PTO/SB/05 (05-03)
Approved for use through 04/30/2003. OMB 0651-0032
U.S. Patent and Trademark Office. U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

00727 U.S.PTO 10/702486 110703

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 C.F.R. 1.53(b))

Attorney Docket No.	000479.00112
First Inventor	Victor Larson
Title	Method For Establishing Secure Communication Link Between Computers Of Virtual Private Network
Express Mail Label No.	

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

ADDRESS TO: Commissioner for Patents
Mail Stop Patent Application
P.O. Box 1450
Alexandria VA 22313-1450

1. ☒ Fee Transmittal Form (e.g., PTO/SB/17)
(Submit an original and a duplicate for fee processing)
2. ☐ Applicant claims small entity status. See 37 CFR 1.27.
3. ☒ Specification [*Total Pages* 83]
(preferred arrangement set forth below)
- Descriptive title of the Invention
- Cross Reference to Related Applications
- Statement Regarding Fed sponsored R & D
- Reference to sequence listing, a table, or a computer program listing appendix
- Background of the Invention
- Brief Summary of the Invention
- Brief Description of the Drawings *(if filed)*
- Detailed Description
- Claim(s)
- Abstract of the Disclosure

4. ☒ Drawing(s) (*35 U.S.C.113*) [*Total Sheets* 40]
☒ Formal ☐ Informal
5. Oath or Declaration [*Total Sheets* 2]
a. ☒ Newly executed (original or copy)
b. ☐ Copy from a prior application (37 CFR 1.63 (d))
(for a continuation/divisional with Box 18 completed)
i. ☐ DELETION OF INVENTOR(S)
Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b).
6. ☒ Application Data Sheet. See 37 CFR 1.76
7. ☐ CD-ROM or CD-R in duplicate, large table or Computer Program *(Appendix)*
8. Nucleotide and/or Amino Acid Sequence Submission *(if applicable, all necessary)*
a. ☐ Computer Readable Form (CRF)
b. Specification Sequence Listing on:
i. ☐ CD-ROM or CD-R (2 copies); or
ii. ☐ paper
c. ☐ Statements verifying identity of above copies

ACCOMPANYING APPLICATIONS PARTS

9. ☒ Assignment Papers (cover sheet & document(s))
10. ☐ 37 C.F.R. 3.73(b) Statement ☐ Power of Attorney *(when there is an assignee)*
11. ☐ English Translation Document *(if applicable)*
12. ☒ Information Disclosure Statement (IDS)/PTO-1449 ☐ Copies of IDS Citations
13. ☐ Preliminary Amendment
14. ☒ Return Receipt Postcard (MPEP 503) *(Should be specifically itemized)*
15. ☐ Certified Copy of Priority Document(s) *(if foreign priority is claimed)*
16. ☐ Nonpublication Request under 35 U.S.C. 122 (b)(2)(B)(i). Applicant must attach form PTO/SB/35 or its equivalent.
17. ☐ Other: _____

18. If a CONTINUING APPLICATION, *check appropriate box, and supply the requisite information below and in a preliminary amendment, or in an Application Data Sheet under 37 CFR 1.76:*

☐ Continuation ☒ Divisional ☐ Continuation-in-part (CIP) of prior application No: 09 / 558,209

Prior application information: *Examiner* Krisna Lim *Art Unit:* 2153

For CONTINUATION or DIVISIONAL APPS only: The entire disclosure of the prior application, from which an oath or declaration is supplied under Box 5b, is considered a part of the disclosure of the accompanying or divisional application and is hereby incorporated by reference. The incorporation can only be relied upon when a portion has been inadvertently omitted from the submitted application parts.

19. CORRESPONDENCE ADDRESS

☒ *Customer Number or Bar Code Label* 22907 *(Insert Customer No. or Attach bar code label here)* *or* ☐ *Correspondence address below*

Name					
Address					
City		*State*		*Zip Code*	
Country		*Telephone*		*Fax*	

Name (Print/Type)	Ross A. Dannenberg	*Registration No. (Attorney/Agent)*	49,024
Signature	Ross Dannenberg	*Date*	November 7, 2003

This collection of information is required by 37 CFR 1.53(b). The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Mail Stop Patent Application, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Approved for use through 07/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

FEE TRANSMITTAL for FY 2004

Effective 10/01/2003. Patent fees are subject to annual revision.

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT ($) 882

Complete if Known	
Application Number	TBA
Filing Date	November 7, 2003
First Named Inventor	Victor Larson
Examiner Name	TBA
Art Unit	TBA
Attorney Docket No.	000479.00112

METHOD OF PAYMENT *(check all that apply)*

☐ Check ☐ Credit card ☐ Money Order ☐ Other ☐ None

☒ Deposit Account:

Deposit Account Number: 19-0733

Deposit Account Name: Banner & Witcoff, LTD.

The Director is authorized to: *(check all that apply)*
☒ Charge fee(s) indicated below ☒ Credit any overpayments
☐ Charge any additional fee(s) during the pendency of this application
☐ Charge fee(s) indicated below, **except for the filing fee** to the above-identified deposit account.

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1001	770	2001	385	Utility filing fee	770
1002	340	2002	170	Design filing fee	
1003	530	2003	265	Plant filing fee	
1004	770	2004	385	Reissue filing fee	
1005	160	2005	80	Provisional filling fee	

SUBTOTAL (1) ($) 770

2. EXTRA CLAIM FEES FOR UTILITY AND REISSUE

			Extra Claims		Fee from below	=	Fee Paid
Total Claims	24	-20 **	4	X	18	=	72
Independent Claims	2	-3 **	0	X		=	0
Multiple Dependent				X		=	0

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
1202	18	2202	9	Claims in excess of 20
1201	86	2201	43	Independent claims in excess of 3
1203	290	2203	145	Multiple dependent claim, if not paid
1204	86	2204	43	** Reissue independent claims over original patent
1205	18	2205	9	** Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($) 72

***or number previously paid, if greater; For Reissues, see above*

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1051	130	2051	65	Surcharge - late filing fee or oath	
1052	50	2052	25	Surcharge - late provisional filing fee or cover sheet.	
1053	130	1053	130	Non-English specification	
1812	2,520	1812	2,520	For filing a request for reexamination	
1804	920*	1804	920*	Requesting publication of SIR prior to Examiner action	
1805	1,840*	1805	1,840*	Requesting publication of SIR after Examiner action	
1251	110	2251	55	Extension for reply within first month	
1252	420	2252	210	Extension for reply within second month	
1253	950	2253	475	Extension for reply within third month	
1254	1,480	2254	740	Extension for reply within fourth month	
1255	2,010	2255	1,005	Extension for reply within fifth month	
1401	330	2401	165	Notice of Appeal	
1402	330	2402	165	Filing a brief in support of an appeal	
1403	290	2403	145	Request for oral hearing	
1451	1,510	1451	1,510	Petition to institute a public use proceeding	
1452	110	2452	55	Petition to revive – unavoidable	
1453	1,330	2453	665	Petition to revive – unintentional	
1501	1,330	2501	665	Utility issue fee (or reissue)	
1502	480	2502	240	Design issue fee	
1503	640	2503	320	Plant issue fee	
1460	130	1460	130	Petitions to the Commissioner	
1807	50	1807	50	Processing fee under 37 CFR 1.17 (q)	
1806	180	1806	180	Submission of Information Disclosure Stmt	
8021	40	8021	40	Recording each patent assignment per property (times number of properties)	40
1809	770	2809	385	Filing a submission after final rejection (37 CFR § 1.129(a))	
1810	770	2810	385	For each additional invention to be examined (37 CFR § 1.129(b))	
1801	770	2801	385	Request for Continued Examination (RCE)	
1802	900	1802	900	Request for expedited examination of a design application	

Other fee (specify) ____

*Reduced by Basic Filing Fee Paid SUBTOTAL (3) ($) 40

SUBMITTED BY				Complete *(if applicable)*	
Name (Print/Type)	Ross A. Dannenberg	*Registration No. (Attorney/Agent)*	49,024	*Telephone*	(202) 824-3153
Signature				*Date*	November 7, 2003

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

PTO/SB/05 (05-03)
Approved for use through 04/30/2003. OMB 0651-0032
U.S. Patent and Trademark Office. U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

00727 U.S.PTO 10/702486 110703

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 C.F.R. 1.53(b))

Attorney Docket No.	000479.00112
First Inventor	Victor Larson
Title	Method For Establishing Secure Communication Link Between Computers Of Virtual Private Network
Express Mail Label No.	

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

ADDRESS TO: Commissioner for Patents
Mail Stop Patent Application
P.O. Box 1450
Alexandria VA 22313-1450

1. ☒ Fee Transmittal Form (e.g., PTO/SB/17)
(Submit an original and a duplicate for fee processing)
2. ☐ Applicant claims small entity status. See 37 CFR 1.27.
3. ☒ Specification [*Total Pages* 83]
(preferred arrangement set forth below)
- Descriptive title of the Invention
- Cross Reference to Related Applications
- Statement Regarding Fed sponsored R & D
- Reference to sequence listing, a table, or a computer program listing appendix
- Background of the Invention
- Brief Summary of the Invention
- Brief Description of the Drawings *(if filed)*
- Detailed Description
- Claim(s)
- Abstract of the Disclosure

4. ☒ Drawing(s) (*35 U.S.C.113*) [*Total Sheets* 40]
☒ Formal ☐ Informal
5. Oath or Declaration [*Total Sheets* 2]
a. ☒ Newly executed (original or copy)
b. ☐ Copy from a prior application (37 CFR 1.63 (d))
(for a continuation/divisional with Box 18 completed)
i. ☐ DELETION OF INVENTOR(S)
Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b).
6. ☒ Application Data Sheet. See 37 CFR 1.76
7. ☐ CD-ROM or CD-R in duplicate, large table or Computer Program *(Appendix)*
8. Nucleotide and/or Amino Acid Sequence Submission *(if applicable, all necessary)*
a. ☐ Computer Readable Form (CRF)
b. Specification Sequence Listing on:
i. ☐ CD-ROM or CD-R (2 copies); or
ii. ☐ paper
c. ☐ Statements verifying identity of above copies

ACCOMPANYING APPLICATIONS PARTS

9. ☒ Assignment Papers (cover sheet & document(s))
10. ☐ 37 C.F.R. 3.73(b) Statement ☐ Power of Attorney *(when there is an assignee)*
11. ☐ English Translation Document *(if applicable)*
12. ☒ Information Disclosure Statement (IDS)/PTO-1449 ☐ Copies of IDS Citations
13. ☐ Preliminary Amendment
14. ☒ Return Receipt Postcard (MPEP 503) *(Should be specifically itemized)*
15. ☐ Certified Copy of Priority Document(s) *(if foreign priority is claimed)*
16. ☐ Nonpublication Request under 35 U.S.C. 122 (b)(2)(B)(i). Applicant must attach form PTO/SB/35 or its equivalent.
17. ☐ Other: _____

18. If a CONTINUING APPLICATION, *check appropriate box, and supply the requisite information below and in a preliminary amendment, or in an Application Data Sheet under 37 CFR 1.76:*

☐ Continuation ☒ Divisional ☐ Continuation-in-part (CIP) of prior application No: 09 / 558,209

Prior application information: *Examiner* Krisna Lim *Art Unit:* 2153

For CONTINUATION or DIVISIONAL APPS only: The entire disclosure of the prior application, from which an oath or declaration is supplied under Box 5b, is considered a part of the disclosure of the accompanying or divisional application and is hereby incorporated by reference. The incorporation can only be relied upon when a portion has been inadvertently omitted from the submitted application parts.

19. CORRESPONDENCE ADDRESS

☒ *Customer Number or Bar Code Label* 22907 *(Insert Customer No. or Attach bar code label here)* *or* ☐ *Correspondence address below*

Name					
Address					
City		*State*		*Zip Code*	
Country		*Telephone*		*Fax*	

Name (Print/Type)	Ross A. Dannenberg	*Registration No. (Attorney/Agent)*	49,024
Signature	Ross Dannenberg	*Date*	November 7, 2003

This collection of information is required by 37 CFR 1.53(b). The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Mail Stop Patent Application, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

PTO/SB/17 (10-03)
Approved for use through 07/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

FEE TRANSMITTAL for FY 2004

Effective 10/01/2003. Patent fees are subject to annual revision.

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT ($) 882

Complete if Known	
Application Number	TBA
Filing Date	November 7, 2003
First Named Inventor	Victor Larson
Examiner Name	TBA
Art Unit	TBA
Attorney Docket No.	000479.00112

METHOD OF PAYMENT *(check all that apply)*

☐ Check ☐ Credit card ☐ Money Order ☐ Other ☐ None

☒ Deposit Account:

Deposit Account Number: 19-0733

Deposit Account Name: Banner & Witcoff, LTD.

The Director is authorized to: *(check all that apply)*

☒ Charge fee(s) indicated below ☒ Credit any overpayments

☐ Charge any additional fee(s) during the pendency of this application

☐ Charge fee(s) indicated below, **except for the filing fee** to the above-identified deposit account.

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1001	770	2001	385	Utility filing fee	770
1002	340	2002	170	Design filing fee	
1003	530	2003	265	Plant filing fee	
1004	770	2004	385	Reissue filing fee	
1005	160	2005	80	Provisional filling fee	

SUBTOTAL (1) ($) 770

2. EXTRA CLAIM FEES FOR UTILITY AND REISSUE

			Extra Claims		Fee from below	=	Fee Paid
Total Claims	24	-20 **	4	X	18	=	72
Independent Claims	2	-3 **	0	X		=	0
Multiple Dependent				X		=	0

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
1202	18	2202	9	Claims in excess of 20
1201	86	2201	43	Independent claims in excess of 3
1203	290	2203	145	Multiple dependent claim, if not paid
1204	86	2204	43	** Reissue independent claims over original patent
1205	18	2205	9	** Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($) 72

***or number previously paid, if greater; For Reissues, see above*

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1051	130	2051	65	Surcharge - late filing fee or oath	
1052	50	2052	25	Surcharge - late provisional filing fee or cover sheet.	
1053	130	1053	130	Non-English specification	
1812	2,520	1812	2,520	For filing a request for reexamination	
1804	920*	1804	920*	Requesting publication of SIR prior to Examiner action	
1805	1,840*	1805	1,840*	Requesting publication of SIR after Examiner action	
1251	110	2251	55	Extension for reply within first month	
1252	420	2252	210	Extension for reply within second month	
1253	950	2253	475	Extension for reply within third month	
1254	1,480	2254	740	Extension for reply within fourth month	
1255	2,010	2255	1,005	Extension for reply within fifth month	
1401	330	2401	165	Notice of Appeal	
1402	330	2402	165	Filing a brief in support of an appeal	
1403	290	2403	145	Request for oral hearing	
1451	1,510	1451	1,510	Petition to institute a public use proceeding	
1452	110	2452	55	Petition to revive – unavoidable	
1453	1,330	2453	665	Petition to revive – unintentional	
1501	1,330	2501	665	Utility issue fee (or reissue)	
1502	480	2502	240	Design issue fee	
1503	640	2503	320	Plant issue fee	
1460	130	1460	130	Petitions to the Commissioner	
1807	50	1807	50	Processing fee under 37 CFR 1.17 (q)	
1806	180	1806	180	Submission of Information Disclosure Stmt	
8021	40	8021	40	Recording each patent assignment per property (times number of properties)	40
1809	770	2809	385	Filing a submission after final rejection (37 CFR § 1.129(a))	
1810	770	2810	385	For each additional invention to be examined (37 CFR § 1.129(b))	
1801	770	2801	385	Request for Continued Examination (RCE)	
1802	900	1802	900	Request for expedited examination of a design application	

Other fee (specify) ____

*Reduced by Basic Filing Fee Paid

SUBTOTAL (3) ($) 40

SUBMITTED BY				Complete *(if applicable)*	
Name (Print/Type)	Ross A. Dannenberg	*Registration No. (Attorney/Agent)*	49,024	*Telephone*	(202) 824-3153
Signature				*Date*	November 7, 2003

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

This collection of information is required by 37 CFR 1.17 and 1.27. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing this form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN COMPUTERS OF VIRTUAL PRIVATE NETWORK

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims priority from and is a divisional patent application of co-pending U.S. application serial number 09/558,209, filed April 26, 2000, which is a continuation-in-part patent application of previously-filed U.S. application serial number 09/504,783, filed on February 15, 2000, now U.S. Pat. No. 6,502,135, issued December 31, 2002, which claims priority from and is a continuation-in-part patent application of previously-filed U.S. application serial number 09/429,643, filed on October 29, 1999. The subject matter of U.S. application serial number 09/429,643, which is bodily incorporated herein, derives from provisional U.S. application numbers 60/106,261 (filed October 30, 1998) and 60/137,704 (filed June 7, 1999). The present application is also related to U.S. application serial number 09/558,210, filed April 26, 2000, and which is incorporated by reference herein.

BACKGROUND OF THE INVENTION

A tremendous variety of methods have been proposed and implemented to provide security and anonymity for communications over the Internet. The variety stems, in part, from the different needs of different Internet users. A basic heuristic framework to aid in discussing these different security techniques is illustrated in FIG. 1. Two terminals, an originating terminal 100 and a destination terminal 110 are in communication over the Internet. It is desired for the communications to be secure, that is, immune to eavesdropping. For example, terminal 100 may transmit secret information to terminal 110 over the Internet 107. Also, it may be desired to prevent an eavesdropper from discovering that terminal 100 is in communication with terminal 110. For example, if terminal 100 is a user and terminal 110 hosts a web site, terminal 100's user may not want anyone in the intervening networks to know what web sites he is "visiting." Anonymity would thus be an issue, for example, for companies that want to keep their market research interests private and thus would prefer to prevent outsiders from knowing which web-sites or other Internet resources they are "visiting." These two security issues may be called data security and anonymity, respectively.

Data security is usually tackled using some form of data encryption. An encryption key 48 is known at both the originating and terminating terminals 100 and 110. The keys may be

private and public at the originating and destination terminals 100 and 110, respectively or they may be symmetrical keys (the same key is used by both parties to encrypt and decrypt). Many encryption methods are known and usable in this context.

To hide traffic from a local administrator or ISP, a user can employ a local proxy server in communicating over an encrypted channel with an outside proxy such that the local administrator or ISP only sees the encrypted traffic. Proxy servers prevent destination servers from determining the identities of the originating clients. This system employs an intermediate server interposed between client and destination server. The destination server sees only the Internet Protocol (IP) address of the proxy server and not the originating client. The target server only sees the address of the outside proxy. This scheme relies on a trusted outside proxy server. Also, proxy schemes are vulnerable to traffic analysis methods of determining identities of transmitters and receivers. Another important limitation of proxy servers is that the server knows the identities of both calling and called parties. In many instances, an originating terminal, such as terminal A, would prefer to keep its identity concealed from the proxy, for example, if the proxy server is provided by an Internet service provider (ISP).

To defeat traffic analysis, a scheme called Chaum's mixes employs a proxy server that transmits and receives fixed length messages, including dummy messages. Multiple originating terminals are connected through a mix (a server) to multiple target servers. It is difficult to tell which of the originating terminals are communicating to which of the connected target servers, and the dummy messages confuse eavesdroppers' efforts to detect communicating pairs by analyzing traffic. A drawback is that there is a risk that the mix server could be compromised. One way to deal with this risk is to spread the trust among multiple mixes. If one mix is compromised, the identities of the originating and target terminals may remain concealed. This strategy requires a number of alternative mixes so that the intermediate servers interposed between the originating and target terminals are not determinable except by compromising more than one mix. The strategy wraps the message with multiple layers of encrypted addresses. The first mix in a sequence can decrypt only the outer layer of the message to reveal the next destination mix in sequence. The second mix can decrypt the message to reveal the next mix and so on. The target server receives the message and, optionally, a multi-layer encrypted payload containing return information to send data back in the same fashion. The only way to defeat such

a mix scheme is to collude among mixes. If the packets are all fixed-length and intermixed with dummy packets, there is no way to do any kind of traffic analysis.

Still another anonymity technique, called 'crowds,' protects the identity of the originating terminal from the intermediate proxies by providing that originating terminals belong to groups of proxies called crowds. The crowd proxies are interposed between originating and target terminals. Each proxy through which the message is sent is randomly chosen by an upstream proxy. Each intermediate proxy can send the message either to another randomly chosen proxy in the "crowd" or to the destination. Thus, even crowd members cannot determine if a preceding proxy is the originator of the message or if it was simply passed from another proxy.

ZKS (Zero-Knowledge Systems) Anonymous IP Protocol allows users to select up to any of five different pseudonyms, while desktop software encrypts outgoing traffic and wraps it in User Datagram Protocol (UDP) packets. The first server in a 2+-hop system gets the UDP packets, strips off one l ayer of encryption to add another, then sends the traffic to the next server, which strips off yet another layer of encryption and adds a new one. The user is permitted to control the number of hops. At the final server, traffic is decrypted with an untraceable IP address. The technique is called onion-routing. This method can be defeated using traffic analysis. For a simple example, bursts of packets from a user during low-duty periods can reveal the identities of sender and receiver.

Firewalls attempt to protect LANs from unauthorized access and hostile exploitation or damage to computers connected to the LAN. Firewalls provide a server through which all access to the LAN must pass. Firewalls are centralized systems that require administrative overhead to maintain. They can be compromised by virtual-machine applications ("applets"). They instill a false sense of security that leads to security breaches for example by users sending sensitive information to servers outside the firewall or encouraging use of modems to sidestep the firewall security. Firewalls are not useful for distributed systems such as business travelers, extranets, small teams, etc.

SUMMARY OF THE INVENTION

A secure mechanism for communicating over the internet, including a protocol referred to as the Tunneled Agile Routing Protocol (TARP), uses a unique two-layer encryption format and special TARP routers. TARP routers are similar in function to regular IP routers. Each TARP router has one or more IP addresses and uses normal IP protocol to send IP packet

messages ("packets" or "datagrams"). The IP packets exchanged between TARP terminals via TARP routers are actually encrypted packets whose true destination address is concealed except to TARP routers and servers. The normal or "clear" or "outside" IP header attached to TARP IP packets contains only the address of a next hop router or destination server. That is, instead of indicating a final destination in the destination field of the IP header, the TARP packet's IP header always points to a next-hop in a series of TARP router hops, or to the final destination. This means there is no overt indication from an intercepted TARP packet of the true destination of the TARP packet since the destination could always be next-hop TARP router as well as the final destination.

Each TARP packet's true destination is concealed behind a layer of encryption generated using a link key. The link key is the encryption key used for encrypted communication between the hops intervening between an originating TARP terminal and a destination TARP terminal. Each TARP router can remove the outer layer of encryption to reveal the destination router for each TARP packet. To identify the link key needed to decrypt the outer layer of encryption of a TARP packet, a receiving TARP or routing terminal may identify the transmitting terminal by the sender/receiver IP numbers in the cleartext IP header.

Once the outer layer of encryption is removed, the TARP router determines the final destination. Each TARP packet 140 undergoes a minimum number of hops to help foil traffic analysis. The hops may be chosen at random or by a fixed value. As a result, each TARP packet may make random trips among a number of geographically disparate routers before reaching its destination. Each trip is highly likely to be different for each packet composing a given message because each trip is independently randomly determined. This feature is called *agile routing*. The fact that different packets take different routes provides distinct advantages by making it difficult for an interloper to obtain all the packets forming an entire multi-packet message. The associated advantages have to do with the inner layer of encryption discussed below. Agile routing is combined with another feature that furthers this purpose; a feature that ensures that any message is broken into multiple packets.

The IP address of a TARP router can be changed, a feature called *IP agility*. Each TARP router, independently or under direction from another TARP terminal or router, can change its IP address. A separate, unchangeable identifier or address is also defined. This address, called the TARP address, is known only to TARP routers and terminals and may be correlated at any time

by a TARP router or a TARP terminal using a Lookup Table (LUT). When a TARP router or terminal changes its IP address, it updates the other TARP routers and terminals which in turn update their respective LUTs.

The message payload is hidden behind an inner layer of encryption in the TARP packet that can only be unlocked using a session key. The session key is not available to any of the intervening TARP routers. The session key is used to decrypt the payloads of the TARP packets permitting the data stream to be reconstructed.

Communication may be made private using link and session keys, which in turn may be shared and used according to any desired method. For example, public/private keys or symmetric keys may be used.

To transmit a data stream, a TARP originating terminal constructs a series of TARP packets from a series of IP packets generated by a network (IP) layer process. (Note that the terms "network layer," "data link layer," "application layer," etc. used in this specification correspond to the Open Systems Interconnection (OSI) network terminology.) The payloads of these packets are assembled into a block and chain-block encrypted using the session key. This assumes, of course, that all the IP packets are destined for the same TARP terminal. The block is then interleaved and the interleaved encrypted block is broken into a series of payloads, one for each TARP packet to be generated. Special TARP headers IP_T are then added to each payload using the IP headers from the data stream packets. The TARP headers can be identical to normal IP headers or customized in some way. They should contain a formula or data for deinterleaving the data at the destination TARP terminal, a time-to-live (TTL) parameter to indicate the number of hops still to be executed, a data type identifier which indicates whether the payload contains, for example, TCP or UDP data, the sender's TARP address, the destination TARP address, and an indicator as to whether the packet contains real or decoy data or a formula for filtering out decoy data if decoy data is spread in some way through the TARP payload data.

Note that although chain-block encryption is discussed here with reference to the session key, any encryption method may be used. Preferably, as in chain block encryption, a method should be used that makes unauthorized decryption difficult without an entire result of the encryption process. Thus, by separating the encrypted block among multiple packets and making it difficult for an interloper to obtain access to all of such packets, the contents of the communications are provided an extra layer of security.

Decoy or dummy data can be added to a stream to help foil traffic analysis by reducing the peak-to-average network load. It may be desirable to provide the TARP process with an ability to respond to the time of day or other criteria to generate more decoy data during low traffic periods so that communication bursts at one point in the Internet cannot be tied to communication bursts at another point to reveal the communicating endpoints.

Dummy data also helps to break the data into a larger number of inconspicuously-sized packets permitting the interleave window size to be increased while maintaining a reasonable size for each packet. (The packet size can be a single standard size or selected from a fixed range of sizes.) One primary reason for desiring for each message to be broken into multiple packets is apparent if a chain block encryption scheme is used to form the first encryption layer prior to interleaving. A single block encryption may be applied to portion, or entirety, of a message, and that portion or entirety then interleaved into a number of separate packets. Considering the agile IP routing of the packets, and the attendant difficulty of reconstructing an entire sequence of packets to form a single block-encrypted message element, decoy packets can significantly increase the difficulty of reconstructing an entire data stream.

The above scheme may be implemented entirely by processes operating between the data link layer and the network layer of each server or terminal participating in the TARP system. Because the encryption system described above is insertable between the data link and network layers, the processes involved in supporting the encrypted communication may be completely transparent to processes at the IP (network) layer and above. The TARP processes may also be completely transparent to the data link layer processes as well. Thus, no operations at or above the Network layer, or at or below the data link layer, are affected by the insertion of the TARP stack. This provides additional security to all processes at or above the network layer, since the difficulty of unauthorized penetration of the network layer (by, for example, a hacker) is increased substantially. Even newly developed servers running at the session layer leave all processes below the session layer vulnerable to attack. Note that in this architecture, security is distributed. That is, notebook computers used by executives on the road, for example, can communicate over the Internet without any compromise in security.

IP address changes made by TARP terminals and routers can be done at regular intervals, at random intervals, or upon detection of "attacks." The variation of IP addresses hinders traffic analysis that might reveal which computers are communicating, and also provides a degree of

immunity from attack. The level of immunity from attack is roughly proportional to the rate at which the IP address of the host is changing.

As mentioned, IP addresses may be changed in response to attacks. An attack may be revealed, for example, by a regular series of messages indicating that a router is being probed in some way. Upon detection of an attack, the TARP layer process may respond to this event by changing its IP address. In addition, it may create a subprocess that maintains the original IP address and continues interacting with the attacker in some manner.

Decoy packets may be generated by each TARP terminal on some basis determined by an algorithm. For example, the algorithm may be a random one which calls for the generation of a packet on a random basis when the terminal is idle. Alternatively, the algorithm may be responsive to time of day or detection of low traffic to generate more decoy packets during low traffic times. Note that packets are preferably generated in groups, rather than one by one, the groups being sized to simulate real messages. In addition, so that decoy packets may be inserted in normal TARP message streams, the background loop may have a latch that makes it more likely to insert decoy packets when a message stream is being received. Alternatively, if a large number of decoy packets is received along with regular TARP packets, the algorithm may increase the rate of dropping of decoy packets rather than forwarding them. The result of dropping and generating decoy packets in this way is to make the apparent incoming message size different from the apparent outgoing message size to help foil traffic analysis.

In various other embodiments of the invention, a scalable version of the system may be constructed in which a plurality of IP addresses are preassigned to each pair of communicating nodes in the network. Each pair of nodes agrees upon an algorithm for "hopping" between IP addresses (both sending and receiving), such that an eavesdropper sees apparently continuously random IP address pairs (source and destination) for packets transmitted between the pair. Overlapping or "reusable" IP addresses may be allocated to different users on the same subnet, since each node merely verifies that a particular packet includes a valid source/destination pair from the agreed-upon algorithm. Source/destination pairs are preferably not reused between any two nodes during any given end-to-end session, though limited IP block sizes or lengthy sessions might require it.

Further improvements described in this continuation-in-part application include: (1) a load balancer that distributes packets across different transmission paths according to

transmission path quality; (2) a DNS proxy server that transparently creates a virtual private network in response to a domain name inquiry; (3) a large-to-small link bandwidth management feature that prevents denial-of-service attacks at system chokepoints; (4) a traffic limiter that regulates incoming packets by limiting the rate at which a transmitter can be synchronized with a receiver; and (5) a signaling synchronizer that allows a large number of nodes to communicate with a central node by partitioning the communication function between two separate entities

The present invention provides key technologies for implementing a secure virtual Internet by using a new agile network protocol that is built on top of the existing Internet protocol (IP). The secure virtual Internet works over the existing Internet infrastructure, and interfaces with client applications the same way as the existing Internet. The key technologies provided by the present invention that support the secure virtual Internet include a "one-click" and "no-click" technique to become part of the secure virtual Internet, a secure domain name service (SDNS) for the secure virtual Internet, and a new approach for interfacing specific client applications onto the secure virtual Internet. According to the invention, the secure domain name service interfaces with existing applications, in addition to providing a way to register and serve domain names and addresses.

According to one aspect of the present invention, a user can conveniently establish a VPN using a "one-click" or a "no-click" technique without being required to enter user identification information, a password and/or an encryption key for establishing a VPN. The advantages of the present invention are provided by a method for establishing a secure communication link between a first computer and a second computer over a computer network, such as the Internet. In one embodiment, a secure communication mode is enabled at a first computer without a user entering any cryptographic information for establishing the secure communication mode of communication, preferably by merely selecting an icon displayed on the first computer. Alternatively, the secure communication mode of communication can be enabled by entering a command into the first computer. Then, a secure communication link is established between the first computer and a second computer over a computer network based on the enabled secure communication mode of communication. According to the invention, it is determined whether a secure communication software module is stored on the first computer in response to the step of enabling the secure communication mode of communication. A predetermined computer network address is then accessed for loading the secure communication

software module when the software module is not stored on the first computer. Subsequently, the proxy software module is stored in the first computer. The secure communication link is a virtual private network communication link over the computer network. Preferably, the virtual private network can be based on inserting into each data packet one or more data values that vary according to a pseudo-random sequence. Alternatively, the virtual private network can be based on a computer network address hopping regime that is used to pseudorandomly change computer network addresses or other data values in packets transmitted between the first computer and the second computer, such that the second computer compares the data values in each data packet transmitted between the first computer and the second computer to a moving window of valid values. Yet another alternative provides that the virtual private network can be based on a comparison between a discriminator field in each data packet to a table of valid discriminator fields maintained for the first computer.

According to another aspect of the invention, a command is entered to define a setup parameter associated with the secure communication link mode of communication. Consequently, the secure communication mode is automatically established when a communication link is established over the computer network.

The present invention also provides a computer system having a communication link to a computer network, and a display showing a hyperlink for establishing a virtual private network through the computer network. When the hyperlink for establishing the virtual private network is selected, a virtual private network is established over the computer network. A non-standard top-level domain name is then sent over the virtual private network communication to a predetermined computer network address, such as a computer network address for a secure domain name service (SDNS).

The present invention provides a domain name service that provides secure computer network addresses for secure, non-standard top-level domain names. The advantages of the present invention are provided by a secure domain name service for a computer network that includes a portal connected to a computer network, such as the Internet, and a domain name database connected to the computer network through the portal. According to the invention, the portal authenticates a query for a secure computer network address, and the domain name database stores secure computer network addresses for the computer network. Each secure

computer network address is based on a non-standard top-level domain name, such as .scom, .sorg, .snet, .snet, .sedu, .smil and .sint.

The present invention provides a way to encapsulate existing application network traffic at the application layer of a client computer so that the client application can securely communicate with a server protected by an agile network protocol. The advantages of the present invention are provided by a method for communicating using a private communication link between a client computer and a server computer over a computer network, such as the Internet. According to the invention, an information packet is sent from the client computer to the server computer over the computer network. The information packet contains data that is inserted into the payload portion of the packet at the application layer of the client computer and is used for forming a virtual private connection between the client computer and the server computer. The modified information packet can be sent through a firewall before being sent over the computer network to the server computer and by working on top of existing protocols (i.e., UDP, ICMP and TCP), the present invention more easily penetrates the firewall. The information packet is received at a kernel layer of an operating system on the server side. It is then determined at the kernel layer of the operating system on the host computer whether the information packet contains the data that is used for forming the virtual private connection. The server side replies by sending an information packet to the client computer that has been modified at the kernel layer to containing virtual private connection information in the payload portion of the reply information packet. Preferably, the information packet from the client computer and the reply information packet from the server side are each a UDP protocol information packet. Alternative, both information packets could be a TCP/IP protocol information packet, or an ICMP protocol information packet.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is an illustration of secure communications over the Internet according to a prior art embodiment.

FIG. 2 is an illustration of secure communications over the Internet according to a an embodiment of the invention.

FIG. 3a is an illustration of a process of forming a tunneled IP packet according to an embodiment of the invention.

FIG. 3b is an illustration of a process of forming a tunneled IP packet according to another embodiment of the invention.

FIG. 4 is an illustration of an OSI layer location of processes that may be used to implement the invention.

FIG. 5 is a flow chart illustrating a process for routing a tunneled packet according to an embodiment of the invention.

FIG. 6 is a flow chart illustrating a process for forming a tunneled packet according to an embodiment of the invention.

FIG. 7 is a flow chart illustrating a process for receiving a tunneled packet according to an embodiment of the invention.

FIG. 8 shows how a secure session is established and synchronized between a client and a TARP router.

FIG. 9 shows an IP address hopping scheme between a client computer and TARP router using transmit and receive tables in each computer.

FIG. 10 shows physical link redundancy among three Internet Service Providers (ISPs) and a client computer.

FIG. 11 shows how multiple IP packets can be embedded into a single "frame" such as an Ethernet frame, and further shows the use of a discriminator field to camouflage true packet recipients.

FIG. 12A shows a system that employs hopped hardware addresses, hopped IP addresses, and hopped discriminator fields.

FIG. 12B shows several different approaches for hopping hardware addresses, IP addresses, and discriminator fields in combination.

FIG. 13 shows a technique for automatically re-establishing synchronization between sender and receiver through the use of a partially public sync value.

FIG. 14 shows a "checkpoint" scheme for regaining synchronization between a sender and recipient.

FIG. 15 shows further details of the checkpoint scheme of FIG. 14.

FIG. 16 shows how two addresses can be decomposed into a plurality of segments for comparison with presence vectors.

FIG. 17 shows a storage array for a receiver's active addresses.

FIG. 18 shows the receiver's storage array after receiving a sync request.

FIG. 19 shows the receiver's storage array after new addresses have been generated.

FIG. 20 shows a system employing distributed transmission paths.

FIG. 21 shows a plurality of link transmission tables that can be used to route packets in the system of FIG. 20.

FIG. 22A shows a flowchart for adjusting weight value distributions associated with a plurality of transmission links.

FIG. 22B shows a flowchart for setting a weight value to zero if a transmitter turns off.

FIG. 23 shows a system employing distributed transmission paths with adjusted weight value distributions for each path.

FIG. 24 shows an example using the system of FIG. 23.

FIG. 25 shows a conventional domain-name look-up service.

FIG. 26 shows a system employing a DNS proxy server with transparent VPN creation.

FIG. 27 shows steps that can be carried out to implement transparent VPN creation based on a DNS look-up function.

FIG. 28 shows a system including a link guard function that prevents packet overloading on a low-bandwidth link LOW BW.

FIG. 29 shows one embodiment of a system employing the principles of FIG. 28.

FIG. 30 shows a system that regulates packet transmission rates by throttling the rate at which synchronizations are performed.

FIG. 31 shows a signaling server 3101 and a transport server 3102 used to establish a VPN with a client computer.

FIG. 32 shows message flows relating to synchronization protocols of FIG. 31.

FIG. 33 shows a system block diagram of a computer network in which the "one-click" secure communication link of the present invention is suitable for use.

FIG. 34 shows a flow diagram for installing and establishing a "one-click" secure communication link over a computer network according to the present invention.

FIG. 35 shows a flow diagram for registering a secure domain name according to the present invention.

FIG. 36 shows a system block diagram of a computer network in which a private connection according to the present invention can be configured to more easily traverse a firewall between two computer networks.

FIG. 37 shows a flow diagram for establishing a virtual private connection that is encapsulated using an existing network protocol.

DETAILED DESCRIPTION OF THE INVENTION

Referring to FIG. 2, a secure mechanism for communicating over the internet employs a number of special routers or servers, called TARP routers 122-127 that are similar to regular IP routers 128-132 in that each has one or more IP addresses and uses normal IP protocol to send normal-looking IP packet messages, called TARP packets 140. TARP packets 140 are identical to normal IP packet messages that are routed by regular IP routers 128-132 because each TARP packet 140 contains a destination address as in a normal IP packet. However, instead of indicating a final destination in the destination field of the IP header, the TARP packet's 140 IP header always points to a next-hop in a series of TARP router hops, or the final destination, TARP terminal 110. Because the header of the TARP packet contains only the next-hop destination, there is no overt indication from an intercepted TARP packet of the true destination of the TARP packet 140 since the destination could always be the next-hop TARP router as well as the final destination, TARP terminal 110.

Each TARP packet's true destination is concealed behind an outer layer of encryption generated using a link key 146. The link key 146 is the encryption key used for encrypted communication between the end points (TARP terminals or TARP routers) of a single link in the chain of hops connecting the originating TARP terminal 100 and the destination TARP terminal 110. Each TARP router 122-127, using the link key 146 it uses to communicate with the previous hop in a chain, can use the link key to reveal the true destination of a TARP packet. To identify the link key needed to decrypt the outer layer of encryption of a TARP packet, a receiving TARP or routing terminal may identify the transmitting terminal (which may indicate the link key used) by the sender field of the clear IP header. Alternatively, this identity may be hidden behind another layer of encryption in available bits in the clear IP header. Each TARP router, upon receiving a TARP message, determines if the message is a TARP message by using authentication data in the TARP packet. This could be recorded in available bytes in the TARP packet's IP header. Alternatively, TARP packets could be authenticated by attempting to decrypt

using the link key 146 and determining if the results are as expected. The former may have computational advantages because it does not involve a decryption process.

Once the outer layer of decryption is completed by a TARP router 122-127, the TARP router determines the final destination. The system is preferably designed to cause each TARP packet 140 to undergo a minimum number of hops to help foil traffic analysis. The time to live counter in the IP header of the TARP message may be used to indicate a number of TARP router hops yet to be completed. Each TARP router then would decrement the counter and determine from that whether it should forward the TARP packet 140 to another TARP router 122-127 or to the destination TARP terminal 110. If the time to live counter is zero or below zero after decrementing, for an example of usage, the TARP router receiving the TARP packet 140 may forward the TARP packet 140 to the destination TARP terminal 110. If the time to live counter is above zero after decrementing, for an example of usage, the TARP router receiving the TARP packet 140 may forward the TARP packet 140 to a TARP router 122-127 that the current TARP terminal chooses at random. As a result, each TARP packet 140 is routed through some minimum number of hops of TARP routers 122-127 which are chosen at random.

Thus, each TARP packet, irrespective of the traditional factors determining traffic in the Internet, makes random trips among a number of geographically disparate routers before reaching its destination and each trip is highly likely to be different for each packet composing a given message because each trip is independently randomly determined as described above. This feature is called *agile routing*. For reasons that will become clear shortly, the fact that different packets take different routes provides distinct advantages by making it difficult for an interloper to obtain all the packets forming an entire multi-packet message. Agile routing is combined with another feature that furthers this purpose, a feature that ensures that any message is broken into multiple packets.

A TARP router receives a TARP packet when an IP address used by the TARP router coincides with the IP address in the TARP packet's IP header IP_C. The IP address of a TARP router, however, may not remain constant. To avoid and manage attacks, each TARP router, independently or under direction from another TARP terminal or router, may change its IP address. A separate, unchangeable identifier or address is also defined. This address, called the TARP address, is known only to TARP routers and terminals and may be correlated at any time by a TARP router or a TARP terminal using a Lookup Table (LUT). When a TARP router or

terminal changes its IP address, it updates the other TARP routers and terminals which in turn update their respective LUTs. In reality, whenever a TARP router looks up the address of a destination in the encrypted header, it must convert a TARP address to a real IP address using its LUT.

While every TARP router receiving a TARP packet has the ability to determine the packet's final destination, the message payload is embedded behind an inner layer of encryption in the TARP packet that can only be unlocked using a session key. The session key is not available to any of the TARP routers 122-127 intervening between the originating 100 and destination 110 TARP terminals. The session key is used to decrypt the payloads of the TARP packets 140 permitting an entire message to be reconstructed.

In one embodiment, communication may be made private using link and session keys, which in turn may be shared and used according any desired method. For example, a public key or symmetric keys may be communicated between link or session endpoints using a public key method. Any of a variety of other mechanisms for securing data to ensure that only authorized computers can have access to the private information in the TARP packets 140 may be used as desired.

Referring to FIG. 3a, to construct a series of TARP packets, a data stream 300 of IP packets 207a, 207b, 207c, etc., such series of packets being formed by a network (IP) layer process, is broken into a series of small sized segments. In the present example, equal-sized segments 1-9 are defined and used to construct a set of interleaved data packets A, B, and C. Here it is assumed that the number of interleaved packets A, B, and C formed is three and that the number of IP packets 207a-207c used to form the three interleaved packets A, B, and C is exactly three. Of course, the number of IP packets spread over a group of interleaved packets may be any convenient number as may be the number of interleaved packets over which the incoming data stream is spread. The latter, the number of interleaved packets over which the data stream is spread, is called the *interleave window.*

To create a packet, the transmitting software interleaves the normal IP packets 207a *et. seq.* to form a new set of interleaved payload data 320. This payload data 320 is then encrypted using a session key to form a set of session-key-encrypted payload data 330, each of which, A, B, and C, will form the payload of a TARP packet. Using the IP header data, from the original packets 207a-207c, new TARP headers IP_T are formed. The TARP headers IP_T can be identical

to normal IP headers or customized in some way. In a preferred embodiment, the TARP headers IP_T are IP headers with added data providing the following information required for routing and reconstruction of messages, some of which data is ordinarily, or capable of being, contained in normal IP headers:

1. A window sequence number – an identifier that indicates where the packet belongs in the original message sequence.
2. An interleave sequence number – an identifier that indicates the interleaving sequence used to form the packet so that the packet can be deinterleaved along with other packets in the interleave window.
3. A time-to-live (TTL) datum – indicates the number of TARP-router-hops to be executed before the packet reaches its destination. Note that the TTL parameter may provide a datum to be used in a probabilistic formula for determining whether to route the packet to the destination or to another hop.
4. Data type identifier – indicates whether the payload contains, for example, TCP or UDP data.
5. Sender's address – indicates the sender's address in the TARP network.
6. Destination address – indicates the destination terminal's address in the TARP network.
7. Decoy/Real – an indicator of whether the packet contains real message data or dummy decoy data or a combination.

Obviously, the packets going into a single interleave window must include only packets with a common destination. Thus, it is assumed in the depicted example that the IP headers of IP packets 207a-207c all contain the same destination address or at least will be received by the same terminal so that they can be deinterleaved. Note that dummy or decoy data or packets can be added to form a larger interleave window than would otherwise be required by the size of a given message. Decoy or dummy data can be added to a stream to help foil traffic analysis by leveling the load on the network. Thus, it may be desirable to provide the TARP process with an ability to respond to the time of day or other criteria to generate more decoy data during low traffic periods so that communication bursts at one point in the Internet cannot be tied to communication bursts at another point to reveal the communicating endpoints.

Dummy data also helps to break the data into a larger number of inconspicuously-sized packets permitting the interleave window size to be increased while maintaining a reasonable size for each packet. (The packet size can be a single standard size or selected from a fixed range of sizes.) One primary reason for desiring for each message to be broken into multiple packets is apparent if a chain block encryption scheme is used to form the first encryption layer prior to interleaving. A single block encryption may be applied to a portion, or the entirety, of a message, and that portion or entirety then interleaved into a number of separate packets.

Referring to FIG. 3b, in an alternative mode of TARP packet construction, a series of IP packets are accumulated to make up a predefined interleave window. The payloads of the packets are used to construct a single block 520 for chain block encryption using the session key. The payloads used to form the block are presumed to be destined for the same terminal. The block size may coincide with the interleave window as depicted in the example embodiment of FIG. 3b. After encryption, the encrypted block is broken into separate payloads and segments which are interleaved as in the embodiment of Fig 3a. The resulting interleaved packets A, B, and C, are then packaged as TARP packets with TARP headers as in the Example of FIG. 3a. The remaining process is as shown in, and discussed with reference to, FIG. 3a.

Once the TARP packets 340 are formed, each entire TARP packet 340, including the TARP header IP_T, is encrypted using the link key for communication with the first-hop-TARP router. The first hop TARP router is randomly chosen. A final unencrypted IP header IP_C is added to each encrypted TARP packet 340 to form a normal IP packet 360 that can be transmitted to a TARP router. Note that the process of constructing the TARP packet 360 does not have to be done in stages as described. The above description is just a useful heuristic for describing the final product, namely, the TARP packet.

Note that, TARP header IP_T could be a completely custom header configuration with no similarity to a normal IP header except that it contain the information identified above. This is so since this header is interpreted by only TARP routers.

The above scheme may be implemented entirely by processes operating between the data link layer and the network layer of each server or terminal participating in the TARP system. Referring to FIG. 4, a TARP transceiver 405 can be an originating terminal 100, a destination terminal 110, or a TARP router 122-127. In each TARP Transceiver 405, a transmitting process is generated to receive normal packets from the Network (IP) layer and generate TARP packets

for communication over the network. A receiving process is generated to receive normal IP packets containing TARP packets and generate from these normal IP packets which are "passed up" to the Network (IP) layer. Note that where the TARP Transceiver 405 is a router, the received TARP packets 140 are not processed into a stream of IP packets 415 because they need only be authenticated as proper TARP packets and then passed to another TARP router or a TARP destination terminal 110. The intervening process, a "TARP Layer" 420, could be combined with either the data link layer 430 or the Network layer 410. In either case, it would intervene between the data link layer 430 so that the process would receive regular IP packets containing embedded TARP packets and "hand up" a series of reassembled IP packets to the Network layer 410. As an example of combining the TARP layer 420 with the data link layer 430, a program may augment the normal processes running a communications card, for example, an Ethernet card. Alternatively, the TARP layer processes may form part of a dynamically loadable module that is loaded and executed to support communications between the network and data link layers.

Because the encryption system described above can be inserted between the data link and network layers, the processes involved in supporting the encrypted communication may be completely transparent to processes at the IP (network) layer and above. The TARP processes may also be completely transparent to the data link layer processes as well. Thus, no operations at or above the network layer, or at or below the data link layer, are affected by the insertion of the TARP stack. This provides additional security to all processes at or above the network layer, since the difficulty of unauthorized penetration of the network layer (by, for example, a hacker) is increased substantially. Even newly developed servers running at the session layer leave all processes below the session layer vulnerable to attack. Note that in this architecture, security is distributed. That is, notebook computers used by executives on the road, for example, can communicate over the Internet without any compromise in security.

Note that IP address changes made by TARP terminals and routers can be done at regular intervals, at random intervals, or upon detection of "attacks." The variation of IP addresses hinders traffic analysis that might reveal which computers are communicating, and also provides a degree of immunity from attack. The level of immunity from attack is roughly proportional to the rate at which the IP address of the host is changing.

As mentioned, IP addresses may be changed in response to attacks. An attack may be revealed, for example, by a regular series of messages indicates that a router is being probed in some way. Upon detection of an attack, the TARP layer process may respond to this event by changing its IP address. To accomplish this, the TARP process will construct a TARP-formatted message, in the style of Internet Control Message Protocol (ICMP) datagrams as an example; this message will contain the machine's TARP address, its previous IP address, and its new IP address. The TARP layer will transmit this packet to at least one known TARP router; then upon receipt and validation of the message, the TARP router will update its LUT with the new IP address for the stated TARP address. The TARP router will then format a similar message, and broadcast it to the other TARP routers so that they may update their LUTs. Since the total number of TARP routers on any given subnet is expected to be relatively small, this process of updating the LUTs should be relatively fast. It may not, however, work as well when there is a relatively large number of TARP routers and/or a relatively large number of clients; this has motivated a refinement of this architecture to provide scalability; this refinement has led to a second embodiment, which is discussed below.

Upon detection of an attack, the TARP process may also create a subprocess that maintains the original IP address and continues interacting with the attacker. The latter may provide an opportunity to trace the attacker or study the attacker's methods (called "fishbowling" drawing upon the analogy of a small fish in a fish bowl that "thinks" it is in the ocean but is actually under captive observation). A history of the communication between the attacker and the abandoned (fishbowled) IP address can be recorded or transmitted for human analysis or further synthesized for purposes of responding in some way.

As mentioned above, decoy or dummy data or packets can be added to outgoing data streams by TARP terminals or routers. In addition to making it convenient to spread data over a larger number of separate packets, such decoy packets can also help to level the load on inactive portions of the Internet to help foil traffic analysis efforts.

Decoy packets may be generated by each TARP terminal 100, 110 or each router 122-127 on some basis determined by an algorithm. For example, the algorithm may be a random one which calls for the generation of a packet on a random basis when the terminal is idle. Alternatively, the algorithm may be responsive to time of day or detection of low traffic to generate more decoy packets during low traffic times. Note that packets are preferably generated

in groups, rather than one by one, the groups being sized to simulate real messages. In addition, so that decoy packets may be inserted in normal TARP message streams, the background loop may have a latch that makes it more likely to insert decoy packets when a message stream is being received. That is, when a series of messages are received, the decoy packet generation rate may be increased. Alternatively, if a large number of decoy packets is received along with regular TARP packets, the algorithm may increase the rate of dropping of decoy packets rather than forwarding them. The result of dropping and generating decoy packets in this way is to make the apparent incoming message size different from the apparent outgoing message size to help foil traffic analysis. The rate of reception of packets, decoy or otherwise, may be indicated to the decoy packet dropping and generating processes through perishable decoy and regular packet counters. (A perishable counter is one that resets or decrements its value in response to time so that it contains a high value when it is incremented in rapid succession and a small value when incremented either slowly or a small number of times in rapid succession.) Note that destination TARP terminal 110 may generate decoy packets equal in number and size to those TARP packets received to make it appear it is merely routing packets and is therefore not the destination terminal.

Referring to FIG. 5, the following particular steps may be employed in the above-described method for routing TARP packets.

- S0. A background loop operation is performed which applies an algorithm which determines the generation of decoy IP packets. The loop is interrupted when an encrypted TARP packet is received.
- S2. The TARP packet may be probed in some way to authenticate the packet before attempting to decrypt it using the link key. That is, the router may determine that the packet is an authentic TARP packet by performing a selected operation on some data included with the clear IP header attached to the encrypted TARP packet contained in the payload. This makes it possible to avoid performing decryption on packets that are not authentic TARP packets.
- S3. The TARP packet is decrypted to expose the destination TARP address and an indication of whether the packet is a decoy packet or part of a real message.
- S4. If the packet is a decoy packet, the perishable decoy counter is incremented.

- S5. Based on the decoy generation/dropping algorithm and the perishable decoy counter value, if the packet is a decoy packet, the router may choose to throw it away. If the received packet is a decoy packet and it is determined that it should be thrown away (S6), control returns to step S0.
- S7. The TTL parameter of the TARP header is decremented and it is determined if the TTL parameter is greater than zero.
- S8. If the TTL parameter is greater than zero, a TARP address is randomly chosen from a list of TARP addresses maintained by the router and the link key and IP address corresponding to that TARP address memorized for use in creating a new IP packet containing the TARP packet.
- S9. If the TTL parameter is zero or less, the link key and IP address corresponding to the TARP address of the destination are memorized for use in creating the new IP packet containing the TARP packet.
- S10. The TARP packet is encrypted using the memorized link key.
- S11. An IP header is added to the packet that contains the stored IP address, the encrypted TARP packet wrapped with an IP header, and the completed packet transmitted to the next hop or destination.

Referring to FIG. 6, the following particular steps may be employed in the above-described method for generating TARP packets.

- S20. A background loop operation applies an algorithm that determines the generation of decoy IP packets. The loop is interrupted when a data stream containing IP packets is received for transmission.
- S21. The received IP packets are grouped into a set consisting of messages with a constant IP destination address. The set is further broken down to coincide with a maximum size of an interleave window The set is encrypted, and interleaved into a set of payloads destined to become TARP packets.
- S22. The TARP address corresponding to the IP address is determined from a lookup table and stored to generate the TARP header. An initial TTL count is generated and stored in the

header. The TTL count may be random with minimum and maximum values or it may be fixed or determined by some other parameter.

- S23. The window sequence numbers and interleave sequence numbers are recorded in the TARP headers of each packet.
- S24. One TARP router address is randomly chosen for each TARP packet and the IP address corresponding to it stored for use in the clear IP header. The link key corresponding to this router is identified and used to encrypt TARP packets containing interleaved and encrypted data and TARP headers.
- S25. A clear IP header with the first hop router's real IP address is generated and added to each of the encrypted TARP packets and the resulting packets.

Referring to FIG. 7, the following particular steps may be employed in the above-described method for receiving TARP packets.

- S40. A background loop operation is performed which applies an algorithm which determines the generation of decoy IP packets. The loop is interrupted when an encrypted TARP packet is received.
- S42. The TARP packet may be probed to authenticate the packet before attempting to decrypt it using the link key.
- S43. The TARP packet is decrypted with the appropriate link key to expose the destination TARP address and an indication of whether the packet is a decoy packet or part of a real message.
- S44. If the packet is a decoy packet, the perishable decoy counter is incremented.
- S45. Based on the decoy generation/dropping algorithm and the perishable decoy counter value, if the packet is a decoy packet, the receiver may choose to throw it away.
- S46. The TARP packets are cached until all packets forming an interleave window are received.
- S47. Once all packets of an interleave window are received, the packets are deinterleaved.
- S48. The packets block of combined packets defining the interleave window is then decrypted using the session key.

- S49. The decrypted block is then divided using the window sequence data and the IP_T headers are converted into normal IP_C headers. The window sequence numbers are integrated in the IP_C headers.
- S50. The packets are then handed up to the IP layer processes.

1. SCALABILITY ENHANCEMENTS

The IP agility feature described above relies on the ability to transmit IP address changes to all TARP routers. The embodiments including this feature will be referred to as "boutique" embodiments due to potential limitations in scaling these features up for a large network, such as the Internet. (The "boutique" embodiments would, however, be robust for use in smaller networks, such as small virtual private networks, for example). One problem with the boutique embodiments is that if IP address changes are to occur frequently, the message traffic required to update all routers sufficiently quickly creates a serious burden on the Internet when the TARP router and/or client population gets large. The bandwidth burden added to the networks, for example in ICMP packets, that would be used to update all the TARP routers could overwhelm the Internet for a large scale implementation that approached the scale of the Internet. In other words, the boutique system's scalability is limited.

A system can be constructed which trades some of the features of the above embodiments to provide the benefits of IP agility without the additional messaging burden. This is accomplished by IP address-hopping according to shared algorithms that govern IP addresses used between links participating in communications sessions between nodes such as TARP nodes. (Note that the IP hopping technique is also applicable to the boutique embodiment.) The IP agility feature discussed with respect to the boutique system can be modified so that it becomes decentralized under this scalable regime and governed by the above-described shared algorithm. Other features of the boutique system may be combined with this new type of IP-agility.

The new embodiment has the advantage of providing IP agility governed by a local algorithm and set of IP addresses exchanged by each communicating pair of nodes. This local governance is session-independent in that it may govern communications between a pair of nodes, irrespective of the session or end points being transferred between the directly communicating pair of nodes.

In the scalable embodiments, blocks of IP addresses are allocated to each node in the network. (This scalability will increase in the future, when Internet Protocol addresses are increased to 128-bit fields, vastly increasing the number of distinctly addressable nodes). Each node can thus use any of the IP addresses assigned to that node to communicate with other nodes in the network. Indeed, each pair of communicating nodes can use a plurality of source IP addresses and destination IP addresses for communicating with each other.

Each communicating pair of nodes in a chain participating in any session stores two blocks of IP addresses, called netblocks, and an algorithm and randomization seed for selecting, from each netblock, the next pair of source/destination IP addresses that will be used to transmit the next message. In other words, the algorithm governs the sequential selection of IP-address pairs, one sender and one receiver IP address, from each netblock. The combination of algorithm, seed, and netblock (IP address block) will be called a "hopblock." A router issues separate transmit and receive hopblocks to its clients. The send address and the receive address of the IP header of each outgoing packet sent by the client are filled with the send and receive IP addresses generated by the algorithm. The algorithm is "clocked" (indexed) by a counter so that each time a pair is used, the algorithm turns out a new transmit pair for the next packet to be sent.

The router's receive hopblock is identical to the client's transmit hopblock. The router uses the receive hopblock to predict what the send and receive IP address pair for the next expected packet from that client will be. Since packets can be received out of order, it is not possible for the router to predict with certainty what IP address pair will be on the next sequential packet. To account for this problem, the router generates a range of predictions encompassing the number of possible transmitted packet send/receive addresses, of which the next packet received could leap ahead. Thus, if there is a vanishingly small probability that a given packet will arrive at the router ahead of 5 packets transmitted by the client before the given packet, then the router can generate a series of 6 send/receive IP address pairs (or "hop window") to compare with the next received packet. When a packet is received, it is marked in the hop window as such, so that a second packet with the same IP address pair will be discarded. If an out-of-sequence packet does not arrive within a predetermined timeout period, it can be requested for retransmission or simply discarded from the receive table, depending upon the protocol in use for that communications session, or possibly by convention.

When the router receives the client's packet, it compares the send and receive IP addresses of the packet with the next N predicted send and receive IP address pairs and rejects the packet if it is not a member of this set. Received packets that do not have the predicted source/destination IP addresses falling with the window are rejected, thus thwarting possible hackers. (With the number of possible combinations, even a fairly large window would be hard to fall into at random.) If it is a member of this set, the router accepts the packet and processes it further. This link-based IP-hopping strategy, referred to as "IHOP," is a network element that stands on its own and is not necessarily accompanied by elements of the boutique system described above. If the routing agility feature described in connection with the boutique embodiment is combined with this link-based IP-hopping strategy, the router's next step would be to decrypt the TARP header to determine the destination TARP router for the packet and determine what should be the next hop for the packet. The TARP router would then forward the packet to a random TARP router or the destination TARP router with which the source TARP router has a link-based IP hopping communication established.

Figure 8 shows how a client computer 801 and a TARP router 811 can establish a secure session. When client 801 seeks to establish an IHOP session with TARP router 811, the client 801 sends "secure synchronization" request ("SSYN") packet 821 to the TARP router 811. This SYN packet 821 contains the client's 801 authentication token, and may be sent to the router 811 in an encrypted format. The source and destination IP numbers on the packet 821 are the client's 801 current fixed IP address, and a "known" fixed IP address for the router 811. (For security purposes, it may be desirable to reject any packets from outside of the local network that are destined for the router's known fixed IP address.) Upon receipt and validation of the client's 801 SSYN packet 821, the router 811 responds by sending an encrypted "secure synchronization acknowledgment" ("SSYN ACK") 822 to the client 801. This SSYN ACK 822 will contain the transmit and receive hopblocks that the client 801 will use when communicating with the TARP router 811. The client 801 will acknowledge the TARP router's 811 response packet 822 by generating an encrypted SSYN ACK ACK packet 823 which will be sent from the client's 801 fixed IP address and to the TARP router's 811 known fixed IP address. The client 801 will simultaneously generate a SSYN ACK ACK packet; this SSYN ACK packet, referred to as the Secure Session Initiation (SSI) packet 824, will be sent with the first {sender, receiver} IP pair in the client's transmit table 921 (FIG. 9), as specified in the transmit hopblock provided by the

TARP router 811 in the SSYN ACK packet 822. The TARP router 811 will respond to the SSI packet 824 with an SSI ACK packet 825, which will be sent with the first {sender, receiver} IP pair in the TARP router's transmit table 923. Once these packets have been successfully exchanged, the secure communications session is established, and all further secure communications between the client 801 and the TARP router 811 will be conducted via this secure session, as long as synchronization is maintained. If synchronization is lost, then the client 801 and TARP router 802 may re-establish the secure session by the procedure outlined in Figure 8 and described above.

While the secure session is active, both the client 901 and TARP router 911 (FIG. 9) will maintain their respective transmit tables 921, 923 and receive tables 922, 924, as provided by the TARP router during session synchronization 822. It is important that the sequence of IP pairs in the client's transmit table 921 be identical to those in the TARP router's receive table 924; similarly, the sequence of IP pairs in the client's receive table 922 must be identical to those in the router's transmit table 923. This is required for the session synchronization to be maintained. The client 901 need maintain only one transmit table 921 and one receive table 922 during the course of the secure session. Each sequential packet sent by the client 901 will employ the next {send, receive} IP address pair in the transmit table, regardless of TCP or UDP session. The TARP router 911 will expect each packet arriving from the client 901 to bear the next IP address pair shown in its receive table.

Since packets can arrive out of order, however, the router 911 can maintain a "look ahead" buffer in its receive table, and will mark previously-received IP pairs as invalid for future packets; any future packet containing an IP pair that is in the look-ahead buffer but is marked as previously received will be discarded. Communications from the TARP router 911 to the client 901 are maintained in an identical manner; in particular, the router 911 will select the next IP address pair from its transmit table 923 when constructing a packet to send to the client 901, and the client 901 will maintain a look-ahead buffer of expected IP pairs on packets that it is receiving. Each TARP router will maintain separate pairs of transmit and receive tables for each client that is currently engaged in a secure session with or through that TARP router.

While clients receive their hopblocks from the first server linking them to the Internet, routers exchange hopblocks. When a router establishes a link-based IP-hopping communication regime with another router, each router of the pair exchanges its transmit hopblock. The transmit

hopblock of each router becomes the receive hopblock of the other router. The communication between routers is governed as described by the example of a client sending a packet to the first router.

While the above strategy works fine in the IP milieu, many local networks that are connected to the Internet are Ethernet systems. In Ethernet, the IP addresses of the destination devices must be translated into hardware addresses, and vice versa, using known processes ("address resolution protocol," and "reverse address resolution protocol"). However, if the link-based IP-hopping strategy is employed, the correlation process would become explosive and burdensome. An alternative to the link-based IP hopping strategy may be employed within an Ethernet network. The solution is to provide that the node linking the Internet to the Ethernet (call it the border node) use the link-based IP-hopping communication regime to communicate with nodes outside the Ethernet LAN. Within the Ethernet LAN, each TARP node would have a single IP address which would be addressed in the conventional way. Instead of comparing the {sender, receiver} IP address pairs to authenticate a packet, the intra-LAN TARP node would use one of the IP header extension fields to do so. Thus, the border node uses an algorithm shared by the intra-LAN TARP node to generate a symbol that is stored in the free field in the IP header, and the intra-LAN TARP node generates a range of symbols based on its prediction of the next expected packet to be received from that particular source IP address. The packet is rejected if it does not fall into the set of predicted symbols (for example, numerical values) or is accepted if it does. Communications from the intra-LAN TARP node to the border node are accomplished in the same manner, though the algorithm will necessarily be different for security reasons. Thus, each of the communicating nodes will generate transmit and receive tables in a similar manner to that of Figure 9; the intra-LAN TARP nodes transmit table will be identical to the border node's receive table, and the intra-LAN TARP node's receive table will be identical to the border node's transmit table.

The algorithm used for IP address-hopping can be any desired algorithm. For example, the algorithm can be a given pseudo-random number generator that generates numbers of the range covering the allowed IP addresses with a given seed. Alternatively, the session participants can assume a certain type of algorithm and specify simply a parameter for applying the algorithm. For example the assumed algorithm could be a particular pseudo-random number generator and the session participants could simply exchange seed values.

Note that there is no permanent physical distinction between the originating and destination terminal nodes. Either device at either end point can initiate a synchronization of the pair. Note also that the authentication/synchronization-request (and acknowledgment) and hopblock-exchange may all be served by a single message so that separate message exchanges may not be required.

As another extension to the stated architecture, multiple physical paths can be used by a client, in order to provide link redundancy and further thwart attempts at denial of service and traffic monitoring. As shown in Figure 10, for example, client 1001 can establish three simultaneous sessions with each of three TARP routers provided by different ISPs 1011, 1012, 1013. As an example, the client 1001 can use three different telephone lines 1021, 1022, 1023 to connect to the ISPs, or two telephone lines and a cable modem, etc. In this scheme, transmitted packets will be sent in a random fashion among the different physical paths. This architecture provides a high degree of communications redundancy, with improved immunity from denial-of-service attacks and traffic monitoring.

2. FURTHER EXTENSIONS

The following describes various extensions to the techniques, systems, and methods described above. As described above, the security of communications occurring between computers in a computer network (such as the Internet, an Ethernet, or others) can be enhanced by using seemingly random source and destination Internet Protocol (IP) addresses for data packets transmitted over the network. This feature prevents eavesdroppers from determining which computers in the network are communicating with each other while permitting the two communicating computers to easily recognize whether a given received data packet is legitimate or not. In one embodiment of the above-described systems, an IP header extension field is used to authenticate incoming packets on an Ethernet.

Various extensions to the previously described techniques described herein include: (1) use of hopped hardware or "MAC" addresses in broadcast type network; (2) a self-synchronization technique that permits a computer to automatically regain synchronization with a sender; (3) synchronization algorithms that allow transmitting and receiving computers to quickly re-establish synchronization in the event of lost packets or other events; and (4) a fast-packet rejection mechanism for rejecting invalid packets. Any or all of these extensions can be combined with the features described above in any of various ways.

A. Hardware Address Hopping

Internet protocol-based communications techniques on a LAN—or across any dedicated physical medium—typically embed the IP packets within lower-level packets, often referred to as "frames." As shown in FIG. 11, for example, a first Ethernet frame 1150 comprises a frame header 1101 and two embedded IP packets IP1 and IP2, while a second Ethernet frame 1160 comprises a different frame header 1104 and a single IP packet IP3. Each frame header generally includes a source hardware address 1101A and a destination hardware address 1101B; other well-known fields in frame headers are omitted from FIG. 11 for clarity. Two hardware nodes communicating over a physical communication channel insert appropriate source and destination hardware addresses to indicate which nodes on the channel or network should receive the frame.

It may be possible for a nefarious listener to acquire information about the contents of a frame and/or its communicants by examining frames on a local network rather than (or in addition to) the IP packets themselves. This is especially true in broadcast media, such as Ethernet, where it is necessary to insert into the frame header the hardware address of the machine that generated the frame and the hardware address of the machine to which frame is being sent. All nodes on the network can potentially "see" all packets transmitted across the network. This can be a problem for secure communications, especially in cases where the communicants do not want for any third party to be able to identify who is engaging in the information exchange. One way to address this problem is to push the address-hopping scheme down to the hardware layer. In accordance with various embodiments of the invention, hardware addresses are "hopped" in a manner similar to that used to change IP addresses, such that a listener cannot determine which hardware node generated a particular message nor which node is the intended recipient.

FIG. 12A shows a system in which Media Access Control ("MAC") hardware addresses are "hopped" in order to increase security over a network such as an Ethernet. While the description refers to the exemplary case of an Ethernet environment, the inventive principles are equally applicable to other types of communications media. In the Ethernet case, the MAC address of the sender and receiver are inserted into the Ethernet frame and can be observed by anyone on the LAN who is within the broadcast range for that frame. For secure

communications, it becomes desirable to generate frames with MAC addresses that are not attributable to any specific sender or receiver.

As shown in FIG. 12A, two computer nodes 1201 and 1202 communicate over a communication channel such as an Ethernet. Each node executes one or more application programs 1203 and 1218 that communicate by transmitting packets through communication software 1204 and 1217, respectively. Examples of application programs include video conferencing, e-mail, word processing programs, telephony, and the like. Communication software 1204 and 1217 can comprise, for example, an OSI layered architecture or "stack" that standardizes various services provided at different levels of functionality.

The lowest levels of communication software 1204 and 1217 communicate with hardware components 1206 and 1214 respectively, each of which can include one or more registers 1207 and 1215 that allow the hardware to be reconfigured or controlled in accordance with various communication protocols. The hardware components (an Ethernet network interface card, for example) communicate with each other over the communication medium. Each hardware component is typically pre-assigned a fixed hardware address or MAC number that identifies the hardware component to other nodes on the network. One or more interface drivers control the operation of each card and can, for example, be configured to accept or reject packets from certain hardware addresses. As will be described in more detail below, various embodiments of the inventive principles provide for "hopping" different addresses using one or more algorithms and one or more moving windows that track a range of valid addresses to validate received packets. Packets transmitted according to one or more of the inventive principles will be generally referred to as "secure" packets or "secure communications" to differentiate them from ordinary data packets that are transmitted in the clear using ordinary, machine-correlated addresses.

One straightforward method of generating non-attributable MAC addresses is an extension of the IP hopping scheme. In this scenario, two machines on the same LAN that desire to communicate in a secure fashion exchange random-number generators and seeds, and create sequences of quasi-random MAC addresses for synchronized hopping. The implementation and synchronization issues are then similar to that of IP hopping.

This approach, however, runs the risk of using MAC addresses that are currently active on the LAN—which, in turn, could interrupt communications for those machines. Since an

Ethernet MAC address is at present 48 bits in length, the chance of randomly misusing an active MAC address is actually quite small. However, if that figure is multiplied by a large number of nodes (as would be found on an extensive LAN), by a large number of frames (as might be the case with packet voice or streaming video), and by a large number of concurrent Virtual Private Networks (VPNs), then the chance that a non-secure machine's MAC address could be used in an address-hopped frame can become non-trivial. In short, any scheme that runs even a small risk of interrupting communications for other machines on the LAN is bound to receive resistance from prospective system administrators. Nevertheless, it is technically feasible, and can be implemented without risk on a LAN on which there is a small number of machines, or if all of the machines on the LAN are engaging in MAC-hopped communications.

Synchronized MAC address hopping may incur some overhead in the course of session establishment, especially if there are multiple sessions or multiple nodes involved in the communications. A simpler method of randomizing MAC addresses is to allow each node to receive and process *every* incident frame on the network. Typically, each network interface driver will check the destination MAC address in the header of every incident frame to see if it matches that machine's MAC address; if there is no match, then the frame is discarded. In one embodiment, however, these checks can be disabled, and every incident packet is passed to the TARP stack for processing. This will be referred to as "promiscuous" mode, since every incident frame is processed. Promiscuous mode allows the sender to use completely random, unsynchronized MAC addresses, since the destination machine is guaranteed to process the frame. The decision as to whether the packet was truly intended for that machine is handled by the TARP stack, which checks the source and destination IP addresses for a match in its IP synchronization tables. If no match is found, the packet is discarded; if there is a match, the packet is unwrapped, the inner header is evaluated, and if the inner header indicates that the packet is destined for that machine then the packet is forwarded to the IP stack—otherwise it is discarded.

One disadvantage of purely-random MAC address hopping is its impact on processing overhead; that is, since every incident frame must be processed, the machine's CPU is engaged considerably more often than if the network interface driver is discriminating and rejecting packets unilaterally. A compromise approach is to select either a single fixed MAC address or a small number of MAC addresses (e.g., one for each virtual private network on an Ethernet) to

use for MAC-hopped communications, regardless of the actual recipient for which the message is intended. In this mode, the network interface driver can check each incident frame against one (or a few) pre-established MAC addresses, thereby freeing the CPU from the task of physical-layer packet discrimination. This scheme does not betray any useful information to an interloper on the LAN; in particular, every secure packet can already be identified by a unique packet type in the outer header. However, since all machines engaged in secure communications would either be using the same MAC address, or be selecting from a small pool of predetermined MAC addresses, the association between a specific machine and a specific MAC address is effectively broken.

In this scheme, the CPU will be engaged more often than it would be in non-secure communications (or in synchronized MAC address hopping), since the network interface driver cannot always unilaterally discriminate between secure packets that are destined for that machine, and secure packets from other VPNs. However, the non-secure traffic is easily eliminated at the network interface, thereby reducing the amount of processing required of the CPU. There are boundary conditions where these statements would not hold, of course—e.g., if *all* of the traffic on the LAN is secure traffic, then the CPU would be engaged to the same degree as it is in the purely-random address hopping case; alternatively, if each VPN on the LAN uses a different MAC address, then the network interface can perfectly discriminate secure frames destined for the local machine from those constituting other VPNs. These are engineering tradeoffs that might be best handled by providing administrative options for the users when installing the software and/or establishing VPNs.

Even in this scenario, however, there still remains a slight risk of selecting MAC addresses that are being used by one or more nodes on the LAN. One solution to this problem is to formally assign one address or a range of addresses for use in MAC-hopped communications. This is typically done via an assigned numbers registration authority; e.g., in the case of Ethernet, MAC address ranges are assigned to vendors by the Institute of Electrical and Electronics Engineers (IEEE). A formally-assigned range of addresses would ensure that secure frames do not conflict with any properly-configured and properly-functioning machines on the LAN.

Reference will now be made to FIGS. 12A and 12B in order to describe the many combinations and features that follow the inventive principles. As explained above, two

computer nodes 1201 and 1202 are assumed to be communicating over a network or communication medium such as an Ethernet. A communication protocol in each node (1204 and 1217, respectively) contains a modified element 1205 and 1216 that performs certain functions that deviate from the standard communication protocols. In particular, computer node 1201 implements a first "hop" algorithm 1208X that selects seemingly random source and destination IP addresses (and, in one embodiment, seemingly random IP header discriminator fields) in order to transmit each packet to the other computer node. For example, node 1201 maintains a transmit table 1208 containing triplets of source (S), destination (D), and discriminator fields (DS) that are inserted into outgoing IP packet headers. The table is generated through the use of an appropriate algorithm (e.g., a random number generator that is seeded with an appropriate seed) that is known to the recipient node 1202. As each new IP packet is formed, the next sequential entry out of the sender's transmit table 1208 is used to populate the IP source, IP destination, and IP header extension field (e.g., discriminator field). It will be appreciated that the transmit table need not be created in advance but could instead be created on-the-fly by executing the algorithm when each packet is formed.

At the receiving node 1202, the same IP hop algorithm 1222X is maintained and used to generate a receive table 1222 that lists valid triplets of source IP address, destination IP address, and discriminator field. This is shown by virtue of the first five entries of transmit table 1208 matching the second five entries of receive table 1222. (The tables may be slightly offset at any particular time due to lost packets, misordered packets, or transmission delays). Additionally, node 1202 maintains a receive window W3 that represents a list of valid IP source, IP destination, and discriminator fields that will be accepted when received as part of an incoming IP packet. As packets are received, window W3 slides down the list of valid entries, such that the possible valid entries change over time. Two packets that arrive out of order but are nevertheless matched to entries within window W3 will be accepted; those falling outside of window W3 will be rejected as invalid. The length of window W3 can be adjusted as necessary to reflect network delays or other factors.

Node 1202 maintains a similar transmit table 1221 for creating IP packets and frames destined for node 1201 using a potentially different hopping algorithm 1221X, and node 1201 maintains a matching receive table 1209 using the same algorithm 1209X. As node 1202 transmits packets to node 1201 using seemingly random IP source, IP destination, and/or

discriminator fields, node 1201 matches the incoming packet values to those falling within window W1 maintained in its receive table. In effect, transmit table 1208 of node 1201 is synchronized (i.e., entries are selected in the same order) to receive table 1222 of receiving node 1202. Similarly, transmit table 1221 of node 1202 is synchronized to receive table 1209 of node 1201. It will be appreciated that although a common algorithm is shown for the source, destination and discriminator fields in FIG. 12A (using, e.g., a different seed for each of the three fields), an entirely different algorithm could in fact be used to establish values for each of these fields. It will also be appreciated that one or two of the fields can be "hopped" rather than all three as illustrated.

In accordance with another aspect of the invention, hardware or "MAC" addresses are hopped instead of or in addition to IP addresses and/or the discriminator field in order to improve security in a local area or broadcast-type network. To that end, node 1201 further maintains a transmit table 1210 using a transmit algorithm 1210X to generate source and destination hardware addresses that are inserted into frame headers (e.g., fields 1101A and 1101B in FIG. 11) that are synchronized to a corresponding receive table 1224 at node 1202. Similarly, node 1202 maintains a different transmit table 1223 containing source and destination hardware addresses that is synchronized with a corresponding receive table 1211 at node 1201. In this manner, outgoing hardware frames appear to be originating from and going to completely random nodes on the network, even though each recipient can determine whether a given packet is intended for it or not. It will be appreciated that the hardware hopping feature can be implemented at a different level in the communications protocol than the IP hopping feature (e.g., in a card driver or in a hardware card itself to improve performance).

FIG. 12B shows three different embodiments or modes that can be employed using the aforementioned principles. In a first mode referred to as "promiscuous" mode, a common hardware address (e.g., a fixed address for source and another for destination) or else a completely random hardware address is used by all nodes on the network, such that a particular packet cannot be attributed to any one node. Each node must initially accept all packets containing the common (or random) hardware address and inspect the IP addresses or discriminator field to determine whether the packet is intended for that node. In this regard, either the IP addresses or the discriminator field or both can be varied in accordance with an algorithm as described above. As explained previously, this may increase each node's overhead

since additional processing is involved to determine whether a given packet has valid source and destination hardware addresses.

In a second mode referred to as "promiscuous per VPN" mode, a small set of fixed hardware addresses are used, with a fixed source/destination hardware address used for all nodes communicating over a virtual private network. For example, if there are six nodes on an Ethernet, and the network is to be split up into two private virtual networks such that nodes on one VPN can communicate with only the other two nodes on its own VPN, then two sets of hardware addresses could be used: one set for the first VPN and a second set for the second VPN. This would reduce the amount of overhead involved in checking for valid frames since only packets arriving from the designated VPN would need to be checked. IP addresses and one or more discriminator fields could still be hopped as before for secure communication within the VPN. Of course, this solution compromises the anonymity of the VPNs (i.e., an outsider can easily tell what traffic belongs in which VPN, though he cannot correlate it to a specific machine/person). It also requires the use of a discriminator field to mitigate the vulnerability to certain types of DoS attacks. (For example, without the discriminator field, an attacker on the LAN could stream frames containing the MAC addresses being used by the VPN; rejecting those frames could lead to excessive processing overhead. The discriminator field would provide a low-overhead means of rejecting the false packets.)

In a third mode referred to as "hardware hopping" mode, hardware addresses are varied as illustrated in FIG. 12A, such that hardware source and destination addresses are changed constantly in order to provide non-attributable addressing. Variations on these embodiments are of course possible, and the invention is not intended to be limited in any respect by these illustrative examples.

B. Extending the Address Space

Address hopping provides security and privacy. However, the level of protection is limited by the number of addresses in the blocks being hopped. A hopblock denotes a field or fields modulated on a packet-wise basis for the purpose of providing a VPN. For instance, if two nodes communicate with IP address hopping using hopblocks of 4 addresses (2 bits) each, there would be 16 possible address-pair combinations. A window of size 16 would result in most address pairs being accepted as valid most of the time. This limitation can be overcome by using a discriminator field in addition to or instead of the hopped address fields. The discriminator

field would be hopped in exactly the same fashion as the address fields and it would be used to determine whether a packet should be processed by a receiver.

Suppose that two clients, each using four-bit hopblocks, would like the same level of protection afforded to clients communicating via IP hopping between two A blocks (24 address bits eligible for hopping). A discriminator field of 20 bits, used in conjunction with the 4 address bits eligible for hopping in the IP address field, provides this level of protection. A 24-bit discriminator field would provide a similar level of protection if the address fields were not hopped or ignored. Using a discriminator field offers the following advantages: (1) an arbitrarily high level of protection can be provided, and (2) address hopping is unnecessary to provide protection. This may be important in environments where address hopping would cause routing problems.

C. Synchronization Techniques

It is generally assumed that once a sending node and receiving node have exchanged algorithms and seeds (or similar information sufficient to generate quasi-random source and destination tables), subsequent communication between the two nodes will proceed smoothly. Realistically, however, two nodes may lose synchronization due to network delays or outages, or other problems. Consequently, it is desirable to provide means for re-establishing synchronization between nodes in a network that have lost synchronization.

One possible technique is to require that each node provide an acknowledgment upon successful receipt of each packet and, if no acknowledgment is received within a certain period of time, to re-send the unacknowledged packet. This approach, however, drives up overhead costs and may be prohibitive in high-throughput environments such as streaming video or audio, for example.

A different approach is to employ an automatic synchronizing technique that will be referred to herein as "self-synchronization." In this approach, synchronization information is embedded into each packet, thereby enabling the receiver to re-synchronize itself upon receipt of a single packet if it determines that is has lost synchronization with the sender. (If communications are already in progress, and the receiver determines that it is still in sync with the sender, then there is no need to re-synchronize.) A receiver could detect that it was out of synchronization by, for example, employing a "dead-man" timer that expires after a certain

period of time, wherein the timer is reset with each valid packet. A time stamp could be hashed into the public sync field (see below) to preclude packet-retry attacks.

In one embodiment, a "sync field" is added to the header of each packet sent out by the sender. This sync field could appear in the clear or as part of an encrypted portion of the packet. Assuming that a sender and receiver have selected a random-number generator (RNG) and seed value, this combination of RNG and seed can be used to generate a random-number sequence (RNS). The RNS is then used to generate a sequence of source/destination IP pairs (and, if desired, discriminator fields and hardware source and destination addresses), as described above. It is not necessary, however, to generate the entire sequence (or the first N-1 values) in order to generate the Nth random number in the sequence; if the sequence index N is known, the random value corresponding to that index can be directly generated (see below). Different RNGs (and seeds) with different fundamental periods could be used to generate the source and destination IP sequences, but the basic concepts would still apply. For the sake of simplicity, the following discussion will assume that IP source and destination address pairs (only) are hopped using a single RNG sequencing mechanism.

In accordance with a "self-synchronization" feature, a sync field in each packet header provides an index (i.e., a sequence number) into the RNS that is being used to generate IP pairs. Plugging this index into the RNG that is being used to generate the RNS yields a specific random number value, which in turn yields a specific IP pair. That is, an IP pair can be generated directly from knowledge of the RNG, seed, and index number; it is not necessary, in this scheme, to generate the entire sequence of random numbers that precede the sequence value associated with the index number provided.

Since the communicants have presumably previously exchanged RNGs and seeds, the only new information that must be provided in order to generate an IP pair is the sequence number. If this number is provided by the sender in the packet header, then the receiver need only plug this number into the RNG in order to generate an IP pair – and thus verify that the IP pair appearing in the header of the packet is valid. In this scheme, if the sender and receiver lose synchronization, the receiver can immediately re-synchronize upon receipt of a single packet by simply comparing the IP pair in the packet header to the IP pair generated from the index number. Thus, synchronized communications can be resumed upon receipt of a single packet, making this scheme ideal for multicast communications. Taken to the extreme, it could obviate

the need for synchronization tables entirely; that is, the sender and receiver could simply rely on the index number in the sync field to validate the IP pair on each packet, and thereby eliminate the tables entirely.

The aforementioned scheme may have some inherent security issues associated with it — namely, the placement of the sync field. If the field is placed in the outer header, then an interloper could observe the values of the field and their relationship to the IP stream. This could potentially compromise the algorithm that is being used to generate the IP-address sequence, which would compromise the security of the communications. If, however, the value is placed in the inner header, then the sender must decrypt the inner header before it can extract the sync value and validate the IP pair; this opens up the receiver to certain types of denial-of-service (DoS) attacks, such as packet replay. That is, if the receiver must decrypt a packet before it can validate the IP pair, then it could potentially be forced to expend a significant amount of processing on decryption if an attacker simply retransmits previously valid packets. Other attack methodologies are possible in this scenario.

A possible compromise between algorithm security and processing speed is to split up the sync value between an inner (encrypted) and outer (unencrypted) header. That is, if the sync value is sufficiently long, it could potentially be split into a rapidly-changing part that can be viewed in the clear, and a fixed (or very slowly changing) part that must be protected. The part that can be viewed in the clear will be called the "public sync" portion and the part that must be protected will be called the "private sync" portion.

Both the public sync and private sync portions are needed to generate the complete sync value. The private portion, however, can be selected such that it is fixed or will change only occasionally. Thus, the private sync value can be stored by the recipient, thereby obviating the need to decrypt the header in order to retrieve it. If the sender and receiver have previously agreed upon the frequency with which the private part of the sync will change, then the receiver can selectively decrypt a single header in order to extract the new private sync if the communications gap that has led to lost synchronization has exceeded the lifetime of the previous private sync. This should not represent a burdensome amount of decryption, and thus should not open up the receiver to denial-of-service attack simply based on the need to occasionally decrypt a single header.

One implementation of this is to use a hashing function with a one-to-one mapping to generate the private and public sync portions from the sync value. This implementation is shown in FIG. 13, where (for example) a first ISP 1302 is the sender and a second ISP 1303 is the receiver. (Other alternatives are possible from FIG. 13.) A transmitted packet comprises a public or "outer" header 1305 that is not encrypted, and a private or "inner" header 1306 that is encrypted using for example a link key. Outer header 1305 includes a public sync portion while inner header 1306 contains the private sync portion. A receiving node decrypts the inner header using a decryption function 1307 in order to extract the private sync portion. This step is necessary only if the lifetime of the currently buffered private sync has expired. (If the currently-buffered private sync is still valid, then it is simply extracted from memory and "added" (which could be an inverse hash) to the public sync, as shown in step 1308.) The public and decrypted private sync portions are combined in function 1308 in order to generate the combined sync 1309. The combined sync (1309) is then fed into the RNG (1310) and compared to the IP address pair (1311) to validate or reject the packet.

An important consideration in this architecture is the concept of "future" and "past" where the public sync values are concerned. Though the sync values, themselves, should be random to prevent spoofing attacks, it may be important that the receiver be able to quickly identify a sync value that has already been sent — even if the packet containing that sync value was never actually received by the receiver. One solution is to hash a time stamp or sequence number into the public sync portion, which could be quickly extracted, checked, and discarded, thereby validating the public sync portion itself.

In one embodiment, packets can be checked by comparing the source/destination IP pair generated by the sync field with the pair appearing in the packet header. If (1) they match, (2) the time stamp is valid, and (3) the dead-man timer has expired, then re-synchronization occurs; otherwise, the packet is rejected. If enough processing power is available, the dead-man timer and synchronization tables can be avoided altogether, and the receiver would simply resynchronize (e.g., validate) on every packet.

The foregoing scheme may require large-integer (e.g., 160-bit) math, which may affect its implementation. Without such large-integer registers, processing throughput would be affected, thus potentially affecting security from a denial-of-service standpoint. Nevertheless, as large-

integer math processing features become more prevalent, the costs of implementing such a feature will be reduced.

D. Other Synchronization Schemes

As explained above, if W or more consecutive packets are lost between a transmitter and receiver in a VPN (where W is the window size), the receiver's window will not have been updated and the transmitter will be transmitting packets not in the receiver's window. The sender and receiver will not recover synchronization until perhaps the random pairs in the window are repeated by chance. Therefore, there is a need to keep a transmitter and receiver in synchronization whenever possible and to re-establish synchronization whenever it is lost.

A "checkpoint" scheme can be used to regain synchronization between a sender and a receiver that have fallen out of synchronization. In this scheme, a checkpoint message comprising a random IP address pair is used for communicating synchronization information. In one embodiment, two messages are used to communicate synchronization information between a sender and a recipient:

1. SYNC_REQ is a message used by the sender to indicate that it wants to synchronize; and
2. SYNC_ACK is a message used by the receiver to inform the transmitter that it has been synchronized.

According to one variation of this approach, both the transmitter and receiver maintain three checkpoints (see FIG. 14):

1. In the transmitter, ckpt_o ("checkpoint old") is the IP pair that was used to re-send the last SYNC_REQ packet to the receiver. In the receiver, ckpt_o ("checkpoint old") is the IP pair that receives repeated SYNC_REQ packets from the transmitter.
2. In the transmitter, ckpt_n ("checkpoint new") is the IP pair that will be used to send the next SYNC_REQ packet to the receiver. In the receiver, ckpt_n ("checkpoint new") is the IP pair that receives a new SYNC_REQ packet from the transmitter and which causes the receiver's window to be re-aligned, ckpt_o set to ckpt_n, a new ckpt_n to be generated and a new ckpt_r to be generated.
3. In the transmitter, ckpt_r is the IP pair that will be used to send the next SYNC_ACK packet to the receiver. In the receiver, ckpt_r is the IP pair that receives a new SYNC_ACK packet from the transmitter and which causes a new ckpt_n to be

generated. Since SYNC_ACK is transmitted from the receiver ISP to the sender ISP, the transmitter ckpt_r refers to the ckpt_r of the receiver and the receiver ckpt_r refers to the ckpt_r of the transmitter (see FIG. 14).

When a transmitter initiates synchronization, the IP pair it will use to transmit the next data packet is set to a predetermined value and when a receiver first receives a SYNC_REQ, the receiver window is updated to be centered on the transmitter's next IP pair. This is the primary mechanism for checkpoint synchronization.

Synchronization can be initiated by a packet counter (e.g., after every N packets transmitted, initiate a synchronization) or by a timer (every S seconds, initiate a synchronization) or a combination of both. See FIG. 15. From the transmitter's perspective, this technique operates as follows: (1) Each transmitter periodically transmits a "sync request" message to the receiver to make sure that it is in sync. (2) If the receiver is still in sync, it sends back a "sync ack" message. (If this works, no further action is necessary). (3) If no "sync ack" has been received within a period of time, the transmitter retransmits the sync request again. If the transmitter reaches the next checkpoint without receiving a "sync ack" response, then synchronization is broken, and the transmitter should stop transmitting. The transmitter will continue to send sync_reqs until it receives a sync_ack , at which point transmission is reestablished.

From the receiver's perspective, the scheme operates as follows: (1) when it receives a "sync request" request from the transmitter, it advances its window to the next checkpoint position (even skipping pairs if necessary), and sends a "sync ack" message to the transmitter. If sync was never lost, then the "jump ahead" really just advances to the next available pair of addresses in the table (i.e., normal advancement).

If an interloper intercepts the "sync request" messages and tries to interfere with communication by sending new ones, it will be ignored if the synchronization has been established or it will actually help to re-establish synchronization.

A window is realigned whenever a re-synchronization occurs. This realignment entails updating the receiver's window to straddle the address pairs used by the packet transmitted immediately after the transmission of the SYNC_REQ packet. Normally, the transmitter and receiver are in synchronization with one another. However, when network events occur, the receiver's window may have to be advanced by many steps during resynchronization. In this

case, it is desirable to move the window ahead without having to step through the intervening random numbers sequentially. (This feature is also desirable for the auto-sync approach discussed above).

E. Random Number Generator with a Jump-Ahead capability

An attractive method for generating randomly hopped addresses is to use identical random number generators in the transmitter and receiver and advance them as packets are transmitted and received. There are many random number generation algorithms that could be used. Each one has strengths and weaknesses for address hopping applications.

Linear congruential random number generators (LCRs) are fast, simple and well characterized random number generators that can be made to jump ahead *n* steps efficiently. An LCR generates random numbers $X_1, X_2, X_3 \ldots X_k$ starting with seed X_0 using a recurrence

$$X_i=(a X_{i-1} + b) \bmod c, \qquad (1)$$

where a, b and c define a particular LCR. Another expression for X_i,

$$X_i=((a^i(X_0+b)-b)/(a-1)) \bmod c \quad (2)$$

enables the jump-ahead capability. The factor a^i can grow very large even for modest i if left unfettered. Therefore some special properties of the modulo operation can be used to control the size and processing time required to compute (2). (2) can be rewritten as:

$$X_i=(a^i (X_0(a-1)+b)-b)/(a-1) \bmod c. \quad (3)$$

It can be shown that:

$$(a^i(X_0(a-1)+b)-b)/(a-1) \bmod c =$$

$$((a^i \bmod((a-1)c)(X_0(a-1)+b) -b) /(a-1)) \bmod c \qquad (4).$$

$(X_0(a-1)+b)$ can be stored as $(X_0(a-1)+b) \bmod c$, b as b mod c and compute $a^i \bmod((a-1)c)$ (this requires O(log(i)) steps).

A practical implementation of this algorithm would jump a fixed distance, n, between synchronizations; this is tantamount to synchronizing every *n* packets. The window would commence *n* IP pairs from the start of the previous window. Using X_j^w, the random number at the j^{th} checkpoint, as X_0 and *n* as *i*, a node can store $a^n \bmod((a-1)c)$ once per LCR and set

$$X_{j+1}{}^w=X_{n(j+1)}=((a^n \bmod((a-1)c) (X_j{}^w (a-1)+b)-b)/(a-1))\bmod c, \quad (5)$$

to generate the random number for the $j+1^{th}$ synchronization. Using this construction, a node could jump ahead an arbitrary (but fixed) distance between synchronizations in a constant amount of time (independent of n).

Pseudo-random number generators, in general, and LCRs, in particular, will eventually repeat their cycles. This repetition may present vulnerability in the IP hopping scheme. An adversary would simply have to wait for a repeat to predict future sequences. One way of coping with this vulnerability is to create a random number generator with a known long cycle. A random sequence can be replaced by a new random number generator before it repeats. LCRs can be constructed with known long cycles. This is not currently true of many random number generators.

Random number generators can be cryptographically insecure. An adversary can derive the RNG parameters by examining the output or part of the output. This is true of LCGs. This vulnerability can be mitigated by incorporating an encryptor, designed to scramble the output as part of the random number generator. The random number generator prevents an adversary from mounting an attack—e.g., a known plaintext attack—against the encryptor.

F. Random Number Generator Example

Consider a RNG where a=31,b=4 and c=15. For this case equation (1) becomes:

$X_i=(31\ X_{i-1} + 4) \bmod 15$. (6)

If one sets $X_0=1$, equation (6) will produce the sequence 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 0, 4, 8, 12. This sequence will repeat indefinitely. For a jump ahead of 3 numbers in this sequence $a^n= 31^3=29791$, $c*(a-1)=15*30=450$ and $a^n \bmod((a-1)c) = 31^3 \bmod(15*30)=29791 \bmod(450)=91$. Equation (5) becomes:

$((91\ (X_i30+4)-4)/30) \bmod 15$ (7).

Table 1 shows the jump ahead calculations from (7) . The calculations start at 5 and jump ahead 3.

TABLE 1

I	X_i	(X_i30+4)	$91\ (X_i30+4)-4$	$((91\ (X_i30+4)-4)/30$	X_{i+3}
1	5	154	14010	467	2
4	2	64	5820	194	14
7	14	424	38580	1286	11
10	11	334	30390	1013	8
13	8	244	22200	740	5

G. Fast Packet Filter

Address hopping VPNs must rapidly determine whether a packet has a valid header and thus requires further processing, or has an invalid header (a hostile packet) and should be immediately rejected. Such rapid determinations will be referred to as "fast packet filtering." This capability protects the VPN from attacks by an adversary who streams hostile packets at the receiver at a high rate of speed in the hope of saturating the receiver's processor (a so-called "denial of service" attack). Fast packet filtering is an important feature for implementing VPNs on shared media such as Ethernet.

Assuming that all participants in a VPN share an unassigned "A" block of addresses, one possibility is to use an experimental "A" block that will never be assigned to any machine that is not address hopping on the shared medium. "A" blocks have a 24 bits of address that can be hopped as opposed to the 8 bits in "C" blocks. In this case a hopblock will be the "A" block. The use of the experimental "A" block is a likely option on an Ethernet because:

1. The addresses have no validity outside of the Ethernet and will not be routed out to a valid outside destination by a gateway.
2. There are 2^{24} (~16 million) addresses that can be hopped within each "A" block. This yields >280 trillion possible address pairs making it very unlikely that an adversary would guess a valid address. It also provides acceptably low probability of collision between separate VPNs (all VPNs on a shared medium independently generate random address pairs from the same "A" block).
3. The packets will not be received by someone on the Ethernet who is not on a VPN (unless the machine is in promiscuous mode) minimizing impact on non-VPN computers.

The Ethernet example will be used to describe one implementation of fast packet filtering. The ideal algorithm would quickly examine a packet header, determine whether the packet is hostile, and reject any hostile packets or determine which active IP pair the packet header matches. The problem is a classical associative memory problem. A variety of techniques have been developed to solve this problem (hashing, B–trees etc). Each of these approaches has its strengths and weaknesses. For instance, hash tables can be made to operate quite fast in a statistical sense, but can occasionally degenerate into a much slower algorithm. This slowness can persist for a period of time. Since there is a need to discard hostile packets quickly at all times, hashing would be unacceptable.

H. Presence Vector Algorithm

A presence vector is a bit vector of length 2^n that can be indexed by n-bit numbers (each ranging from 0 to 2^n–1). One can indicate the presence of k n-bit numbers (not necessarily unique), by setting the bits in the presence vector indexed by each number to 1. Otherwise, the bits in the presence vector are 0. An n-bit number, x, is one of the k numbers if and only if the x^{th} bit of the presence vector is 1. A fast packet filter can be implemented by indexing the presence vector and looking for a 1, which will be referred to as the "test."

For example, suppose one wanted to represent the number 135 using a presence vector. The 135^{th} bit of the vector would be set. Consequently, one could very quickly determine whether an address of 135 was valid by checking only one bit: the 135^{th} bit. The presence vectors could be created in advance corresponding to the table entries for the IP addresses. In effect, the incoming addresses can be used as indices into a long vector, making comparisons very fast. As each RNG generates a new address, the presence vector is updated to reflect the information. As the window moves, the presence vector is updated to zero out addresses that are no longer valid.

There is a trade-off between efficiency of the test and the amount of memory required for storing the presence vector(s). For instance, if one were to use the 48 bits of hopping addresses as an index, the presence vector would have to be 35 terabytes. Clearly, this is too large for practical purposes. Instead, the 48 bits can be divided into several smaller fields. For instance, one could subdivide the 48 bits into four 12-bit fields (see FIG. 16). This reduces the storage requirement to 2048 bytes at the expense of occasionally having to process a hostile packet. In effect, instead of one long presence vector, the decomposed address portions must match all four shorter presence vectors before further processing is allowed. (If the first part of the address portion doesn't match the first presence vector, there is no need to check the remaining three presence vectors).

A presence vector will have a 1 in the y^{th} bit if and only if one or more addresses with a corresponding field of y are active. An address is active only if each presence vector indexed by the appropriate sub-field of the address is 1.

Consider a window of 32 active addresses and 3 checkpoints. A hostile packet will be rejected by the indexing of one presence vector more than 99% of the time. A hostile packet will be rejected by the indexing of all 4 presence vectors more than 99.9999995% of the time. On average, hostile packets will be rejected in less than 1.02 presence vector index operations.

The small percentage of hostile packets that pass the fast packet filter will be rejected when matching pairs are not found in the active window or are active checkpoints. Hostile packets that serendipitously match a header will be rejected when the VPN software attempts to decrypt the header. However, these cases will be extremely rare. There are many other ways this method can be configured to arbitrate the space/speed tradeoffs.

I. Further Synchronization Enhancements

A slightly modified form of the synchronization techniques described above can be employed. The basic principles of the previously described checkpoint synchronization scheme remain unchanged. The actions resulting from the reception of the checkpoints are, however, slightly different. In this variation, the receiver will maintain between OoO ("Out of Order") and 2×WINDOW_SIZE+OoO active addresses ($1 \leq OoO \leq WINDOW\_SIZE$ and $WINDOW\_SIZE \geq 1$). OoO and WINDOW_SIZE are engineerable parameters, where OoO is the minimum number of addresses needed to accommodate lost packets due to events in the network or out of order arrivals and WINDOW_SIZE is the number of packets transmitted before a SYNC_REQ is issued. FIG. 17 depicts a storage array for a receiver's active addresses.

The receiver starts with the first 2×WINDOW_SIZE addresses loaded and active (ready to receive data). As packets are received, the corresponding entries are marked as "used" and are no longer eligible to receive packets. The transmitter maintains a packet counter, initially set to 0, containing the number of data packets transmitted since the last *initial* transmission of a SYNC_REQ for which SYNC_ACK has been received. When the transmitter packet counter equals WINDOW_SIZE, the transmitter generates a SYNC_REQ and does its initial transmission. When the receiver receives a SYNC_REQ corresponding to its current CKPT_N, it generates the next WINDOW_SIZE addresses and starts loading them in order starting at the first location after the last active address wrapping around to the beginning of the array after the end of the array has been reached. The receiver's array might look like FIG. 18 when a SYNC_REQ has been received. In this case a couple of packets have been either lost or will be received out of order when the SYNC_REQ is received.

FIG. 19 shows the receiver's array after the new addresses have been generated. If the transmitter does not receive a SYNC_ACK, it will re-issue the SYNC_REQ at regular intervals. When the transmitter receives a SYNC_ACK, the packet counter is decremented by WINDOW_SIZE. If the packet counter reaches 2×WINDOW_SIZE – OoO then the transmitter

ceases sending data packets until the appropriate SYNC_ACK is finally received. The transmitter then resumes sending data packets. Future behavior is essentially a repetition of this initial cycle. The advantages of this approach are:

1. There is no need for an efficient jump ahead in the random number generator,
2. No packet is ever transmitted that does not have a corresponding entry in the receiver side
3. No timer based re-synchronization is necessary. This is a consequence of 2.
4. The receiver will always have the ability to accept data messages transmitted within OoO messages of the most recently transmitted message.

J. Distributed Transmission Path Variant

Another embodiment incorporating various inventive principles is shown in FIG. 20. In this embodiment, a message transmission system includes a first computer 2001 in communication with a second computer 2002 through a network 2011 of intermediary computers. In one variant of this embodiment, the network includes two edge routers 2003 and 2004 each of which is linked to a plurality of Internet Service Providers (ISPs) 2005 through 2010. Each ISP is coupled to a plurality of other ISPs in an arrangement as shown in FIG. 20, which is a representative configuration only and is not intended to be limiting. Each connection between ISPs is labeled in FIG. 20 to indicate a specific physical transmission path (e.g., AD is a physical path that links ISP A (element 2005) to ISP D (element 2008)). Packets arriving at each edge router are selectively transmitted to one of the ISPs to which the router is attached on the basis of a randomly or quasi-randomly selected basis.

As shown in FIG. 21, computer 2001 or edge router 2003 incorporates a plurality of link transmission tables 2100 that identify, for each potential transmission path through the network, valid sets of IP addresses that can be used to transmit the packet. For example, AD table 2101 contains a plurality of IP source/destination pairs that are randomly or quasi-randomly generated. When a packet is to be transmitted from first computer 2001 to second computer 2002, one of the link tables is randomly (or quasi-randomly) selected, and the next valid source/destination address pair from that table is used to transmit the packet through the network. If path AD is randomly selected, for example, the next source/destination IP address pair (which is pre-determined to transmit between ISP A (element 2005) and ISP B (element 2008)) is used to transmit the packet. If one of the transmission paths becomes degraded or inoperative, that link

table can be set to a "down" condition as shown in table 2105, thus preventing addresses from being selected from that table. Other transmission paths would be unaffected by this broken link.

3. CONTINUATION-IN-PART IMPROVEMENTS

The following describes various improvements and features that can be applied to the embodiments described above. The improvements include: (1) a load balancer that distributes packets across different transmission paths according to transmission path quality; (2) a DNS proxy server that transparently creates a virtual private network in response to a domain name inquiry; (3) a large-to-small link bandwidth management feature that prevents denial-of-service attacks at system chokepoints; (4) a traffic limiter that regulates incoming packets by limiting the rate at which a transmitter can be synchronized with a receiver; and (5) a signaling synchronizer that allows a large number of nodes to communicate with a central node by partitioning the communication function between two separate entities. Each is discussed separately below.

A. Load Balancer

Various embodiments described above include a system in which a transmitting node and a receiving node are coupled through a plurality of transmission paths, and wherein successive packets are distributed quasi-randomly over the plurality of paths. See, for example, FIGS. 20 and 21 and accompanying description. The improvement extends this basic concept to encompass distributing packets across different paths in such a manner that the loads on the paths are generally balanced according to transmission link quality.

In one embodiment, a system includes a transmitting node and a receiving node that are linked via a plurality of transmission paths having potentially varying transmission quality. Successive packets are transmitted over the paths based on a weight value distribution function for each path. The rate that packets will be transmitted over a given path can be different for each path. The relative "health" of each transmission path is monitored in order to identify paths that have become degraded. In one embodiment, the health of each path is monitored in the transmitter by comparing the number of packets transmitted to the number of packet acknowledgements received. Each transmission path may comprise a physically separate path (e.g., via dial-up phone line, computer network, router, bridge, or the like), or may comprise logically separate paths contained within a broadband communication medium (e.g., separate channels in an FDM, TDM, CDMA, or other type of modulated or unmodulated transmission link).

When the transmission quality of a path falls below a predetermined threshold and there are other paths that can transmit packets, the transmitter changes the weight value used for that path, making it less likely that a given packet will be transmitted over that path. The weight will preferably be set no lower than a minimum value that keeps nominal traffic on the path. The weights of the other available paths are altered to compensate for the change in the affected path. When the quality of a path degrades to where the transmitter is turned off by the synchronization function (i.e., no packets are arriving at the destination), the weight is set to zero. If all transmitters are turned off, no packets are sent.

Conventional TCP/IP protocols include a "throttling" feature that reduces the transmission rate of packets when it is determined that delays or errors are occurring in transmission. In this respect, timers are sometimes used to determine whether packets have been received. These conventional techniques for limiting transmission of packets, however, do not involve multiple transmission paths between two nodes wherein transmission across a particular path relative to the others is changed based on link quality.

According to certain embodiments, in order to damp oscillations that might otherwise occur if weight distributions are changed drastically (e.g., according to a step function), a linear or an exponential decay formula can be applied to gradually decrease the weight value over time that a degrading path will be used. Similarly, if the health of a degraded path improves, the weight value for that path is gradually increased.

Transmission link health can be evaluated by comparing the number of packets that are acknowledged within the transmission window (see embodiments discussed above) to the number of packets transmitted within that window and by the state of the transmitter (i.e., on or off). In other words, rather than accumulating general transmission statistics over time for a path, one specific implementation uses the "windowing" concepts described above to evaluate transmission path health.

The same scheme can be used to shift virtual circuit paths from an "unhealthy" path to a "healthy" one, and to select a path for a new virtual circuit.

FIG. 22A shows a flowchart for adjusting weight values associated with a plurality of transmission links. It is assumed that software executing in one or more computer nodes executes the steps shown in FIG. 22A. It is also assumed that the software can be stored on a computer-readable medium such as a magnetic or optical disk for execution by a computer.

Beginning in step 2201, the transmission quality of a given transmission path is measured. As described above, this measurement can be based on a comparison between the number of packets transmitted over a particular link to the number of packet acknowledgements received over the link (e.g., per unit time, or in absolute terms). Alternatively, the quality can be evaluated by comparing the number of packets that are acknowledged within the transmission window to the number of packets that were transmitted within that window. In yet another variation, the number of missed synchronization messages can be used to indicate link quality. Many other variations are of course possible.

In step 2202, a check is made to determine whether more than one transmitter (e.g., transmission path) is turned on. If not, the process is terminated and resumes at step 2201.

In step 2203, the link quality is compared to a given threshold (e.g., 50%, or any arbitrary number). If the quality falls below the threshold, then in step 2207 a check is made to determine whether the weight is above a minimum level (e.g., 1%). If not, then in step 2209 the weight is set to the minimum level and processing resumes at step 2201. If the weight is above the minimum level, then in step 2208 the weight is gradually decreased for the path, then in step 2206 the weights for the remaining paths are adjusted accordingly to compensate (e.g., they are increased).

If in step 2203 the quality of the path was greater than or equal to the threshold, then in step 2204 a check is made to determine whether the weight is less than a steady-state value for that path. If so, then in step 2205 the weight is increased toward the steady-state value, and in step 2206 the weights for the remaining paths are adjusted accordingly to compensate (e.g., they are decreased). If in step 2204 the weight is not less than the steady-state value, then processing resumes at step 2201 without adjusting the weights.

The weights can be adjusted incrementally according to various functions, preferably by changing the value gradually. In one embodiment, a linearly decreasing function is used to adjust the weights; according to another embodiment, an exponential decay function is used. Gradually changing the weights helps to damp oscillators that might otherwise occur if the probabilities were abruptly.

Although not explicitly shown in FIG. 22A the process can be performed only periodically (e.g., according to a time schedule), or it can be continuously run, such as in a background mode of operation. In one embodiment, the combined weights of all potential paths

should add up to unity (e.g., when the weighting for one path is decreased, the corresponding weights that the other paths will be selected will increase).

Adjustments to weight values for other paths can be prorated. For example, a decrease of 10% in weight value for one path could result in an evenly distributed increase in the weights for the remaining paths. Alternatively, weightings could be adjusted according to a weighted formula as desired (e.g., favoring healthy paths over less healthy paths). In yet another variation, the difference in weight value can be amortized over the remaining links in a manner that is proportional to their traffic weighting.

FIG. 22B shows steps that can be executed to shut down transmission links where a transmitter turns off. In step 2210, a transmitter shut-down event occurs. In step 2211, a test is made to determine whether at least one transmitter is still turned on. If not, then in step 2215 all packets are dropped until a transmitter turns on. If in step 2211 at least one transmitter is turned on, then in step 2212 the weight for the path is set to zero, and the weights for the remaining paths are adjusted accordingly.

FIG. 23 shows a computer node 2301 employing various principles of the above-described embodiments. It is assumed that two computer nodes of the type shown in FIG. 23 communicate over a plurality of separate physical transmission paths. As shown in FIG. 23, four transmission paths X1 through X4 are defined for communicating between the two nodes. Each node includes a packet transmitter 2302 that operates in accordance with a transmit table 2308 as described above. (The packet transmitter could also operate without using the IP-hopping features described above, but the following description assumes that some form of hopping is employed in conjunction with the path selection mechanism.). The computer node also includes a packet receiver 2303 that operates in accordance with a receive table 2309, including a moving window W that moves as valid packets are received. Invalid packets having source and destination addresses that do not fall within window W are rejected.

As each packet is readied for transmission, source and destination IP addresses (or other discriminator values) are selected from transmit table 2308 according to any of the various algorithms described above, and packets containing these source/destination address pairs, which correspond to the node to which the four transmission paths are linked, are generated to a transmission path switch 2307. Switch 2307, which can comprise a software function, selects from one of the available transmission paths according to a weight distribution table 2306. For

example, if the weight for path X1 is 0.2, then every fifth packet will be transmitted on path X1. A similar regime holds true for the other paths as shown. Initially, each link's weight value can be set such that it is proportional to its bandwidth, which will be referred to as its "steady-state" value.

Packet receiver 2303 generates an output to a link quality measurement function 2304 that operates as described above to determine the quality of each transmission path. (The input to packet receiver 2303 for receiving incoming packets is omitted for clarity). Link quality measurement function 2304 compares the link quality to a threshold for each transmission link and, if necessary, generates an output to weight adjustment function 2305. If a weight adjustment is required, then the weights in table 2306 are adjusted accordingly, preferably according to a gradual (e.g., linearly or exponentially declining) function. In one embodiment, the weight values for all available paths are initially set to the same value, and only when paths degrade in quality are the weights changed to reflect differences.

Link quality measurement function 2304 can be made to operate as part of a synchronizer function as described above. That is, if resynchronization occurs and the receiver detects that synchronization has been lost (e.g., resulting in the synchronization window W being advanced out of sequence), that fact can be used to drive link quality measurement function 2304. According to one embodiment, load balancing is performed using information garnered during the normal synchronization, augmented slightly to communicate link health from the receiver to the transmitter. The receiver maintains a count, MESS_R(W), of the messages received in synchronization window W. When it receives a synchronization request (SYNC_REQ) corresponding to the end of window W, the receiver includes counter MESS_R in the resulting synchronization acknowledgement (SYNC_ACK) sent back to the transmitter. This allows the transmitter to compare messages sent to messages received in order to asses the health of the link.

If synchronization is completely lost, weight adjustment function 2305 decreases the weight value on the affected path to zero. When synchronization is regained, the weight value for the affected path is gradually increased to its original value. Alternatively, link quality can be measured by evaluating the length of time required for the receiver to acknowledge a synchronization request. In one embodiment, separate transmit and receive tables are used for each transmission path.

When the transmitter receives a SYNC_ACK, the MESS_R is compared with the number of messages transmitted in a window (MESS_T). When the transmitter receives a SYNC_ACK, the traffic probabilities will be examined and adjusted if necessary. MESS_R is compared with the number of messages transmitted in a window (MESS_T). There are two possibilities:

1. If MESS_R is less than a threshold value, THRESH, then the link will be deemed to be unhealthy. If the transmitter was turned off, the transmitter is turned on and the weight P for that link will be set to a minimum value MIN. This will keep a trickle of traffic on the link for monitoring purposes until it recovers. If the transmitter was turned on, the weight P for that link will be set to:

$$P' = \alpha \times MIN + (1 - \alpha) \times P \quad (1)$$

Equation 1 will exponentially damp the traffic weight value to MIN during sustained periods of degraded service.

2. If MESS_R for a link is greater than or equal to THRESH, the link will be deemed healthy. If the weight P for that link is greater than or equal to the steady state value S for that link, then P is left unaltered. If the weight P for that link is less than THRESH then P will be set to:

$$P' = \beta \times S + (1 - \beta) \times P \quad (2)$$

where β is a parameter such that $0 <= \beta <= 1$ that determines the damping rate of P.

Equation 2 will increase the traffic weight to S during sustained periods of acceptable service in a damped exponential fashion.

A detailed example will now be provided with reference to FIG. 24. As shown in FIG. 24, a first computer 2401 communicates with a second computer 2402 through two routers 2403 and 2404. Each router is coupled to the other router through three transmission links. As described above, these may be physically diverse links or logical links (including virtual private networks).

Suppose that a first link L1 can sustain a transmission bandwidth of 100 Mb/s and has a window size of 32; link L2 can sustain 75 Mb/s and has a window size of 24; and link L3 can sustain 25 Mb/s and has a window size of 8. The combined links can thus sustain 200Mb/s. The steady state traffic weights are 0.5 for link L1; 0.375 for link L2, and 0.125 for link L3. MIN=1Mb/s, THRESH =0.8 MESS_T for each link, α=.75 and β=.5. These traffic weights will

remain stable until a link stops for synchronization or reports a number of packets received less than its THRESH. Consider the following sequence of events:

1. Link L1 receives a SYNC_ACK containing a MESS_R of 24, indicating that only 75% of the MESS_T (32) messages transmitted in the last window were successfully received. Link 1 would be below THRESH (0.8). Consequently, link L1's traffic weight value would be reduced to 0.12825, while link L2's traffic weight value would be increased to 0.65812 and link L3's traffic weight value would be increased to 0.217938.

2. Link L2 and L3 remained healthy and link L1 stopped to synchronize. Then link L1's traffic weight value would be set to 0, link L2's traffic weight value would be set to 0.75, and link L33's traffic weight value would be set to 0.25.

3. Link L1 finally received a SYNC_ACK containing a MESS_R of 0 indicating that none of the MESS_T (32) messages transmitted in the last window were successfully received. Link L1 would be below THRESH. Link L1's traffic weight value would be increased to .005, link L2's traffic weight value would be decreased to 0.74625, and link L3's traffic weight value would be decreased to 0.24875.

4. Link L1 received a SYNC_ACK containing a MESS_R of 32 indicating that 100% of the MESS_T (32) messages transmitted in the last window were successfully received. Link L1 would be above THRESH. Link L1's traffic weight value would be increased to 0.2525, while link L2's traffic weight value would be decreased to 0.560625 and link L3's traffic weight value would be decreased to .186875.

5. Link L1 received a SYNC_ACK containing a MESS_R of 32 indicating that 100% of the MESS_T (32) messages transmitted in the last window were successfully received. Link L1 would be above THRESH. Link L1's traffic weight value would be increased to 0.37625; link L2's traffic weight value would be decreased to 0.4678125, and link L3's traffic weight value would be decreased to 0.1559375.

6. Link L1 remains healthy and the traffic probabilities approach their steady state traffic probabilities.

B. Use of a DNS Proxy to Transparently Create Virtual Private Networks

A second improvement concerns the automatic creation of a virtual private network (VPN) in response to a domain-name server look-up function.

Conventional Domain Name Servers (DNSs) provide a look-up function that returns the IP address of a requested computer or host. For example, when a computer user types in the web name "Yahoo.com," the user's web browser transmits a request to a DNS, which converts the name into a four-part IP address that is returned to the user's browser and then used by the browser to contact the destination web site.

This conventional scheme is shown in FIG. 25. A user's computer 2501 includes a client application 2504 (for example, a web browser) and an IP protocol stack 2505. When the user enters the name of a destination host, a request DNS REQ is made (through IP protocol stack 2505) to a DNS 2502 to look up the IP address associated with the name. The DNS returns the IP address DNS RESP to client application 2504, which is then able to use the IP address to communicate with the host 2503 through separate transactions such as PAGE REQ and PAGE RESP.

In the conventional architecture shown in FIG. 25, nefarious listeners on the Internet could intercept the DNS REQ and DNS RESP packets and thus learn what IP addresses the user was contacting. For example, if a user wanted to set up a secure communication path with a web site having the name "Target.com," when the user's browser contacted a DNS to find the IP address for that web site, the true IP address of that web site would be revealed over the Internet as part of the DNS inquiry. This would hamper anonymous communications on the Internet.

One conventional scheme that provides secure virtual private networks over the Internet provides the DNS server with the public keys of the machines that the DNS server has the addresses for. This allows hosts to retrieve automatically the public keys of a host that the host is to communicate with so that the host can set up a VPN without having the user enter the public key of the destination host. One implementation of this standard is presently being developed as part of the FreeS/WAN project(RFC 2535).

The conventional scheme suffers from certain drawbacks. For example, any user can perform a DNS request. Moreover, DNS requests resolve to the same value for all users.

According to certain aspects of the invention, a specialized DNS server traps DNS requests and, if the request is from a special type of user (e.g., one for which secure communication services are defined), the server does not return the true IP address of the target node, but instead automatically sets up a virtual private network between the target node and the user. The VPN is preferably implemented using the IP address "hopping" features of the basic

invention described above, such that the true identity of the two nodes cannot be determined even if packets during the communication are intercepted. For DNS requests that are determined to not require secure services (e.g., an unregistered user), the DNS server transparently "passes through" the request to provide a normal look-up function and return the IP address of the target web server, provided that the requesting host has permissions to resolve unsecured sites. Different users who make an identical DNS request could be provided with different results.

FIG. 26 shows a system employing various principles summarized above. A user's computer 2601 includes a conventional client (e.g., a web browser) 2605 and an IP protocol stack 2606 that preferably operates in accordance with an IP hopping function 2607 as outlined above. A modified DNS server 2602 includes a conventional DNS server function 2609 and a DNS proxy 2610. A gatekeeper server 2603 is interposed between the modified DNS server and a secure target site 2704. An "unsecure" target site 2611 is also accessible via conventional IP protocols.

According to one embodiment, DNS proxy 2610 intercepts all DNS lookup functions from client 2605 and determines whether access to a secure site has been requested. If access to a secure site has been requested (as determined, for example, by a domain name extension, or by reference to an internal table of such sites), DNS proxy 2610 determines whether the user has sufficient security privileges to access the site. If so, DNS proxy 2610 transmits a message to gatekeeper 2603 requesting that a virtual private network be created between user computer 2601 and secure target site 2604. In one embodiment, gatekeeper 2603 creates "hopblocks" to be used by computer 2601 and secure target site 2604 for secure communication. Then, gatekeeper 2603 communicates these to user computer 2601. Thereafter, DNS proxy 2610 returns to user computer 2601 the resolved address passed to it by the gatekeeper (this address could be different from the actual target computer) 2604, preferably using a secure administrative VPN. The address that is returned need not be the actual address of the destination computer.

Had the user requested lookup of a non-secure web site such as site 2611, DNS proxy would merely pass through to conventional DNS server 2609 the look-up request, which would be handled in a conventional manner, returning the IP address of non-secure web site 2611. If the user had requested lookup of a secure web site but lacked credentials to create such a connection, DNS proxy 2610 would return a "host unknown" error to the user. In this manner,

different users requesting access to the same DNS name could be provided with different look-up results.

Gatekeeper 2603 can be implemented on a separate computer (as shown in FIG. 26) or as a function within modified DNS server 2602. In general, it is anticipated that gatekeeper 2703 facilitates the allocation and exchange of information needed to communicate securely, such as using "hopped" IP addresses. Secure hosts such as site 2604 are assumed to be equipped with a secure communication function such as an IP hopping function 2608.

It will be appreciated that the functions of DNS proxy 2610 and DNS server 2609 can be combined into a single server for convenience. Moreover, although element 2602 is shown as combining the functions of two servers, the two servers can be made to operate independently.

FIG. 27 shows steps that can be executed by DNS proxy server 2610 to handle requests for DNS look-up for secure hosts. In step 2701, a DNS look-up request is received for a target host. In step 2702, a check is made to determine whether access to a secure host was requested. If not, then in step 2703 the DNS request is passed to conventional DNS server 2609, which looks up the IP address of the target site and returns it to the user's application for further processing.

In step 2702, if access to a secure host was requested, then in step 2704 a further check is made to determine whether the user is authorized to connect to the secure host. Such a check can be made with reference to an internally stored list of authorized IP addresses, or can be made by communicating with gatekeeper 2603 (e.g., over an "administrative" VPN that is secure). It will be appreciated that different levels of security can also be provided for different categories of hosts. For example, some sites may be designated as having a certain security level, and the security level of the user requesting access must match that security level. The user's security level can also be determined by transmitting a request message back to the user's computer requiring that it prove that it has sufficient privileges.

If the user is not authorized to access the secure site, then a "host unknown" message is returned (step 2705). If the user has sufficient security privileges, then in step 2706 a secure VPN is established between the user's computer and the secure target site. As described above, this is preferably done by allocating a hopping regime that will be carried out between the user's computer and the secure target site, and is preferably performed transparently to the user (i.e., the user need not be involved in creating the secure link). As described in various embodiments of

this application, any of various fields can be "hopped" (e.g., IP source/destination addresses; a field in the header; etc.) in order to communicate securely.

Some or all of the security functions can be embedded in gatekeeper 2603, such that it handles all requests to connect to secure sites. In this embodiment, DNS proxy 2610 communicates with gatekeeper 2603 to determine (preferably over a secure administrative VPN) whether the user has access to a particular web site. Various scenarios for implementing these features are described by way of example below:

Scenario #1: Client has permission to access target computer, and gatekeeper has a rule to make a VPN for the client. In this scenario, the client's DNS request would be received by the DNS proxy server 2610, which would forward the request to gatekeeper 2603. The gatekeeper would establish a VPN between the client and the requested target. The gatekeeper would provide the address of the destination to the DNS proxy, which would then return the resolved name as a result. The resolved address can be transmitted back to the client in a secure administrative VPN.

Scenario #2: Client does not have permission to access target computer. In this scenario, the client's DNS request would be received by the DNS proxy server 2610, which would forward the request to gatekeeper 2603. The gatekeeper would reject the request, informing DNS proxy server 2610 that it was unable to find the target computer. The DNS proxy 2610 would then return a "host unknown" error message to the client.

Scenario #3: Client has permission to connect using a normal non-VPN link, and the gatekeeper does not have a rule to set up a VPN for the client to the target site. In this scenario, the client's DNS request is received by DNS proxy server 2610, which would check its rules and determine that no VPN is needed. Gatekeeper 2603 would then inform the DNS proxy server to forward the request to conventional DNS server 2609, which would resolve the request and return the result to the DNS proxy server and then back to the client.

Scenario #4: Client does not have permission to establish a normal/non-VPN link, and the gatekeeper does not have a rule to make a VPN for the client to the target site. In this scenario, the DNS proxy server would receive the client's DNS request and forward it to gatekeeper 2603. Gatekeeper 2603 would determine that no special VPN was needed, but that the client is not authorized to communicate with non-VPN members. The gatekeeper would reject the request, causing DNS proxy server 2610 to return an error message to the client.

C. Large Link to Small Link Bandwidth Management

One feature of the basic architecture is the ability to prevent so-called "denial of service" attacks that can occur if a computer hacker floods a known Internet node with packets, thus preventing the node from communicating with other nodes. Because IP addresses or other fields are "hopped" and packets arriving with invalid addresses are quickly discarded, Internet nodes are protected against flooding targeted at a single IP address.

In a system in which a computer is coupled through a link having a limited bandwidth (e.g., an edge router) to a node that can support a much higher-bandwidth link (e.g., an Internet Service Provider), a potential weakness could be exploited by a determined hacker. Referring to FIG. 28, suppose that a first host computer 2801 is communicating with a second host computer 2804 using the IP address hopping principles described above. The first host computer is coupled through an edge router 2802 to an Internet Service Provider (ISP) 2803 through a low bandwidth link (LOW BW), and is in turn coupled to second host computer 2804 through parts of the Internet through a high bandwidth link (HIGH BW). In this architecture, the ISP is able to support a high bandwidth to the internet, but a much lower bandwidth to the edge router 2802.

Suppose that a computer hacker is able to transmit a large quantity of dummy packets addressed to first host computer 2801 across high bandwidth link HIGH BW. Normally, host computer 2801 would be able to quickly reject the packets since they would not fall within the acceptance window permitted by the IP address hopping scheme. However, because the packets must travel across low bandwidth link LOW BW, the packets overwhelm the lower bandwidth link before they are received by host computer 2801. Consequently, the link to host computer 2801 is effectively flooded before the packets can be discarded.

According to one inventive improvement, a "link guard" function 2805 is inserted into the high-bandwidth node (e.g., ISP 2803) that quickly discards packets destined for a low-bandwidth target node if they are not valid packets. Each packet destined for a low-bandwidth node is cryptographically authenticated to determine whether it belongs to a VPN. If it is not a valid VPN packet, the packet is discarded at the high-bandwidth node. If the packet is authenticated as belonging to a VPN, the packet is passed with high preference. If the packet is a valid non-VPN packet, it is passed with a lower quality of service (e.g., lower priority).

In one embodiment, the ISP distinguishes between VPN and non-VPN packets using the protocol of the packet. In the case of IPSEC [rfc 2401], the packets have IP protocols 420 and

421. In the case of the TARP VPN, the packets will have an IP protocol that is not yet defined. The ISP's link guard, 2805, maintains a table of valid VPNs which it uses to validate whether VPN packets are cryptographically valid. According to one embodiment, packets that do not fall within any hop windows used by nodes on the low-bandwidth link are rejected, or are sent with a lower quality of service. One approach for doing this is to provide a copy of the IP hopping tables used by the low-bandwidth nodes to the high-bandwidth node, such that both the high-bandwidth and low-bandwidth nodes track hopped packets (e.g., the high-bandwidth node moves its hopping window as valid packets are received). In such a scenario, the high-bandwidth node discards packets that do not fall within the hopping window before they are transmitted over the low-bandwidth link. Thus, for example, ISP 2903 maintains a copy 2910 of the receive table used by host computer 2901. Incoming packets that do not fall within this receive table are discarded. According to a different embodiment, link guard 2805 validates each VPN packet using a keyed hashed message authentication code (HMAC) [rfc 2104].

According to another embodiment, separate VPNs (using, for example, hopblocks) can be established for communicating between the low-bandwidth node and the high-bandwidth node (i.e., packets arriving at the high-bandwidth node are converted into different packets before being transmitted to the low-bandwidth node).

As shown in FIG. 29, for example, suppose that a first host computer 2900 is communicating with a second host computer 2902 over the Internet, and the path includes a high bandwidth link HIGH BW to an ISP 2901 and a low bandwidth link LOW BW through an edge router 2904. In accordance with the basic architecture described above, first host computer 2900 and second host computer 2902 would exchange hopblocks (or a hopblock algorithm) and would be able to create matching transmit and receive tables 2905, 2906, 2912 and 2913. Then in accordance with the basic architecture, the two computers would transmit packets having seemingly random IP source and destination addresses, and each would move a corresponding hopping window in its receive table as valid packets were received.

Suppose that a nefarious computer hacker 2903 was able to deduce that packets having a certain range of IP addresses (e.g., addresses 100 to 200 for the sake of simplicity) are being transmitted to ISP 2901, and that these packets are being forwarded over a low-bandwidth link. Hacker computer 2903 could thus "flood" packets having addresses falling into the range 100 to 200, expecting that they would be forwarded along low bandwidth link LOW BW, thus causing

the low bandwidth link to become overwhelmed. The fast packet reject mechanism in first host computer 3000 would be of little use in rejecting these packets, since the low bandwidth link was effectively jammed before the packets could be rejected. In accordance with one aspect of the improvement, however, VPN link guard 2911 would prevent the attack from impacting the performance of VPN traffic because the packets would either be rejected as invalid VPN packets or given a lower quality of service than VPN traffic over the lower bandwidth link. A denial-of-service flood attack could, however, still disrupt non-VPN traffic.

According to one embodiment of the improvement, ISP 2901 maintains a separate VPN with first host computer 2900, and thus translates packets arriving at the ISP into packets having a different IP header before they are transmitted to host computer 2900. The cryptographic keys used to authenticate VPN packets at the link guard 2911 and the cryptographic keys used to encrypt and decrypt the VPN packets at host 2902 and host 2901 can be different, so that link guard 2911 does not have access to the private host data; it only has the capability to authenticate those packets.

According to yet a third embodiment, the low-bandwidth node can transmit a special message to the high-bandwidth node instructing it to shut down all transmissions on a particular IP address, such that only hopped packets will pass through to the low-bandwidth node. This embodiment would prevent a hacker from flooding packets using a single IP address. According to yet a fourth embodiment, the high-bandwidth node can be configured to discard packets transmitted to the low-bandwidth node if the transmission rate exceeds a certain predetermined threshold for any given IP address; this would allow hopped packets to go through. In this respect, link guard 2911 can be used to detect that the rate of packets on a given IP address are exceeding a threshold rate; further packets addressed to that same IP address would be dropped or transmitted at a lower priority (e.g., delayed).

D. Traffic Limiter

In a system in which multiple nodes are communicating using "hopping" technology, a treasonous insider could internally flood the system with packets. In order to prevent this possibility, one inventive improvement involves setting up "contracts" between nodes in the system, such that a receiver can impose a bandwidth limitation on each packet sender. One technique for doing this is to delay acceptance of a checkpoint synchronization request from a sender until a certain time period (e.g., one minute) has elapsed. Each receiver can effectively

control the rate at which its hopping window moves by delaying "SYNC ACK" responses to "SYNC_REQ" messages.

A simple modification to the checkpoint synchronizer will serve to protect a receiver from accidental or deliberate overload from an internally treasonous client. This modification is based on the observation that a receiver will not update its tables until a SYNC_REQ is received on hopped address CKPT_N. It is a simple matter of deferring the generation of a new CKPT_N until an appropriate interval after previous checkpoints.

Suppose a receiver wished to restrict reception from a transmitter to 100 packets a second, and that checkpoint synchronization messages were triggered every 50 packets. A compliant transmitter would not issue new SYNC_REQ messages more often than every 0.5 seconds. The receiver could delay a non-compliant transmitter from synchronizing by delaying the issuance of CKPT_N for 0.5 second after the last SYNC_REQ was accepted.

In general, if M receivers need to restrict N transmitters issuing new SYNC_REQ messages after every W messages to sending R messages a second in aggregate, each receiver could defer issuing a new CKPT_N until MxNxW/R seconds have elapsed since the last SYNC_REQ has been received and accepted. If the transmitter exceeds this rate between a pair of checkpoints, it will issue the new checkpoint before the receiver is ready to receive it, and the SYNC_REQ will be discarded by the receiver. After this, the transmitter will re-issue the SYNC_REQ every T1 seconds until it receives a SYNC_ACK. The receiver will eventually update CKPT_N and the SYNC_REQ will be acknowledged. If the transmission rate greatly exceeds the allowed rate, the transmitter will stop until it is compliant. If the transmitter exceeds the allowed rate by a little, it will eventually stop after several rounds of delayed synchronization until it is in compliance. Hacking the transmitter's code to not shut off only permits the transmitter to lose the acceptance window. In this case it can recover the window and proceed only after it is compliant again.

Two practical issues should be considered when implementing the above scheme:

1. The receiver rate should be slightly higher than the permitted rate in order to allow for statistical fluctuations in traffic arrival times and non-uniform load balancing.

2. Since a transmitter will rightfully continue to transmit for a period after a SYNC_REQ is transmitted, the algorithm above can artificially reduce the transmitter's bandwidth. If events prevent a compliant transmitter from synchronizing for a period (e.g. the network dropping a

SYNC_REQ or a SYNC_ACK) a SYNC_REQ will be accepted later than expected. After this, the transmitter will transmit fewer than expected messages before encountering the next checkpoint. The new checkpoint will not have been activated and the transmitter will have to retransmit the SYNC_REQ. This will appear to the receiver as if the transmitter is not compliant. Therefore, the next checkpoint will be accepted late from the transmitter's perspective. This has the effect of reducing the transmitter's allowed packet rate until the transmitter transmits at a packet rate below the agreed upon rate for a period of time.

To guard against this, the receiver should keep track of the times that the last C SYNC_REQs were received and accepted and use the minimum of MxNxW/R seconds after the last SYNC_REQ has been received and accepted, 2xMxNxW/R seconds after next to the last SYNC_REQ has been received and accepted, CxMxNxW/R seconds after $(C-1)^{th}$ to the last SYNC_REQ has been received, as the time to activate CKPT_N. This prevents the receiver from inappropriately limiting the transmitter's packet rate if at least one out of the last C SYNC_REQs was processed on the first attempt.

FIG. 30 shows a system employing the above-described principles. In FIG. 30, two computers 3000 and 3001 are assumed to be communicating over a network N in accordance with the "hopping" principles described above (e.g., hopped IP addresses, discriminator values, etc.). For the sake of simplicity, computer 3000 will be referred to as the receiving computer and computer 3001 will be referred to as the transmitting computer, although full duplex operation is of course contemplated. Moreover, although only a single transmitter is shown, multiple transmitters can transmit to receiver 3000.

As described above, receiving computer 3000 maintains a receive table 3002 including a window W that defines valid IP address pairs that will be accepted when appearing in incoming data packets. Transmitting computer 3001 maintains a transmit table 3003 from which the next IP address pairs will be selected when transmitting a packet to receiving computer 3000. (For the sake of illustration, window W is also illustrated with reference to transmit table 3003). As transmitting computer moves through its table, it will eventually generate a SYNC_REQ message as illustrated in function 3010. This is a request to receiver 3000 to synchronize the receive table 3002, from which transmitter 3001 expects a response in the form of a CKPT_N (included as part of a SYNC_ACK message). If transmitting computer 3001 transmits more messages than its allotment, it will prematurely generate the SYNC_REQ message. (If it has

been altered to remove the SYNC_REQ message generation altogether, it will fall out of synchronization since receiver 3000 will quickly reject packets that fall outside of window W, and the extra packets generated by transmitter 3001 will be discarded).

In accordance with the improvements described above, receiving computer 3000 performs certain steps when a SYNC_REQ message is received, as illustrated in FIG. 30. In step 3004, receiving computer 3000 receives the SYNC_REQ message. In step 3005, a check is made to determine whether the request is a duplicate. If so, it is discarded in step 3006. In step 3007, a check is made to determine whether the SYNC_REQ received from transmitter 3001 was received at a rate that exceeds the allowable rate R (i.e., the period between the time of the last SYNC_REQ message). The value R can be a constant, or it can be made to fluctuate as desired. If the rate exceeds R, then in step 3008 the next activation of the next CKPT_N hopping table entry is delayed by W/R seconds after the last SYNC_REQ has been accepted.

Otherwise, if the rate has not been exceeded, then in step 3109 the next CKPT_N value is calculated and inserted into the receiver's hopping table prior to the next SYNC_REQ from thetransmitter 3101. Transmitter 3101 then processes the SYNC_REQ in the normal manner.

E. Signaling Synchronizer

In a system in which a large number of users communicate with a central node using secure hopping technology, a large amount of memory must be set aside for hopping tables and their supporting data structures. For example, if one million subscribers to a web site occasionally communicate with the web site, the site must maintain one million hopping tables, thus using up valuable computer resources, even though only a small percentage of the users may actually be using the system at any one time. A desirable solution would be a system that permits a certain maximum number of simultaneous links to be maintained, but which would "recognize" millions of registered users at any one time. In other words, out of a population of a million registered users, a few thousand at a time could simultaneously communicate with a central server, without requiring that the server maintain one million hopping tables of appreciable size.

One solution is to partition the central node into two nodes: a signaling server that performs session initiation for user log-on and log-off (and requires only minimally sized tables), and a transport server that contains larger hopping tables for the users. The signaling server listens for the millions of known users and performs a fast-packet reject of other (bogus) packets.

When a packet is received from a known user, the signaling server activates a virtual private link (VPL) between the user and the transport server, where hopping tables are allocated and maintained. When the user logs onto the signaling server, the user's computer is provided with hop tables for communicating with the transport server, thus activating the VPL. The VPLs can be torn down when they become inactive for a time period, or they can be torn down upon user log-out. Communication with the signaling server to allow user log-on and log-off can be accomplished using a specialized version of the checkpoint scheme described above.

FIG. 31 shows a system employing certain of the above-described principles. In FIG. 31, a signaling server 3101 and a transport server 3102 communicate over a link. Signaling server 3101 contains a large number of small tables 3106 and 3107 that contain enough information to authenticate a communication request with one or more clients 3103 and 3104. As described in more detail below, these small tables may advantageously be constructed as a special case of the synchronizing checkpoint tables described previously. Transport server 3102, which is preferably a separate computer in communication with signaling server 3101, contains a smaller number of larger hopping tables 3108, 3109, and 3110 that can be allocated to create a VPN with one of the client computers.

According to one embodiment, a client that has previously registered with the system (e.g., via a system administration function, a user registration procedure, or some other method) transmits a request for information from a computer (e.g., a web site). In one variation, the request is made using a "hopped" packet, such that signaling server 3101 will quickly reject invalid packets from unauthorized computers such as hacker computer 3105. An "administrative" VPN can be established between all of the clients and the signaling server in order to ensure that a hacker cannot flood signaling server 3101 with bogus packets. Details of this scheme are provided below.

Signaling server 3101 receives the request 3111 and uses it to determine that client 3103 is a validly registered user. Next, signaling server 3101 issues a request to transport server 3102 to allocate a hopping table (or hopping algorithm or other regime) for the purpose of creating a VPN with client 3103. The allocated hopping parameters are returned to signaling server 3101 (path 3113), which then supplies the hopping parameters to client 3103 via path 3114, preferably in encrypted form.

Thereafter, client 3103 communicates with transport server 3102 using the normal hopping techniques described above. It will be appreciated that although signaling server 3101 and transport server 3102 are illustrated as being two separate computers, they could of course be combined into a single computer and their functions performed on the single computer. Alternatively, it is possible to partition the functions shown in FIG. 31 differently from as shown without departing from the inventive principles.

One advantage of the above-described architecture is that signaling server 3101 need only maintain a small amount of information on a large number of potential users, yet it retains the capability of quickly rejecting packets from unauthorized users such as hacker computer 3105. Larger data tables needed to perform the hopping and synchronization functions are instead maintained in a transport server 3102, and a smaller number of these tables are needed since they are only allocated for "active" links. After a VPN has become inactive for a certain time period (e.g., one hour), the VPN can be automatically torn down by transport server 3102 or signaling server 3101.

A more detailed description will now be provided regarding how a special case of the checkpoint synchronization feature can be used to implement the signaling scheme described above.

The signaling synchronizer may be required to support many (millions) of standing, low bandwidth connections. It therefore should minimize per-VPL memory usage while providing the security offered by hopping technology. In order to reduce memory usage in the signaling server, the data hopping tables can be completely eliminated and data can be carried as part of the SYNC_REQ message. The table used by the server side (receiver) and client side (transmitter) is shown schematically as element 3106 in FIG. 31.

The meaning and behaviors of CKPT_N, CKPT_O and CKPT_R remain the same from the previous description, except that CKPT_N can receive a combined data and SYNC_REQ message or a SYNC_REQ message without the data.

The protocol is a straightforward extension of the earlier synchronizer. Assume that a client transmitter is on and the tables are synchronized. The initial tables can be generated "out of band." For example, a client can log into a web server to establish an account over the Internet. The client will receive keys etc encrypted over the Internet. Meanwhile, the server will set up the signaling VPN on the signaling server.

Assuming that a client application wishes to send a packet to the server on the client's standing signaling VPL:

1. The client sends the message marked as a data message on the inner header using the transmitter's CKPT_N address. It turns the transmitter off and starts a timer T1 noting CKPT_O. Messages can be one of three types: DATA, SYNC_REQ and SYNC_ACK. In the normal algorithm, some potential problems can be prevented by identifying each message type as part of the encrypted inner header field. In this algorithm, it is important to distinguish a data packet and a SYNC_REQ in the signaling synchronizer since the data and the SYNC_REQ come in on the same address.

2. When the server receives a data message on its CKPT_N, it verifies the message and passes it up the stack. The message can be verified by checking message type and and other information (i.e., user credentials) contained in the inner header It replaces its CKPT_O with CKPT_N and generates the next CKPT_N. It updates its transmitter side CKPT_R to correspond to the client's receiver side CKPT_R and transmits a SYNC_ACK containing CKPT_O in its payload.

3. When the client side receiver receives a SYNC_ACK on its CKPT_R with a payload matching its transmitter side CKPT_O and the transmitter is off, the transmitter is turned on and the receiver side CKPT_R is updated. If the SYNC_ACK's payload does not match the transmitter side CKPT_O or the transmitter is on, the SYNC_ACK is simply discarded.

4. T1 expires: If the transmitter is off and the client's transmitter side CKPT_O matches the CKPT_O associated with the timer, it starts timer T1 noting CKPT_O again, and a SYNC_REQ is sent using the transmitter's CKPT_O address. Otherwise, no action is taken.

5. When the server receives a SYNC_REQ on its CKPT_N, it replaces its CKPT_O with CKPT_N and generates the next CKPT_N. It updates its transmitter side CKPT_R to correspond to the client's receiver side CKPT_R and transmits a SYNC_ACK containing CKPT_O in its payload.

6. When the server receives a SYNC_REQ on its CKPT_O, it updates its transmitter side CKPT_R to correspond to the client's receiver side CKPT_R and transmits a SYNC_ACK containing CKPT_O in its payload.

FIG. 32 shows message flows to highlight the protocol. Reading from top to bottom, the client sends data to the server using its transmitter side CKPT_N. The client side transmitter is

turned off and a retry timer is turned off. The transmitter will not transmit messages as long as the transmitter is turned off. The client side transmitter then loads CKPT_N into CKPT_O and updates CKPT_N. This message is successfully received and a passed up the stack. It also synchronizes the receiver i.e., the server loads CKPT_N into CKPT_O and generates a new CKPT_N, it generates a new CKPT_R in the server side transmitter and transmits a SYNC_ACK containing the server side receiver's CKPT_O the server. The SYNC_ACK is successfully received at the client. The client side receiver's CKPT_R is updated, the transmitter is turned on and the retry timer is killed. The client side transmitter is ready to transmit a new data message.

Next, the client sends data to the server using its transmitter side CKPT_N. The client side transmitter is turned off and a retry timer is turned off. The transmitter will not transmit messages as long as the transmitter is turned off. The client side transmitter then loads CKPT_N into CKPT_O and updates CKPT_N. This message is lost. The client side timer expires and as a result a SYNC_REQ is transmitted on the client side transmitter's CKPT_O (this will keep happening until the SYNC_ACK has been received at the client). The SYNC_REQ is successfully received at the server. It synchronizes the receiver i.e., the server loads CKPT_N into CKPT_O and generates a new CKPT_N, it generates an new CKPT_R in the server side transmitter and transmits a SYNC_ACK containing the server side receiver's CKPT_O the server. The SYNC_ACK is successfully received at the client. The client side receiver's CKPT_R is updated, the transmitter is turned off and the retry timer is killed. The client side transmitter is ready to transmit a new data message.

There are numerous other scenarios that follow this flow. For example, the SYNC_ACK could be lost. The transmitter would continue to re-send the SYNC_REQ until the receiver synchronizes and responds.

The above-described procedures allow a client to be authenticated at signaling server 3201 while maintaining the ability of signaling server 3201 to quickly reject invalid packets, such as might be generated by hacker computer 3205. In various embodiments, the signaling synchronizer is really a derivative of the synchronizer. It provides the same protection as the hopping protocol, and it does so for a large number of low bandwidth connections.

F. One-Click Secure On-line Communications and Secure Domain Name Service

The present invention provides a technique for establishing a secure communication link between a first computer and a second computer over a computer network. Preferably, a user

enables a secure communication link using a single click of a mouse, or a corresponding minimal input from another input device, such as a keystroke entered on a keyboard or a click entered through a trackball. Alternatively, the secure link is automatically established as a default setting at boot-up of the computer (i.e., no click). FIG. 33 shows a system block diagram 3300 of a computer network in which the one-click secure communication method of the present invention is suitable. In FIG. 33, a computer terminal or client computer 3301, such as a personal computer (PC), is connected to a computer network 3302, such as the Internet, through an ISP 3303. Alternatively, computer 3301 can be connected to computer network 3302 through an edge router. Computer 3301 includes an input device, such as a keyboard and/or mouse, and a display device, such as a monitor. Computer 3301 can communicate conventionally with another computer 3304 connected to computer network 3302 over a communication link 3305 using a browser 3306 that is installed and operates on computer 3301 in a well-known manner.

Computer 3304 can be, for example, a server computer that is used for conducting e-commerce. In the situation when computer network 3302 is the Internet, computer 3304 typically will have a standard top-level domain name such as .com, .net, .org, .edu, .mil or .gov.

FIG. 34 shows a flow diagram 3400 for installing and establishing a "one-click" secure communication link over a computer network according to the present invention. At step 3401, computer 3301 is connected to server computer 3304 over a non-VPN communication link 3305. Web browser 3306 displays a web page associated with server 3304 in a well-known manner. According to one variation of the invention, the display of computer 3301 contains a hyperlink, or an icon representing a hyperlink, for selecting a virtual private network (VPN) communication link ("go secure" hyperlink) through computer network 3302 between terminal 3301 and server 3304. Preferably, the "go secure" hyperlink is displayed as part of the web page downloaded from server computer 3304, thereby indicating that the entity providing server 3304 also provides VPN capability.

By displaying the "go secure" hyperlink, a user at computer 3301 is informed that the current communication link between computer 3301 and server computer 3304 is a non-secure, non-VPN communication link. At step 3402, it is determined whether a user of computer 3301 has selected the "go secure" hyperlink. If not, processing resumes using a non-secure (conventional) communication method (not shown). If, at step 3402, it is determined that the user has selected the "go secure" hyperlink, flow continues to step 3403 where an object

associated with the hyperlink determines whether a VPN communication software module has already been installed on computer 3301. Alternatively, a user can enter a command into computer 3301 to "go secure."

If, at step 3403, the object determines that the software module has been installed, flow continues to step 3407. If, at step 3403, the object determines that the software module has not been installed, flow continues to step 3404 where a non-VPN communication link 3307 is launched between computer 3301 and a website 3308 over computer network 3302 in a well-known manner. Website 3308 is accessible by all computer terminals connected to computer network 3302 through a non-VPN communication link. Once connected to website 3308, a software module for establishing a secure communication link over computer network 3302 can be downloaded and installed. Flow continues to step 3405 where, after computer 3301 connects to website 3308, the software module for establishing a communication link is downloaded and installed in a well-known manner on computer terminal 3301 as software module 3309. At step 3405, a user can optionally select parameters for the software module, such as enabling a secure communication link mode of communication for all communication links over computer network 3302. At step 3406, the -communication link between computer 3301 and website 3308 is then terminated in a well-known manner.

By clicking on the "go secure" hyperlink, a user at computer 3301 has enabled a secure communication mode of communication between computer 3301 and server computer 3304. According to one variation of the invention, the user is not required to do anything more than merely click the "go secure" hyperlink. The user does not need to enter any user identification information, passwords or encryption keys for establishing a secure communication link. All procedures required for establishing a secure communication link between computer 3301 and server computer 3304 are performed transparently to a user at computer 3301.

At step 3407, a secure VPN communications mode of operation has been enabled and software module 3309 begins to establish a VPN communication link. In one embodiment, software module 3309 automatically replaces the top-level domain name for server 3304 within browser 3406 with a secure top-level domain name for server computer 3304. For example, if the top-level domain name for server 3304 is .com, software module 3309 replaces the .com top-level domain name with a .scom top-level domain name, where the "s" stands for secure. Alternatively, software module 3409 can replace the top-level domain name of server 3304 with

any other non-standard top-level domain name.

Because the secure top-level domain name is a non-standard domain name, a query to a standard domain name service (DNS) will return a message indicating that the universal resource locator (URL) is unknown. According to the invention, software module 3409 contains the URL for querying a secure domain name service (SDNS) for obtaining the URL for a secure top-level domain name. In this regard, software module 3309 accesses a secure portal 3310 that interfaces a secure network 3311 to computer network 3302. Secure network 3311 includes an internal router 3312, a secure domain name service (SDNS) 3313, a VPN gatekeeper 3314 and a secure proxy 3315. The secure network can include other network services, such as e-mail 3316, a plurality of chatrooms (of which only one chatroom 3317 is shown), and a standard domain name service (STD DNS) 3318. Of course, secure network 3311 can include other resources and services that are not shown in FIG. 33.

When software module 3309 replaces the standard top-level domain name for server 3304 with the secure top-level domain name, software module 3309 sends a query to SDNS 3313 at step 3408 through secure portal 3310 preferably using an administrative VPN communication link 3319. In this configuration, secure portal 3310 can only be accessed using a VPN communication link. Preferably, such a VPN communication link can be based on a technique of inserting a source and destination IP address pair into each data packet that is selected according to a pseudo-random sequence; an IP address hopping regime that pseudorandomly changes IP addresses in packets transmitted between a client computer and a secure target computer; periodically changing at least one field in a series of data packets according to a known sequence; an Internet Protocol (IP) address in a header of each data packet that is compared to a table of valid IP addresses maintained in a table in the second computer; and/or a comparison of the IP address in the header of each data packet to a moving window of valid IP addresses, and rejecting data packets having IP addresses that do not fall within the moving window. Other types of VPNs can alternatively be used. Secure portal 3310 authenticates the query from software module 3309 based on the particular information hopping technique used for VPN communication link 3319.

SDNS 3313 contains a cross-reference database of secure domain names and corresponding secure network addresses. That is, for each secure domain name, SDNS 3313 stores a computer network address corresponding to the secure domain name. An entity can

register a secure domain name in SDNS 3313 so that a user who desires a secure communication link to the website of the entity can automatically obtain the secure computer network address for the secure website. Moreover, an entity can register several secure domain names, with each respective secure domain name representing a different priority level of access in a hierarchy of access levels to a secure website. For example, a securities trading website can provide users secure access so that a denial of service attack on the website will be ineffectual with respect to users subscribing to the secure website service. Different levels of subscription can be arranged based on, for example, an escalating fee, so that a user can select a desired level of guarantee for connecting to the secure securities trading website. When a user queries SDNS 3313 for the secure computer network address for the securities trading website, SDNS 3313 determines the particular secure computer network address based on the user's identity and the user's subscription level.

At step 3409, SDNS 3313 accesses VPN gatekeeper 3314 for establishing a VPN communication link between software module 3309 and secure server 3320. Server 3320 can only be accessed through a VPN communication link. VPN gatekeeper 3314 provisions computer 3301 and secure web server computer 3320, or a secure edge router for server computer 3320, thereby creating the VPN. Secure server computer 3320 can be a separate server computer from server computer 3304, or can be the same server computer having both non-VPN and VPN communication link capability, such as shown by server computer 3322. Returning to FIG. 34, in step 3410, SDNS 3313 returns a secure URL to software module 3309 for the .scom server address for a secure server 3320 corresponding to server 3304.

Alternatively, SDNS 3313 can be accessed through secure portal 3310 "in the clear", that is, without using an administrative VPN communication link. In this situation, secure portal 3310 preferably authenticates the query using any well-known technique, such as a cryptographic technique, before allowing the query to proceed to SDNS 3319. Because the initial communication link in this situation is not a VPN communication link, the reply to the query can be "in the clear." The querying computer can use the clear reply for establishing a VPN link to the desired domain name. Alternatively, the query to SDNS 3313 can be in the clear, and SDNS 3313 and gatekeeper 3314 can operate to establish a VPN communication link to the querying computer for sending the reply.

At step 3411, software module 3309 accesses secure server 3320 through VPN

communication link 3321 based on the VPN resources allocated by VPN gatekeeper 3314. At step 3412, web browser 3306 displays a secure icon indicating that the current communication link to server 3320 is a secure VPN communication link. Further communication between computers 3301 and 3320 occurs via the VPN, e.g., using a "hopping" regime as discussed above. When VPN link 3321 is terminated at step 3413, flow continues to step 3414 where software module 3309 automatically replaces the secure top-level domain name with the corresponding non-secure top-level domain name for server 3304. Browser 3306 accesses a standard DNS 3325 for obtaining the non-secure URL for server 3304. Browser 3306 then connects to server 3304 in a well-known manner. At step 3415, browser 3306 displays the "go secure" hyperlink or icon for selecting a VPN communication link between terminal 3301 and server 3304. By again displaying the "go secure" hyperlink, a user is informed that the current communication link is a non-secure, non-VPN communication link.

When software module 3309 is being installed or when the user is off-line, the user can optionally specify that all communication links established over computer network 3302 are secure communication links. Thus, anytime that a communication link is established, the link is a VPN link. Consequently, software module 3309 transparently accesses SDNS 3313 for obtaining the URL for a selected secure website. In other words, in one embodiment, the user need not "click" on the secure option each time secure communication is to be effected.

Additionally, a user at computer 3301 can optionally select a secure communication link through proxy computer 3315. Accordingly, computer 3301 can establish a VPN communication link 3323 with secure server computer 3320 through proxy computer 3315. Alternatively, computer 3301 can establish a non-VPN communication link 3324 to a non-secure website, such as non-secure server computer 3304.

FIG. 35 shows a flow diagram 3500 for registering a secure domain name according to the present invention. At step 3501, a requester accesses website 3308 and logs into a secure domain name registry service that is available through website 3308. At step 3502, the requestor completes an online registration form for registering a secure domain name having a top-level domain name, such as .com, .net, .org, .edu, .mil or .gov. Of course, other secure top-level domain names can also be used. Preferably, the requestor must have previously registered a non-secure domain name corresponding to the equivalent secure domain name that is being

requested. For example, a requestor attempting to register secure domain name "website.scom" must have previously registered the corresponding non-secure domain name "website.com".

At step 3503, the secure domain name registry service at website 3308 queries a non-secure domain name server database, such as standard DNS 3322, using, for example, a whois query, for determining ownership information relating to the non-secure domain name corresponding to the requested secure domain name. At step 3504, the secure domain name registry service at website 3308 receives a reply from standard DNS 3322 and at step 3505 determines whether there is conflicting ownership information for the corresponding non-secure domain name. If there is no conflicting ownership information, flow continues to step 3507, otherwise flow continues to step 3506 where the requestor is informed of the conflicting ownership information. Flow returns to step 3502.

When there is no conflicting ownership information at step 3505, the secure domain name registry service (website 3308) informs the requestor that there is no conflicting ownership information and prompts the requestor to verify the information entered into the online form and select an approved form of payment. After confirmation of the entered information and appropriate payment information, flow continues to step 3508 where the newly registered secure domain name sent to SDNS 3313 over communication link 3326.

If, at step 3505, the requested secure domain name does not have a corresponding equivalent non-secure domain name, the present invention informs the requestor of the situation and prompts the requestor for acquiring the corresponding equivalent non-secure domain name for an increased fee. By accepting the offer, the present invention automatically registers the corresponding equivalent non-secure domain name with standard DNS 3325 in a well-known manner. Flow then continues to step 3508.

G. Tunneling Secure Address Hopping Protocol Through Existing Protocol Using Web Proxy

The present invention also provides a technique for implementing the field hopping schemes described above in an application program on the client side of a firewall between two computer networks, and in the network stack on the server side of the firewall. The present invention uses a new secure connectionless protocol that provides good denial of service rejection capabilities by layering the new protocol on top of an existing IP protocol, such as the ICMP, UDP or TCP protocols. Thus, this aspect of the present invention does not require

changes in the Internet infrastructure.

According to the invention, communications are protected by a client-side proxy application program that accepts unencrypted, unprotected communication packets from a local browser application. The client-side proxy application program tunnels the unencrypted, unprotected communication packets through a new protocol, thereby protecting the communications from a denial of service at the server side. Of course, the unencrypted, unprotected communication packets can be encrypted prior to tunneling.

The client-side proxy application program is not an operating system extension and does not involve any modifications to the operating system network stack and drivers. Consequently, the client is easier to install, remove and support in comparison to a VPN. Moreover, the client-side proxy application can be allowed through a corporate firewall using a much smaller "hole" in the firewall and is less of a security risk in comparison to allowing a protocol layer VPN through a corporate firewall.

The server-side implementation of the present invention authenticates valid field-hopped packets as valid or invalid very early in the server packet processing, similar to a standard virtual private network, for greatly minimizing the impact of a denial of service attempt in comparison to normal TCP/IP and HTTP communications, thereby protecting the server from invalid communications.

FIG. 36 shows a system block diagram of a computer network 3600 in which a virtual private connection according to the present invention can be configured to more easily traverse a firewall between two computer networks. FIG. 37 shows a flow diagram 3700 for establishing a virtual private connection that is encapsulated using an existing network protocol.

In FIG. 36 a local area network (LAN) 3601 is connected to another computer network 3602, such as the Internet, through a firewall arrangement 3603. Firewall arrangement operates in a well-known manner to interface LAN 3601 to computer network 3602 and to protect LAN 3601 from attacks initiated outside of LAN 3601.

A client computer 3604 is connected to LAN 3601 in a well-known manner. Client computer 3604 includes an operating system 3605 and a web browser 3606. Operating system 3605 provides kernel mode functions for operating client computer 3604. Browser 3606 is an application program for accessing computer network resources connected to LAN 3601 and computer network 3602 in a well-known manner. According to the present invention, a proxy

application 3607 is also stored on client computer 3604 and operates at an application layer in conjunction with browser 3606. Proxy application 3607 operates at the application layer within client computer 3604 and when enabled, modifies unprotected, unencrypted message packets generated by browser 3606 by inserting data into the message packets that are used for forming a virtual private connection between client computer 3604 and a server computer connected to LAN 3601 or computer network 3602. According to the invention, a virtual private connection does not provide the same level of security to the client computer as a virtual private network. A virtual private connection can be conveniently authenticated so that, for example, a denial of service attack can be rapidly rejected, thereby providing different levels of service that can be subscribed to by a user.

Proxy application 3607 is conveniently installed and uninstalled by a user because proxy application 3607 operates at the application layer within client computer 3604. On installation, proxy application 3607 preferably configures browser 3606 to use proxy application for all web communications. That is, the payload portion of all message packets is modified with the data for forming a virtual private connection between client computer 3604 and a server computer. Preferably, the data for forming the virtual private connection contains field-hopping data, such as described above in connection with VPNs. Also, the modified message packets preferably conform to the UDP protocol. Alternatively, the modified message packets can conform to the TCP/IP protocol or the ICMP protocol. Alternatively, proxy application 3606 can be selected and enabled through, for example, an option provided by browser 3606. Additionally, proxy application 3607 can be enabled so that only the payload portion of specially designated message packets is modified with the data for forming a virtual private connection between client computer 3604 and a designated host computer. Specially designated message packets can be, for example, selected predetermined domain names.

Referring to FIG. 37, at step 3701, unprotected and unencrypted message packets are generated by browser 3606. At step 3702, proxy application 3607 modifies the payload portion of all message packets by tunneling the data for forming a virtual private connection between client computer 3604 and a destination server computer into the payload portion. At step, 3703, the modified message packets are sent from client computer 3604 to, for example, website (server computer) 3608 over computer network 3602.

Website 3608 includes a VPN guard portion 3609, a server proxy portion 3610 and a web server portion 3611. VPN guard portion 3609 is embedded within the kernel layer of the operating system of website 3608 so that large bandwidth attacks on website 3608 are rapidly rejected. When client computer 3604 initiates an authenticated connection to website 3608, VPN guard portion 3609 is keyed with the hopping sequence contained in the message packets from client computer 3604, thereby performing a strong authentication of the client packet streams entering website 3608 at step 3704. VPN guard portion 3609 can be configured for providing different levels of authentication and, hence, quality of service, depending upon a subscribed level of service. That is, VPN guard portion 3609 can be configured to let all message packets through until a denial of service attack is detected, in which case VPN guard portion 3609 would allow only client packet streams conforming to a keyed hopping sequence, such as that of the present invention.

Server proxy portion 3610 also operates at the kernel layer within website 3608 and catches incoming message packets from client computer 3604 at the VPN level. At step 3705, server proxy portion 3610 authenticates the message packets at the kernel level within host computer 3604 using the destination IP address, UDP ports and discriminator fields. The authenticated message packets are then forwarded to the authenticated message packets to web server portion 3611 as normal TCP web transactions.

At step 3705, web server portion 3611 responds to message packets received from client computer 3604 in accordance with the particular nature of the message packets by generating reply message packets. For example, when a client computer requests a webpage, web server portion 3611 generates message packets corresponding to the requested webpage. At step 3706, the reply message packets pass through server proxy portion 3610, which inserts data into the payload portion of the message packets that are used for forming the virtual private connection between host computer 3608 and client computer 3604 over computer network 3602. Preferably, the data for forming the virtual private connection is contains field-hopping data, such as described above in connection with VPNs. Server proxy portion 3610 operates at the kernel layer within host computer 3608 to insert the virtual private connection data into the payload portion of the reply message packets. Preferably, the modified message packets sent by host computer 3608 to client computer 3604 conform to the UDP protocol. Alternatively, the modified message packets can conform to the TCP/IP protocol or the ICMP protocol.

At step 3707, the modified packets are sent from host computer 3608 over computer network 3602 and pass through firewall 3603. Once through firewall 3603, the modified packets are directed to client computer 3604 over LAN 3601 and are received at step 3708 by proxy application 3607 at the application layer within client computer 3604. Proxy application 3607 operates to rapidly evaluate the modified message packets for determining whether the received packets should be accepted or dropped. If the virtual private connection data inserted into the received information packets conforms to expected virtual private connection data, then the received packets are accepted. Otherwise, the received packets are dropped.

While the present invention has been described in connection with the illustrated embodiments, it will be appreciated and understood that modifications may be made without departing from the true spirit and scope of the invention.

CLAIMS

What is claimed is:

1. A method for accessing a secure computer network address, comprising steps of:

receiving a secure domain name;

sending a query message to a secure domain name service, the query message requesting a secure computer network address corresponding to the secure domain name;

receiving a response message containing the secure computer network address corresponding to the secure domain name; and

sending an access request message to the secure computer network address using a virtual private network communication link.

2. The method according to claim 1, wherein the step of receiving the secure domain name includes steps of:

receiving a command to establish the virtual private network communication link with a secure computer network address corresponding to a predetermined non-secure domain name; and

automatically generating a secure domain name corresponding to the non-secure domain name.

3. The method according to claim 2, wherein the step of receiving a command to establish the virtual private network communication link includes a step of selecting a predetermined icon displayed on a computer display.

4. The method according to claim 1, wherein the response message contains provisioning information for the virtual private network.

5. The method according to claim 4, wherein the virtual private network is based on inserting one or more data values into each data packet sent to the secure computer network address, the one or more data values varying according to a pseudo-random sequence.

6. The method according to claim 4, wherein the virtual private network is based on inserting into at least one data packet at least one data value representing a predetermined level of service associated with the virtual private network.

7. The method according to claim 4, wherein the virtual private network is based on a computer network address hopping regime that is used to pseudorandomly change computer network addresses in packets transmitted between the first computer and the second computer.

8. The method according to claim 4, wherein the virtual private network is based on comparing a value in each data packet transmitted to the secure computer network address to a moving window of valid values.

9. The method according to claim 4, wherein the virtual private network is based on a comparison of a discriminator field in a header of each data packet to the secure computer network address to a table of valid discriminator fields.

10. The method according to claim 1, wherein the computer network includes the Internet.

11. The method according to claim 1, wherein the secure domain name has a top-level domain name that includes one of .scom, .snet, .sorg, .sedu, .smil or .sgov.

12. A computer-readable storage medium, comprising:

a storage area; and

computer-readable instructions for a method for accessing a secure computer network address, the method comprising steps of:

receiving a secure domain name;

sending a query message to a secure domain name service, the query message requesting a secure computer network address corresponding to the secure domain name;

receiving a response message containing the secure computer network address corresponding to the secure domain name; and

sending an access request message to the secure computer network address using a virtual private network communication link.

13. The computer-readable medium according to claim 12, wherein the step of receiving the secure domain name includes steps of:

receiving a command to establish the virtual private network communication link with a secure computer network address corresponding to a predetermined non-secure domain name; and

automatically generating a secure domain name corresponding to the non-secure domain name.

14. The computer-readable medium according to claim 13, wherein the step of receiving a command to establish the virtual private network communication link includes a step of selecting a predetermined icon displayed on a computer display.

15. The computer-readable medium according to claim 12, wherein the response message contains provisioning information for the virtual private network.

16. The computer-readable medium according to claim 15, wherein the virtual private network is based on inserting one or more data values into each data packet sent to the secure computer network address, the one or more data values varying according to a pseudo-random sequence.

17. The computer-readable medium according to claim 15, wherein the virtual private network is based on inserting into at least one data packet at least one data value representing a predetermined level of service associated with the virtual private network.

18. The computer-readable medium according to claim 15, wherein the virtual private network is based on a computer network address hopping regime that is used to pseudorandomly change computer network addresses in packets transmitted between the first computer and the second computer.

19. The computer-readable medium according to claim 15, wherein the virtual private network is based on comparing a value in each data packet transmitted to the secure computer network address to a moving window of valid values.

20. The computer-readable medium according to claim 15, wherein the virtual private network is based on a comparison of a discriminator field in a header of each data packet to the secure computer network address to a table of valid discriminator fields.

21. The computer-readable medium according to claim 12, wherein the computer network includes the Internet.

22. The computer-readable medium according to claim 12, wherein the secure domain name has a top-level domain name that includes one of .scom, .snet, .sorg, .sedu, .smil or .sgov.

23. The method of claim 1, wherein the access request message containing a request for information stored at the secure computer network address.

24. The computer readable medium of claim 12, wherein the access request message containing a request for information stored at the secure computer network address.

ABSTRACT

A technique is disclosed for establishing a secure communication link between a first computer and a second computer over a computer network. Initially, a secure communication mode of communication is enabled at a first computer without a user entering any cryptographic information for establishing the secure communication mode of communication. Then, a secure communication link is established between the first computer and a second computer over a computer network based on the enabled secure communication mode of communication. The secure communication link is a virtual private network communication link over the computer network in which one or more data values that vary according to a pseudo-random sequence are inserted into each data packet.

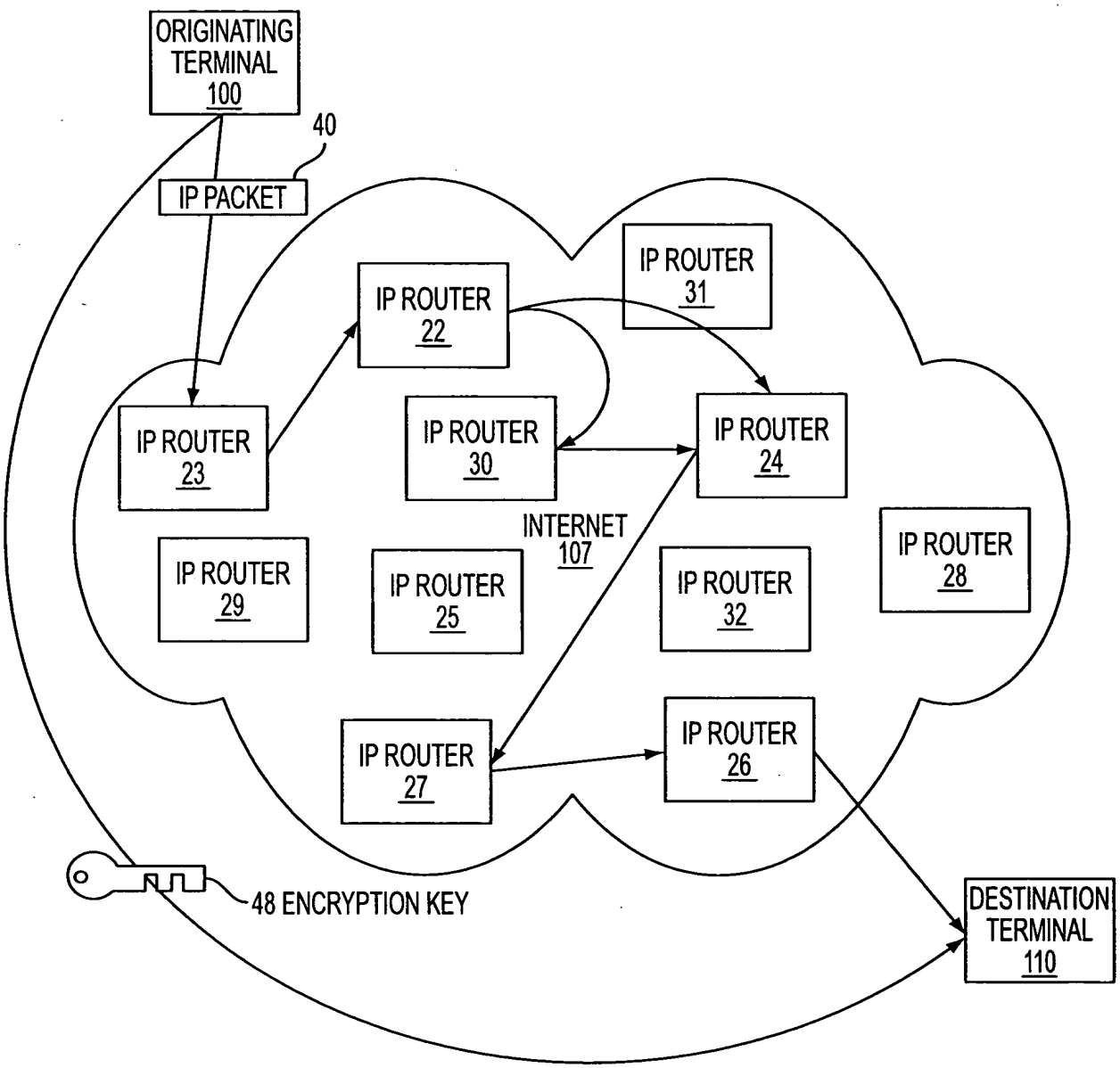

ORIGINATING TERMINAL 100
40
IP PACKET
IP ROUTER 22
IP ROUTER 31
IP ROUTER 23
IP ROUTER 30
IP ROUTER 24
INTERNET 107
IP ROUTER 29
IP ROUTER 25
IP ROUTER 32
IP ROUTER 28
IP ROUTER 27
IP ROUTER 26
48 ENCRYPTION KEY
DESTINATION TERMINAL 110

FIG. 1

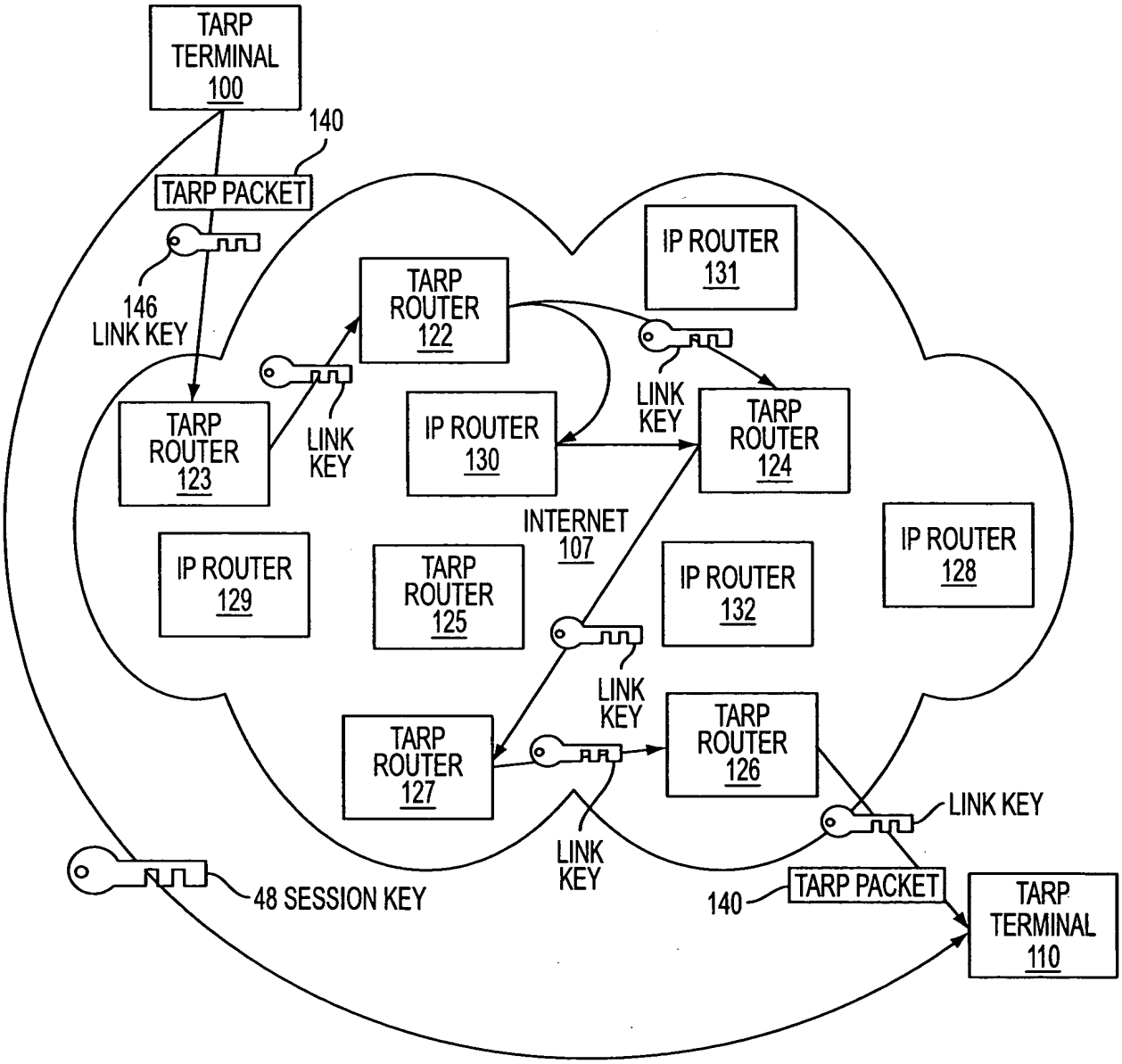

TARP TERMINAL 100
140
TARP PACKET
146 LINK KEY
TARP ROUTER 122
IP ROUTER 131
LINK KEY
TARP ROUTER 123
LINK KEY
IP ROUTER 130
TARP ROUTER 124
INTERNET 107
IP ROUTER 129
TARP ROUTER 125
IP ROUTER 132
IP ROUTER 128
LINK KEY
TARP ROUTER 127
TARP ROUTER 126
LINK KEY
48 SESSION KEY
LINK KEY
140
TARP PACKET
TARP TERMINAL 110

FIG. 2

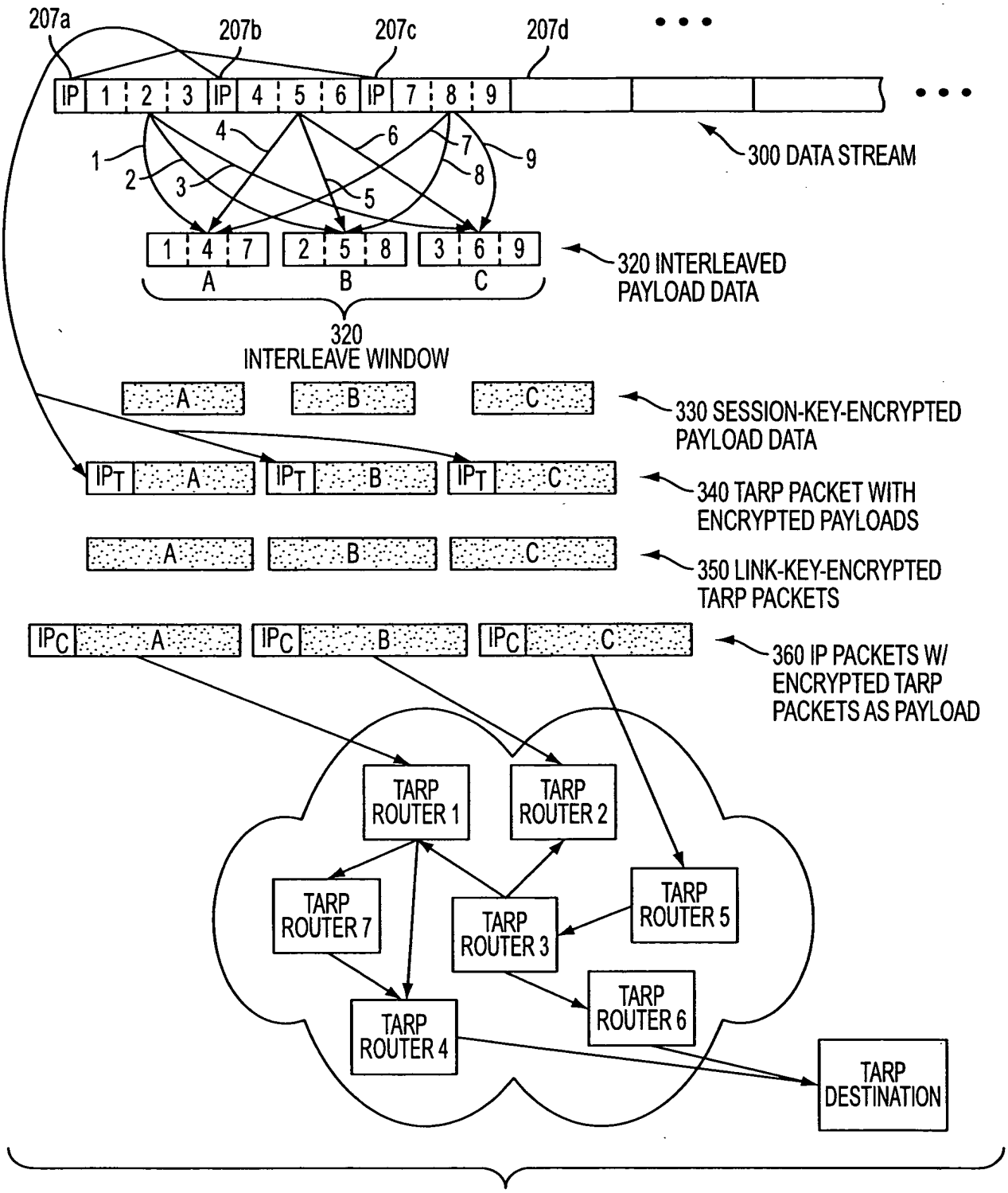

207a
207b
207c
207d
IP 1 2 3 IP 4 5 6 IP 7 8 9
300 DATA STREAM
1 4 7
2 5 8
3 6 9
A
B
C
320 INTERLEAVED PAYLOAD DATA
320 INTERLEAVE WINDOW
330 SESSION-KEY-ENCRYPTED PAYLOAD DATA
IPT A IPT B IPT C
340 TARP PACKET WITH ENCRYPTED PAYLOADS
350 LINK-KEY-ENCRYPTED TARP PACKETS
IPC A IPC B IPC C
360 IP PACKETS W/ ENCRYPTED TARP PACKETS AS PAYLOAD
TARP ROUTER 1
TARP ROUTER 2
TARP ROUTER 7
TARP ROUTER 3
TARP ROUTER 5
TARP ROUTER 4
TARP ROUTER 6
TARP DESTINATION

FIG. 3A

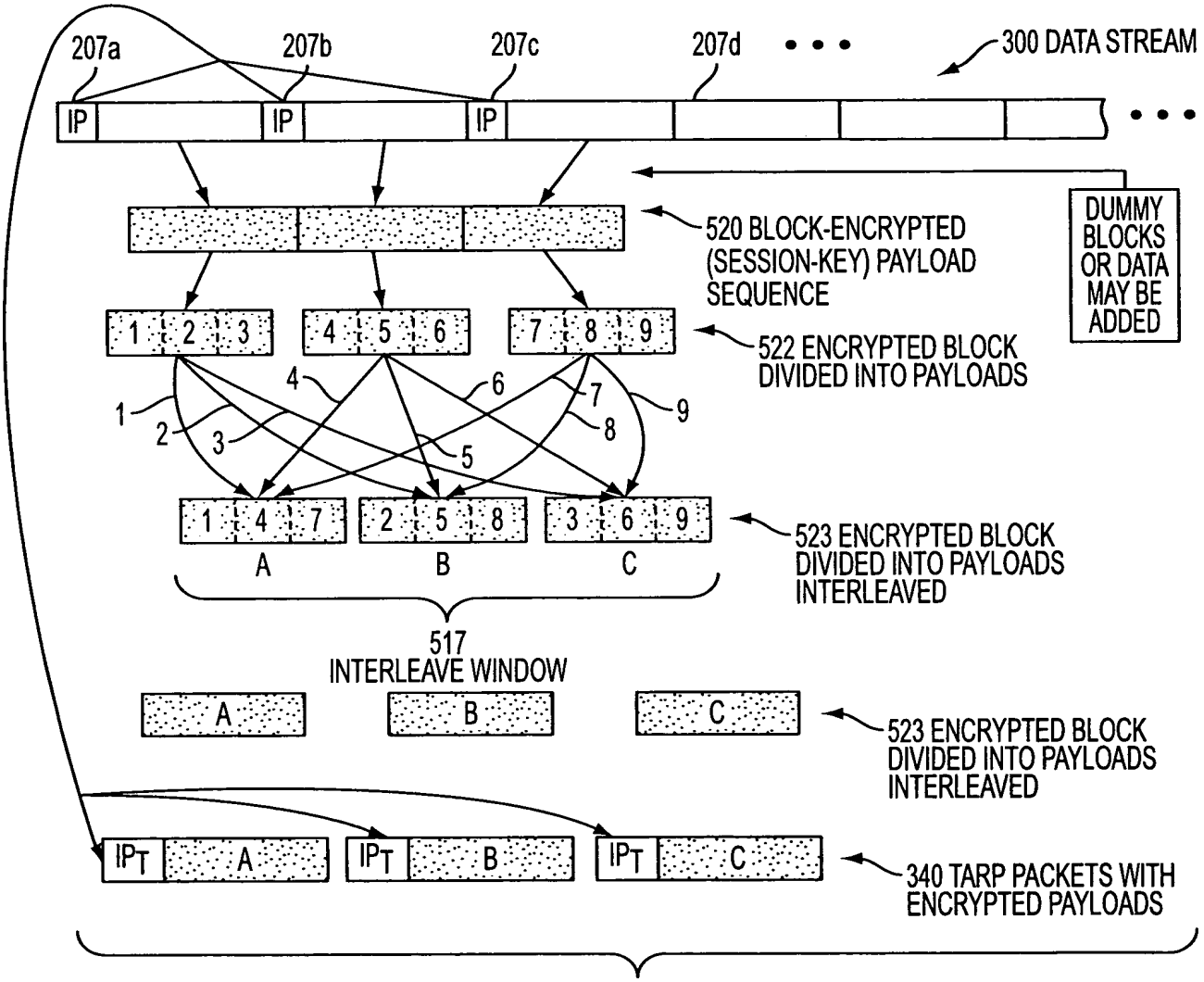

207a
207b
207c
207d
300 DATA STREAM
IP
IP
IP
520 BLOCK-ENCRYPTED (SESSION-KEY) PAYLOAD SEQUENCE
DUMMY BLOCKS OR DATA MAY BE ADDED
1 2 3
4 5 6
7 8 9
522 ENCRYPTED BLOCK DIVIDED INTO PAYLOADS
1 4 7
2 5 8
3 6 9
A
B
C
523 ENCRYPTED BLOCK DIVIDED INTO PAYLOADS INTERLEAVED
517
INTERLEAVE WINDOW
A
B
C
523 ENCRYPTED BLOCK DIVIDED INTO PAYLOADS INTERLEAVED
IPT A
IPT B
IPT C
340 TARP PACKETS WITH ENCRYPTED PAYLOADS

FIG. 3B

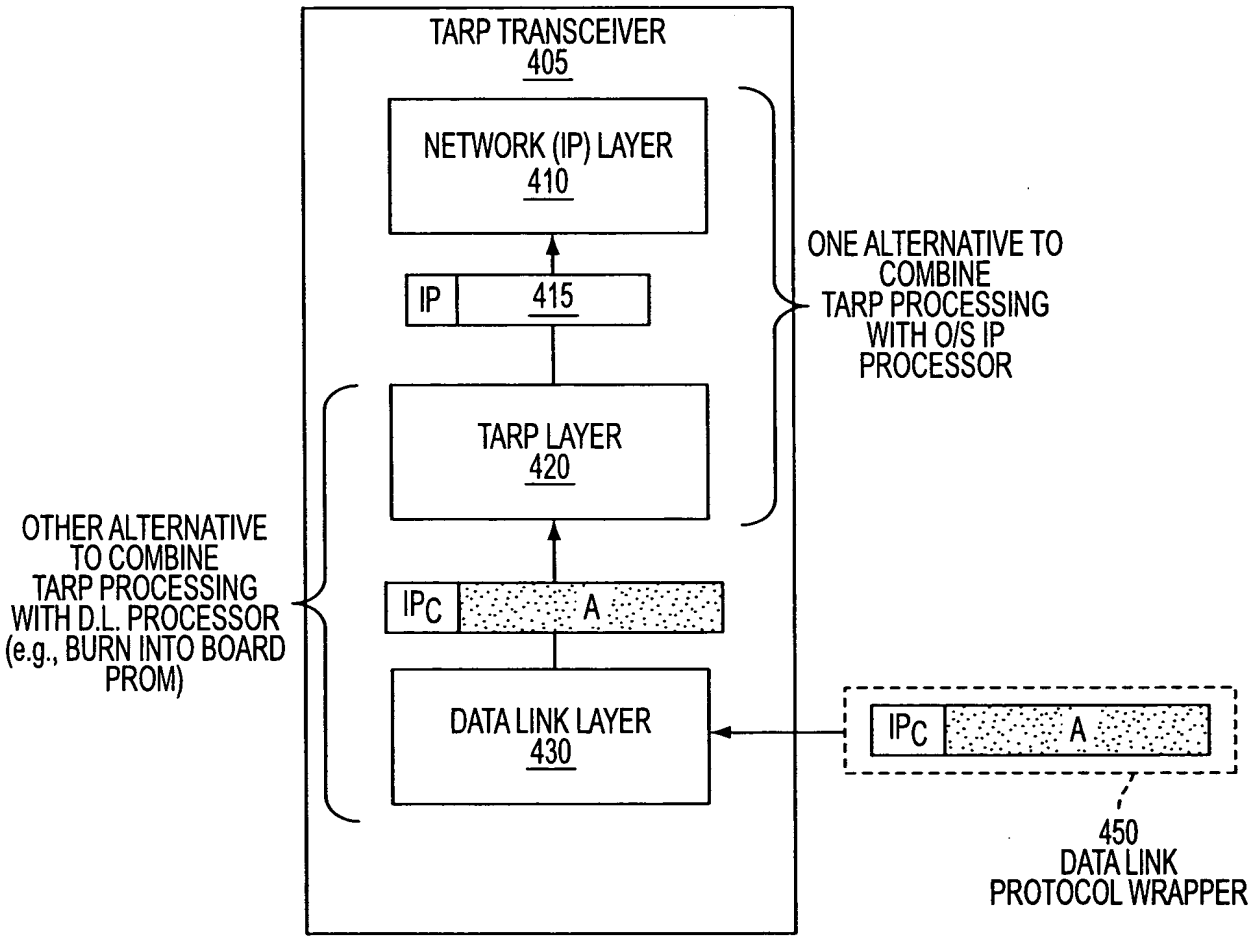

TARP TRANSCEIVER
405
NETWORK (IP) LAYER
410
IP
415
TARP LAYER
420
IPC
A
DATA LINK LAYER
430
ONE ALTERNATIVE TO COMBINE TARP PROCESSING WITH O/S IP PROCESSOR
OTHER ALTERNATIVE TO COMBINE TARP PROCESSING WITH D.L. PROCESSOR (e.g., BURN INTO BOARD PROM)
IPC
A
450
DATA LINK PROTOCOL WRAPPER

FIG. 4

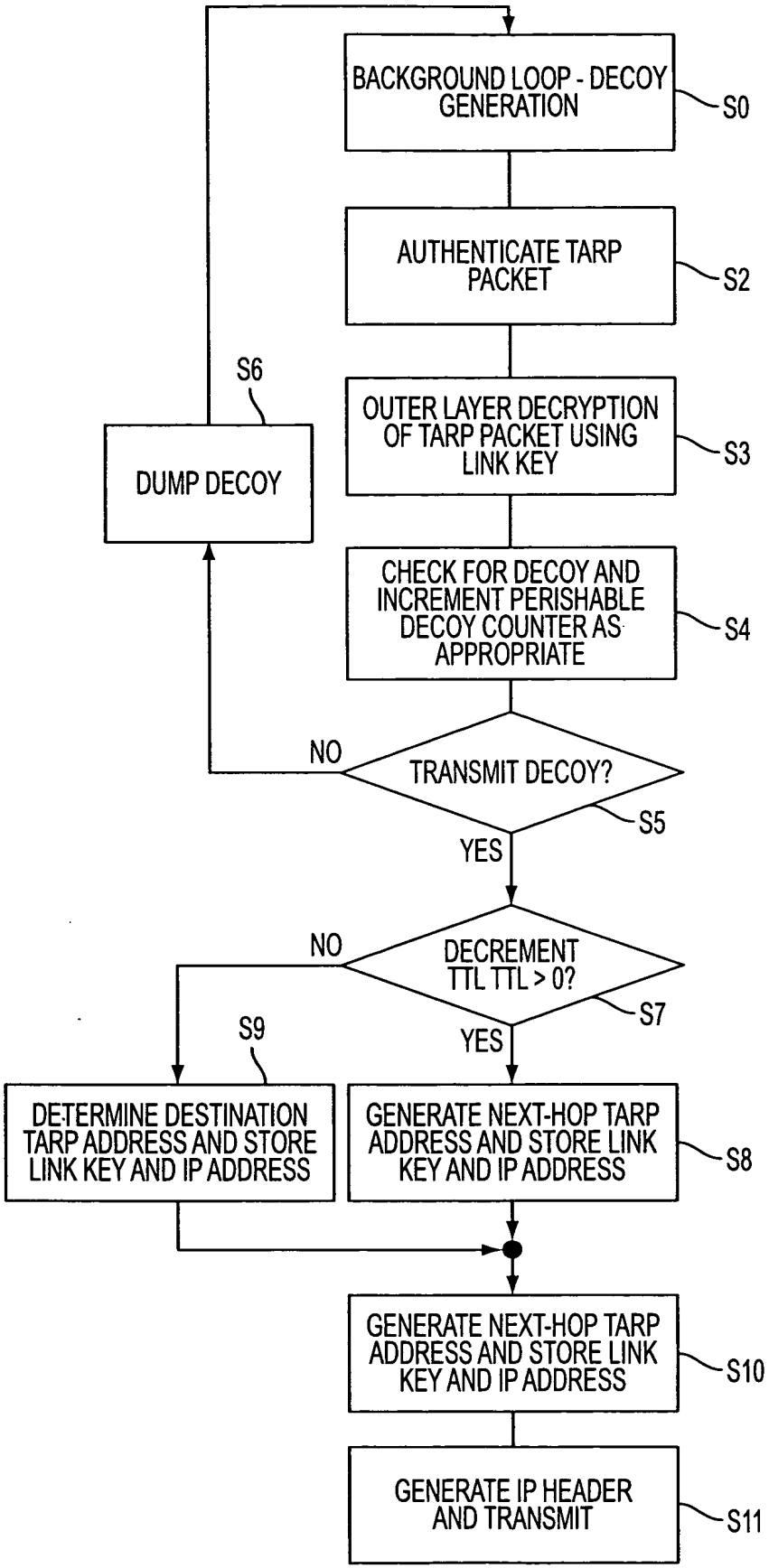

BACKGROUND LOOP - DECOY GENERATION
S0
AUTHENTICATE TARP PACKET
S2
OUTER LAYER DECRYPTION OF TARP PACKET USING LINK KEY
S3
CHECK FOR DECOY AND INCREMENT PERISHABLE DECOY COUNTER AS APPROPRIATE
S4
TRANSMIT DECOY?
S5
NO
YES
DUMP DECOY
S6
DECREMENT TTL TTL > 0?
S7
NO
YES
DETERMINE DESTINATION TARP ADDRESS AND STORE LINK KEY AND IP ADDRESS
S9
GENERATE NEXT-HOP TARP ADDRESS AND STORE LINK KEY AND IP ADDRESS
S8
GENERATE NEXT-HOP TARP ADDRESS AND STORE LINK KEY AND IP ADDRESS
S10
GENERATE IP HEADER AND TRANSMIT
S11

FIG. 5

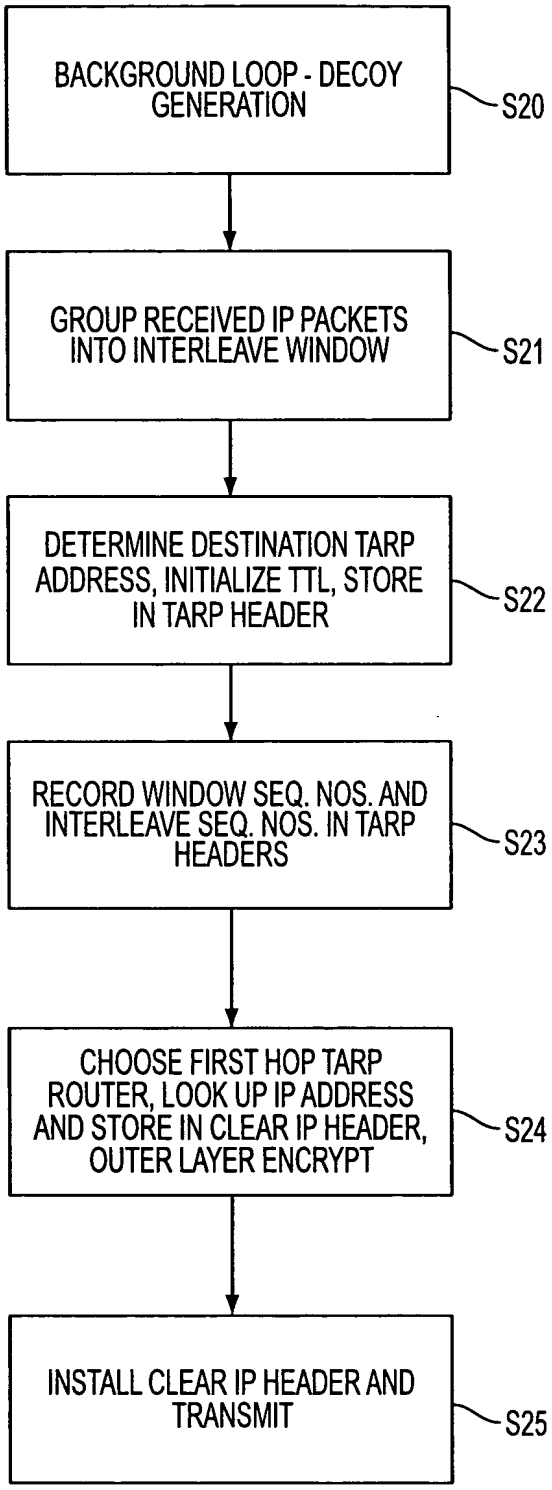

BACKGROUND LOOP - DECOY GENERATION
S20
GROUP RECEIVED IP PACKETS INTO INTERLEAVE WINDOW
S21
DETERMINE DESTINATION TARP ADDRESS, INITIALIZE TTL, STORE IN TARP HEADER
S22
RECORD WINDOW SEQ. NOS. AND INTERLEAVE SEQ. NOS. IN TARP HEADERS
S23
CHOOSE FIRST HOP TARP ROUTER, LOOK UP IP ADDRESS AND STORE IN CLEAR IP HEADER, OUTER LAYER ENCRYPT
S24
INSTALL CLEAR IP HEADER AND TRANSMIT
S25

FIG. 6

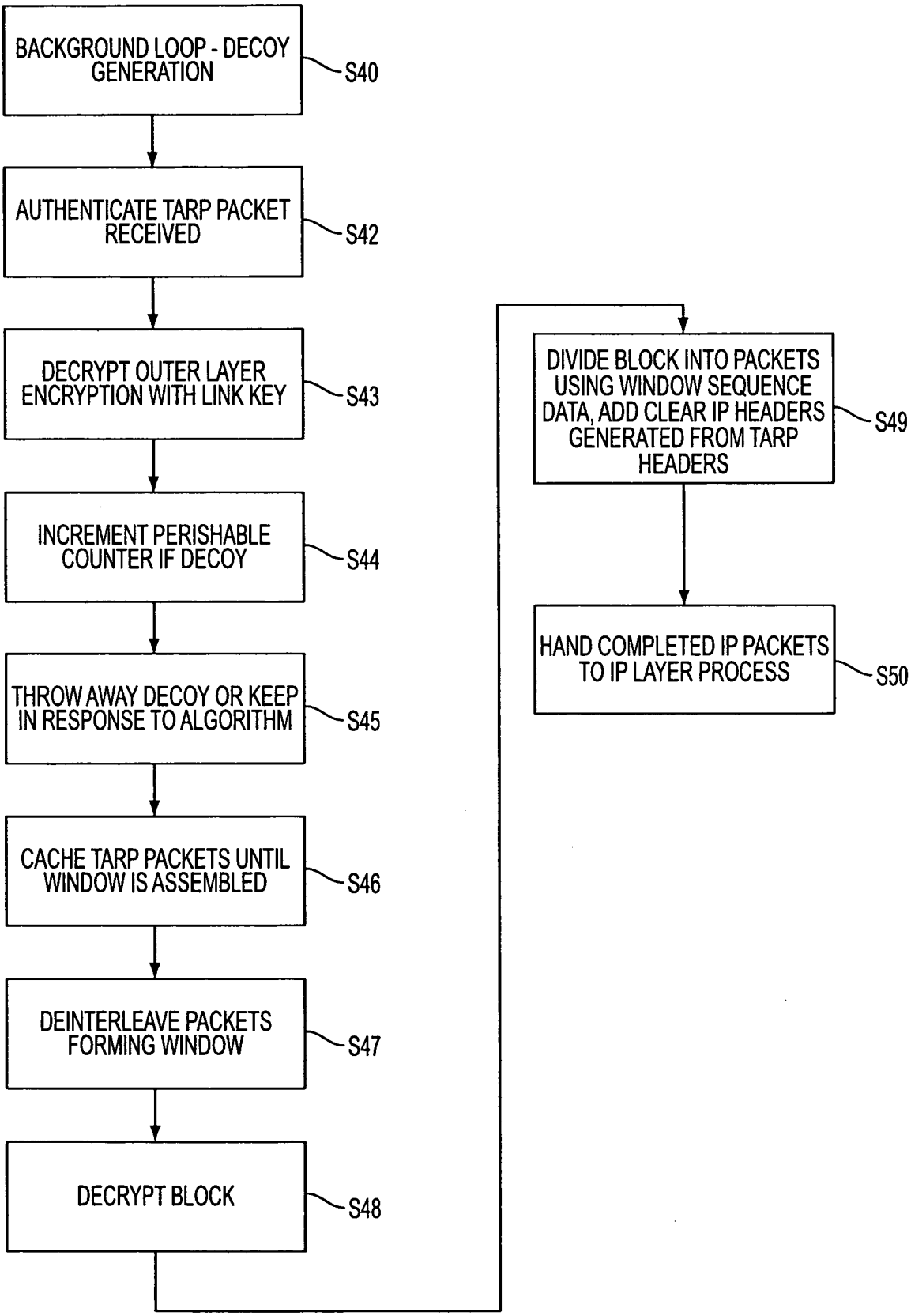

BACKGROUND LOOP - DECOY GENERATION
S40
AUTHENTICATE TARP PACKET RECEIVED
S42
DECRYPT OUTER LAYER ENCRYPTION WITH LINK KEY
S43
INCREMENT PERISHABLE COUNTER IF DECOY
S44
THROW AWAY DECOY OR KEEP IN RESPONSE TO ALGORITHM
S45
CACHE TARP PACKETS UNTIL WINDOW IS ASSEMBLED
S46
DEINTERLEAVE PACKETS FORMING WINDOW
S47
DECRYPT BLOCK
S48
DIVIDE BLOCK INTO PACKETS USING WINDOW SEQUENCE DATA, ADD CLEAR IP HEADERS GENERATED FROM TARP HEADERS
S49
HAND COMPLETED IP PACKETS TO IP LAYER PROCESS
S50

FIG. 7

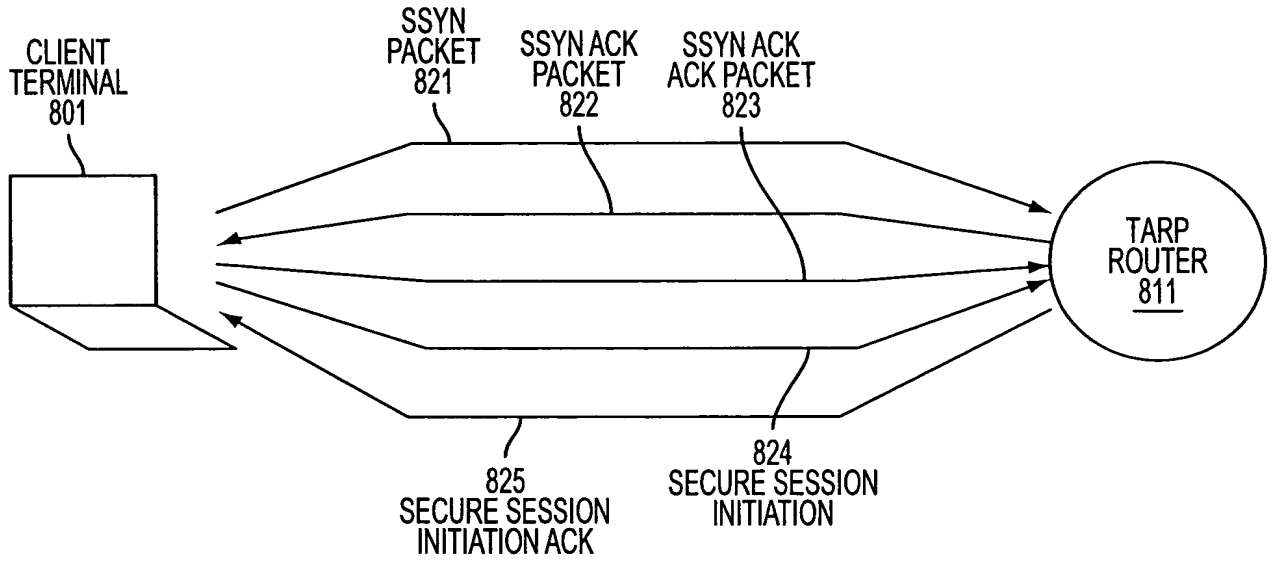

CLIENT TERMINAL 801
SSYN PACKET 821
SSYN ACK PACKET 822
SSYN ACK ACK PACKET 823
TARP ROUTER 811
825 SECURE SESSION INITIATION ACK
824 SECURE SESSION INITIATION

FIG. 8

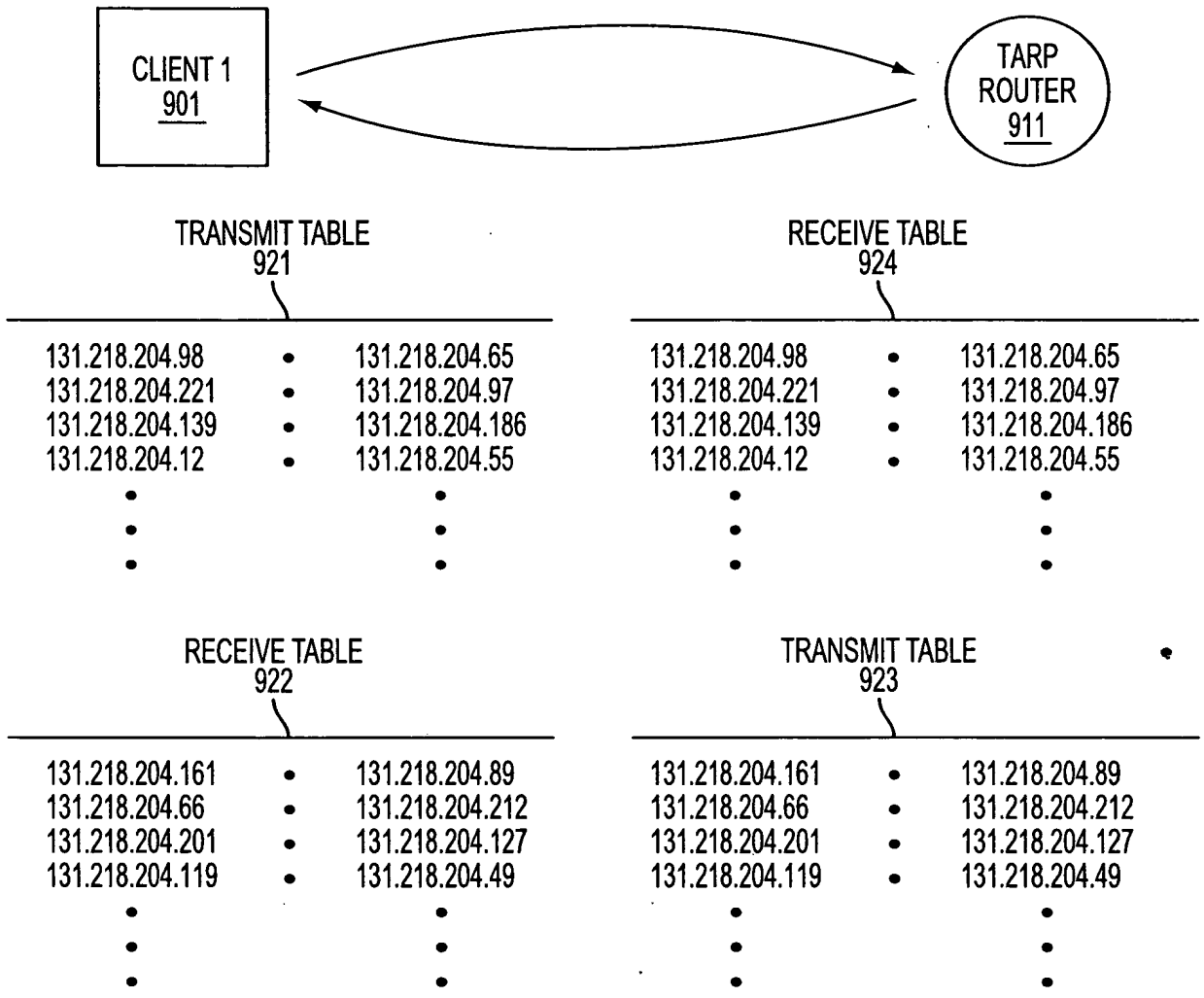

CLIENT 1
901
TARP
ROUTER
911
TRANSMIT TABLE
921
131.218.204.98 • 131.218.204.65
131.218.204.221 • 131.218.204.97
131.218.204.139 • 131.218.204.186
131.218.204.12 • 131.218.204.55
RECEIVE TABLE
924
131.218.204.98 • 131.218.204.65
131.218.204.221 • 131.218.204.97
131.218.204.139 • 131.218.204.186
131.218.204.12 • 131.218.204.55
RECEIVE TABLE
922
131.218.204.161 • 131.218.204.89
131.218.204.66 • 131.218.204.212
131.218.204.201 • 131.218.204.127
131.218.204.119 • 131.218.204.49
TRANSMIT TABLE
923
131.218.204.161 • 131.218.204.89
131.218.204.66 • 131.218.204.212
131.218.204.201 • 131.218.204.127
131.218.204.119 • 131.218.204.49

FIG. 9

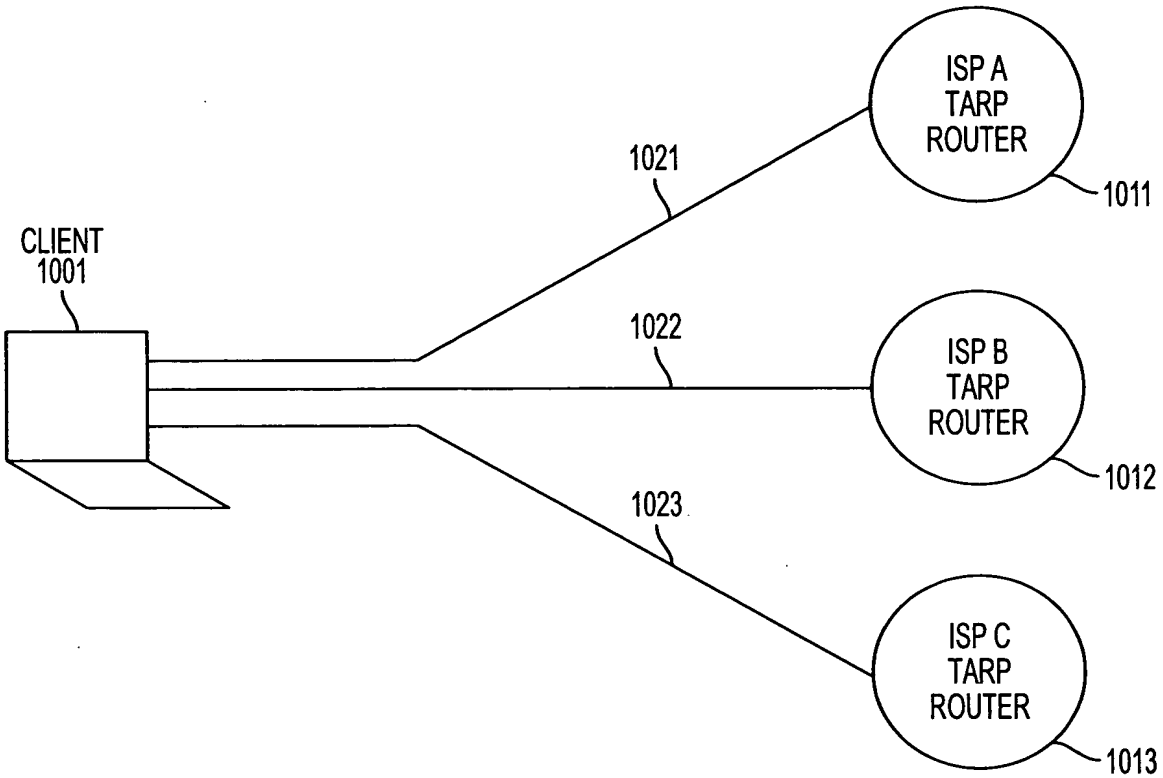

ISP A
TARP
ROUTER
1011
1021
CLIENT
1001
1022
ISP B
TARP
ROUTER
1012
1023
ISP C
TARP
ROUTER
1013

FIG. 10

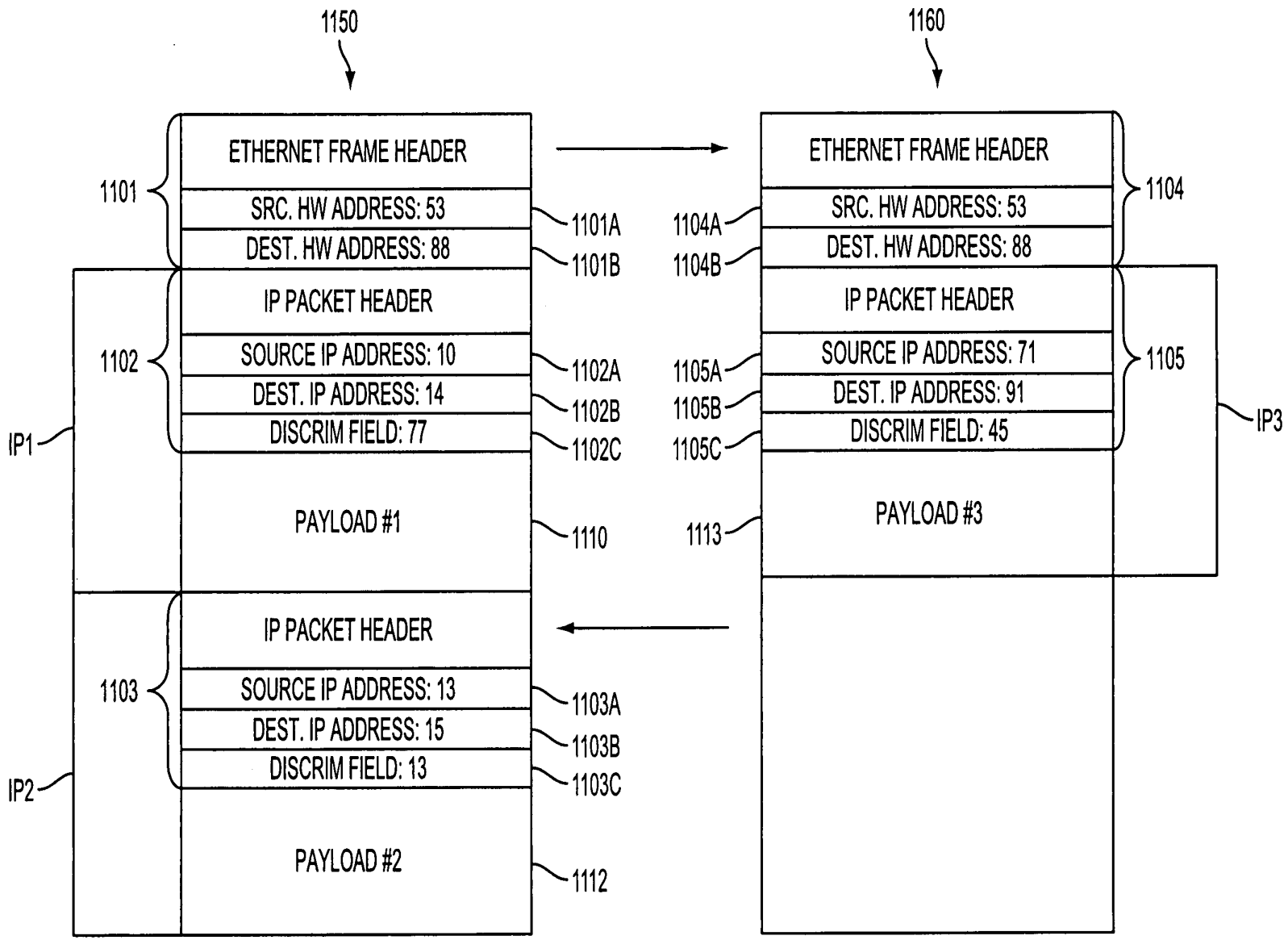


FIG. 11

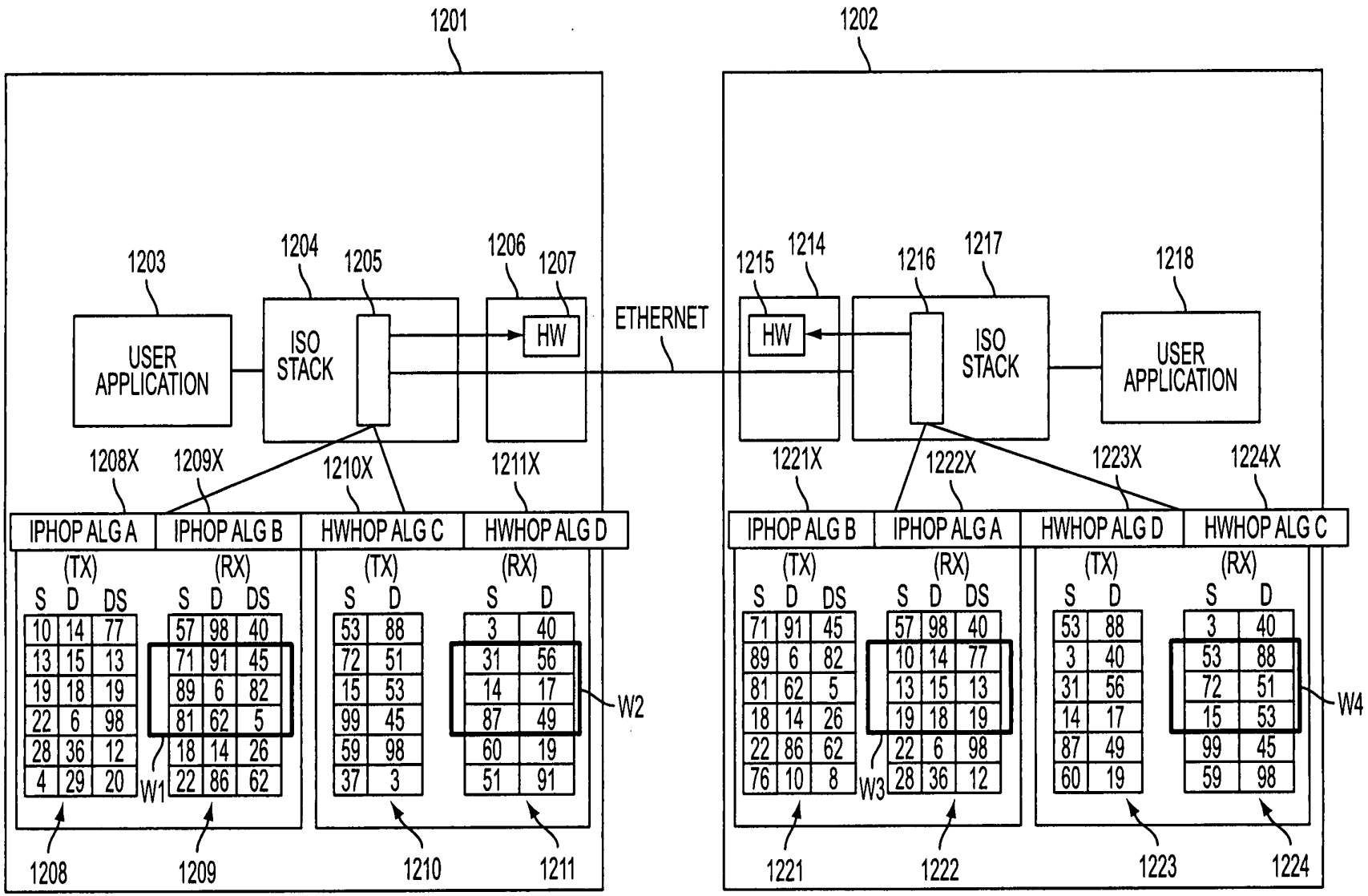


FIG. 12A

MODE OR EMBODIMENT	HARDWARE ADDRESSES	IP ADDRESSES	DISCRIMINATOR FIELD VALUES
1. PROMISCUOUS	SAME FOR ALL NODES OR COMPLETELY RANDOM	CAN BE VARIED IN SYNC	CAN BE VARIED IN SYNC
2. PROMISCUOUS PER VPN	FIXED FOR EACH VPN	CAN BE VARIED IN SYNC	CAN BE VARIED IN SYNC
3. HARDWARE HOPPING	CAN BE VARIED IN SYNC	CAN BE VARIED IN SYNC	CAN BE VARIED IN SYNC

FIG. 12B

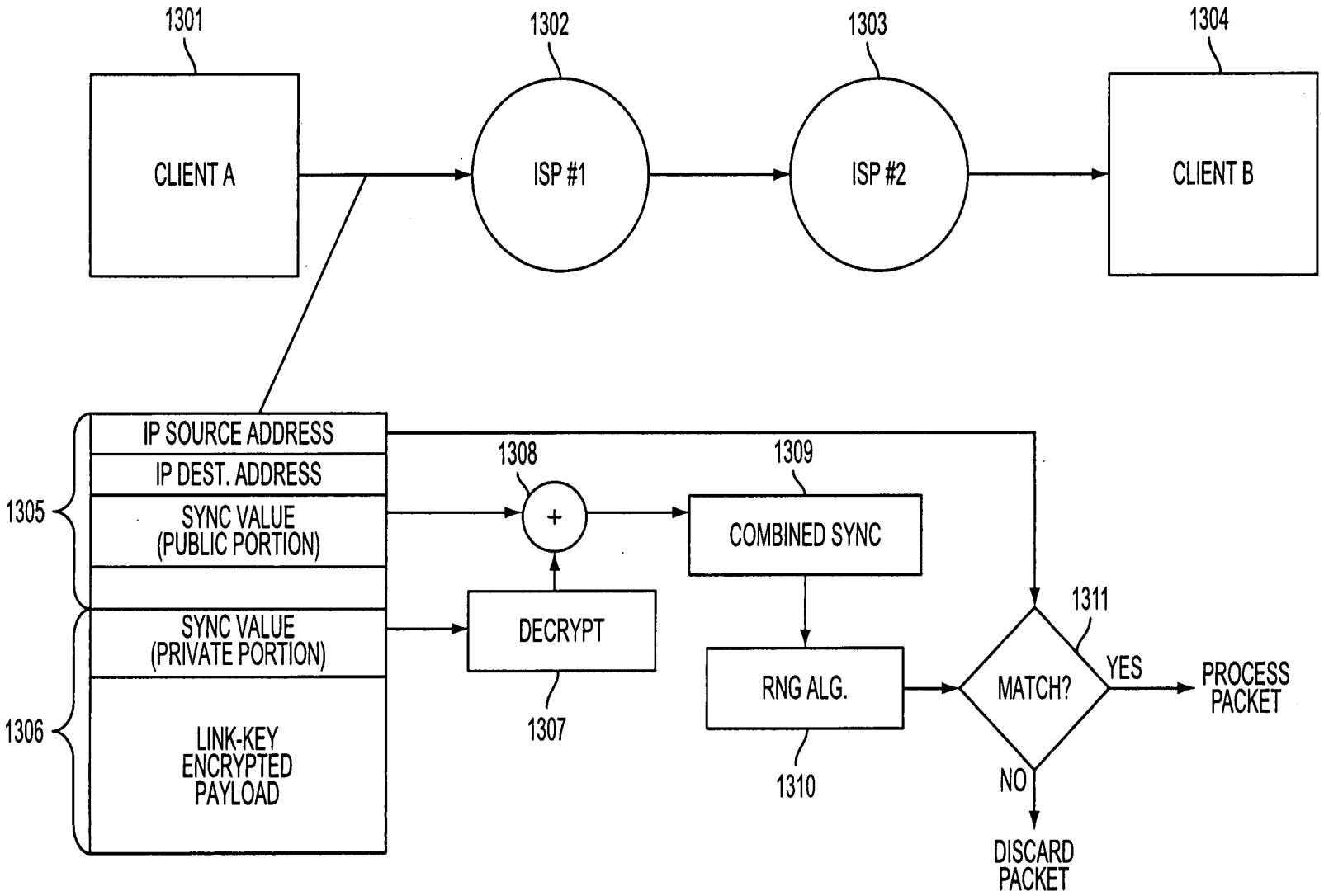


FIG. 13

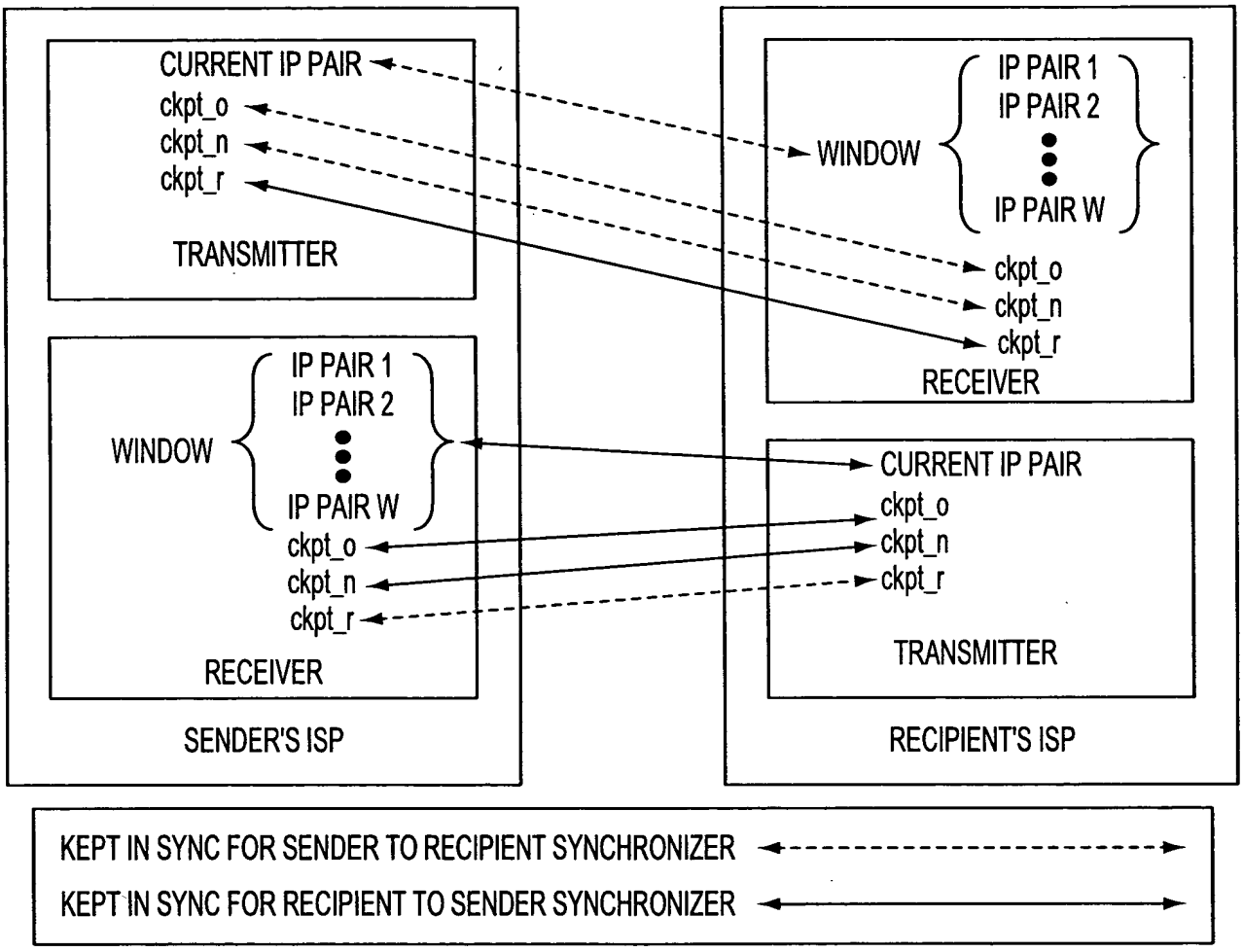

CURRENT IP PAIR
ckpt_o
ckpt_n
ckpt_r
TRANSMITTER
WINDOW
IP PAIR 1
IP PAIR 2
IP PAIR W
ckpt_o
ckpt_n
ckpt_r
RECEIVER
SENDER'S ISP
WINDOW
IP PAIR 1
IP PAIR 2
IP PAIR W
ckpt_o
ckpt_n
ckpt_r
RECEIVER
CURRENT IP PAIR
ckpt_o
ckpt_n
ckpt_r
TRANSMITTER
RECIPIENT'S ISP
KEPT IN SYNC FOR SENDER TO RECIPIENT SYNCHRONIZER
KEPT IN SYNC FOR RECIPIENT TO SENDER SYNCHRONIZER

FIG. 14

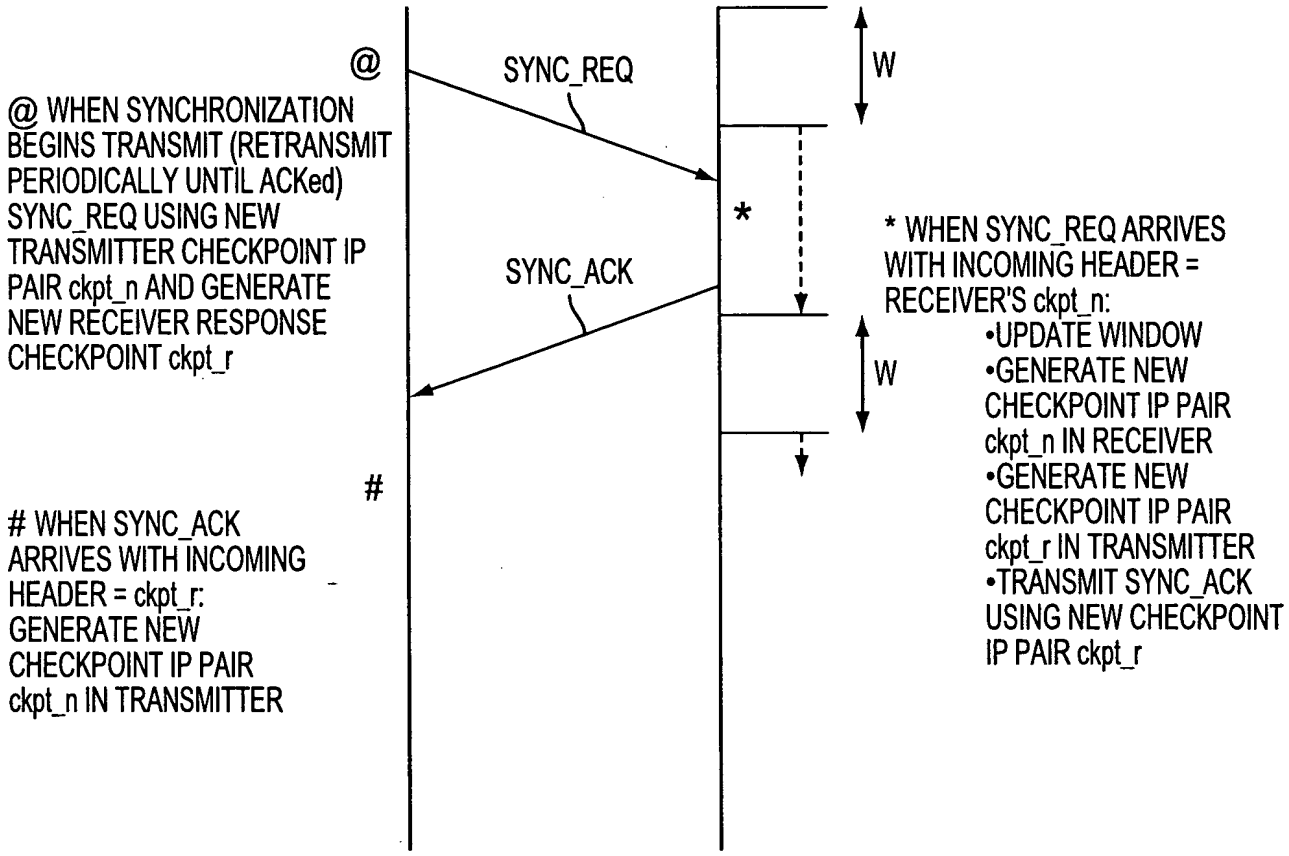

@
SYNC_REQ
W
@ WHEN SYNCHRONIZATION BEGINS TRANSMIT (RETRANSMIT PERIODICALLY UNTIL ACKed) SYNC_REQ USING NEW TRANSMITTER CHECKPOINT IP PAIR ckpt_n AND GENERATE NEW RECEIVER RESPONSE CHECKPOINT ckpt_r
*
* WHEN SYNC_REQ ARRIVES WITH INCOMING HEADER = RECEIVER'S ckpt_n:
•UPDATE WINDOW
•GENERATE NEW CHECKPOINT IP PAIR ckpt_n IN RECEIVER
•GENERATE NEW CHECKPOINT IP PAIR ckpt_r IN TRANSMITTER
•TRANSMIT SYNC_ACK USING NEW CHECKPOINT IP PAIR ckpt_r
SYNC_ACK
W
#
WHEN SYNC_ACK ARRIVES WITH INCOMING HEADER = ckpt_r: GENERATE NEW CHECKPOINT IP PAIR ckpt_n IN TRANSMITTER

FIG. 15

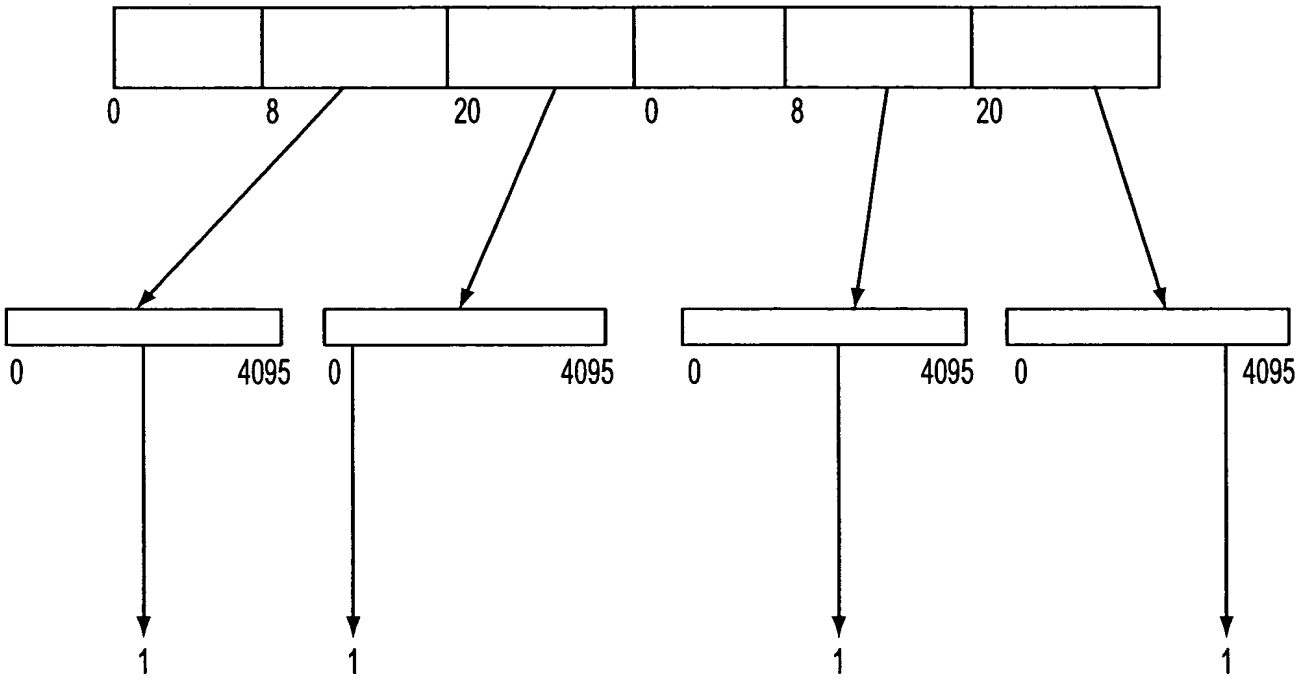


FIG. 16

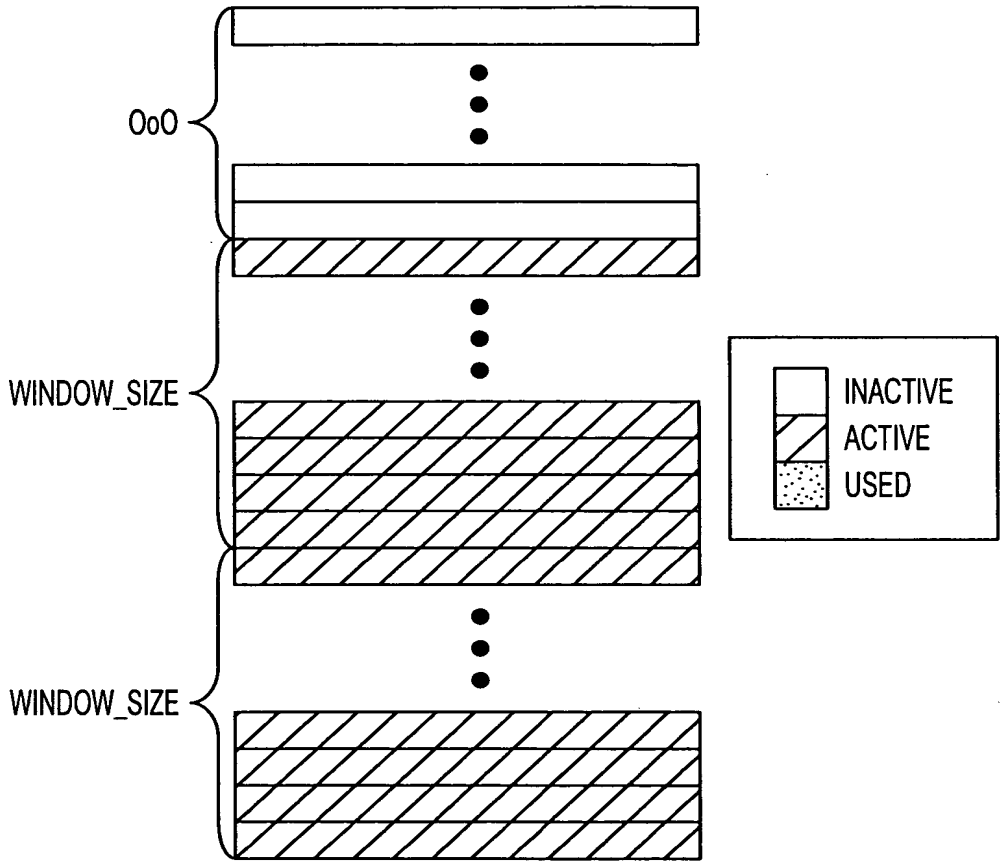


FIG. 17

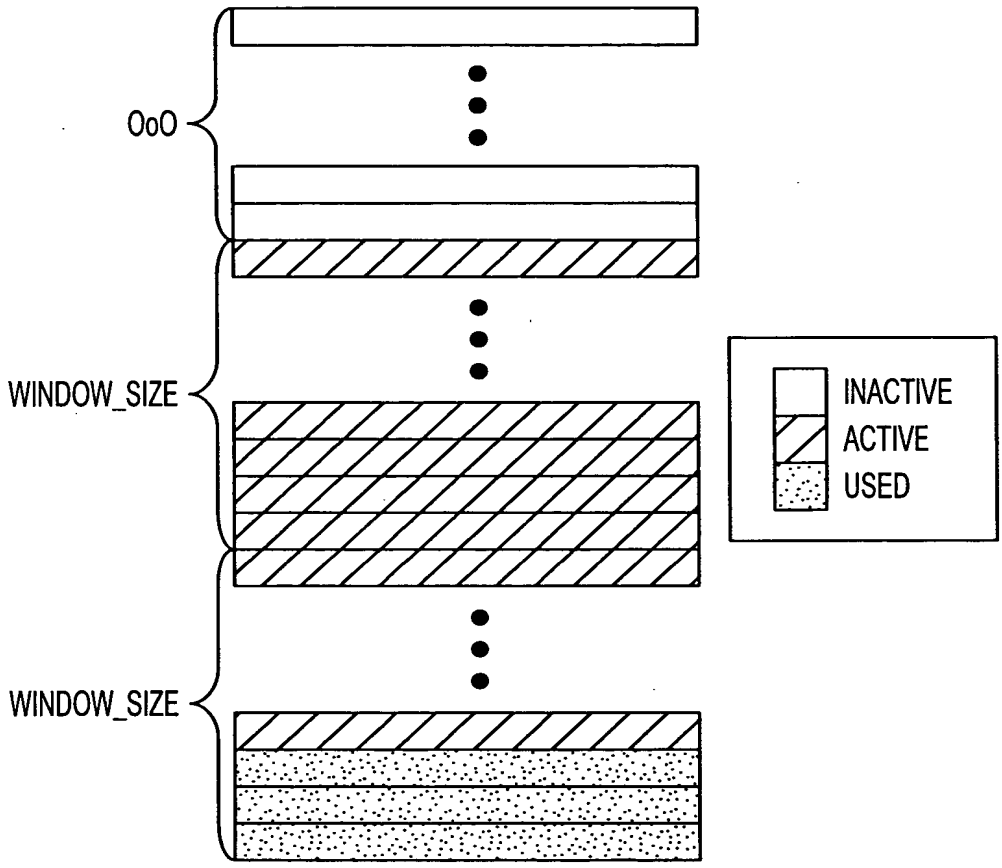


FIG. 18

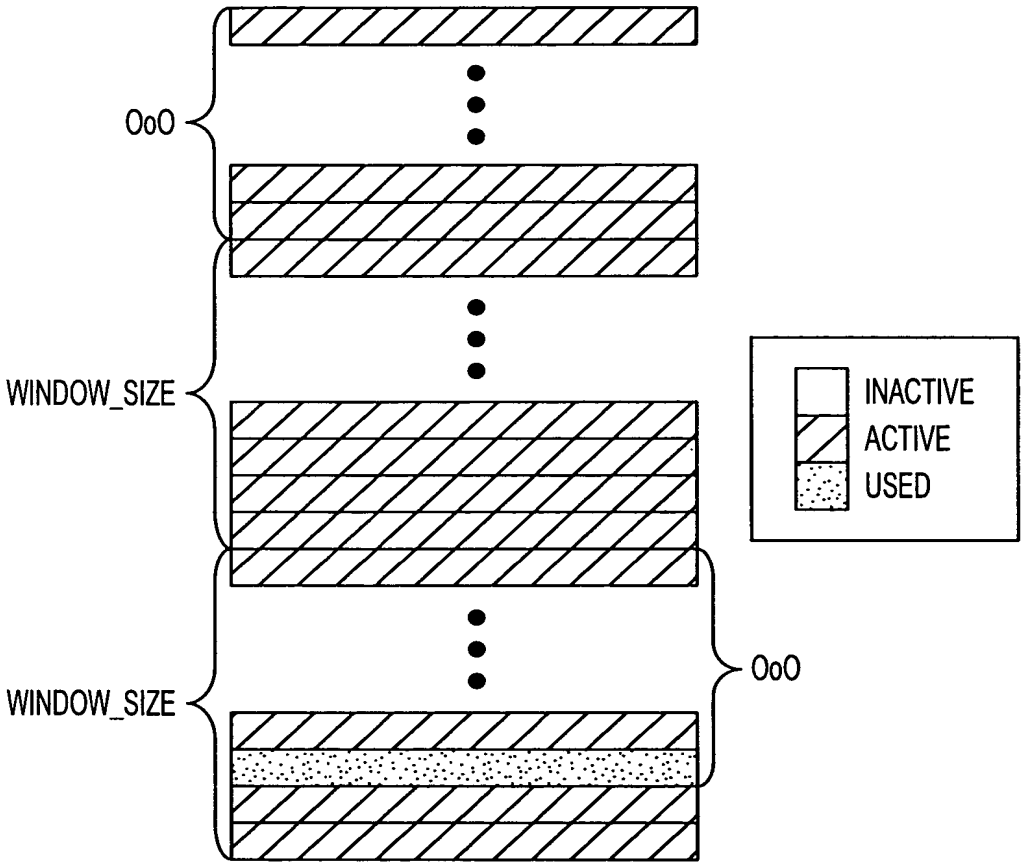


FIG. 19

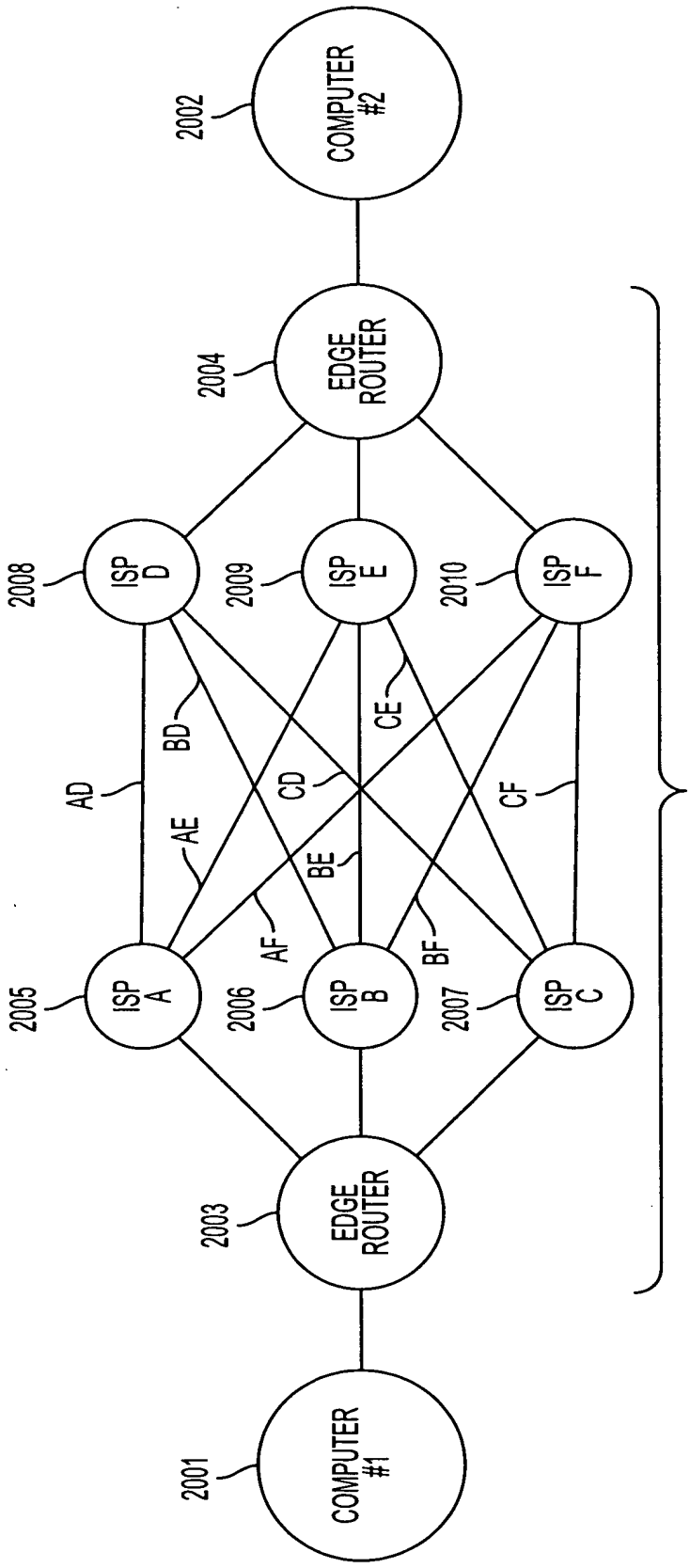

COMPUTER #1
2001
EDGE ROUTER
2003
ISP A
2005
ISP B
2006
ISP C
2007
ISP D
2008
ISP E
2009
ISP F
2010
EDGE ROUTER
2004
COMPUTER #2
2002
AD
AE
AF
BD
BE
BF
CD
CE
CF
2011

FIG. 20

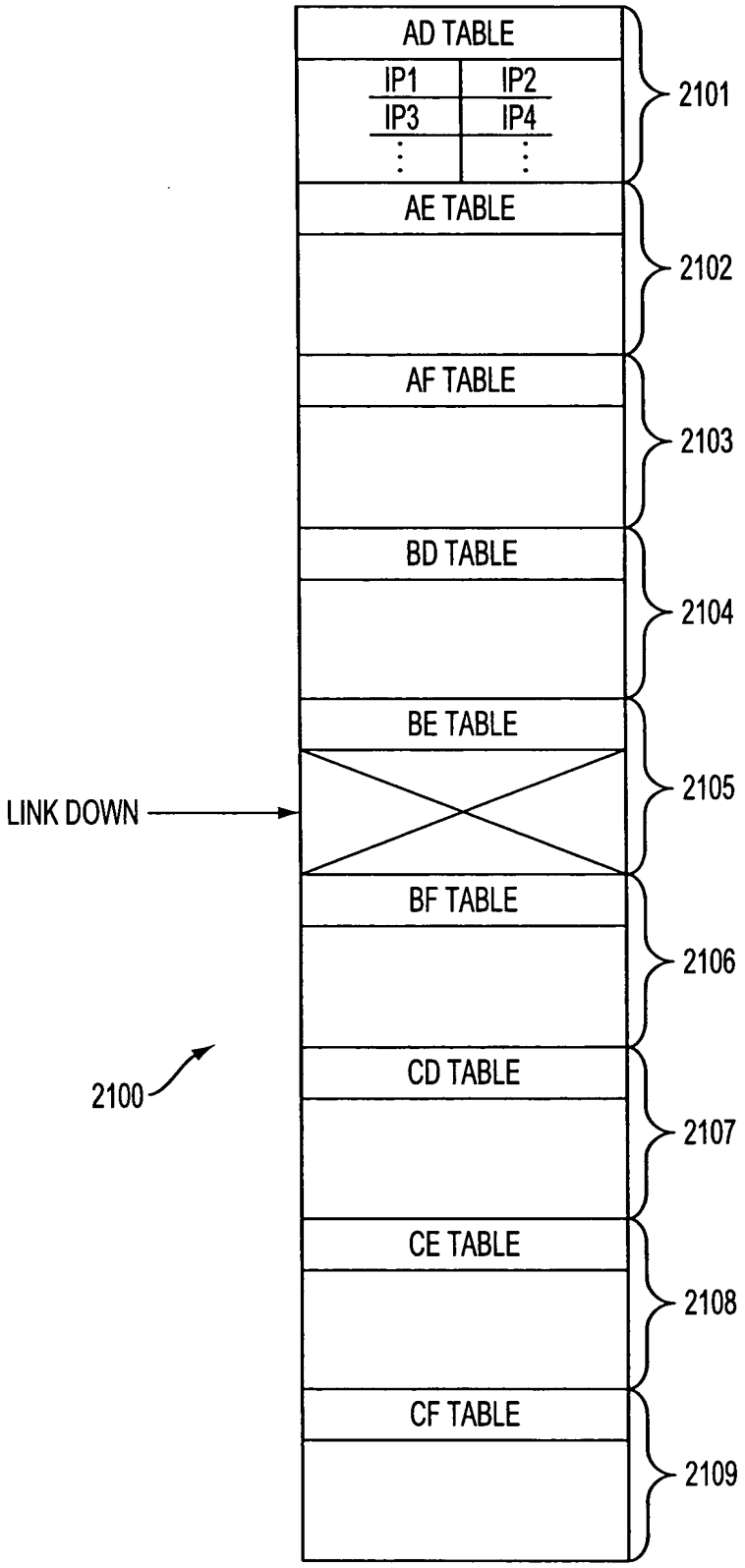

AD TABLE
IP1
IP2
IP3
IP4
2101
AE TABLE
2102
AF TABLE
2103
BD TABLE
2104
BE TABLE
2105
LINK DOWN
BF TABLE
2106
2100
CD TABLE
2107
CE TABLE
2108
CF TABLE
2109

FIG. 21

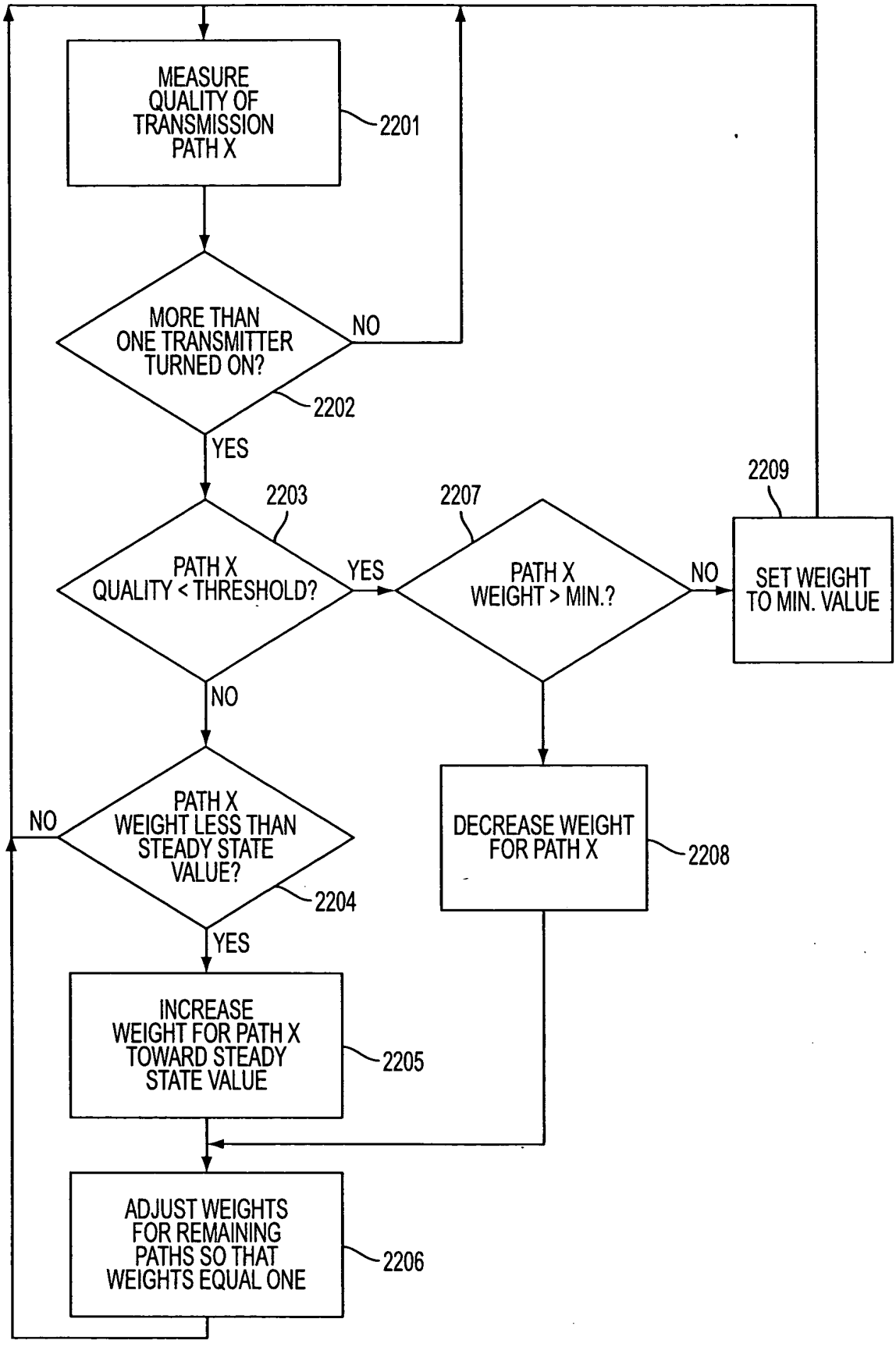

MEASURE QUALITY OF TRANSMISSION PATH X
2201
MORE THAN ONE TRANSMITTER TURNED ON?
NO
2202
YES
2203
PATH X QUALITY < THRESHOLD?
YES
2207
PATH X WEIGHT > MIN.?
NO
2209
SET WEIGHT TO MIN. VALUE
NO
PATH X WEIGHT LESS THAN STEADY STATE VALUE?
NO
2204
YES
DECREASE WEIGHT FOR PATH X
2208
INCREASE WEIGHT FOR PATH X TOWARD STEADY STATE VALUE
2205
ADJUST WEIGHTS FOR REMAINING PATHS SO THAT WEIGHTS EQUAL ONE
2206

FIG. 22A

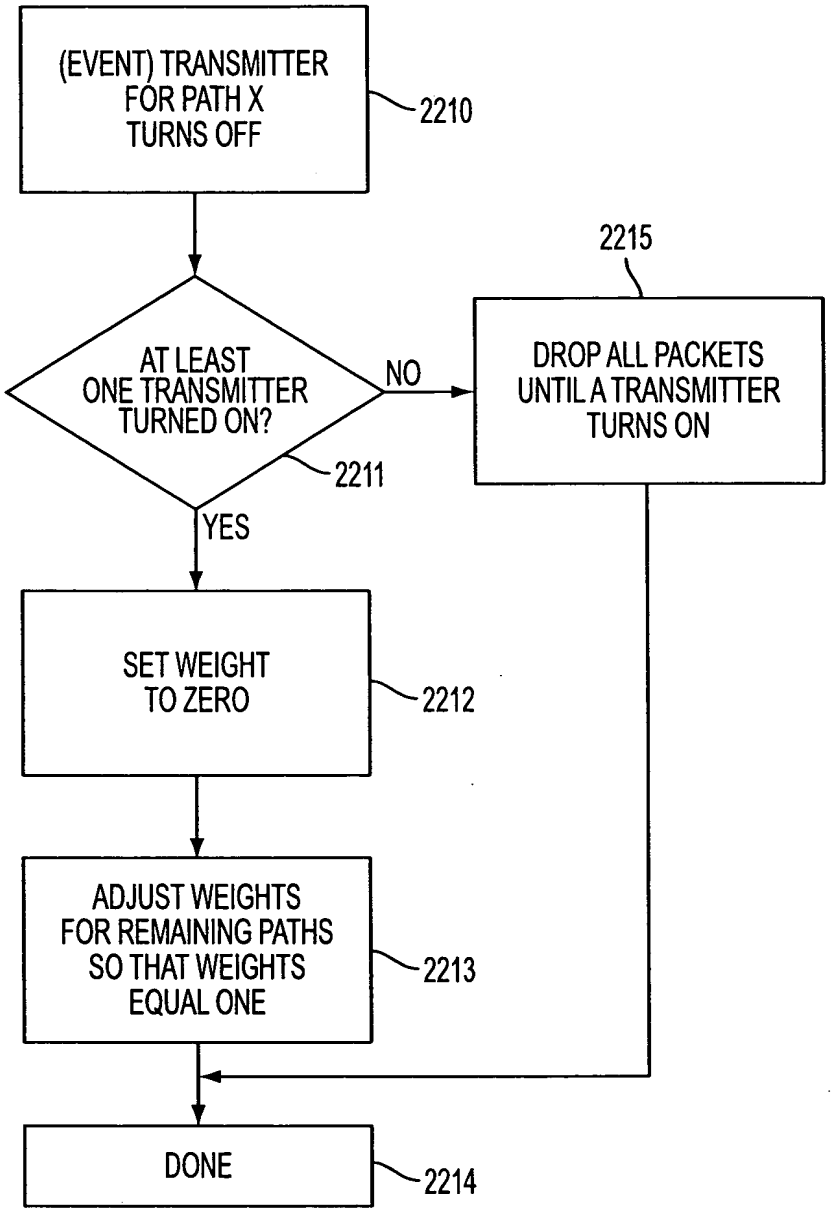

(EVENT) TRANSMITTER FOR PATH X TURNS OFF
2210
AT LEAST ONE TRANSMITTER TURNED ON?
NO
2215
DROP ALL PACKETS UNTIL A TRANSMITTER TURNS ON
2211
YES
SET WEIGHT TO ZERO
2212
ADJUST WEIGHTS FOR REMAINING PATHS SO THAT WEIGHTS EQUAL ONE
2213
DONE
2214

FIG. 22B

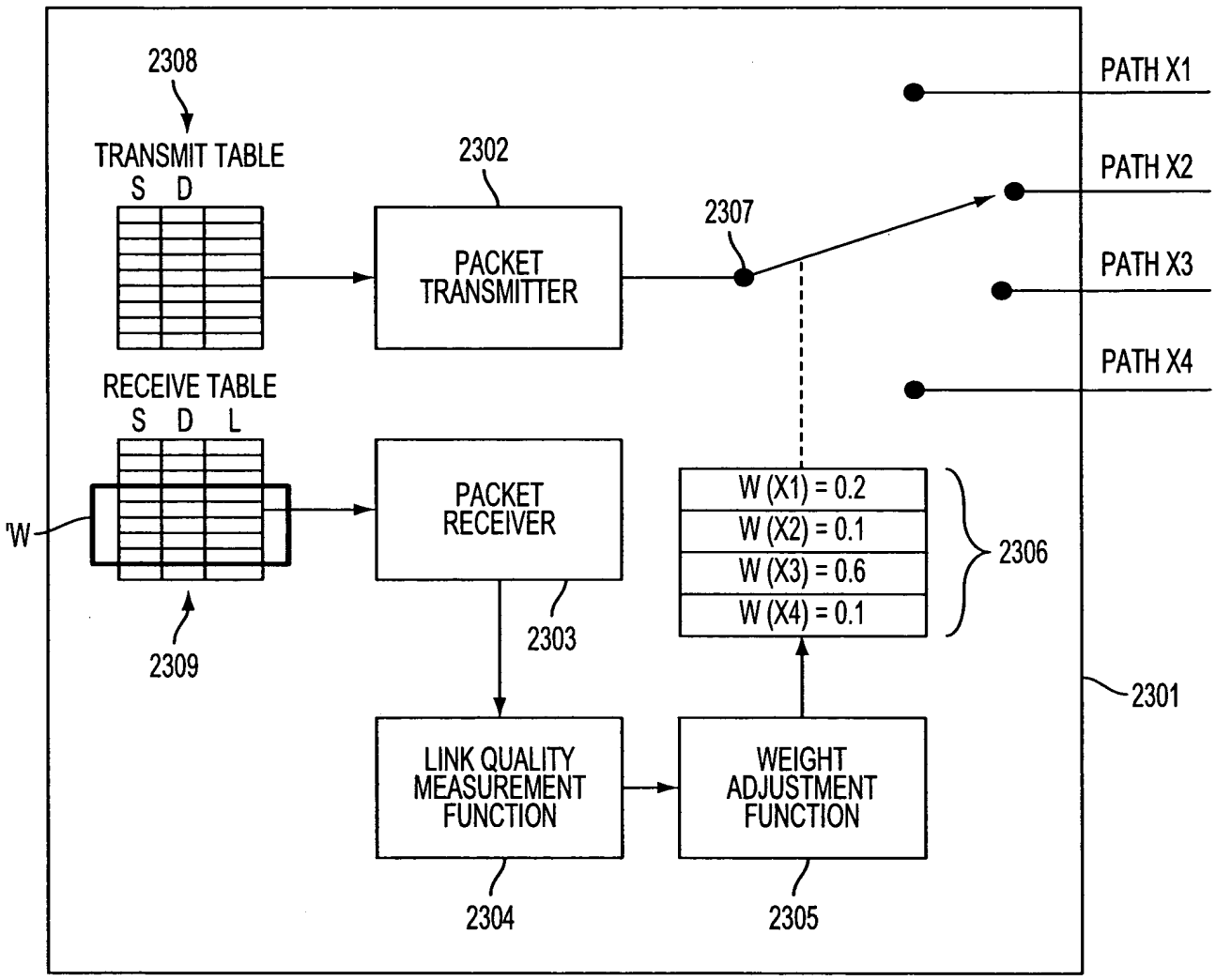

2308
TRANSMIT TABLE
S D
2302
PACKET TRANSMITTER
2307
PATH X1
PATH X2
PATH X3
PATH X4
RECEIVE TABLE
S D L
'W
PACKET RECEIVER
W (X1) = 0.2
W (X2) = 0.1
W (X3) = 0.6
W (X4) = 0.1
2306
2303
2309
2301
LINK QUALITY MEASUREMENT FUNCTION
WEIGHT ADJUSTMENT FUNCTION
2304
2305

FIG. 23

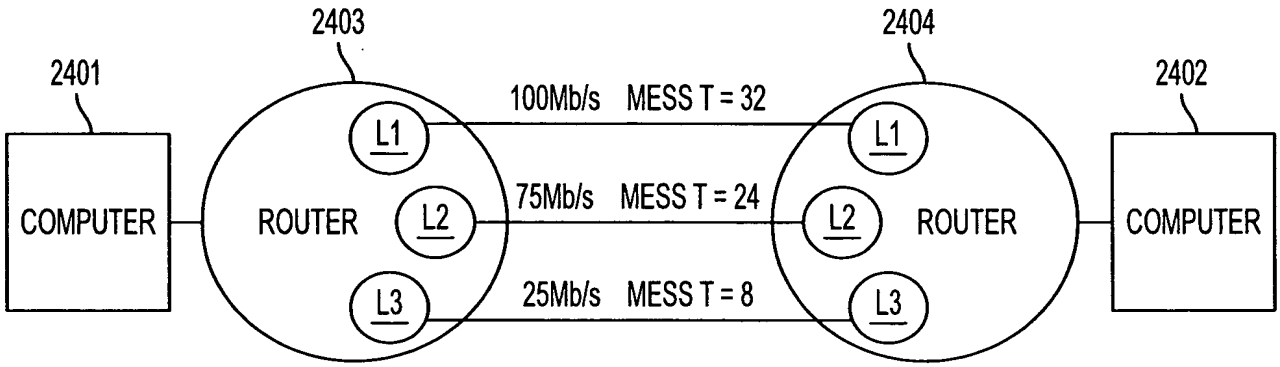

2403
2404
2401
2402
100Mb/s MESS T = 32
L1
L1
COMPUTER
ROUTER
L2
75Mb/s MESS T = 24
L2
ROUTER
COMPUTER
L3
25Mb/s MESS T = 8
L3

FIG. 24

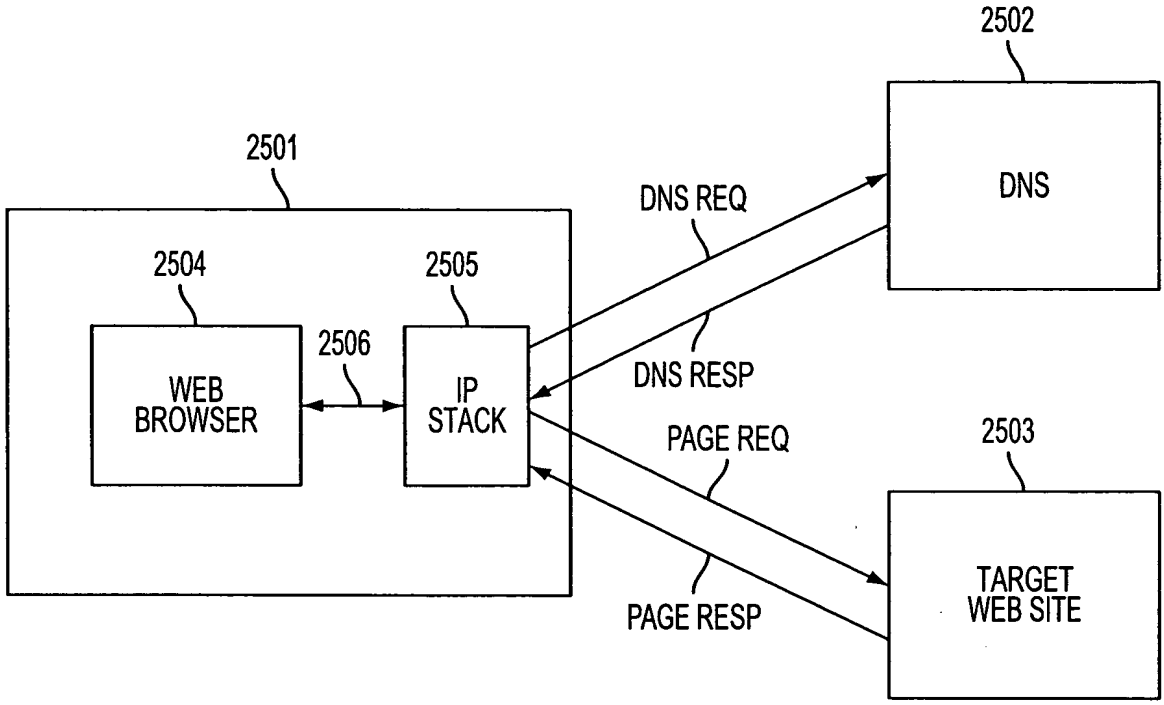


FIG. 25
(PRIOR ART)

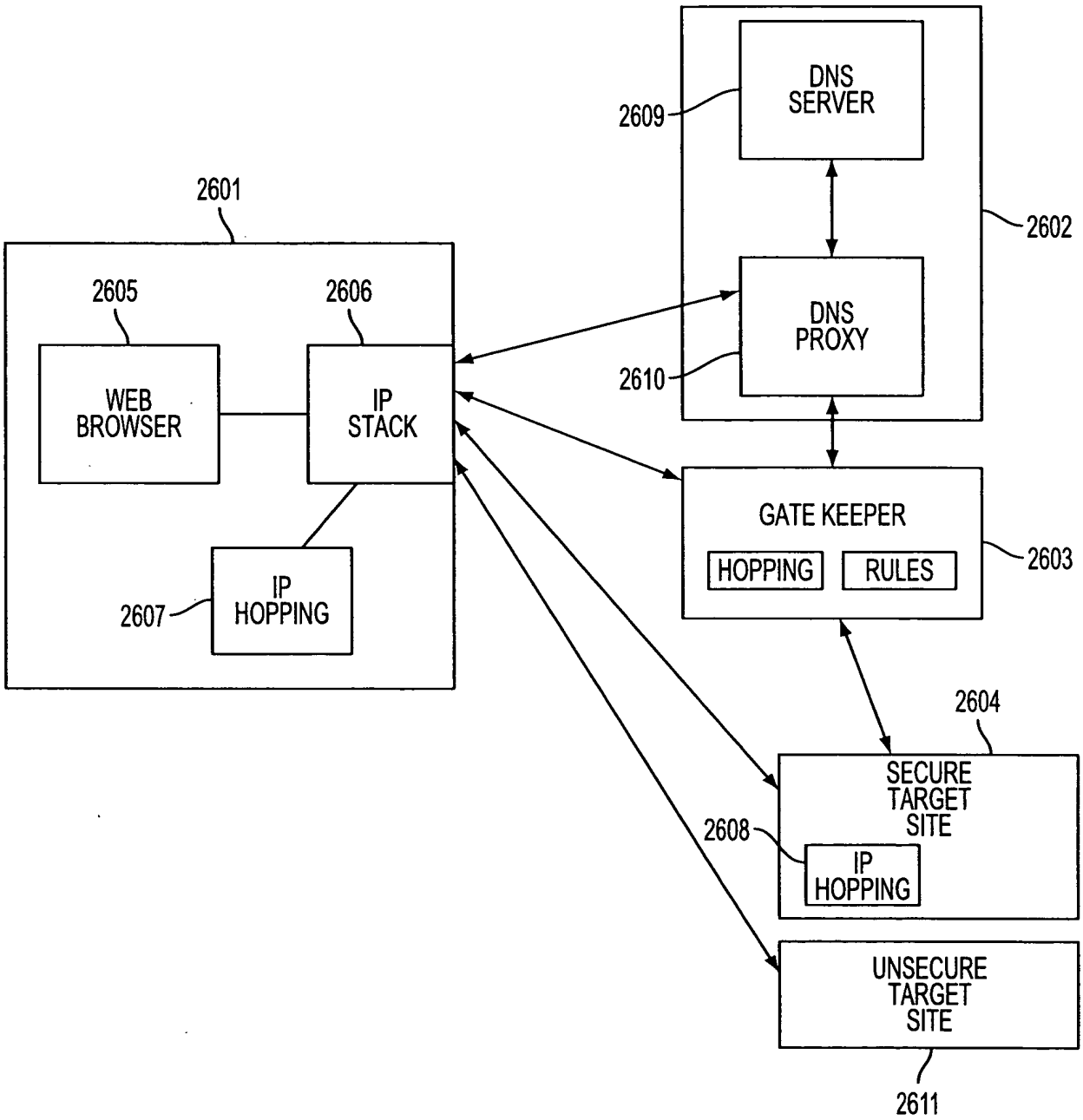

DNS SERVER
2609
2601
2602
2605
2606
DNS PROXY
WEB BROWSER
IP STACK
2610
GATE KEEPER
HOPPING
RULES
2603
IP HOPPING
2607
2604
SECURE TARGET SITE
2608
IP HOPPING
UNSECURE TARGET SITE
2611

FIG. 26

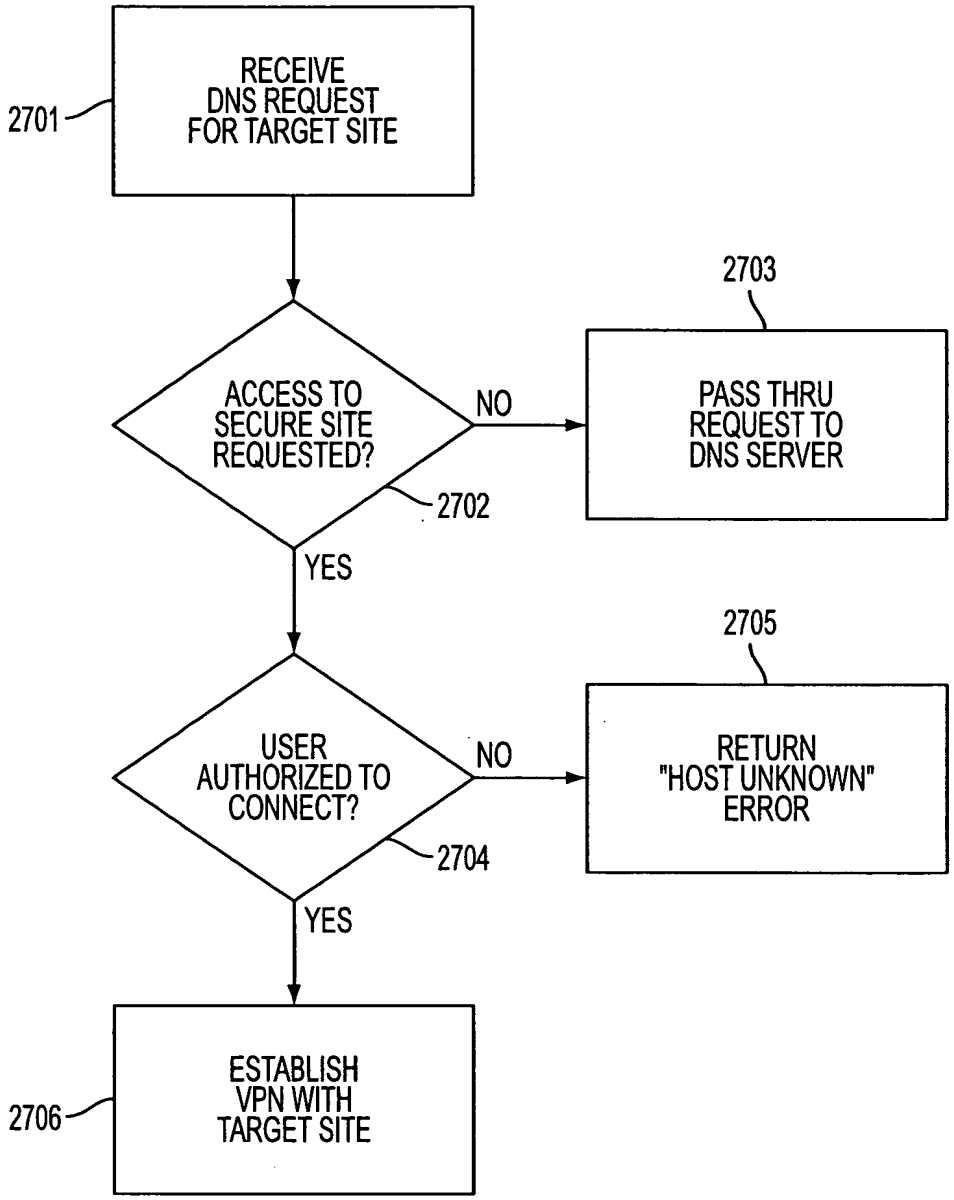

RECEIVE DNS REQUEST FOR TARGET SITE
2701
ACCESS TO SECURE SITE REQUESTED?
2702
NO
2703
PASS THRU REQUEST TO DNS SERVER
YES
USER AUTHORIZED TO CONNECT?
2704
NO
2705
RETURN "HOST UNKNOWN" ERROR
YES
ESTABLISH VPN WITH TARGET SITE
2706

FIG. 27

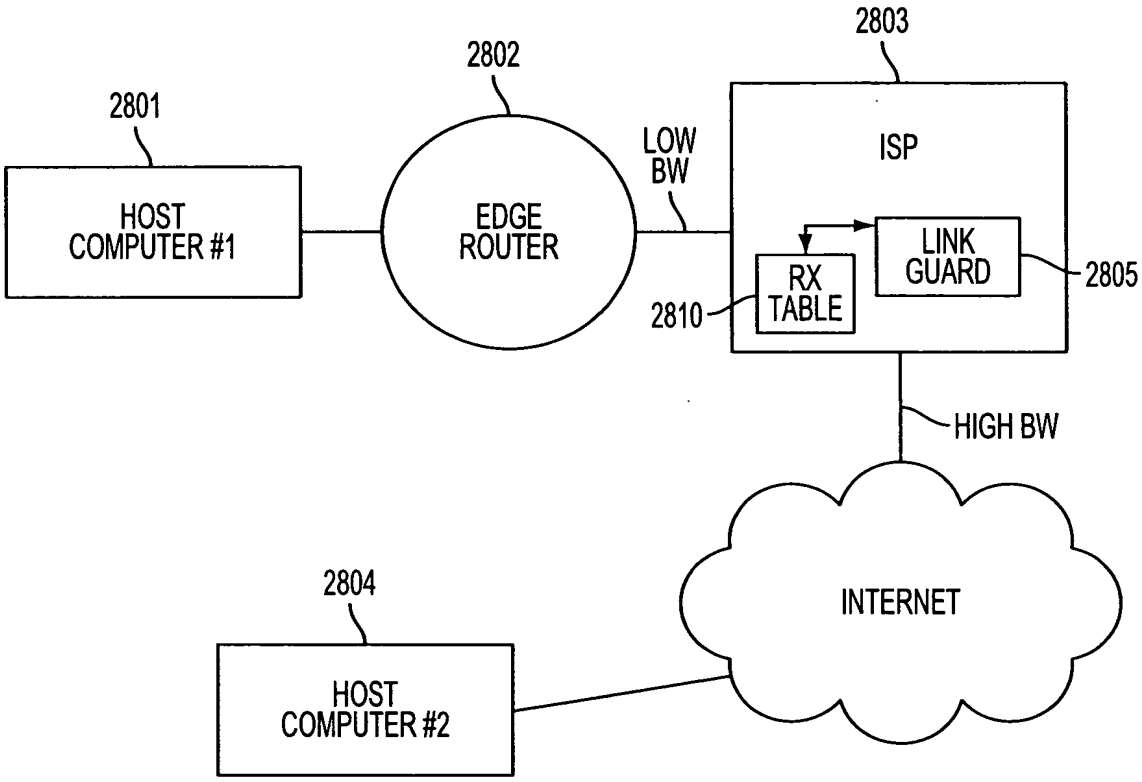

2803
2802
2801
ISP
LOW
BW
HOST
COMPUTER #1
EDGE
ROUTER
LINK
GUARD
2805
RX
TABLE
2810
HIGH BW
INTERNET
2804
HOST
COMPUTER #2

FIG. 28

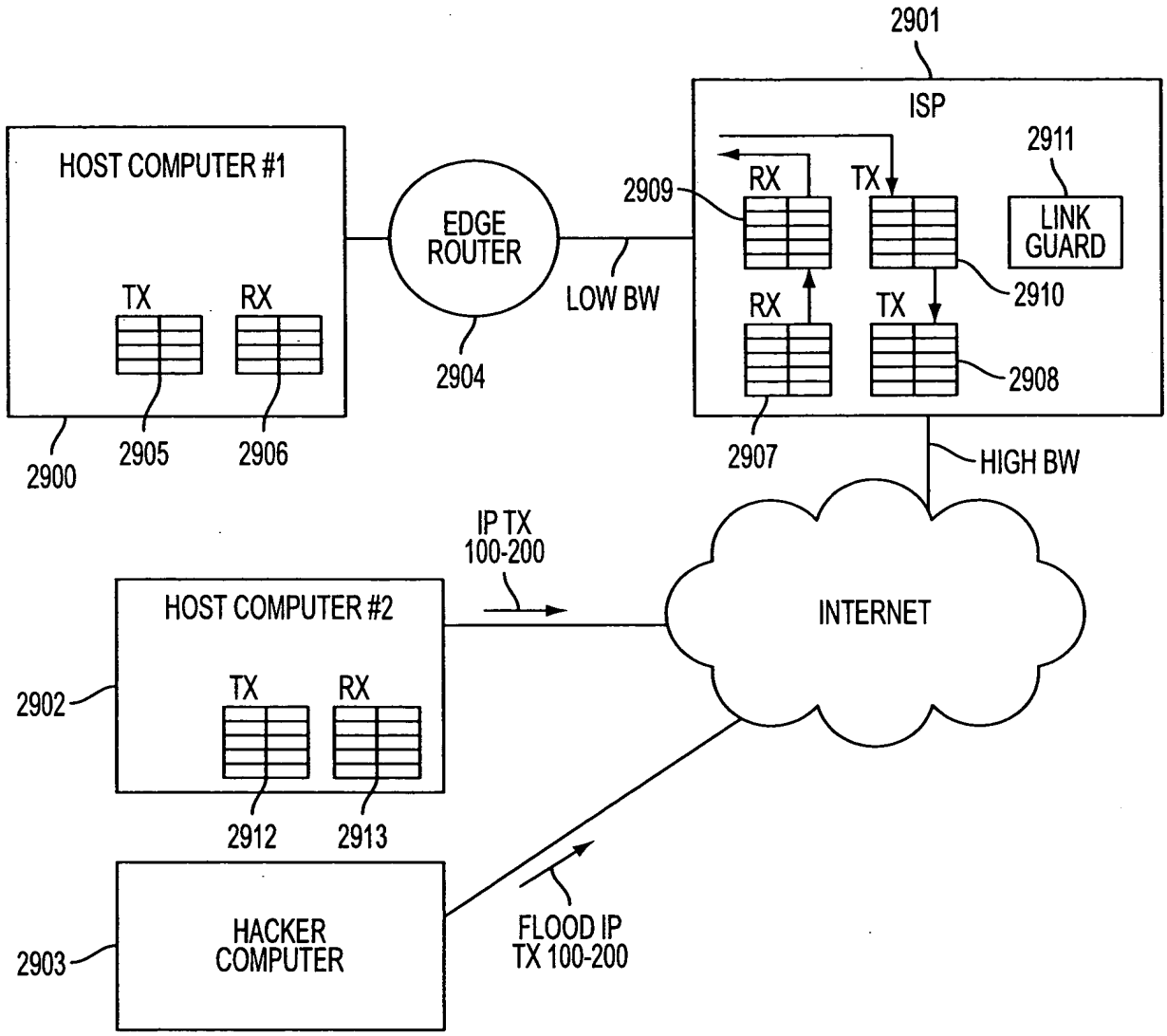

2901
ISP
HOST COMPUTER #1
EDGE ROUTER
2909
RX
TX
2911
LINK GUARD
2910
LOW BW
TX
RX
2904
RX
TX
2908
2900
2905
2906
2907
HIGH BW
IP TX 100-200
INTERNET
HOST COMPUTER #2
2902
TX
RX
2912
2913
HACKER COMPUTER
2903
FLOOD IP TX 100-200

FIG. 29

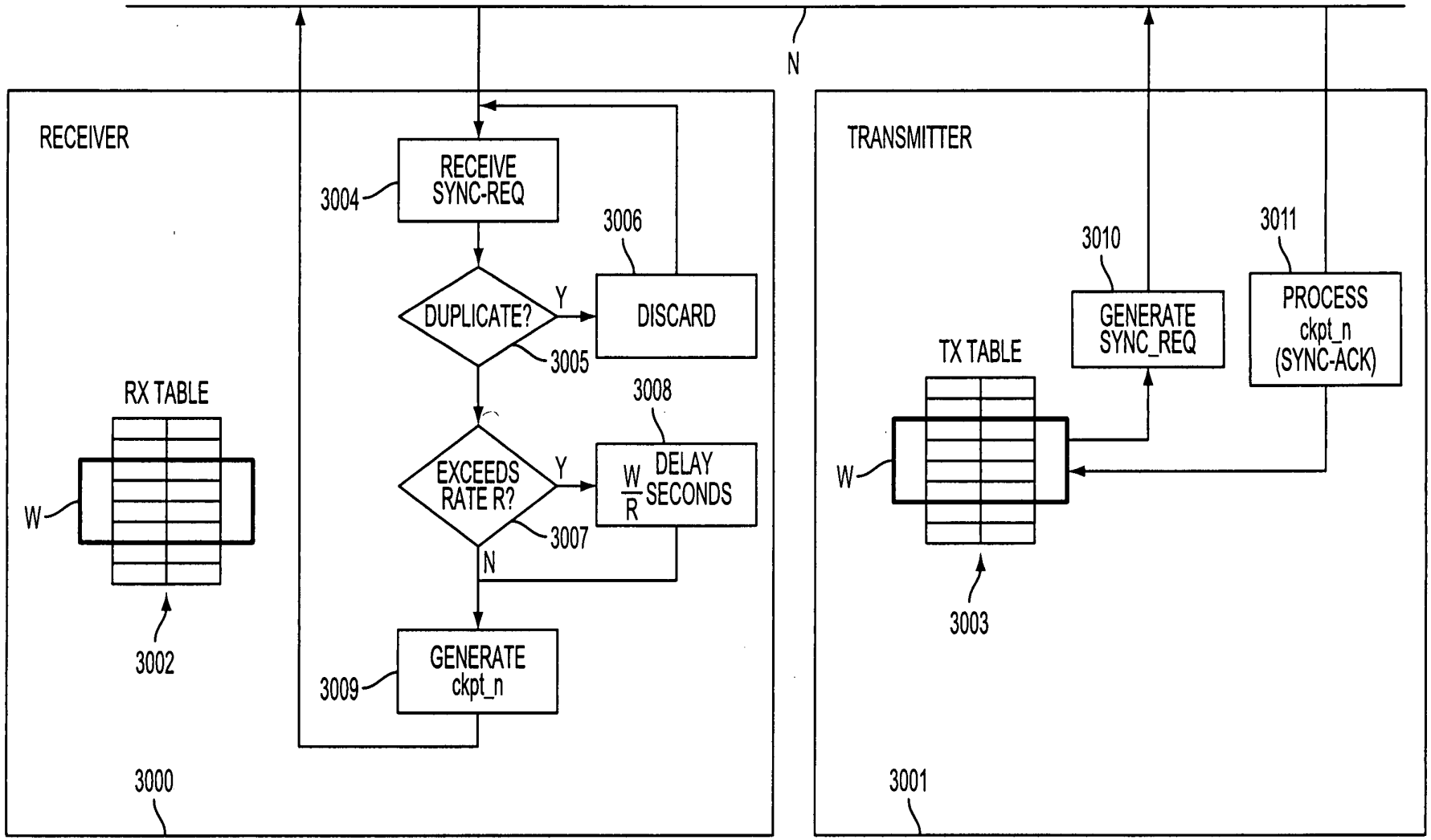


FIG. 30

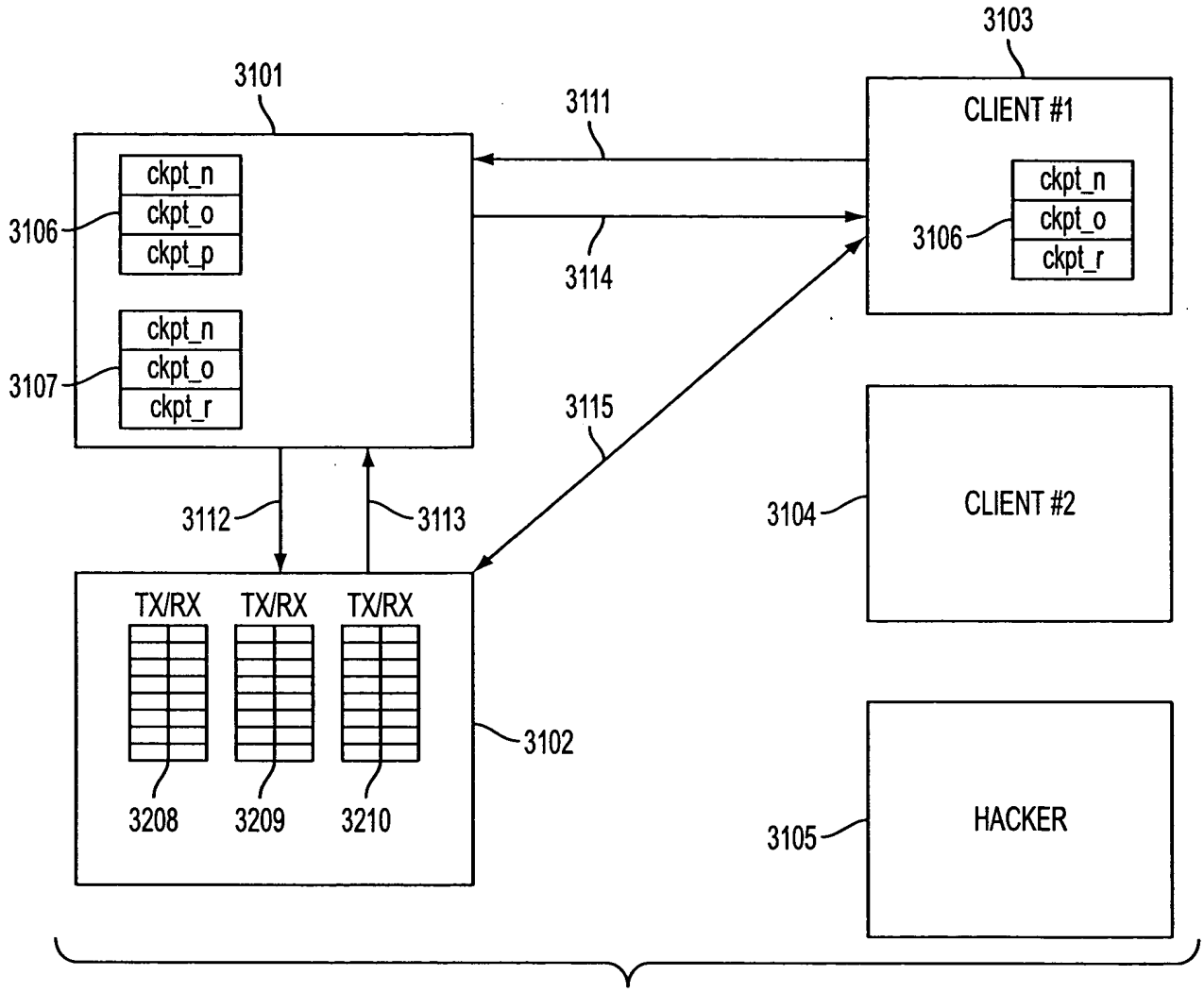

3103
3101
3111
CLIENT #1
ckpt_n
ckpt_o
ckpt_p
3106
ckpt_n
ckpt_o
ckpt_r
3106
3114
ckpt_n
ckpt_o
ckpt_r
3107
3115
3112
3113
3104
CLIENT #2
TX/RX
TX/RX
TX/RX
3102
3208
3209
3210
3105
HACKER

FIG. 31

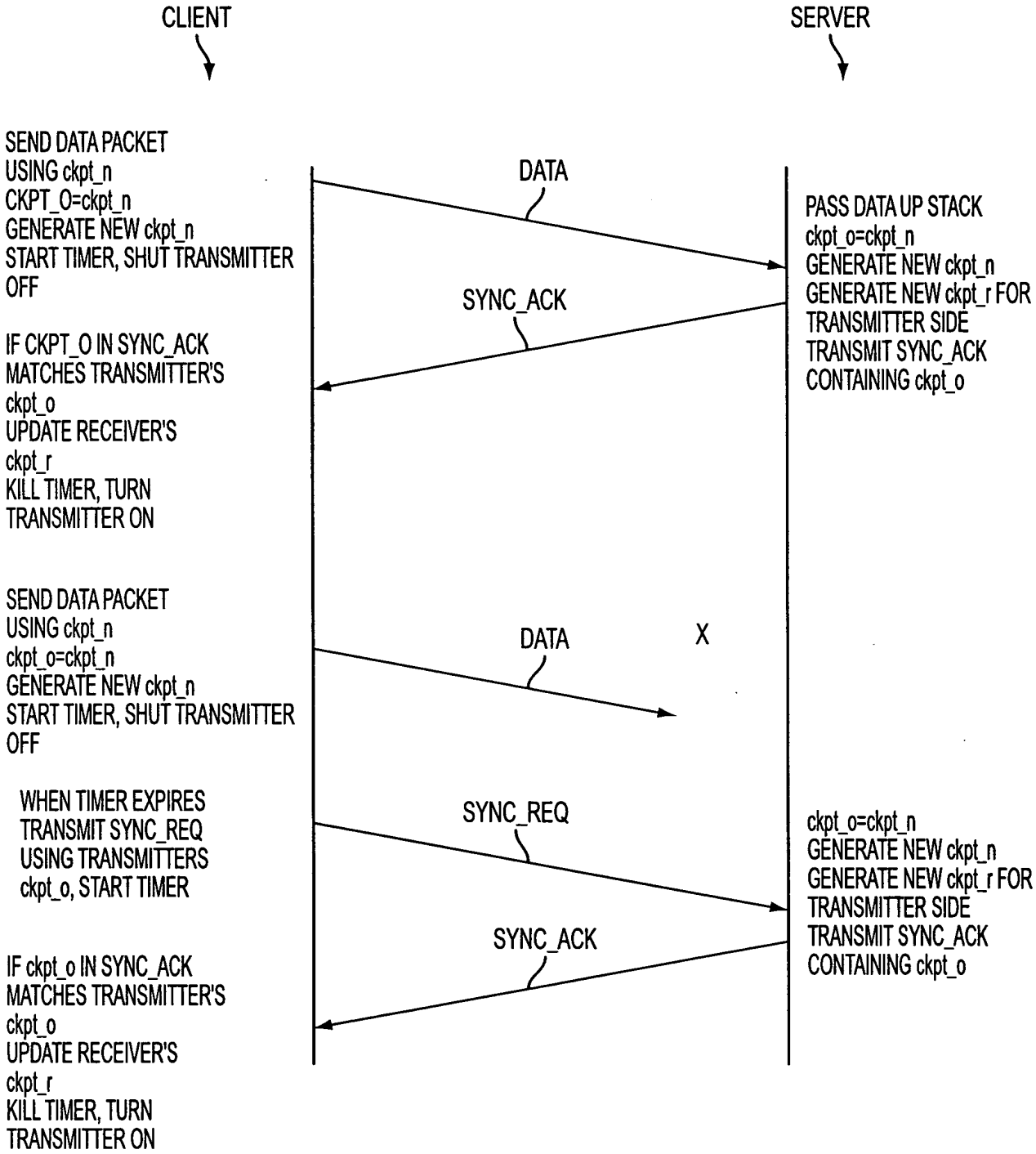

CLIENT
SERVER
SEND DATA PACKET
USING ckpt_n
CKPT_O=ckpt_n
GENERATE NEW ckpt_n
START TIMER, SHUT TRANSMITTER OFF
DATA
PASS DATA UP STACK
ckpt_o=ckpt_n
GENERATE NEW ckpt_n
GENERATE NEW ckpt_r FOR TRANSMITTER SIDE
TRANSMIT SYNC_ACK CONTAINING ckpt_o
SYNC_ACK
IF CKPT_O IN SYNC_ACK MATCHES TRANSMITTER'S ckpt_o
UPDATE RECEIVER'S ckpt_r
KILL TIMER, TURN TRANSMITTER ON
SEND DATA PACKET
USING ckpt_n
ckpt_o=ckpt_n
GENERATE NEW ckpt_n
START TIMER, SHUT TRANSMITTER OFF
DATA
X
WHEN TIMER EXPIRES
TRANSMIT SYNC_REQ
USING TRANSMITTERS
ckpt_o, START TIMER
SYNC_REQ
ckpt_o=ckpt_n
GENERATE NEW ckpt_n
GENERATE NEW ckpt_r FOR TRANSMITTER SIDE
TRANSMIT SYNC_ACK CONTAINING ckpt_o
SYNC_ACK
IF ckpt_o IN SYNC_ACK MATCHES TRANSMITTER'S ckpt_o
UPDATE RECEIVER'S ckpt_r
KILL TIMER, TURN TRANSMITTER ON

FIG. 32

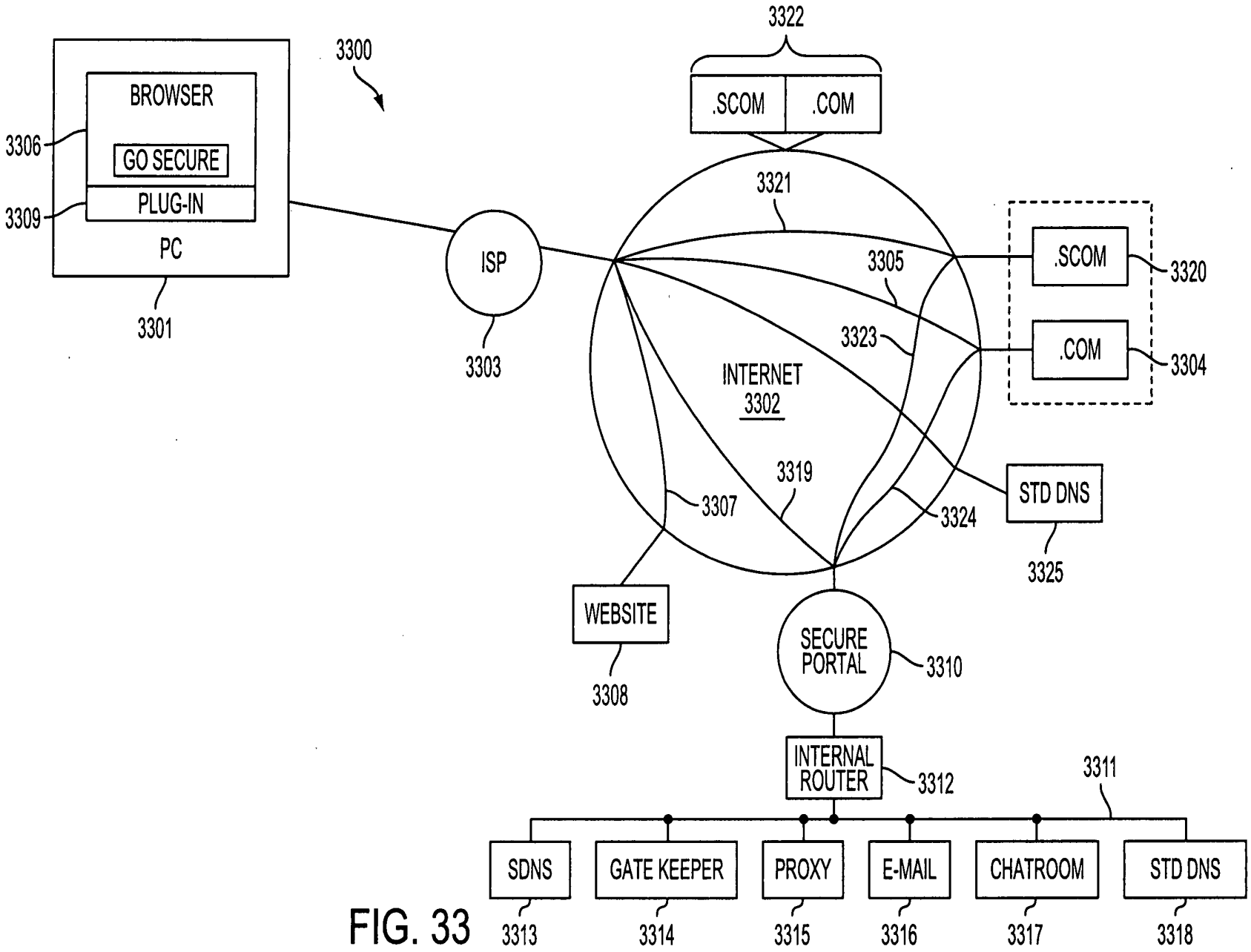


FIG. 33

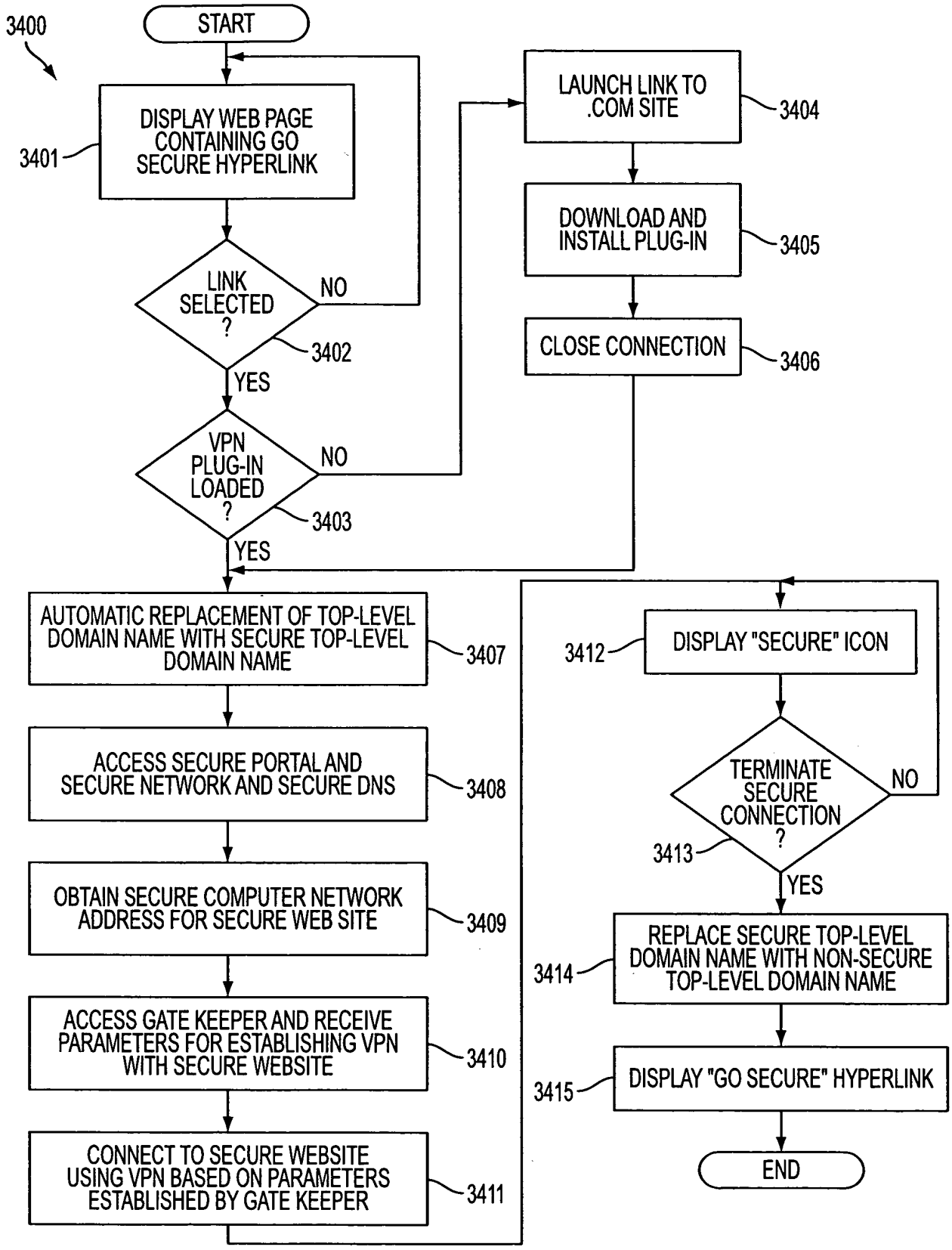

3400
START
DISPLAY WEB PAGE CONTAINING GO SECURE HYPERLINK
3401
LINK SELECTED ?
NO
3402
YES
VPN PLUG-IN LOADED ?
NO
3403
YES
LAUNCH LINK TO .COM SITE
3404
DOWNLOAD AND INSTALL PLUG-IN
3405
CLOSE CONNECTION
3406
AUTOMATIC REPLACEMENT OF TOP-LEVEL DOMAIN NAME WITH SECURE TOP-LEVEL DOMAIN NAME
3407
ACCESS SECURE PORTAL AND SECURE NETWORK AND SECURE DNS
3408
OBTAIN SECURE COMPUTER NETWORK ADDRESS FOR SECURE WEB SITE
3409
ACCESS GATE KEEPER AND RECEIVE PARAMETERS FOR ESTABLISHING VPN WITH SECURE WEBSITE
3410
CONNECT TO SECURE WEBSITE USING VPN BASED ON PARAMETERS ESTABLISHED BY GATE KEEPER
3411
3412
DISPLAY "SECURE" ICON
TERMINATE SECURE CONNECTION ?
NO
3413
YES
REPLACE SECURE TOP-LEVEL DOMAIN NAME WITH NON-SECURE TOP-LEVEL DOMAIN NAME
3414
DISPLAY "GO SECURE" HYPERLINK
3415
END

FIG. 34

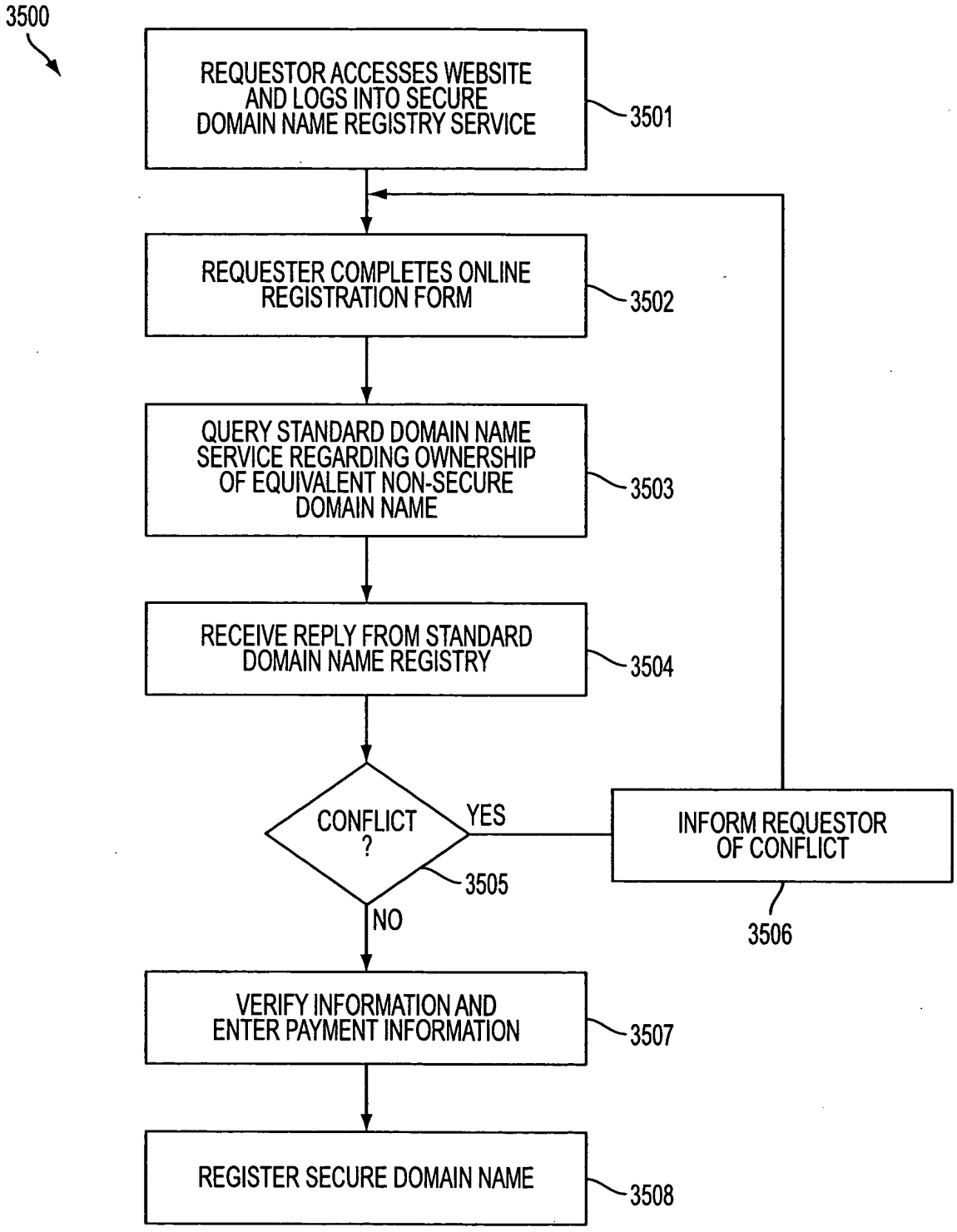

3500
REQUESTOR ACCESSES WEBSITE AND LOGS INTO SECURE DOMAIN NAME REGISTRY SERVICE
3501
REQUESTER COMPLETES ONLINE REGISTRATION FORM
3502
QUERY STANDARD DOMAIN NAME SERVICE REGARDING OWNERSHIP OF EQUIVALENT NON-SECURE DOMAIN NAME
3503
RECEIVE REPLY FROM STANDARD DOMAIN NAME REGISTRY
3504
CONFLICT ?
YES
3505
NO
INFORM REQUESTOR OF CONFLICT
3506
VERIFY INFORMATION AND ENTER PAYMENT INFORMATION
3507
REGISTER SECURE DOMAIN NAME
3508

FIG. 35

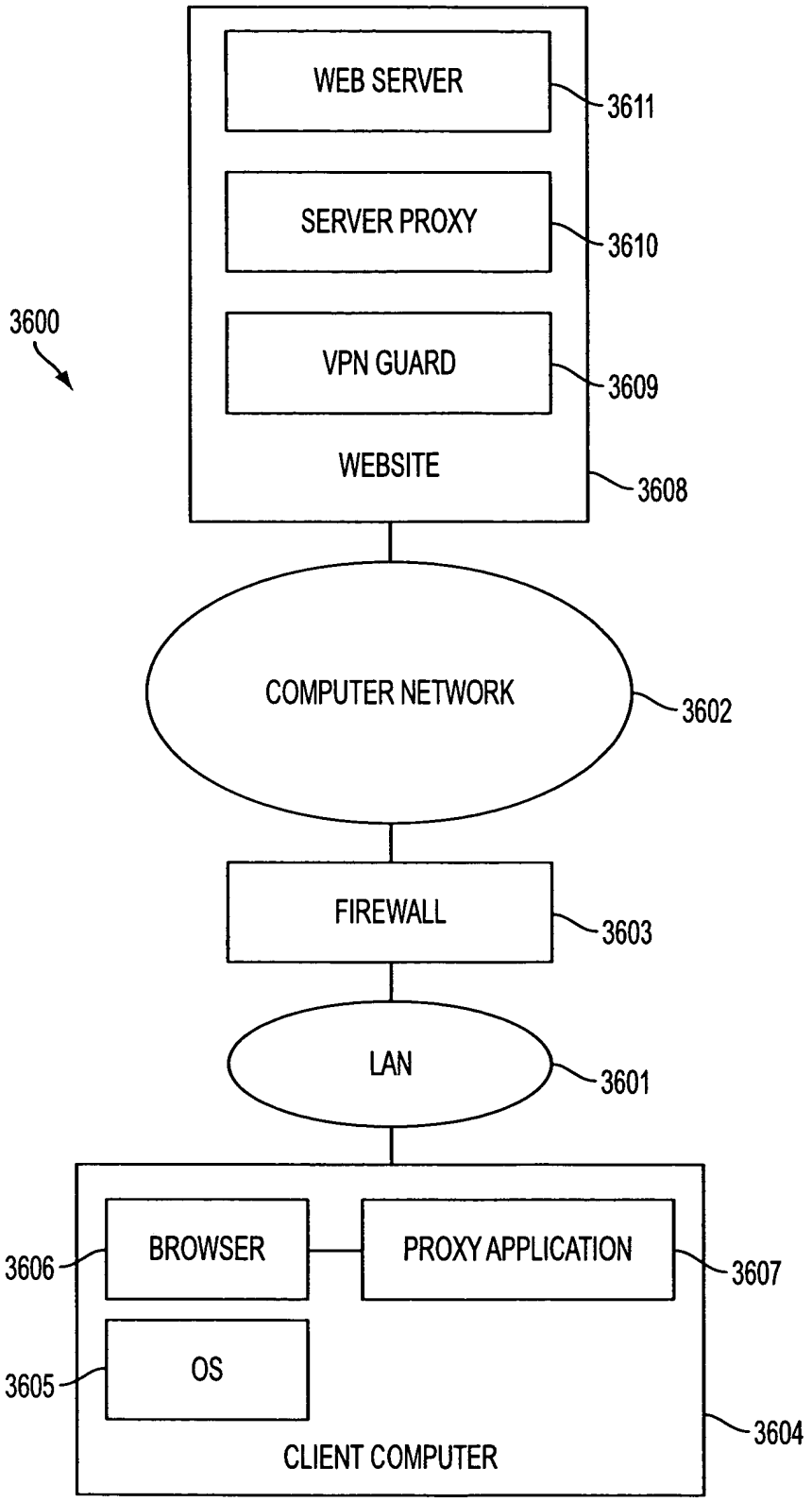

3600
WEB SERVER
3611
SERVER PROXY
3610
VPN GUARD
3609
WEBSITE
3608
COMPUTER NETWORK
3602
FIREWALL
3603
LAN
3601
BROWSER
PROXY APPLICATION
3606
3607
OS
3605
CLIENT COMPUTER
3604

FIG. 36

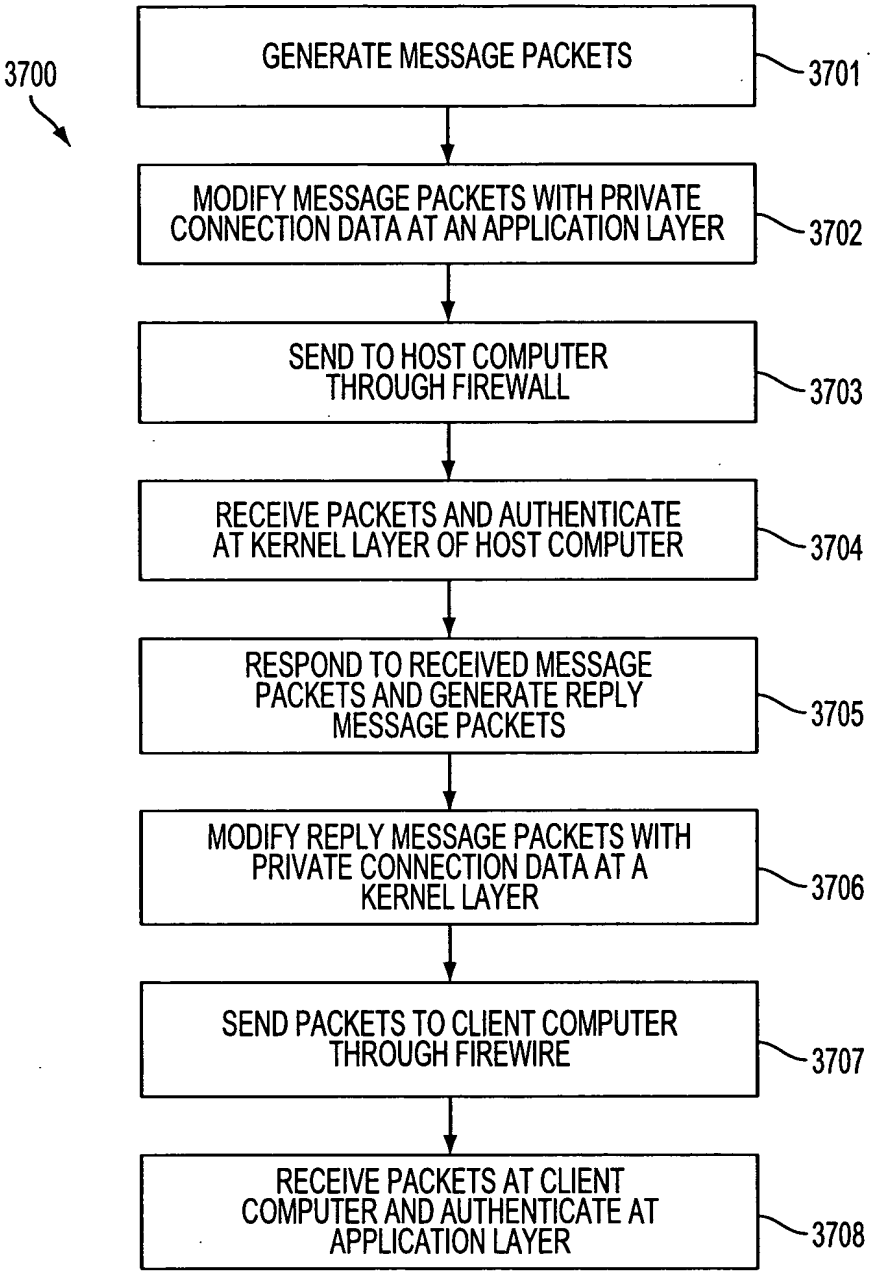

3700
GENERATE MESSAGE PACKETS
3701
MODIFY MESSAGE PACKETS WITH PRIVATE CONNECTION DATA AT AN APPLICATION LAYER
3702
SEND TO HOST COMPUTER THROUGH FIREWALL
3703
RECEIVE PACKETS AND AUTHENTICATE AT KERNEL LAYER OF HOST COMPUTER
3704
RESPOND TO RECEIVED MESSAGE PACKETS AND GENERATE REPLY MESSAGE PACKETS
3705
MODIFY REPLY MESSAGE PACKETS WITH PRIVATE CONNECTION DATA AT A KERNEL LAYER
3706
SEND PACKETS TO CLIENT COMPUTER THROUGH FIREWIRE
3707
RECEIVE PACKETS AT CLIENT COMPUTER AND AUTHENTICATE AT APPLICATION LAYER
3708

FIG. 37

Banner & Witcoff Ref. No. 000479.00112
Client Ref. No. 10006-Div. (1)

JOINT DECLARATION FOR PATENT APPLICATION

As the below named inventors, we hereby declare that:

Our residence, post office address and citizenship are as stated below next to our names;

We believe we are the original, first and joint inventors of the subject matter which is claimed and for which a patent is sought on the invention entitled METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN COMPUTERS OF VIRTUAL PRIVATE NETWORK, the specification of which

- ☒ is attached hereto.
- ☐ was filed on _____as Application Serial Number _____and was amended on _____ (if applicable).
- ☐ was filed under the Patent Cooperation Treaty (PCT) and accorded International Application No. _____, filed _____, and amended on _____ (if any).

We hereby state that we have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

We hereby acknowledge the duty to disclose information which is material to patentability in accordance with Title 37, Code of Federal Regulations, §1.56(a).

Prior Foreign Application(s)

We hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application(s) for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Country	Application No.	Date of Filing (day month year)	Date of Issue (day month year)	Priority Claimed Under 35 U.S.C. §119

Prior United States Provisional Application(s)

We hereby claim priority benefits under Title 35, United States Code, §119(e)(1) of any U.S. provisional application listed below:

U.S. Provisional Application No.	Date of Filing (day month year)	Priority Claimed Under 35 U.S.C. §119(e)(1)
60/106,261	30 October 1998	Yes
60/137,704	7 June 1999	Yes

Prior United States Application(s)

We hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §112, we acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

Banner & Witcoff Ref. No. 000479.00112
Client Ref. No. 10006-Div. (1)

Application Serial No.	Date of Filing (Day, Month, Year)	Status — Patented, Pending, Abandoned
09/558,209	26 April 2000	Pending
09/504,783	15 February 2000	Patented
09/429,643	29 October 1999	Pending

Power of Attorney

And we hereby appoint, both jointly and severally, as our attorneys with full power of substitution and revocation, to prosecute this application and to transact all business in the Patent and Trademark Office connected herewith the practitioners at:

Customer Number: 22907 (WDC)

Please address all correspondence and telephone communications to the address and telephone number for this Customer Number.

We hereby declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Signature Victor Larson Date 11/6/03
Full Name of First Inventor Larson (Family Name) Victor (First Given Name) (Second Given Name)
Residence Fairfax, Virginia Citizenship USA
Post Office Address 12026 Lisa Marie Court, Fairfax, Virginia 22033

Signature Robert D. Short Date 10/27/03
Full Name of Second Inventor Short, III (Family Name) Robert (First Given Name) Dunham (Second Given Name)
Residence Leesburg, Virginia Citizenship USA
Post Office Address 38710 Goose Creek Lane, Leesburg, Virginia 20175

Signature Edmund Colby Munger Date 11/5/03
Full Name of Third Inventor Munger (Family Name) Edmund (First Given Name) Colby (Second Given Name)
Residence Crownsville, Maryland Citizenship USA
Post Office Address 1101 Opaca Court, Crownsville, Maryland 21032

Signature Michael Williamson Date 11/5/03
Full Name of Fourth Inventor Williamson (Family Name) Michael (First Given Name) (Second Given Name)
Residence South Riding, Virginia Citizenship USA
Post Office Address 26203 Ocala Circle, South Riding, Virginia 20152

Application Data Sheet

Application Information

Application number::	
Filing Date::	
Application Type::	Regular
Subject Matter::	Utility
Suggested classification::	
Suggested Group Art Unit::	
CD-ROM or CD-R?::	None
Number of CD disks::	
Number of copies of CDs::	
Sequence submission?::	
Computer Readable Form (CRF)?::	
Number of copies of CRF::	
Title::	METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN COMPUTERS OF VIRTUAL PRIVATE NETWORK
Attorney Docket Number::	000479.00112
Request for Early Publication?::	NO
Request for Non-Publication?::	NO
Suggested Drawing Figure::	1
Total Drawing Sheets::	40
Small Entity?::	NO
Latin name::	
Variety denomination name::	
Petition included?::	NO
Petition Type::	
Licensed US Govt. Agency::	

Contract or Grant Numbers::
Secrecy Order in Parent Appl.?:: NO

Applicant Information

Applicant Authority Type:: Inventor
Primary Citizenship Country:: USA
Status:: Full Capacity
Given Name:: Victor
Middle Name::
Family Name:: Larson
Name Suffix::
City of Residence:: Fairfax
State or Province of Residence:: VA
Country of Residence:: USA
Street of mailing address:: 12026 Lisa Marie Court
City of mailing address:: Fairfax
State or Province of mailing address:: VA
Country of mailing address:: USA
Postal or Zip Code of mailing address:: 22033

Applicant Authority Type:: Inventor
Primary Citizenship Country:: USA
Status:: Full Capacity
Given Name:: Robert
Middle Name:: Durham
Family Name:: Short
Name Suffix:: III
City of Residence:: Leesburg
State or Province of Residence:: VA
Country of Residence:: USA
38710 Goose Creek Lane

Street of mailing address::	38710 Goose Creek Lane
City of mailing address::	Leesburg
State or Province of mailing address::	VA
Country of mailing address::	USA
Postal or Zip Code of mailing address::	20175
Applicant Authority Type::	Inventor
Primary Citizenship Country::	USA
Status::	Full Capacity
Given Name::	Edmund
Middle Name::	Colby
Family Name::	Munger
Name Suffix::	
City of Residence::	Crownsville
State or Province of Residence::	MD
Country of Residence::	USA
Street of mailing address::	1101 Opaca Court
City of mailing address::	Crownsville
State or Province of mailing address::	MD
Country of mailing address::	USA
Postal or Zip Code of mailing address::	21032
Applicant Authority Type::	Inventor
Primary Citizenship Country::	USA
Status::	Full Capacity
Given Name::	Michael
Middle Name::	
Family Name::	Williamson
Name Suffix::	
City of Residence::	South Riding

State or Province of Residence:: VA
Country of Residence:: USA
Street of mailing address:: 26203 Ocla Circle
City of mailing address:: South Riding
State or Province of mailing address:: VA
Country of mailing address:: USA
Postal or Zip Code of mailing address:: 20152

Correspondence Information

Correspondence Customer Number:: 22907

Representative Information

Representative Customer Number:: 22907

Domestic Priority Information

Application::	Continuity Type::	Parent Application::	Parent Filing Date::
This Application	Division of	09/558,209	04/26/00
	Continuation-in-Part of	09/504,783	02/15/00
	Continuation-in-Part of	09/429,643	10/29/99
	Provisional	60/106,261	10/30/98
	Provisional	60/137,704	06/07/99

Foreign Priority Information

Country::	Application number::	Filing Date::	Priority Claimed::

Assignee Information

Assignee name::	Science Applications International Corporation
Street of mailing address::	10260 Campus Point Drive
City of mailing address::	San Diego
State or Province of mailing address::	CA
Country of mailing address::	USA
Postal or Zip Code of mailing address::	92121

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of	)	
	)	
Victor Larson et al.	)	Group Art Unit: TBA
	)	
Serial No.: TBA (DIV of 09/558,209)	)	Examiner: TBA
	)	
	)	
Filed: Herewith	)	Atty. Docket No.: 00479.00112
	)	

For: METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN COMPUTERS OF VIRTUAL PRIVATE NETWORK

INFORMATION DISCLOSURE STATEMENT

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

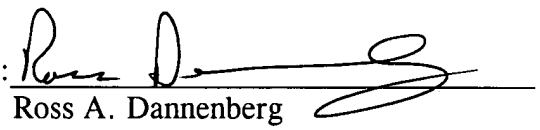
Sir:

Pursuant to 37 C.F.R. 1.56, the attention of the Patent and Trademark Office is hereby directed to the reference(s) listed on the attached PTO-1449. A copy of each cited prior art reference was provided or cited in the prior application in accordance with 37 C.F.R. 1.98(d). It is respectfully requested that the information be expressly considered during the prosecution of this application, and that the reference(s) be made of record therein and appear among the "References Cited" on any patent to issue therefrom.

Applicant does not waive any right to take appropriate action to establish patentability over the listed documents should they be applied as a reference against the claims of the present application.

The accompanying Information Disclosure Statement is being filed within three months of the U.S. filing date OR before the mailing date of a first Office Action on the merits. No certification or fee is required.

Respectfully submitted,
BANNER & WITCOFF, LTD.

By: 
Ross A. Dannenberg
Registration No. 49,024

1001 G Street, N.W.
Eleventh Floor
Washington, D.C. 20001-4597
(202) 824-3000

Dated: November 7, 2003

PTO-1449 (Modified) U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE INFORMATION DISCLOSURE STATEMENT BY APPLICANT	ATTY. DOCKET NO. 000479.00112	SERIAL NUMBER TBD
	APPLICANT Victor Larson et al.	
	FILING DATE Herewith	GROUP ART UNIT TBD

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	NAME	CLASS	SUB CLASS	FILING DATE
	6,119,171	9/2000	Alkhatib			
	5,588,060	12/24/96	Aziz			
	5,689,566	11/18/9	Nguyen			
	5,842,040	11/24/98	Hughes et al.			
	4,933,846	06/12/90	Humphrey et al.			

FOREIGN PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUB CLASS	TRANSLATION YES/NO	
	199 24 575	12/2/99	DE				
	0 838 930	4/29/98	EPO				
	2 317 792	4/1/98	GB				
	0 814 589	12/29/97	EPO				
	WO 98/27783	6/25/98	PCT				

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

	Search Report (dated 6/18/02), International Application No. PCT/US01/13260
	Search Report (dated 6/28/02), International Application No. PCT/US01/13261
	Donald E. Eastlake, "Domain Name System Security Extensions", DNS Security Working Group, April 1998, 51 pages
	D. B. Chapman et al., "Building Internet Firewalls", November 1995, pages 278-297 and pages 351-375
	P. Srisuresh et al., "DNS extensions to Network Address Translators", July 1998, 27 pages
	Laurie Wells, "Security Icon", October 19, 1998, 1 page
	W. Stallings, "Cryptography And Network Security", 2nd Edition, Chapter 13, IP Security, June 8, 1998, pages 399-440
	W. Stallings, "New Cryptography and Network Security Book", June 8, 1998, 3 pages
	FASBENDER, KESDOGAN, and KUBITZ: "Variable and Scalable Security: Protection of Location Information in Mobile IP", IEEE publication, 1996, pages 963-967

EXAMINER	DATE CONSIDERED
EXAMINER: Initial citation if reference was considered. Draw line through citation if not in conformance to MPEP 609 and not considered. Include copy of this form with next communication to applicant.	

PTO-1449 (Modified) U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE INFORMATION DISCLOSURE STATEMENT BY APPLICANT	ATTY. DOCKET NO. 000479.00112	SERIAL NUMBER TBD
	APPLICANT Victor Larson et al.	
	FILING DATE Herewith	GROUP ART UNIT TBD

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	NAME	CLASS	SUB CLASS	FILING DATE
	6,353,614	3/5/02	Borella et al.			
	6,332,158	12/18/01	Risley et al.			
	6,330,562	12/11/01	Boden et al.			
	6,286,047	9/4/01	Ramanathan et al.			
	6,243,749	6/5/01	Sitaraman et al.			
	6,226,751	5/1/01	Arrow et al.			
	6,178,505	1/23/01	Schneider et al.			
	6,119,171	9/12/00	Alkhatib			
	6,079,020	6/20/00	Liu			
	6,052,788	4/18/00	Wesinger, Jr. et al.			
	6,016,318	1/18/00	Tomoike			
	6,006,259	12/21/99	Adelman et al.			

FOREIGN PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUB CLASS	TRANSLATION YES/NO	

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

EXAMINER	DATE CONSIDERED
EXAMINER: Initial citation if reference was considered. Draw line through citation if not in conformance to MPEP 609 and not considered. Include copy of this form with next communication to applicant.	

PTO-1449 (Modified) U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE INFORMATION DISCLOSURE STATEMENT BY APPLICANT	ATTY. DOCKET NO. 000479.00112	SERIAL NUMBER TBD
	APPLICANT Victor Larson et al.	
	FILING DATE Herewith	GROUP ART UNIT TBD

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	NAME	CLASS	SUB CLASS	FILING DATE
	5,905,859	5/18/99	Holloway et al.			
	5,898,830	4/27/99	Wesinger, Jr. et al.			
	5,892,903	4/6/99	Klaus			
	5,878,231	3/2/99	Baehr et al.			
	5,805,801	9/8/98	Holloway et al.			
	5,796,942	8/18/98	Esbensen			

FOREIGN PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUB CLASS	TRANSLATION YES/NO	
	WO 00/70458	11/23/00	PCT				

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

	Linux FreeS/WAN Index File, printed from http://liberty.freeswan.org/freeswan_trees/freeswan-1.3/doc/ on February 21, 2002, 3 Pages
	J. Gilmore, "Swan: Securing the Internet against Wiretapping", printed from http://liberty.freeswan.org/freeswan_trees/freeswan-1.3/doc/rationale.html on February 21, 2002, 4 pages
	Glossary for the Linux FreeS/WAN project, printed from http://liberty.freeswan.org/freeswan_trees/freeswan-1.3/doc/glossary.html on February 21, 2002, 25 pages
	Alan O. Frier et al., "The SSL Protocol Version 3.0", November 18, 1996, printed from http://www.netscape.com/eng/ssl3/draft302.txt on February 4, 2002, 56 pages

EXAMINER	DATE CONSIDERED
EXAMINER: Initial citation if reference was considered. Draw line through citation if not in conformance to MPEP 609 and not considered. Include copy of this form with next communication to applicant.	

PTO-1449 (Modified) U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE INFORMATION DISCLOSURE STATEMENT BY APPLICANT	ATTY. DOCKET NO. 000479.00112	SERIAL NUMBER TBD
	APPLICANT Victor Larson et al.	
	FILING DATE Herewith	GROUP ART UNIT TBD

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	NAME	CLASS	SUB CLASS	FILING DATE
	5,341,426	8/1994	Barney et al.			
	5,787,172	7/1998	Terry Sutton Arnold			
	6,092,200	7/2000	Muniyappa et al.			
	6,158,011	12/2000	Chen et al.			
	6,168,409	1/2001	Weber et al.			

FOREIGN PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUB CLASS	TRANSLATION YES/NO	
	0 858 189	8/12/98	EPO				
	WO 01 50688	7/12/01	PCT				
	WO 98 59470	12/30/98	PCT				
	WO 99 48303	9/23/99	PCT				
	WO 99 38081	7/29/99	PCT				

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

	Search Report (dated 8/20/02), International Application No. PCT/US01/04340
	Search Report (dated 8/23/02), International Application No. PCT/US01/13260
	Shree Murthy et al., "Congestion-Oriented Shortest Multipath Routing", Proceedings of IEEE INFOCOM, 1996, pages 1028-1036
	Jim Jones et al., "Distributed Denial of Service Attacks: Defenses", Global Integrity Corporation, 2000, pages 1-14
	James E. Bellaire, "New Statement of Rules – Naming Internet Domains", Internet Newsgroup, July 30, 1995, 1 page
	D. Clark, "US Calls for Private Domain-Name System", Computer, IEEE Computer Society, August 1, 1998, pages 22-25
	August Bequai, "Balancing Legal Concerns Over Crime and Security in Cyberspace", Computer & Security, Vol. 17, No. 4, 1998, pages 293-298
	Rich Winkel, "CAQ: Networking With Spooks: The NET & The Control Of Information", Internet Newsgroup, June 21, 1997, 4 pages

EXAMINER	DATE CONSIDERED
EXAMINER: Initial citation if reference was considered. Draw line through citation if not in conformance to MPEP 609 and not considered. Include copy of this form with next communication to applicant.	

PTO-1449 (Modified) U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE INFORMATION DISCLOSURE STATEMENT BY APPLICANT	ATTY. DOCKET NO. 000479.00112	SERIAL NUMBER TBD
	APPLICANT Victor Larson et al.	
	FILING DATE Herewith	GROUP ART UNIT TBD

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	NAME	CLASS	SUB CLASS	FILING DATE

FOREIGN PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUB CLASS	TRANSLATION YES/NO	
	WO 98 55930	12/10/98	PCT				

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

	Search Report (dated 10/7/02), International Application No. PCT/US01/13261
	F. Halsall, "Data Communications, Computer Networks And Open Systems", Chapter 4, Protocol Basics, 1996, pages 198-203
	Reiter, Michael K. and Rubin, Aviel D. (AT&T Labs – Research), "Crowds: Anonymity for Web Transmissions", pages 1-23
	Dolev, Shlomi and Ostrovsky, Rafil, "Efficient Anonymous Multicast and Reception" (Extended Abstract), 16 pages
	Rubin, Aviel D., Greer, Daniel, and Ranum, Marcus J. (Wiley Computer Publishing), "Web Security Sourcebook", pages 82-94
	FASBENDER, KESDOGAN, and KUBITZ: "Variable and Scalable Security" Protection of Location Information in Mobile IP", IEEE publication, 1996, pages 963-967

EXAMINER	DATE CONSIDERED
EXAMINER: Initial citation if reference was considered. Draw line through citation if not in conformance to MPEP 609 and not considered. Include copy of this form with next communication to applicant.	

PATENT APPLICATION FEE DETERMINATION RECORD
Effective October 1, 2003

Application or Docket Number: 10702486

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS	24	
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	24 minus 20=	* 4
INDEPENDENT CLAIMS	2 minus 3 =	* 0
MULTIPLE DEPENDENT CLAIM PRESENT		☐

* If the difference in column 1 is less than zero, enter "0" in column 2

SMALL ENTITY TYPE ☐		OR	OTHER THAN SMALL ENTITY	
RATE	FEE		RATE	FEE
BASIC FEE	385.00	OR	BASIC FEE	770.00
X$ 9=		OR	X$18=	72
X43=		OR	X86=	
+145=		OR	+290=	
TOTAL		OR	TOTAL	842

CLAIMS AS AMENDED - PART II

AMENDMENT A

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X43=		OR	X86=	
+145=		OR	+290=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT B

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X43=		OR	X86=	
+145=		OR	+290=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT C

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X43=		OR	X86=	
+145=		OR	+290=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
***If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

PATENT APPLICATION SERIAL NO. _____

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE
FEE RECORD SHEET

11/10/2003 HLE333 00000016 190733 10702486

01 FC:1001 770.00 DA
02 FC:1202 72.00 DA

PTO-1556
(5/87)

*U.S. Government Printing Office: 2002 — 489-267/69033

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/702,486	11/07/2003	Victor Larson	000479.00112	8949

22907 7590 05/19/2006

BANNER & WITCOFF
1001 G STREET N W
SUITE 1100
WASHINGTON, DC 20001

EXAMINER	
LIM, KRISNA	

ART UNIT	PAPER NUMBER
2153	

DATE MAILED: 05/19/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/702,486	LARSON ET AL.	
	Examiner	Art Unit	
	Krisna Lim	2153	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☐ Responsive to communication(s) filed on _____.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-24 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☐ Claim(s) _____ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

1. Claims 1-24 are presented for examination.

2. The title of the invention is neither descriptive nor precise. A new title is required which should include, using twenty words or fewer, claimed features that differentiate the invention from the Prior Art. The title should reflect the gist of or the improvement of the present invention.

3. The disclosure is objected to because of the following informalities:

(a) On page 1, the text of the first paragraph should be updated with the current status of the cited applications such as U.S. Patent Application Serial No., a filing date, U.S. Patent No., and the issued date. Appropriate correction is required.

4. Claims 1-2 are rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

In claim 1, line 3, it is unclear from a query message is sent. At line 4, it unclear from the query message is requesting a secure computer network address.

At line 5, it is unclear where the response message is received and from where is the response message is received. At line 7, it is unclear from where an access request is sent.

In claim 2, it is unclear from where a command is received and where a secure domain name is generated.

In claim 3, it is unclear from where a command is received.

Claims 12-14 contain similar problems as in claims 1-3.

5. Claims 1 and 12 would be allowable if rewritten or amended to overcome the rejection(s) under 35 U.S.C. 112, 2nd paragraph, set forth in this Office action.

Claims 2-3 and 13-14 would be allowable if rewritten to overcome the rejection(s) under 35 U.S.C. 112, 2nd paragraph, set forth in this Office action and to include all of the limitations of the base claim and any intervening claims.

6. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

The references are cited in the Form PTO-892 for the applicant's review.

A shortened statutory period for response to this action is set to expire 3 (three) months and 0 (zero) days from the mail date of this letter. Failure to respond within the period for response will result in **ABANDONMENT** of the application (see 35 U.S.C 133, M.P.E.P 710.02, 710.02(b)).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Krisna Lim whose telephone number is 571-272-3956 The examiner can normally be reached on Monday to Wednesday and Friday from 7:30 AM to 4:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Glenton Burgess, can be reached on 571-272-3949. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

KI

May 15, 2006

KRISNA LIM
PRIMARY EXAMINER

PTO-1449 (Modified) U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE INFORMATION DISCLOSURE STATEMENT BY APPLICANT	ATTY. DOCKET NO. 000479.00112	SERIAL NUMBER ~~TBD~~ 10/702,486
	APPLICANT Victor Larson et al.	
	FILING DATE ~~Herewith~~ 11/7/2003	GROUP ART UNIT ~~TBD~~ 2153

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	NAME	CLASS	SUB CLASS	FILING DATE
VL	6,119,171	9/2000	Alkhatib			
VL	5,588,060	12/24/96	Aziz			
VL	5,689,566	11/18/9	Nguyen			
VL	5,842,040	11/24/98	Hughes et al.			
VL	4,933,846	06/12/90	Humphrey et al.			

FOREIGN PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUB CLASS	TRANSLATION YES/NO	
VL	199 24 575	12/2/99	DE				
	0 838 930	4/29/98	EPO				
	2 317 792	4/1/98	GB				
	0 814 589	12/29/97	EPO				
V	WO 98/27783	6/25/98	PCT				

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

VL	Search Report (dated 6/18/02), International Application No. PCT/US01/13260
	Search Report (dated 6/28/02), International Application No. PCT/US01/13261
	Donald E. Eastlake, "Domain Name System Security Extensions", DNS Security Working Group, April 1998, 51 pages
	D. B. Chapman et al., "Building Internet Firewalls", November 1995, pages 278-297 and pages 351-375
	P. Srisuresh et al., "DNS extensions to Network Address Translators", July 1998, 27 pages
	Laurie Wells, "Security Icon", October 19, 1998, 1 page
	W. Stallings, "Cryptography And Network Security", 2nd Edition, Chapter 13, IP Security, June 8, 1998, pages 399-440
	W. Stallings, "New Cryptography and Network Security Book", June 8, 1998, 3 pages
V	FASBENDER, KESDOGAN, and KUBITZ: "Variable and Scalable Security: Protection of Location Information in Mobile IP", IEEE publication, 1996, pages 963-967

EXAMINER KRISNA LIM	DATE CONSIDERED 5/15/06

EXAMINER: Initial citation if reference was considered. Draw line through citation if not in conformance to MPEP 609 and not considered. Include copy of this form with next communication to applicant.

PTO-1449 (Modified) U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE INFORMATION DISCLOSURE STATEMENT BY APPLICANT	ATTY. DOCKET NO. 000479.00112	SERIAL NUMBER TBD
	APPLICANT Victor Larson et al.	
	FILING DATE Herewith	GROUP ART UNIT TBD

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	NAME	CLASS	SUB CLASS	FILING DATE
KL	6,353,614	3/5/02	Borella et al.			
	6,332,158	12/18/01	Risley et al.			
	6,330,562	12/11/01	Boden et al.			
	6,286,047	9/4/01	Ramanathan et al.			
	6,243,749	6/5/01	Sitaraman et al.			
	6,226,751	5/1/01	Arrow et al.			
	6,178,505	1/23/01	Schneider et al.			
	6,119,171	9/12/00	Alkhatib			
	6,079,020	6/20/00	Liu			
	6,052,788	4/18/00	Wesinger, Jr. et al.			
	6,016,318	1/18/00	Tomoike			
	6,006,259	12/21/99	Adelman et al.			

FOREIGN PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUB CLASS	TRANSLATION YES/NO	

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

EXAMINER	KRISNA LIM	DATE CONSIDERED	5/15/06

EXAMINER: Initial citation if reference was considered. Draw line through citation if not in conformance to MPEP 609 and not considered. Include copy of this form with next communication to applicant.

PTO-1449 (Modified) U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE INFORMATION DISCLOSURE STATEMENT BY APPLICANT	ATTY. DOCKET NO. 000479.00112	SERIAL NUMBER TBD
	APPLICANT Victor Larson et al.	
	FILING DATE Herewith	GROUP ART UNIT TBD

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	NAME	CLASS	SUB CLASS	FILING DATE
VC	5,905,859	5/18/99	Holloway et al.			
	5,898,830	4/27/99	Wesinger, Jr. et al.			
	5,892,903	4/6/99	Klaus			
	5,878,231	3/2/99	Baehr et al.			
	5,805,801	9/8/98	Holloway et al.			
	5,796,942	8/18/98	Esbensen			

FOREIGN PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUB CLASS	TRANSLATION YES/NO	
	WO 00/70458	11/23/00	PCT				

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

VC	Linux FreeS/WAN Index File, printed from http://liberty.freeswan.org/freeswan_trees/freeswan-1.3/doc/ on February 21, 2002, 3 Pages
VC	J. Gilmore, "Swan: Securing the Internet against Wiretapping", printed from http://liberty.freeswan.org/freeswan_trees/freeswan-1.3/doc/rationale.html on February 21, 2002, 4 pages
VC	Glossary for the Linux FreeS/WAN project, printed from http://liberty.freeswan.org/freeswan_trees/freeswan-1.3/doc/glossary.html on February 21, 2002, 25 pages
VC	Alan O. Frier et al., "The SSL Protocol Version 3.0", November 18, 1996, printed from http://www.netscape.com/eng/ssl3/draft302.txt on February 4, 2002, 56 pages

EXAMINER KRISNA LIM	DATE CONSIDERED 5/15/06

EXAMINER: Initial citation if reference was considered. Draw line through citation if not in conformance to MPEP 609 and not considered. Include copy of this form with next communication to applicant.

PTO-1449 (Modified) U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE INFORMATION DISCLOSURE STATEMENT BY APPLICANT	ATTY. DOCKET NO. 000479.00112	SERIAL NUMBER TBD
	APPLICANT Victor Larson et al.	
	FILING DATE Herewith	GROUP ART UNIT TBD

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	NAME	CLASS	SUB CLASS	FILING DATE
KL	5,341,426	8/1994	Barney et al.			
	5,787,172	7/1998	Terry Sutton Arnold			
	6,092,200	7/2000	Muniyappa et al.			
	6,158,011	12/2000	Chen et al.			
V	6,168,409	1/2001	Weber et al.			

FOREIGN PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUB CLASS	TRANSLATION YES/NO	
KL	0 858 189	8/12/98	EPO				
	WO 01 50688	7/12/01	PCT				
	WO 98 59470	12/30/98	PCT				
	WO 99 48303	9/23/99	PCT				
V	WO 99 38081	7/29/99	PCT				

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

KL	Search Report (dated 8/20/02), International Application No. PCT/US01/04340
	Search Report (dated 8/23/02), International Application No. PCT/US01/13260
	Shree Murthy et al., "Congestion-Oriented Shortest Multipath Routing", Proceedings of IEEE INFOCOM, 1996, pages 1028-1036
	Jim Jones et al., "Distributed Denial of Service Attacks: Defenses", Global Integrity Corporation, 2000, pages 1-14
	James E. Bellaire, "New Statement of Rules – Naming Internet Domains", Internet Newsgroup, July 30, 1995, 1 page
	D. Clark, "US Calls for Private Domain-Name System", Computer, IEEE Computer Society, August 1, 1998, pages 22-25
	August Bequai, "Balancing Legal Concerns Over Crime and Security in Cyberspace", Computer & Security, Vol. 17, No. 4, 1998, pages 293-298
V	Rich Winkel, "CAQ: Networking With Spooks: The NET & The Control Of Information", Internet Newsgroup, June 21, 1997, 4 pages

EXAMINER KRISNA LIM	DATE CONSIDERED 5/15/06

EXAMINER: Initial citation if reference was considered. Draw line through citation if not in conformance to MPEP 609 and not considered. Include copy of this form with next communication to applicant.

PTO-1449 (Modified) U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE INFORMATION DISCLOSURE STATEMENT BY APPLICANT	ATTY. DOCKET NO. 000479.00112	SERIAL NUMBER TBD
	APPLICANT Victor Larson et al.	
	FILING DATE Herewith	GROUP ART UNIT TBD

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	NAME	CLASS	SUB CLASS	FILING DATE

FOREIGN PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUB CLASS	TRANSLATION YES/NO	
KC	WO 98 55930	12/10/98	PCT				

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

KC	Search Report (dated 10/7/02), International Application No. PCT/US01/13261
	F. Halsall, "Data Communications, Computer Networks And Open Systems", Chapter 4, Protocol Basics, 1996, pages 198-203
	Reiter, Michael K. and Rubin, Aviel D. (AT&T Labs – Research), "Crowds: Anonymity for Web Transmissions", pages 1-23
	Dolev, Shlomi and Ostrovsky, Rafil, "Efficient Anonymous Multicast and Reception" (Extended Abstract), 16 pages
	Rubin, Aviel D., Greer, Daniel, and Ranum, Marcus J. (Wiley Computer Publishing), "Web Security Sourcebook", pages 82-94
V	FASBENDER, KESDOGAN, and KUBITZ: "Variable and Scalable Security" Protection of Location Information in Mobile IP", IEEE publication, 1996, pages 963-967

EXAMINER KRISNA LIM	DATE CONSIDERED 5/15/06

EXAMINER: Initial citation if reference was considered. Draw line through citation if not in conformance to MPEP 609 and not considered. Include copy of this form with next communication to applicant.

Notice of References Cited	Application/Control No. 10/702,486	Applicant(s)/Patent Under Reexamination LARSON ET AL.	
	Examiner Krisna Lim	Art Unit 2153	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-			
	B	US-			
	C	US-			
	D	US-			
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Index of Claims

Application/Control No.	Applicant(s)/Patent under Reexamination
10/702,486	LARSON ET AL.
Examiner	**Art Unit**
Krisna Lim	2153

R	Rejected	–	(Through numeral) Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

Claim		Date								
Final	Original	5/15/06								
	1	R								
	2	R								
	3	R								
	4	R								
	5	R								
	6	R								
	7	R								
	8	R								
	9	R								
	10	R								
	11	R								
	12	R								
	13	R								
	14	R								
	15	R								
	16	R								
	17	R								
	18	R								
	19	R								
	20	R								
	21	R								
	22	R								
	23	R								
	24	R								
	25									
	26									
	27									
	28									
	29									
	30									
	31									
	32									
	33									
	34									
	35									
	36									
	37									
	38									
	39									
	40									
	41									
	42									
	43									
	44									
	45									
	46									
	47									
	48									
	49									
	50									

Claim		Date								
Final	Original									
	51									
	52									
	53									
	54									
	55									
	56									
	57									
	58									
	59									
	60									
	61									
	62									
	63									
	64									
	65									
	66									
	67									
	68									
	69									
	70									
	71									
	72									
	73									
	74									
	75									
	76									
	77									
	78									
	79									
	80									
	81									
	82									
	83									
	84									
	85									
	86									
	87									
	88									
	89									
	90									
	91									
	92									
	93									
	94									
	95									
	96									
	97									
	98									
	99									
	100									

Claim		Date								
Final	Original									
	101									
	102									
	103									
	104									
	105									
	106									
	107									
	108									
	109									
	110									
	111									
	112									
	113									
	114									
	115									
	116									
	117									
	118									
	119									
	120									
	121									
	122									
	123									
	124									
	125									
	126									
	127									
	128									
	129									
	130									
	131									
	132									
	133									
	134									
	135									
	136									
	137									
	138									
	139									
	140									
	141									
	142									
	143									
	144									
	145									
	146									
	147									
	148									
	149									
	150									

Search Notes

Application/Control No.	Applicant(s)/Patent under Reexamination	
10/702,486	LARSON ET AL.	
Examiner	**Art Unit**	
Krisna Lim	2153	

SEARCHED			
Class	Subclass	Date	Examiner
709	225, 229, 245	5/15/06	KL

INTERFERENCE SEARCHED			
Class	Subclass	Date	Examiner

SEARCH NOTES (INCLUDING SEARCH STRATEGY)		
	DATE	EXMR
EAST TEXT SEARCH	5/15/2006	KL

RECEIVED
CENTRAL FAX CENTER

AUG 17 2006

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of (first named inventor): **Larson *et al.***		Atty. Docket No.:	000479.00112
Serial No.:	10/702,486	Group Art Unit:	2153
Filed:	November 7, 2003	Examiner:	Lim, Krisna
For:	METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN COMPUTERS OF VIRTUAL PRIVATE NETWORK USING SECURE DOMAIN NAMES (As amended)	Confirmation No.:	8949

AMENDMENT

Mail Stop Amendment
U.S. Patent and Trademark Office
Customer Service Window
Randolph Building
401 Dulany Street
Alexandria, VA 22314

I hereby certify that this correspondence is being facsimile transmitted to the Patent and Trademark Office on August 17, 2006, to (571) 273-8300.
Signature: /Ross Dannenberg/
Ross A. Dannenberg, Reg. No. 49,024

Sir:

In response to the Office Action mailed May 19, 2006, please amend the instant application as follows:

Amendments to the Specification begin on page 2 of this paper.

Amendments to the Claims are reflected in the Listing of Claims, which begins on page 3 of this paper.

Remarks/Arguments begin on page 9 of this paper.

If any fees are required or if an overpayment is made, the Commissioner is authorized to debit or credit our Deposit Account No. 19-0733, accordingly.

PAGE 1/11 * RCVD AT 8/17/2006 11:53:41 AM [Eastern Daylight Time] * SVR:USPTO-EFXRF-5/20 * DNIS:2738300 * CSID:12028243001 * DURATION (mm-ss):04-24

RECEIVED
CENTRAL FAX CENTER
AUG 17 2006

Amendments to the Specification:

Please make the following changes to the title of the invention:

-- METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN COMPUTERS OF VIRTUAL PRIVATE NETWORK USING SECURE DOMAIN NAMES --

Please amend the paragraph on page 1, immediately preceding the "Background Of The Invention," as follows:

--This application claims priority from and is a divisional patent application of co-pending U.S. application serial number 09/558,209, filed April 26, 2000, which is a continuation-in-part patent application of previously-filed U.S. application serial number 09/504,783, filed on February 15, 2000, now U.S. Pat. No. 6,502,135, issued December 31, 2002, which claims priority from and is a continuation-in-part patent application of previously-filed U.S. application serial number 09/429,643, filed on October 29, 1999, now U.S. Pat. No. 7,010,604, issued March 7, 2006. The subject matter of U.S. application serial number 09/429,643, which is bodily incorporated herein, derives from provisional U.S. application numbers 60/106,261 (filed October 30, 1998) and 60/137,704 (filed June 7, 1999). The present application is also related to U.S. application serial number 09/558,210, filed April 26, 2000, and which is incorporated by reference herein.--

PAGE 2/11 * RCVD AT 8/17/2006 11:53:41 AM [Eastern Daylight Time] * SVR:USPTO-EFXRF-5/20 * DNIS:2738300 * CSID:12028243001 * DURATION (mm-ss):04-24

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A method for accessing a secure computer network address, comprising steps of:

receiving a secure domain name;

sending a query message to a secure domain name service, the query message requesting from the secure domain name service a secure computer network address corresponding to the secure domain name;

receiving from the secure domain name service a response message containing the secure computer network address corresponding to the secure domain name; and

sending an access request message to the secure computer network address using a virtual private network communication link.

2. (Original) The method according to claim 1, wherein the step of receiving the secure domain name includes steps of:

receiving a command to establish the virtual private network communication link with a secure computer network address corresponding to a predetermined non-secure domain name; and

automatically generating a secure domain name corresponding to the non-secure domain name.

3. (Original) The method according to claim 2, wherein the step of receiving a command to establish the virtual private network communication link includes a step of selecting a predetermined icon displayed on a computer display.

4. (Original) The method according to claim 1, wherein the response message contains provisioning information for the virtual private network.

5. (Original) The method according to claim 4, wherein the virtual private network is based on inserting one or more data values into each data packet sent to the secure computer network address, the one or more data values varying according to a pseudo-random sequence.

6. (Original) The method according to claim 4, wherein the virtual private network is based on inserting into at least one data packet at least one data value representing a predetermined level of service associated with the virtual private network.

7. (Currently Amended) The method according to claim 4, wherein the virtual private network is based on a computer network address hopping regime that is used to pseudorandomly change computer network addresses in packets transmitted between ~~the~~ a first computer and ~~the~~ a second computer.

8. (Original) The method according to claim 4, wherein the virtual private network is based on comparing a value in each data packet transmitted to the secure computer network address to a moving window of valid values.

9. (Original) The method according to claim 4, wherein the virtual private network is based on a comparison of a discriminator field in a header of each data packet to the secure computer network address to a table of valid discriminator fields.

10. (Currently Amended) The method according to claim 1, wherein the virtual private ~~computer~~ network includes the Internet.

11. (Original) The method according to claim 1, wherein the secure domain name has a top-level domain name that includes one of .scom, .snet, .sorg, .sedu, .smil or .sgov.

12. (Currently Amended) A computer-readable storage medium, comprising:
a storage area; and

computer-readable instructions for a method for accessing a secure computer network address, the method comprising steps of:

receiving a secure domain name;

sending a query message to a secure domain name service, the query message requesting from the domain name service a secure computer network address corresponding to the secure domain name;

receiving from the domain name service a response message containing the secure computer network address corresponding to the secure domain name; and

sending an access request message to the secure computer network address using a virtual private network communication link.

13. (Original) The computer-readable medium according to claim 12, wherein the step of receiving the secure domain name includes steps of:

receiving a command to establish the virtual private network communication link with a secure computer network address corresponding to a predetermined non-secure domain name; and

automatically generating a secure domain name corresponding to the non-secure domain name.

14. (Original) The computer-readable medium according to claim 13, wherein the step of receiving a command to establish the virtual private network communication link includes a step of selecting a predetermined icon displayed on a computer display.

15. (Original) The computer-readable medium according to claim 12, wherein the response message contains provisioning information for the virtual private network.

16. (Original) The computer-readable medium according to claim 15, wherein the virtual private network is based on inserting one or more data values into each data packet sent to the secure computer network address, the one or more data values varying according to a pseudo-random sequence.

17. (Original) The computer-readable medium according to claim 15, wherein the virtual private network is based on inserting into at least one data packet at least one data value representing a predetermined level of service associated with the virtual private network.

18. (Currently Amended) The computer-readable medium according to claim 15, wherein the virtual private network is based on a computer network address hopping regime that is used to pseudorandomly change computer network addresses in packets transmitted between ~~the~~ a first computer and ~~the~~ a second computer.

19. (Original) The computer-readable medium according to claim 15, wherein the virtual private network is based on comparing a value in each data packet transmitted to the secure computer network address to a moving window of valid values.

20. (Original) The computer-readable medium according to claim 15, wherein the virtual private network is based on a comparison of a discriminator field in a header of each data packet to the secure computer network address to a table of valid discriminator fields.

21. (Currently Amended) The computer-readable medium according to claim 12, wherein the ~~computer~~ virtual private network includes the Internet.

22. (Original) The computer-readable medium according to claim 12, wherein the secure domain name has a top-level domain name that includes one of .scom, .snet, .sorg, .sedu, .smil or .sgov.

23. (Currently Amended) The method of claim 1, wherein the access request message ~~containing~~ contains a request for information stored at the secure computer network address.

24. (Currently Amended) The computer readable medium of claim 12, wherein the access request message ~~containing~~ contains a request for information stored at the secure computer network address.

25. (New) The method of claim 1,

wherein receiving the secure domain name comprises receiving the secure domain name at a client computer from a user;

wherein sending the query message comprises sending the query message at the client computer;

wherein receiving the response message comprises receiving the response message at the client computer;

wherein sending the access request message comprises sending the access request message at the client computer.

26. (New) The method of claim 1, performed by a software module.

27. (New) The method of claim 1, performed by a client computer.

28. (New) The method of claim 2, wherein receiving the command comprises receiving the command at a client computer from a user.

29. (New) The computer-readable medium according to claim 12,

wherein receiving the secure domain name comprises receiving the secure domain name at a client computer from a user;

wherein sending the query message comprises sending the query message at the client computer;

wherein receiving the response message comprises receiving the response message at the client computer;

wherein sending the access request message comprises sending the access request message at the client computer.

30. (New) The computer-readable medium according to claim 12, wherein the method is performed by a software module.

31. (New) The computer-readable medium according to claim 12, wherein the method is performed by a client computer.

32. (New) The computer-readable medium according to claim 13, wherein receiving the command comprises receiving the command at a client computer from a user.

33. (New) A data processing apparatus, comprising:

a processor; and

memory storing computer executable instructions which, when executed by the processor, cause the apparatus to perform a method for accessing a secure computer network address, said method comprising steps of:

receiving a secure domain name;

sending a query message to a secure domain name service, the query message requesting from the secure domain name service a secure computer network address corresponding to the secure domain name;

receiving from the secure domain name service a response message containing the secure computer network address corresponding to the secure domain name; and

sending an access request message to the secure computer network address using a virtual private network communication link.

34. (New) The apparatus of claim 33, wherein the step of receiving the secure domain name includes steps of:

receiving a command to establish the virtual private network communication link with a secure computer network address corresponding to a predetermined non-secure domain name; and

automatically generating a secure domain name corresponding to the non-secure domain name.

35. (New) The apparatus of claim 33, wherein the response message contains provisioning information for the virtual private network.

36. (New) The apparatus of claim 35, wherein the virtual private network is based on inserting one or more data values into each data packet sent to the secure computer network address, the one or more data values varying according to a pseudo-random sequence.

37. (New) The apparatus of claim 35, wherein the virtual private network is based on inserting into at least one data packet at least one data value representing a predetermined level of service associated with the virtual private network.

38. (New) The apparatus of claim 35, wherein the virtual private network is based on a computer network address hopping regime that is used to pseudorandomly change computer network addresses in packets transmitted between a first computer and a second computer.

39. (New) The apparatus of claim 35, wherein the virtual private network is based on comparing a value in each data packet transmitted to the secure computer network address to a moving window of valid values.

40. (New) The apparatus of claim 35, wherein the virtual private network is based on a comparison of a discriminator field in a header of each data packet to the secure computer network address to a table of valid discriminator fields.

41. (New) The apparatus of claim 33, wherein the secure domain name has a top-level domain name that includes one of .scom, .snet, .sorg, .sedu, .smil or .sgov.

REMARKS/ARGUMENTS

The Office Action of May 19, 2006, has been carefully reviewed and these remarks are responsive thereto. Claims 1, 7, 10, 12, 18, 21, 23, and 24 have been amended, and new claims 25-41 have been added. Claims 1-41 are thus pending in this application. Reconsideration and allowance of the instant application are respectfully requested.

Rejections Under 35 U.S.C. § 112

Claims 1-3 and 12-14 stand rejected under 35 U.S.C. § 112, second paragraph, as being indefinite. Applicants have amended the claims to present the claims in a more preferred form, and respectfully request the rejection be withdrawn.

The Office Action's allegation that it is unclear where the steps of claim 1 are performed is misplaced. What is claimed is a method that, for example, receives "a secure domain name," and sends "a query message to a secure domain name service." Thus, while it is the method that receives a secure domain name and sends a query message, where the method is embodied is unclaimed, and thus immaterial with respect to claim 1. The method could be performed, for example, by a software module, in which case it would be the software module that receives a secure domain name and sends a query message. This is but one example, and claim 1 is not limited as such. In the same vein, where a secure domain name is received from in claim 1 or where a command is received from in dependent claims 2-3 are irrelevant insomuch as they pertain to entities outside the scope of the claims. Again, what is claimed is the receiving and processing of such information, not that the method can only receive such information from certain specified entities.

In this regard, applicants have added new claims 25-32, which recite examples of illustrative entities from where the steps of the claims may originate. Claim 25, for example, recites that the secure domain name is received at a client computer from a user, and that sending the query message, receiving the response message, and sending the access request message, is at the client computer.

Finally, amended claim 1 further recites, inter alia, "sending a query message to a secure domain name service, the query message requesting *from the secure domain name service* a

RECEIVED
CENTRAL FAX CENTER

AUG 17 2006

secure computer network address corresponding to the secure domain name," and "receiving *from the secure domain name service* a response message containing the secure computer network address corresponding to the secure domain name." (Emphasis added). Thus, the claim now explicitly states what was previously inherent in the claim, i.e., from where the query message is requested and from where a response message is received.

Thus, amended claim 1 is allowable for at least the reasons stated above. Dependent claims 2-11 are allowable for the same reasons as claim 1.

Amended claim 12 is the computer medium claim of claim 1 and is therefore allowable for at least the same reasons. Dependent claims 13-24 are likewise allowable for the same reasons as claim 12.

Finally, new claims 25-32 are presented to address the Office Action's rejections of their base claims and should therefore be allowable for at least the same reasons as their respective base claims.

New claims 33-41 are allowable at least for similar reasons as claims 1 and 12.

CONCLUSION

All rejections having been addressed, applicant respectfully submits that the instant application is in condition for allowance, and respectfully solicits prompt notification of the same. However, if for any reason the Examiner believes the application is not in condition for allowance or there are any questions, the Examiner is requested to contact the undersigned at (202) 824-3153.

Respectfully submitted,

BANNER & WITCOFF, LTD.

Dated this 17th day of August, 2006 By: /Ross Dannenberg/

Ross Dannenberg, Registration No. 49,024
1001 G Street, N.W.
Washington, D.C. 20001-4597
Tel: (202) 824-3000
Fax: (202) 824-3001

RAD/mmd

PAGE 11/11 * RCVD AT 8/17/2006 11:53:41 AM [Eastern Daylight Time] * SVR:USPTO-EFXRF-5/20 * DNIS:2738300 * CSID:12028243001 * DURATION (mm-ss):04-24

PATENT APPLICATION FEE DETERMINATION RECORD
Effective October 1, 2003

Application or Docket Number: 10702486

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS	24	
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	24 minus 20=	* 4
INDEPENDENT CLAIMS	2 minus 3 =	* 0
MULTIPLE DEPENDENT CLAIM PRESENT		☐

* If the difference in column 1 is less than zero, enter "0" in column 2

SMALL ENTITY TYPE ☐		OR	OTHER THAN SMALL ENTITY	
RATE	FEE		RATE	FEE
BASIC FEE	385.00	OR	BASIC FEE	770.00
XS 9=		OR	X$18=	72
X43=		OR	X86=	
+145=		OR	+290=	
TOTAL		OR	TOTAL	842

CLAIMS AS AMENDED - PART II

AMENDMENT A

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	* 24	Minus	** 26	=0
Independent	* 2	Minus	*** 2	=0
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X43=		OR	X86=	
+145=		OR	+290=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT B

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X43=		OR	X86=	
+145=		OR	+290=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT C

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X43=		OR	X86=	
+145=		OR	+290=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
***If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

PATENT APPLICATION FEE DETERMINATION RECORD
Effective October 1, 2003

Application or Docket Number: 10702486

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS	24	
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	24 minus 20=	* 4
INDEPENDENT CLAIMS	2 minus 3 =	* 0
MULTIPLE DEPENDENT CLAIM PRESENT		☐

* If the difference in column 1 is less than zero, enter "0" in column 2

SMALL ENTITY TYPE ☐		OR	OTHER THAN SMALL ENTITY	
RATE	FEE		RATE	FEE
BASIC FEE	385.00	OR	BASIC FEE	770.00
XS 9=		OR	X$18=	72
X43=		OR	X86=	
+145=		OR	+290=	
TOTAL		OR	TOTAL	842

CLAIMS AS AMENDED - PART II

AMENDMENT A	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	* 24	Minus	** 24	=0
Independent	* 2	Minus	*** 2	=0
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
XS 9=		OR	X$18=	
X43=		OR	X86=	
+145=		OR	+290=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT B	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X43=		OR	X86=	
+145=		OR	+290=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT C	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X43=		OR	X86=	
+145=		OR	+290=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
***If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

PTO/SB/08a (08-03)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Not for submission under 37 CFR 1.99)

Application Number	10702486
Filing Date	2003-11-07
First Named Inventor	Victor Larson
Art Unit	2153
Examiner Name	Lim, Krisna
Attorney Docket Number	000479.00112

U.S.PATENTS

Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	5870610	A	1999-02-09	Beyda et al.	

If you wish to add additional U.S. Patent citation information please click the Add button.

U.S.PATENT APPLICATION PUBLICATIONS

Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1					

If you wish to add additional U.S. Published Application citation information please click the Add button.

FOREIGN PATENT DOCUMENTS

Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2] i	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]
	1	0 838 930	EP	A	1998-04-29	Compaq Computer Corp.		☐
	2	0 814 589	EP	A	1997-12-29	AT&T Corp.		☐
	3	2 334 181	GB	A	1999-08-11	NEC Technologies; Globalmart Ltd.		☐

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	10702486
	Filing Date	2003-11-07
	First Named Inventor	Victor Larson
	Art Unit	2153
	Examiner Name	Lim, Krisna
	Attorney Docket Number	000479.00112

	4	9827783	WO	A	1998-06-25	Northern Telecom Limited; Antonio, G; Hui, Margare		☐
	5	2 317 792	GB	A	1998-04-01	Secure Computing Corporation		☐
	6							☐

If you wish to add additional Foreign Patent Document citation information please click the Add button

NON-PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T5
	1	Laurie Wells (LANCASTERBIBELMAIL MSN COM); "Subject: Security Icon" USENET NEWSGROUP, 19 October 1998, XP002200606	☐
	2	Davila J et al, "Implementation of Virtual Private Networks at the Transport Layer", Information Security, Second International Workshop, ISW '99. Proceedings (Lecture Springer-Verlag Berlin, Germany, [Online] 1999, pages 85-102, XP002399276, ISBN 3-540-66695-B, retrieved from the Internet: URL: http://www.springerlink.com/content/4uac0tb0heccma89/fulltext.pdf> (Abstract)	☐
	3		☐
	4		☐

If you wish to add additional non-patent literature document citation information please click the Add button

EXAMINER SIGNATURE

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	10702486
	Filing Date	2003-11-07
	First Named Inventor	Victor Larson
	Art Unit	2153
	Examiner Name	Lim, Krisna
	Attorney Docket Number	000479.00112

[1] See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. [2] Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). [3] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [4] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [5] Applicant is to place a check mark here if English language translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	10702486
	Filing Date	2003-11-07
	First Named Inventor	Victor Larson
	Art Unit	2153
	Examiner Name	Lim, Krisna
	Attorney Docket Number	000479.00112

CERTIFICATION STATEMENT

Please see 37 CFR 1.97 and 1.98 to make the appropriate selection(s):

☒ That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).

OR

☐ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

☐ See attached certification statement.

☐ Fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

☐ None

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	/Steve Chang/	Date (YYYY-MM-DD)	2006-11-09
Name/Print	Steve S. Chang	Registration Number	42,402

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1 hour to complete, including gathering, preparing and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these record s.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

WO9827783

Publication Title:

VIRTUAL PRIVATE NETWORK SERVICE PROVIDER FOR ASYNCHRONOUS TRANSFER MODE NETWORK

Abstract:

Abstract of WO9827783

A virtual private network service provider is used to transfer data over a data network to a final destination, with third-party billing. The method comprises the steps of: prompting the user at a data terminal to select a destination, password, and call type; sending a set-up message to the data network; selecting a virtual private network provider through the data network; the virtual pri 104d vate network provider giving an encryption key to the user, and then prompting the user for a password and a user identification; encrypting the password, and sending the user identification and the encrypted password to the virtual private network provider; the virtual private network provider decrypting the encrypted password, and verifying the password; the virtual private network provider providing an authorization code; and the data terminal transferring the data through the data network to the final destination, using the authorization code. Data supplied from the esp@cenet database - Worldwide

---

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :
H04Q 11/04, H04L 12/22

A1

(11) International Publication Number: **WO 98/27783**

(43) International Publication Date: 25 June 1998 (25.06.98)

(21) International Application Number: PCT/IB97/01563

(22) International Filing Date: 12 December 1997 (12.12.97)

(30) Priority Data:
08/769,649 19 December 1996 (19.12.96) US

(71) Applicant *(for all designated States except US)*: NORTHERN TELECOM LIMITED [CA/CA]; World Trade Center of Montreal, 8th floor, 380 St. Antoine Street West, Montreal, Quebec H2Y 3Y4 (CA).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: TELLO, Antonio, G. [US/US]; 114 Fountain Hills Drive, Garland, TX 75044 (US). HUI, Margaret [US/US]; 9920 Forest Lane #208, Dallas, TX 75243 (US). HOLMES, Kim [US/US]; 5409 Scenic Drive, Rowlett, TX 75088 (US).

(74) Agents: MCCOMBS, David et al.; Haynes and Boone, L.L.P., Suite 3100, 901 Main Street, Dallas, TX 75202–3789 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) Title: VIRTUAL PRIVATE NETWORK SERVICE PROVIDER FOR ASYNCHRONOUS TRANSFER MODE NETWORK

(57) Abstract

A virtual private network service provider is used to transfer data over a data network to a final destination, with third–party billing. The method comprises the steps of: prompting the user at a data terminal to select a destination, password, and call type; sending a set–up message to the data network; selecting a virtual private network provider through the data network; the virtual private network provider giving an encryption key to the user, and then prompting the user for a password and a user identification; encrypting the password, and sending the user identification and the encrypted password to the virtual private network provider; the virtual private network provider decrypting the encrypted password, and verifying the password; the virtual private network provider providing an authorization code; and the data terminal transferring the data through the data network to the final destination, using the authorization code.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

VIRTUAL PRIVATE NETWORK SERVICE PROVIDER FOR ASYNCHRONOUS TRANSFER MODE NETWORK

Technical Field

The invention relates generally to asynchronous transfer mode ("ATM") networks and virtual private networks ("VPN"), such as those offered by MCI and Sprint, and, more particularly, to a method of using a VPN to transfer data over a data network, with third-party billing.

Background of the Invention

Telephone service providers offer third-party billing. For example, local and long distance telephone companies offer calling cards for third party billing.

VPNs exist to provide the sense of a private network among a company's locations. The lines/trunks of a VPN are actually shared among several companies, to reduce costs, yet to each company the VPN appears to be that company's own private network. However, a user at a remote data terminal, such as a portable computer in a hotel room, can not immediately charge his company for the access time to a data net, such as the Internet. Instead, his access time is charged to his hotel room, and so he must pay the inflated rates that hotels charge for phone service.

What is needed is a VPN service provider that offers remote access for users belonging to a VPN, user authorizations to prevent delinquent access into the VPN, and convenient third-party billing.

Summary of the Invention

The present invention, accordingly, provides a system and method for using a VPN service provider to transfer data over a data network to a final destination, with third-party billing. The method comprises the steps of: prompting the user at a data terminal to select a destination, password, and call type; selecting a VPN through the data network; giving an encryption key to the user, and then prompting the user for a password and a user identification; verifying the password, and providing an authorization code to

the user; and allowing the user to transfer the data through the data network to the final destination, using the authorization code.

In another feature of the invention, the method further comprises negotiating for more bandwidth for the user, and including within the authorization code a grant of additional bandwidth.

In another feature of the invention, the method further comprises encrypting the user's password, and sending the user identification and the encrypted password to the VPN service provider.

In another feature of the invention, the method further comprises a step of sending a set-up message to the data network.

In another feature of the invention, the method further comprises a step of the VPN service provider decrypting the encrypted password.

A technical advantage achieved with the invention is that it shifts or defers costs from an end user to a bulk purchaser of data network services. Another technical advantage achieved with the invention is that it permits end users mobility while attaining a virtual appearance on a corporate intranet.

Brief Description of the Drawings

Fig. 1 is a system block diagram of a VPN service provider of the present invention.

Fig. 2 is a flow chart depicting the method of the present invention, as implemented by application software on a user terminal.

Fig. 3 is the initial screen display of the user interface of the application software.

Figs. 4A and 4B are call flow diagrams, illustrating the preferred sequence of steps of the method of the present invention.

Figs. 5A, 5B, 5C, 5D, 5E, and 5F comprise a flow chart depicting the method of the present invention, as implemented by switching control point software.

Description of the Preferred Embodiment

In Fig. 1, the VPN service provider system of the present invention is designated generally by a reference numeral 10. The VPN service provider system 10 includes a VPN 12. The VPN 12 may be a corporate, government, association, or other organization's telephone/data line `network. The VPN service provider system 10 also includes access lines 13 from the VPN 12 to a data network 14, such as the Internet, or an ATM network. The VPN service provider system 10 also includes access lines 16 from the data network 14 to a long distance phone company 18, such as AT&T, MCI, or Sprint. The VPN service provider system 10 also includes access lines 20 from the data network 14 to a called party 22, such as, for example, American Express reservations service. The VPN service provider system 10 also includes access lines 24 from the data network 14 to a remote user terminal 26, such as a portable computer in a hotel room. The user terminal 26 includes user application software 28, which provides the interface for the user to enter the number to be called, the user identification number, and the user's authorization code. The VPN service provider system 10 also includes VPN service provider software 30, located in a switching control point (SCP) device 32, which, in the preferred embodiment may be physically located anywhere. The SCP 32 connects to the data network 14 via access lines 36. One possible physical location for the SCP 32 is on the premises of a local phone company central switch building 34. However, even when located within the building 34, the SCP 32 connects to the local phone company switches via the data network 14. The local phone company switches connect to the data network 14 via access lines 38.

In an alternate embodiment, the VPN service provider software 30 and the SCP device 32 may be located on the premises of an independent provider of local phone service, or on the premises of an independent VPN service provider.

Referring now to Fig. 2, the application software 28 begins the data transfer process in step 50. In step 52, the user is presented with a screen display.

Referring now to Fig. 3, a screen display 100 displays the following information requests: whether the call is a direct call 102 or a VPN call 104, the number the user desires to call 106, the VPN user ID 108, and the user password 110. The user is also presented with the option to make the call 112, or to quit 114.

Referring back to Fig. 2, in step 54 the user terminal sends to the SCP 32 the information captured through the graphical user interface ("GUI") in step 52 within a user network interface ("UNI") setup message. In step 56 the user terminal 26 waits for a connect message from the SCP 32. In step 58 the user terminal 26 determines if a connection was made. If no connection was made, then in step 60 the user application software 28 displays an error message to the user, and returns to step 50 to begin again the data transfer process.

If a connection was made, then in step 62 the user terminal 26 sends the VPN user ID to the SCP 32. In step 64 the user terminal 26 waits for an encryption key from the SCP 32. In step 66, having received the encryption key from the SCP 32, the user application software 28 encrypts the user's password, and sends it to the SCP 32. In step 68 the user terminal 26 waits for authentication of the user. In step 70 the user application software 28 determines if the SCP 32 authorizes the user to make the call.

If the user is not authorized, then in step 72 the user terminal 26 displays an error message, terminates the connection, blanks the screen display 100, and returns to step 50 to begin again the data transfer process. If the user is authorized, then in step 74 the VPN service provider software 30 sets up the billing, and authorizes it. In step 76 the user terminal 26 sends a "release", meaning to terminate or disconnect the connection, to the SCP 32. In step 78 the user terminal 26 sends a setup message to the number listed by

the user as the "number to call", that is, to the final destination. In step 80 the user terminal 26 waits for a connection. In step 82 the user terminal 26 determines if a connection was made.

If a connection to the final destination was not made, then the user application software 28 returns to step 72, in which step the user terminal 26 displays an error message, terminates the connection, blanks the screen display 100, and returns to step 50 to begin again the data transfer process. If a connection to the final destination was made, then in step 84 the user terminal 26 exchanges user data, services, and/or value added or user specific applications with the computer at the address, that is, the telephone number, of the final destination. In step 86 the user selects the option presented to him to release, or terminate, the call. In step 88 the user terminal 26 sends a release message to the final destination. In step 90 the data network 14 sends billing information to the SCP 32. In step 92 the application software 28 ends the data transfer process.

Fig. 4A and Fig. 4B are call flow diagrams, showing the sequence of messages in the method of the preferred embodiment. These diagrams present the same method as the flow chart of Fig. 2. The horizontal arrows represent the messages sent and received. The vertical lines represent the various devices involved in sending and receiving the messages. For example, the top left arrow in Fig. 4A represents a message sent from the user terminal 26, labeled "Macintosh" in Fig.4A, to an interface with a public network. The user terminal 26 can be any brand of a work station computer, a desktop computer, a laptop computer, or even a notebook computer. The interface could be any interface, but in the example of Fig. 4A and Fig. 4B, the interface is imagined to be at a hotel, where a business traveler is using the method of the present invention. Thus, the interface is labeled "Hotel ATM Interface", which is not shown in Fig. 1. The vertical line labeled "Public ATM Network" is the same as the data network 14 in Fig. 1. The vertical line labeled "Moe's VPN Service" represents the VPN service provider software 30

within the SCP 32. The vertical line labeled "Travel ATM Interface" is not shown in Fig. 1, but is located between the called party 22 and the data network 14. The vertical line labeled "Travel Service" is one example of the called party 22 shown in Fig. 1. In the example of Fig. 4A and Fig. 4B, the business traveler is imagined to be using the method of the present invention to contact a travel service to make reservations for his next airline flight. In Figs. 4A and 4B the designation "Ack" represents "acknowledge", and the designation "Cmp" represents "complete".

Referring now to Fig. 5, the VPN service provider software 30 begins the data transfer process in step 300 by waiting for an event. The event it waits for is a setup message on a signaling port of the SCP 32, to be received from the user terminal 26. In step 302, having monitored the signaling ports, and the SCP 32 having received a setup message, the VPN service provider software 30 assigns a call condense block ("CCB") to the setup message, based on a call reference number. The CCB is a software data structure for tracking resources associated with the call. The call reference number is a number, internal to the SCP, for tracking calls. In step 304 the VPN service provider software 30 compiles the connect message. In step 306 the VPN service provider software 30 sends a connect message to the calling address, that is, the hotel room from which the user is calling. In step 308 the VPN service provider software 30 condenses, that is, it remains in a wait state for that call.

Referring now to Fig. 5B, in step 310 the VPN service provider software 30 waits for an event by monitoring the signaling ports of the SCP 32. After the SCP 32 receives a connect acknowledge message from the user terminal 26, then in step 312 the VPN service provider software 30 accesses the CCB, based on the call reference number. In step 314 the VPN service provider software 30 condenses.

Referring now to Fig. 5C, in step 316 the VPN service provider software 30 waits for dialog on a data port of the SCP 32. After the SCP 32 receives a

VPN ID on a data port, the VPN service provider software 30 verifies the VPN ID in step 318. In step 320 the VPN service provider software 30 determines if the VPN ID is valid. If the VPN ID is not valid, then in step 322 the SCP 32 sends a reject message over an assigned switch virtual circuit ("SVC"). The SVC is a channel over the data network 14. In step 324 the VPN service provider software 30 waits for dialog. In step 326, because the VPN ID is valid, the VPN service provider software 30 assigns an encryption key to the user terminal 26, in step 328 sends the encryption key over the assigned SVC to the user terminal 26, and in step 330 waits for dialog.

Referring now to Fig. 5D, in step 332 the VPN service provider software 30 waits for dialog. When the SCP 32 receives the encrypted password from the user terminal 26 at a data port, then in step 334 the VPN service provider software 30 verifies the password, and determines in step 336 if the password is valid. If the password is not valid, then in step 338 the SCP 32 sends a reject message over the assigned SVC to the user terminal, and in step 340 waits for dialog. If the password is valid, then in step 342 the VPN service provider software 30 assigns an authorization token to the user terminal 26, in step 344 sends the token over an assigned SVC to the user terminal 26, and in step 346 waits for dialog.

Referring now to Fig. 5E, in step 348 the VPN service provider software 30 waits for an event. When the VPN service provider software 30 senses that the SCP 32 has received on a signaling port a release message from the user terminal 26, then in step 350 the VPN service provider software 30 accesses the CCB, based on the call reference number of the user terminal 26, in step 352 compiles a release complete message, in step 354 sends a release complete message to the user terminal 26, and in step 356 condenses.

Referring now to Fig. 5F, in step 358 the VPN service provider software 30 waits for an event. When the VPN service provider software 30 senses that the SCP 32 has received on a signaling port a third-party billing setup message from the user terminal 26, then in step 360 the VPN service provider

software 30 verifies the token just received from the user terminal 26, to determine, in step 362, if it is the same token that the VPN service provider software 30 sent to the user terminal 26 in step 344. If the token is not valid, then in step 364 the SCP 32 sends a release message to the terminal 26, and in step 366 condenses. If the token is valid, then in step 368 the SCP 32 sends a modified third-party billing setup message to the data network 14, and in step 370 condenses.

Although an illustrative embodiment of the invention has been shown and described, other modifications, changes, and substitutions are intended in the foregoing disclosure. Accordingly, it is appropriate that the appended claims be construed broadly and in a manner consistent with the scope of the invention.

WHAT IS CLAIMED IS:

1. A computerized method of a virtual private network service provider with third party billing, using a virtual private network to transfer data over a data network to a final destination, the method comprising the steps of:

a. prompting the user at a data terminal to select a destination, password, and call type;

b. selecting a virtual private network through the data network;

c. giving an encryption key to the user, and then prompting the user for a password and a user identification;

d. verifying the password, and providing an authorization code to the user; and

e. allowing the user to transfer the data through the data network to the final destination, using the authorization code.

2. The method of claim 1, wherein step (d) further comprises negotiating for more bandwidth for the user, and including within the authorization code a grant of additional bandwidth.

3. The method of claim 2, wherein step (c) further comprises encrypting the user's password, and sending the user identification and the encrypted password to the virtual private network service provider.

4. The method of claim 3, further comprising, after step (a), the step of sending a set-up message to the data network.

5. The method of claim 4, further comprising, after step (c), the step of the virtual private network service provider decrypting the encrypted password.

6. An apparatus for providing a datalink connection from a user terminal to a data network and to a virtual private network, with third party billing, comprising:

a. an interface between the user terminal and the data network;

b. a switching control point device connected to the data network, the switching control point device connected to a computer; and

c. a computer-readable medium encoded with a method of using the virtual private network and the data network, with third party billing, the computer-readable medium accessible by the computer.

7. The apparatus of claim 6, wherein the method comprises negotiating for more bandwidth for the user, and including within an authorization code a grant of additional bandwidth.

8. The apparatus of claim 7, wherein the method further comprises encrypting a user's password, and temporarily storing the user identification and the encrypted password.

9. The apparatus of claim 8, wherein the method further comprises sending a set-up message to the data network.

10. The apparatus of claim 9, wherein the method further comprises decrypting the encrypted password.

11. A computer-readable medium encoded with a method of using a virtual private network, with third party billing, the method comprising the steps of:

a. prompting the user at a data terminal to select a destination, password, and call type;

b. selecting a virtual private network through the data network;

c. giving an encryption key to the user, and then prompting the user for a password and a user identification;

d. verifying the password, and providing an authorization code to the user; and

e. allowing the user to transfer the data through the data network to the final destination, using the authorization code.

12. The computer-readable medium of claim 11 wherein step (d) further comprises negotiating for more bandwidth for the user, and including within the authorization code a grant of additional bandwidth.

13. The computer-readable medium of claim 12 wherein step (c) further comprises encrypting the user's password, and sending the user identification and the encrypted password to the virtual private network service provider.

14. The computer-readable medium of claim 13 further comprising, after step (a), the step of sending a set-up message to the data network.

15. The computer-readable medium of claim 14 further comprising, after step (c), the step of the virtual private network service provider decrypting the encrypted password.

16. An apparatus for providing a datalink connection from a user terminal to a data network and to a virtual private network, with third party billing, comprising:

a. means for prompting a user at the data terminal to select a destination, password, and call type;

b. means for selecting the virtual private network through the data network;

c. means for giving an encryption key to the user, and then prompting the user for a password and a user identification;

d. means for verifying the password, and providing an authorization code to the user; and

e. means for allowing the user to transfer data through the data network to a final destination, using the authorization code.

17. The apparatus of claim 16, further comprising means for negotiating for more bandwidth for the user, and including within the authorization code a grant of additional bandwidth.

18. The apparatus of claim 17, further comprising means for encrypting the user's password, and sending the user identification and the encrypted password to the virtual private network service provider.

19. The apparatus of claim 18, further comprising means for sending a set-up message to the data network.

20. The apparatus of claim 19, further comprising means for decrypting the encrypted password.

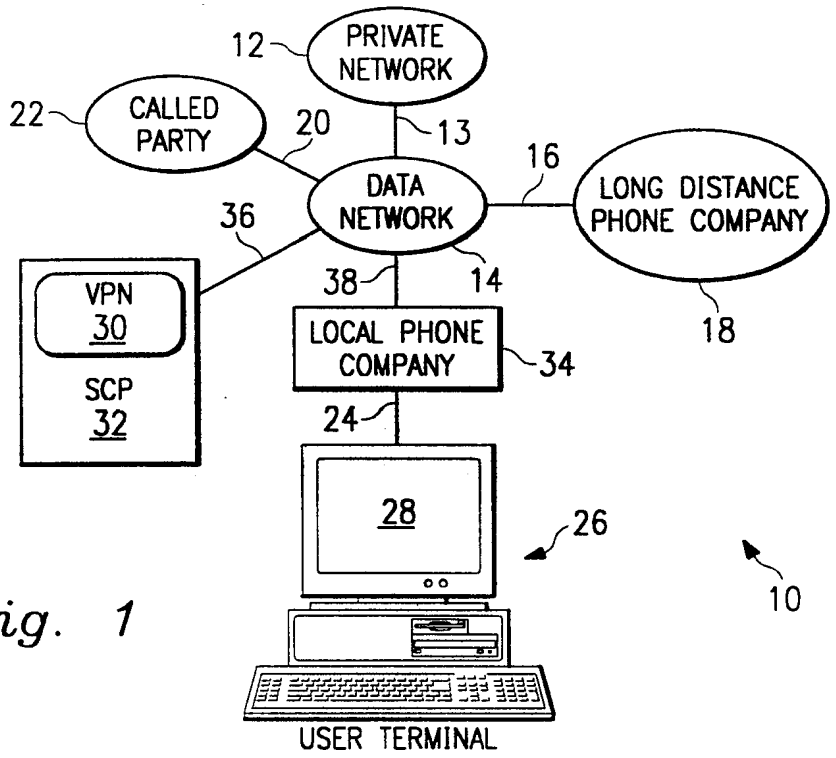

12
PRIVATE NETWORK
22
CALLED PARTY
20
13
16
DATA NETWORK
LONG DISTANCE PHONE COMPANY
36
38
14
18
VPN 30
SCP 32
LOCAL PHONE COMPANY
34
24
28
26
10
USER TERMINAL

Fig. 1

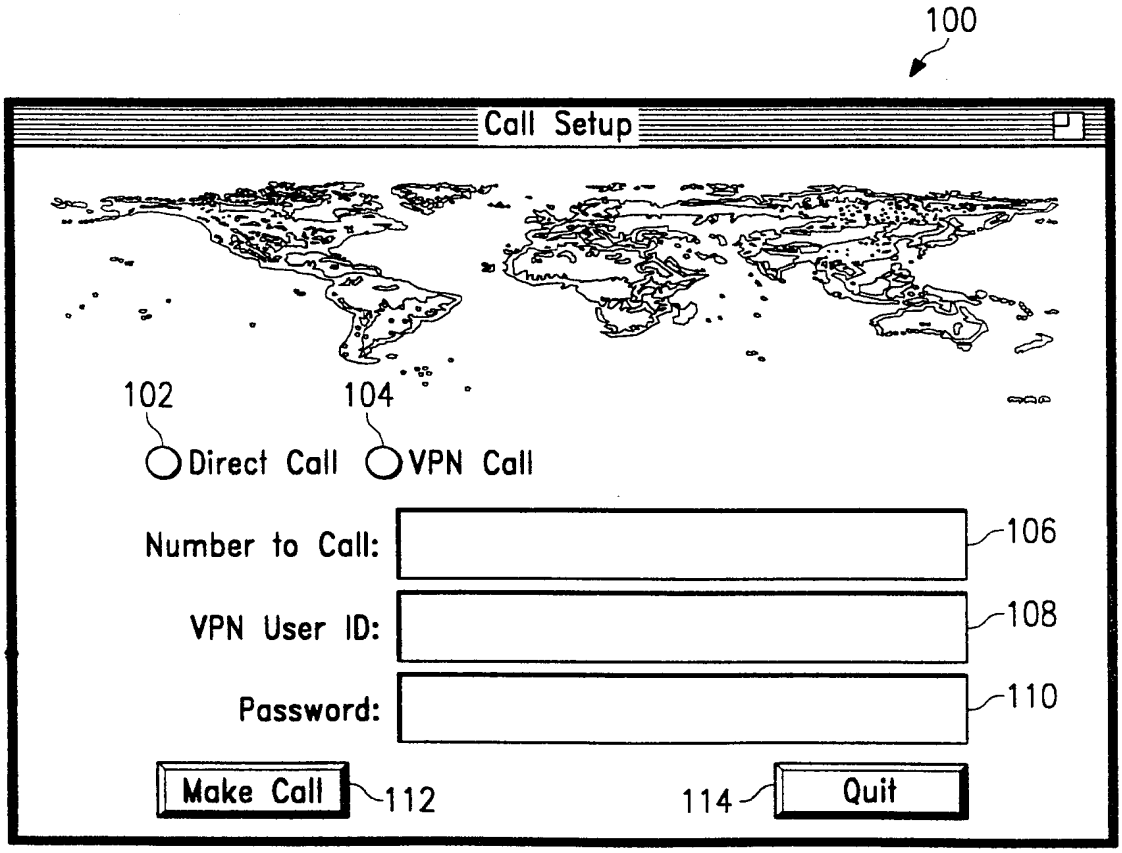

100
Call Setup
102
104
Direct Call
VPN Call
Number to Call:
106
VPN User ID:
108
Password:
110
Make Call
112
114
Quit

Fig. 3

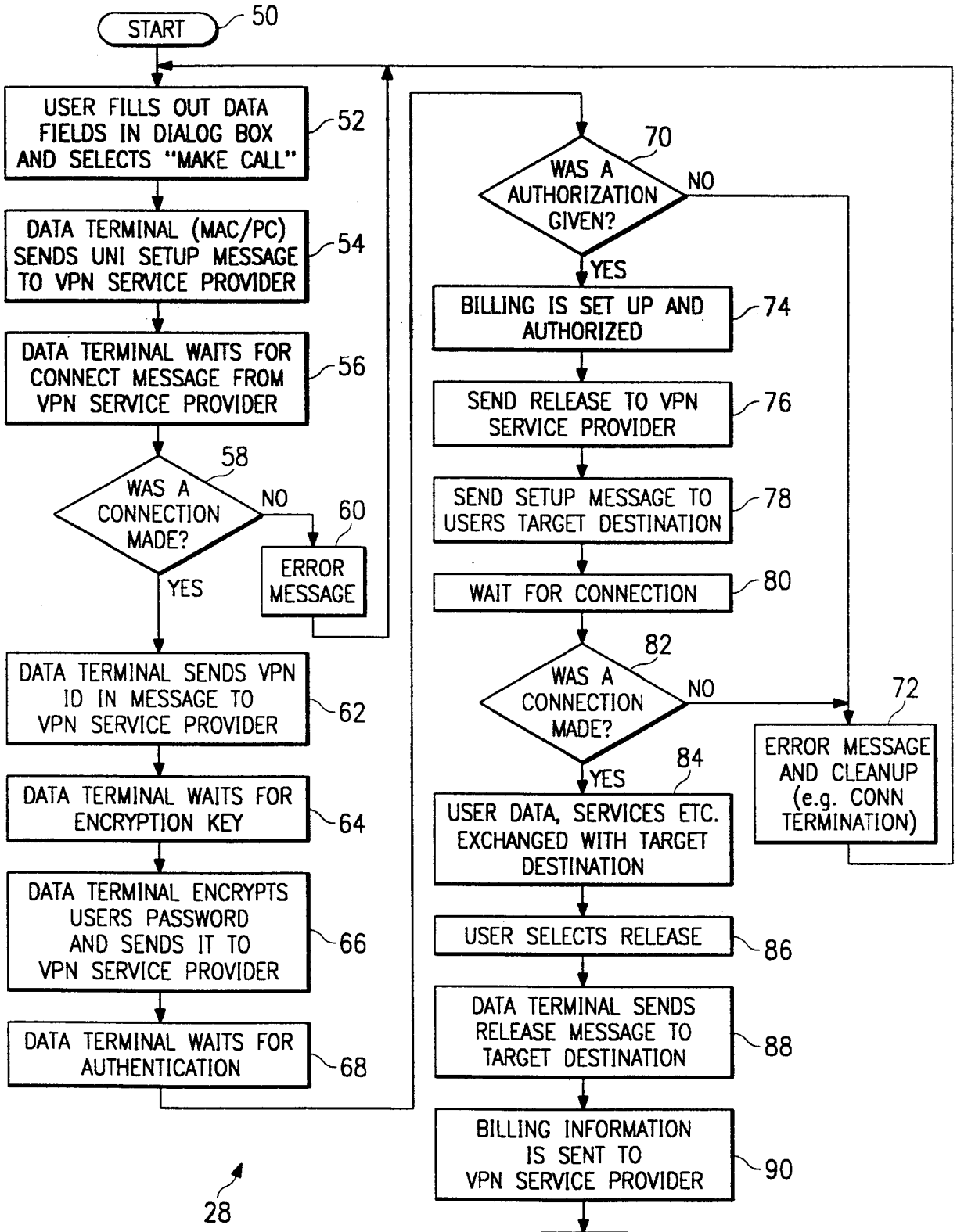

START
50
USER FILLS OUT DATA FIELDS IN DIALOG BOX AND SELECTS "MAKE CALL"
52
DATA TERMINAL (MAC/PC) SENDS UNI SETUP MESSAGE TO VPN SERVICE PROVIDER
54
DATA TERMINAL WAITS FOR CONNECT MESSAGE FROM VPN SERVICE PROVIDER
56
WAS A CONNECTION MADE?
58
NO
60
ERROR MESSAGE
YES
DATA TERMINAL SENDS VPN ID IN MESSAGE TO VPN SERVICE PROVIDER
62
DATA TERMINAL WAITS FOR ENCRYPTION KEY
64
DATA TERMINAL ENCRYPTS USERS PASSWORD AND SENDS IT TO VPN SERVICE PROVIDER
66
DATA TERMINAL WAITS FOR AUTHENTICATION
68
28
WAS A AUTHORIZATION GIVEN?
70
NO
YES
BILLING IS SET UP AND AUTHORIZED
74
SEND RELEASE TO VPN SERVICE PROVIDER
76
SEND SETUP MESSAGE TO USERS TARGET DESTINATION
78
WAIT FOR CONNECTION
80
WAS A CONNECTION MADE?
82
NO
72
ERROR MESSAGE AND CLEANUP (e.g. CONN TERMINATION)
YES
84
USER DATA, SERVICES ETC. EXCHANGED WITH TARGET DESTINATION
USER SELECTS RELEASE
86
DATA TERMINAL SENDS RELEASE MESSAGE TO TARGET DESTINATION
88
BILLING INFORMATION IS SENT TO VPN SERVICE PROVIDER
90
END
92

Fig. 2

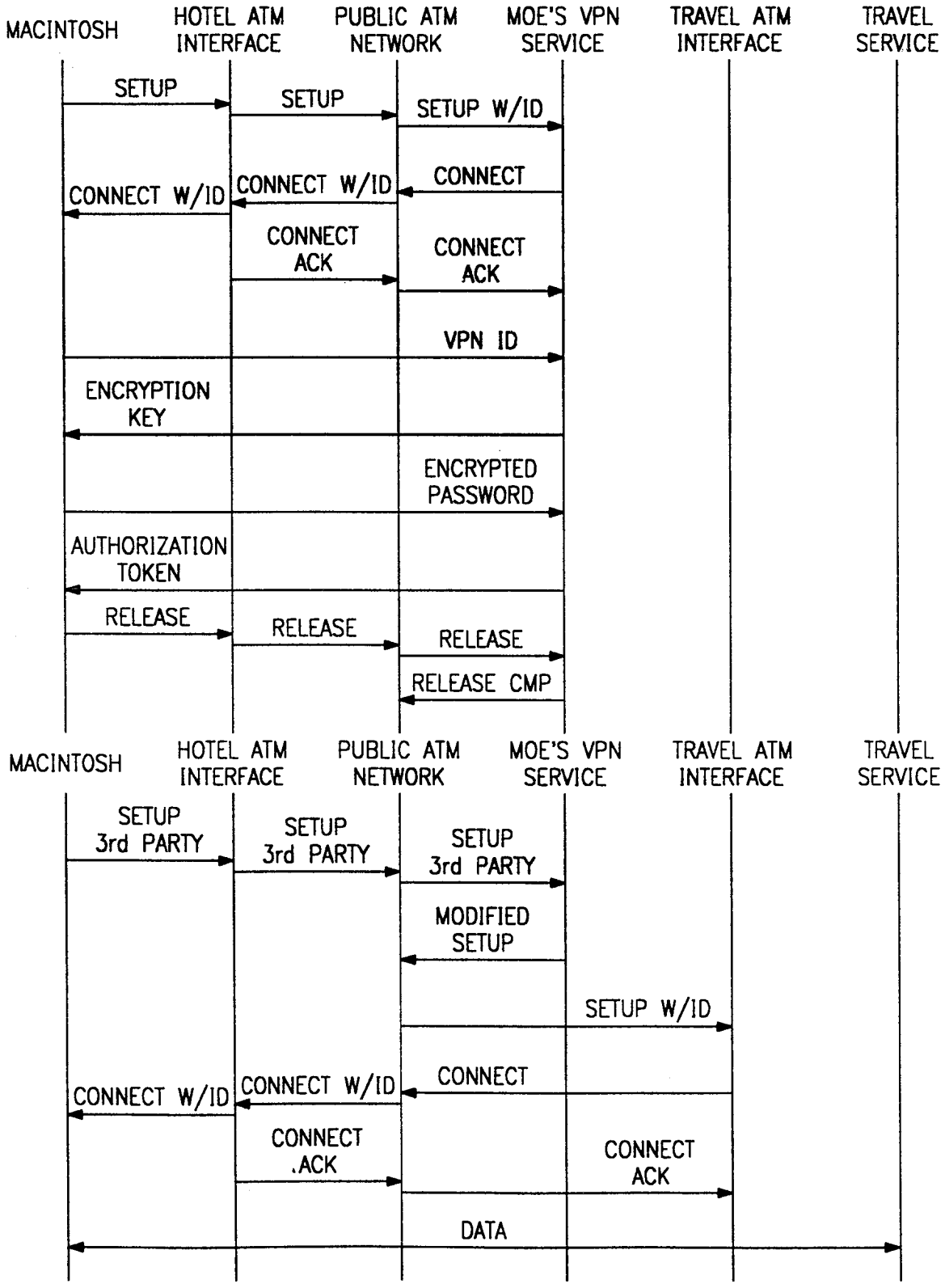

MACINTOSH
HOTEL ATM INTERFACE
PUBLIC ATM NETWORK
MOE'S VPN SERVICE
TRAVEL ATM INTERFACE
TRAVEL SERVICE
SETUP
SETUP
SETUP W/ID
CONNECT W/ID
CONNECT W/ID
CONNECT
CONNECT ACK
CONNECT ACK
VPN ID
ENCRYPTION KEY
ENCRYPTED PASSWORD
AUTHORIZATION TOKEN
RELEASE
RELEASE
RELEASE
RELEASE CMP
MACINTOSH
HOTEL ATM INTERFACE
PUBLIC ATM NETWORK
MOE'S VPN SERVICE
TRAVEL ATM INTERFACE
TRAVEL SERVICE
SETUP 3rd PARTY
SETUP 3rd PARTY
SETUP 3rd PARTY
MODIFIED SETUP
SETUP W/ID
CONNECT W/ID
CONNECT W/ID
CONNECT
CONNECT .ACK
CONNECT ACK
DATA

Fig. 4A

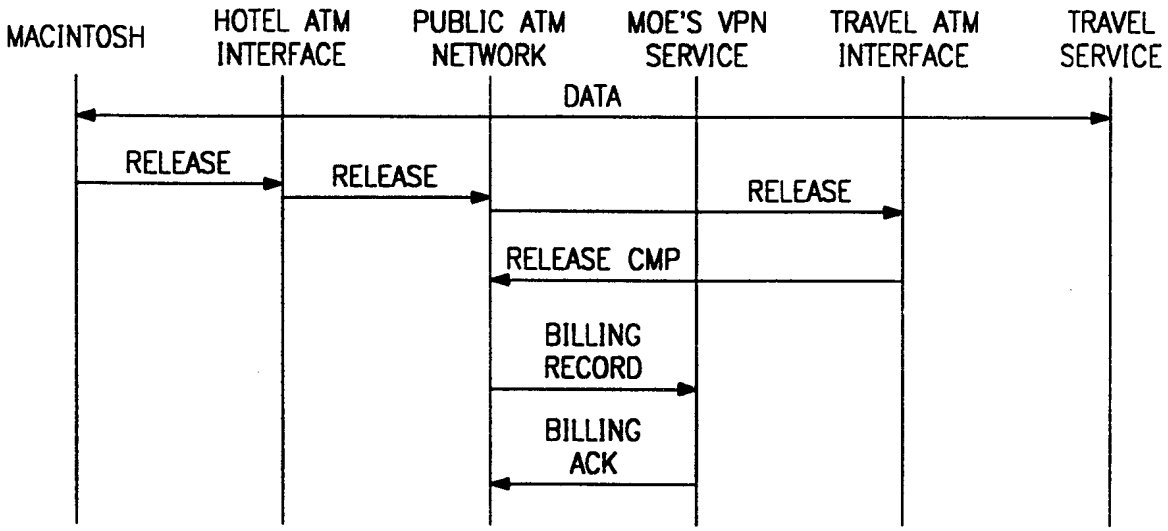


Fig. 4B

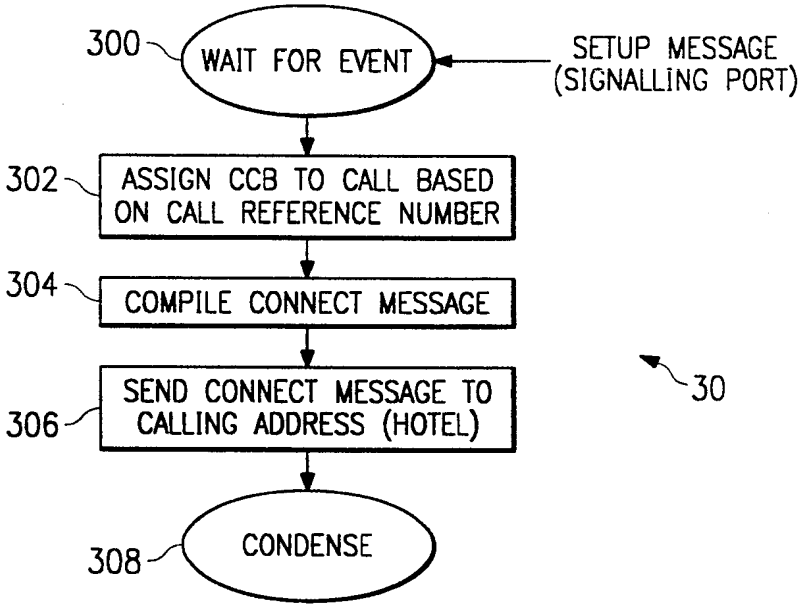


Fig. 5A

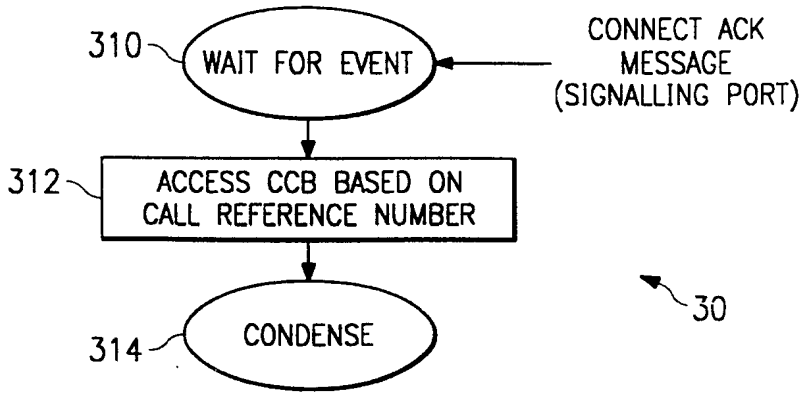


Fig. 5B

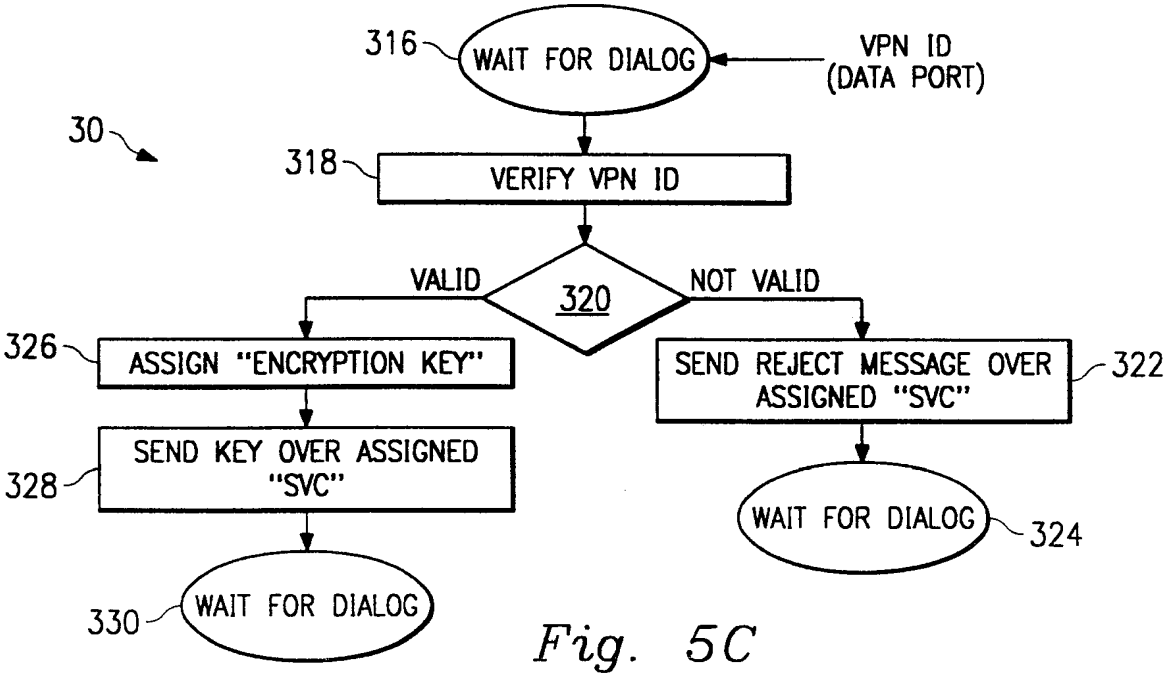

316
WAIT FOR DIALOG
VPN ID
(DATA PORT)
30
318
VERIFY VPN ID
VALID
320
NOT VALID
326
ASSIGN "ENCRYPTION KEY"
SEND REJECT MESSAGE OVER
ASSIGNED "SVC"
322
328
SEND KEY OVER ASSIGNED
"SVC"
WAIT FOR DIALOG
324
330
WAIT FOR DIALOG

Fig. 5C

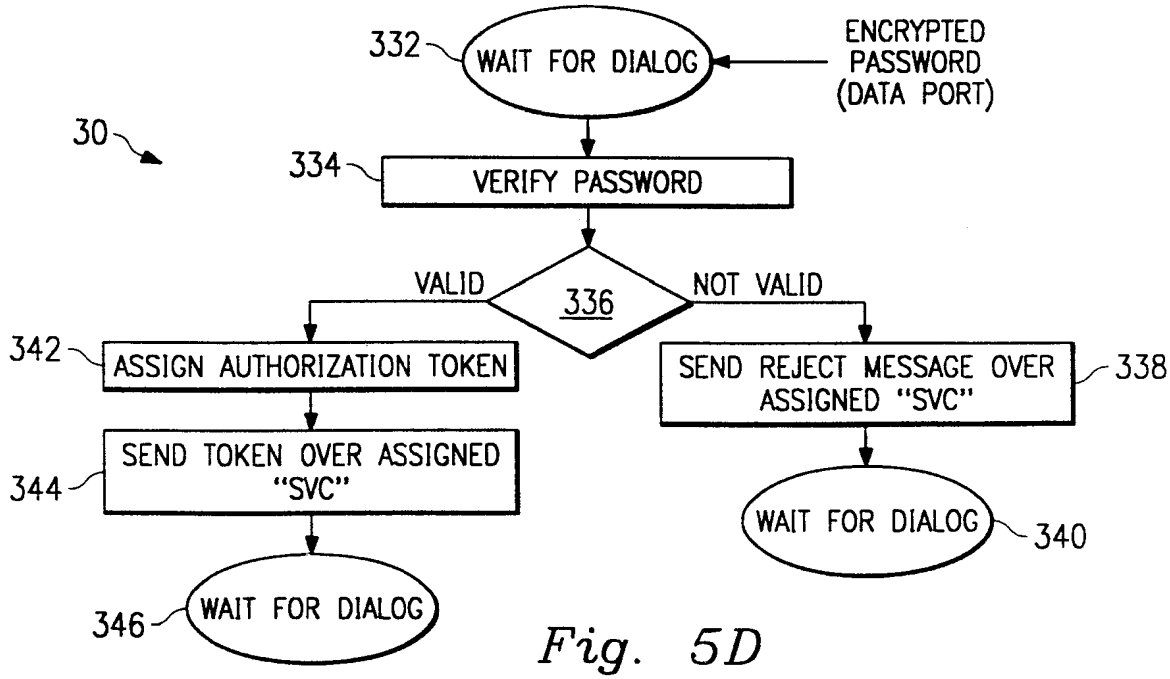

332
WAIT FOR DIALOG
ENCRYPTED
PASSWORD
(DATA PORT)
30
334
VERIFY PASSWORD
VALID
336
NOT VALID
342
ASSIGN AUTHORIZATION TOKEN
SEND REJECT MESSAGE OVER
ASSIGNED "SVC"
338
344
SEND TOKEN OVER ASSIGNED
"SVC"
WAIT FOR DIALOG
340
346
WAIT FOR DIALOG

Fig. 5D

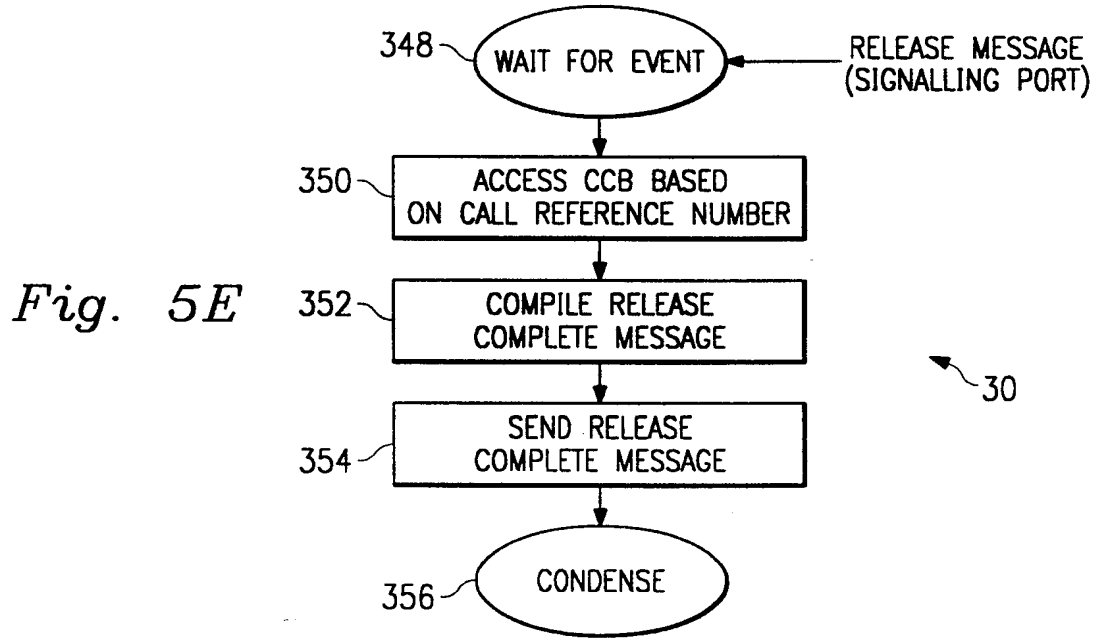

348
WAIT FOR EVENT
RELEASE MESSAGE (SIGNALLING PORT)
350
ACCESS CCB BASED ON CALL REFERENCE NUMBER
352
COMPILE RELEASE COMPLETE MESSAGE
354
SEND RELEASE COMPLETE MESSAGE
356
CONDENSE
30

Fig. 5E

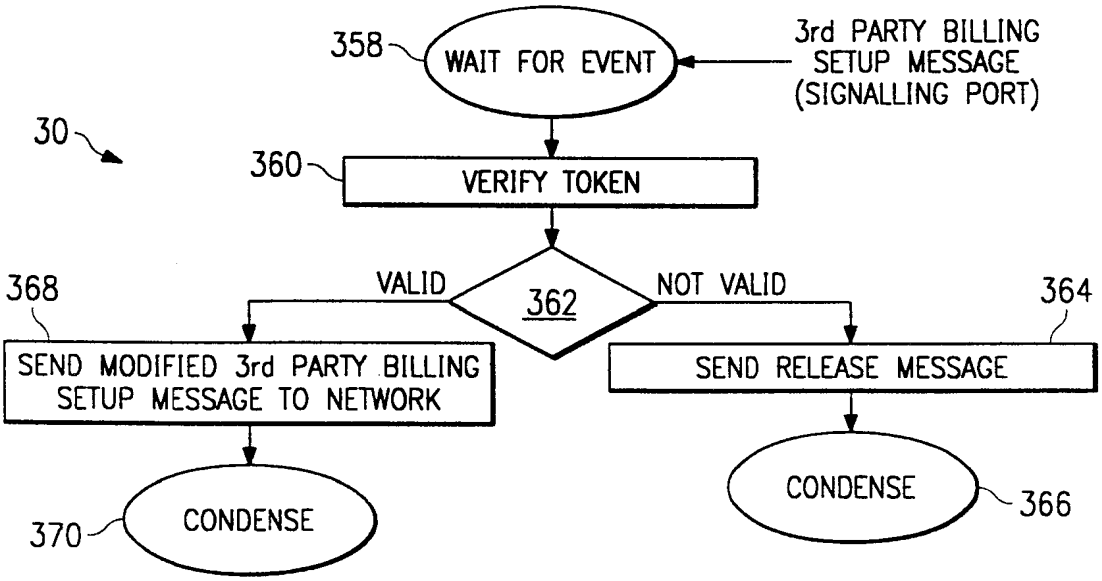

358
WAIT FOR EVENT
3rd PARTY BILLING SETUP MESSAGE (SIGNALLING PORT)
30
360
VERIFY TOKEN
362
VALID
NOT VALID
368
364
SEND MODIFIED 3rd PARTY BILLING SETUP MESSAGE TO NETWORK
SEND RELEASE MESSAGE
CONDENSE
370
CONDENSE
366

Fig. 5F

INTERNATIONAL SEARCH REPORT

Inte onal Application No
PCT/IB 97/01563

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 H04Q11/04 H04L12/22

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 H04Q H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	MUN CHOON CHAN ET AL: "AN ARCHITECTURE FOR BROADBAND VIRTUAL NETWORKS UNDER CUSTOMER CONTROL" NOMS '96 IEEE NETWORK OPERATIONS AND MANAGEMENT SYMPOSIUM , vol. 1, 15 April 1996, KYOTO, JP, pages 135-144, XP000641086 see abstract ---	1-20
A	BIC V: "VOICE PERIPHERALS IN THE INTELLIGENT NETWORK" TELECOMMUNICATIONS, vol. 28, no. 6, June 1994, page 29/30, 32, 34 XP000600293 see the whole document --- -/--	1-20

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

1

Date of the actual completion of the international search	Date of mailing of the international search report
19 March 1998	02/04/1998

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Staessen, B

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Inter nal Application No
PCT/IB 97/01563

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 729 256 A (NEDERLAND PTT) 28 August 1996 see abstract figures of pages 136 and 140	1-20
A	CROCETTI P ET AL: "ATM VIRTUAL PRIVATE NETWORKS: ALTERNATIVES AND PERFORMANCES COMPARISONS" SUPERCOMM/ICC '94, 1 May 1994, NEW ORLEANS, US, pages 608-612, XP000438985 see abstract	1-20

1

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

information on patent family members

Inte onal Application No
PCT/IB 97/01563

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0729256 A	28-08-96	NL 9500339 A	01-10-96

Form PCT/ISA/210 (patent family annex) (July 1992)

GB2334181

Publication Title:

Over-the-air re-programming of radio transceivers

Abstract:

Abstract of GB2334181

A method of downloading reprogramming data from a network for installation in a mobile station makes use of a dedicated small bandwidth pilot channel. The mobile station obtains from the base station the radio access parameters of a second channel. The second channel is a large bandwidth (bootstrap) channel suitable for fast transfer of data. The bootstrap channel is logically mapped onto a local transmission mode such as DECT or GSM by the mobile station and re-programming data may be downloaded from the base station via the bootstrap channel.

(12) UK Patent Application (19) GB (11) 2 334 181 (13) A

(43) Date of A Publication 11.08.1999

(21) Application No 9802545.5

(22) Date of Filing 06.02.1998

(71) Applicant(s)
NEC Technologies (UK) Ltd
(Incorporated in the United Kingdom)
Castle Farm Campus, Priorslee, TELFORD, Shropshire,
TF2 9SA, United Kingdom

(72) Inventor(s)
Charles Marie Herve Noblet

(74) Agent and/or Address for Service
J W White
NEC Technologies (UK) Ltd, Level 3, The Imperium,
Imperial Way, READING, Berks, RG2 0TD,
United Kingdom

(51) INT CL[6]
H04Q 7/32

(52) UK CL (Edition Q)
H4L LDSC

(56) Documents Cited
US 5613204 A US 5109403 A

(58) Field of Search
UK CL (Edition P) H4L LDSC LDSU LECC LECX
INT CL[6] H04Q 7/32 7/38
Online: WPI

(54) Abstract Title
Over-the-air re-programming of radio transceivers

(57) A method of downloading reprogramming data from a network for installation in a mobile station makes use of a dedicated small bandwidth pilot channel. The mobile station obtains from the base station the radio access parameters of a second channel. The second channel is a large bandwidth (bootstrap) channel suitable for fast transfer of data. The bootstrap channel is logically mapped onto a local transmission mode such as DECT or GSM by the mobile station and re-programming data may be downloaded from the base station via the bootstrap channel.

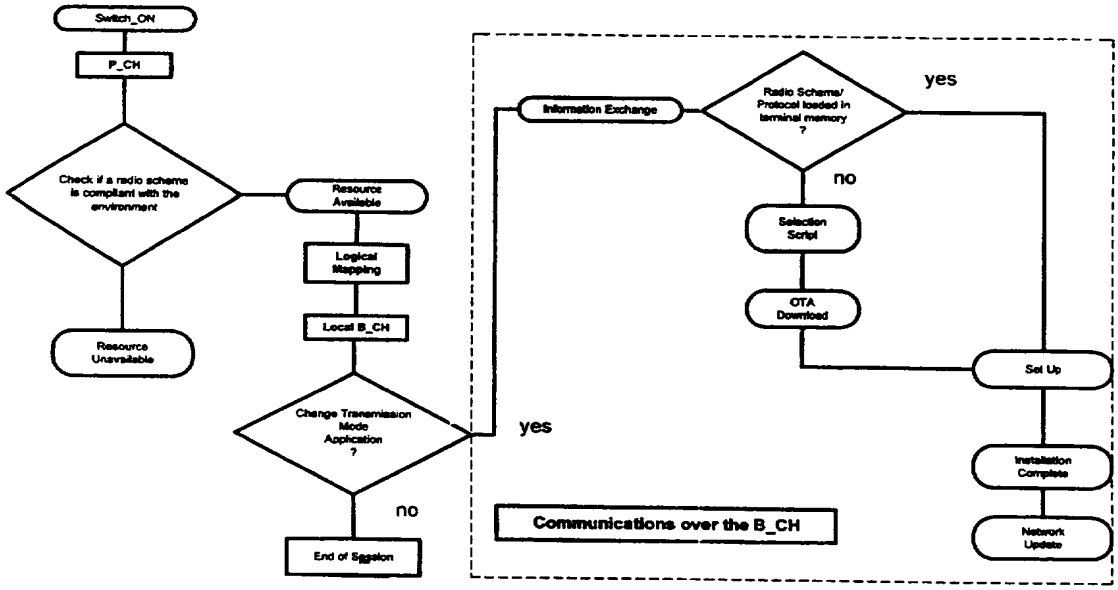


Figure 2

GB 2 334 181 A

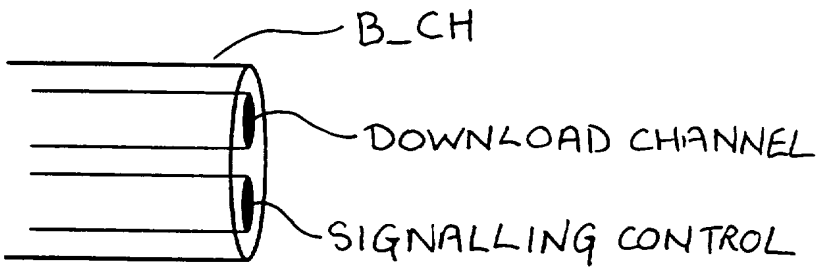


Figure: 1

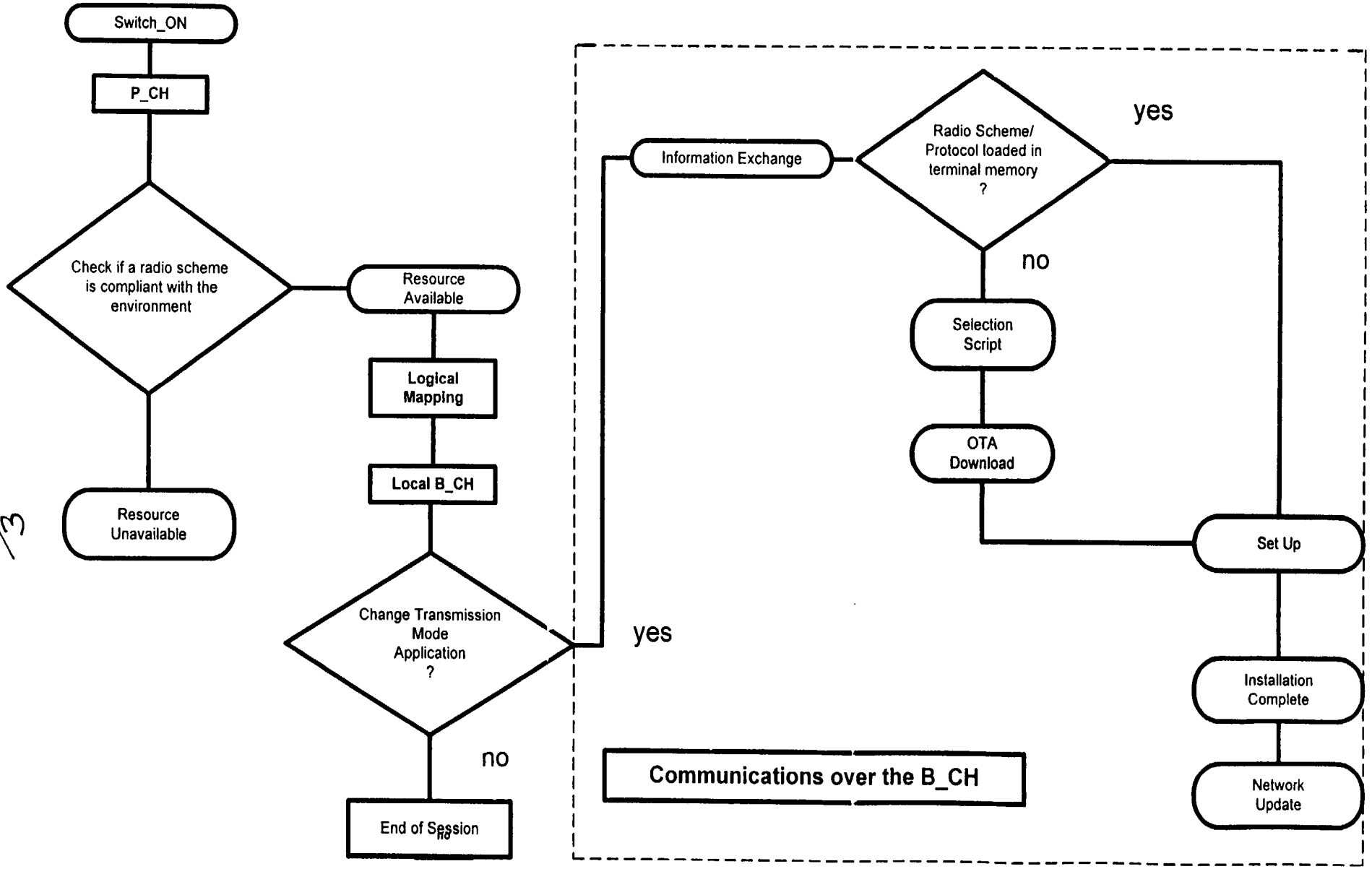
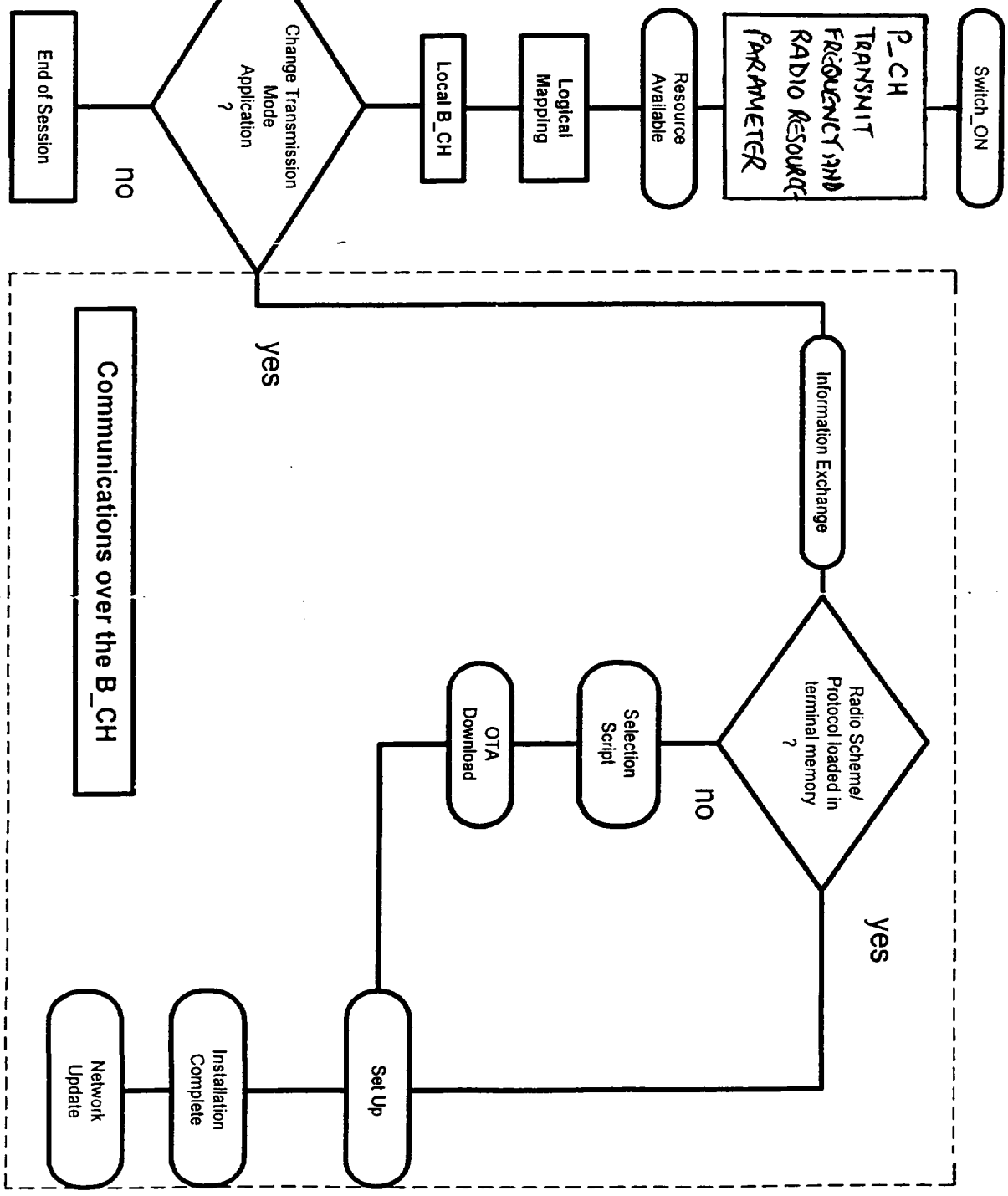


Figure 2

Figure 3


Switch_ON
P_CH TRANSMIT FREQUENCY AND RADIO RESOURCE PARAMETER
Resource Available
Logical Mapping
Local B_CH
Change Transmission Mode Application ?
no
End of Session
yes
Information Exchange
Radio Scheme/ Protocol loaded in terminal memory ?
no
Selection Script
OTA Download
yes
Set Up
Installation Complete
Network Update
Communications over the B_CH

2334181

Over-the-air re-programming of radio transceivers

This invention relates to radio transmitter/receivers and in particular it relates to a method of re-programming radio transmitter/receivers over-the-air.

A radio transmitter/receiver (transceiver) such as a radiotelephone is designed for operation with particular types of networks such as GSM 900 or DCS 1800. Intended use of the radiotelephone with a particular network(s) in a restricted geographical area, however, requires that the telephone be configured so as properly to communicate with the particular network (s). The user of a radiotelephone will usually have a telephone which has been configured for communication with a so called "home network". The home network is the local network usually most used by the subscriber.

The area within which a user of e.g. a GSM radiotelephone may operate, however, is considerable and is not limited to the home network but may be used on many other networks throughout the world. Use of a handset outside the home network is known as "roaming".

When the radiotelephone is to be used in roaming it is often necessary for it to have a configuration different to that for use with the home network. It is possible for re-configuration of radio transmitter/receivers to be effected by means of signals received across the air interface.

It is also convenient for the radio to be re-configurable over the air interface so as to support different types of communication and user applications e.g. addition of address book manager, whether or not it is located in the home network.

Over the air re-programming of radio receivers is well known in the art and reference may be made to US patent 5 381 138 for example. The capability to obtain programming data from a network is particularly useful for a roaming radio transmitter/receiver.

When beginning operation in an area for which the radiotelephone is not configured and it is required to download the data for reconfiguration from one of the available networks, a communication link must first be established with the network of interest. It has been proposed that a pilot channel be established in all areas from which the roaming radiotelephone may obtain the data necessary for reconfiguration.

A pilot channel of this type, however, will require a relatively large bandwidth to allow a sufficiently fast transfer of the data required.

According to the invention there is provided a method of downloading reprogramming data from a network for installation in a radio transmitter/receiver comprising initial communication from a first dedicated channel of relatively small bandwidth broadcasting at least the frequency and radio access parameters of a second channel of relatively large bandwidth from which reprogramming data may be downloaded.

Examples of the invention will now be described in more detail with reference to the accompanying figures in which

figure 1 Illustrates the logical structure of the bootstrap channel

Figure 2 Is a flow diagram of a reconfiguration process

figure 3 Is a flow diagram of an alternative reconfiguration process

A roaming radio transmitter/receiver (mobile) is located in a region served by one or more networks and the user wishes to communicate with a network from which he can obtain reprogramming data and subsequently begin communicating with the network in the communication mode selected.

A pilot channel broadcast is maintained in the region and contained in the pilot channel broadcast there is at least sufficient information for the mobile to connect to a second channel which we shall call the bootstrap channel. Conveniently the pilot channel will be broadcast in all regions over a standardised radio interface. Only a small bandwidth is required for the pilot channel because of the small amount of information contained in the broadcast.

The small bandwidth requirement makes the task of standardisation much easier with respect to the pilot channel. The wider bandwidth channels are more conveniently assigned locally for ease of implementation.

The Pilot Channel (P_CH) broadcasts a list of sets of parameters corresponding to networks available in the region. The mobile receives the network transmission through the P_CH. If the existing configuration of the mobile is matched to the available regional radio schemes, then a second channel the bootstrap channel (B_CH) is logically mapped onto the selected transmission mode. The base station and mobile exchange information over this dedicated logical channel.

The Bootstrap channel is logically mapped on top of one of the default modes of the terminal; a mapping of a logical B_CH onto the physical GSM channel for instance may be implemented. Once the mapping has been effected the terminal may download data from the base station. The bootstrap channels provided by each operator may accommodate differing services with regard to the applications available for downloading.

The flow diagram shown at fig 3 depicts a reconfiguration procedure.

When the mobile is switched on, it reads the Pilot Channel broadcast. The mobile must be configured to support the (standardised) radio interface of the Pilot Channel. The Pilot Channel carries local radio parameters (standards supported in the regional environment in which the mobile is located). After processing the received information, the mobile

communicates with the base station through the Bootstrap Channel, provided that the mobile has the minimum resources required by its local radio environment. Prior to the change of channel, P_CH to B_CH, a logical mapping of the Bootstrap Channel is performed within the mobile on the selected air interface.

When operation on a local B_CH transmission has been established, the user may wish to change some properties or the performance of his mobile and can request supply of the desired services from the network. If no changes are required then the mobile adopts the default transmission mode in stand-by and releases the allocated B_CH.

If the user requests a change then communication between the base station and mobile is maintained for the exchange, the nature of which will depend on the capabilities of both mobile and network. At least 3 conditions can affect the nature of this information exchange.

Firstly, the mobile may not be able to support the required software. Where the mobile is not able to support the required software, no communication channel is available to the mobile from the existing network resources and use of the mobile within the region will therefore not be possible.

Secondly, the required software may be stored already in the mobile's memory. In this situation there is no need to download a software module but the allocated B_CH connection is maintained for further operations as described.

Thirdly, the software module required to support a different type of communication or user application may need to be downloaded from the base station. Where the download of a software module is required, initially a selection script is downloaded to the mobile followed by downloading and installation of the required software.

When the installation of the required software into the mobile has been completed, the mobile signals to the network the achievement of correct reconfiguration. On receipt of the "correct reconfiguration" signal from the mobile details of the mobile identity and its present configuration are entered on the network database (to license the product for instance) .

With reference to figure 1, the logical structure of the bootstrap channel will include 2 logical sub-channels : a download channel and a signalling control channel (S_CH). The signalling control channel assists in the reduction of errors in transmission so as to allow correct software download.

In the above example, the first channel, the Pilot Channel, is standardised and the mobile must be configured to support the radio interface for the Pilot Channel. The second (bootstrap) channel may be subject to local definition through logical mapping on a local transmission mode e.g. GSM, DECT and the mobile is not initially configured to support the radio interface for the bootstrap channel..

An example of a method of reprogramming providing greater flexibility will now be given. In this example the mobile is configured to support the radio interfaces for both the first, dedicated relatively small bandwidth (Pilot) channel and the second relatively large bandwidth (bootstrap) channel. That is to say that when the mobile is switched on in most and preferably all regions, the network can communicate with the mobile via both pilot and bootstrap channels.

In order for the mobile always to have the appropriate radio interface for the bootstrap channel then this channel would need also to be standardised (in addition to the Pilot Channel). The parameters of the bootstrap channels provided in different regions may have local variations in terms of e.g. allocated frequency, data rate and available user applications.

With reference to figure 3 which is a flow diagram of the reconfiguration process for this example, the mobile when switched on reads the Pilot Channel broadcast. The allocated frequency and radio resource parameters for the bootstrap channel contained in the pilot channel broadcast are processed and any required logical mapping effected. After processing the received information, the mobile communicates with the base station through the Bootstrap Channel.

The condition likely to be experienced in the previous example whereby the mobile is not able to support the required software and no communication channel is available to the mobile from the existing network resources does not apply in this arrangement. The communication via the bootstrap

channel allows the request for and supply of the software module necessary to establish communication with the network. The transfer to the bootstrap channel does not depend on the existing configuration of the mobile since the bootstrap channel is standardised in this example and the mobile is equipped to interface, via the pilot channel, with the bootstrap channel.

The services and structure offered by the Bootstrap Channel are common for both of the above examples, however, the requirements on the terminals and networks differ.

The bootstrap channel will provide the following services by means of over -the-air (OTA) reconfiguration :
capability Exchange - the terminal provides some information to the network on its current configuration and capabilities.
module Selection : at this stage the user specifies the software that his terminal requires to download. This operation could be compared to an installation script.
data download : transfer of the data. In some cases software code will have to be downloaded whilst in other cases the software may already be implemented in the mobile. In the latter case, a set-up mechanism would be sufficient to initiate the reconfiguration.

Once the mobile and the base station are synchronised on the bootstrap channel, information exchange can begin.

Claims

1. A method of downloading reprogramming data from a network for installation in a radio transmitter/receiver comprising initial communication from a first dedicated channel of relatively small bandwidth broadcasting at least the frequency and radio access parameters of a second channel of relatively large bandwidth from which reprogramming data may be downloaded.

2. A method of downloading reprogramming data from a network as in claim 1 where first, dedicated relatively small bandwidth channel has a standard radio interface common to many network locations.

3. A method of downloading reprogramming data from a network as in claim 2 where second relatively large bandwidth channel has a standard radio interface common to many network locations.

4. A method of downloading reprogramming data from a network as in claims 1 to 3 where first, dedicated relatively small bandwidth channel broadcasts a list of sets of parameters corresponding to networks available in the region.

5. A method of downloading reprogramming data from a network as in claim 1 where the radio transmitter/receiver is configured to support the radio interfaces for both the first, dedicated relatively small bandwidth channel and the second relatively large bandwidth channel.

The Patent Office

10

Application No: GB 9802545.5 **Examiner:** Glyn Hughes
Claims searched: 1 to 5 **Date of search:** 17 August 1998

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.P): H4L (LDSC, LDSU, LECC, LECX)

Int Cl (Ed.6): H04Q 7/32, 7/38

Other: Online: WPI

Documents considered to be relevant:

Category	Identity of document and relevant passage		Relevant to claims
X	US 5613204	(HABERMAN ET AL) see in particular column 15 lines 48 to 50	1
X	US 5109403	(SUTPHIN) see abstract	1

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.

EP0814589

Publication Title:

System and method for automated network reconfiguration

Abstract:

Abstract of EP0814589

A method is disclosed for providing an enhanced level of security for sensitive or proprietary information associated with information transactions in a public network, such as the Internet. In carrying out that method, an on-line information transaction is bifurcated between a generalized information access portion of such a transaction and an exchange of sensitive user information. With such a bifurcation, the generalized information access portion of the transaction, which generally would constitute the more substantial (in terms of network resources) portion of the transaction, would be handled via a non-secure network, usually a public network such as the Internet. The portion of the transaction involving sensitive user information, on the other hand, would be handled by a separate secure connection, such as a private network, or in 10a7 tranetwork. An important characteristic of this bifurcation arrangement is the provision of a means for automated reconfiguration of a user terminal as between accessing the Ageneralized information via the non-secure network and access to the secure communications network for the exchange of sensitive user information. Such an automated reconfiguration will be carried out without the necessity for any action on the part of the user, and indeed will be largely invisible to the user.

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 814 589 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
29.12.1997 Bulletin 1997/52

(51) Int. Cl.[6]: **H04L 29/06**

(21) Application number: **97109792.8**

(22) Date of filing: **16.06.1997**

(84) Designated Contracting States:
AT BE CH DE DK ES FR GB GR IE IT LI LU MC NL PT SE

(30) Priority: **19.06.1996 US 667524**

(71) Applicant: **AT&T Corp.**
New York, NY 10013-2412 (US)

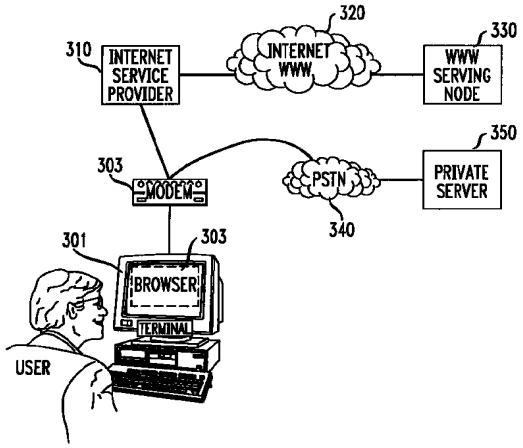
(72) Inventors:
- **Harwood, Jonathan P.**
Morganville, N.J. 07751 (US)
- **Kimmeth, Thomas**
Gladstone, N.J. 07977 (US)
- **Nusbaum, Kurt**
Downers Grove, Illinois 60515 (US)

(74) Representative:
KUHNEN, WACKER & PARTNER
Alois-Steinecker-Strasse 22
85354 Freising (DE)

(54) System and method for automated network reconfiguration

(57) A method is disclosed for providing an enhanced level of security for sensitive or proprietary information associated with information transactions in a public network, such as the Internet. In carrying out that method, an on-line information transaction is bifurcated between a generalized information access portion of such a transaction and an exchange of sensitive user information. With such a bifurcation, the generalized information access portion of the transaction, which generally would constitute the more substantial (in terms of network resources) portion of the transaction, would be handled via a non-secure network, usually a public network such as the Internet. The portion of the transaction involving sensitive user information, on the other hand, would be handled by a separate secure connection, such as a private network, or intranetwork. An important characteristic of this bifurcation arrangement is the provision of a means for automated reconfiguration of a user terminal as between accessing the Ageneralized information via the non-secure network and access to the secure communications network for the exchange of sensitive user information. Such an automated reconfiguration will be carried out without the necessity for any action on the part of the user, and indeed will be largely invisible to the user.

FIG. 3



EP 0 814 589 A2

Description

FIELD OF THE INVENTION

This invention is related to the field of data communications, and more particularly to a method and means for establishing an automatic reconfiguration of a user terminal among alternative tasks.

BACKGROUND OF THE INVENTION

With the increasing popularity of personal computers over the last several years has come a striking growth in transaction-oriented computer-to-computer communications (as opposed to bulk-data transfers among such computers). For convenience herein such transaction-oriented computer-to-computer communications will be described by the shorthand term "information transaction". That growth in the use of computers for such information transactions has unquestionably been fueled by the existence of an international infrastructure for implementing such data communications, known as the Internet. And, driven by the burgeoning demand for such information transaction services, the Internet has itself experienced explosive growth in the amount of traffic handled.

At least partly in response to that demand, a new level of accessibility to various information sources has recently been introduced to the Internet, known as the World Wide Web ("WWW"). The WWW allows a user to access a universe of information which combines text, audio, graphics and animation within a hypermedia document. Links are contained within a WWW document which allow simple and rapid access to related documents. Using a system known as the HyperText Markup Language ("HTML"), pages of information in the WWW contain pointers to other pages, those pointers typically being a key word (commonly known as a hyperlink word). When a user selects one of those key words, a hyperlink is created to another information layer (which may be in the same, or a different information server), where typically additional detail related to that key word will be found.

In order to facilitate implementation of the WWW on the Internet, new software tools have been developed for user terminals, usually known as Web Browsers, which provide a user with a graphical user interface means for accessing information on the Web, and navigating among information layers therein. A commonly used such Web Browser is that provided by Netscape.

The substantial growth in the use of computer networks, and particularly the WWW, for such information transactions, has predictably led to significant commercialization of this communications medium. For example, with the WWW, a user is not only able to access numerous information sources, some public and some commercial, but is also able to access "catalogs" of merchandise, where individual items from such a catalog can be identified and ordered, and is able to carry out a number of banking and other financial transactions. As will be obvious, such commercial transactions will typically involve sensitive and proprietary information, such as credit card numbers and financial information of a user. Thus, with the growth of commercial activity in the Internet, has also come a heightened concern with security.

It is well known that there are persons with a high level of skill in the computer arts, commonly known as "hackers", who have both the ability and the will to intercept communications via the Internet. Such persons are thereby able to gain unauthorized access to various sensitive user information, potentially compromising or misappropriating such information.

The vulnerability of such sensitive user information to misuse when so transmitted via the Internet is a phenomena which has only recently received wide public attention. Unless such security concerns can be quickly addressed and alleviated, the commercial development of this new communications medium may be slowed or even stalled altogether.

SUMMARY OF THE INVENTION

Accordingly, it is an object of the invention to provide an acceptable level of security for sensitive or proprietary information associated with information transactions in a public network, such as the Internet. That object is realized through an arrangement whereby an on-line information transaction is bifurcated between a generalized information access portion of such a transaction and an exchange of sensitive user information. With such a bifurcation, the generalized information access portion of the transaction, which generally would constitute the more substantial (in terms of network resources) portion of the transaction would be handled via a non-secure network, usually a public network such as the Internet. The portion of the transaction involving sensitive user information, on the other hand, would be handled by a separate secure connection, such as a private network, or intranetwork. An important characteristic of this bifurcation arrangement is the provision of a means for automated reconfiguration of a user terminal as between accessing the generalized information via the non-secure network and access to the secure communications network for the exchange of sensitive user information. Such an automated reconfiguration will be carried out without the necessity for any action on the part of the user, and indeed will be largely invisible to the user. In a further embodiment of the invention, a transfer of data is provided from a public to a private network, wherein data selected by a user from a public net-

work site may be arranged and displayed at a user terminal and, subject to further user selection/confirmation activity, thereafter transferred to a private network.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 depicts an illustrative case of information transactions carried out via a public network such as the Internet.

Figure 2 shows the architecture of a browser as would typically be applied for accessing a hypermedia web page.

Figure 3 illustrates the primary elements of the reconfigurable dual-path method of the invention.

Figure 4 depicts in flow chart form the basic jump capability of the methodology of the invention.

Figures 5A & 5B (generally designated collectively herein as "Figure 5") depict in flow chart form the "shopping cart" capability of the methodology of the invention.

Figure 6A & 6B (generally designated collectively herein as "Figure 6") depict in flow chart form the stored configuration capability of the methodology of the invention.

Figure 7A & 7B (generally designated collectively herein as "Figure 7") depict in flow chart form the off-line form capability of the methodology of the invention.

DETAILED DESCRIPTION

For clarity of explanation, the illustrative embodiment of the present invention is presented as comprising individual functional blocks. The functions these blocks represent may be provided through the use of either shared or dedicated hardware, including, but not limited to, hardware capable of executing software.

Figure 1 depicts an illustrative case of information transactions carried out via the Internet. As seen in the figure, an exemplary user obtains access to the Internet by First connecting, via a Terminal **110** having an associated Browser **111**, to an Internet Service Provider **112** selected by the user. That connection between the user and the Internet Service Provider will typically be made via the Public Switched Telephone Network (PSTN) from a modem associated with the user's Terminal to a network node in the Internet maintained by the selected Internet Service Provider.

Once the user has obtained access to the selected Internet Service Provider, an address is provided for connection to another user or other termination site and such a connection is made via the Internet to that destination location. As can be seen from the figure, communication via the Internet may be either user-to-user, as from Terminal **110** to Terminal **130**, or from a user to a node representing an information source accessed via the Internet, such as Public Site **120**.

It will of course be understood that the Internet provides service to a large number of users and includes a large number of such Public Sites, but the illustration provides the essential idea of the communication paths established for such Internet communication. It will also be understood that a number of service classifications are supported by the Internet, with the World Wide Web service, which represents a preferred embodiment for the public network aspect of the method of the invention, being one of the currently most heavily trafficked of such services.

The Web Browser, such as depicted at **111**, can be seen as a software application operating in conjunction with a user terminal (such as Terminal **110**) which provides an interface between such a user terminal and the particular functionality of the WWW information site. The architecture of such a browser is generally described in terms of three main components, as illustrated in Figure 2. At the top level is the Browser **201**, which enables the acquisition of information pages from a WWW server (beginning, in all cases, with the "home page" for that server), for display at a display device associated with the terminal. The Browser also provides the necessary interface for the terminal with the HTML functionality used by the server to provide access to other linked information layers.

The second level of the browser architecture is the TCP/IP Stack **202**, which handles the communications protocols used for connecting the terminal to the WWW server. The bottom level of this architecture is the Dialer **203**, which typically handles the function of providing dialing and setup digits to a modem, as illustrated at **204**, such a modem generally being a part of the terminal. Normally, upon receiving dialing and other setup information from the dialer, the modem would cause a connection to be made via the PSTN to the Internet Service Provider selected for that terminal.

After a connection is established in this manner to the Internet Service Provider, an address would be provided for the WWW information node sought to be contacted, a connection to that node made through the Internet, and the home page for that node caused to be displayed at the terminal's display device. A user would then select a key word in that home page, typically by clicking on the word with a mouse or similar device, and, upon transmission of that selection signal to the WWW server, a hyperlink would be created to the linked information layer and the open page of that layer would be caused to be displayed at the user terminal.

As explained above, serious questions have been raised in respect to the security of communications via the public Internet. (Note, that the discussion herein is focused on the Internet, and particularly the WWW functionality of the Internet, as a preferred embodiment of such public data communication networks generally, but the methodology of the invention will be applicable to any such network.) To address this problem, the methodology of the invention begins with a bifurcation of the information transaction between a user and the selected information transaction provider into a por-

tion related to sensitive or proprietary user information, and other information comprising that transaction. With such a bifurcation, it becomes possible to provide substantial security for that proprietary information by use of an alternative communications path for that separated portion of the transaction via a private network, or intranetwork -- *i.e.*, a connection between a user's terminal and a secure serving node on that private network. It is anticipated that a coordination means will be established in respect to the management of information among the public and private network elements of the bifurcated information transaction.

In its basic form, this methodology may be carried out by the user terminal initiating a call via the Internet to a selected WWW node, and upon establishing connection to that node, proceeding with the desired information transaction up to the point where an exchange of sensitive or proprietary information were required. At that point the user terminal would be instructed by the WWW server to terminate that connection (*i.e.*, hangup) and to place a new call to an identified private network server for the necessary exchange of sensitive information.

However, in order to accomplish such a dual-path transaction, it is necessary that the browser at the user terminal be reconfigured to provide the dialing, authorization (*i.e.*, login and password), and other needed information for accessing the alternative private network, in order to implement the proprietary portion of the transaction. It will also usually be the case that, upon completion of that private-network transaction, the original dialer, stack and browser configurations will need to be restored, in order for the terminal to retain its normal Internet access functionality. Such a reconfiguration and subsequent restoral of the necessary parameters in the browser, stack and dialer is likely to be well beyond the capabilities of the average user.

Accordingly, as a further embodiment of the inventive methodology, an automated browser reconfiguration means is provided which interoperates with the browser. This browser reconfiguration means is described in detail hereafter and will be referred to as the "Bridging Software".

Figure 3 provides an illustration of the primary elements of the reconfigurable dual-path method of the invention. As seen in the figure, a first path comparable to the Internet link shown in Figure 1, between User Terminal **301** and WWW Serving Node **330** (via Browser **302**, Modem **303**, Internet Service Provider **310**, and Internet **320**) is provided. However, an alternative path is now provided from the output of Modem **303** to Private Server **350**. That path is illustrated as being via the PSTN, which is generally regarded as being highly secure, but an alternative dedicated or other more-secure path between the User Terminal **301** and the Private Server **350** could as well be provided. In keeping with the discussion above, Browser **302** shown in Figure 3 would also include the Bridging Software installed as a helper application for implementing the automatic reconfiguration of the Browser.

In the operation of this system, a user would normally make an initial connection to an Internet application, such as the application represented by WWW Serving Node **330**, which, *e.g.*, might be a shopping application, a financial transaction, or the provision of an enrollment form for off-line preparation. After conducting all, or some portion of an information transaction short of an exchange of sensitive or proprietary information, including a capture by the user's terminal of needed information from the public site, a user provides a signal indicative of an end to that portion of that transaction. During the course of the public portion of the information transaction, specially configured files are sent from the WWW serving node to the Bridging Software associated with Browser **302**. Such files contain instructions for the Bridging Software to store information-like products -- e.g., for selected items from a catalog, forms for enrollment, or non-secure portions of a financial transaction, and reconfiguration information for dialing and logging into the private portion of the transaction. The Bridging Software then hangs up the Internet connection, edits the user terminal's browser, stack and dialer files to reconfigure the terminal to connect to the private server. Prior to automatic redialing of the new private site for the user, the Bridging Software may be instructed by the application operating at WWW Server Node **330** to display items chosen for purchase, or to display a form for the end-user to complete off-line before dialing the private application. Upon connecting to the private application and completing the transaction as to the user sensitive information in a private environment, the Bridging Software then restores the end-user software to the dialing and authorization parameters required to dial to the public Internet.

A particularly advantageous application of the automated reconfiguration and information transfer methodology of the Bridging Software is that it adds value to certain WWW servers which do not possess the Common Gateway Interface ("CGI") capability -- *i.e.*, a provision of specialized functions on the server beyond just displaying HTML files, and are accordingly unable to accomplish any transactional processing in respect to items selected by a user. In effect, such a non-CGI server, on its own, can only serve as a "billboard" for the items represented in its database.

However, with the collection and redelivery process of the Bridging Software, a data capture and processing mechanism can be implemented for servers operating in a non-CGI environment -- such servers being incapable of more than the simple delivery of static data packets corresponding to available items. The data set enabled by the Bridging Software is a mechanism for augmenting such limited server capabilities by defining a flexible mechanism for the receipt, display, and delivery of arbitrary data from one site to another.

In such a scenario, the Bridging Software receives a "shopping cart" item list from the host as a data-set defined with a static MIME data packet associated with the Bridging Software. This information comprising the data-set may be updated, displayed to the user in a "read-only" fashion, or presented to the user for order selection.

During the process of interacting with the WWW server, a user may trigger HTML links resulting in additional MIME packets for the Bridging Software being delivered to the client. These packets allow items to be added and/or removed from the specified data set or presented to the user for local confirmation. The user will interact with a pop-up screen provided by the Bridging Software which presents the items available with product information, such as part number, description, unit cost, etc. The user identifies those items which are to be placed into the "shopping cart" and the quantity of items desired. Upon completion of the form, the Bridging Software stores the order in a format suitable for subsequent delivery to the private server site.

An additional feature provided by the methodology of the Bridging Software is an automated mechanism for providing compatibility with user terminals not previously having the Bridging Software included with the terminal's browser. To that end, the Bridging Software located at an accessed public network site initially checks to see if the browser counterpart for that software is loaded at the calling user terminal. If yes, the heretofore described processes of the Bridging Software go forward. If not however, a request is sent through the public host to download the Bridging Software to the calling terminal. After such a download, a helper application loads the Bridging Software to the terminal's browser.

I. Illustrative Embodiments

A variety of browser reconfiguration applications are supported by the automated browser reconfiguration means of the invention. Four essentially diverse capabilities of this invention, which support such applications, are described hereafter as illustrative embodiments of the invention.

A. Basic Jump Capabilities

In this configuration, which is illustrated in flow chart form in Figure 4, an end-user is connected to a chosen WWW serving node (where a desired information product is made available) via a modem and an Internet browser associated with the user's terminal (**Step 401** of Figure 4). After conducting an information transaction with the selected WWW serving node for some interval (determined in relation to the specific application accessed), the user clicks on a hypertext link, or picture, to begin an automated process which will cause that public session to be terminated and a new connection established to an alternate private data network (**Step 402**).

In response to that user action, a data message containing parameter reconfiguration instructions is passed from the WWW server application to the Bridging Software at the user's terminal (**Step 403**). Upon receiving such instructions, the Bridging Software edits the user's on-line communications software parameters, reconfiguring that software to dial the alternate data network (**Step 404**). This reconfiguration is fully automatic and transparent to the user, and includes parameters such as modem dial number, login, password, and TCP/IP addresses. At that point, the Bridging Software causes the modem to disconnect the current data network connection, shutting down the browser, and to then dial the alternate private data network (**Step 405**).

With the establishment of a connection to the private server on the alternate data network, the user interacts with the alternate data network application as appropriate (**Step 406**), and after an interval completes his activity with the alternate data network and provides an indication of such completion (**Step 407**). A data message containing parameter reconfiguration instructions is then passed from the alternate data network application to the Bridging Software (**Step 408**).

At that point, the Bridging Software again edits the user's on-line communications software parameters, reconfiguring them to dial the original public data network, or another preselected network (**Step 409**). As with the first reconfiguration, this configuration is automatic and includes parameters such as modem dial number, login, password, and TCP/IP addresses. The Bridging Software automatically causes the current private data network to be disconnected by the modem (**Step 410**), and if appropriate, causes the original public data network to be redialed (**Step 411**). When such a reconnection to the public data network is established, the end-user would then continue his application in the public data network.

B. "Shopping Cart" Capability

With this configuration, illustrated in flow chart form in Figure 5, a user begins by establishing a connection to a WWW application (assuming for the moment that the application is non-CGI enabled) at a serving node for that application, using the Internet browser and modem associated with the user's terminal (**Step 501** of Figure 5). Upon finding an item in that application to be saved, or remembered for later consideration, or purchase, the user clicks on a hypertext link, or picture, representing that item (**Step 502**). That application then sends a data message to the Bridging Software containing information about the items selected (**Step 503**) and such information is stored by the Bridging Soft-

ware in the "shopping cart" file in the user's terminal (**Step 504**). Such selection download and storage steps (*i.e.*, steps 502, 503 & 504) are repeated for as many items as the user chooses to select. At any point after the Bridging Software has received the first set of item selection information, the user can instruct the Bridging Software to cause those selected items about which such information has been received to be displayed locally (at the user's terminal), where the user may review or edit (including deletion if desired) the collection of items theretofore selected. The application may also control display characteristics such as color and font for such locally displayed items. Note that in the case of a CGI-enabled application, the application itself will keep track of the items selected by the user and only download the totality of the selected items at the end of the selection process, and accordingly, the described local display option will not be applicable to such a CGI-enabled application.

At the point of completion of his "shopping", the user clicks on a hyper-text link or picture to "check out" (**Step 505**), which will begin a process of causing a jump to an alternate data network for the completion of sensitive portions of the transaction. To that end, a data message containing parameter reconfiguration instructions is passed from the WWW application to the Bridging Software (**Step 506**). It is to be noted that, as a security measure, information such as the new dial number, IP address, home page, configuration data (*e.g.*, login, password, DNS address) may be passed over the public network in encrypted form.

Upon receiving such reconfiguration instructions, the Bridging Software edits the user's on-line communications software parameters, reconfiguring that software to dial the alternate data network (**Step 507**). This reconfiguration is fully automatic and transparent to the user, and includes parameters such as modem dial number, login, password, and TCP/IP addresses. At that point, the Bridging Software causes the modem to disconnect the current data network connection, shutting down the browser, and to then dial the alternate data network (**Step 508**).

The Bridging Software passes the stored "shopping cart" data captured from the WWW application to the alternate network application (**Step 509**), where that data may be displayed for the user, permitting the user to confirm and/or modify the data (**Step 510**). The user interacts with the alternate data network application as appropriate, and after an interval completes his activity with the alternate data network (**Step 511**) and thus, by providing an appropriate completion signal to the application, completing the private portion of the information transaction (**Step 512**). A data message containing parameter reconfiguration instructions is then passed from the alternate data network application to the Bridging Software (**Step 513**).

The Bridging Software, at this point, again edits the user's on-line communications software parameters, reconfiguring them to dial the original (or another pre-defined) data network (**Step 514**). As with the first reconfiguration, this configuration is automatic and includes parameters such as modem dial number, login, password, and TCP/IP addresses. The Bridging Software automatically causes the current private data network to be disconnected by the modem (**Step 515**), and if appropriate, causes the original public data network to be redialed (**Step 516**). When such a reconnection is established to the point in the public data network where the user had left off to handle the secured aspects of his information transaction, the user would then continue his application in the public data network.

C. Stored Configuration Capabilities

For this configuration, depicted in flow chart form in Figure 6, an end-user is connected to a chosen WWW serving node (where a desired information product is made available) via a modem and an Internet browser associated with the user's terminal (**Step 601** of Figure 6). The user selects a hypertext link or picture associated with the WWW application by clicking on such link or picture (**Step 602**). A data message containing parameter reconfiguration instructions and an application icon (related to the selected hypertext link or picture) is passed from the WWW application to the Bridging Software (**Step 603**).

The Bridging Software creates an icon for display at the user's terminal, and saves a Bridging Software configuration file that is associated with that icon (**Step 604**). Such Bridging Software actions are automatic and multiple selections may he captured in this manner. At this point the user may continue the on-line session, or, if all desired selections have been made, a signal is provided from the user that the session should be discontinued (**Step 605**). The Bridging Software then automatically disconnects the current data network connection (**Step 606**).

After disconnecting from the WWW application, and following an interval determined by the user, a new application is selected by the user by clicking on the appropriate new icon displayed at the user's terminal (**Step 607**). The Bridging Software receives the reconfiguration instructions from the file associated with the selected icon (**Step 608**).

The Bridging Software edits the user's on-line communications software parameters, reconfiguring that software to dial the alternate data network (**Step 609**). The Bridging Software then automatically starts the user's Internet browser software and causes the alternate network application to be dialed by the modem associated with that terminal (**Step 610**). Upon establishing a connection to the alternate network, the user interacts with that application and completes the transaction to the user's satisfaction (**Step 611**). After a signal is sent to the alternate network indicating such completion of the user's activity (**Step 612**), a data message containing parameter reconfiguration instructions is passed from the alternate data network application to the Bridging Software (**Step 613**). That Software then causes the user's

terminal configuration parameters to be reset (**Step 614**) and the alternate data network to be automatically disconnected (**Step 615**).

D. Off-Line Form Capability

In this configuration, depicted in flow chart form in Figure 7, an end-user is connected to a chosen WWW serving node (where a desired information product is made available) via a modem and an Internet browser associated with the user's terminal (**Step 701** of Figure 7). The user selects a hypertext link or picture associated with an off-line form application -- an exemplary such form being an HTML-based form -- by clicking on such link or picture (**Step 702**). A data message containing parameter reconfiguration instructions for the Bridging Software, the selected off-line-form application, and an optional icon (related to the selected hypertext link or picture) is passed from the WWW application to the Bridging Software (**Step 703**). Note that the selected off-line form may be for either single or multiple use.

In the case of a delayed or multiple use of the selected form, the Bridging Software may create an icon for display at the user's terminal, and will save a Bridging Software configuration file that is associated with that icon (**Step 704**). The form in question is also saved on the user's terminal. Such Bridging Software actions are automatic. At this point the user may continue the on-line session, or, if all desired selections have been made, a signal is provided from the user that the session should be discontinued (**Step 705**). The Bridging Software then automatically disconnects the current data network connection (**Step 706**).

After disconnecting from the WWW application, two cases are to be considered as to the further processing of the selected form: (1) an immediate single use of the form and (2) either a delayed or multiple use of the form. In the first case, the Bridging Software edits the user's on-line communications software parameters, reconfiguring that software to dial the alternate data network. The Bridging Software then automatically starts the user's Internet browser software which is caused to display the off-line form. The user then completes the off-line form and chooses a "Submit Form" button displayed at his terminal.

In the second case, the Bridging Software will have created an icon for display at the user's terminal and saved a Bridging Software configuration file associated with that icon. Following an interval determined by the user, the off-line-form application is started by the user by clicking on the new form icon displayed at the user's terminal (**Step 707**). The Bridging Software receives the reconfiguration instructions from the file associated with the selected icon (**Step 708**).

The Bridging Software edits the user's on-line communications software parameters, reconfiguring that software to dial the alternate data network (**Step 709**). The Bridging Software then automatically starts the user's Internet browser software which is caused to display the off-line form (**Step 710**). The user then completes the off-line form and chooses a "Submit Form" button displayed at his terminal (**Step 711**).

In either the first or second case, following activation of the "Submit Form" button, the alternate network application is then caused to be dialed by the Bridging Software. Upon establishing a connection to the alternate network, the form data is passed to the alternate network (**Step 712**). The user then interacts with that application and completes the application (**Step 713**). After a signal is sent to the alternate network indicating such completion of the user's activity (**Step 714**),a data message containing parameter reconfiguration instructions is passed from the alternate data network application to the Bridging Software (**Step 715**). That Software then causes the user's terminal configuration parameters to be reset (**Step 716**) and the alternate data network to be automatically disconnected (**Step 717**).

CONCLUSION

A system and method has been described for the automatic switching of an information transaction between two or more alternate networks. This functionality, which incorporates a reconfiguration means designated herein as the Bridging Software, supports the movement of application specific data from one on-line environment to another. Among potential applications of this process for passing data between different environments are: selected items for purchase ("shopping cart"), captured data from forms, and other server captured data such as web pages visited.

The Bridging Software reconfiguration means is intended to work with various Web Browser software implementations, including the Netscape Personal Edition (NPE) Software for Windows 3.1 and 3.11, and which represents a working embodiment for the invention. The Bridging Software installs itself as a helper application within the browser application and utilizes a special MIME type configuration file to pass reconfiguration and "shopping cart" information from the server to the client software.

When an application requires a user to re-connect to a private application, a reconfiguration file is passed to the Bridging Software helper application via a CGI script or simple hyper-text link. The helper application disconnects the current data connection, reconfigures the dial parameters (dial #, login password, DNS address, and home page) and initiates the dial program so the end-user can access the private application.

When the end-user connects to the private application, the Bridging Software reconfiguration means provides the new "private server" application with data collected from the "public server", and the application resumes in a private,

secure environment.

The Bridging Software allows both short term and long term storage of dial configurations. Configurations passed to the Bridging Software can be designated as single use configurations and discarded after the application has terminated, or saved and displayed to the end-user as a dial choice by the Bridging Software.

Although the present embodiment of the invention has been described in detail, it should be understood that various changes, alterations and substitutions can be made therein without departing from the spirit and scope of the invention as defined by the appended claims. In particular, it is noted that, while the invention has been primarily described in terms of a preferred embodiment based on an automatic reconfiguration between a public and a private data network, any the methodology of the invention will be equally applicable to any set of alternate networks.

Claims

1. A method for managing a transaction via a communications path between a terminal device and a serving node in a data network, said method comprising the steps of:

   establishing an initial communications path via a first connection between said terminal device and a serving node in a first data network;
   receiving information from said serving node in said first data network for effecting a reconfiguration of said communications path for said transaction from said first connection in said first data network to a second connection in a second data network; and
   automatically connecting said terminal device to a serving node in said second data network via said second connection.

2. A method for managing a transaction via a communications path between a terminal device and a serving node in a data network, said method comprising the steps of:

   establishing an initial communications path via a first connection between said terminal device and a serving node in a first data network;
   selecting at least one information item from a data base of said information items provided at said serving node in said first data network;
   causing said selected information items to be downloaded to said terminal device via said first connection;
   receiving information from said serving node in said first data network for effecting a reconfiguration of said communications path for said transaction from said first connection in said first data network to a second connection in a second data network; and
   automatically connecting said terminal device to a serving node in said second data network via said second connection.

3. A method for managing a transaction via a communications path between a terminal device and a serving node in a data network, said method comprising the steps of:

   establishing an initial communications path via a first connection between said terminal device and a serving node in a first data network;
   identifying at least one data network application from a data base of said data network applications provided at said serving node in said first data network;
   receiving information from said serving node in said first data network for reconfiguring said terminal device for implementation of a communication path via an alternate connection between said terminal device and at least one of said identified data network applications in a second data network; and
   in response to a selection signal from a user, automatically connecting said terminal device to a selected one of said identified data network applications via said alternate connection.

4. A method for managing a transaction via a communications path between a terminal device and a serving node in a data network, said method comprising the steps of:

   establishing an initial communications path via a first connection between said terminal device and a serving node in a first data network;
   selecting an off-line form application from a data base provided at said serving node in said first data network;
   receiving information from said serving node in said first data network for reconfiguring said terminal device for implementation of a communication path via a second connection between said terminal device and said

selected off-line form application in a second data network; and

in response to, a selection signal from a user, automatically connecting said terminal device to said selected off-line form application.

5. The method for managing a transaction of Claim 1 or 2 including the further step of recognizing a signal to reconfigure said communications path from said first connection to said second connection.

6. The method for managing a transaction of Claim 3 wherein said selected data network application is operated at a serving node in said second data network.

7. The method for managing a transaction of Claim 4 wherein said selected off-line form application is operated at a serving node in said second data network.

8. The method for managing a transaction of one of the Claims 1, 2, 6 or 7 wherein said serving nodes in said first and said second data networks are manifested in a common node.

9. The method for managing a transaction of Claim 1 or 2 wherein said step of receiving information includes the further step of effecting said reconfiguration of said communications path.

10. The method for managing a transaction of Claim 1 or 2 wherein said step of automatically connecting includes the step of automatically disconnecting said first connection prior to implementation of said second connection.

11. The method for managing a transaction of Claim 1 or 2 including the further steps of:

automatically disconnecting said second connection in response to a user signal; and

reconfiguring said terminal device to enable, in response to user instruction, an implementation of a connection via an identified data network.

12. The method for managing a transaction of Claim 11 wherein said step of automatically reconfiguring said terminal device includes the step of effecting said implementation of said connection via said identified data network.

13. The method for managing a transaction of Claim 2 wherein said step of causing said selected information items to be downloaded includes the further step of causing said selected information items to be displayed at said terminal device.

14. The method for managing a transaction of Claim 13 wherein said displayed selected items can be edited by a user at said terminal device.

15. The method for managing a transaction of Claim 13 wherein display characteristics for said displayed selected items can be controlled at said terminal device.

16. The method for managing a transaction of Claim 2 wherein said step of automatically connecting includes the step of uploading said selected information items from said terminal device to said service provider via said second connection.

17. The method for managing a transaction of Claim 3 including the further steps of:

automatically disconnecting said alternate connection in response to a user signal; and

reconfiguring said terminal device to enable implementation of a pre-selected connection between said terminal device and an identified data network.

18. The method for managing a transaction of Claim 17 wherein said step of automatically reconfiguring said terminal device includes the further step of effecting said implementation of said pre-selected connection.

19. The method for managing a transaction of Claim 4 including the further step of downloading from said serving node in said first data network to said terminal device of an off-line form related to said off-line form application.

20. The method for managing a transaction of Claim 4 including the further step of uploading said downloaded off-line

form from said terminal device to said selected off-line form application, after processing by a user.

21. The method for managing a transaction of Claim 4 including the further steps of:

automatically disconnecting said connection to said selected off-line form application in response to a user signal; and
reconfiguring said terminal device to enable implementation of a pre-selected connection between said terminal device and an identified data network.

22. The method for managing a transaction of Claim 21 wherein said step of automatically reconfiguring said terminal device includes the further step of effecting said implementation of said pre-selected connection.

23. A method for managing connections between a terminal device and at least one information source/processor wherein at least two of said connections are implemented via separate communications networks, comprising the steps of:

recognizing a signal for connection to an information source/processor via a communications network other than a communications network for which a predetermined connection is configured;
causing said terminal device to implement a connection to said information source/processor via said other communications network; and
upon termination of said information source/processor connection via said other communications network, automatically reconfiguring a connection criteria in said terminal device to enable said terminal device to implement, in response to user instruction, a connection via an alternative one of said communications networks.

24. The method for managing connections of Claim 23 wherein said recognizing step occurs at a point when said terminal device is connected to a given source/processor.

25. The method for managing connections of Claim 23 wherein information items may be selected by a user at said terminal device from said given source/processor, and including the further step of causing said selected information items to be downloaded from said source/processor to said terminal device.

26. The method for managing connections of Claim 25 wherein said step of effecting connection includes the further step of uploading said selected information items from said terminal device to said other information source/processor.

27. The method for managing connections of Claim 26 wherein said selected information items are processed by said user at said terminal device prior to uploading to said other information source/processor.

28. The method for managing connections of Claim 24 including the further step of causing said given source/processor to download to said terminal device configuration data for enabling said step of effecting connection to said other information source/processor.

29. The method for managing connections of Claim 24 including the further step of causing said other source/processor to download to said terminal device configuration data for enabling said step of automatically restoring a prior connection criteria in said terminal device.

30. A method for enhancing security of certain data in an on-line information transaction comprising the steps of:

bifurcating said information transaction into a first portion comprising said certain data and a remaining portion, wherein said remaining portion is carried out via a public on-line communications connection between a terminal device and a public information server;
causing said first portion to be carried out via a secure private on-line communications connection between said terminal device and a private information server; and
automatically reconfiguring network access means in said terminal device to switch between said public connection and said private connection.

FIG. 1

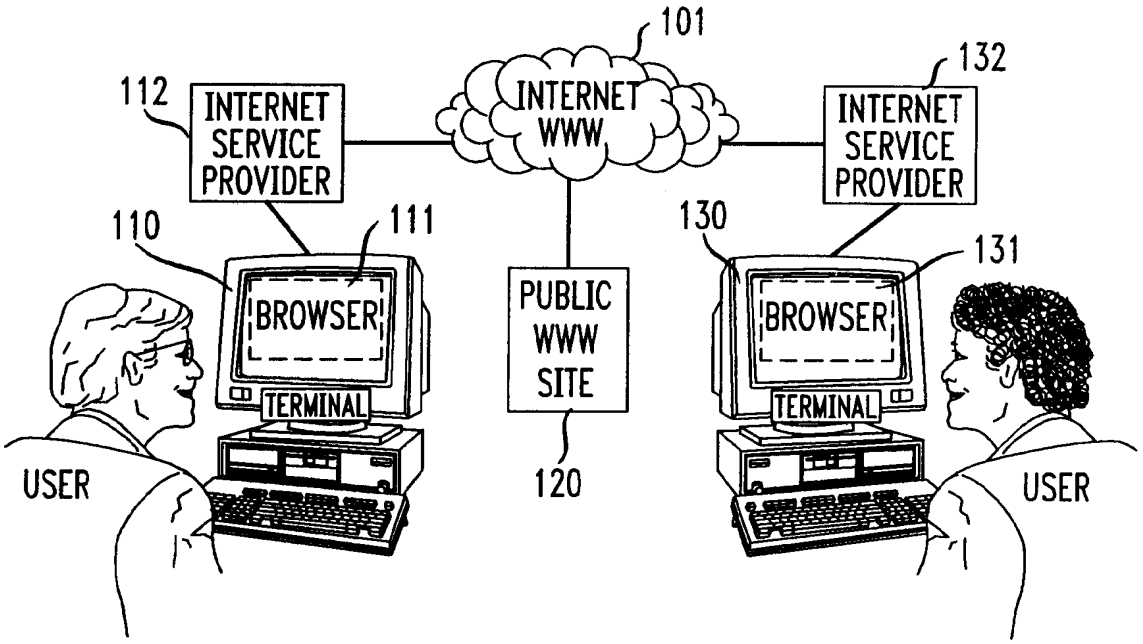


FIG. 2

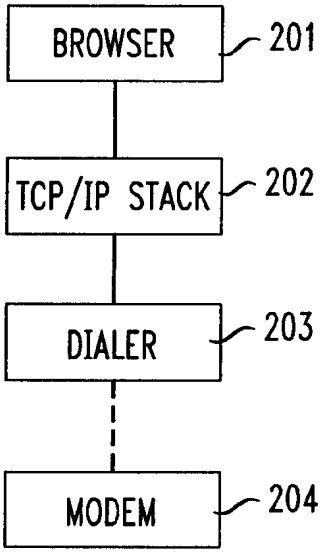


FIG. 3

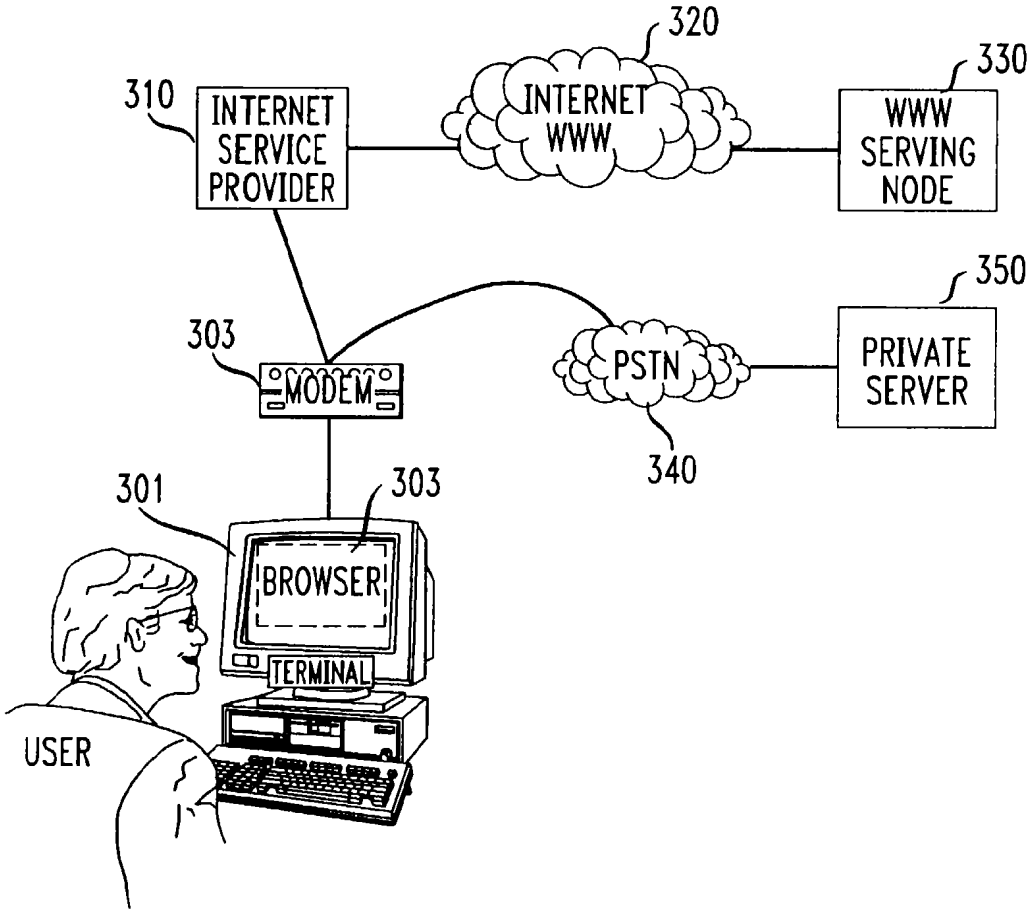


FIG. 4

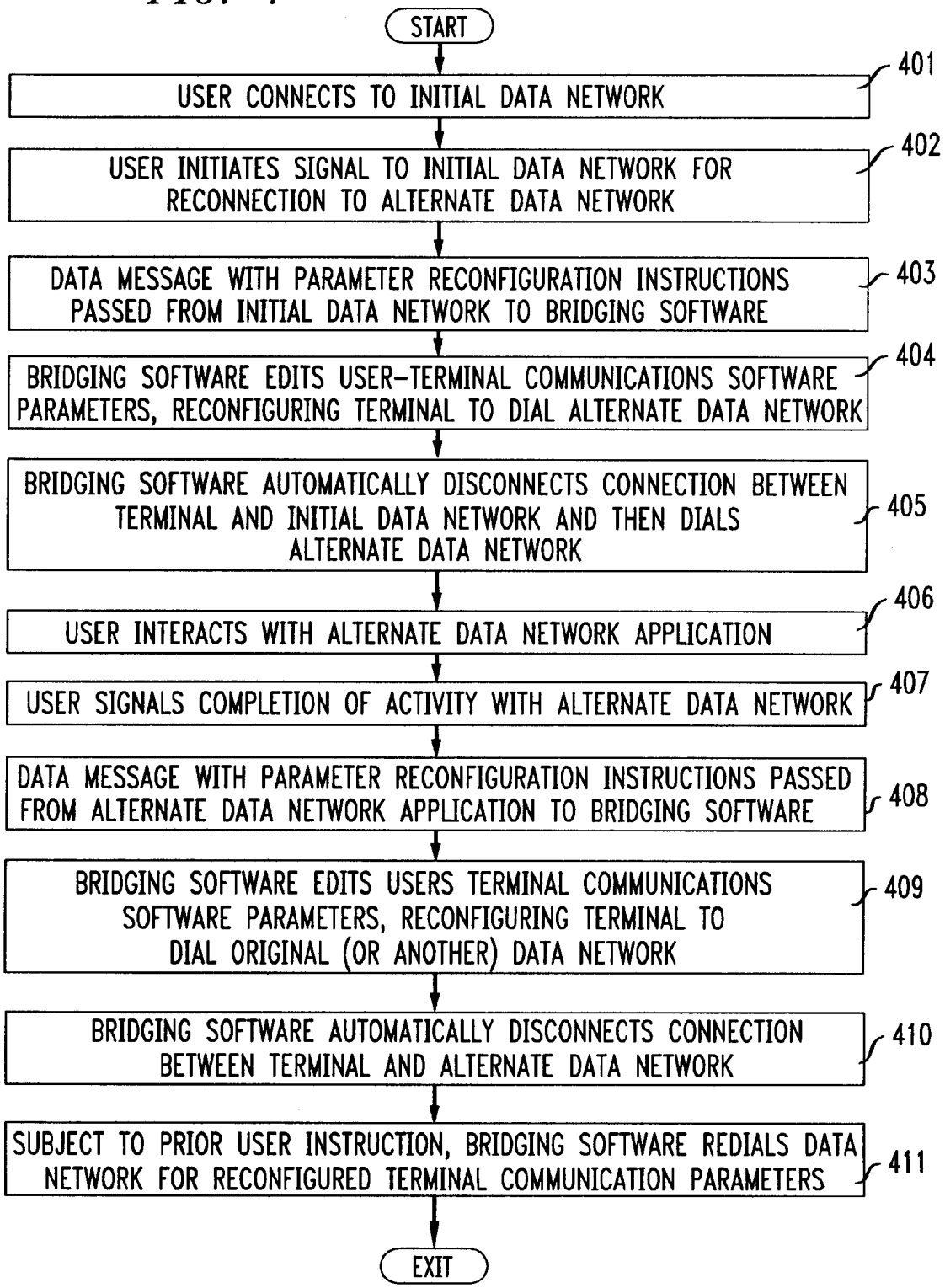


FIG. 5A

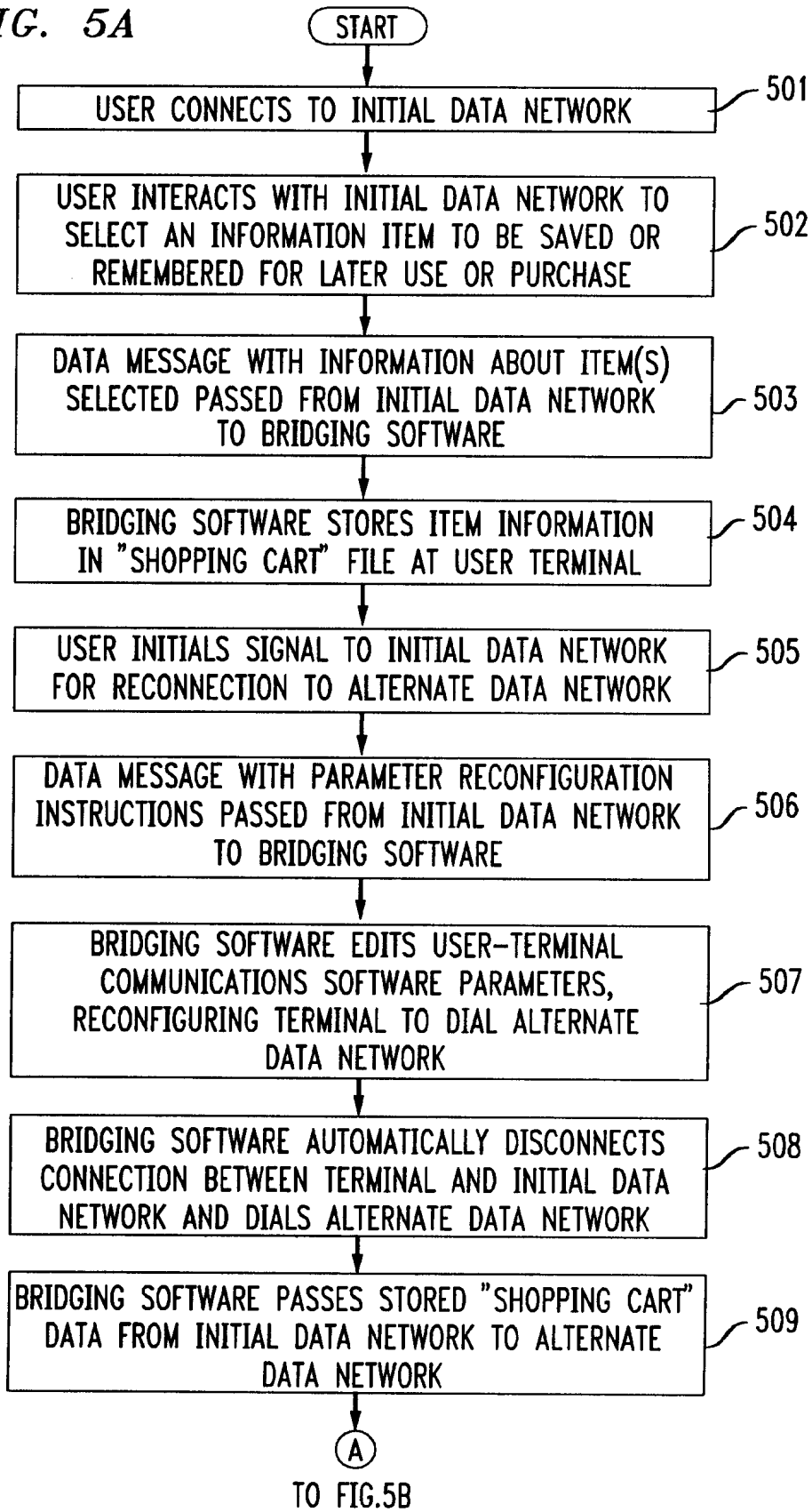


FIG. 5B

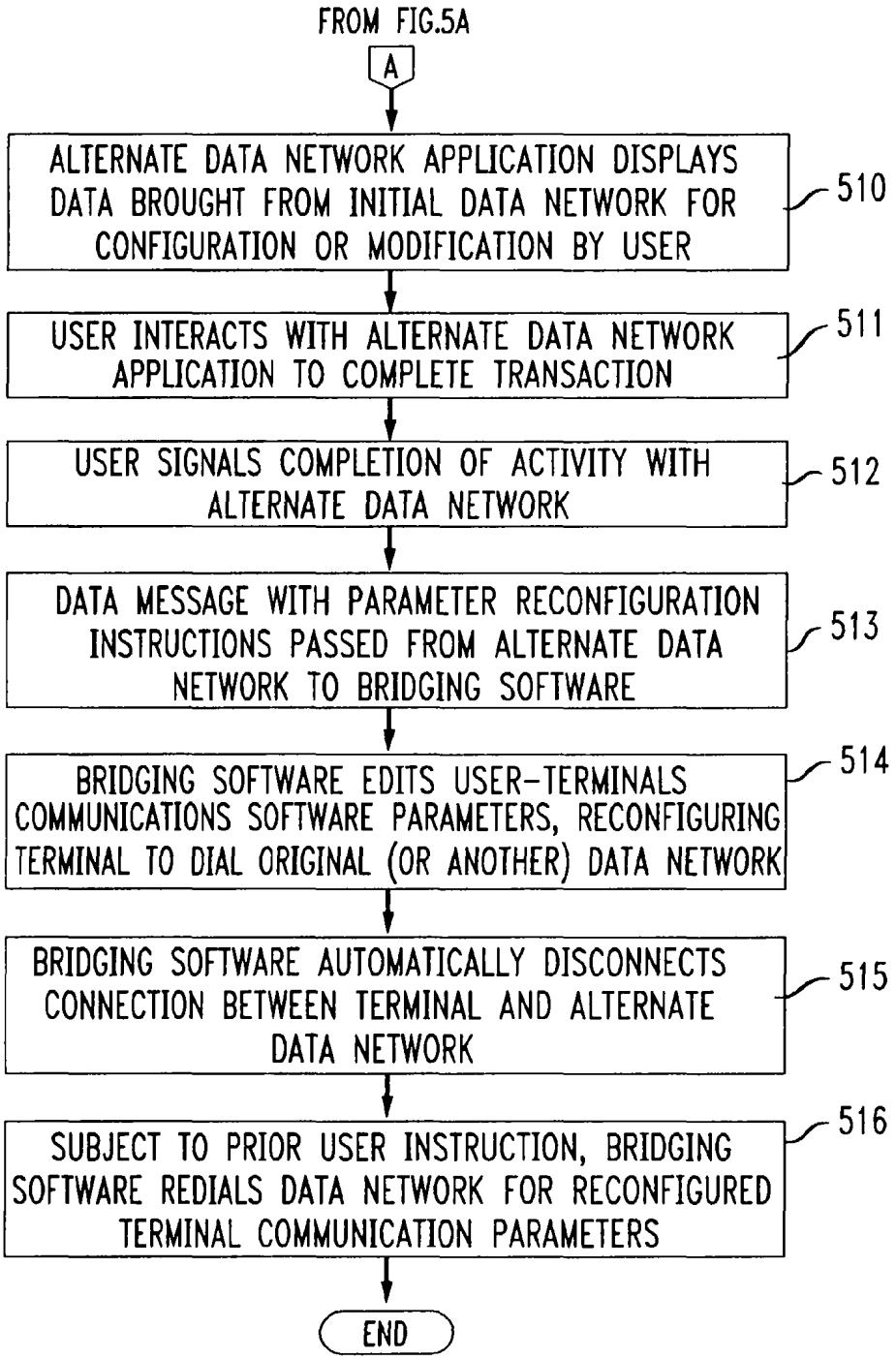


FIG. 6A

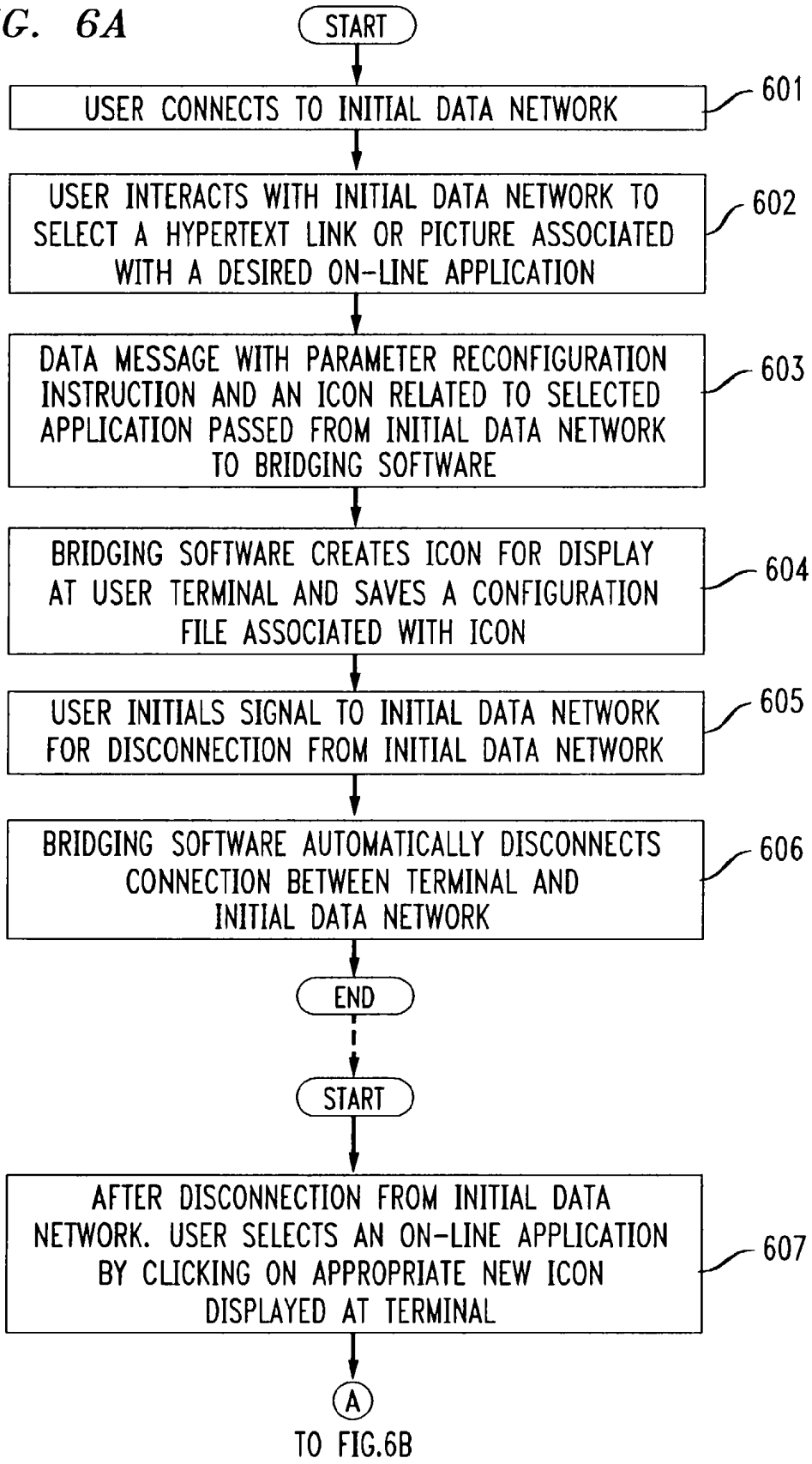


FIG. 6B

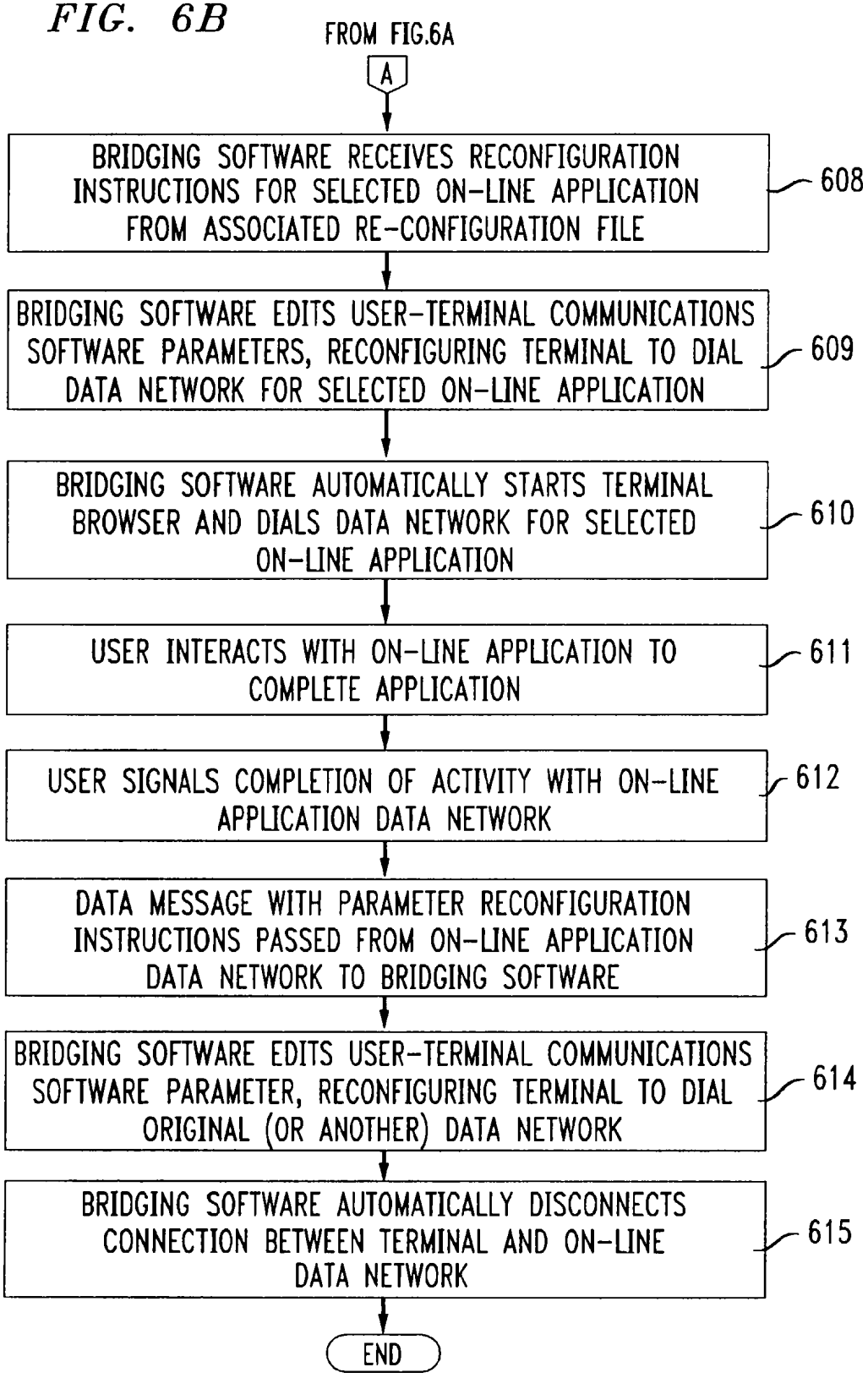


FIG. 7A

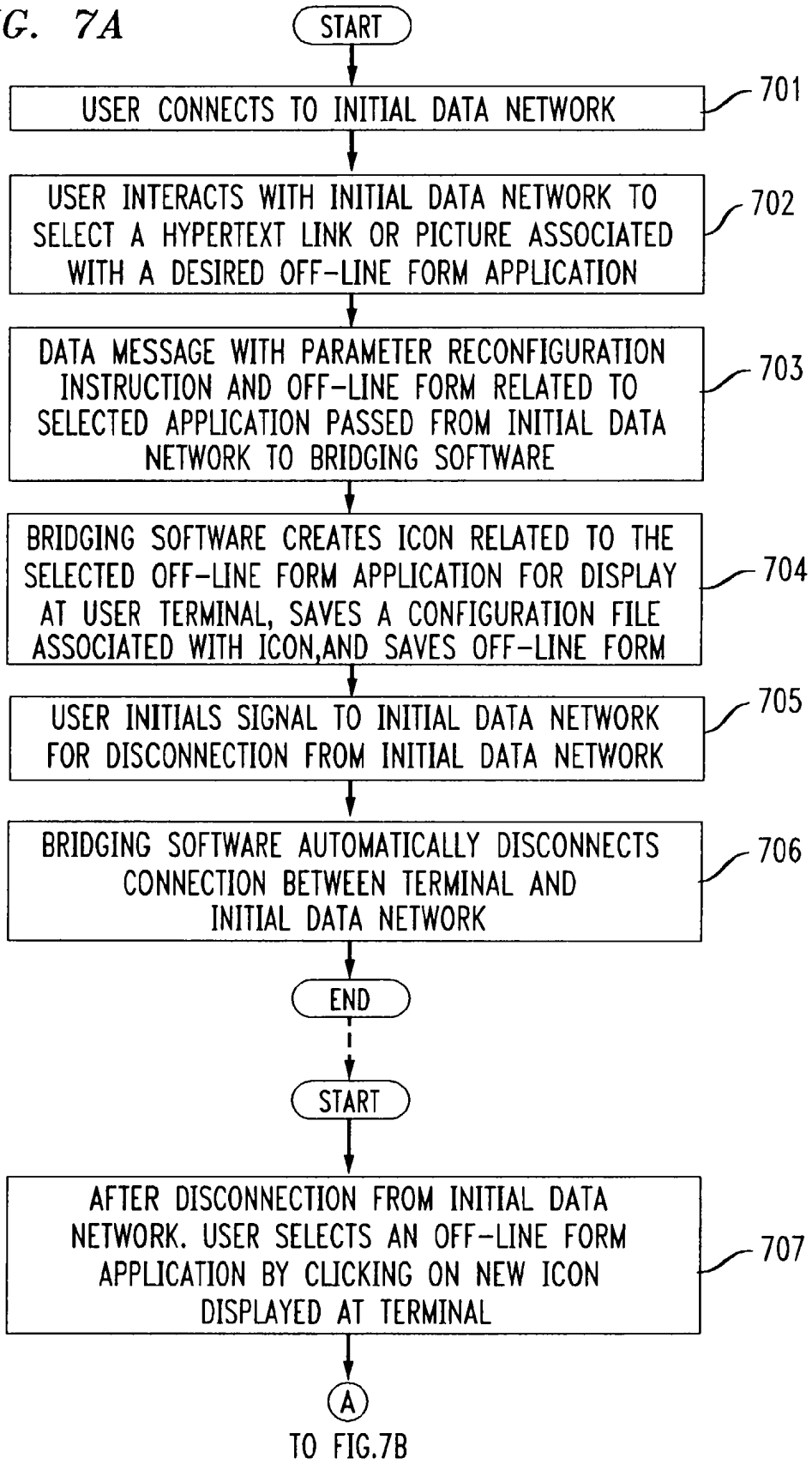
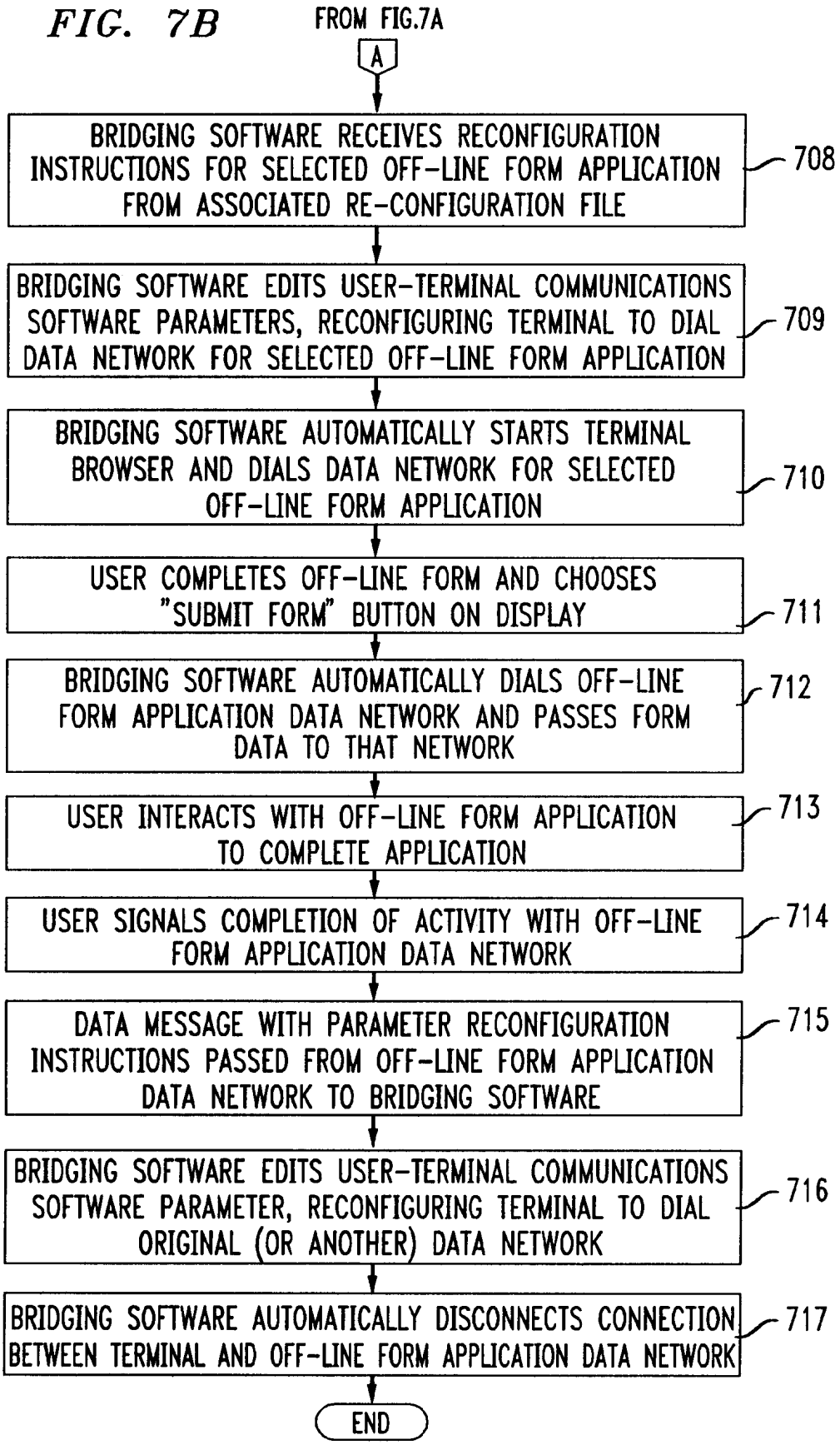


FIG. 7B



EP0838930

Publication Title:

Pseudo network adapter for frame capture, encapsulation and encryption

Abstract:

Abstract of EP0838930

A new pseudo network adapter provides an interface for capturing packets from a local communications protocol stack for transmission on the virtual private network, and includes a Dynamic Host Configuration Protocol (DHCP) server emulator, and an Address Resolution Protocol (ARP) server emulator. The new system indicates to the local communications protocol stack that nodes on a remote private network are reachable through a gateway that is in turn reachable through the pseudo network adapter. A transmit path in the system processes data packets from the local communications protocol stack for transmission through the pseudo network adapter. An encryption engine encrypts the data packets and an encapsulation engine encapsulates the encrypted data packets into tunnel data frames. The network adapter further includes an interface into a transport layer of the local communications protocol stack for capturing received data packets from the remote server node, and a receive path for processing received data packets captured from the transport layer of the local communications protocol stack. The receive path includes a decapsulation engine, and a decryption engine, and passes the decrypted, decapsulated data packets back to the local communications protocol stack for de ed1 livery to a user.

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 838 930 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
29.04.1998 Bulletin 1998/18

(51) Int. Cl.[6]: **H04L 29/06**

(21) Application number: **97118556.6**

(22) Date of filing: **24.10.1997**

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV RO SI

(30) Priority: **25.10.1996 US 738155**

(71) Applicant:
DIGITAL EQUIPMENT CORPORATION
Maynard, Massachusetts 01754 (US)

(72) Inventors:
- **Alden, Kenneth F.**
  Boylston, Massachusetts 01505 (US)
- **Lichtenberg, Mitchell P.**
  Sunnyvale, CA 94087 (US)
- **Wobber, Edward P.**
  Menlo Park, California 94025 (US)

(74) Representative: **Betten & Resch**
Reichenbachstrasse 19
80469 München (DE)

(54) Pseudo network adapter for frame capture, encapsulation and encryption

(57) A new pseudo network adapter provides an interface for capturing packets from a local communications protocol stack for transmission on the virtual private network, and includes a Dynamic Host Configuration Protocol (DHCP) server emulator, and an Address Resolution Protocol (ARP) server emulator. The new system indicates to the local communications protocol stack that nodes on a remote private network are reachable through a gateway that is in turn reachable through the pseudo network adapter. A transmit path in the system processes data packets from the local communications protocol stack for transmission through the pseudo network adapter. An encryption engine encrypts the data packets and an encapsulation engine encapsulates the encrypted data packets into tunnel data frames. The network adapter further includes an interface into a transport layer of the local communications protocol stack for capturing received data packets from the remote server node, and a receive path for processing received data packets captured from the transport layer of the local communications protocol stack. The receive path includes a decapsulation engine, and a decryption engine, and passes the decrypted, decapsulated data packets back to the local communications protocol stack for delivery to a user.

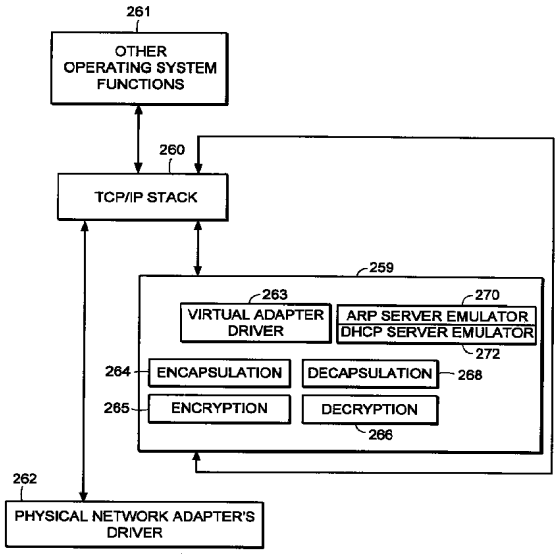


FIG. 15

EP 0 838 930 A2

Description

FIELD OF THE INVENTION

The invention relates generally to establishing secure virtual private networks. The invention relates specifically to a pseudo network adapter for capturing, encapsulating and encrypting messages or frames.

BACKGROUND

In data communications it is often required that secure communications be provided between users of network stations (also referred to as "network nodes") at different physical locations. Secure communications must potentially extend over public networks as well as through secure private networks. Secure private networks are protected by "firewalls", which separate the private network from a public network. Firewalls ordinarily provide some combination of packet filtering, circuit gateway, and application gateway technology, insulating the private network from unwanted communications with the public network.

One approach to providing secure communications is to form a virtual private network. In a virtual private network, secure communications are provided by encapsulating and encrypting messages. Encapsulated messaging in general is referred to as "tunneling". Tunnels using encryption may provide protected communications between users separated by a public network, or among a subset of users of a private network.

Encryption may for example be performed using an encryption algorithm using one or more encryption "keys". When an encryption key is used, the value of the key determines how the data is encrypted and decrypted. When a public-key encryption system is used, a key pair is associated with each communicating entity. The key pair consists of an encryption key and a decryption key. The two keys are formed such that it is unfeasible to generate one key from the other. Each entity makes its encryption key public, while keeping its decryption key secret. When sending a message to node A, for example, the transmitting entity uses the public key of node A to encrypt the message, and then the message can only be decrypted by node A using node A's private key.

In a symmetric key encryption system a single key is used as the basis for both encryption and decryption. An encryption key in a symmetric key encryption system is sometimes referred to as a "shared" key. For example, a pair of communicating nodes A and B could communicate securely as follows: a first shared key is used to encrypt data sent from node A to node B, while a second shared key is to be used to encrypt data sent from node B to node A. In such a system, the two shared keys must be known by both node A and node B. More examples of encryption algorithms and keyed encryption are disclosed in many textbooks, for example "Applied Cryptography - Protocols, Algorithms, and Source Code in C", by Bruce Schneier, published by John Wiley and Sons, New York, New York, copyright 1994.

Information regarding what encryption key or keys are to be used, and how they are to be used to encrypt data for a given secure communications session is referred to as "key exchange material". Key exchange material may for example determine what keys are used and a time duration for which each key is valid. Key exchange material for a pair of communicating stations must be known by both stations before encrypted data can be exchanged in a secure communications session. How key exchange material is made known to the communicating stations for a given secure communications session is referred to as "session key establishment".

A tunnel may be implemented using a virtual or "pseudo" network adapter that appears to the communications protocol stack as a physical device and which provides a virtual private network. A pseudo network adapter must have the capability to receive packets from the communications protocol stack, and to pass received packets back through the protocol stack either to a user or to be transmitted.

A tunnel endpoint is the point at which any encryption/decryption and encapsulation/decapsulation provided by a tunnel is performed. In existing systems, the tunnel end points are pre-determined network layer addresses. The source network layer address in a received message is used to determine the "credentials" of an entity requesting establishment of a tunnel connection. For example, a tunnel server uses the source network layer address to determine whether a requested tunnel connection is authorized. The source network layer address is also used to determine which cryptographic key or keys to use to decrypt received messages.

Existing tunneling technology is typically performed by encapsulating encrypted network layer packets (also referred to as "frames") at the network layer. Such systems provide "network layer within network layer" encapsulation of encrypted messages. Tunnels in existing systems are typically between firewall nodes which have statically allocated IP addresses. In such existing systems, the statically allocated IP address of the firewall is the address of a tunnel end point within the firewall. Existing systems fail to provide a tunnel which can perform authorization based for an entity which must dynamically allocate its network layer address. This is especially problematic for a user wishing to establish a tunnel in a mobile computing environment, and who requests a dynamically allocated IP address from an Internet Service Provider (ISP).

Because existing virtual private networks are based on network layer within network layer encapsulation, they are generally only capable of providing connectionless datagram type services. Because datagram type services do not guarantee delivery of packets, existing

tunnels can only easily employ encryption methods over the data contained within each transmitted packet. Encryption based on the contents of multiple packets is desirable, such as cipher block chaining or stream ciphering over multiple packets. For example, encrypted data would advantageously be formed based not only on the contents of the present packet data being encrypted, but also based on some attribute of the connection or session history between the communicating stations. Examples of encryption algorithms and keyed encryption are disclosed in many textbooks, for example "Applied Cryptography - Protocols, Algorithms, and Source Code in C", by Bruce Schneier, published by John Wiley and Sons, New York, New York, copyright 1994.

Thus there is required a new pseudo network adapter providing a virtual private network having a dynamically determined end point to support a user in a mobile computing environment. The new pseudo network adapter should appear to the communications protocol stack of the node as an interface to an actual physical device. The new pseudo network adapter should support guaranteed, in-order delivery of frames over a tunnel to conveniently support cipher block chaining mode or stream cipher encryption over multiple packets.

SUMMARY OF THE INVENTION

A new pseudo network adapter is disclosed providing a virtual private network. The new system includes an interface for capturing packets from a local communications protocol stack for transmission on the virtual private network. The interface appears to the local communications stack as a network adapter device driver for a network adapter.

The invention, in its broad form, includes a pseudo network adapter as recited in claim 1, providing a virtual network and a method therefor as recited in claim 9.

The system as described hereinafter further includes a Dynamic Host Configuration Protocol (DHCP) server emulator, and an Address Resolution Protocol (ARP) server emulator. The new system indicates to the local communications protocol stack that nodes on a remote private network are reachable through a gateway that is in turn reachable through the pseudo network adapter. The new pseudo network adapter includes a transmit path for processing data packets from the local communications protocol stack for transmission through the pseudo network adapter. The transmit path includes an encryption engine for encrypting the data packets and an encapsulation engine for encapsulating the encrypted data packets into tunnel data frames. The pseudo network adapter passes the tunnel data frames back to the local communications protocol stack for transmission to a physical network adapter on a remote server node.

Preferably, as described hereinafter, the pseudo network adapter includes a digest value in a digest field in each of the tunnel data frames. A keyed hash function is a hash function which takes data and a shared cryptographic key as inputs, and outputs a digital signature referred to as a digest. The value of the digest field is equal to an output of a keyed hash function applied to data consisting of the data packet encapsulated within the tunnel data frame concatenated with a counter value equal to a total number of tunnel data frames previously transmitted to the remote server node. In another aspect of the system, the pseudo network adapter processes an Ethernet header in each one of the captured data packets, including removing the Ethernet header.

The new pseudo network adapter further includes an interface into a transport layer of the local communications protocol stack for capturing received data packets from the remote server node, and a receive path for processing received data packets captured from the transport layer of the local communications protocol stack. The receive path includes a decapsulation engine, and a decryption engine, and passes the decrypted, decapsulated data packets back to the local communications protocol stack for delivery to a user.

Thus there is disclosed a new pseudo network adapter providing a virtual private network having dynamically determined end points to support users in a mobile computing environment. The new pseudo network adapter provides a system for capturing a fully formed frame prior to transmission. The new pseudo network adapter appears to the communications protocol stack of the station as an interface to an actual physical device. The new pseudo network adapter further includes encryption capabilities to conveniently provide secure communications between tunnel end points using stream mode encryption or cipher block chaining over multiple packets.

BRIEF DESCRIPTION OF THE DRAWINGS

A more detailed understanding of the invention may be had from the following description of a preferred embodiment, given by way of example and to be understood in conjunction with the accompanying drawing in which:

- Fig. 1 is a block diagram showing the Open Systems Interconnection (OSI) reference model;
- Fig. 2 is a block diagram showing the TCP/IP internet protocol suite;
- Fig. 3 is a block diagram showing an examplary embodiment of a tunnel connection across a public network between two tunnel servers;
- Fig. 4 is a flow chart showing an examplary embodiment of steps performed to establish a tunnel con-

nection;

- Fig. 5 is a flow chart showing an examplary embodiment of steps performed to perform session key management for a tunnel connection;
- Fig. 6 is a block diagram showing an examplary embodiment of a relay frame;
- Fig. 7 is a block diagram showing an examplary embodiment of a connection request frame;
- Fig. 8 is a block diagram showing an examplary embodiment of a connection response frame;
- Fig. 9 is a block diagram showing an examplary embodiment of a data frame;
- Fig. 10 is a block diagram showing an examplary embodiment of a close connection frame;
- Fig. 11 is a state diagram showing an examplary embodiment of a state machine forming a tunnel connection in a network node initiating a tunnel connection;
- Fig. 12 is a state diagram showing an examplary embodiment of a state machine forming a tunnel connection in a server computer;
- Fig. 13 is a state diagram showing an examplary embodiment of a state machine forming a tunnel connection in a relay node;
- Fig. 14 is a block diagram showing an examplary embodiment of a tunnel connection between a client computer (tunnel client) and a server computer (tunnel server);
- Fig. 15 is a block diagram showing an examplary embodiment of a pseudo network adapter;
- Fig. 16 is a block diagram showing an examplary embodiment of a pseudo network adapter;
- Fig. 17 is a flow chart showing steps performed by an examplary embodiment of a pseudo network adapter during packet transmission;
- Fig. 18 is a flow chart showing steps performed by an examplary embodiment of a pseudo network adapter during packet receipt;
- Fig. 19 is a data flow diagram showing data flow in an examplary embodiment of a pseudo network adapter during packet transmission;
- Fig. 20 is a data flow diagram showing data flow in an examplary embodiment of a pseudo network adapter during packet receipt;
- Fig. 21 is a diagram showing the movement of encrypted and unencrypted data in an examplary embodiment of a system including a pseudo network adapter;
- Fig. 22 is a diagram showing the movement of encrypted and unencrypted data in an examplary embodiment of a system including a pseudo network adapter; and
- Fig. 23 is a flow chart showing steps initialization of an examplary embodiment of a system including a pseudo network adapter.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Now with reference to Fig. 1 there is described for purposes of explanation, communications based on the Open Systems Interconnection (OSI) reference model. In Fig. 1 there is shown communications 12 between a first protocol stack 10 and a second protocol stack 14. The first protocol stack 10 and second protocol stack 14 are implementations of the seven protocol layers (Application layer, Presentation layer, Session layer, Transport layer, Network layer, Data link layer, and Physical layer) of the OSI reference model. A protocol stack implementation is typically in some combination of software and hardware. Descriptions of the specific services provided by each protocol layer in the OSI reference model are found in many text books, for example "Computer Networks", Second Edition, by Andrew S. Tannenbaum, published by Prentice-Hall, Englewood Cliffs, New Jersey, copyright 1988.

As shown in Fig. 1, data 11 to be transmitted from a sending process 13 to a receiving process 15 is passed down through the protocol stack 10 of the sending process to the physical layer 9 for transmission on the data path 7 to the receiving process 15. As the data 11 is passed down through the protocol stack 10, each protocol layer prepends a header (and possibly also appends a trailer) portion to convey information used by that protocol layer. For example, the data link layer 16 of the sending process wraps the information received from the network layer 17 in a data link header 18 and a data link layer trailer 20 before the message is passed to the physical layer 9 for transmission on the actual transmission path 7.

Fig. 2 shows the TCP/IP protocol stack. Some protocol layers in the TCP/IP protocol stack correspond with layers in the OSI protocol stack shown in Fig. 1. The detailed services and header formats of each layer in the TCP/IP protocol stack are described in many texts, for example "Internetworking with TCP/IP, Vol. 1: Principles, Protocols, and Architecture", Second Edi-

tion, by Douglas E. Comer, published by Prentice-Hall, Englewood Cliffs, New Jersey, copyright 1991. The Transport Control Protocol (TCP) 22 corresponds to the Transport layer in the OSI reference model. The TCP protocol 22 provides a connection-oriented, end to end transport service with guaranteed, in-sequence packet delivery. In this way the TCP protocol 22 provides a reliable, transport layer connection.

The IP protocol 26 corresponds to the Network layer of the OSI reference model. The IP protocol 26 provides no guarantee of packet delivery to the upper layers. The hardware link level and access protocols 32 correspond to the Data link and Physical layers of the OSI reference model.

The Address Resolution Protocol (ARP) 28 is used to map IP layer addresses (referred to as "IP addresses") to addresses used by the hardware link level and access protocols 32 (referred to as "physical addresses" or "MAC addresses"). The ARP protocol layer in each network station typically contains a table of mappings between IP addresses and physical addresses (referred to as the "ARP cache"). When a mapping between an IP address and the corresponding physical address is not known, the ARP protocol 28 issues a broadcast packet (an "ARP request" packet) on the local network. The ARP request indicates an IP address for which a physical address is being requested. The ARP protocols 28 in each station connected to the local network examine the ARP request, and if a station recognizes the IP address indicated by the ARP request, it issues a response (an "ARP response" or "ARP reply" packet) to the requesting station indicating the responder's physical address. The requesting ARP protocol reports the received physical address to the local IP layer which then uses it to send datagrams directly to the responding station. As an alternative to having each station respond only for its own IP address, an ARP server may be used to respond for a set of IP addresses it stores internally, thus potentially eliminating the requirement of a broadcast request. In that case, the ARP request can be sent directly to the ARP server for physical addresses corresponding to any IP address mappings stored within the ARP server.

At system start up, each station on a network must determine an IP address for each of its network interfaces before it can communicate using TCP/IP. For example, a station may need to contact a server to dynamically obtain an IP address for one or more of its network interfaces. The station may use what is referred to as the Dynamic Host Configuration Protocol (DHCP) to issue a request for an IP address to a DHCP server. For example, a DHCP module broadcasts a DHCP request packet at system start up requesting allocation of an IP address for an indicated network interface. Upon receiving the DHCP request packet, the DHCP server allocates an IP address to the requesting station for use with the indicated network interface. The requesting station then stores the IP address in the response from the server as the IP address to associate with that network interface when communicating using TCP/IP.

Fig. 3 shows an example configuration of network nodes for which the presently disclosed system is applicable. In the example of Fig. 3, the tunnel server A is an initiator of the tunnel connection. As shown in Fig. 3, the term "tunnel relay" node is used to refer to a station which forwards data packets between transport layer connections (for example TCP connections).

For example, in the present system a tunnel relay may be dynamically configured to forward packets between transport layer connection 1 and transport layer connection 2. The tunnel relay replaces the header information of packets received over transport layer connection 1 with header information indicating transport layer connection 2. The tunnel relay can then forward the packet to a firewall, which may be conveniently programmed to pass packets received over transport layer connection 2 into a private network on the other side of the firewall. In the present system, the tunnel relay dynamically forms transport layer connections when a tunnel connection is established. Accordingly the tunnel relay is capable of performing dynamic load balancing or providing redundant service for fault tolerance over one or more tunnel servers at the time the tunnel connection is established.

Fig. 3 shows a Tunnel Server A 46 in a private network N1 48, physically connected with a first Firewall 50. The first Firewall 50 separates the private network N1 48 from a public network 52, for example the Internet. The first Firewall 50 is for example physically connected with a Tunnel Relay B 54, which in turn is virtually connected through the public network 52 with a Tunnel Relay C. The connection between Tunnel Relay B and Tunnel Relay C may for example span multiple intervening forwarding nodes such as routers or gateways through the public network 52.

The Tunnel Relay C is physically connected with a second Firewall 58, which separates the public network 52 from a private network N2 60. The second Firewall 58 is physically connected with a Tunnel Server D 62 on the private network N2 60. During operation of the elements shown in Fig. 3, the Tunnel Server D 62 provides routing of IP packets between the tunnel connection with Tunnel Server A 46 and other stations on the private network N2 60. In this way the Tunnel Server D 62 acts as a router between the tunnel connection and the private network N2 60.

During operation of the elements shown in Fig. 3, the present system establishes a tunnel connection between the private network N1 48 and the private network N2 60. The embodiment of Fig. 3 thus eliminates the need for a dedicated physical cable or line to provide secure communications between the private network 48 and the private network 60. The tunnel connection between Tunnel Server A 46 and Tunnel Server D 62 is

composed of reliable, pair-wise transport layer connections between Tunnel Server A 46 (node "A"), Tunnel Relay B 54 (node "B"), Tunnel Relay C 56 (node "C"), and Tunnel Server D 62 (node "D"). For example, such pair-wise connections may be individual transport layer connections between each node A and node B, node B and node C, and node C and node D. In an alternative embodiment, as will be described below, a tunnel connection may alternatively be formed between a standalone PC in a public network and a tunnel server within a private network.

Fig. 4 and Fig. 5 show an example embodiment of steps performed during establishment of the tunnel connection between Tunnel Server A 46 (node "A") and Tunnel Server D 62 (node "D") as shown in Fig. 3. Prior to the steps shown in Fig. 4, node A selects a tunnel path to reach node D. The tunnel path includes the tunnel end points and any intervening tunnel relays. The tunnel path is for example predetermined by a system administrator for node A. Each tunnel relay along the tunnel path is capable of finding a next node in the tunnel path, for example based on a provided next node name (or "next node arc"), using a predetermined naming convention and service, for example the Domain Name System (DNS) of the TCP/IP protocol suite.

During the steps shown in Fig. 4, each of the nodes A, B and C perform the following steps:

- resolve the node name of the next node in the tunnel path, for example as found in a tunnel relay frame;
- establish a reliable transport layer (TCP) connection to the next node in the tunnel path;
- forward the tunnel relay frame down the newly formed reliable transport layer connection to the next node in the tunnel path.

As shown for example in Fig. 4, at step 70 node A establishes a reliable transport layer connection with node B. At step 72 node A identifies the next downstream node to node B by sending node B a tunnel relay frame over the reliable transport layer connection between node A and node B. The tunnel relay frame contains a string buffer describing all the nodes along the tunnel path (see below description of an example tunnel relay frame format). At step 74, responsive to the tunnel relay frame from node A, node B searches the string buffer in the relay frame to determine if the string buffer includes node B's node name. If node B finds its node name in the string buffer, it looks at the next node name in the string buffer to find the node name of the next node in the tunnel path.

Node B establishes a reliable transport layer connection with the next node in the tunnel path, for example node C. Node B further forms an association between the reliable transport layer connection between Node A and Node B, over which the relay frame was received, and the newly formed reliable transport layer connection between Node B and Node C, and as a result forwards subsequent packets received over the reliable transport layer connection with Node A onto the reliable transport layer connection with Node C, and vice versa. At step 76 node B forwards the tunnel relay frame on the newly formed reliable transport layer connection to node C.

At step 78, responsive to the relay frame forwarded from node B, node C determines that the next node in the tunnel path is the last node in the tunnel path, and accordingly is a tunnel server. Node C may actively determine whether alternative tunnel servers are available to form the tunnel connection. Node C may select one of the alternative available tunnel servers to form the tunnel connection in order to provide load balancing or fault tolerance. As a result node C may form a transport layer connections with one of several available tunnel servers, for example a tunnel server that is relatively underutilized at the time the tunnel connection is established. In the example embodiment, node C establishes a reliable transport layer connection with the next node along the tunnel path, in this case node D.

Node C further forms an association between the reliable transport layer connection between Node B and Node C, over which the relay frame was received, and the newly formed reliable transport layer connection between Node C and Node D, and as a result forwards subsequent packets received over the reliable transport layer connection with Node B to the reliable transport layer connection with Node D, and vice versa. At step 80 node C forwards the relay frame to node D on the newly formed reliable transport layer connection.

Fig. 5 shows an example of tunnel end point authentication and sharing of key exchange material provided by the present system. The present system supports passing authentication data and key exchange material through the reliable transport layer connections previously established on the tunnel path. The following are provided by use of a key exchange/authentication REQUEST frame and a key exchange/authentication RESPONSE frame:

a) mutual authentication of both endpoints of the tunnel connection;

b) establishment of shared session encryption keys and key lifetimes for encrypting/authenticating subsequent data sent through the tunnel connection;

d) agreement on a shared set of cryptographic transforms to be applied to subsequent data; and

e) exchange of any other connection-specific data between the tunnel endpoints, for example strength and type of cipher to be used, any compression of the data to be used, etc. This data can also be used

by clients of this protocol to qualify the nature of the authenticated connection.

At step 90 a key exchange/authentication request frame is forwarded over the reliable transport layer connections formed along the tunnel path from node A to node D. At step 92, a key exchange/authentication response frame is forwarded from node D back to node A through the reliable transport layer connections. The attributes exchanged using the steps shown in Fig. 5 may be used for the lifetime of the tunnel connection. In an alternative embodiment the steps shown in Fig. 5 are repeated as needed for the tunnel end points to exchange sufficient key exchange material to agree upon a set of session parameters for use during the tunnel connection such as cryptographic keys, key durations, and choice of encryption/decryption algorithms.

Further in the disclosed system, the names used for authentication and access control with regard to node A and node D need not be the network layer address or physical address of the nodes. For example, in an alternative embodiment where the initiating node sending the tunnel relay frame is a stand-alone PC located within a public network, the user's name may be used for authentication and/or access control purposes. This provides a significant improvement over existing systems which base authorization on predetermined IP addresses.

Fig. 6 shows the format of an example embodiment of a tunnel relay frame. The tunnel frame formats shown in Figs. 6, 7, 8 and 9 are encapsulated within the data portion of a transport layer (TCP) frame when transmitted. Alternatively, another equivalent, connection-oriented transport layer protocol having guaranteed, in-sequence frame delivery may be used. The example TCP frame format, including TCP header fields, is conventional and not shown.

The field 100 contains a length of the frame. The field 102 contains a type of the frame, for example a type of RELAY. The field 104 contains a tunnel protocol version number. The field 106 contains an index into a string buffer field 112 at which a name of the originating node is located, for example a DNS host name of the node initially issuing the relay frame (node A in Fig. 3). The fields following the origin index field 106 contain indexes into the string buffer 112 at which names of nodes along the tunnel path are located. For example each index may be the offset of a DNS host name within the string buffer 112. In this way the field 108 contains the index of the name of the first node in the tunnel path, for example node B (Fig. 3). The field 110 contains the index of the name of the second node in the tunnel path, etc. The field 112 contains a string of node names of nodes in the tunnel path.

During operation of the present system, the initiating node, for example node A as shown in Fig. 3, transmits a tunnel relay frame such as the tunnel relay frame shown in Fig. 6. Node A sends the tunnel relay frame to the first station along the tunnel path, for example node B (Fig. 3), over a previously established reliable transport layer connection. Node B searches the string buffer in the tunnel relay frame to find its node name, for example its DNS host name. Node B finds its node name in the string buffer indexed by path index 0, and then uses the contents of path index 1 110 to determine the location within the string buffer 112 of the node name of the next node along the tunnel path. Node B uses this node name to establish a reliable transport layer connection with the next node along the tunnel path. Node B then forwards the relay frame to the next node. This process continues until the end node of the tunnel route, for example tunnel server D 62 (Fig. 3) is reached.

Fig. 7 shows the format of an example embodiment of a key exchange/key authentication request frame. The field 120 contains a length of the frame. The field 122 contains a type of the frame, for example a type of REQUEST indicating a key exchange/key authentication request frame. The field 124 contains a tunnel protocol version number. The field 126 contains an offset of the name of the entity initiating the tunnel connection, for example the name of a user on the node originally issuing the request frame. This name and key exchange material in the request frame are used by the receiving tunnel end point to authenticate the key exchange/authentication REQUEST. The name of the entity initiating the tunnel connection is also use to authorize any subsequent tunnel connection, based on predetermined security policies of the system. The field 128 contains an offset into the frame of the node name of the destination node, for example the end node of the tunnel shown as node D 62 in Fig. 3.

The field 130 contains an offset into the frame at which key exchange data as is stored, for example within the string buffer field 138. The key exchange data for example includes key exchange material used to determine a shared set of encryption parameters for the life of the tunnel connection such as cryptographic keys and any validity times associated with those keys. The key exchange data, as well as the field 132, further include information regarding any shared set of cryptographic transforms to be used and any other connection-specific parameters, such as strength and type of cipher to be used, type of compression of the data to be used, etc. The field 134 contains flags, for example indicating further information about the frame. The field 136 contains client data used in the tunnel end points to configure the local routing tables so that packets for nodes reachable through the virtual private network are sent through the pseudo network adapters. In an example embodiment, the string buffer 138 is encrypted using a public encryption key of the receiving tunnel end point.

During operation of the present system, one of the end nodes of the tunnel sends a key exchange/authentication REQUEST frame as shown in Fig. 7 to the other end node of the tunnel in order to perform key exchange and authentication as described in step 90 of Fig. 5.

Fig. 8 shows the format of an example embodiment of a key exchange/key authentication response frame, referred to as a connection RESPONSE frame. The field 150 contains a length of the frame. The field 152 contains a type of the frame, for example a type of connection RESPONSE indicating a key exchange/key authentication request frame. The field 154 contains a tunnel protocol version number.

The field 156 contains an offset into the frame at which key exchange data as is stored, for example within the string buffer field 163. The key exchange data for example includes key exchange material to be used for encryption/decryption over the life of the tunnel connection and any validity times associated with that key exchange material. The key exchange data, as well as the field 158, further includes information regarding any shared set of cryptographic transforms to be applied to subsequent data and any other connection-specific parameters, such as strength and type of cipher to be used, any compression of the data to be used, etc. The field 160 contains flags, for example indicating other information about the frame. The client data field 162 contains data used by the pseudo network adapters in the tunnel end points to configure the local routing tables so that packets for nodes in the virtual private network are sent through the pseudo network adapters. The string buffer includes key exchange material. The string buffer is for example encrypted using a public encryption key of the receiving tunnel end point, in the this case the initiator of the tunnel connection.

During operation of the present system, one of the end nodes of the tunnel sends a key exchange/authentication RESPONSE frame as shown in Fig. 7 to the other end node of the tunnel in order to perform key exchange and authentication as described in step 92 of Fig. 5.

Fig. 9 shows the format of an example embodiment of an tunnel data frame used to communicate through a tunnel connection. Fig. 9 shows how an IP datagram may be encapsulated within a tunnel frame by the present system for secure communications through a virtual private network. The field 170 contains a length of the frame. The field 172 contains a type of the frame, for example a type of DATA indicating a tunnel data frame. The field 174 contains a tunnel protocol version number.

The fields 176, 178 and 182 contain information regarding the encapsulated datagram. The field 180 contains flags indicating information regarding the frame. The field 184 contains a value indicating the length of the optional padding 189 at the end of the frame. The frame format allows for optional padding in the event that the amount of data in the frame needs to be padded to an even block boundary for the purpose of being encrypted using a block cipher. The field 186 contains a value indicating the length of the digest field 187.

The data frame format includes a digital signature generated by the transmitting tunnel end point referred to as a "digest". The value of the digest ensures data integrity, for example by detecting invalid frames and replays of previously transmitted valid frames. The digest is the output of a conventional keyed cryptographic hash function applied to both the encapsulated datagram 190 and a monotonically increasing sequence number. The resulting hash output is passed as the value of the digest field 187. The sequence number is not included in the data frame. In the example embodiment, the sequence number is a counter maintained by the transmitter (for example node A in Fig. 3) of all data frames sent to the receiving node (for example node D in Fig. 3) since establishment of the tunnel connection.

In order to determine if the data frame is invalid or a duplicate, the receiving node decrypts the encapsulated datagram 190, and applies the keyed cryptographic hash function (agreed to by the tunnel end nodes during the steps shown in Fig. 5) to both the decrypted encapsulated datagram and the value of a counter indicating the number of data frames received from the transmitter since establishment of the tunnel connection. For example the keyed hash function is applied to the datagram concatenated to the counter value. If the resulting hash output matches the value of the digest field 187, then the encapsulated datagram 190 was received correctly and is not a duplicate. If the hash output does not match the value of the digest field 187, then the integrity check fails, and the tunnel connection is closed. The field 188 contains an encrypted network layer datagram, for example an encrypted IP datagram.

The encapsulated datagram may be encrypted using various encryption techniques. An example embodiment of the present system advantageously encrypts the datagram 190 using either a stream cipher or cipher block chaining encryption over all data transmitted during the life of the tunnel connection. This is enabled by the reliable nature of the transport layer connections within the tunnel connection. The specific type of encryption and any connection specific symmetric encryption keys used is determined using the steps shown in Fig. 5. The fields in the tunnel data frame other than the encapsulated datagram 188 are referred to as the tunnel data frame header fields.

Fig. 10 is a block diagram showing an example embodiment of a "close connection" frame. The field 190 contains the length of the frame. The field 191 contains a frame type, for example having a value equal to CLOSE. Field 192 contains a value equal to the current protocol version number of the tunnel protocol. The field 193 contains a status code indicating the reason the tunnel connection is being closed.

During operation of the present system, when end point of a tunnel connection determines that the tunnel connection should be closed, a close connection frame as shown in Fig. 10 is transmitted to the other end point of the tunnel connection. When a close connection close frame is received, the receiver closes the tunnel

connection and no further data will be transmitted or received through the tunnel connection.

Fig. 11 is a state diagram showing an example embodiment of forming a tunnel connection in a node initiating a tunnel connection. In Fig. 11, Fig. 12, and Fig. 13, states are indicated by ovals and actions or events are indicated by rectangles. For example the tunnel server node A as shown in Fig. 3 may act as a tunnel connection initiator when establishing a tunnel connection with the tunnel server node D. Similarly the client system 247 in Fig. 14 may act as a tunnel connection initiator when establishing a tunnel connection with the tunnel server. The tunnel initiator begins in an idle state 194. Responsive to an input from a user indicating that a tunnel connection should be established, the tunnel initiator transitions from the idle state 194 to a TCP Open state 195. In the TCP Open state 195, the tunnel initiator establishes a reliable transport layer connection with a first node along the tunnel path. For example, the tunnel initiator opens a socket interface associated with a TCP connection to the first node along the tunnel path. In Fig. 3 node A opens a socket interface associated with a TCP connection with node B.

Following establishment of the reliable transport layer connection in the TCP Open state 195, the tunnel initiator enters a Send Relay state 197. In the Send Relay state 197, the tunnel initiator transmits a relay frame at 198 over the reliable transport layer connection. Following transmission of the relay frame, the tunnel initiator enters the connect state 199. If during transmission of the relay frame there is a transmission error, the tunnel initiator enters the Network Error state 215 followed by the Dying state 208. In the Dying state 208, the tunnel initiator disconnects the reliable transport layer connection formed in the TCP Open state 195, for example by disconnecting a TCP connection with Node B. Following the disconnection at 209, the tunnel initiator enters the Dead state 210. The tunnel initiator subsequently transitions back to the Idle state 194 at a point in time predetermined by system security configuration parameters.

In the Connect state 199, the tunnel initiator sends a key exchange/authentication REQUEST frame at 200 to the tunnel server. Following transmission of the key exchange/authentication REQUEST frame 200, the tunnel initiator enters the Response Wait state 201. The tunnel initiator remains in the Response Wait state 201 until it receives a key exchange/authentication RESPONSE frame 202 from the tunnel server. After the key exchange/authentication RESPONSE frame is received at 202, the tunnel initiator enters the Authorized state 203, in which it may send or receive tunnel data frames. Upon receipt of a CLOSE connection frame at 216 in the Authorized state 203, the tunnel initiator transitions to the Dying state 208.

Upon expiration of a session encryption key at 211, the tunnel initiator enters the Reconnect state 212, and sends a CLOSE connection frame at 213 and disconnects the TCP connection with the first node along the tunnel path at 214. Subsequently the tunnel initiator enters the TCP Open state 195.

If during the authorized state 203, a local user issues an End Session command at 204, or there is a detection of an authentication or cryptography error in a received data frame at 205, the tunnel initiator enters the Close state 206. During the Close state 206 the tunnel initiator sends a CLOSE connection frame at 207 to the tunnel server. The tunnel initiator then enters the Dying state at 208.

Figure 12 is a state diagram showing the states within an example embodiment of a tunnel server, for example node D in Fig. 3 or tunnel server 253 in Fig. 14. The tunnel server begins in an Accept Wait state 217. In the Accept Wait state 217, the tunnel server receives a request for a reliable transport layer connection, for example a TCP connection reguest 218 from the last node in the tunnel path prior to the tunnel server, for example Node C in Fig. 3. In response to a TCP connection request 218 the tunnel server accepts the request and establishes a socket interface associated with the resulting TCP connection with Node C.

Upon establishment of the TCP connection with the last node in the tunnel path prior to the tunnel server, the tunnel server enters the Receive Relay state 219. In the Receive Relay state 219, the tunnel server waits to receive a relay frame at 220, at which time the tunnel server enters the Connect Wait state 221. If there is some sort of network error 234 during receipt of the relay frame at 219, the tunnel server enters the Dying state 230. During the Dying state 230 the tunnel server disconnects at 231 the transport layer connection with the last node in the tunnel path prior to the tunnel server. After disconnecting the connection, the tunnel server enters the Dead state 232.

In the Connect Wait state 221, the tunnel server waits for receipt of a key exchange/authentication REQUEST frame at 222. Following receipt of the key exchange/authentication REQUEST frame at 222, the tunnel server determines whether the requested tunnel connection is authorized at step 223. The determination of whether the tunnel connection is authorized is based on a name of the tunnel initiator, and the key exchange material within the key exchange/authentication REQUEST frame.

If the requested tunnel connection is authorized the tunnel server sends a key exchange/authentication RESPONSE frame at 224 back to the tunnel initiator. If the requested tunnel connection is not authorized, the tunnel server enters the Close state 228, in which it sends a close connection frame at 229 to the tunnel client. Following transmission of the CLOSE connection frame at 229, the tunnel server enters the Dying state 230.

If the requested tunnel connection is determined to be authorized at step 223, the tunnel server enters the Authorized state 225. In the Authorized state, the tunnel

server transmits and receives tunnel data frames between itself and the tunnel initiator. If during the Authorized state 225, the tunnel server receives a CLOSE connection frame at 233, the tunnel server transitions to the Dying state 230. If during the authorized state 225, the tunnel server receives an end session command from a user at 226, then the tunnel server transitions to the Close state 228, and transmits a close connection frame at 229 to the tunnel initiator. If the tunnel server in the Authorized state 225 detects an integrity failure in a received packet, the tunnel server transitions to the Close state 228. In the close state 228 the tunnel server sends a CLOSE connection frame at 229 and subsequently enters the Dying state 230.

Fig. 13 is a state diagram showing an example embodiment of a state machine within a tunnel relay node. The tunnel relay node begins in an Accept Wait state 235. When a request is received to form a reliable transport layer connection at 236, a reliable transport layer connection is accepted with the requesting node. For example, a TCP connection is accepted between the relay node and the preceding node in the tunnel path.

The relay node then transitions to the Receive Relay state 237. During the Receive Relay state 237, the relay node receives a relay frame at 238. Following receipt of the relay frame at 238, the relay node determines what forwarding address should be used to forward frames received from the TCP connection established responsive to the TCP connect event 236. If the next node in the tunnel path is a tunnel server, the forwarding address may be selected at 239 so as to choose an underutilized tunnel server from a group of available tunnel servers or to choose an operational server where others are not operational.

Following determination of the forwarding address or addresses in step 239, the relay node enters the Forward Connect state 240. In the Forward Connect state 240, the relay node establishes a reliable transport layer connection with the node or nodes indicated by the forwarding address or addresses determined in step 239.

Following establishment of the new connection at event 241, the tunnel relay enters the Forward state 242. During the Forward state 242, the relay node forwards all frames between the connection established at 236 and those connections established at 241. Upon detection of a network error or receipt of a frame indicating a closure of the tunnel connection at 243, the tunnel relay enters the Dying state 244. Following the Dying state 244, the relay node disconnects any connections established at event 241. The relay node then enters the Dead state 246.

Fig. 14 shows an example embodiment of a virtual private network 249 formed by a pseudo network adapter 248 and a tunnel connection between a tunnel client 247 and a tunnel server 253 across a public network 251. The tunnel server 253 and tunnel client 247 are for example network stations including a CPU or microprocessor, memory, and various I/O devices. The tunnel server 253 is shown physically connected to a private LAN 256 including a Network Node 1 257 and a Network Node 2 258, through a physical network adapter 254. The tunnel server 253 is further shown physically connected with a firewall 252 which separates the private LAN 256 from the public network 251. The firewall 252 is physically connected with the public network 251. The tunnel server 253 is further shown including a pseudo network adapter 255. The client system 247 is shown including a physical network adapter 250 physically connected to the public network 251.

During operation of the elements shown in Fig. 14, nodes within the virtual private network 249 appear to the tunnel client 247 as if they were physically connected to the client system through the pseudo network adapter 248. Data transmissions between the tunnel client and any nodes that appear to be within the virtual private network are passed through the pseudo network adapter 248. Data transmissions between the tunnel client 247 and the tunnel server 253 are physically accomplished using a tunnel connection between the tunnel client 247 and the tunnel server 253.

Fig. 15 shows elements in an example embodiment of a pseudo network adapter such as the pseudo network adapter 248 in Fig. 14. In an example embodiment the elements shown in Fig. 15 are implemented as software executing on the tunnel client 247 as shown in Fig. 14. In Fig. 15 there is shown a pseudo network adapter 259 including a virtual adapter driver interface 263, an encapsulation engine 264, an encryption engine 265, a decapsulation engine 268, and a decryption engine 266. Further shown in the pseudo network adapter 259 are an ARP server emulator 270 and a Dynamic Host Configuration Protocol (DHCP) server emulator.

The pseudo network adapter 259 is shown interfaced to a TCP/IP protocol stack 260, through the virtual adapter driver interface 260. The TCP/IP protocol stack 260 is shown interfaced to other services in an operating system 261, as well as a physical network adapter's driver 262. The physical network adapter's driver 262 is for example a device driver which controls the operation of a physical network adapter such as physical network adapter 250 as shown in Fig. 14.

During operation of the elements shown in Fig. 15, the pseudo network adapter 259 registers with the network layer in the TCP/IP stack 260 that it is able to reach the IP addresses of nodes within the virtual private network 249 as shown in Fig. 14. For example, the pseudo network adapter on the client system registers that it can reach the pseudo network adapter on the server. Subsequently, a message from the tunnel client addressed to a node reachable through the virtual private network will be passed by the TCP/IP stack to the pseudo network adapter 259. The pseudo network adapter 259 then encrypts the message, and encapsulates the message into a tunnel data frame. The pseudo network adapter 259 then passes the tunnel data frame

back to the TCP/IP protocol stack 260 to be sent through to the physical network adapter in the tunnel server. The tunnel server passes the received data frame to the pseudo network adapter in the server, which de-encapsulates and decrypts the message.

Fig. 16 shows a more detailed example embodiment of a pseudo network adapter 280. The pseudo network adapter 280 includes a virtual network adapter driver interface 288. The transmit path 290 includes an encryption engine 292, and an encapsulation engine 294. The encapsulation engine 294 is interfaced with a TCP/IP transmit interface 312 within a TCP/IP protocol stack, for example a socket interface associated with the first relay node in the tunnel path, or with the remote tunnel end point if the tunnel path includes no relays.

In the example embodiment of Fig. 16, the pseudo network adapter 280 appears to the TCP/IP protocol stack 282 as an Ethernet adapter. Accordingly, ethernet packets 286 for a destination addresses understood by the TCP/IP protocol stack to be reachable through the virtual private network are passed from the TCP/IP protocol stack 282 to the virtual network adapter interface 288 and through the transmit path 290. Similarly, ethernet packets 284 received through the pseudo network adapter 280 are passed from the receive path 296 to the virtual network adapter interface 288 and on to the TCP/IP protocol stack 282.

Further shown in the pseudo network adapter 280 of Fig. 16 is a receive path 296 having a decryption engine 298 interfaced to the virtual network adapter interface 288 and a decapsulation engine 300. The decapsulation engine 300 in turn is interfaced to a TCP/IP receive function 314 in the TCP/IP protocol stack 282, for example a socket interface associated with the first relay in the tunnel path, or with the remote tunnel end point if the tunnel path includes no relays. The pseudo network adapter 280 further includes an ARP server emulator 304 and a DHCP server emulator 306. ARP and DHCP request packets 302 are passed to the ARP server emulator 304 and DHCP server emulator 306 respectively. When a received packet is passed from the receive path 296 to the TCP/IP stack 282, a receive event must be indicated to the TCP/IP stack 282, for example through an interface such the Network Device Interface Specification (NDIS), defined by Microsoft™ Corporation.

Also in Fig. 16 is shown is an operating system 310 coupled with the TCP/IP protocol stack 282. The TCP/IP protocol stack 282 is generally considered to be a component part of the operating system. The operating system 310 in Fig. 16 is accordingly the remaining operating system functions and procedures outside the TCP/IP protocol stack 282. A physical network adapter 308 is further shown operated by the TCP/IP protocol stack 282.

During operation of the elements shown in Fig. 16, a user passes data for transmission to the TCP/IP protocol stack 282, and indicates the IP address of the node to which the message is to be transmitted, for example through a socket interface to the TCP layer. The TCP/IP protocol stack 282 then determines whether the destination node is reachable through the virtual private network. If the message is for a node that is reachable through the virtual private network, the TCP/IP protocol stack 282 an ethernet packet 286 corresponding to the message to the pseudo network adapter 280. The pseudo network adapter 280 then passes the ethernet packet 286 through the transmit path, in which the ethernet packet is encrypted and encapsulated into a tunnel data frame. The tunnel data frame is passed back into the TCP/IP protocol stack 282 through the TCP/IP transmit function 312 to be transmitted to the tunnel server through the tunnel connection. In an example embodiment, a digest value is calculated for the tunnel data frame before encryption within the transmit path within the pseudo network adapter.

Further during operation of the elements shown in Fig. 16, when the TCP/IP protocol stack 282 receives a packet from the remote endpoint of the TCP/IP tunnel connection, for example the tunnel server, the packet is passed to the pseudo network adapter 280 responsive to a TCP receive event. The pseudo network adapter 280 then decapsulates the packet by removing the tunnel header. The pseudo network adapter further decrypts the decapsulated data and passes it back to the TCP/IP protocol stack 282. The data passed from the pseudo network adapter 280 appears to the TCP/IP protocol stack 282 as an ethernet packet received from an actual physical device, and is the data it contains is passed on to the appropriate user by the TCP/IP protocol stack 282 based on information in the ethernet packet header provided by the pseudo network adapter.

Fig. 17 is a flow chart showing steps performed by an example embodiment of a pseudo network adapter during packet transmission, such as in the transmit path 290 of Fig. 14. The TCP/IP protocol stack determines that the destination node of a packet to be transmitted is reachable through the virtual LAN based on the destination IP address of the packet and a network layer routing table. At step 320 the packet is passed to the pseudo network adapter from the TCP/IP protocol stack. As a result, a send routine in the pseudo adapter is triggered for example in the virtual network adapter interface 288 of Fig. 16.

At step 322 the pseudo network adapter send routine processes the Ethernet header of the packet provided by the TCP/IP stack, and removes it. At step 324, the send routine determines whether the packet is an ARP request packet. If the packet is an ARP request packet for an IP address of a node on the virtual LAN, such as the pseudo network adapter of the tunnel server, then step 324 is followed by step 326. Otherwise, step 324 is followed by step 330.

At step 326, the ARP server emulator in the pseudo network adapter generates an ARP reply packet. For example, if the ARP request were for a physical address

corresponding to the IP address of the pseudo network adapter on the tunnel server, the ARP reply would indicate a predetermined, reserved physical address to be associated with that IP address. At step 328 the pseudo network adapter passes the ARP response to the virtual network adapter interface. The virtual network adapter interface then indicates a received packet to the TCP/IP protocol stack, for example using an NDIS interface. The TCP/IP protocol stack then processes the ARP response as if it had been received over an actual physical network.

At step 330 the send routine determines whether the packet is a DHCP request packet requesting an IP address for the pseudo network adapter. If so, then step 330 is followed by step 332. Otherwise, step 330 is followed by step 334.

At step 334, the DHCP server emulator in the pseudo network adapter generates a DHCP response. The format of DHCP is generally described in the DHCP RFC. At step 328 the pseudo network adapter passes the DHCP response to the virtual network adapter interface, for example indicating an IP address received from the tunnel server in the client data field of the key exchange/authentication RESPONSE frame. The virtual network adapter interface then indicates a received packet to the TCP/IP protocol stack. The TCP/IP protocol stack then processes the DHCP response as if it had been received over an actual physical network.

At step 334 the pseudo network adapter encrypts the message using an encryption engine such that only the receiver is capable of decrypting and reading the message. At step 336 the pseudo network adapter encapsulates the encrypted message into a tunnel data frame. At step 338 the pseudo network adapter transmits the tunnel data frame through the tunnel connection using the TCP/IP protocol stack.

Fig. 18 is a flow chart showing steps performed by an example embodiment of a pseudo network adapter during packet receipt, such as in the receive path 296 of Fig. 14.

At step 350, the pseudo network adapter is notified that a packet has been received over the tunnel connection. At step 352 the pseudo network adapter decapsulates the received message by removing the header fields of the tunnel data frame. At step 354 the pseudo network adapter decrypts the decapsulated datagram from the tunnel data frame. At step 356, in an example embodiment, the pseudo network adapter forms an Ethernet packet from the decapsulated message. At step 358 the pseudo network adapter indicates that an Ethernet packet has been received to the TCP/IP protocol stack through the virtual network adapter interface. This causes the TCP/IP protocol stack to behave as if it had received an Ethernet packet from an actual Ethernet adapter.

Fig. 19 shows the data flow within the transmit path in an example embodiment of a pseudo network adapter. At step 1 370, an application submits data to be transmitted to the TCP protocol layer 372 within the TCP/IP protocol stack. The application uses a conventional socket interface to the TCP protocol layer 372 to pass the data, and indicates the destination IP address the data is to be transmitted to. The TCP protocol layer 372 then passes the data to the IP protocol layer 374 within the TCP/IP protocol stack. At step 2 376, the TCP/IP protocol stack refers to the routing table 378 to determine which network interface should be used to reach the destination IP address.

Because in the example the destination IP address is of a node reachable through the virtual private network, the IP layer 374 determines from the routing table 378 that the destination IP address is reachable through pseudo network adapter. Accordingly at step 3 380 the TCP/IP protocol stack passes a packet containing the data to the pseudo network adapter 382.

At step 4 384, the pseudo network adapter 382 encrypts the data packets and encapsulates them into tunnel data frames.

The pseudo network adapter 382 then passes the tunnel data frames packets back to the TCP protocol layer 372 within the TCP/IP protocol stack through a conventional socket interface to the tunnel connection with the first node in the tunnel path.

The TCP protocol layer 372 then forms a TCP layer packet for each tunnel data frame, having the tunnel data frame as its data. The TCP frames are passed to the IP layer 374. At step 5 386 the routing table 378 is again searched, and this time the destination IP address is the IP address associated with the physical network adapter on the tunnel server, and accordingly is determined to be reachable over the physical network adapter 390. Accordingly at step 6 388 the device driver 390 for the physical network adapter is called to pass the packets to the physical network adapter. At step 7 392 the physical network adapter transmits the data onto the physical network 394.

Fig. 20 is a data flow diagram showing data flow in an example embodiment of packet receipt involving a pseudo network adapter. At step 1 410 data arrives over the physical network 412 and is received by the physical network adapter and passed to the physical network driver 414. The physical network driver 414 passes the data at step 2 418 through the IP layer 420 and TCP layer 422 to the pseudo network adapter 426 at step 3 424, for example through a conventional socket interface. At step 4 428 the pseudo network adapter 426 decrypts and decapsulates the received data and passes it back to the IP layer of the TCP/IP protocol stack, for example through the TDI (Transport Layer Dependent Interface API) of the TCP/IP stack. The data is then passed through the TCP/IP protocol stack and to the user associated with the destination IP address in the decapsulated datagrams at step 5 430.

Fig. 21 shows data flow in an example embodiment of packet transmission involving a pseudo network adapter. Fig. 21 shows an example embodiment for use

on a Microsoft™ Windows 95™ PC platform. In Fig. 21 a user application 450 passes unencrypted data to an interface into the TCP layer of the TCP/IP protocol, for example the WinSock API 452. The user indicates a destination IP address associated with a node reachable through a virtual private network accessible through the pseudo network adapter.

The TCP layer 454 passes the data to the IP layer 456, which in turn passes the data to the Network Device Interface Specification Media Access Control (NDIS MAC) interface 458. The pseudo network adapter 459 has previously registered with the routing layer (IP) that it is able to reach a gateway address associated with the destination IP address for the user data. Accordingly the IP layer uses the NDIS MAC layer interface to invoke the virtual device driver interface 460 to the pseudo network adapter 459. The pseudo network adapter 459 includes a virtual device driver interface 460, an ARP server emulator 462, and a DHCP server emulator 464.

In the example embodiment of Fig. 19, the pseudo network adapter 459 passes the data to a tunnel application program 466. The tunnel application program 466 encrypts the IP packet received from the IP layer and encapsulates it into a tunnel data frame. The tunnel application then passes the tunnel data frame including the encrypted data to the WinSock interface 452, indicating a destination IP address of the remote tunnel end point. The tunnel data frame is then passed through the TCP layer 454, IP layer 456, NDIS MAC layer interface 458, and physical layer 468, and transmitted on the network 470. Since the resulting packets do not contain a destination IP address which the pseudo network adapter has registered to convey, these packets will not be diverted to the pseudo network adapter.

Fig. 22 is a data flow diagram showing data flow in an example embodiment of packet transmission involving a pseudo network adapter. The embodiment shown in Fig. 22 is for use on a UNIX platform. In Fig. 20 a user application 472 passes unencrypted data to a socket interface to the TCP/IP protocol stack in the UNIX socket layer 474, indicating a destination IP address of a node reachable through the virtual private network.

The UNIX socket layer 474 passes the data through the TCP layer 476 and the IP layer 478. The pseudo network adapter 480 has previously registered with the routing layer (IP) that it is able to reach a gateway associated with the destination IP address for the user data. Accordingly the IP layer 478 invokes the virtual device driver interface 482 to the pseudo network adapter 480. The IP layer 478 passes the data to the pseudo network adapter 480. The pseudo network adapter 480 includes a virtual device driver interface 482, and a DHCP server emulator 484.

In the example embodiment of Fig. 22, the pseudo network adapter 480 passes IP datagrams to be transmitted to a UNIX Daemon 486 associated with the tunnel connection. The UNIX Daemon 486 encrypts the IP packet(s) received from the IP layer 478 and encapsulates them into tunnel data frames. The UNIX Daemon 486 then passes the tunnel data frames to the UNIX socket layer 474, through a socket associated with the tunnel connection. The tunnel data frames are then processed by the TCP layer 476, IP layer 478, data link layer 488, and physical layer 490 to be transmitted on the network 492. Since the resulting packets are not addressed to an IP address which the pseudo network adapter 480 has registered to convey, the packets will not be diverted to the pseudo network adapter 480.

Fig. 23 is a flow chart showing steps to initialize a example embodiment of a virtual private network. The steps shown in Fig. 23 are performed for example in the tunnel client 247 as shown in Fig. 14. At step 500 a tunnel application program executing in the tunnel client sends a tunnel relay frame to the tunnel server. At step 502 the tunnel application program sends a tunnel key exchange/authentication REQUEST frame to the tunnel server. The tunnel application in the tunnel server ignores the contents of the client data field in the tunnel key exchange/authentication REQUEST frame. The tunnel application in the tunnel server fills in the client data field in the tunnel key exchange/authentication RESPONSE frame with Dynamic Host Configuration Protocol (DHCP) information, for example including the following information in standard DHCP format:

1) IP Address for tunnel client Pseudo Network Adapter
2) IP Address for tunnel server Pseudo Network Adapter
3) Routes to nodes on the private network physically connected to the tunnel server which are to be reachable over the tunnel connection.

At step 504 the tunnel application receives a tunnel key exchange/authentication RESPONSE frame from the tunnel server. The client data field 508 in the tunnel connection response is made available to the pseudo network adapter in the tunnel client. The tunnel application in the tunnel client tells the TCP/IP stack that the pseudo network adapter in the tunnel client is active. The pseudo network adapter in the tunnel client is active and ready to be initialized at step 510.

The tunnel client system is configured such that it must obtain an IP address for the tunnel client pseudo network adapter dynamically. Therefore the TCP/IP stack in the tunnel client broadcasts a DHCP request packet through the pseudo network adapter. Accordingly, at step 512 the pseudo network adapter in the client receives a conventional DHCP request packet from the TCP/IP stack requesting a dynamically allocated IP address to associate with the pseudo network adapter. The pseudo network adapter passes the DHCP request packet to the DHCP server emulator within the pseudo network adapter, which forms a DHCP response based on the client data 508 received from the tunnel applica-

tion. The DHCP response includes the IP address for the client pseudo adapter provided by the tunnel server in the client data. At step 514 the pseudo network adapter passes the DHCP response to the TCP/IP stack.

At step 520, the tunnel application modifies the routing tables within the tunnel client TCP/IP stack to indicate that the routes to the nodes attached to the private network to which the tunnel server is attached all are reachable only through the pseudo network adapter in the tunnel server. The IP address of the pseudo network adapter in the tunnel server provided in the client data is in this way specified as a gateway to the nodes on the private network to which the tunnel server is attached. In this way those remote nodes are viewed by the TCP/IP stack as being reachable via the virtual private network through the client pseudo network adapter.

At step 516 the pseudo network adapter in the tunnel client receives an ARP request for a physical address associated with the IP address of the pseudo network adapter in the tunnel server. The pseudo network adapter passes the ARP request to the ARP server emulator, which forms an ARP reply indicating a reserved physical address to be associated with the IP address of the pseudo network adapter in the tunnel server. At step 518 the pseudo network adapter passes the ARP response to the TCP/IP stack in the tunnel client. In response to the ARP response, the TCP/IP stack determines that packets addressed to any node on the virtual private network must be initially transmitted through the pseudo network adapter.

In an example embodiment the present system reserves two physical addresses to be associated with the pseudo network adapter in the client and the pseudo network adapter in the server respectively. These reserved physical addresses are used in responses to ARP requests passed through the pseudo network adapter for physical addresses corresponding to the IP addresses for the pseudo network adapter in the client and the pseudo network adapter in the server respectively. The reserved physical addresses should have a high likelihood of not being used in any actual network interface.

While the invention has been described with reference to specific example embodiments, the description is not meant to be construed in a limiting sense. Various modifications of the disclosed embodiments, as well as other embodiments of the invention, will be apparent to persons skilled in the art upon reference to this description. Specifically, while various embodiments have been described using the TCP/IP protocol stack, the invention may advantageously be applied where other communications protocols are used. Also, while various flow charts have shown steps performed in an example order, various implementations may use altered orders of step in order to apply the invention. And further, while certain specific software and/or hardware platforms have been used in the description, the invention may be applied on other platforms with similar advantage. It is therefore contemplated that the appended claims will cover any such modifications or embodiments which fall within the scope of the invention.

Claims

1. A pseudo network adapter providing a virtual private network, comprising:

   an interface for capturing packets from a local communications protocol stack for transmission on said virtual private network, said interface appearing to said local communications protocol stack as a network adapter device driver for a network adapter connected to said virtual private network;
   a first server emulator, providing a first reply packet responsive to a first request packet captured by said interface for capturing packets from said local communications protocol stack for transmission on said virtual private network, said first request packet requesting a network layer address for said pseudo network adapter, said first reply indicating a network layer address for said pseudo network adapter; and
   a second server emulator, providing a second reply packet responsive to an second request packet captured by said interface for capturing packets from said local communications protocol stack for transmission on said virtual private network, said second request packet requesting a physical address corresponding to a network layer address of a second pseudo network adapter, said second pseudo network adapter located on a remote server node, said second reply indicating a predetermined, reserved physical address.

2. The pseudo network adapter of claim 1, further comprising a means for indicating to said local communications protocol stack that said predetermined, reserved physical address is reachable through said pseudo network adapter, wherein said means for indicating modifies a data structure in said local communications protocol stack indicating which nodes or networks are reachable through each network interface of the local system.

3. The pseudo network adapter of claim 1, further comprising a means for indicating to said local communications protocol stack that one or more nodes on a remote private network connected to said remote server node are reachable through a gateway node equal to said second pseudo network adapter on said remote server node.

4. The pseudo network adapter of claim 1, further comprising:

a transmit path for processing data packets captured by said interface for capturing packets from said local communications protocol stack for transmission on said virtual private network;
an encryption engine, within said transmit path, for encrypting said data packets;
an encapsulation engine, within said transmit path, for encapsulating said encrypted data packets into tunnel data frames; and
a means for passing said tunnel data frames back to said local communications protocol stack for transmission to a physical network adapter on said remote server node.

5. The pseudo network adapter of claim 4, wherein said transmit path further includes means for storing a digest value in a digest field in each of said tunnel data frames, said digest value equal to an output of a keyed hash function applied to said data packet encapsulated within said tunnel data frame concatenated with a counter value equal to a total number of tunnel data frames previously transmitted to said remote server node.

6. The pseudo network adapter of claim 4, wherein said transmit path further includes means for processing an Ethernet header in each one of said captured data packets, said processing of said Ethernet header including removing said Ethernet header.

7. The pseudo network adapter of claim 1, further comprising:

an interface into a transport layer of said local communications protocol stack for capturing received data packets from said remote server node.

8. The pseudo network adapter of claim 7, further comprising:

a receive path for processing received data packets captured by said interface into said transport layer of said local communications protocol stack for capturing received data packets from said remote server node;
an decapsulation engine, within said receive path, for decapsulating said received data packets by removing a tunnel frame header;
an decryption engine, within said receive path, for decrypting said received data packets; and
a means for passing said received data packets back to said local communications protocol stack for delivery to a user.

9. A method for providing a pseudo network adapter for a virtual private network, comprising the steps of:

capturing packets from a local communications protocol stack for transmission on said virtual private network, said capturing through an interface appearing to said local communications stack as a network adapter device driver for a network adapter connected to said virtual private network;
issuing a first reply packet responsive to a first request packet captured by said interface for capturing packets from said local communications protocol stack for transmission on said virtual private network, said first request packet requesting a network layer address for said pseudo network adapter, said first reply indicating a network layer address for said pseudo network adapter; and
issuing a second reply packet responsive to a second request packet captured by said interface for capturing packets from said local communications protocol stack for transmission on said virtual private network, said second request packet requesting a physical address corresponding to a network layer address of a second pseudo network adapter, said second pseudo network adapter located on a remote server node, said ARP Reply indicating a predetermined, reserved physical address.

10. The method of claim 9, further comprising indicating to said local communications protocol stack that said predetermined, reserved physical address is reachable through said pseudo network adapter, wherein said step of indicating to said local communications protocol stack modifies a data structure in said local communications protocol stack indicating which nodes or networks are reachable through each network interface of the local system.

11. The method of claim 9, further comprising indicating to said local communications protocol stack that one or more nodes on a remote private network connected to said remote server node are reachable through a gateway node equal to said second pseudo network adapter on said remote server node, wherein said step of indicating to said local communications protocol stack that one or more nodes on said remote private network connected to said remote server node are reachable through a gateway node equal to said second pseudo network adapter on said remote server node modifies a network layer routing table in said local communications protocol stack.

12. The method of claim 9, further comprising:

processing data packets captured by said interface for capturing packets from said local communications protocol stack for transmission on said virtual private network in a transmit data path;

encrypting said data packets in an encryption engine, within said transmit path;

encapsulating said encrypted data packets into tunnel data frames by an encapsulation engine, within said transmit path; and

passing said tunnel data frames back to said local communications protocol stack for transmission to a physical network adapter on said remote server node, wherein said transmit path further includes storing a digest value in a digest field in each of said tunnel data frames, said digest value equal to an output of a keyed hash function applied to said data packet encapsulated within said tunnel data frame concatenated with a counter value equal to a total number of tunnel data frames previously transmitted to said remote server node.

13. The method of claim 12, wherein said transmit path further includes processing an Ethernet header in each one of said captured data packets, said processing of said Ethernet header including removing said Ethernet header.

14. The method of claim 9, further comprising capturing received data packets from said remote server node through an interface into a transport layer of said local communications protocol stack, further comprising:

processing received data packets captured by said interface into said transport layer of said local communications protocol stack for capturing received data packets from said remote server node in a receive path;

decapsulating said received data packets by removing a tunnel frame header in an decapsulation engine, within said receive path;

decrypting said received data packets in a decryption engine within said receive path; and

passing said received data frames packets back to said local communications protocol stack for delivery to a user.

15. The method of claim 9, wherein said network layer address for said pseudo network adapter and said predetermined, reserved physical address is communicated to said pseudo network adapter from said remote server node as client data in a connection response frame.

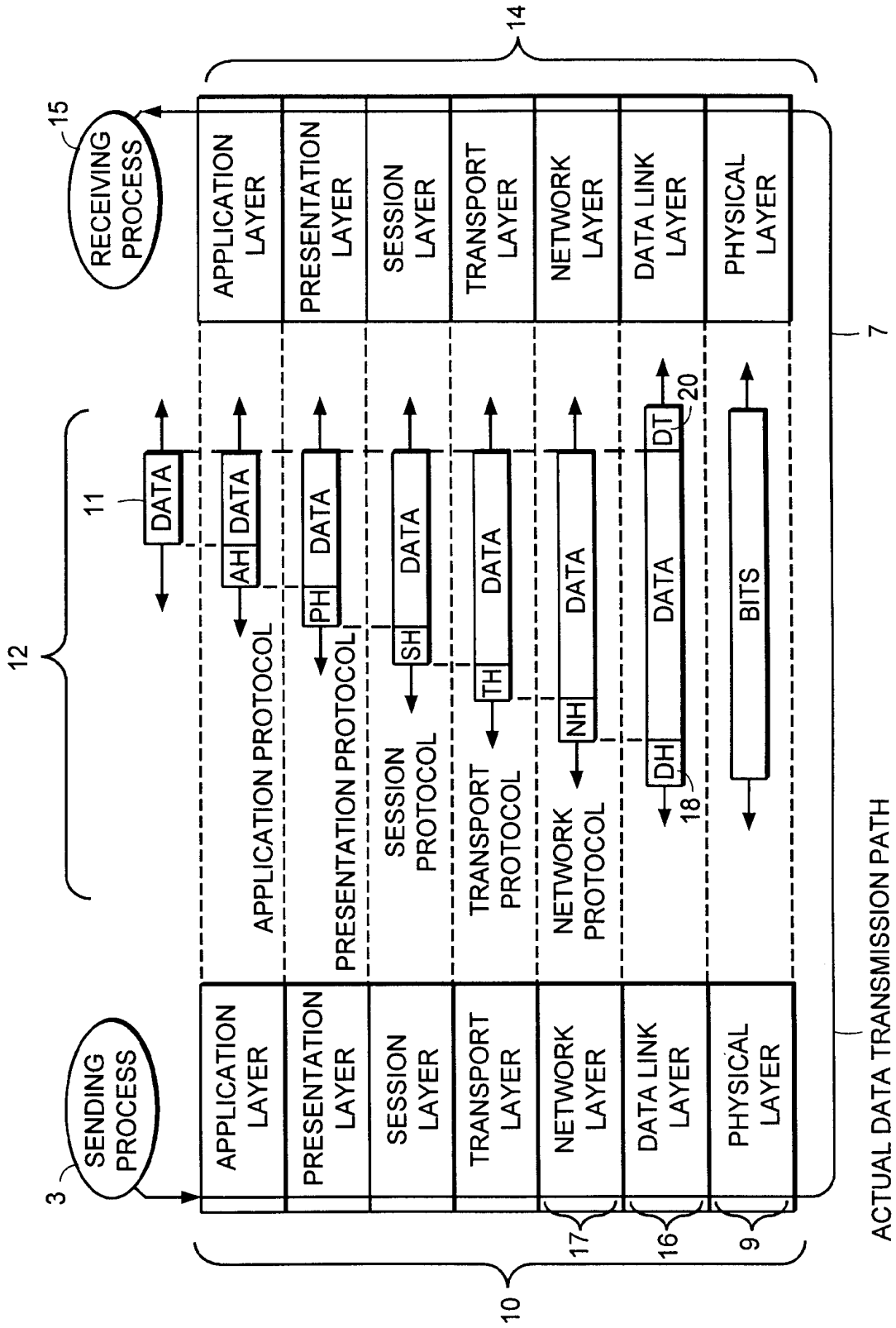


FIG. 1

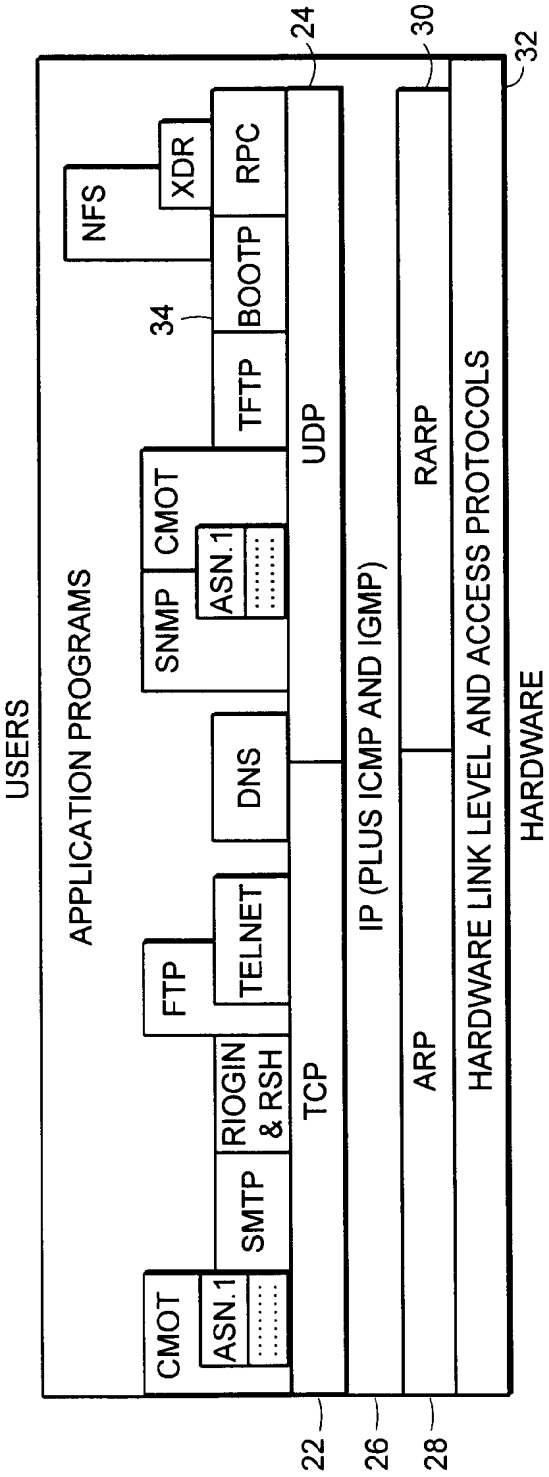


FIG. 2

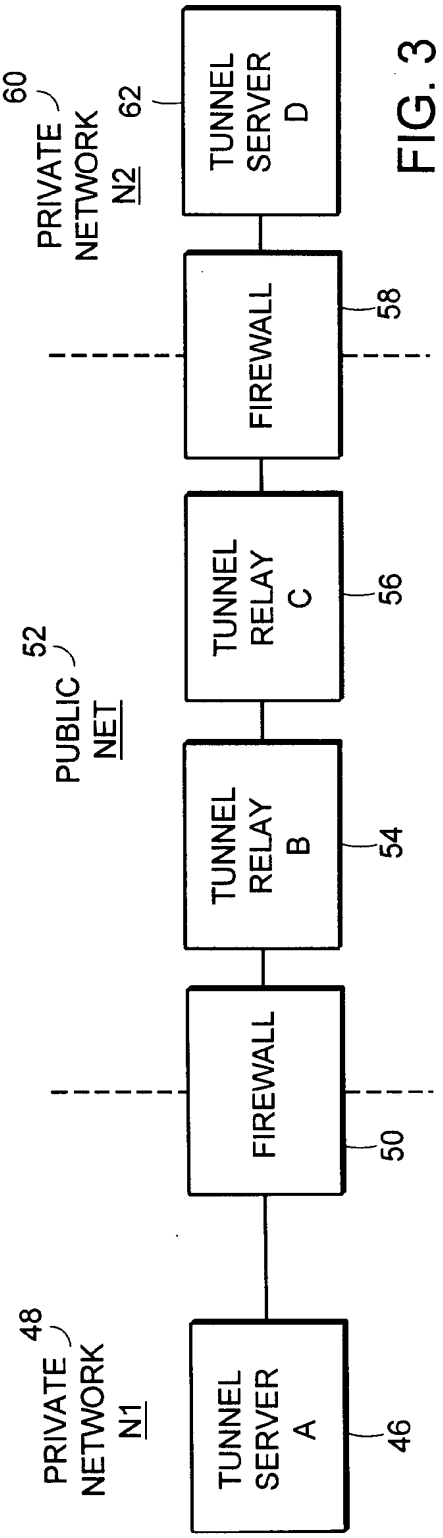


FIG. 3

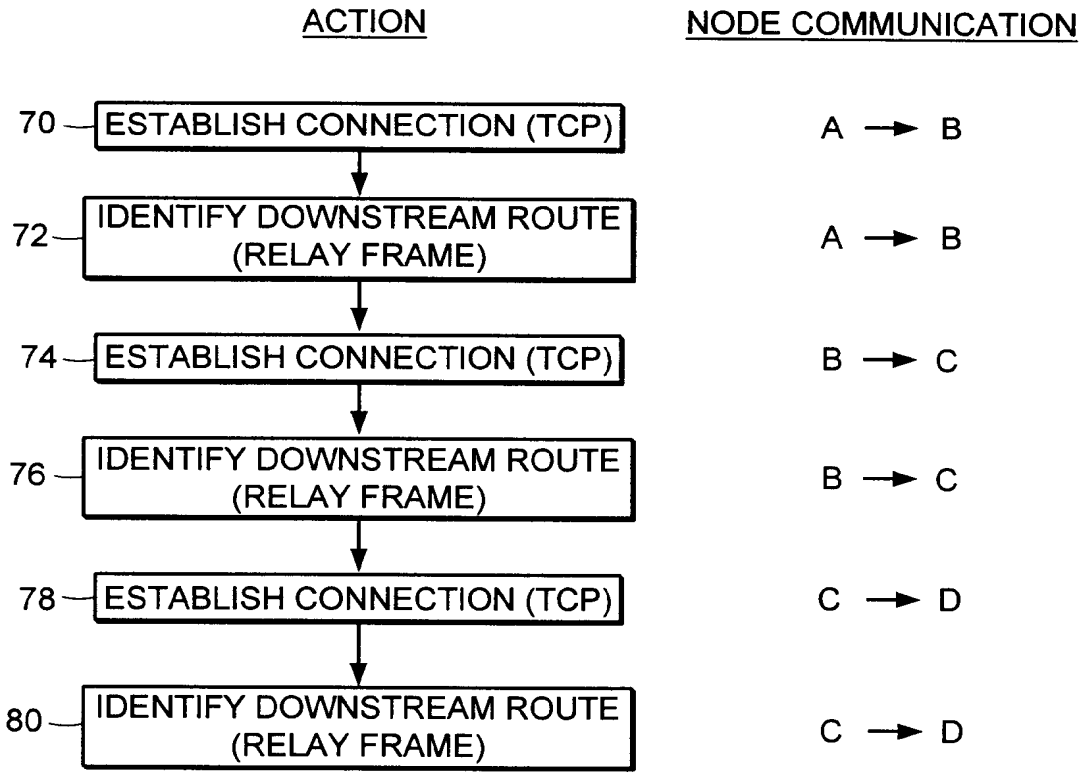


FIG. 4

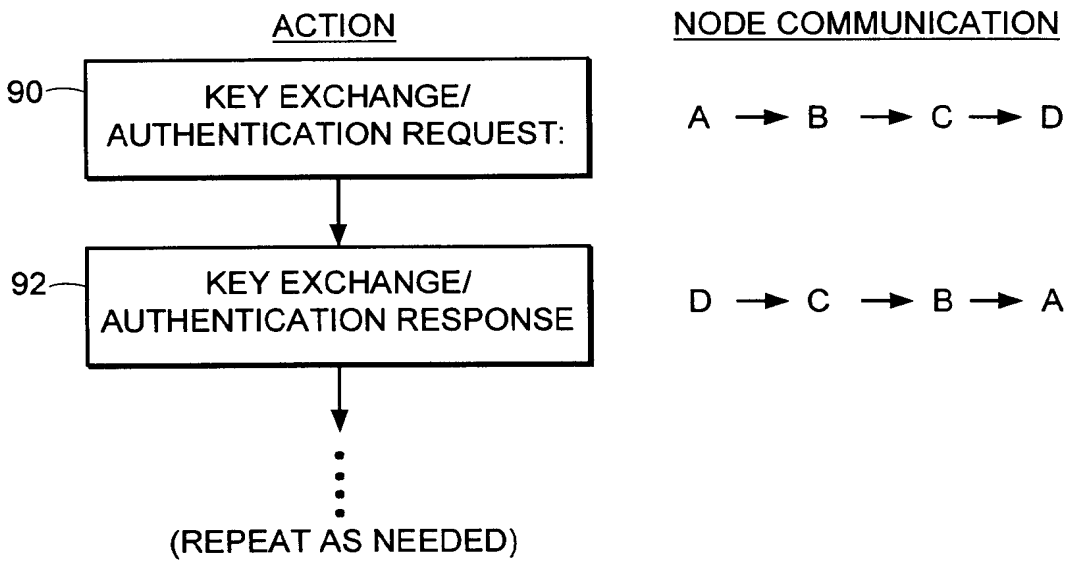


FIG. 5

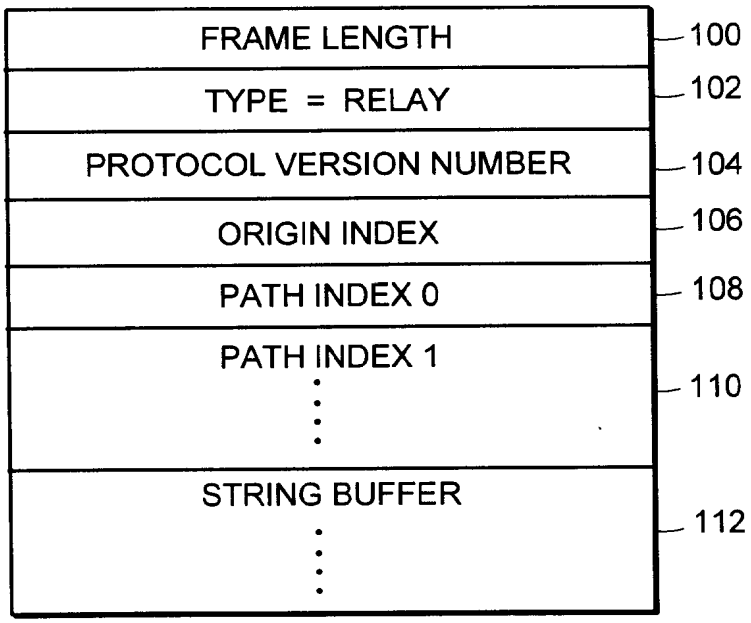


FIG. 6

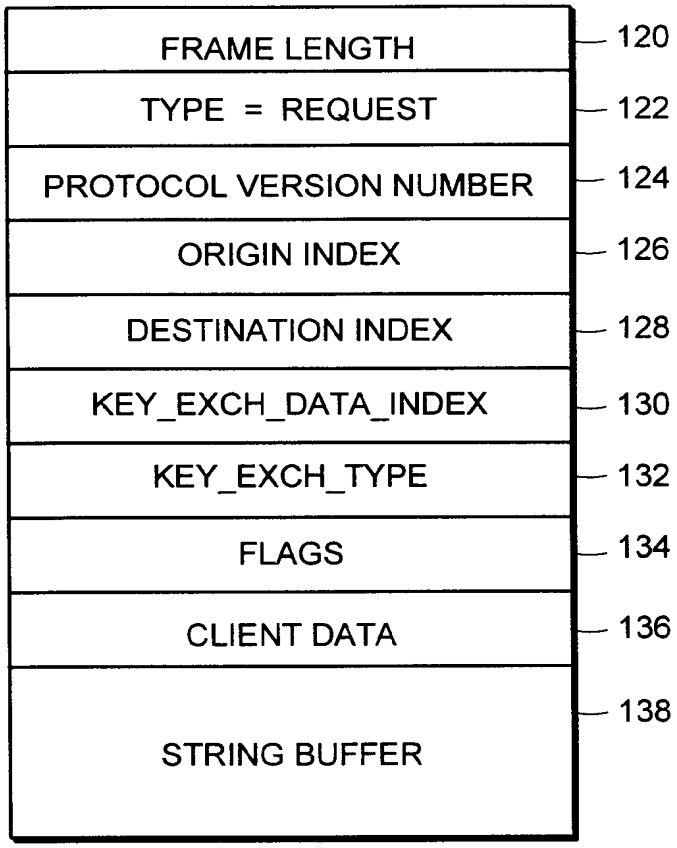


FIG. 7

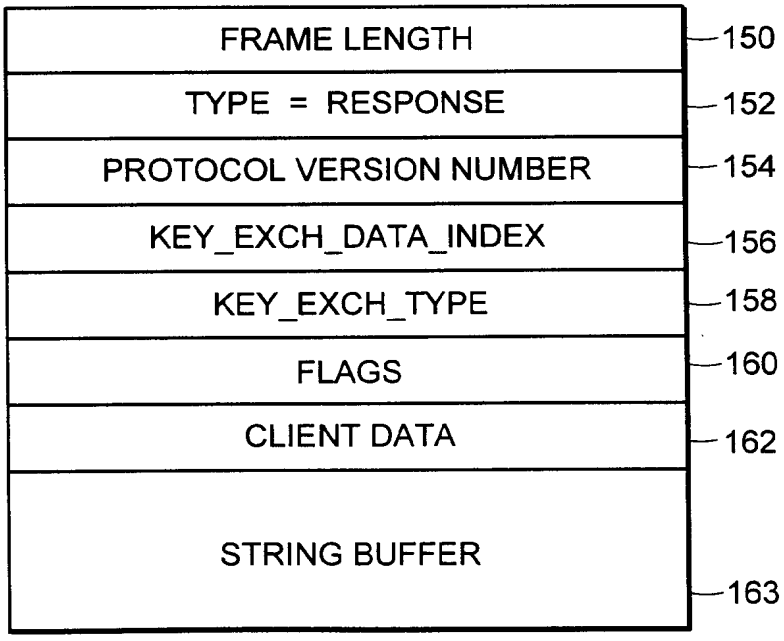


FIG. 8

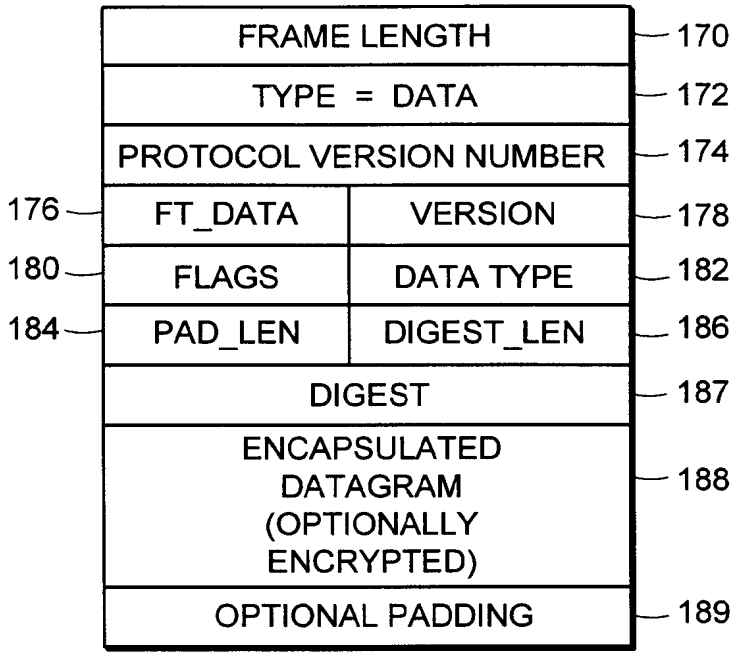


FIG. 9

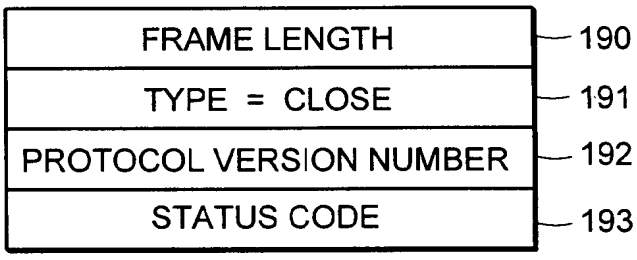


FIG. 10

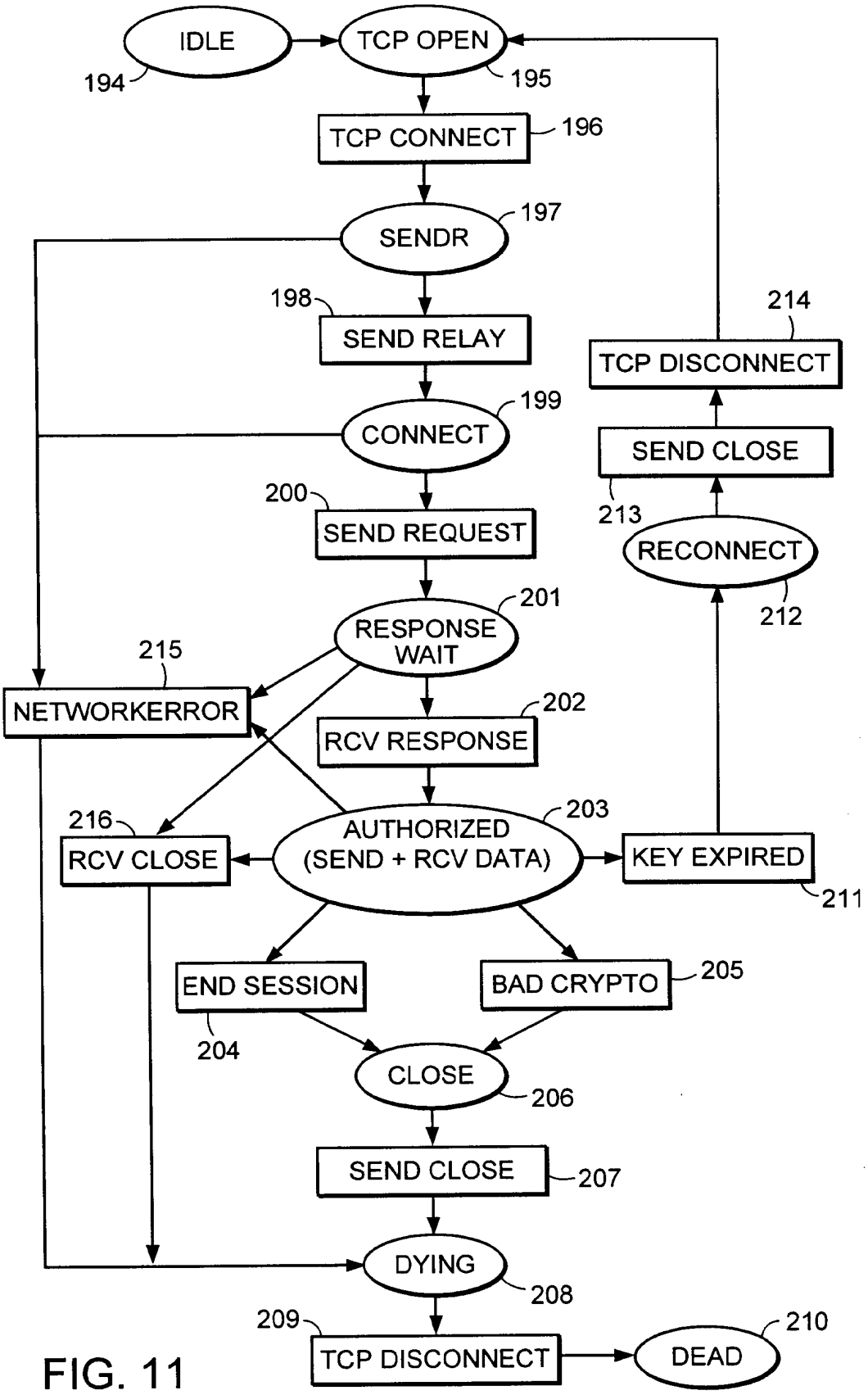


FIG. 11

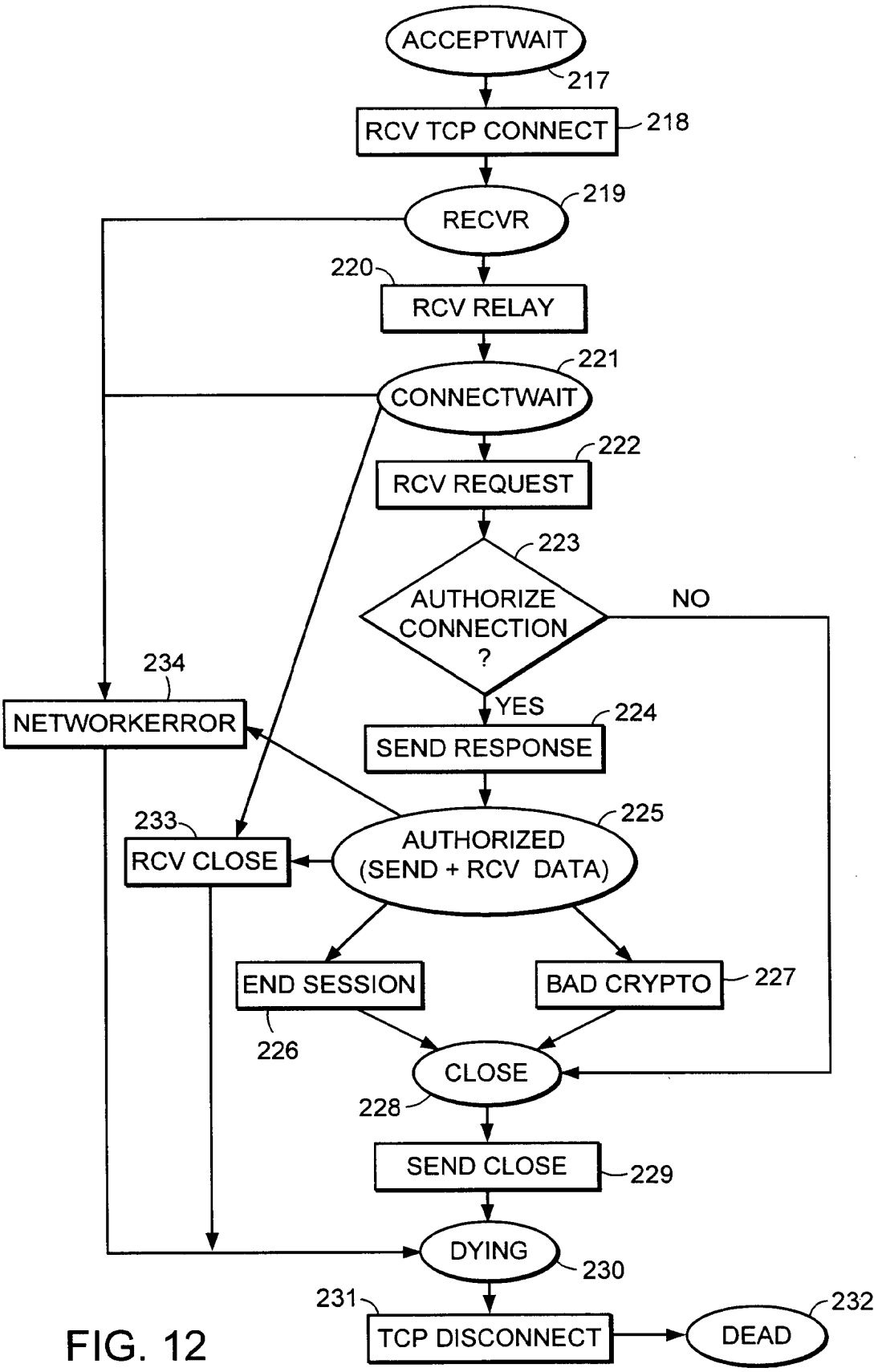


FIG. 12

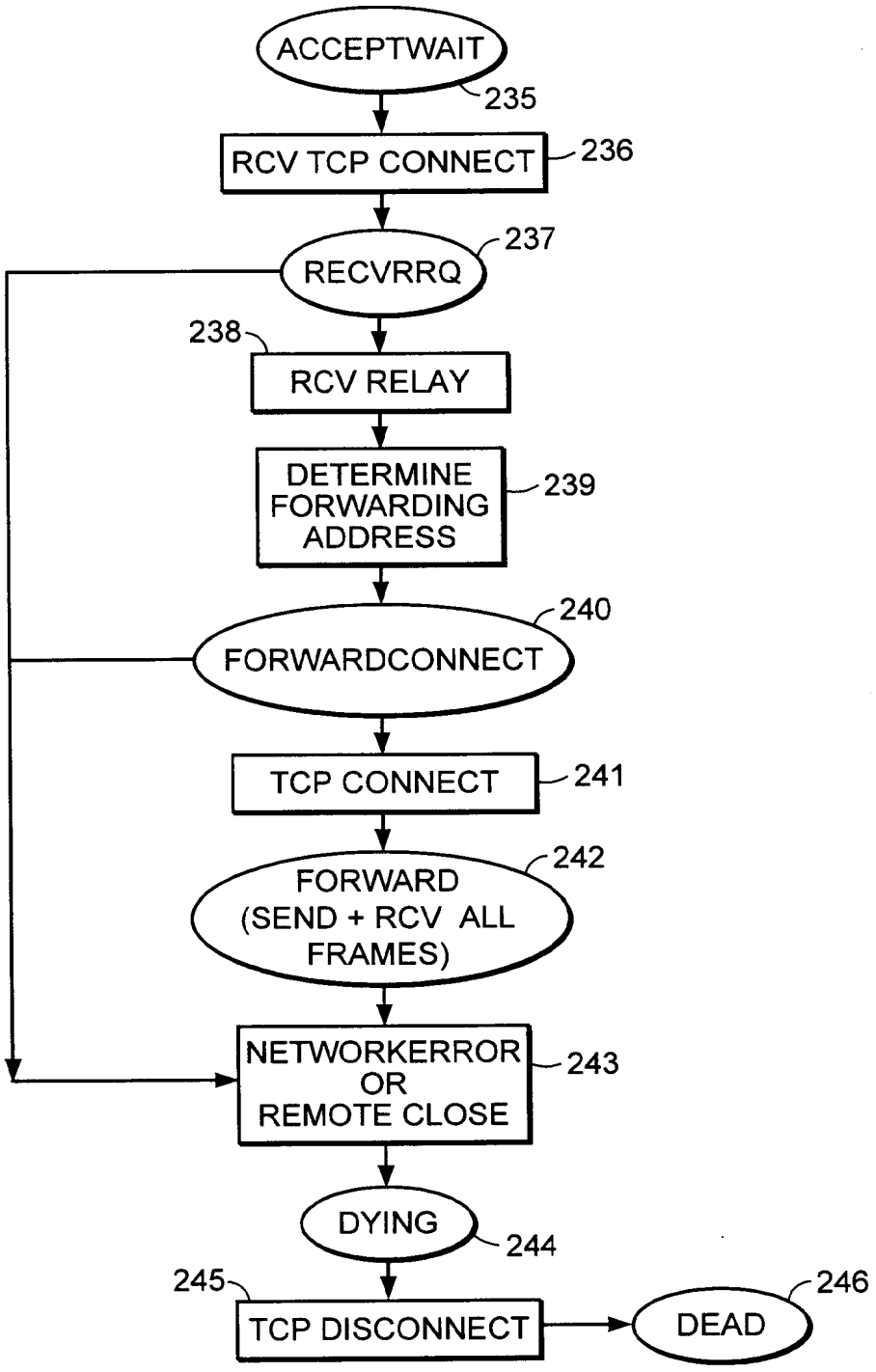


FIG. 13

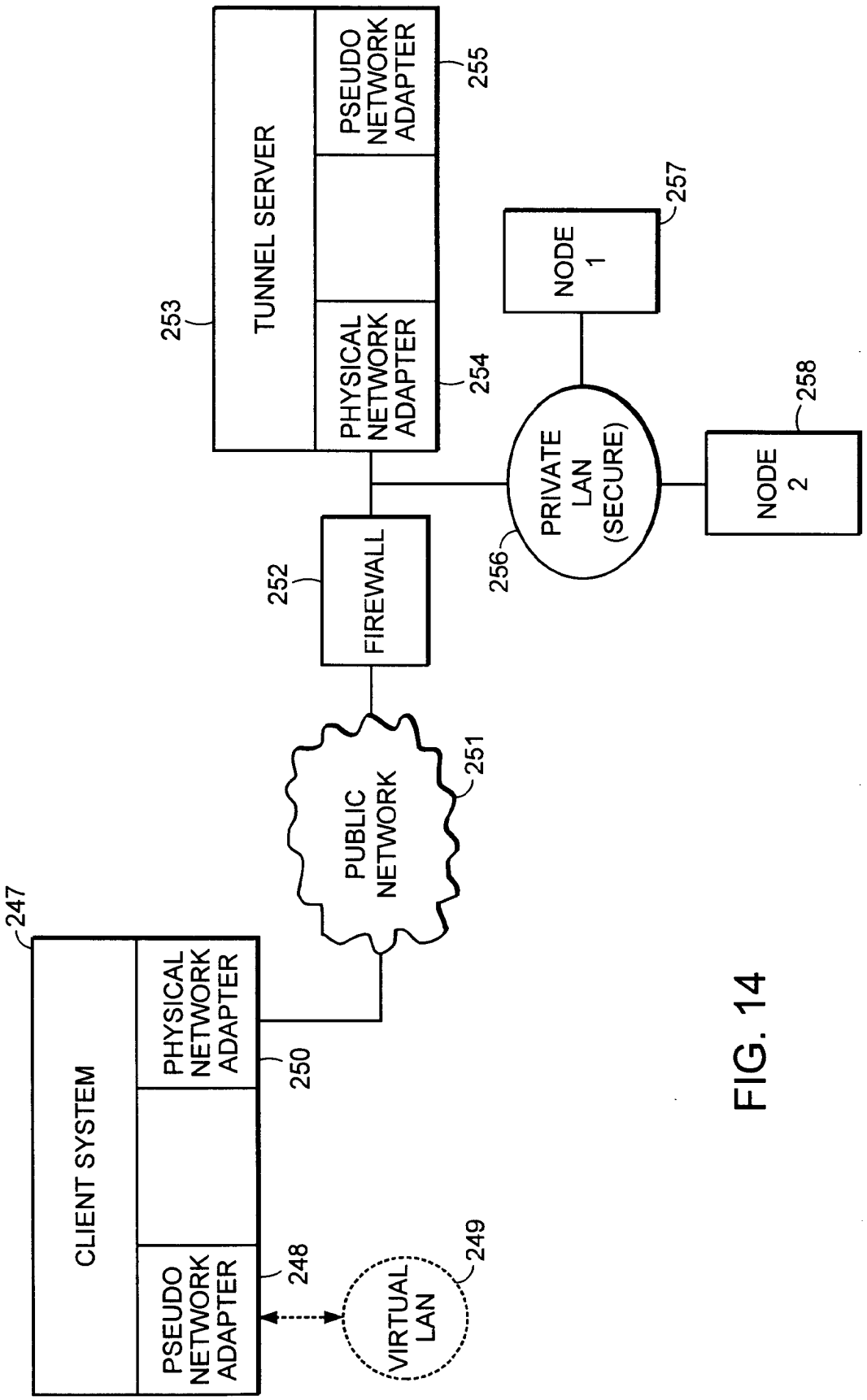


FIG. 14

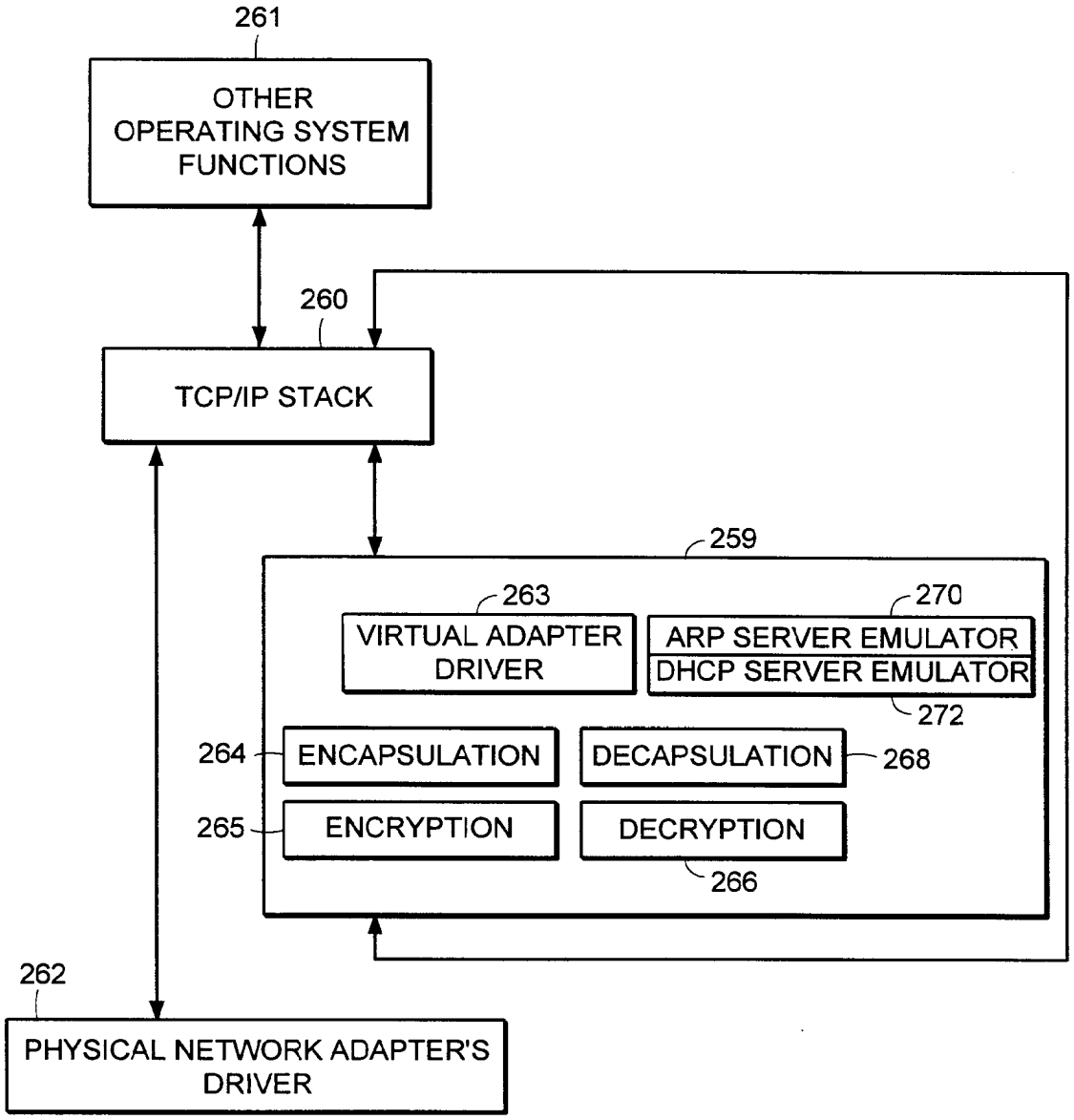


FIG. 15

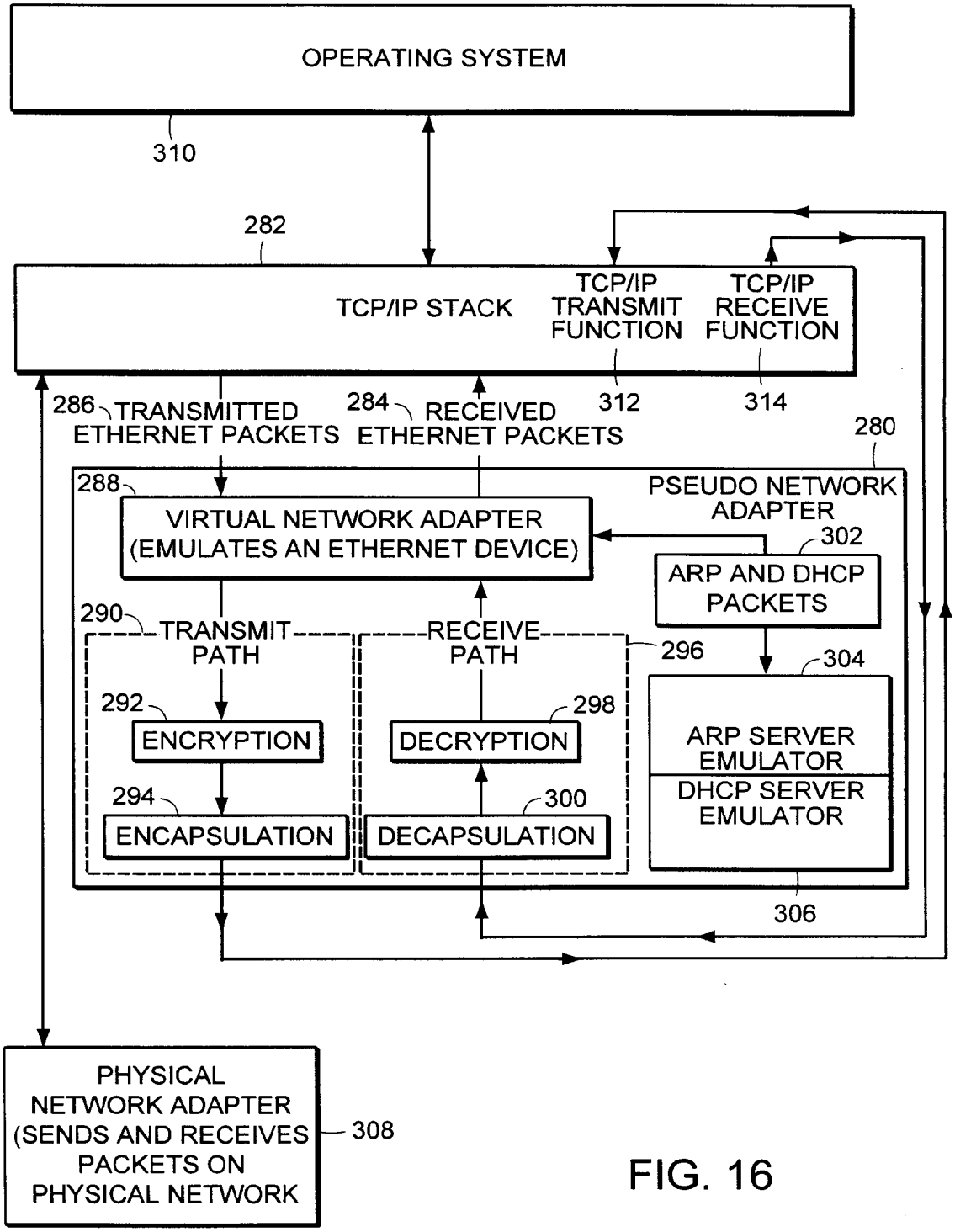


FIG. 16

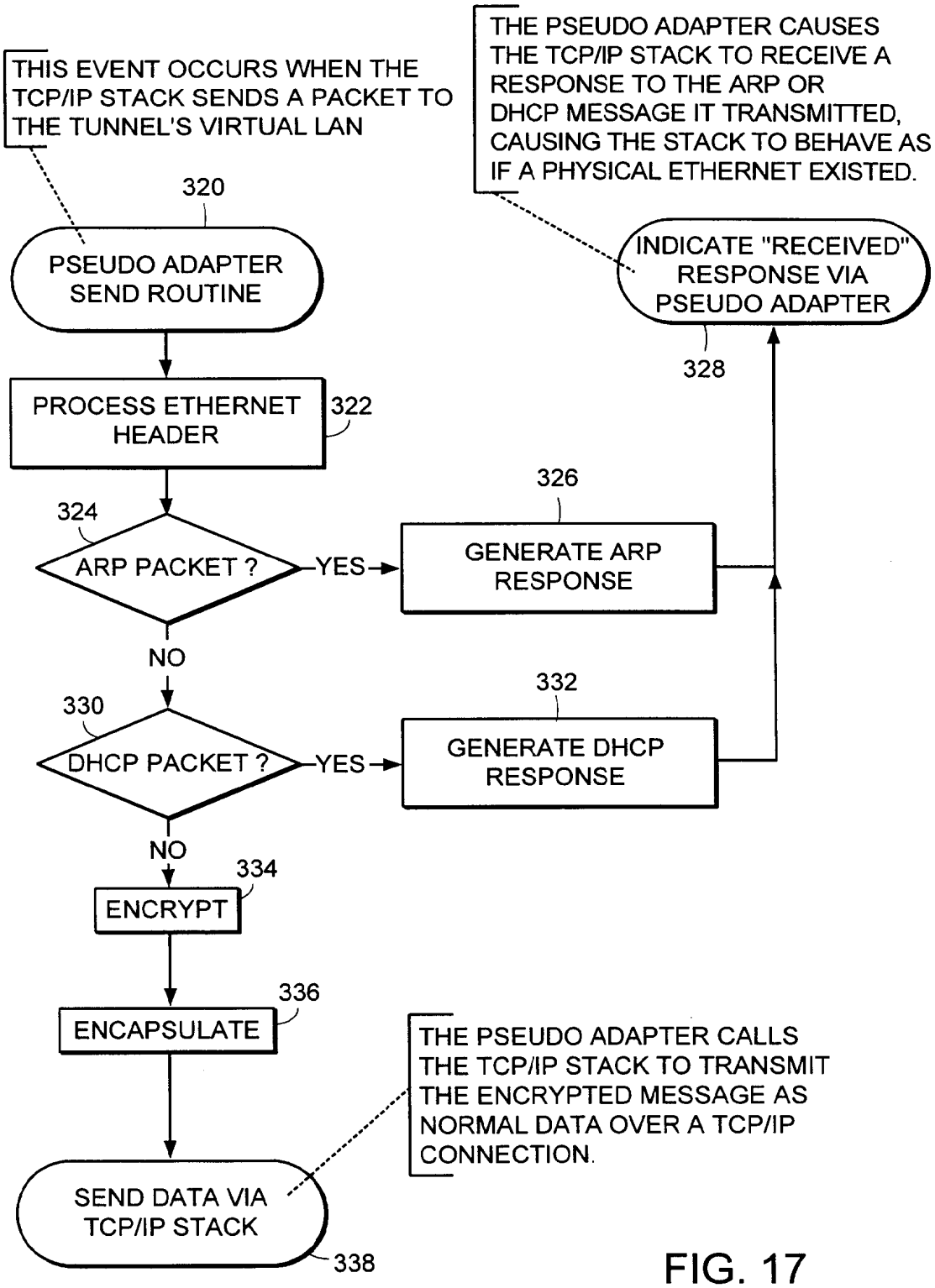


FIG. 17

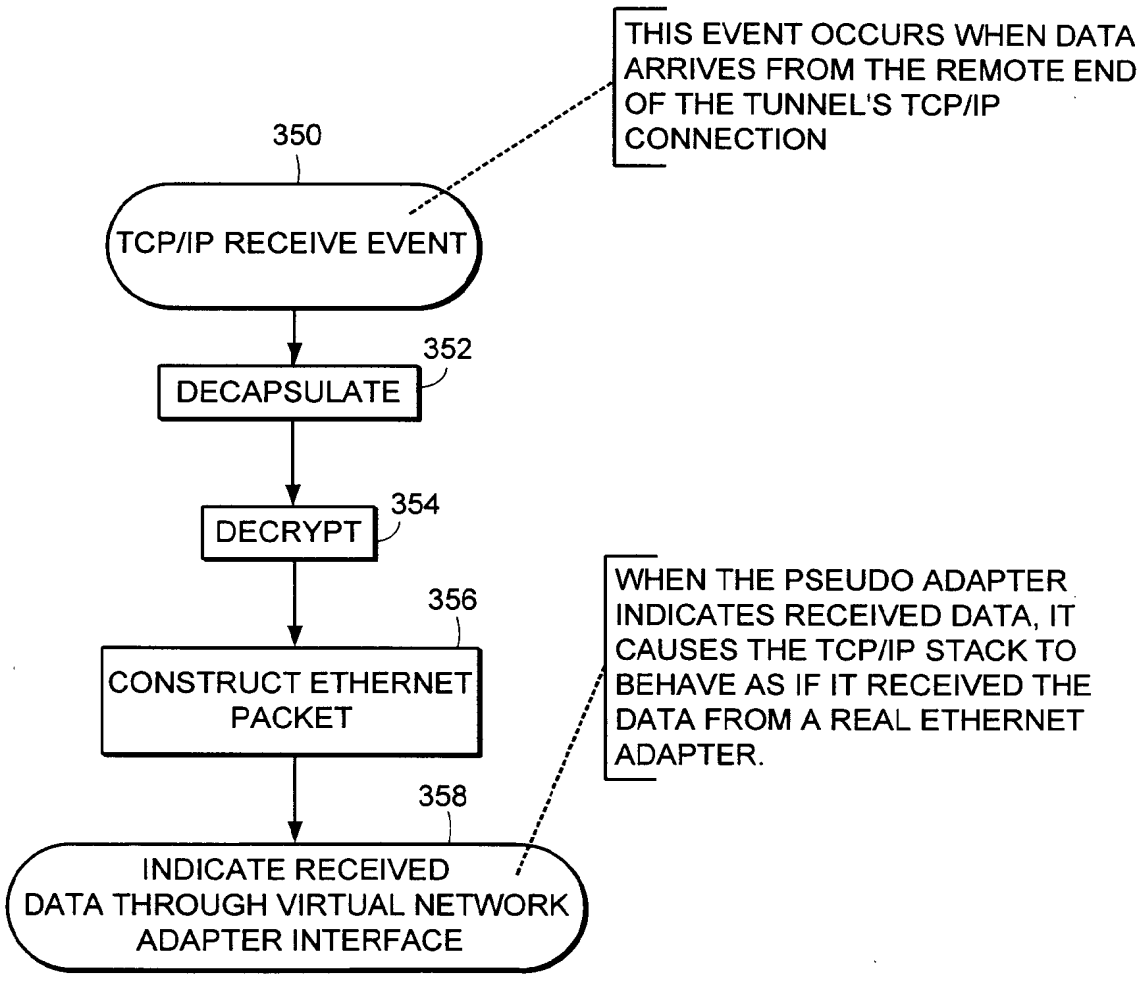


FIG. 18

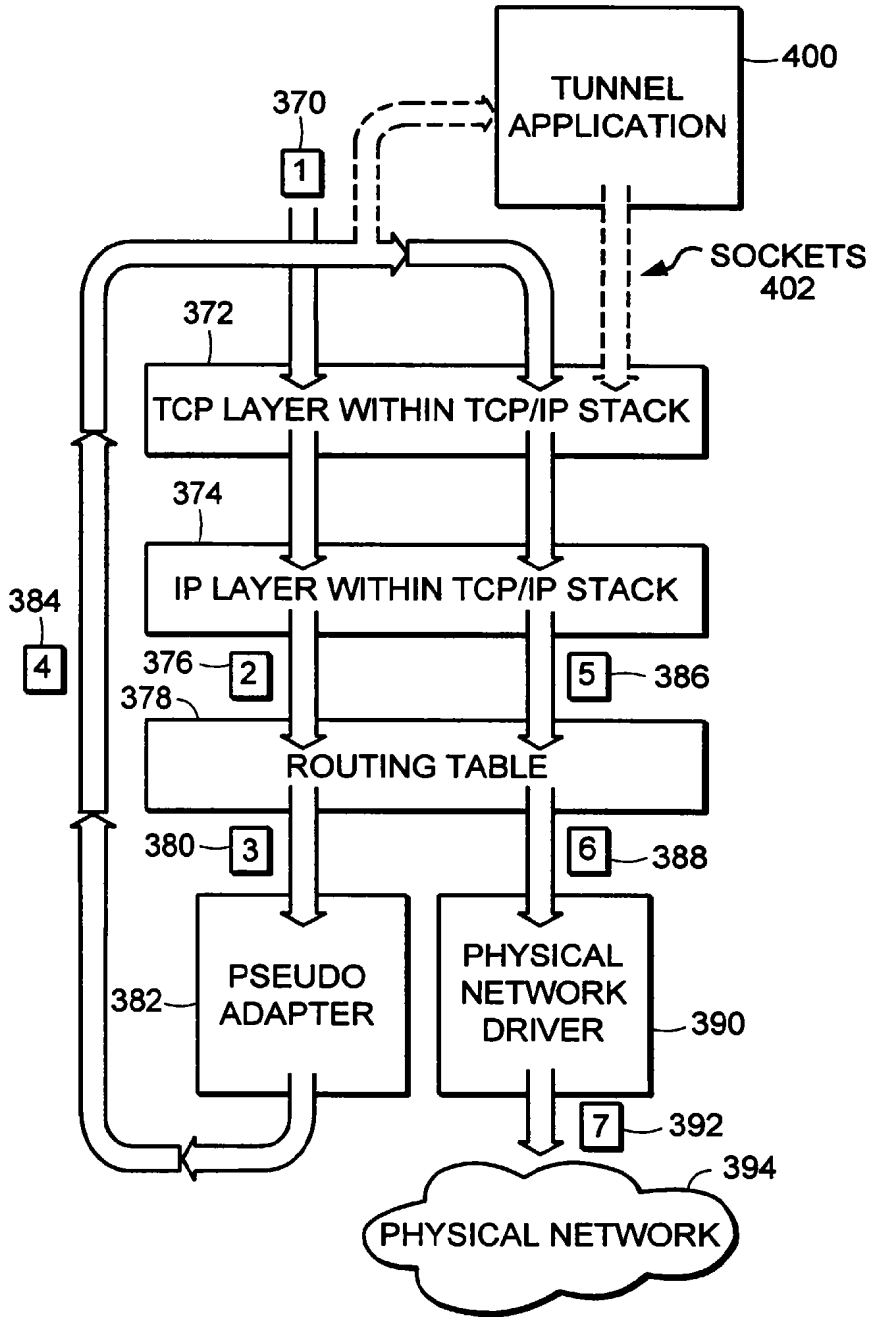


FIG. 19

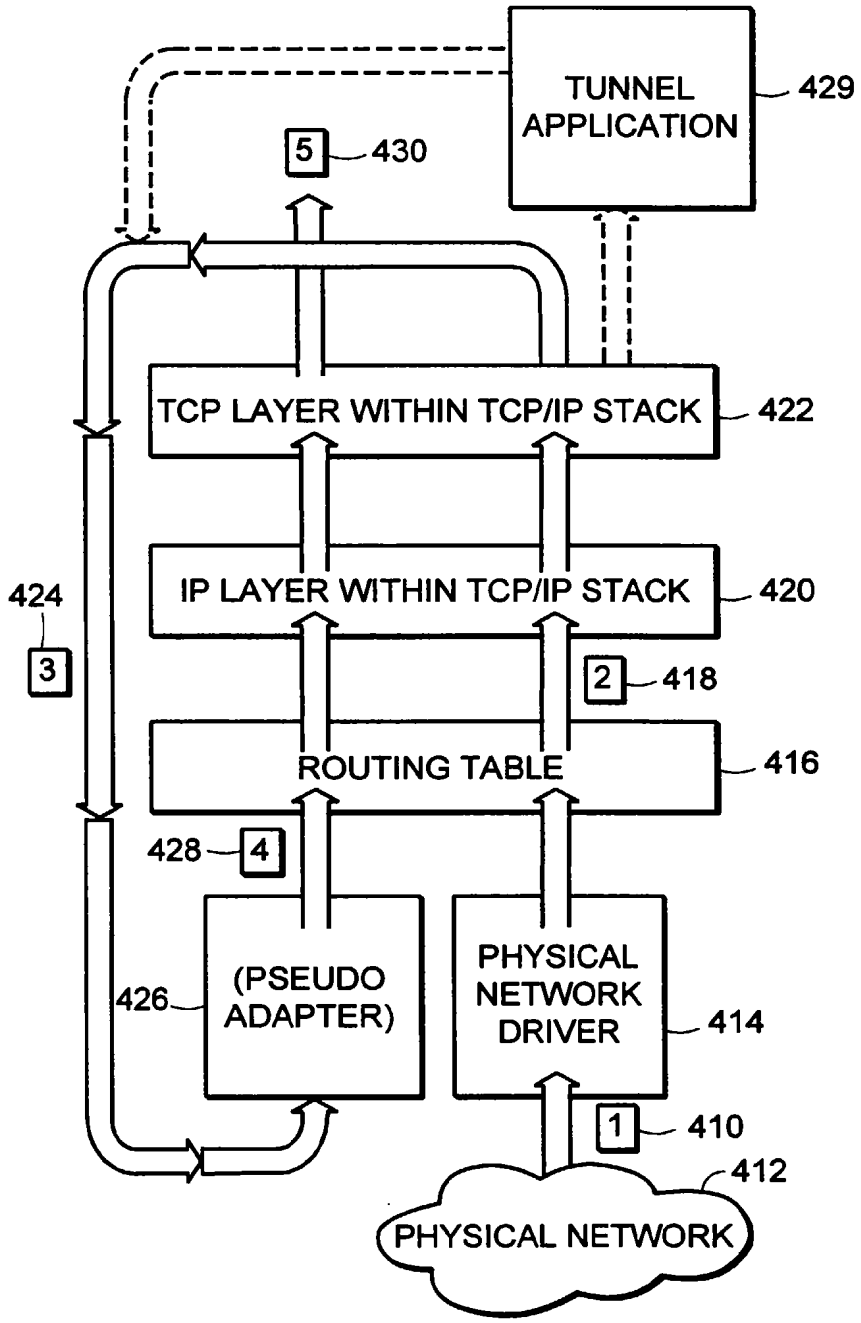


FIG. 20

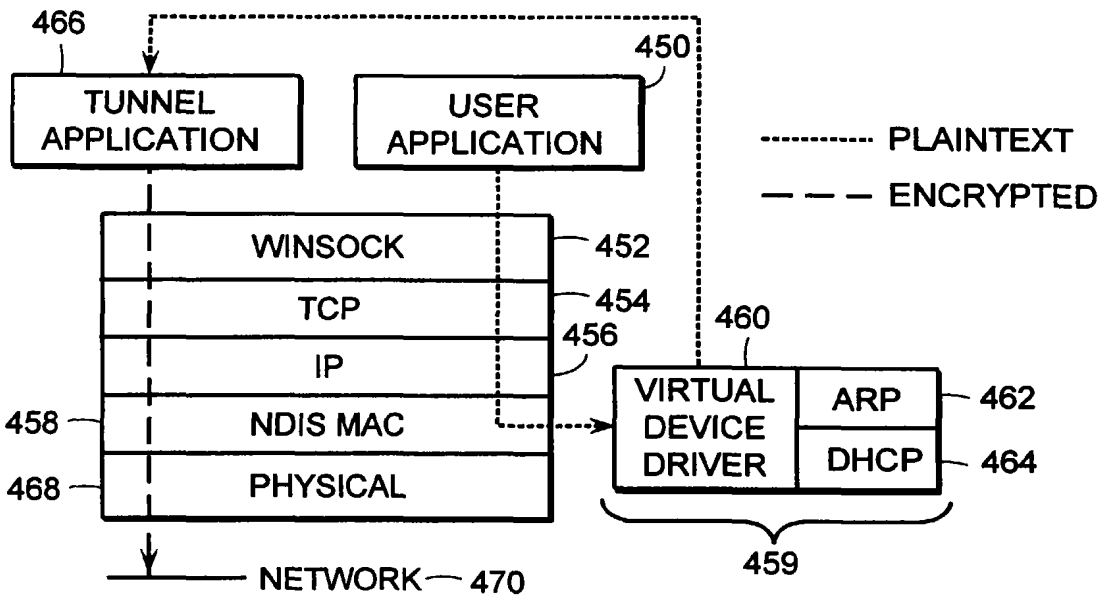


FIG. 21

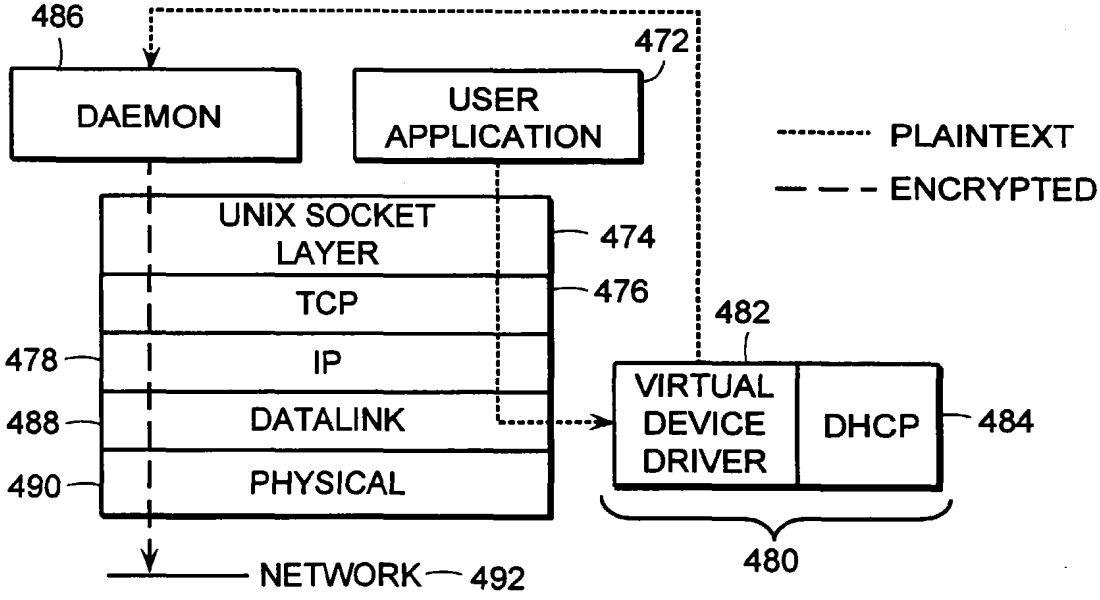


FIG. 22

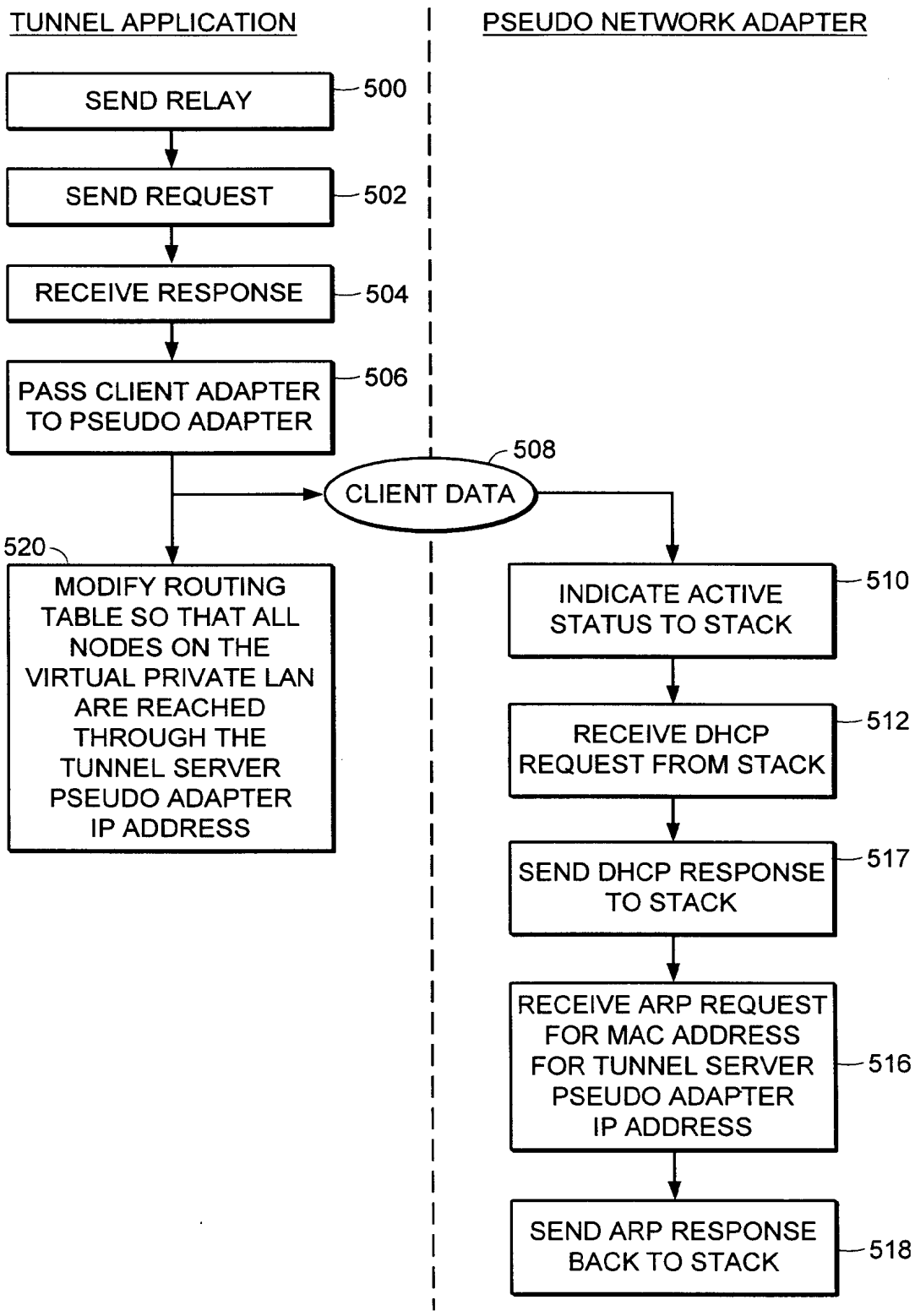


FIG. 23

GB2317792

Publication Title:

Virtual Private Network for encrypted firewall

Abstract:

Abstract of GB2317792

A system (10) for regulating the flow of messages through a firewall (18) having a network protocol stack, wherein the network protocol stack includes an Internet Protocol (IP) layer where if the message is not encrypted, it passes the unencrypted message up the network protocol stack to an application level proxy (50), and if the message is encrypted, it decrypts the message and passes the decrypted message up the network protocol stack to the application level proxy. The step of decrypting the message includes the step of executing a process at the IP layer to decrypt the message. Data supplied from the esp@cenet database - Worldwide

(12) UK Patent Application (19) GB (11) 2 317 792 (13) A

(43) Date of A Publication 01.04.1998

(21) Application No 9719816.2

(22) Date of Filing 17.09.1997

(30) Priority Data
(31) 08715343 (32) 18.09.1996 (33) US
08715668 18.09.1996

(71) Applicant(s)
Secure Computing Corporation

(Incorporated in USA - Delaware)

2675 Long Lake Road, Roseville,
Minnesota 55113-2536, United States of America

(72) Inventor(s)
Spence Minear
Edward B Stockwell
Troy De Jongh

(74) Agent and/or Address for Service
Beresford & Co
2-5 Warwick Court, High Holborn, LONDON,
WC1R 5DJ, United Kingdom

(51) INT CL^6
H04L 9/00

(52) UK CL (Edition P)
H4P PPEB
U1S S2124 S2209

(56) Documents Cited
WO 97/26735 A1 **WO 97/26734 A1** **WO 97/26731 A1**
WO 97/23972 A1 **WO 97/13340 A1**

(58) Field of Search
UK CL (Edition P) **H4P PDCSA PDCSC PPEB**
INT CL^6 **H04L 9/00 9/32 29/06 29/08**
Online: WPI, INSPEC

(54) Virtual Private Network for encrypted firewall

(57) A system (10) for regulating the flow of messages through a firewall (18) having a network protocol stack, wherein the network protocol stack includes an Internet Protocol (IP) layer where if the message is not encrypted, it passes the unencrypted message up the network protocol stack to an application level proxy (50), and if the message is encrypted, it decrypts the message and passes the decrypted message up the network protocol stack to the application level proxy. The step of decrypting the message includes the step of executing a process at the IP layer to decrypt the message.

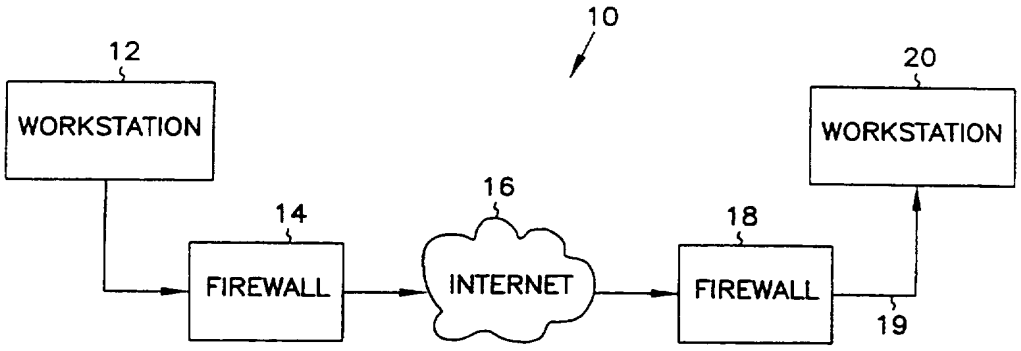


FIG. 1

GB 2 317 792 A

At least one drawing originally filed was informal and the print reproduced here is taken from a later filed formal copy.

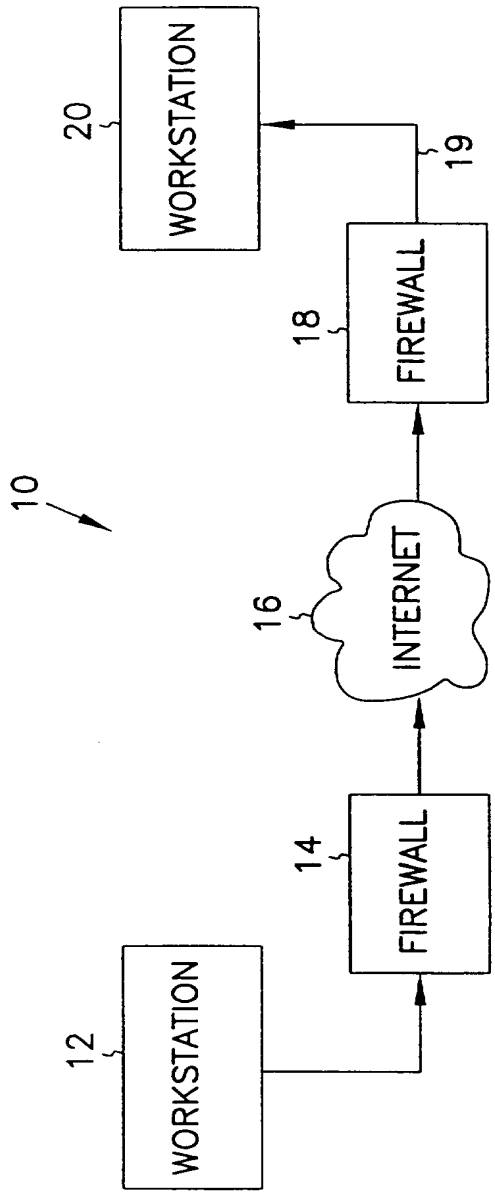

10
12
WORKSTATION
14
FIREWALL
16
INTERNET
18
FIREWALL
19
20
WORKSTATION

FIG. 1

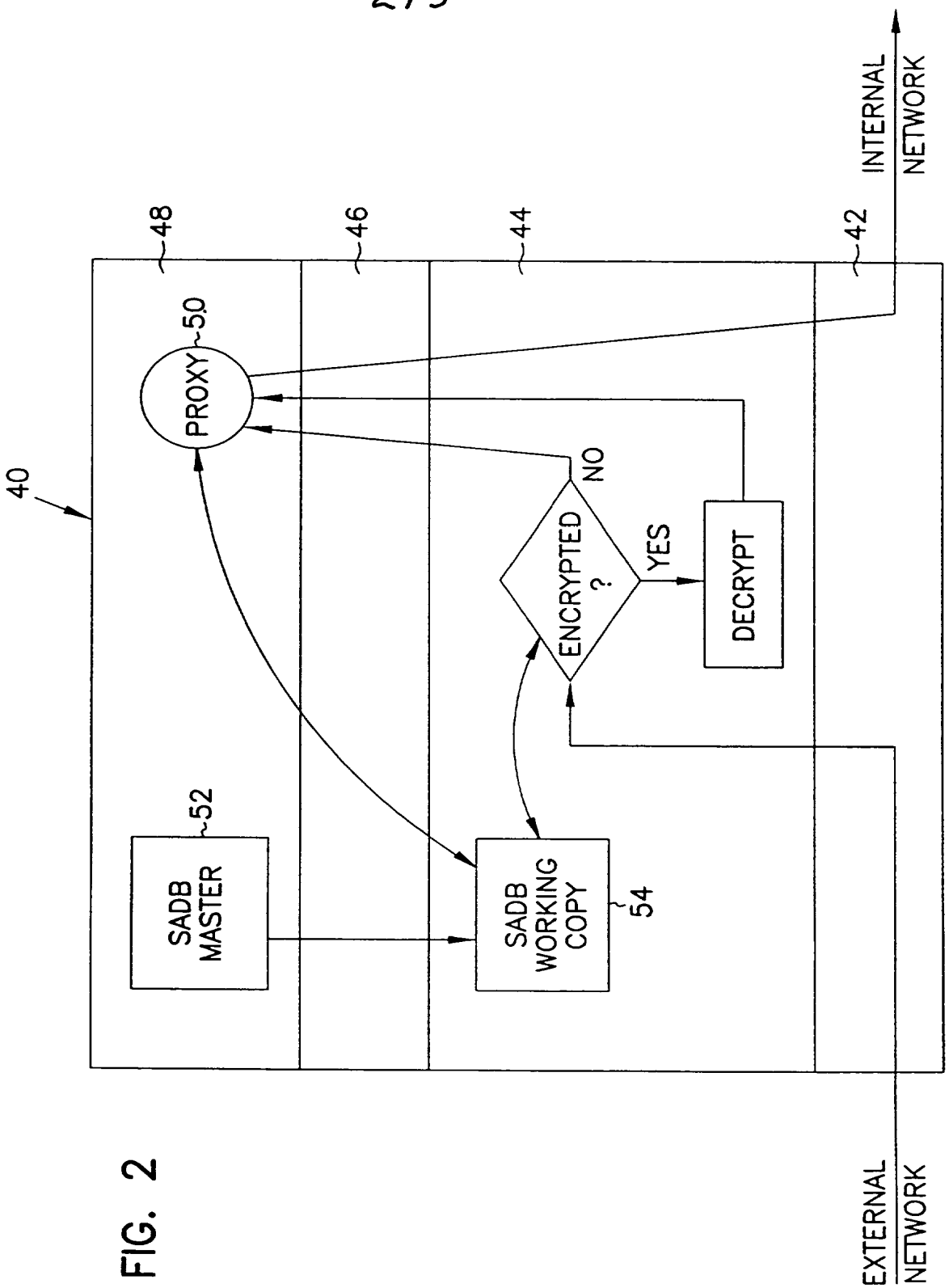

40
48
46
44
42
INTERNAL NETWORK
PROXY
50
NO
ENCRYPTED ?
YES
DECRYPT
SADB MASTER
52
SADB WORKING COPY
54
EXTERNAL NETWORK

FIG. 2

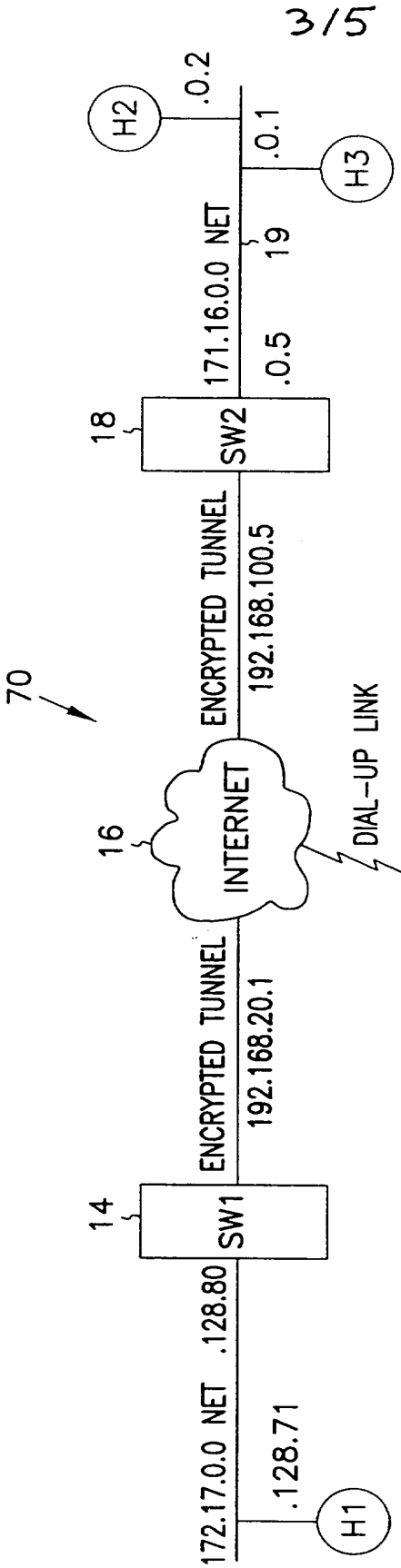

70
H1
.128.71
172.17.0.0 NET
.128.80
14
SW1
ENCRYPTED TUNNEL
192.168.20.1
16
INTERNET
DIAL-UP LINK
L1
ENCRYPTED TUNNEL
192.168.100.5
18
SW2
.0.5
171.16.0.0 NET
19
.0.1
H3
.0.2
H2

FIG. 3

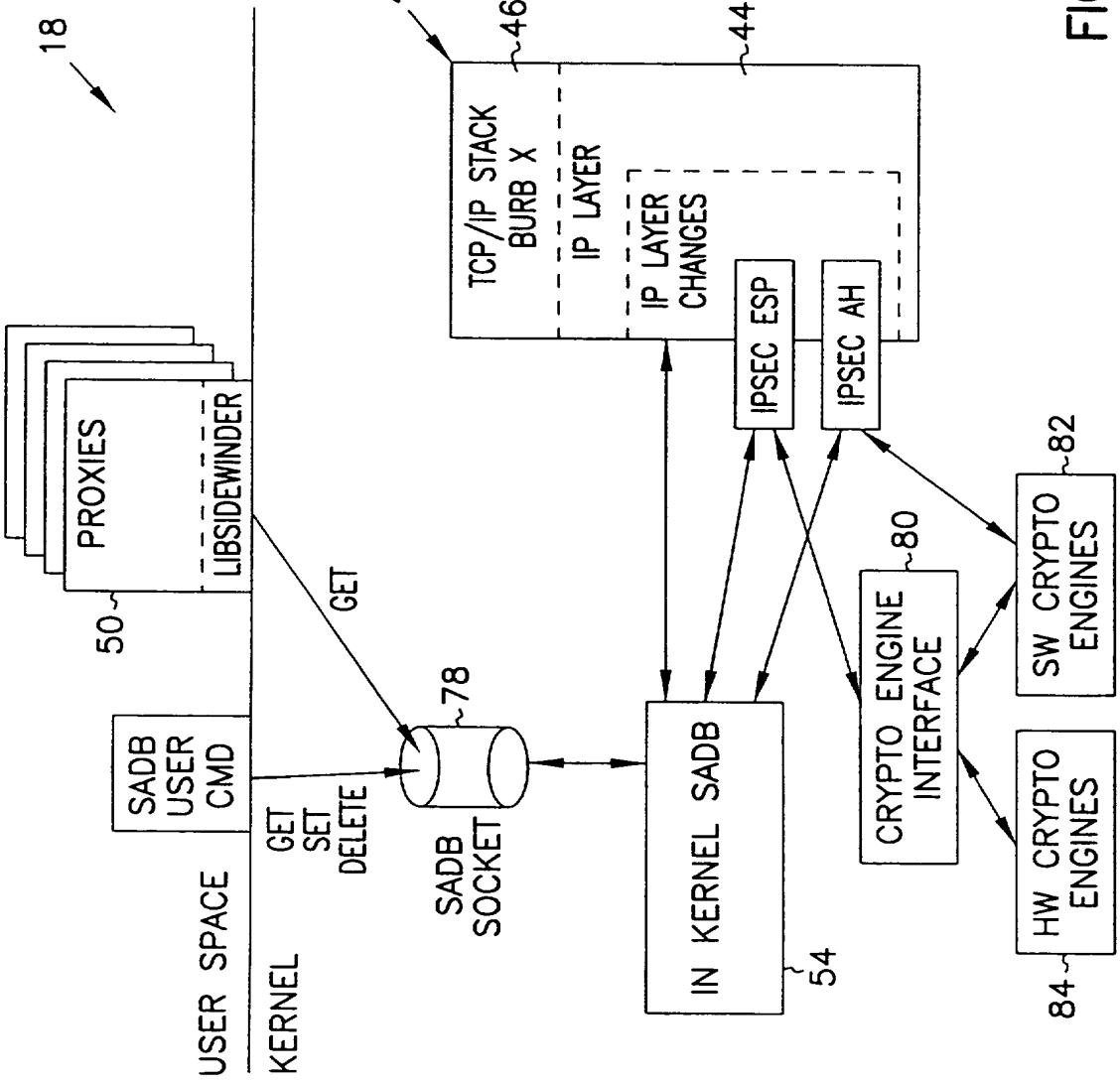

18
40x
46x
44x
TCP/IP STACK
BURB X
IP LAYER
IP LAYER
CHANGES
IPSEC ESP
IPSEC AH
PROXIES
LIBSIDEWINDER
50
GET
78
SADB
USER
CMD
GET
SET
DELETE
SADB
SOCKET
IN KERNEL SADB
54
CRYPTO ENGINE
INTERFACE
80
SW CRYPTO
ENGINES
82
HW CRYPTO
ENGINES
84
USER SPACE
KERNEL

FIG. 4

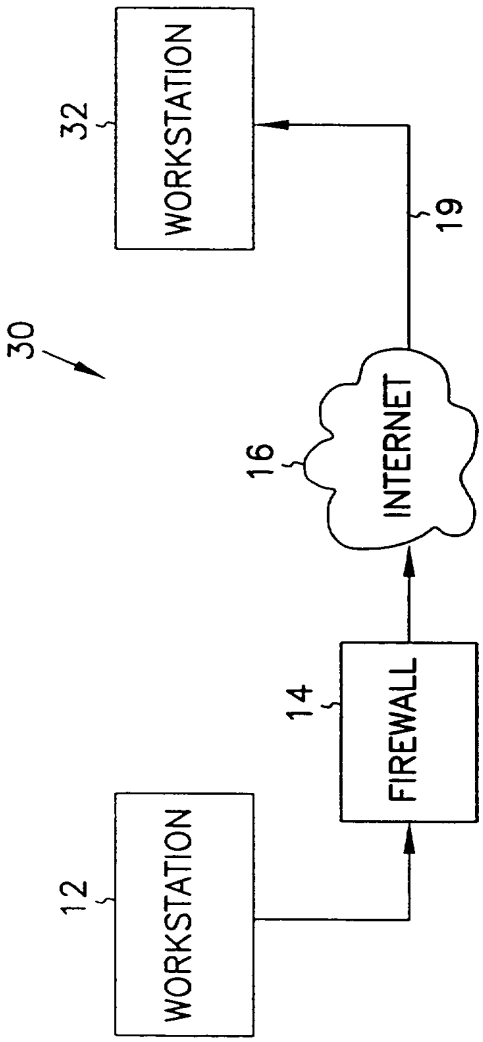

30
12
WORKSTATION
14
FIREWALL
16
INTERNET
19
32
WORKSTATION

FIG. 5

VIRTUAL PRIVATE NETWORK ON APPLICATION GATEWAY

Background of the Invention

Field of the Invention

The present invention pertains generally to network communications, and in particular to a system and method for securely transferring information between firewalls over an unprotected network.

Background Information

Firewalls have become an increasingly important part of network design. Firewalls provide protection of valuable resources on a private network while allowing communication and access with systems located on an unprotected network such as the Internet. In addition, they operate to block attacks on a private network arriving from the unprotected network by providing a single connection with limited services. A well designed firewall limits the security problems of an Internet connection to a single firewall computer system. This allows an organization to focus their network security efforts on the definition of the security policy enforced by the firewall. An example of a firewall is given in "SYSTEM AND METHOD FOR PROVIDING SECURE INTERNETWORK SERVICES" by Boebert et al. (PCT Published Application No. WO 96/13113, published on May 2, 1996), the description of which is hereby incorporated by reference. Another description of a firewall is provided by Dan Thomsen in "Type Enforcement: the new security model", *Proceedings: Multimedia: Full-Service Impact on Business, Education, and the Home*, SPIE Vol. 2617, p. 143, August 1996. Yet another such system is described in "SYSTEM AND METHOD FOR ACHIEVING NETWORK SEPARATION" by Gooderum et al. (PCT Published Application No. WO 97/29413, published on August 14, 1997), the description of which is hereby incorporated by reference. All the above systems are examples of application level gateways. Application level gateways use proxies or other such mechanisms operating at the application layer to process traffic through the firewall. As such, they can review not only the

message traffic but also message content. In addition, they provide authentication and identification services, access control and auditing.

Data to be transferred on unprotected networks like the Internet is susceptible to electronic eavesdropping and accidental (or deliberate) corruption. Although a firewall can protect data within a private network from attacks launched from the unprotected network, even that data is vulnerable to both eavesdropping and corruption when transferred from the private network to an external machine. To address this danger, the Internet Engineering Task Force (IETF) developed a standard for protecting data transferred between firewalls over an unprotected network. The Internet Protocol Security (IPSEC) standard calls for encrypting data before it leaves the first firewall, and then decrypting the data when it is received by the second firewall. The decrypted data is then delivered to its destination, usually a user workstation connected to the second firewall. For this reason IPSEC encryption is sometimes called *firewall-to-firewall encryption* (FFE) and the connection between a workstation connected to the first firewall and a client or server connected to the second firewall is termed a *virtual private network*, or VPN.

The two main components of IPSEC security are data encryption and sender authentication. Data encryption increases the cost and time required for the eavesdropping party to read the transmitted data. Sender authentication ensures that the destination system can verify whether or not the encrypted data was actually sent from the workstation that it was supposed to be sent from. The IPSEC standard defines an encapsulated payload (ESP) as the mechanism used to transfer encrypted data. The standard defines an authentication header (AH) as the mechanism for establishing the sending workstation's identity.

Through the proper use of encryption, the problems of eavesdropping and corruption can be avoided; in effect, a protected connection is established from the internal network connected to one firewall through to an internal network connected to the second firewall. In addition, IPSEC can be used to provide a protected connection to an external computing system such as a portable personal computer.

IPSEC encryption and decryption work within the IP layer of the network protocol stack. This means that all communication between two IP addresses will be protected because all interfirewall communication must go through the IP layer. Such an approach is preferable over encryption and decryption at higher levels in the network protocol stack since when encryption is performed at layers higher than the IP layer more work is required to ensure that all supported communication is properly protected. In addition, since IPSEC encryption is handled below the Transport layer, IPSEC can encrypt data sent by any application. IPSEC therefore becomes a transparent add-on to such protocols as TCP and UDP.

Since, however, IPSEC decryption occurs at the IP layer, it can be difficult to port IPSEC to an application level gateway while still maintaining control at the proxy over authentication, message content, access control and auditing. Although the IPSEC specification in RFC 1825 suggests the use of a mandatory access control mechanism in a multi-level secure (MLS) network to compare a security level associated with the message with the security level of the receiving process, such an approach provides only limited utility in an application level gateway environment. In fact, implementations on application level gateways to date have simply relied on the fact that the message was IPSEC-encrypted as assurance that the message is legitimate and have simply decoded and forwarded the message to its destination. This creates, however, a potential chink in the firewall by assuming that the encrypted communication has access to all services.

What is needed is a method of handling IPSEC messages within an application level gateway which overcomes the above deficiencies. The method should allow control over access by an IPSEC connection to individual services within the internal network.

Summary of the Invention

The present invention is a system and method for regulating the flow of messages through a firewall having a network protocol stack, wherein the network protocol stack includes an Internet Protocol (IP) layer, the method

comprising the steps of determining, at the IP layer, if a message is encrypted, if the message is not encrypted, passing the unencrypted message up the network protocol stack to an application level proxy, and if the message is encrypted, decrypting the message and passing the decrypted message up the network protocol stack to the application level proxy, wherein the step of decrypting the message includes the step of executing a procedure at the IP layer to decrypt the message.

According to another aspect of the present invention, a system and method is described for authenticating the sender of a message within a computer system having a network protocol stack, wherein the network protocol stack includes an Internet Protocol (IP) layer, the method comprising the steps of determining, at the IP layer, if the message is encrypted, if the message is encrypted, decrypting the message, wherein the step of decrypting the message includes the step of executing a procedure at the IP layer to decrypt the message, passing the decrypted message up the network protocol stack to an application level proxy, determining an authentication protocol appropriate for the message, and executing the authentication protocol to authenticate the sender of the message.

Brief Description of the Drawings

In the following detailed description of example embodiments of the invention, reference is made to the accompanying drawings which form a part hereof, and which is shown by way of illustration only, specific embodiments in which the invention may be practiced. It is to be understood that other embodiments may be utilized and structural changes may be made without departing from the scope of the present invention.

In the drawings, where like numerals refer to like components throughout the several views:

Figure 1 is a functional block diagram of an application level gateway-implemented firewall-to-firewall encryption scheme according to the present invention;

Figure 2 is a block diagram showing access control checking of both encrypted and unencrypted messages in network protocol stack according to the present invention;

Figure 3 is a block diagram of a representative application level gateway-implemented firewall-to-firewall encryption scheme;

Figure 4 is a block diagram of one embodiment of a network-separated protocol stack implementing IPSEC according to the present invention; and

Figure 5 is a functional block diagram of a firewall-to-workstation encryption scheme according to the present invention.

Description of the Preferred Embodiments

In the following detailed description of the preferred embodiment, references made to the accompanying drawings which form a part hereof, and in which is shown by way of illustration specific preferred embodiments in which the invention may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the invention, and it is to be understood that other embodiments may be utilized and that structural, logical, physical, architectural, and electrical changes may be made without departing from the spirit and scope of the present invention. The following detailed description is, therefore, not to be taken in a limiting sense, and the scope of the present invention is defined only by the appended claims and their equivalents.

A system 10 which can be used for firewall-to-firewall encryption (FFE) is shown in Figure 1. In Figure 1, system 10 includes a workstation 12 communicating through a firewall 14 to an unprotected network 16 such as the Internet. System 10 also includes a workstation 20 communicating through a firewall 18 to unprotected network 16. In one embodiment, firewall 18 is an application level gateway.

As noted above, IPSEC encryption and decryption work within the IP layer of the network protocol stack. This means that all communications between two IP addresses will be protected because all interfirewall communication must pass through the IP layer. IPSEC takes the standard

Internet packet and converts it into a carrier packet. The carrier packet is designed to do two things: to conceal the contents of the original packet (encryption) and to provide a mechanism by which the receiving firewall can verify the source of the packet (authentication). In one embodiment of the present invention, each IPSEC carrier packet includes both an authentication header used to authenticate the sending machine and an encapsulated payload containing encrypted data. The authentication header and the encapsulated payload features of IPSEC can, however, be used independently. As required in RFC 1825, DES-CBC is provided for use in encrypting the encapsulated payload while the authentication header uses keyed MD5.

To use IPSEC, you must create a *security association* (SA) for each destination IP address. In one embodiment, each SA contains the following information:

- Security Parameters Index (SPI) - The index used to find a SA on receipt of an IPSEC datagram.
- Destination IP address - The address used to find the SA and trigger use of IPSEC processing on output.
- The peer SPI - The SPI value to put on a IPSEC datagram on output.
- The peer IP address - The destination IP address to be put into the packet header if IPSEC Tunnel mode is used.
- The Encryption Security Payload (ESP) algorithm to be used.
- The ESP key to used for decryption of input datagrams.
- The ESP key to used for encryption of output datagrams.
- The authentication (AH) algorithm to be used.
- The AH key to be used for validation of input packets.
- The AH key to be used for generation of the authentication data for output datagrams.

The combination of a given Security Parameter Index and Destination IP address uniquely identifies a particular "Security Association." In one

embodiment, the sending firewall uses the sending userid and Destination Address to select an appropriate Security Association (and hence SPI value). The receiving firewall uses the combination of SPI value and Source address to obtain the appropriate Security Association.

A security association is normally one-way. An authenticated communications session between two firewalls will normally have two Security Parameter Indexes in use (one in each direction). The combination of a particular Security Parameter Index and a particular Destination Address uniquely identifies the Security Association.

More information on the specifics of an IPSEC FFE implementation can be obtained from the standards developed by the IPSEC work group and documented in *Security Architecture for IP* (RFC 1825) and in RFC's 1826-1829.

When a datagram is received from unprotected network 16 or is to be transmitted to a destination across unprotected network 16, the firewall must be able to determine the algorithms, keys, etc. that must be used to process the datagram correctly. In one embodiment, this information is obtained via a security association lookup. In one such embodiment, the lookup routine is passed several arguments: the source IP address if the datagram is being received from network 16 or the destination IP address if the datagram is to be transmitted across network 16, the SPI, and a flag that is used to indicate whether the lookup is being done to receive or transmit a datagram.

When an IPSEC datagram is received by firewall 18 from unprotected network 16, the SPI and source IP address are determined by looking in the datagram. In one embodiment a Security Association Database (SADB) stored within firewall 18 is searched for the entry with a matching SPI. In one such embodiment, security associations can be set up based on network address as well as a more granular host address. This allows the network administrator to create a security association between two firewalls with only a couple of lines in a configuration file on each machine. For such embodiments, the entry in the Security Association Database that has both the matching SPI and the longest

address match is selected as the SA entry. In another such embodiment, each SA has a prefix length value associated with the address. An address match on a SA entry means that the addresses match for the number of bits specified by the prefix length value.

There are two exceptions to this search process. First, when an SA entry is set marked as being dynamic it implies that the user of this SA may not have a fixed IP address. In this case the match is fully determined by the SPI value. Thus it is necessary that the SPI values for such SA entries be unique in the SADB. The second exception is for SA entries marked as tunnel mode entries. In this case it is normally the case that the sending entity will hide its source address so that all that is visible on the public wire is the destination address. In this case, like in the case where the SA entries are for dynamic IP addresses, the search is done exclusively on the basis of the SPI.

When transmitting a datagram across unprotected network 16 the SADB is searched using only the destination address as an input. In this case the entry which has the longest address match is selected and returned to the calling routine.

In one embodiment, if firewall 18 receives datagrams which are identified as either an IP_PROTO_IPSEC_ESP or IP_PROTO_IPSEC_AH protocol datagram, there must be a corresponding SA in the SADB or else firewall 18 will drop the packet and an audit message will be generated. Such an occurrence might indicate a possible attack or it might simply be a symptom of an erroneous key entry in the Security Association Database.

In a system such as system 10, application level gateway firewall 18 acts as a buffer between unprotected network 16 and workstations such as workstation 20. Messages coming from unprotected network 16 are reviewed and a determination is made as to whether execution of an authentication and identification protocol is warranted. In contrast to previous systems, system 10 also performs this same determination on IPSEC-encrypted messages. If desired, the same authentication and identification can be made on messages to be transferred from workstation 20 to unprotected network 16. Figure 2

illustrates one way of authenticating both encrypted and unencrypted messages in a system such as system 10.

In the system of Figure 2 a network protocol stack 40 includes a physical layer 42, an Internet protocol (IP) layer 44, a Transport layer 46 and an application layer 48. Such a protocol stack exists, for instance on application level gateway firewall 18 of Figure 1. An application executing in application layer 48 can communicate to an application executing on another system by preparing a message and transmitting it through one of the existing transport services executing on transport layer 46. Transport layer 46 in turn uses a process executing in IP layer 44 to continue the transfer. Physical layer 42 provides the software needed to transfer data through the communication hardware (e.g., a network interface card or a modem). As noted above, IPSEC executes within IP layer 44. Encryption and authentication is transparent to the host as long as the network administrator has the Security Association Database correctly configured and a key management mechanism is in place on the firewall.

In application level gateway firewall 18, a proxy 50 operating within application layer 48 processes messages transferred between internal and external networks. All network-to-network traffic must pass through one of the proxies within application layer 48 before being the transfer across networks is allowed. A message arriving from external network 16 is examined at IP layer 44 and an SADB is queried to determine if the source address and SPI are associated with an SA. In the embodiment shown in Figure 2, an SADB Master copy 52 is maintained in persistent memory at application layer 48 while a copy 54 of SADB is maintained in volatile memory within the kernel. If the message is supposed to be encrypted, the message is decrypted based on the algorithm and key associated with the particular SA and the message is transferred up through transport layer 46 to proxy 50. Proxy 50 examines the source and destination addresses and the type of service desired and decides whether authentication of the sender is warranted. If so, proxy 50 initiates an authentication protocol. The protocol may be as simple as requesting a user

name and password or it may include a challenge/response authentication process. Proxy 50 also looks to see whether the message coming in was encrypted or not and may factor that into whether a particular type of authentication is needed. In Telnet, for instance, user name/password authentication may be sufficient for an FFE link while the security policy may dictate that a more stringent challenge/response protocol is needed for unencrypted links. In that case, proxy 50 will be a Telnet proxy and it will base its authentication protocol on whether the link was encrypted or not.

Since IPSEC executes within IP layer 44 there is no need for host firewalls to update their applications. Users that already have IPSEC available on their own host machine will, however, have to request that the firewall administrator set up SA's in the SADB for their traffic.

In the embodiment shown in Figure 2, a working copy 54 of the Security Association Database consisting of all currently active SA's is kept resident in memory for ready access by IP layer processing as datagrams are received and transmitted. In addition, a working master copy 52 of the SADB is maintained in a file in nonvolatile memory. During system startup and initialization processing the content of all of the required SA's in master SADB 52 is added to the working copy 54 stored in kernel memory.

In one embodiment, firewall 18 maintains different levels of security on internal and external network interfaces. It is desirable for a firewall to have different levels of security on both the internal and external interfaces. In one embodiment, firewall 18 supports three different levels, numbered 0 through 2. These levels provide a simple policy mechanism that controls permission for both in-bound and out-bound packets.

- Level 0 - do not allow any in-bound or out-bound traffic unless there is a security association between the source and destination.

- Level 1 - Allow both in-bound and out-bound non-IPSEC traffic but force the use of IPSEC if a SA exists for the address. (To support this firewall 18 must look for a SA for each in-bound datagram.)
- Level 2 - allow NULL security associations to exist. NULL associations are just like normal security associations, except no encryption or authentication transform is performed on in-bound or out-bound packets that correspond to this NULL association. With Level 2 enabled, the machine will still receive unprotected traffic, but it will not transmit unless Level 1 is enabled.

The default protection level established when the Security Association Database (SADB) is initialized at boot time is 1 for in-bound traffic and 2 for out-bound traffic.

An Access Control List, or ACL, is a list of rules that regulate the flow of Internet connections through a firewall. These rules control how a firewall's servers and proxies will react to connection attempts. When a server or proxy receives an incoming connection, it performs an ACL check on that connection.

An ACL check compares a set of parameters associated with the connection against a list of ACL rules. The rules determine whether the connection is allowed or denied. A rule can also have one or more side effects. A side effect causes the proxy to change its behavior in some fashion. For example, a common side effect is to redirect the destination IP address to an alternate machine. In addition to IP connection attempts, ACL checks can also made on the console logins and on logins made from serial ports. Finally, ACL checks can also be made on behalf of IP access devices, such as a Cisco box, through the use of the industry standard TACACS+ protocol.

In one embodiment, the ACL is managed by an acld daemon running in the kernel of firewalls 10 and 30. The acld daemon receives two types of requests, one to query the ACL and one to administer it. In one such embodiment, the ACL is stored in a relational database such as the Oracle database for fast access. By using such a database, query execution is asynchronous and many queries can be executing concurrently. In addition, these types of databases are designed to manipulate long lists of rules quickly

and efficiently. These qualities ensure that a given query cannot hang up the process that issued the query for any appreciable time (> 1-2 seconds).

In one such embodiment, the database can hold up to 100,000 users and up to 10,000 hosts but can be scaled up to the capacity of the underlying database engine. The results of an ACL check is cached, allowing repeated checks to be turned around very quickly.

Applications on firewalls 10 and 30 can query acld to determine if a given connection attempt should be allowed to succeed. In one embodiment, the types of applications (i.e. "agents") that can make ACL queries can be divided into four classes:

1) Proxies. These allow connections to pass through firewall 10 or 30 in order to provide access to a remote service. They include tnauthp (authenticated telnet proxy), pftp (FTP proxy), httpp (HTTP proxy), and tcpgsp (TCP generic service proxy).

2) Servers. These provide a service on the firewall itself. They include ftpd and httpd.

3) Login agents. Login agent is a program on the firewall that can create a Unix shell. It is not considered a server because it cannot receive IP connections. One example is /usr/bin/login when used to create a dialup session or a console session on firewall 10 or 30. Another example is the command *srole*.

4) Network Access Servers (NAS). NAS is a remote IP access device, typically a dialup box manufactured by such companies as Cisco or Bridge. The NAS usually provides dialup telnet service and may also provide SLIP or PPP service.

Proxies, servers, login agents, and NASes make queries to acld to determine if a given connection attempt should be allowed to succeed. All of the agents except NAS make their queries directly. NAS, because it is remote, must communicate via an auxiliary daemon that typically uses an industry standard protocol such as RADIUS or TACACS+. The auxiliary daemon (e.g., tacradd) in turn forwards the query to local acld.

As a side effect of the query, acld tells the agent if authentication is needed. If no authentication is needed, the connection proceeds immediately. Otherwise acld provides (as another side effect) a list of allowed authentication methods that the user can choose from. The agent can present a menu of choices or simply pick the first authentication method by default. Typical authentication methods include plain password, SNK DSS, SDI SecurID, LOCKout DES, and LOCKout FORTEZZA. In one embodiment, the list of allowed authentication methods varies depending on the host name, user name, time of day, or any combination thereof.

In the case of a Level 0 policy, it would be safe to assume that all incoming traffic is encrypted or authenticated. In the case of Levels 1 through 2, a determination must be made whether or not a security association exists for a given peer. Otherwise an application may believe that in-bound traffic has been authenticated when it really has not. (That is why it is necessary to look for an SA on input of each non-IPSEC datagram.)

In one embodiment, a flag which accompanies the message as it is sent from IP layer 44 to proxy 50 indicates whether the incoming message was or was not encrypted. In another embodiment, proxy 50 accesses Security Association Database 54 (the table in the kernel can be queried via an SADB routing socket (PF-SADB)) to determine whether or not a security association exists for a given peer.. The SADB socket is much like a routing socket found in the stock BSD 4.4 kernel (protocol family PF-ROUTE) except that PF-SADB sockets are used to maintain the Security Association Database (SADB) instead of the routing table. Because the private keys used for encryption, decryption, and keyed authentication are stored in this table, access must be strictly prohibited and allowed to only administrators and key management daemons. Care must be taken when allowing user-level daemons access to `/dev/mem` or `/dev/kmem` as well, since the keys are stored in kernel memory and could be exposed with some creative hacking.

In one embodiment, a command-line tool called `sadb` is used to support the generation and maintenance of in-kernel version 54 of SADB. The primary

interface between this tool and the SADB is the PF-SADB socket. The kernel provides socket processing to receive client requests to add, update, or change entries in in-kernel SADB 54. As noted above, the default protection level established when the Security Association Database (SADB) is initialized at boot time is 1 for in-bound traffic and 2 for out-bound traffic. This may be changed by the use of the `sadb` command.

The existing `sadb` command was derived from the NIST implementation of IPSEC. As noted above, this tool is much like `route` in that it uses a special socket to pass data structures in and out of the kernel. There are three commands recognized by the `sadb` command: *get, set, delete*. The following simple shell script supports adding and removing a single SA entry to SADB 54. It shows one embodiment of a parameter order for adding a SA to the SADB.

```
# ! /bin/sh
if [ $# -ne 1 ]
then
     echo "usage: $0 <on>|<off>" >&2
     exit 1
fi
ONOFF=$1

addsa ()
{
IPADDRESS=$2
PEERADDRESS=0.0.0.0
PREFIXLEN=0                  # Num of bits, 0 => full 32
bit match
LOCALADDRESS=0.0.0.0
REALADDRESS=0.0.0.0
PORT=0
PROTOCOL=0
UID=0
DESALG=1                     # I = DES-CBC
IVLEN=4                      # bytes
DESKEY=0b0b0b0b0b0b0b0b
DESKEYLEN=8                  # bytes
AHALG=1                      # 1 = MD5
AHKEY=30313233343536373031323334353637
AHKEYLEN=16                  # bytes
LOCAL_SPI=$1
```

```
PEER_SPI=$1
TUNNEL_MODE=0
AHRESULTLEN=4
COMBINED_MODE=1          # On output, 1 = ESP, then
AH; 0 = AH, then ESP
DYNAMIC_FLAG=0

if [ "$ONOFF" = "on"
then
     ./sadb add dst $IPADDRESS $PREFIXLEN $LOCAL_SPI
$UID $PEERADDRESS $PEER_SPI $TUNNEL_MODE $LOCALADDRESS
$REALADDRESS $PROTOCOL $PORT $DESALG $IVLEN $DESKEYLEN
$DESKEY $DESKEYLEN $DESKEY $AHALG $AHKEYLEN $AHKEY
$AHKEYLEN $AHKEY $AHRESULTLEN $COMBINED_MODE
$DYNAMIC_FLAG
else
     ./sadb delete dst $IPADDRESS $LOCAL-SPI
fi
}

#    Get down to work:
addsa 500 172.17.128.115          # number6.sctc.com
```

The current status of in-kernel SADB 54 can be obtained with the sadb command. The get option allows dumping the entire SADB or a single entry. In one embodiment, the complete dump approach uses /dev/kmem to find the information. The information may be presented as follows:

```
# sadb get dst

Local-SPI Address-Family Destination-Addr
Preflx_length UID
     Peer-Address Peer-SPI Transport-Type
     Local-Address Real-Address
     Protocol Port
     ESP_Alg_ID ESP_IVEC_Length
          ESP_Enc_Key_length ESP_Enc_ESP_Key
          ESP_Dec_Key_length ESP_Dec_ESP_Key
     AH_Alg_ID AH_Data_Length
          AH_Gen_Key_Length AH_Gen_Key
          AH_Check_Key_Length AH_Check_Key
     Combined_Mode  Dynamic_Flag
```

```
-----------------------------------------------------
-
500 INET: number6.sctc.com 0 0
      0.0.0.0    500 Transport(0) 0
      0.0.0.0 0.0.0.0
      None None
      DES/CBC-RFC1829(1) 4
            8 0b0b0b0b0b0b0b0b
            8 0b0b0b0b0b0b0b0b
      MD5-RFC1828(1) 4
            16 30313233343536373031323334353637
            16 30313233343536373031323334353637
      ESP+AH(1) 0
501 INET: spokes.sctc.com 0 0
      0.0.0.0    501 Transport(0) 0
      0.0.0.0.0.0.0.0
      None None
      DES/CBC-RFC1829(1) 4
            8 0b0b0b0b0b0b0b0b
            8 0b0b0b0b0b0b0b0b
      MD5-RFC1828(1) 4
            16 30313233343536373031323334353637
            16 30313233343536373031323334353637
      ESP+AH(1) 0

End of list.
```

When a new entry is added to in-kernel SADB 54, the add process first checks to see that no existing entry will match the values provided in the new entry. If no match is found then the entry is added to the end of the existing SADB list.

To illustrate the use and administration of an FFE, we'll go through an example using FFE 70 in Figure 3. Firewalls 14 and 18 are both application level gateway firewalls implemented according to the present invention. Workstations H2 and H3 both want to communicate with H1. For the administrator of firewalls 14 and 18, this is easy to accomplish. The administrator sets up a line something like this (we'll only show the IP address part and SPI parts of the SA, since they're the trickiest values to configure. Also, assume that we are using tunnel mode):

```
#  Hypothetical SW1 Config File
```

```
#
#  Fields are laid out in the following manner:
#  srcaddrornet= localSPI= peeraddr= peerSPI=
realsrcaddr= localaddr= key=

# The following entry sets up a tunnel between hosts
behind SW1
# and hosts behind SW2.
src=172.16.0.0 localSPI=666 peer=192.168.100.5
peerSPI=777 \
     realsrcaddr=192.168.100.5 localaddrs=0.0.0.0
     key=0xdeadbeeffadebabe

#  Hypothetical SW2 Config File
#
#  Fields are laid out in the following manner:
#  srcaddrornet= localSPI= peeraddr= peerSPI=
     realsrcaddr= localaddr= key=

#  The following entry sets up a tunnel between hosts
behind SW1 and
#  hosts behind SW2.
src=172.17.0.0 localSPI=777 peer=192.168.20.1
peerSPI=666 \
     realsrcaddr=192.168.20.1 localaddr=0.0.0.0 \
     key=Oxdeadbeeffadebabe
```

With this setup, all traffic is encrypted using one key, no matter who is talking to whom. For example, traffic from H2 to Hl as well as traffic from H3 to Hl will be encrypted with one key. Although this setup is small and simple, it may not be enough.

What happens if H2 cannot trust H3? In this case, the administrator can set up security associations at the host level. In this case, we have to rely on the SPI field of the SA, since the receiving firewall cannot tell from the datagram header which host behind the sending firewall sent the packet. Since the SPI is stored in IPSEC datagrams, we can do a lookup to obtain its value. Below are the sample configuration files for both firewalls again, but this time, each host combination communicates with a different key. Moreover, H2 excludes H3 from communications with Hl, and H3 excludes H2 in the same way.

```
#  Hypothetical SW1 Config File
#
#  Fields are laid out in the following manner:
#  srcaddrornet= localSPI= peeraddr= peerSPI=
realsrcaddr= localaddr= key=

# The following entry sets up a secure link between H2
and H1
src=172.16.0.2 localSPI=666 peer=192.168.100.5
peerSPI=777 \
     realsrcaddr=192.168.100.5
localaddrs=178.17.128.71 \
     key=0x0a0a0a0a0a0a0a0a

# The following entry sets up a secure link between H3
and H1
src=172.16.0.1 localSPI=555 peer=192.168.100.5
peerSPI=888 \
     realsrcaddr=192.168.100.5
localaddrs=178.17.128.71 \
     key=0x0b0b0b0b0b0b0b0b

#  Hypothetical SW2 Config File
#
#  Fields are laid out in the following manner:
#  srcaddrornet= localSPI= peeraddr= peerSPI=
realsrcaddr= localaddr= key=

# The following entry sets up a secure link between H2
and H1
src=172.17.128.71 localSPI=777 peer=192.168.20.1
peerSPI=666 \
     realsrcaddr=192.168.20.1 localaddrs=172.16.0.2 \
     key=0x0a0a0a0a0a0a0a0a

# The following entry sets up a secure link between H3
and H1
src=172.17.128.71 localSPI=888 peer=192.168.20.1
peerSPI=555 \
     realsrcaddr=192.168.20.1 localaddrs=172.16.0.1 \
     key=0x0b0b0b0b0b0b0b0b
```

Figure 4 is a block diagram showing in more detail one embodiment of an IPSEC-enabled application level gateway firewall 18. Application level gateway firewall 18 provides access control checking of both encrypted and

unencrypted messages in a more secure environment due to its network-separated architecture. Network separation divides a system into a set of independent regions or burbs, with a domain and a protocol stack assigned to each burb. Each protocol stack 40x has its own independent set of data structures, including routing information and protocol information. A given socket will be bound to a single protocol stack at creation time and no data can pass between protocol stacks 40 without going through proxy space. A proxy 50 therefore acts as the go-between for transfers between domains. Because of this, a malicious attacker who gains control of one of the regions is prevented from being able to compromise processes executing in other regions. Network separation and its application to an application level gateway is described in "SYSTEM AND METHOD FOR ACHIEVING NETWORK SEPARATION", U.S. Application No. 08/599,232, filed February 9, 1996 by Gooderum et al.

In the system shown in Figure 4, the in-bound and out-bound datagram processing of a security association continues to follow the conventions defined by the network separation model. Thus all datagrams received on or sent to a given burb remain in that burb once decrypted. In one such embodiment SADB socket 78 has been defined to have the type 'sadb'. Each proxy 50 that requires access to SADB socket 78 to execute its query as to whether the received message was encrypted must have create permission to the sadb type.

The following is list of specific requirements that a system such as is shown in Figure 4 must provide. Many of the requirements were discussed in the information provided earlier in this document.

1. Firewall applications may query the IPSEC subsystem to determine if traffic with a given address is guaranteed to be encrypted.
2. Receipt of an unencrypted datagram from an address that has a SA results in the datagram being dropped and an audit message being generated.
3. On receipt of encrypted protocol datagrams the SADB searches will be done using the SPI as the primary key. The source address will a secondary key. The SA returned by the search will be the SA which matches the SPI exactly and has the longest match with the address.

4. A search of the SADB for a SPI that finds an entry that is marked as SA for a dynamic IP will not consider the address in the search process.
5. A search of the SADB for a SPI that finds an entry that is marked as a SA for a tunnel mode connection will to consider the address if it is (0.0.0.0) i.e INADDR.
6. On receipt of a non-IPSEC datagram the SADB will be searched for an entry that matches the src address. If a SA is found the datagram will be dropped and an audit message sent.
7. SADB searches on output will be done using the DST address as key. If more than one SA entry in the SADB has that address the first one with the maximum address match will be returned.
8. The SADB must be structured so that searches are fast regardless if the search is done by SPI or by address.
9. The SADB must provide support for connections to a site with a fixed SPI but changing IP address. SA entries for such connections will be referred to as Dynamic Address Sites, or just Dynamic entries.
10. When a dynamic entry is found by a SPI search, the current datagram's SRC address, which is required to ensure that the return datagrams are properly encrypted, will be recorded in the SA only after the AH checking has passed successfully. (This is because if the address is recorded before AH passes then an attacker can cause return packets of an outgoing connection to be transmitted in the clear.)
11. A failure of an AH check on a dynamic entry results in an audit message.
12. In an embodiment where the firewall requires that all connections use both AH and ESP, on receipt the order should be AH first ESP second.
13. The processing structure on both input and output should try to minimize the number of SADB required lookups.

Returning to Figure 4, in one embodiment firewall 18 includes a crypto engine interface 80 used to encrypt an IPSEC payload. Crypto engine interface 80 may be connected to a software encryption engine 82 or to a hardware

encryption engine 84. Engines 82 and 84 perform the actual encryption function using, for example, DES-CBC. In addition, software encryption engine 82 may include the keyed MD5 algorithm used for AH.

In one embodiment, crypto engine interface 80 is a utility which provides a consistent interface between the software and hardware encryption engines. As shown in Figure 4, in one such embodiment interface 80 only supports the use of the use of hardware cryptographic engine 84 for IPSEC ESP processing. The significant design issue that interface 80 must deal with is that use of a hardware encryption engine requires that the processing be down in disjoint steps operating in different interrupt contexts as engine 84 completes the various processing steps.

The required information is stored in a request structure that is bound to the IP datagram being processed. The request is of type `crypto_request_t`. This structure is quite large and definitely does not contain a minimum state set.

In addition to the definition of the request data structure, this software implementing interface 80 provides two functions which isolate the decision of which cryptographic engine to use. The `crypt_des_encrypt` function is for use by the IP output processing to encrypt a datagram. The `crypt_des_decrypt` function is for use by the IP input processing to decrypt a datagram. If hardware encryption engine 84 is present and other hardware usage criteria are met the request is enqueued on a hardware processing queue and a return code indicating that the cryptographic processing is in progress is returned. If software engine 82 is used, the return code indicates that the cryptographic processing is complete. In the former case, the continuation of the IP processing is delayed until after hardware encryption is done. Otherwise it is completed as immediately in the same processing stream.

There are two software cryptographic engines 82 provided in the IPSEC software. One provides the MD5 algorithm used by the IPSEC AH processing, and the other provides the DES algorithm used by the IPSEC ESP processing. This software can be obtained from the US Government IPSEC implementation.

In one embodiment hardware cryptographic engine 84 is provided by a Cylink SafeNode processing board. The interface to this hardware card is provided by the Cylink device driver. A significant aspect of the Cylink card that plays a major part in the design of the IPSEC Cylink driver is that the card functions much like a low level subroutine interface and requires software support to initiate each processing step. Thus to encrypt or decrypt an individual datagram there are a minimum of two steps, one to set the DES initialization vector and one to do the encryption. Since the IP processing can not suspend itself and wait while the hardware completes and then be rescheduled by the hardware interrupt handler, in one embodiment a finite state machine is used to tie sequences of hardware processing elements together. In one such embodiment the interrupt handler looks at the current state, executes a defined after state function, transitions to the state and then executes that state's start function.

One function, `cyl_enqueue_request`, is used to initiate either an encrypt or a decrypt action. This function is designed to be called by cryptographic engine interface 80. All of the information required to initiate the processing as well as the function to be performed after the encryption operation is completed is provided in the request structure. This function will enqueue the request on the hardware request queue and start the hardware processing if necessary.

A system 30 which can be used for firewall-to-workstation encryption is shown in Figure 5. In Figure 5, system 30 includes a workstation 12 communicating through a firewall 14 to an unprotected network 16 such as the Internet. System 30 also includes a workstation 32 communicating directly with firewall 14 through unprotected network 16. Firewall 14 is an application level gateway incorporating IPSEC handling as described above. (It should be noted that IPSEC security cannot be used to authenticate the personal identity of the sender for a firewall to firewall transfer. When IPSEC is used, however, on a single user machine such as a portable personal computer, IPSEC usage should

be protected with a personal identification number (PIN). In these cases IPSEC can be used to help with user identification to the firewall.)

According to the IPSEC RFC's, you can use either tunnel or transport mode with this embodiment based on your security needs. In certain situations, the communications must be sent in tunnel mode to hide unregistered addresses.

Although specific embodiments have been illustrated and described herein, it will be appreciated by those of ordinary skill in the art that any arrangement which is calculated to achieve the same purpose may be substituted for the specific embodiment shown. This application is intended to cover any adaptations or variations of the present invention. Therefore, it is intended that this invention be limited only by the claims and the equivalents thereof.

What is claimed is:

1. A method of regulating the flow of messages through a firewall having a network protocol stack, wherein the network protocol stack includes an Internet Protocol (IP) layer, the method comprising the steps of:

determining, at the IP layer, if a message is encrypted;

if the message is not encrypted, passing the unencrypted message up the network protocol stack to an application level proxy; and

if the message is encrypted, decrypting the message and passing the decrypted message up the network protocol stack to the application level proxy, wherein the step of decrypting the message includes the step of executing a procedure at the IP layer to decrypt the message.

2. A method of authenticating the sender of a message within a computer system having a network protocol stack, wherein the network protocol stack includes an Internet Protocol (IP) layer, the method comprising the steps of:

determining, at the IP layer, if the message is encrypted;

if the message is encrypted, decrypting the message, wherein the step of decrypting the message includes the step of executing a process at the IP layer to decrypt the message;

passing the decrypted message up the network protocol stack to an application level proxy;

determining an authentication protocol appropriate for the message; and

executing the authentication protocol to authenticate the sender of the message.

3. The method according to claim 2 wherein the step of determining an authentication protocol appropriate for the message includes the steps of:

determining a source IP address associated with the message; and

determining the authentication protocol associated with the source IP address.

4. The method according to claim 2 wherein the message includes security parameters index and wherein the step of determining an authentication protocol appropriate for the message includes the steps of:

determining the authentication protocol associated with a dynamic IP address, wherein the step of determining the authentication protocol includes the step of looking up a security association based on the security parameters index;

determining a current address associated with the dynamic source IP address; and

binding the current address to the security parameters index.

5. A firewall, comprising:

a first communications interface;

a second communications interface;

a network protocol stack connected to the first and the second communications interfaces, wherein the network protocol stack includes an Internet Protocol (IP) layer and a transport layer;

a decryption procedure, operating at the IP layer, wherein the decryption procedure decrypts encrypted messages received at one of said first and second communications interfaces and outputs decrypted messages; and

a proxy, connected to the transport layer of said network protocol stack, wherein the proxy receives decrypted messages from the decryption procedure and executes an authentication protocol based on the content of the decrypted message.

6. A firewall, comprising:

a first communications interface;

a second communications interface;

a first network protocol stack connected to the first communications interface, wherein the first network protocol stack includes an Internet Protocol (IP) layer and a transport layer;

a second network protocol stack connected to the second communications interface, wherein the second network protocol stack includes an Internet Protocol (IP) layer and a transport layer;

a decryption procedure, operating at the IP layer of the first network protocol stack, the decryption procedure receiving encrypted messages received by said first communications interface and outputting decrypted messages; and

a proxy, connected to the transport layers of said first and second network protocol stacks, the proxy receiving decrypted messages from the decryption procedure and executing an authentication protocol based on the content of the decrypted message.

7. The firewall according to claim 6 wherein the firewall further includes:

a third communications interface; and

a third network protocol stack connected to the third communications interface and to the proxy, wherein the third network protocol stack includes an Internet Protocol (IP) layer and a transport layer and wherein the second and third network protocol stacks are restricted to first and second burbs, respectively.

8. A method of establishing a virtual private network between a first and a second network, wherein each network includes an application level gateway firewall which uses a proxy operating at the application layer to process traffic through the firewall, wherein each firewall includes a network protocol stack and wherein each network protocol stack includes an Internet Protocol (IP) layer, the method comprising the steps of:

transferring a connection request from the first network to the second network;

determining, at the IP layer of the network protocol stack of the second network's firewall, if the connection request is encrypted;

if the connection request is encrypted, decrypting the request, wherein the step of decrypting the request includes the step of executing a procedure at the IP layer of the second network's firewall to decrypt the message;

passing the connection request up the network protocol stack to an application level proxy;

determining an authentication protocol appropriate for the connection request;

executing the authentication protocol to authenticate the connection request; and

if the connection request is authentic, establishing an active connection between the first and second networks.

9. The method according to claim 8 wherein the step of executing the authentication protocol includes the step of executing program code within the firewall of the second network to mimic a challenge/response protocol executing on a server internal to the second network.

10. The method according to claim 8 wherein the step of executing the authentication protocol includes the step of executing program code to execute the authentication protocol in line to the session.

11. The method according to claim 8 wherein the step of determining an authentication protocol includes the step of determining if the connection request arrived encrypted and selecting the authentication protocol based on whether the connection request was encrypted or not encrypted.

The Patent Office

28

Application No: GB 9719816.2 **Examiner:** B.J.SPEAR
Claims searched: 1-11 **Date of search:** 21 January 1998

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.P): H4P (PPEB,PDCSA,PDCSC)

Int Cl (Ed.6): H04L 9/00, 9/32, 29/06, 29/08

Other: Online:WPI, INSPEC

Documents considered to be relevant:

Category	Identity of document and relevant passage		Relevant to claims
XP	WO97/26734A1	(Raptor Systems) Whole document, eg Figs 1,3 and pages 6-12	1,2.5,6,8 at least
XP	WO97/26731A1	(Raptor Systems) Whole document, eg Figs 1,3 and pages 7-12	1,2.5,6,8 at least
XP	WO97/26735A1	(Raptor Systems) Whole document, eg Figs 1,3 and pages 4-10	1,2.5,6,8 at least
XP	WO97/23972A1	(V-ONE Corp) Whole document, eg Figs 1,2 and claim 1.	1,2.5,6,8 at least
XP	WO97/13340A1	(Digital Secured Networks) Whole document, eg pages 7-13	1,2.5,6,8 at least

X Document indicating lack of novelty or inventive step
Y Document indicating lack of inventive step if combined with one or more other documents of same category.
& Member of the same patent family
A Document indicating technological background and/or state of the art.
P Document published on or after the declared priority date but before the filing date of this invention.
E Patent document published on or after, but with priority date earlier than, the filing date of this application.

Electronic Acknowledgement Receipt

EFS ID:	1304414
Application Number:	10702486
International Application Number:	
Confirmation Number:	8949
Title of Invention:	Method for establishing secure communication link between computers of virtual private network
First Named Inventor/Applicant Name:	Victor Larson
Customer Number:	22907
Filer:	Steve S. Chang
Filer Authorized By:	
Attorney Docket Number:	000479.00112
Receipt Date:	09-NOV-2006
Filing Date:	07-NOV-2003
Time Stamp:	17:39:45
Application Type:	Utility

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)	Multi Part /.zip	Pages (if appl.)
1	Information Disclosure Statement (IDS) Filed	00112IDS.pdf	138398	no	5
Warnings:					

Information:					
This is not an USPTO supplied IDS fillable form					
2	NPL Documents	wellsref.pdf	92941	no	1
Warnings:					
Information:					
3	Foreign Reference	WO9827783.pdf	851073	no	24
Warnings:					
Information:					
4	Foreign Reference	GB2334181.pdf	434487	no	15
Warnings:					
Information:					
5	Foreign Reference	EP0814589.pdf	1098918	no	20
Warnings:					
Information:					
6	Foreign Reference	EP0838930.pdf	1731451	no	35
Warnings:					
Information:					
7	Foreign Reference	GB2317792.pdf	1262522	no	35
Warnings:					
Information:					
8	NPL Documents	Davilaarticle.pdf	807564	no	18
Warnings:					
Information:					
		Total Files Size (in bytes):	6417354		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	4355237	secure computer network address	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/15 10:29
L2	5884932	secure domain name	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/15 10:30
L3	1902353	l1 and l2	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/15 10:31
L4	456593	l3 and (secure domain name or SDN).ti,ab,clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/15 10:32
L5	867675	(virtual private network or VPN).ti, ab,clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/15 10:32
L6	50519	l4 and l5	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/15 10:34
L8	1996089	(access$3 secure network address). ti,ab,clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/15 10:36
L9	610693	l8 and l2	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/15 10:37

EAST Search History

L10	144386	l9 and l5	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/15 10:38
L11	823930	709/225, "226", "227", "228", "229".ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/15 10:53
L12	21805	l10 and l11	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/15 10:53
L13	5042	l12 and @ad<="19981030"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/15 10:59
L14	2875	l13 and (secure communication link).ti,ab,clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/15 11:00
L15	2875	l14 and l5	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/15 11:01
L16	102	l15 and secur$3.ti,ab,clm. and network.ti,ab,clm. and address.ti,ab,clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/15 11:02
L17	617	establish$3 adj3 secur$4 adj3 link	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/15 11:03

EAST Search History

L18	119	l11 and l17	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/15 11:03
L19	8	l18 and (virtual adj3 private adj3 network).ti,ab,clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/15 11:04

Sign in

Google

Web Images Video News Maps more »

secure domain name Search Advanced Search Preferences

Web

Results 1 - 10 of about 17,400,000 for secure domain name. (0.27 seconds)

Domain Names
NetworkSolutions.com Register names at Network Solutions The pioneer of domains on the web.

Domain Name
www.OfficeLive.com Free **Domain Name**, Web Site, Company Branded Email & More from Microsoft

Secure A Domain Name
How to get a **domain name** cheap or why you may not even want one.
www.knowledgehound.com/khhow2s/a**domain**.htm - 19k - Cached - Similar pages

Domain Name, email and web hosting services from NameSecure ...
An ICANN-accredited **domain name** registrar, offering a free suite of services including web forwarding, e-mail forwarding, and one-page mini-sites.
www.**namesecure**.com/ - 32k - Cached - Similar pages

Yahoo! Domains: **Secure Domain Name** Registration
Yahoo! Domains includes **domain name** registration, 24-hour customer support, a starter web page, **domain** locking to prevent hijacking, DNS management, ...
smallbusiness.yahoo.com/**domains**/ - 25k - Cached - Similar pages

Secure Florida - How to **Secure** Your **Domain Name**
How to **Secure** Your **Domain Name**. As a general rule, owning a **domain name** that fits your business is imperative to your success on the Internet. ...
www.**secure**florida.org/index.php?submenu=How_To&src=gendocs&link=How%20to%20**Secure**%20Your%20**Domain**... - 28k - Cached - Similar pages

Internet **Domain Name** Registration, **Domain** Transfers. Your **domain** ...
Bulk pricing and private **domain name** registration options. ... **Secure Domain** Locking FREE! Total DNS Control FREE! Status Alerts And much more! ...
www.godaddy.com/ - 93k - Cached - Similar pages

[PDF] **Secure Domain Name** System (DNS) Deployment Guide
File Format: PDF/Adobe Acrobat - View as HTML
This publication seeks to assist organizations in understanding the **secure** deployment of **Domain Name**. System (DNS) services in an enterprise. ...
csrc.nist.gov/publications/nistpubs/800-81/SP800-81.pdf - Similar pages

DNSSEC - DNS Security Extensions
DNSSEC services protect against most of the threats to the **Domain Name** System. ... **Secure Domain Name** System (DNS) Deployment Guide ...
www.dnssec.net/ - 30k - Cached - Similar pages

ISC BIND
BIND (Berkeley Internet **Name Domain**) is an implementation of the **Domain Name** ... Draft NIST Special Publication 800-81 **Secure Domain Name** System (DNS) ...
www.isc.org/sw/bind/ - 11k - Cached - Similar pages

rfc 3007
Abstract This document proposes a method for performing **secure Domain Name** System (DNS) dynamic updates. The method described here is intended to be ...
www.ietf.org/rfc/rfc3007.txt - 18k - Cached - Similar pages

Secure Domain Registration Award from World Association of **Domain ...**
Moniker.com, fast growing ICANN accredited **domain name** registrar, wins special **secure domain** registration award from World Association of **Domain Name ...**
www.moniker.com/pressreleases/moniker-pr-2005-10-27.jsp - 18k - Cached - Similar pages

Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

Terms used **secure domain name network address** Found **1,806** of **192,172**

Sort results by: relevance
Display results: expanded form
Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next
Best 200 shown Relevance scale

1 A self-configuring and self-administering name system with dynamic address assignment
February 2002 **ACM Transactions on Internet Technology (TOIT)**, Volume 2 Issue 1
Publisher: ACM Press
Full text available: pdf(908.57 KB) Additional Information: full citation, abstract, references, index terms, review

In this article we present a distributed system that stores name-to-address bindings and provides name resolution to a network of computers. This name system consists of a network of name services that are individually self-configuring and self-administering. The name service consists of an agent program that works in conjunction with the current implementation of the Domain Name System (DNS) program. The DNS agent program automatically configures the Berkeley Internet Name Domain (BIND) process ...

Keywords: Berkeley Internet Name Domain, dynamic reconfiguration, name-to-name address binding, self-administering systems, self-configuring systems

2 Naming in dynamic networks: Names, addresses and identities in ambient networks
Bengt Ahlgren, Lars Eggert, Börje Ohlman, Jarno Rajahalme, Andreas Schieder
September 2005 **Proceedings of the 1st ACM workshop on Dynamic interconnection of networks DIN '05**
Publisher: ACM Press
Full text available: pdf(644.14 KB) Additional Information: full citation, abstract, references, index terms

Ambient Networks interconnect independent realms that may use different local network technologies and may belong to different administrative or legal entities. At the core of these advanced internetworking concepts is a flexible naming architecture based on dynamic indirections between names, addresses and identities. This paper gives an overview of the connectivity abstractions of *Ambient Networks* and then describes its naming architecture in detail, comparing and contrasting the ...

Keywords: addressing, ambient networks, bindings, identities, indirection, internetworking, naming, resolution

3 A public-key based secure mobile IP
John Zao, Joshua Gahm, Gregory Troxel, Matthew Condell, Pam Helinek, Nina Yuan, Isidro Castineyra, Stephen Kent
October 1999 **Wireless Networks**, Volume 5 Issue 5
Publisher: Kluwer Academic Publishers
Full text available: Additional Information:

pdf(255.65 KB) full citation, references, citings, index terms

4 An architecture for secure wide-area service discovery

Todd D. Hodes, Steven E. Czerwinski, Ben Y. Zhao, Anthony D. Joseph, Randy H. Katz
March 2002 **Wireless Networks**, Volume 8 Issue 2/3
Publisher: Kluwer Academic Publishers

Full text available: pdf(365.68 KB) Additional Information: full citation, abstract, references, citings, index terms

The widespread deployment of inexpensive communications technology, computational resources in the networking infrastructure, and network-enabled end devices poses an interesting problem for end users: how to locate a particular network service or device out of hundreds of thousands of accessible services and devices. This paper presents the architecture and implementation of a secure wide-area Service Discovery Service (SDS). Service providers use the SDS to advertise descriptions of available ...

Keywords: location services, name lookup, network protocols, service discovery

5 A public-key based secure mobile IP

John Zao, Stephen Kent, Joshua Gahm, Gregory Troxel, Matthew Condell, Pam Helinek, Nina Yuan, Isidro Castineyra
September 1997 **Proceedings of the 3rd annual ACM/IEEE international conference on Mobile computing and networking**
Publisher: ACM Press
Full text available: pdf(1.95 MB) Additional Information: full citation, references, citings

6 Secure virtual enclaves: Supporting coalition use of distributed application technologies

May 2001 **ACM Transactions on Information and System Security (TISSEC)**, Volume 4 Issue 2
Publisher: ACM Press

Full text available: pdf(462.10 KB) Additional Information: full citation, abstract, references, index terms, review

The Secure Virtual Enclaves (SVE) collaboration infrastructure allows multiple organizations to share their distributed application objects, while respecting organizational autonomy over local resources. The infrastructure is transparent to applications, which may be accessed via a web server, or may be based on Java or Microsoft's DCOM. The SVE infrastructure is implemented in middleware, with no modifications to COTS operating systems or network protocols. The system enables dynamic updates to ...

Keywords: Access control, coalition, collaborative system, group communication, middleware, security policy

7 Secure and mobile networking

Vipul Gupta, Gabriel Montenegro
December 1998 **Mobile Networks and Applications**, Volume 3 Issue 4
Publisher: Kluwer Academic Publishers

Full text available: pdf(223.39 KB) Additional Information: full citation, abstract, references, citings, index terms

The IETF Mobile IP protocol is a significant step towards enabling nomadic Internet users. It allows a mobile node to maintain and use the same IP address even as it changes its point of attachment to the Internet. Mobility implies higher security risks than static operation. Portable devices may be stolen or their traffic may, at times, pass through

links with questionable security characteristics. Most commercial organizations use some combination of source-filtering routers, sophisticate ...

8 Securing wireless applications: ESCORT: a decentralized and localized access control system for mobile wireless access to secured domains

Jiejun Kong, Shirshanka Das, Edward Tsai, Mario Gerla
September 2003 **Proceedings of the 2003 ACM workshop on Wireless security**
Publisher: ACM Press
Full text available: pdf(401.72 KB) Additional Information: full citation, abstract, references, index terms

In this work we design and implement ESCORT, a *backward compatible*, *efficient*, and *secure* access control system, to facilitate mobile wireless access to secured wireless LANs. In mobile environments, a mobile guest may frequently roam into foreign domains while demanding critical network services. ESCORT provides instant yet secure access to the mobile guest based on the concept of "*escort*", which refers to a special network object with four distinct properties: (1) T ...

Keywords: decentralized access control, identity privacy, location privacy, mobile privacy, wireless security

9 An architecture for a secure service discovery service

Steven E. Czerwinski, Ben Y. Zhao, Todd D. Hodes, Anthony D. Joseph, Randy H. Katz
August 1999 **Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking**
Publisher: ACM Press
Full text available: pdf(1.47 MB) Additional Information: full citation, references, citings, index terms

10 DoS and authentication in wireless public access networks

Daniel B. Faria, David R. Cheriton
September 2002 **Proceedings of the 3rd ACM workshop on Wireless security WiSE '02**
Publisher: ACM Press
Full text available: pdf(272.24 KB) Additional Information: full citation, abstract, references, citings, index terms

As WEP has been shown to be vulnerable to multiple attacks, a huge effort has been placed on specifying an access control mechanism to be used in wireless installations. However, properties of the wireless environment have been exploited to perform multiple DoS attacks against current solutions, such as 802.11/802.1X. In this paper we discuss the main wireless idiosyncrasies and the need for taking them into account when designing an access control mechanism that can be used in both wireless and ...

Keywords: DoS, security, wireless networks

11 The design and implementation of a next generation name service for the internet

Venugopalan Ramasubramanian, Emin Gün Sirer
August 2004 **ACM SIGCOMM Computer Communication Review , Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications SIGCOMM '04**, Volume 34 Issue 4
Publisher: ACM Press
Full text available: pdf(472.93 KB) Additional Information: full citation, abstract, references, citings, index terms

Name services are critical for mapping logical resource names to physical resources in large-scale distributed systems. The Domain Name System (DNS) used on the Internet, however, is slow, vulnerable to denial of service attacks, and does not support fast updates. These problems stem fundamentally from the structure of the legacy DNS.This paper describes the design and implementation of the Cooperative Domain Name System

(CoDoNS), a novel name service, which provides high lookup performance thro ...

Keywords: DNS, peer to peer, proactive caching

12 An end-to-end approach to host mobility

Alex C. Snoeren, Hari Balakrishnan
August 2000 **Proceedings of the 6th annual international conference on Mobile computing and networking**
Publisher: ACM Press

Full text available: pdf(1.35 MB) Additional Information: full citation, abstract, references, citings, index terms

We present the design and implementation of an end-to-end architecture for Internet host mobility using dynamic updates to the Domain Name System (DNS) to track host location. Existing TCP connections are retained using secure and efficient connection migration, enabling established connections to seamlessly negotiate a change in endpoint IP addresses without the need for a third party. Our architecture is secure—name updates are effected via the secure DNS update protocol, while TCP ...

13 Towards a secure platform for distributed mobile object computing

Marc Lacoste
April 2000 **ACM SIGOPS Operating Systems Review**, Volume 34 Issue 2
Publisher: ACM Press
Full text available: pdf(1.42 MB) Additional Information: full citation, abstract, index terms

We present some issues relevant to the design of a secure platform for distributed mobile computing, that goes beyond existing ad-hoc approaches to software mobility. This platform aims to support wide-area computing applications such as active network infrastructures or network supervision tools. Our contribution is two-fold: the first part of the paper is a survey of the security features of a few languages and virtual machines as regards authentication, access control, and communications secu ...

14 Integrating security in a large distributed system

M. Satyanarayanan
August 1989 **ACM Transactions on Computer Systems (TOCS)**, Volume 7 Issue 3
Publisher: ACM Press

Full text available: pdf(2.90 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Andrew is a distributed computing environment that is a synthesis of the personal computing and timesharing paradigms. When mature, it is expected to encompass over 5,000 workstations spanning the Carnegie Mellon University campus. This paper examines the security issues that arise in such an environment and describes the mechanisms that have been developed to address them. These mechanisms include the logical and physical separation of servers and clients, support for secure communication ...

15 MobileNAT: a new technique for mobility across heterogeneous address spaces

Milind Buddhikot, Adiseshu Hari, Kundan Singh, Scott Miller
June 2005 **Mobile Networks and Applications**, Volume 10 Issue 3
Publisher: ACM Press
Full text available: pdf(1.60 MB) Additional Information: full citation, abstract, references, index terms

We propose a new network layer mobility architecture called MOBILENAT to efficiently support micro and macro-mobility in and across heterogeneous address spaces common in emerging public networks. The key ideas in this architecture are as follows: (1) Use of two IP addresses - an invariant virtual IP address for host identification at the application layer and an actual routable address at the network layer that changes due to mobility. Since physical address has routing significance only within ...

Keywords: MOBILENAT, design, experimentation, mobility

16 LegionFS: a secure and scalable file system supporting cross-domain high-performance applications

Brian S. White, Michael Walker, Marty Humphrey, Andrew S. Grimshaw
November 2001 **Proceedings of the 2001 ACM/IEEE conference on Supercomputing (CDROM)**
Publisher: ACM Press

Full text available: pdf(499.88 KB) Additional Information: full citation, abstract, references, citings, index terms

Realizing that current file systems can not cope with the diverse requirements of wide-area collaborations, researchers have developed data access facilities to meet their needs. Recent work has focused on comprehensive data access architectures. In order to fulfill the evolving requirements in this environment, we suggest a more fully-integrated architecture built upon the fundamental tenets of naming, security, scalability, extensibility, and adaptability. These form the underpinning of the Le ...

17 Hacking and innovation: Explorations in namespace: white-hat hacking across the domain name system

Dan Kaminsky
June 2006 **Communications of the ACM**, Volume 49 Issue 6
Publisher: ACM Press

Full text available: pdf(291.36 KB) html(28.87 KB) Additional Information: full citation, abstract, references, index terms

DNS cache scanning across a sample set of more than 500,000 name servers revealed the extent of last year's Sony rootkit infestation on client machines.

18 Encryption and Secure Computer Networks

Gerald J. Popek, Charles S. Kline
December 1979 **ACM Computing Surveys (CSUR)**, Volume 11 Issue 4
Publisher: ACM Press

Full text available: pdf(2.50 MB) Additional Information: full citation, references, citings, index terms

19 Mobile networking in the Internet

Charles E. Perkins
December 1998 **Mobile Networks and Applications**, Volume 3 Issue 4
Publisher: Kluwer Academic Publishers

Full text available: pdf(166.90 KB) Additional Information: full citation, abstract, references, citings, index terms

Computers capable of attaching to the Internet from many places are likely to grow in popularity until they dominate the population of the Internet. Consequently, protocol research has shifted into high gear to develop appropriate network protocols for supporting mobility. This introductory article attempts to outline some of the many promising and interesting research directions. The papers in this special issue indicate the diversity of viewpoints within the research community, and it is ...

20 Applications: YouServ: a web-hosting and content sharing tool for the masses

Roberto J. Bayardo Jr., Rakesh Agrawal, Daniel Gruhl, Amit Somani
May 2002 **Proceedings of the 11th international conference on World Wide Web**
Publisher: ACM Press

Full text available: pdf(238.48 KB) Additional Information: full citation, abstract, references, citings, index terms

YouServ is a system that allows its users to pool existing desktop computing resources for *high availability* web hosting and file sharing. By exploiting standard web and internet

protocols (e.g. HTTP and DNS), YouServ does not require those who access YouServ-published content to install special purpose software. Because it requires minimal server-side resources and administration, YouServ can be provided at a very low cost. We describe the design, implementation, and a successful intrane ...

Keywords: decentralized systems, p2p, peer-to-peer networks, web hosting

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Terms used **secure domain name network address** Found **1,806** of **192,172**

Sort results by: relevance
Display results: expanded form
Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next
Best 200 shown Relevance scale

1 A self-configuring and self-administering name system with dynamic address assignment
February 2002 **ACM Transactions on Internet Technology (TOIT)**, Volume 2 Issue 1
Publisher: ACM Press
Full text available: pdf(908.57 KB) Additional Information: full citation, abstract, references, index terms, review

In this article we present a distributed system that stores name-to-address bindings and provides name resolution to a network of computers. This name system consists of a network of name services that are individually self-configuring and self-administering. The name service consists of an agent program that works in conjunction with the current implementation of the Domain Name System (DNS) program. The DNS agent program automatically configures the Berkeley Internet Name Domain (BIND) process ...

Keywords: Berkeley Internet Name Domain, dynamic reconfiguration, name-to-name address binding, self-administering systems, self-configuring systems

2 Naming in dynamic networks: Names, addresses and identities in ambient networks
Bengt Ahlgren, Lars Eggert, Börje Ohlman, Jarno Rajahalme, Andreas Schieder
September 2005 **Proceedings of the 1st ACM workshop on Dynamic interconnection of networks DIN '05**
Publisher: ACM Press
Full text available: pdf(644.14 KB) Additional Information: full citation, abstract, references, index terms

Ambient Networks interconnect independent realms that may use different local network technologies and may belong to different administrative or legal entities. At the core of these advanced internetworking concepts is a flexible naming architecture based on dynamic indirections between names, addresses and identities. This paper gives an overview of the connectivity abstractions of *Ambient Networks* and then describes its naming architecture in detail, comparing and contrasting the ...

Keywords: addressing, ambient networks, bindings, identities, indirection, internetworking, naming, resolution

3 A public-key based secure mobile IP
John Zao, Joshua Gahm, Gregory Troxel, Matthew Condell, Pam Helinek, Nina Yuan, Isidro Castineyra, Stephen Kent
October 1999 **Wireless Networks**, Volume 5 Issue 5
Publisher: Kluwer Academic Publishers
Full text available: Additional Information:

pdf(255.65 KB) full citation, references, citings, index terms

4 An architecture for secure wide-area service discovery

Todd D. Hodes, Steven E. Czerwinski, Ben Y. Zhao, Anthony D. Joseph, Randy H. Katz
March 2002 **Wireless Networks**, Volume 8 Issue 2/3
Publisher: Kluwer Academic Publishers
Full text available: pdf(365.68 KB) Additional Information: full citation, abstract, references, citings, index terms

The widespread deployment of inexpensive communications technology, computational resources in the networking infrastructure, and network-enabled end devices poses an interesting problem for end users: how to locate a particular network service or device out of hundreds of thousands of accessible services and devices. This paper presents the architecture and implementation of a secure wide-area Service Discovery Service (SDS). Service providers use the SDS to advertise descriptions of available ...

Keywords: location services, name lookup, network protocols, service discovery

5 A public-key based secure mobile IP

John Zao, Stephen Kent, Joshua Gahm, Gregory Troxel, Matthew Condell, Pam Helinek, Nina Yuan, Isidro Castineyra
September 1997 **Proceedings of the 3rd annual ACM/IEEE international conference on Mobile computing and networking**
Publisher: ACM Press
Full text available: pdf(1.95 MB) Additional Information: full citation, references, citings

6 Secure virtual enclaves: Supporting coalition use of distributed application technologies

May 2001 **ACM Transactions on Information and System Security (TISSEC)**, Volume 4 Issue 2
Publisher: ACM Press
Full text available: pdf(462.10 KB) Additional Information: full citation, abstract, references, index terms, review

The Secure Virtual Enclaves (SVE) collaboration infrastructure allows multiple organizations to share their distributed application objects, while respecting organizational autonomy over local resources. The infrastructure is transparent to applications, which may be accessed via a web server, or may be based on Java or Microsoft's DCOM. The SVE infrastructure is implemented in middleware, with no modifications to COTS operating systems or network protocols. The system enables dynamic updates to ...

Keywords: Access control, coalition, collaborative system, group communication, middleware, security policy

7 Secure and mobile networking

Vipul Gupta, Gabriel Montenegro
December 1998 **Mobile Networks and Applications**, Volume 3 Issue 4
Publisher: Kluwer Academic Publishers
Full text available: pdf(223.39 KB) Additional Information: full citation, abstract, references, citings, index terms

The IETF Mobile IP protocol is a significant step towards enabling nomadic Internet users. It allows a mobile node to maintain and use the same IP address even as it changes its point of attachment to the Internet. Mobility implies higher security risks than static operation. Portable devices may be stolen or their traffic may, at times, pass through

links with questionable security characteristics. Most commercial organizations use some combination of source-filtering routers, sophisticate ...

8 Securing wireless applications: ESCORT: a decentralized and localized access control system for mobile wireless access to secured domains

Jiejun Kong, Shirshanka Das, Edward Tsai, Mario Gerla
September 2003 **Proceedings of the 2003 ACM workshop on Wireless security**
Publisher: ACM Press
Full text available: pdf(401.72 KB) Additional Information: full citation, abstract, references, index terms

In this work we design and implement ESCORT, a *backward compatible*, *efficient*, and *secure* access control system, to facilitate mobile wireless access to secured wireless LANs. In mobile environments, a mobile guest may frequently roam into foreign domains while demanding critical network services. ESCORT provides instant yet secure access to the mobile guest based on the concept of "*escort*", which refers to a special network object with four distinct properties: (1) T ...

Keywords: decentralized access control, identity privacy, location privacy, mobile privacy, wireless security

9 An architecture for a secure service discovery service

Steven E. Czerwinski, Ben Y. Zhao, Todd D. Hodes, Anthony D. Joseph, Randy H. Katz
August 1999 **Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking**
Publisher: ACM Press
Full text available: pdf(1.47 MB) Additional Information: full citation, references, citings, index terms

10 DoS and authentication in wireless public access networks

Daniel B. Faria, David R. Cheriton
September 2002 **Proceedings of the 3rd ACM workshop on Wireless security WiSE '02**
Publisher: ACM Press
Full text available: pdf(272.24 KB) Additional Information: full citation, abstract, references, citings, index terms

As WEP has been shown to be vulnerable to multiple attacks, a huge effort has been placed on specifying an access control mechanism to be used in wireless installations. However, properties of the wireless environment have been exploited to perform multiple DoS attacks against current solutions, such as 802.11/802.1X. In this paper we discuss the main wireless idiosyncrasies and the need for taking them into account when designing an access control mechanism that can be used in both wireless and ...

Keywords: DoS, security, wireless networks

11 The design and implementation of a next generation name service for the internet

Venugopalan Ramasubramanian, Emin Gün Sirer
August 2004 **ACM SIGCOMM Computer Communication Review , Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications SIGCOMM '04**, Volume 34 Issue 4
Publisher: ACM Press
Full text available: pdf(472.93 KB) Additional Information: full citation, abstract, references, citings, index terms

Name services are critical for mapping logical resource names to physical resources in large-scale distributed systems. The Domain Name System (DNS) used on the Internet, however, is slow, vulnerable to denial of service attacks, and does not support fast updates. These problems stem fundamentally from the structure of the legacy DNS.This paper describes the design and implementation of the Cooperative Domain Name System

(CoDoNS), a novel name service, which provides high lookup performance thro ...

Keywords: DNS, peer to peer, proactive caching

12 An end-to-end approach to host mobility

Alex C. Snoeren, Hari Balakrishnan
August 2000 **Proceedings of the 6th annual international conference on Mobile computing and networking**
Publisher: ACM Press

Full text available: pdf(1.35 MB) Additional Information: full citation, abstract, references, citings, index terms

We present the design and implementation of an end-to-end architecture for Internet host mobility using dynamic updates to the Domain Name System (DNS) to track host location. Existing TCP connections are retained using secure and efficient connection migration, enabling established connections to seamlessly negotiate a change in endpoint IP addresses without the need for a third party. Our architecture is secure—name updates are effected via the secure DNS update protocol, while TCP ...

13 Towards a secure platform for distributed mobile object computing

Marc Lacoste
April 2000 **ACM SIGOPS Operating Systems Review**, Volume 34 Issue 2
Publisher: ACM Press
Full text available: pdf(1.42 MB) Additional Information: full citation, abstract, index terms

We present some issues relevant to the design of a secure platform for distributed mobile computing, that goes beyond existing ad-hoc approaches to software mobility. This platform aims to support wide-area computing applications such as active network infrastructures or network supervision tools. Our contribution is two-fold: the first part of the paper is a survey of the security features of a few languages and virtual machines as regards authentication, access control, and communications secu ...

14 Integrating security in a large distributed system

M. Satyanarayanan
August 1989 **ACM Transactions on Computer Systems (TOCS)**, Volume 7 Issue 3
Publisher: ACM Press

Full text available: pdf(2.90 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Andrew is a distributed computing environment that is a synthesis of the personal computing and timesharing paradigms. When mature, it is expected to encompass over 5,000 workstations spanning the Carnegie Mellon University campus. This paper examines the security issues that arise in such an environment and describes the mechanisms that have been developed to address them. These mechanisms include the logical and physical separation of servers and clients, support for secure communication ...

15 MobileNAT: a new technique for mobility across heterogeneous address spaces

Milind Buddhikot, Adiseshu Hari, Kundan Singh, Scott Miller
June 2005 **Mobile Networks and Applications**, Volume 10 Issue 3
Publisher: ACM Press
Full text available: pdf(1.60 MB) Additional Information: full citation, abstract, references, index terms

We propose a new network layer mobility architecture called MOBILENAT to efficiently support micro and macro-mobility in and across heterogeneous address spaces common in emerging public networks. The key ideas in this architecture are as follows: (1) Use of two IP addresses - an invariant virtual IP address for host identification at the application layer and an actual routable address at the network layer that changes due to mobility. Since physical address has routing significance only within ...

Keywords: MOBILENAT, design, experimentation, mobility

16 LegionFS: a secure and scalable file system supporting cross-domain high-performance applications

Brian S. White, Michael Walker, Marty Humphrey, Andrew S. Grimshaw

November 2001 **Proceedings of the 2001 ACM/IEEE conference on Supercomputing (CDROM)**

Publisher: ACM Press

Full text available: pdf(499.88 KB) Additional Information: full citation, abstract, references, citings, index terms

Realizing that current file systems can not cope with the diverse requirements of wide-area collaborations, researchers have developed data access facilities to meet their needs. Recent work has focused on comprehensive data access architectures. In order to fulfill the evolving requirements in this environment, we suggest a more fully-integrated architecture built upon the fundamental tenets of naming, security, scalability, extensibility, and adaptability. These form the underpinning of the Le ...

17 Hacking and innovation: Explorations in namespace: white-hat hacking across the domain name system

Dan Kaminsky

June 2006 **Communications of the ACM**, Volume 49 Issue 6

Publisher: ACM Press

Full text available: pdf(291.36 KB) html(28.87 KB) Additional Information: full citation, abstract, references, index terms

DNS cache scanning across a sample set of more than 500,000 name servers revealed the extent of last year's Sony rootkit infestation on client machines.

18 Encryption and Secure Computer Networks

Gerald J. Popek, Charles S. Kline

December 1979 **ACM Computing Surveys (CSUR)**, Volume 11 Issue 4

Publisher: ACM Press

Full text available: pdf(2.50 MB) Additional Information: full citation, references, citings, index terms

19 Mobile networking in the Internet

Charles E. Perkins

December 1998 **Mobile Networks and Applications**, Volume 3 Issue 4

Publisher: Kluwer Academic Publishers

Full text available: pdf(166.90 KB) Additional Information: full citation, abstract, references, citings, index terms

Computers capable of attaching to the Internet from many places are likely to grow in popularity until they dominate the population of the Internet. Consequently, protocol research has shifted into high gear to develop appropriate network protocols for supporting mobility. This introductory article attempts to outline some of the many promising and interesting research directions. The papers in this special issue indicate the diversity of viewpoints within the research community, and it is ...

20 Applications: YouServ: a web-hosting and content sharing tool for the masses

Roberto J. Bayardo Jr., Rakesh Agrawal, Daniel Gruhl, Amit Somani

May 2002 **Proceedings of the 11th international conference on World Wide Web**

Publisher: ACM Press

Full text available: pdf(238.48 KB) Additional Information: full citation, abstract, references, citings, index terms

YouServ is a system that allows its users to pool existing desktop computing resources for *high availability* web hosting and file sharing. By exploiting standard web and internet

protocols (e.g. HTTP and DNS), YouServ does not require those who access YouServ-published content to install special purpose software. Because it requires minimal server-side resources and administration, YouServ can be provided at a very low cost. We describe the design, implementation, and a successful intrane ...

Keywords: decentralized systems, p2p, peer-to-peer networks, web hosting

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

NOTICE OF ALLOWANCE AND FEE(S) DUE

22907 7590 11/21/2006

BANNER & WITCOFF
1001 G STREET N W
SUITE 1100
WASHINGTON, DC 20001

EXAMINER	
LIM, KRISNA	
ART UNIT	PAPER NUMBER
2153	

DATE MAILED: 11/21/2006

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/702,486	11/07/2003	Victor Larson	000479.00112	8949

TITLE OF INVENTION: METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN COMPUTERS OF VIRTUAL PRIVATE NETWORK

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1400	$300	$0	$1700	02/21/2007

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the SMALL ENTITY status shown above.

If the SMALL ENTITY is shown as YES, verify your current SMALL ENTITY status:

A. If the status is the same, pay the TOTAL FEE(S) DUE shown above.

B. If the status above is to be removed, check box 5b on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and twice the amount of the ISSUE FEE shown above, or

If the SMALL ENTITY is shown as NO:

A. Pay TOTAL FEE(S) DUE shown above, or

B. If applicant claimed SMALL ENTITY status before, or is now claiming SMALL ENTITY status, check box 5a on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and 1/2 the ISSUE FEE shown above.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

PTOL-85 (Rev. 07/06) Approved for use through 04/30/2007.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

22907 7590 11/21/2006

BANNER & WITCOFF
1001 G STREET N W
SUITE 1100
WASHINGTON, DC 20001

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission
I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/702,486	11/07/2003	Victor Larson	000479.00112	8949

TITLE OF INVENTION: METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN COMPUTERS OF VIRTUAL PRIVATE NETWORK

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1400	$300	$0	$1700	02/21/2007

EXAMINER	ART UNIT	CLASS-SUBCLASS
LIM, KRISNA	2153	709-227000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).
☐ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list
(1) the names of up to 3 registered patent attorneys or agents OR, alternatively,
(2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.
1 _____
2 _____
3 _____

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE (B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent) : ☐ Individual ☐ Corporation or other private group entity ☐ Government

4a. The following fee(s) are submitted:
☐ Issue Fee
☐ Publication Fee (No small entity discount permitted)
☐ Advance Order - # of Copies _____

4b. Payment of Fee(s): **(Please first reapply any previously paid issue fee shown above)**
☐ A check is enclosed.
☐ Payment by credit card. Form PTO-2038 is attached.
☐ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment, to Deposit Account Number _____ (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)
☐ a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27.
☐ b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature _____ Date _____

Typed or printed name _____ Registration No. _____

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/702,486	11/07/2003	Victor Larson	000479.00112	8949

22907 7590 11/21/2006

BANNER & WITCOFF
1001 G STREET N W
SUITE 1100
WASHINGTON, DC 20001

EXAMINER	
LIM, KRISNA	
ART UNIT	PAPER NUMBER
2153	

DATE MAILED: 11/21/2006

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment to date is 413 day(s). If the issue fee is paid on the date that is three months after the mailing date of this notice and the patent issues on the Tuesday before the date that is 28 weeks (six and a half months) after the mailing date of this notice, the Patent Term Adjustment will be 413 day(s).

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

PTOL-85 (Rev. 07/06) Approved for use through 04/30/2007.

Notice of Allowability	Application No.	Applicant(s)	
	10/702,486	LARSON ET AL.	
	Examiner	Art Unit	
	Krisna Lim	2153	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *the amendment filed 8/17/06*.

2. ☒ The allowed claim(s) is/are *1-41*.

3. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
   a) ☐ All b) ☐ Some* c) ☐ None of the:
      1. ☐ Certified copies of the priority documents have been received.
      2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
      3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

4. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.

5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
   (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
      1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____.
   (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☒ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
3. ☒ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date 11/09/06
4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. ☐ Notice of Informal Patent Application
6. ☐ Interview Summary (PTO-413), Paper No./Mail Date _____ .
7. ☐ Examiner's Amendment/Comment
8. ☒ Examiner's Statement of Reasons for Allowance
9. ☐ Other _____.

Pursuant to 37 C.F.R 1.109 and M.P.E.P 1302.14, the following is an Examiner's Statement of Reasons for Allowance:

The prior arts of record do not teach a system and a method for accessing a secure computer network address comprising steps of: requesting a secure computer network address from a secure domain name server according to the secure domain name; and using a virtual private network communication link to send an access request message to the secure computer network address.

The examiner considers the applicants' claims 1-41 to be allowable based on the claim interpretation and the aforesaid prior arts of record.

Any comments considered necessary by applicant must be submitted no later than the payment of the Issue Fee and, to avoid processing delays, should preferably **accompany** the Issue Fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Krisna Lim whose telephone number is 571-272-3956 The examiner can normally be reached on Monday to Wednesday and Friday from 7:30 AM to 4:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Glenton Burgess, can be reached on 571-272-3949. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

KI

November 13, 2006

KRISNA LIM
PRIMARY EXAMINER

Date 11/09/06

PTO/SB/08a (08-03)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT
(Not for submission under 37 CFR 1.99)

Application Number	10702486
Filing Date	2003-11-07
First Named Inventor	Victor Larson
Art Unit	2153
Examiner Name	Lim, Krisna
Attorney Docket Number	000479.00112

U.S.PATENTS

Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
VL	1	5870610	A	1999-02-09	Beyda et al.	

If you wish to add additional U.S. Patent citation information please click the Add button.

U.S.PATENT APPLICATION PUBLICATIONS

Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1					

If you wish to add additional U.S. Published Application citation information please click the Add button.

FOREIGN PATENT DOCUMENTS

Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2] i	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]
VL	1	0 838 930	EP	A	1998-04-29	Compaq Computer Corp.		☐
VL	2	0 814 589	EP	A	1997-12-29	AT&T Corp.		☐
VL	3	2 334 181	GB	A	1999-08-11	NEC Technologies; Globalmart Ltd.		☐

EFS Web 2.0

Date 11/09/06

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	10702486
	Filing Date	2003-11-07
	First Named Inventor	Victor Larson
	Art Unit	2153
	Examiner Name	Lim, Krisna
	Attorney Docket Number	000479.00112

KL	4	9827783	WO	A	1998-06-25	Northern Telecom Limited; Antonio, G; Hui, Margare		☐
KL	5	2 317 792	GB	A	1998-04-01	Secure Computing Corporation		☐
	6							☐

If you wish to add additional Foreign Patent Document citation information please click the Add button

NON-PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T5
KL	1	Laurie Wells (LANCASTERBIBELMAIL MSN COM); "Subject: Security Icon" USENET NEWSGROUP, 19 October 1998, XP002200606	☐
KL	2	Davila J et al, "Implementation of Virtual Private Networks at the Transport Layer", Information Security, Second International Workshop, ISW '99. Proceedings (Lecture Springer-Verlag Berlin, Germany, [Online] 1999, pages 85-102, XP002399276, ISBN 3-540-66695-B, retrieved from the Internet: URL: http://www.springerlink.com/content/4uac0tb0heccma89/fulltext.pdf> (Abstract)	☐
	3		☐
	4		☐

If you wish to add additional non-patent literature document citation information please click the Add button

EXAMINER SIGNATURE

Examiner Signature		Date Considered	11/15/06

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

Date 11/09/06

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	10702486
	Filing Date	2003-11-07
	First Named Inventor	Victor Larson
	Art Unit	2153
	Examiner Name	Lim, Krisna
	Attorney Docket Number	000479.00112

[1] See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. [2] Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). [3] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [4] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [5] Applicant is to place a check mark here if English language translation is attached.

Notice of References Cited	Application/Control No. 10/702,486	Applicant(s)/Patent Under Reexamination LARSON ET AL.	
	Examiner Krisna Lim	Art Unit 2153	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-6,119,171	09-2000	Alkhatib, Hasan S.	709/245
*	B	US-6,119,234	09-2000	Aziz et al.	726/11
*	C	US-6,256,671	07-2001	Strentzsch et al.	709/227
	D	US-			
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Issue Classification	Application/Control No. 10/702,486	Applicant(s)/Patent under Reexamination LARSON ET AL.
	Examiner Krisna Lim	Art Unit 2153

ISSUE CLASSIFICATION

ORIGINAL		INTERNATIONAL CLASSIFICATION									
CLASS	SUBCLASS	CLAIMED					NON-CLAIMED				
709	227	G	06	F	15	/173				/	
						/				/	
						/				/	
						/				/	
						/				/	
						/				/	
						/				/	

CROSS REFERENCES				
CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)			
709	228			

(Assistant Examiner) (Date)

(Legal Instruments Examiner) 11-16-06 (Date)

KRISNA LIM
PRIMARY EXAMINER 11/15/06
(Primary Examiner) (Date)

Total Claims Allowed: 41

O.G. Print Claim(s)	O.G. Print Fig.
1	27

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
1	1	31	31		61		91		121		151		181
2	2	32	32		62		92		122		152		182
3	3	33	33		63		93		123		153		183
4	4	34	34		64		94		124		154		184
5	5	35	35		65		95		125		155		185
6	6	36	36		66		96		126		156		186
7	7	37	37		67		97		127		157		187
8	8	38	38		68		98		128		158		188
9	9	39	39		69		99		129		159		189
10	10	40	40		70		100		130		160		190
11	11	41	41		71		101		131		161		191
17	12		42		72		102		132		162		192
18	13		43		73		103		133		163		193
19	14		44		74		104		134		164		194
20	15		45		75		105		135		165		195
21	16		46		76		106		136		166		196
22	17		47		77		107		137		167		197
23	18		48		78		108		138		168		198
24	19		49		79		109		139		169		199
25	20		50		80		110		140		170		200
26	21		51		81		111		141		171		201
27	22		52		82		112		142		172		202
12	23		53		83		113		143		173		203
28	24		54		84		114		144		174		204
13	25		55		85		115		145		175		205
14	26		56		86		116		146		176		206
15	27		57		87		117		147		177		207
16	28		58		88		118		148		178		208
29	29		59		89		119		149		179		209
30	30		60		90		120		150		180		210

Search Notes

Application/Control No.	Applicant(s)/Patent under Reexamination
10/702,486	LARSON ET AL.
Examiner	**Art Unit**
Krisna Lim	2153

SEARCHED			
Class	Subclass	Date	Examiner
709	225-229	11/15/2006	KL

INTERFERENCE SEARCHED			
Class	Subclass	Date	Examiner
709	**227**	11/5/2006	**KL**

SEARCH NOTES (INCLUDING SEARCH STRATEGY)		
	DATE	EXMR
EAST TEXT SEARCH	11/15/2006	KL
NPL (AMC AND IEEE)	11/15/2006	KL
IGoogle	11/15/2006	KL
Inventor Name Search	11/15/2006	KL

Index of Claims

Application/Control No.	Applicant(s)/Patent under Reexamination
10/702,486	LARSON ET AL.
Examiner	**Art Unit**
Krisna Lim	2153

√	Rejected	–	(Through numeral) Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

Claim		Date
Final	Original	11/15/06
1	1	=
2	2	=
3	3	=
4	4	=
5	5	=
6	6	=
7	7	=
8	8	=
9	9	=
10	10	=
11	11	=
17	12	=
18	13	=
19	14	=
20	15	=
21	16	=
22	17	=
23	18	=
24	19	=
25	20	=
26	21	=
27	22	=
12	23	=
28	24	=
13	25	=
14	26	=
15	27	=
16	28	=
29	29	=
30	30	=
31	31	=
32	32	=
33	33	=
34	34	=
35	35	=
36	36	=
37	37	=
38	38	=
39	39	=
40	40	=
41	41	=
	42	
	43	
	44	
	45	
	46	
	47	
	48	
	49	
	50	

Claim		Date
Final	Original	
	51	
	52	
	53	
	54	
	55	
	56	
	57	
	58	
	59	
	60	
	61	
	62	
	63	
	64	
	65	
	66	
	67	
	68	
	69	
	70	
	71	
	72	
	73	
	74	
	75	
	76	
	77	
	78	
	79	
	80	
	81	
	82	
	83	
	84	
	85	
	86	
	87	
	88	
	89	
	90	
	91	
	92	
	93	
	94	
	95	
	96	
	97	
	98	
	99	
	100	

Claim		Date
Final	Original	
	101	
	102	
	103	
	104	
	105	
	106	
	107	
	108	
	109	
	110	
	111	
	112	
	113	
	114	
	115	
	116	
	117	
	118	
	119	
	120	
	121	
	122	
	123	
	124	
	125	
	126	
	127	
	128	
	129	
	130	
	131	
	132	
	133	
	134	
	135	
	136	
	137	
	138	
	139	
	140	
	141	
	142	
	143	
	144	
	145	
	146	
	147	
	148	
	149	
	150	

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

22907 7590 11/21/2006

BANNER & WITCOFF
1001 G STREET N W
SUITE 1100
WASHINGTON, DC 20001

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission
I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/702,486	11/07/2003	Victor Larson	000479.00112	8949

TITLE OF INVENTION: METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN COMPUTERS OF VIRTUAL PRIVATE NETWORK

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1400	$300	$0	$1700	02/21/2007

EXAMINER	ART UNIT	CLASS-SUBCLASS
LIM, KRISNA	2153	709-227000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).
☐ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list
(1) the names of up to 3 registered patent attorneys or agents OR, alternatively,
(2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.
1 Banner & Witcoff, Ltd.
2
3

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE: VirnetX, Inc.
(B) RESIDENCE: (CITY and STATE OR COUNTRY): Scotts Valley, CA

Please check the appropriate assignee category or categories (will not be printed on the patent): ☐ Individual ☒ Corporation or other private group entity ☐ Government

4a. The following fee(s) are submitted:
☒ Issue Fee
☒ Publication Fee (No small entity discount permitted)
☐ Advance Order - # of Copies _____

4b. Payment of Fee(s): **(Please first reapply any previously paid issue fee shown above)**
☐ A check is enclosed.
☐ Payment by credit card. Form PTO-2038 is attached.
☒ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment, to Deposit Account Number 19-0733 (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)
☐ a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27. ☐ b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature _____ Date January 16, 2007

Typed or printed name Ross A. Dannenberg Registration No. 49,024

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Electronic Patent Application Fee Transmittal

Application Number:	10702486
Filing Date:	07-Nov-2003
Title of Invention:	METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN COMPUTERS OF VIRTUAL PRIVATE NETWORK
First Named Inventor/Applicant Name:	Victor Larson
Filer:	Ross Alan Dannenberg/Allison Anderson
Attorney Docket Number:	000479.00112

Filed as Large Entity

Utility Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Utility Appl issue fee	1501	1	1400	1400
Publication fee for republication	1505	1	300	300

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Extension-of-Time:				
Miscellaneous:				
			Total in USD ($)	**1700**

Electronic Acknowledgement Receipt

EFS ID:	1439592
Application Number:	10702486
International Application Number:	
Confirmation Number:	8949
Title of Invention:	METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN COMPUTERS OF VIRTUAL PRIVATE NETWORK
First Named Inventor/Applicant Name:	Victor Larson
Customer Number:	22907
Filer:	Ross Alan Dannenberg/Allison Anderson
Filer Authorized By:	Ross Alan Dannenberg
Attorney Docket Number:	000479.00112
Receipt Date:	16-JAN-2007
Filing Date:	07-NOV-2003
Time Stamp:	09:44:09
Application Type:	Utility

Payment information:

Submitted with Payment	yes
Payment was successfully received in RAM	$1700
RAM confirmation Number	2056
Deposit Account	190733
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows: Charge any Additional Fees required under 37 C.F.R. Section 1.16 and 1.17	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)	Multi Part /.zip	Pages (if appl.)
1	Issue Fee Payment (PTO-85B)	007170-00029IssueFee.pdf	85585	no	1
Warnings:					
Information:					
2	Fee Worksheet (PTO-06)	fee-info.pdf	8335	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	93920		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

PTO-1449 (Modified) U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE INFORMATION DISCLOSURE STATEMENT BY APPLICANT	ATTY. DOCKET NO. 000479.00112	SERIAL NUMBER ~~TBD~~ 10/702,486
	APPLICANT Victor Larson et al.	
	FILING DATE ~~Herewith~~ 11/7/2003	GROUP ART UNIT ~~TBD~~ 2153

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	NAME	CLASS	SUB CLASS	FILING DATE
VL	6,119,171	9/2000	Alkhatib			
VL	5,588,060	12/24/96	Aziz			
VL	5,689,566	11-18-1997 ~~11/18/97~~	Nguyen			
VL	5,842,040	11/24/98	Hughes et al.			
VL	4,933,846	06/12/90	Humphrey et al.			

FOREIGN PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUB CLASS	TRANSLATION YES/NO	
VL	199 24 575	12/2/99	DE				
	0 838 930	4/29/98	EPO				
	2 317 792	4/1/98	GB				
	0 814 589	12/29/97	EPO				
V	WO 98/27783	6/25/98	PCT				

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

VL	Search Report (dated 6/18/02), International Application No. PCT/US01/13260
	Search Report (dated 6/28/02), International Application No. PCT/US01/13261
	Donald E. Eastlake, "Domain Name System Security Extensions", DNS Security Working Group, April 1998, 51 pages
	D. B. Chapman et al., "Building Internet Firewalls", November 1995, pages 278-297 and pages 351-375
	P. Srisuresh et al., "DNS extensions to Network Address Translators", July 1998, 27 pages
	Laurie Wells, "Security Icon", October 19, 1998, 1 page
	W. Stallings, "Cryptography And Network Security", 2nd Edition, Chapter 13, IP Security, June 8, 1998, pages 399-440
	W. Stallings, "New Cryptography and Network Security Book", June 8, 1998, 3 pages
V	FASBENDER, KESDOGAN, and KUBITZ: "Variable and Scalable Security: Protection of Location Information in Mobile IP", IEEE publication, 1996, pages 963-967

EXAMINER KRISNA LIM	DATE CONSIDERED 5/15/06

EXAMINER: Initial citation if reference was considered. Draw line through citation if not in conformance to MPEP 609 and not considered. Include copy of this form with next communication to applicant.

PTO-1449 (Modified) U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE INFORMATION DISCLOSURE STATEMENT BY APPLICANT	ATTY. DOCKET NO. 000479.00112	SERIAL NUMBER TBD
	APPLICANT Victor Larson et al.	
	FILING DATE Herewith	GROUP ART UNIT TBD

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	NAME	CLASS	SUB CLASS	FILING DATE
KL	5,341,426	8/1994	Barney et al.			
	5,787,172	7/1998	Terry Sutton Arnold			
	6,092,200	7/2000	Muniyappa et al.			
	6,158,011	12/2000	Chen et al.			
V	6,178,409 ~~6,168,409~~	1/2001	Weber et al.			

CAS 1/18/07

FOREIGN PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUB CLASS	TRANSLATION YES/NO	
KL	0 858 189	8/12/98	EPO				
	WO 01 50688	7/12/01	PCT				
	WO 98 59470	12/30/98	PCT				
	WO 99 48303	9/23/99	PCT				
V	WO 99 38081	7/29/99	PCT				

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

KL	Search Report (dated 8/20/02), International Application No. PCT/US01/04340
	Search Report (dated 8/23/02), International Application No. PCT/US01/13260
	Shree Murthy et al., "Congestion-Oriented Shortest Multipath Routing", Proceedings of IEEE INFOCOM, 1996, pages 1028-1036
	Jim Jones et al., "Distributed Denial of Service Attacks: Defenses", Global Integrity Corporation, 2000, pages 1-14
	James E. Bellaire, "New Statement of Rules – Naming Internet Domains", Internet Newsgroup, July 30, 1995, 1 page
	D. Clark, "US Calls for Private Domain-Name System", Computer, IEEE Computer Society, August 1, 1998, pages 22-25
	August Bequai, "Balancing Legal Concerns Over Crime and Security in Cyberspace", Computer & Security, Vol. 17, No. 4, 1998, pages 293-298
V	Rich Winkel, "CAQ: Networking With Spooks: The NET & The Control Of Information", Internet Newsgroup, June 21, 1997, 4 pages

EXAMINER KRISNA LIM	DATE CONSIDERED 5/15/06

EXAMINER: Initial citation if reference was considered. Draw line through citation if not in conformance to MPEP 609 and not considered. Include copy of this form with next communication to applicant.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	ISSUE DATE	PATENT NO.	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/702,486	03/06/2007	7188180	007170.00029	8949

22907 7590 02/14/2007

BANNER & WITCOFF
1001 G STREET N W
SUITE 1100
WASHINGTON, DC 20001

ISSUE NOTIFICATION

The projected patent number and issue date are specified above.

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment is 413 day(s). Any patent to issue from the above-identified application will include an indication of the adjustment on the front page.

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at (571)-272-4200.

APPLICANT(s) (Please see PAIR WEB site http://pair.uspto.gov for additional applicants):

Victor Larson, Fairfax, VA;
Robert Dunham Short III, Leesburg, VA;
Edmund Colby Munger, Crownsville, MD;
Michael Williamson, South Riding, VA;

6-27-07 COFC

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Victor Larson, et al.
U.S. Patent No.:	7,188,180
Issue Date:	March 6, 2007
Serial No.	10/702,486
Filing Date:	November 7, 2003
Title:	Method For Establishing Secure Communication Link Between Computers Of Virtual Private Network
Art Unit:	2153
Examiner:	Lim, Krisna
Confirmation No.:	8949
Docket No.	77580-039 (VRNK-1CP2DV)

Commissioner for Patents
Office of Patent Publication
ATTN: Certificate of Correction Branch
P.O. Box 1450
Alexandria, VA 22313-1450

CERTIFICATE OF MAILING BY EXPRESS MAIL (37 CFR § 1.10)

"Express Mail" Mailing Label Number EV 642601585 US

I hereby certify that this correspondence is being deposited with the United States Postal Service as "Express Mail Post Office to Address" under 37 CFR § 1.10 in an envelope addressed to Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date indicated below.

Date: June 26, 2007

Cynthia Joseph

TRANSMITTAL LETTER

Applicants transmit herewith the following documents in the above-identified patent:

1. Request For Certificate Of Correction; and
2. Certificate Of Correction.

The Commissioner is authorized to charge any filing fees to Deposit Account No. 50-1133.

Respectfully submitted,

Toby H. Kusmer, P.C.
Reg. No. 26,418
McDermott Will & Emery LLP
28 State Street
Boston, MA 02109-1775
Telephone: 617.535.4065
Facsimile: 617.535.3800
e-mail: tkusmer@mwe.com

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Victor Larson, et al.
U.S. Patent No.:	7,188,180
Issue Date:	March 6, 2007
Serial No.	10/702,486
Filing Date:	November 7, 2003
Title:	Method For Establishing Secure Communication Link Between Computers Of Virtual Private Network
Art Unit:	2153
Examiner:	Lim, Krisna
Confirmation No.:	8949
Docket No.	77580-039 (VRNK-1CP2DV)

Commissioner for Patents
Office of Patent Publication
ATTN: Certificate of Correction Branch
P.O. Box 1450
Alexandria, VA 22313-1450

CERTIFICATE OF MAILING BY EXPRESS MAIL (37 CFR § 1.10)

"Express Mail" Mailing Label Number EV 642601585 US

I hereby certify that this correspondence is being deposited with the United States Postal Service as "Express Mail Post Office to Address" under 37 CFR § 1.10 in an envelope addressed to Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date indicated below.

Date: June 26, 2007

Cynthia Joseph

TRANSMITTAL LETTER

Applicants transmit herewith the following documents in the above-identified patent:

1. Request For Certificate Of Correction; and
2. Certificate Of Correction.

The Commissioner is authorized to charge any filing fees to Deposit Account No. 50-1133.

Respectfully submitted,

Toby H. Kusmer, P.C.
Reg. No. 26,418
McDermott Will & Emery LLP
28 State Street
Boston, MA 02109-1775
Telephone: 617.535.4065
Facsimile: 617.535.3800
e-mail: tkusmer@mwe.com

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Victor Larson, et al.
U.S. Patent No.:	7,188,180
Issue Date:	March 6, 2007
Serial No.	10/702,486
Filing Date:	November 7, 2003
Title:	Method For Establishing Secure Communication Link Between Computers Of Virtual Private Network
Art Unit:	2153
Examiner:	Lim, Krisna
Confirmation No.:	8949
Docket No.	77580-039 (VRNK-1CP2DV)

Commissioner for Patents
Office of Patent Publication
ATTN: Certificate of Correction Branch
P.O. Box 1450
Alexandria, VA 22313-1450

CERTIFICATE OF MAILING BY EXPRESS MAIL (37 CFR § 1.10)

"Express Mail" Mailing Label Number EV 642601585 US

I hereby certify that this correspondence is being deposited with the United States Postal Service as "Express Mail Post Office to Address" under 37 CFR § 1.10 in an envelope addressed to Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date indicated below.

Date: June 26, 2007

Cynthia Joseph

Sir:

REQUEST FOR CERTIFICATE OF CORRECTION

Pursuant to 35 U.S.C. § 254, and 37 C.F.R. § 1.322, this is a request for a Certificate of Correction in the above-identified patent. The mistake identified below and in the appended Form occurred through the fault of the Patent Office, as clearly disclosed by the records of the application which matured into this patent.

In Patent Title Page, Item (75), Inventors, delete "Durham" and insert therefor -- Dunham --.

JUN 29 2007

Two (2) copies of PTO Form 1050 are appended. The complete Certificate of Correction involves one (1) page. Issuance of the Certificate of Correction containing the correction is earnestly requested.

Please charge any required fees not included herewith to our Deposit Account No. 50-1133.

Respectfully submitted,

June 26, 2007

Toby H. Kusmer, P.C.
Registration No. 26,418
McDermott Will & Emery LLP
28 State Street
Boston, MA 02109-1775
Telephone: (617) 535-4065
Facsimile: (617) 535-3800
E-mail: tkusmer@mwe.com

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Victor Larson, et al.
U.S. Patent No.:	7,188,180
Issue Date:	March 6, 2007
Serial No.	10/702,486
Filing Date:	November 7, 2003
Title:	Method For Establishing Secure Communication Link Between Computers Of Virtual Private Network
Art Unit:	2153
Examiner:	Lim, Krisna
Confirmation No.:	8949
Docket No.	77580-039 (VRNK-1CP2DV)

Commissioner for Patents
Office of Patent Publication
ATTN: Certificate of Correction Branch
P.O. Box 1450
Alexandria, VA 22313-1450

CERTIFICATE OF MAILING BY EXPRESS MAIL (37 CFR § 1.10)

"Express Mail" Mailing Label Number EV 642601585 US

I hereby certify that this correspondence is being deposited with the United States Postal Service as "Express Mail Post Office to Address" under 37 CFR § 1.10 in an envelope addressed to Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date indicated below.

Date: June 26, 2007

Cynthia Joseph

Sir:

REQUEST FOR CERTIFICATE OF CORRECTION

Pursuant to 35 U.S.C. § 254, and 37 C.F.R. § 1.322, this is a request for a Certificate of Correction in the above-identified patent. The mistake identified below and in the appended Form occurred through the fault of the Patent Office, as clearly disclosed by the records of the application which matured into this patent.

In Patent Title Page, Item (75), Inventors, delete "Durham" and insert therefor -- Dunham --.

Two (2) copies of PTO Form 1050 are appended. The complete Certificate of Correction involves one (1) page. Issuance of the Certificate of Correction containing the correction is earnestly requested.

Please charge any required fees not included herewith to our Deposit Account No. 50-1133.

Respectfully submitted,

June 26, 2007

Toby H. Kusmer, P.C.
Registration No. 26,418
McDermott Will & Emery LLP
28 State Street
Boston, MA 02109-1775
Telephone: (617) 535-4065
Facsimile: (617) 535-3800
E-mail: tkusmer@mwe.com

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO.: **7,188,180**

DATED: **March 6, 2007**

INVENTOR(S): **Victor Larson, et al.**

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

IN PATENT TITLE PAGE:

Item (75), Inventors, delete "Durham" and insert therefor -- Dunham --.

MAILING ADDRESS OF SENDER:
Toby H. Kusmer, P.C.
McDermott Will & Emery LLP
28 State Street
Boston, MA 02109-1775

PATENT NO. 7,188,180

JUN 2 9 2007

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO.: **7,188,180**

DATED: **March 6, 2007**

INVENTOR(S): **Victor Larson, et al.**

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

IN PATENT TITLE PAGE:

Item (75), Inventors, delete "Durham" and insert therefor -- Dunham --.

MAILING ADDRESS OF SENDER:
Toby H. Kusmer, P.C.
McDermott Will & Emery LLP
28 State Street
Boston, MA 02109-1775

PATENT NO. 7,188,180

JUN 2 9 200

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 7,188,180 B2
APPLICATION NO. : 10/702486
DATED : March 6, 2007
INVENTOR(S) : Victor Larson et al.

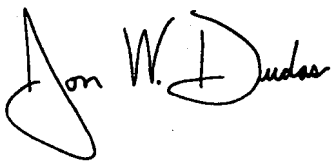
It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

IN PATENT TITLE PAGE:

Item (75), Inventors, delete "Durham" and insert therefor -- Dunham --.

Signed and Sealed this

Seventh Day of August, 2007



JON W. DUDAS
Director of the United States Patent and Trademark Office

PTO/SB/81 (01-09)
Approved for use through 11/30/2011. OMB 0651-0035
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

POWER OF ATTORNEY OR REVOCATION OF POWER OF ATTORNEY WITH A NEW POWER OF ATTORNEY AND CHANGE OF CORRESPONDENCE ADDRESS

Application Number	10/702,486
Filing Date	11/07/2003
First Named Inventor	Larson, Victor
Title	Method for Establishing Secure Communicatio
Art Unit	2153
Examiner Name	Lim, Krisna
Attorney Docket Number	77580-039 (VRNK-1CP2DV)

I hereby revoke all previous powers of attorney given in the above-identified application.

☐ A Power of Attorney is submitted herewith.

OR

☒ I hereby appoint Practitioner(s) associated with the following Customer Number as my/our attorney(s) or agent(s) to prosecute the application identified above, and to transact all business in the United States Patent and Trademark Office connected therewith:

23630

OR

☐ I hereby appoint Practitioner(s) named below as my/our attorney(s) or agent(s) to prosecute the application identified above, and to transact all business in the United States Patent and Trademark Office connected therewith:

Practitioner(s) Name	Registration Number

Please recognize or change the correspondence address for the above-identified application to:

☒ The address associated with the above-mentioned Customer Number.

OR

☐ The address associated with Customer Number:

OR

☐ Firm or Individual Name					
Address					
City		State		Zip	
Country					
Telephone		Email			

I am the:

☐ Applicant/inventor.

OR

☒ Assignee of record of the entire interest. See 37 CFR 3.71.
Statement under 37 CFR 3.73(b) (Form PTO/SB/96) submitted herewith or filed on _____

SIGNATURE of Applicant or Assignee of Record

Signature	Robert Morris	Date	12/15/09
Name	Robert Morris	Telephone	831.426.8200
Title and Company	President VirnetX Inc.		

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below*.

☐ *Total of _____ forms are submitted.

PTO/SB/96 (07-09)
Approved for use through 07/31/2012. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

STATEMENT UNDER 37 CFR 3.73(b)

Applicant/Patent Owner: VirnetX Inc.

Application No./Patent No.: 7,188,180 Filed/Issue Date: 03/06/2007

Titled: METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN TWO COMPUTERS OF VIRTUAL PRIVATE NETWORK

VirnetX Inc., a corporation
(Name of Assignee) (Type of Assignee, e.g., corporation, partnership, university, government agency, etc.

states that it is:

1. [X] the assignee of the entire right, title, and interest in;

2. [ ] an assignee of less than the entire right, title, and interest in
(The extent (by percentage) of its ownership interest is _____ %); or

3. [ ] the assignee of an undivided interest in the entirety of (a complete assignment from one of the joint inventors was made)

the patent application/patent identified above, by virtue of either:

A. [ ] An assignment from the inventor(s) of the patent application/patent identified above. The assignment was recorded in the United States Patent and Trademark Office at Reel _____, Frame _____, or for which a copy therefore is attached.

OR

B. [X] A chain of title from the inventor(s), of the patent application/patent identified above, to the current assignee as follows:

1. From: Larson et al. To Science Applications International Corp.
The document was recorded in the United States Patent and Trademark Office at
Reel 014679, Frame 0947, or for which a copy thereof is attached.

2. From: Science Applications International Corp. To: VirnetX Inc.
The document was recorded in the United States Patent and Trademark Office at
Reel 018757, Frame 0328, or for which a copy thereof is attached.

3. From _____ To: _____
The document was recorded in the United States Patent and Trademark Office at
Reel _____, Frame _____, or for which a copy thereof is attached.

[ ] Additional documents in the chain of title are listed on a supplemental sheet(s).

[X] As required by 37 CFR 3.73(b)(1)(i), the documentary evidence of the chain of title from the original owner to the assignee was, or concurrently is being, submitted for recordation pursuant to 37 CFR 3.11.

[NOTE: A separate copy (*i.e.*, a true copy of the original assignment document(s)) must be submitted to Assignment Division in accordance with 37 CFR Part 3, to record the assignment in the records of the USPTO. See MPEP 302.08]

The undersigned (whose title is supplied below) is authorized to act on behalf of the assignee.

Signature — Date: 11/5/09

Printed or Typed Name: Robert Larson — Title: President

Electronic Acknowledgement Receipt

EFS ID:	6639802
Application Number:	10702486
International Application Number:	
Confirmation Number:	8949
Title of Invention:	METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN COMPUTERS OF VIRTUAL PRIVATE NETWORK
First Named Inventor/Applicant Name:	Victor Larson
Customer Number:	22907
Filer:	Toby H. Kusmer./Kelly Ciarmataro
Filer Authorized By:	Toby H. Kusmer.
Attorney Docket Number:	007170.00029
Receipt Date:	15-DEC-2009
Filing Date:	07-NOV-2003
Time Stamp:	15:23:22
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Power of Attorney	Larson_POA.pdf	755953 c3b35170a2b2492a94aebd0b49a7808e413a8113	no	1

Warnings:	
Information:	

2	Assignee showing of ownership per 37 CFR 3.73(b).	Larson_Statement.pdf	743249 d7d8549a8d0fd27801e95429654a26a3cf6497fd	no	1

Warnings:	
Information:	
Total Files Size (in bytes):	1499202

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
10/702,486	11/07/2003	Victor Larson	77580-039 (VRNK-1CP2DV)

CONFIRMATION NO. 8949

23630
MCDERMOTT WILL & EMERY LLP
28 STATE STREET
BOSTON, MA 02109-1775

POA ACCEPTANCE LETTER

OC000000039341625

Date Mailed: 12/30/2009

NOTICE OF ACCEPTANCE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 12/15/2009.

The Power of Attorney in this application is accepted. Correspondence in this application will be mailed to the above address as provided by 37 CFR 1.33.

/mnguyen/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
10/702,486	11/07/2003	Victor Larson	007170.00029

CONFIRMATION NO. 8949

POWER OF ATTORNEY NOTICE

22907
BANNER & WITCOFF, LTD.
1100 13th STREET, N.W.
SUITE 1200
WASHINGTON, DC 20005-4051

OC000000039341605

Date Mailed: 12/30/2009

NOTICE REGARDING CHANGE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 12/15/2009.

- The Power of Attorney to you in this application has been revoked by the assignee who has intervened as provided by 37 CFR 3.71. Future correspondence will be mailed to the new address of record(37 CFR 1.33).

/mnguyen/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

REEXAMINATION - PATENT OWNER POWER OF ATTORNEY OR REVOCATION OF POWER OF ATTORNEY WITH A NEW POWER OF ATTORNEY AND CHANGE OF CORRESPONDENCE ADDRESS

Control Number(s)	95/001,270
Filing Date(s)	12/08/09
First Named Inventor	Victor Larson
Title	Method for Establishing Secure...
Patent Number	7,188,180
Examiner Name	Lim, Krisna
Attorney Docket No(s).	77580-0090

I hereby revoke all previous patent owner powers of attorney given in the above-identified reexaminaton proceeding control number(s).

☐ A Power of Attorney is submitted herewith.

OR

☒ I hereby appoint Practitioner(s) associated with the following Customer Number as my/our attorney(s) or agent(s) to prosecute the proceeding(s) identified above, and to transact all business in the United States Patent and Trademark Office connected therewith: 23630

OR

☐ I hereby appoint Practitioner(s) named below as my/our attorney(s) or agent(s) to prosecute the proceeding(s) identified above, and to transact all business in the United States Patent and Trademark Office connected therewith:

Practitioner(s) Name	Registration Number

Please recognize or change the correspondence address for the above-identified reexamination proceeding control number(s) (more than one may be changed **only** if they are merged proceedings) to be:

☒ The address associated with the above-mentioned Customer Number.

OR

☐ The address associated with Customer Number:

OR

☐ Firm or Individual Name					
Address					
City		State		Zip	
Country					
Telephone		Email			

I am the:

☐ Inventor, having ownership of the patent being reexamined.

OR

☒ Patent owner.

Statement under 37 CFR 3.73(b) (Form PTO/SB/96) submitted herewith or filed on _____.

SIGNATURE of Inventor or Patent Owner

Signature	Victor J. Larson	Date	1/2/2010
Name	Victor J. Larson	Telephone	703-359-4649
Title and Company	R&D Director VirnetX		

NOTE: Signatures of all the inventors or patent owners of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below*.

☒ *Total of 1 forms are submitted.

Electronic Acknowledgement Receipt

EFS ID:	6813965
Application Number:	10702486
International Application Number:	
Confirmation Number:	8949
Title of Invention:	METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN COMPUTERS OF VIRTUAL PRIVATE NETWORK
First Named Inventor/Applicant Name:	Victor Larson
Customer Number:	23630
Filer:	Toby H. Kusmer./Kelly Ciarmataro
Filer Authorized By:	Toby H. Kusmer.
Attorney Docket Number:	77580-039 (VRNK-1CP2DV)
Receipt Date:	14-JAN-2010
Filing Date:	07-NOV-2003
Time Stamp:	17:33:01
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Power of Attorney	ReexamPOA.pdf	67354 978fe33d73fbad051007559e599897ecffc68870	no	1

Warnings:

The page size in the PDF is too large. The pages should be 8.5 x 11 or A4. If this PDF is submitted, the pages will be resized upon entry into the Image File Wrapper and may affect subsequent processing	
Information:	
Total Files Size (in bytes):	67354

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

AO 120 (Rev. 3/04)

TO:	REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK
Mail Stop 8 Director of the U.S. Patent and Trademark Office P.O. Box 1450 Alexandria, VA 22313-1450	

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court Eastern District of Texas on the following ☑ Patents or ☐ Trademarks:

DOCKET NO.	DATE FILED	U.S. DISTRICT COURT
6:10-cv-417	8/11/2010	Eastern District of Texas

PLAINTIFF	DEFENDANT
VirnetX Inc.,	Aastra USA, Inc., Aastra Technologies Ltd., Apple, Inc., Cisco Systems, Inc., NEC Corporation, and NEC Corporation of America

	PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1	6,502,135	12/31/2002	VirnetX Inc.
2	6,839,759	1/4/2005	VirnetX Inc.
3	7,188,180	3/6/2007	VirnetX Inc.
4	7,418,504	8/26/2008	VirnetX Inc.
5	7,490,151	2/10/2009	VirnetX Inc.

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY ☐ Amendment ☐ Answer ☐ Cross Bill ☐ Other Pleading	
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1--Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2--Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy

#10702486

IN THE UNITED STATES
PATENT AND TRADEMARK OFFICE

In re patent of
VirnetX Inc.
Patent No. 7,188,180
Issued: March 6, 2007
For: Method for establishing secure communication link between computers of virtual private network

Submission of Prior Art Under 37 CFR 1.501

Hon. Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

The undersigned herewith submits in the above-identified patent the following prior art which is pertinent and applicable to the patent and is believed to have a bearing on the patentability of at least claim 1 thereof:

Valencia U.S. 6,308,213 October 23, 2001

The reference discloses a method for creating a secure dial-up session from a remote client to a local network through an internet service provider strikingly similar to the device of VirnetX Inc. It is believed that the reference has a bearing on the patentability of at least claim 1 of the VirnetX Inc. patent.

Insofar as claim 1 is concerned, the reference clearly anticipates the claimed subject matter under 35 U.S.C. 102.

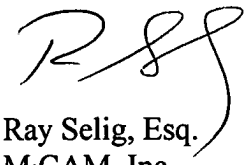
Below is a list of other references which affect one or more of the claims in the patent.

US6874090	US6751738	US6487598	US6339830	US6311218	US6308213	US6240513	US6226748
US6173399	US6072942	US6070243	US5935245	US5918019	US5918018	US5864683	US5850446
US5835726	US5790800	US5790548	US5768271	US5708655	US5559883	US5550984	US5311593
US5276735	US5164988	US5164986	US7861166	US7853723	US7849393	US7844743	US7836481
US7835989	US7831477	US7809847	US7809644	US7788182	US7770196	US7761585	US7752649
US7730299	US7720076	US7716349	US7702540	US7694024	US7664871	US7647243	US7631188
US7627684	US7627001	US7624180	US7620726	US7617527	US7613633	US7593999	US7586939
US7584260	US7583668	US7583665	US7580919	US7546251	US7526644	US7523072	US7515712
US7509270	US7502869	US7496198	US7475156	US7472156	US7462746	US7461160	US7451193
US7443858	US7424737	US7415617	US7401286	US7392395	US7386880	US7380273	US7360244
US7336788	US7299501	US7298851	US7287271	US7281133	US7272625	US7260518	US7251784
US7249378	US7249376	US7224798	US7210035	US7181613	US7165174	US7149208	US7145898
US7143438	US7143290	US7136359	US7133940	US7133846	US7133845	US7124302	US7120802
US7120800	US7117165	US7113508	US7100199	US7095854	US7076652	US7073056	US7072948

US7069451	US7062500	US7058720	US7047415	US7039802	US7039679	US7020700	US7017046
US7010702	US6978017	US6971008	US6963923	US6950436	US6944657	US6937729	US6912222
US6909708	US6901509	US6849045	US6816966	US6798776	US6775692	US6772332	US6766454
US6754212	US6754181	US6748082	US6744892	US6742040	US6741909	US6731625	US6728242
US6721286	US6718319	US6711171	US6701377	US6697836	US6687551	US6681213	US6668278
US6603857	US6598081	US6598075	US6595417	US6590894	US6584480	US6577734	US6571338
US6571296	US6571290	US6560340	US6560236	US6553002	US6546011	US6532540	US6526508
US6516412	US6515968	US6513116	US6510519	US6510154	US6505302	US6505301	US6505241
US6505232	US6501767	US6482156	US6473406	US6470383	US6466987	US6466780	US6463443
US6463057	US6457039	US6456716	US6453345	US6446164	US6445710	US6445703	US6442689
US6438127	US6424717	US6412074	US6408367	US6408341	US6400371	US6396928	US6396831
US6393270	US6378028	US6377691	US6373950	US6363363	US6359882	US6353856	US6349274
US6345303	US6341310	US6339596	US6336141	US6335927	US6332195	US6330608	US6330240
US6326969	US6324525	US6324161	US6317729	US6314520	US6314409	US6314406	US6311207
US6307837	US6304915	US6304908	US6301257	US6301223	US6292568	US6286106	US6285675
US6282172	US6275941	US6272632	US6272110	US6269481	US6266809	US6266704	US6263444
US6263394	US6260145	US6253027	US6252964	US6247132	US6247129	US6246767	US6243812
US6243360	US6237093	US6237006	US6233686	US6230173	US6222842	US6219803	US6219707
US6219421	US6212558	US6206829	US6205483	US6202096	US6202060	US6199115	US6199082
US6192408	US6189032	US6182139	US6182116	US6181711	US6170012	US6163844	US6163843
US6163772	US6161180	US6161145	US6160811	US6157935	US6157647	US6154775	US6151679
US6148400	US6147976	US6145004	US6144962	US6144934	US6141749	US6134658	US6134591
US6119105	US6115780	US6112085	US6111883	US6108786	US6108692	US6104716	US6101531
US6101182	US6098172	US6096094	US6094715	US6091951	US6088796	US6085238	US6078582
US6076113	US6073202	US6073185	US6073176	US6072870	US6072781	US6063129	US6061796
US6061734	US6061721	US6061346	US6058478	US6055562	US6052629	US6049872	US6047338
US6047325	US6046988	US6044402	US6044224	US6041379	US6041358	US6041355	US6041342
US6041041	US6035360	US6035105	US6034680	US6026379	US6023724	US6022315	US6016317
US6014686	US6012066	US6011797	US6011782	US6009467	US6009274	US6009177	US6009173
US6006264	US6005843	US6003137	US6002767	US5999940	US5999629	US5999525	US5996077
US5996076	US5996016	US5991406	US5987611	US5987572	US5987132	US5983350	US5983327
US5983233	US5983208	US5978880	US5978840	US5978594	US5978568	US5978378	US5974453
US5970058	US5968177	US5968176	US5968158	US5968133	US5968116	US5966705	US5966528
US5966509	US5963746	US5963745	US5963556	US5961644	US5960179	US5958052	US5958012
US5958008	US5956403	US5951694	US5950195	US5949975	US5949883	US5946464	US5944783
US5943424	US5941988	US5940591	US5937163	US5937162	US5936940	US5931917	US5926463
US5926458	US5918016	US5917911	US5915087	US5913041	US5913024	US5907704	US5907680
US5903651	US5898780	US5892910	US5892900	US5890005	US5889953	US5889863	US5884246
US5884033	US5878241	US5878212	US5872847	US5870559	US5870550	US5870545	US5867667
US5867665	US5867660	US5867650	US5867494	US5867484	US5864678	US5862339	US5860073
US5856974	US5855020	US5854901	US5852607	US5848258	US5848233	US5845091	US5845070
US5844888	US5842043	US5842031	US5838683	US5835727	US5835725	US5835720	US5835718
US5835714	US5832222	US5832216	US5832092	US5828894	US5828876	US5828833	US5825774
US5822608	US5822531	US5822431	US5815723	US5815665	US5812819	US5812775	US5812771
US5812668	US5812666	US5812552	US5809292	US5805915	US5805820	US5805818	US5805785
US5805595	US5802554	US5802304	US5802291	US5802053	US5799016	US5796951	US5796944
US5796727	US5794059	US5793965	US5793768	US5793763	US5784566	US5781743	US5781534
US5778174	US5777989	US5774689	US5774660	US5771459	US5771353	US5765015	US5765012
US5764935	US5764909	US5764890	US5761523	US5758085	US5758083	US5754871	US5754656
US5752260	US5752067	US5748871	US5748736	US5748633	US5745576	US5745573	US5742762
US5742686	US5742682	US5740402	US5740375	US5740362	US5737525	US5734921	US5734865
US5734853	US5734654	US5732406	US5727147	US5727146	US5724355	US5717944	US5717943
US5717686	US5713037	US5712981	US5710935	US5708836	US5708659	US5706427	US5699532
US5699528	US5699521	US5699513	US5699500	US5687235	US5685004	US5682480	US5680456
US5678045	US5678006	US5673322	US5671279	US5668992	US5668876	US5668857	US5666486
US5664199	US5661803	US5659542	US5657452	US5657390	US5654695	US5651066	US5651002
US5649099	US5644751	US5644720	US5644576	US5638448	US5636371	US5636216	US5634099
US5634074	US5633371	US5632029	US5632011	US5630162	US5625836	US5625626	US5625622
US5623605	US5623601	US5623492	US5621889	US5621727	US5619716	US5619657	US5617577
US5617547	US5617540	US5615340	US5615277	US5614891	US5613204	US5612865	US5611075
US5610981	US5608908	US5606668	US5606493	US5604807	US5602918	US5600644	US5594918
US5592470	US5590299	US5590285	US5590199	US5588152	US5588059	US5587726	US5586263
US5586046	US5577209	US5574475	US5570360	US5566170	US5561669	US5559986	US5557747

US5557742	US5555416	US5553239	US5550816	US5548721	US5548646	US5544340	US5541927
US5537535	US5537099	US5533033	US5533029	US5526489	US5524238	US5515508	US5513346
US5509006	US5506973	US5504921	US5495580	US5495533	US5491808	US5491800	US5491752
US5490212	US5488715	US5485579	US5485510	US5483661	US5483654	US5481613	US5478993
US5476259	US5475687	US5473692	US5473599	US5469554	US5465291	US5463755	US5463752
US5459304	US5457797	US5457786	US5455407	US5453601	US5452420	US5452352	US5450489
US5448045	US5446880	US5446796	US5444491	US5442633	US5442624	US5440723	US5440719
US5440634	US5437024	US5432783	US5430715	US5423020	US5423002	US5421024	US5421019
US5420405	US5418922	US5418854	US5416842	US5414833	US5406628	US5404562	US5404402
US5394408	US5392357	US5392223	US5390336	US5386548	US5381541	US5379289	US5375219
US5375207	US5373559	US5371877	US5371852	US5371794	US5369707	US5369640	US5367643
US5367517	US5365585	US5361256	US5359717	US5349693	US5349686	US5347642	US5347450
US5341477	US5339356	US5337309	US5329521	US5325504	US5323146	US5309562	US5305385
US5301337	US5301247	US5297242	US5291609	US5291442	US5289585	US5285477	US5280480
US5280475	US5276789	US5276678	US5272755	US5268962	US5263165	US5261070	US5261064
US5245606	US5241625	US5241599	US5241594	US5239648	US5237611	US5230020	US5227778
US5226172	US5224163	US5224099	US5222140	US5221838	US5220655	US5220603	US5220516
US5218637	US5216715	US5214767	US5214702	US5214701	US5212806	US5210710	US5208859
US5208858	US5208856	US5204966	US5201000	US5199069	US5197002	US5195089	US5191611
US5185860	US5185796	US5185795	US5181200	US5179704	US5179591	US5177788	US5175416
US5173938	US5166978	US5163154	US5163049	US5159592	US5153919	US5150411	US5150408
US5150401	US5146581	US5146574	US5146498	US5146497	US5144667	US5144664	US5142622
US5142167	US5140634	US5136716	US5136643	US5136642	US5134700	US5115467	US5115466
US5111504	US5109403	US5103476	US5101402	US5093921	US5093860	US5091938	US5091851
US5083265	US5081677	US5077790	US5073852	US5062060	US5060263	US5054067	US5051982
US5048087	US5033084	US5032979	US5030806	US5029207	US5029164	US5025256	US5018137
US5016274	US5014265	US5010549	US5003597	US5003593	US5003591	US5001753	US5001752
US4995082	US4992646	US4988990	US4982430	US4982324	US4968873	US4962531	US4941089
US4935962	US4935870	US4933970	US4933937	US4932056	US4930159	US4926479	US4924500
US4922486	US4919545	US4916737	US4916704	US4906828	US4905176	US4891781	US4888801
US4885777	US4881264	US4877950	US4864615	US4860201	US4859837	US4843026	US4823338
US4817091	US4811393	US4807286	US4803725	US4799156	US4799153	US4792973	US4782326
US4771461	US4769811	US4769810	US4766293	US4748668	US4722502	US4713753	US4712238
US4694491	US4691355	US4689478	US4680753	US4677670	US4672535	US4670857	US4667337
US4658093	US4652993	US4638356	US4634808	US4633434	US4633036	US4630201	US4614861
US4613935	US4613901	US4587627	US4578531	US4577313	US4574350	US4567600	US4533948
US4531929	US4531021	US4529870	US4484025	US4470114	US4454414	US4449181	US4438493
US4424414	US4413315	US4405829	US4403282	US4400770	US4376299	US4371929	US4325116
US4309569	US4303904	US4282572	US4277837	US4227253	US4223380	US4218582	US4200770
US4164787	US4034347	US3956615	US3697768	USRE40187	USRE36751	USH1641	EP1251653
EP0801481	EP0752674	EP0743777	EP0737907	EP0729256	EP0590861	EP0535863	EP0363122
EP0324277	EP0256768	EP0172670	EP0155762	WO9923538	WO9922485	WO9916209	WO9857465
WO9748246	WO9726735	WO9726734	WO9726731	WO9723972	WO9713340	WO9700471	WO9613774
WO9605549	WO9600485	WO9501023	WO9316538	WO9315581	WO9313481	WO9308545	WO9303562
WO9116691	GB2277181	DE4202852	DE19823666	JP09261265	AU0760742	AU0692872	

Respectfully submitted,



Ray Selig, Esq.
M·CAM, Inc.
210 Ridge-McIntire Road, Suite 300
Charlottesville, VA 22903

Certificate of Service

I hereby certify on this 14th day of January 2011, that a true and correct copy of the forgoing "Submission of Prior Art" was mailed by first-class mail, postage paid, to:

VirnetX Inc..
c/o McDermott Will & Emery
600 13th Street, NW
Washington DC 20005-3096

Ray Selig

PATENT
Attorney Docket No. 77580-0039 (VRNK-1CP2DV)
Customer Number 23630

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Larson et al.
Patent No.	7,188,180
Application Serial No.:	10/702,486
Filing Date:	November 7, 2003
Title:	Method For Establishing Secure Communication Link Between Computers Of Virtual Private Network
Examiner:	Lim, Krisna
Art Unit:	2153
Confirmation No.:	8949
Atty. Docket No.:	77580-0039 (VRNK-1CP2DV)

CERTIFICATE OF MAILING OR TRANSMISSION

I hereby certify that this correspondence is being facsimile transmitted (571) 273-8300 to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Mail Stop Certificate of Correction, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 or filed via EFS-Web on the date shown below:

Date: April 3, 2012

Jacqueline Andreu

Jacqueline J. Andreu

Mail Stop Certificate of Correction Branch
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

REQUEST FOR CERTIFICATE OF CORRECTION UNDER 37 CFR 1.322

Sir:

In reviewing the above-identified patent, a printing error was discovered therein requiring correction in order to conform with the Official Record in the application.

The error noted is set forth below and on the attached copy of form PTO/SB/44 in the manner required by the Commissioner's Notice.

IN PATENT TITLE PAGE

Item (73), Assignee:, delete the assignee name listed, and insert the following name:

--VIRNETX, INC., Zephyr Cove, NV (US)--

The error identified in the appended PTO/SB/44 form occurred through the fault of the Patent Office, as disclosed by the records of the application which matured into this patent. See, for example, the attached copy of the Issue Fee Transmittal filed in this case on Jan. 16, 2007, as appears on the USPTO's Public PAIR system, and the attached copy of the Assignment Records as filed in this case and as appear on the USPTO's Assignment website. Since the error in the spelling of the assignee name occurred through the fault of the Patent Office, it is respectfully requested that a Certificate of Correction be issued without expense to the Applicant under Rule 37 CFR § 1.322.

The change requested herein corrects the typographical error in the spelling of the assignee name. Also, since the assignee's address has recently changed, which has been recorded with the Assignment Division, the appended PTO/SB/44 form reflects the assignee's new address. The Commissioner is authorized to charge any fees that may be required with respect to the Certificate of Correction reflecting the Assignee's new address to Deposit Account 50-1133.

Two (2) copies of PTO Form PTO/SB/44 are appended. The complete Certificate of Correction involves one (1) page. Issuance of the Certificate of Correction containing the correction is earnestly requested. No fee is believed to be due for the filing of this request; however, the Commissioner is authorized to charge any shortage in fees, or credit any overpayment, in connection with this filing to Deposit Account 50-1133.

Respectfully submitted,

Dated: April 3, 2012

By: _____

Atabak R. Royaee, Ph.D., Reg. No. 59,037
McDERMOTT WILL & EMERY LLP
28 State Street
Boston, MA 02109-1775
Tel. (617) 535-4108
Fax: (617) 535-3800

DM_US 32659767-1.077580 0039

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail Mail Stop ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

22907 7590 11/21/2006

BANNER & WITCOFF
1001 G STREET N W
SUITE 1100
WASHINGTON, DC 20001

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission
I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/702,486	11/07/2003	Victor Larson	000479.00112	8949

TITLE OF INVENTION: METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN COMPUTERS OF VIRTUAL PRIVATE NETWORK

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1400	$300	$0	$1700	02/21/2007

EXAMINER	ART UNIT	CLASS-SUBCLASS
LIM, KRISNA	2153	709-227000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).
☐ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list
(1) the names of up to 3 registered patent attorneys or agents OR, alternatively,
(2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.
1 Banner & Witcoff, Ltd.
2
3

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE: VirnetX, Inc.
(B) RESIDENCE: (CITY and STATE OR COUNTRY): Scotts Valley, CA

Please check the appropriate assignee category or categories (will not be printed on the patent): ☐ Individual ☒ Corporation or other private group entity ☐ Government

4a. The following fee(s) are submitted:
☒ Issue Fee
☒ Publication Fee (No small entity discount permitted)
☐ Advance Order - # of Copies _____

4b. Payment of Fee(s): **(Please first reapply any previously paid issue fee shown above)**
☐ A check is enclosed.
☐ Payment by credit card. Form PTO-2038 is attached.
☒ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment, to Deposit Account Number 19-0733 (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)
☐ a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27.
☐ b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature _____ Date January 16, 2007

Typed or printed name Ross A. Dannenberg Registration No. 49,024

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Patent Assignment Abstract of Title

NOTE:Results display only for issued patents and published applications. For pending or abandoned applications please consult USPTO staff.

Total Assignments: 3

Patent #: 7188180 **Issue Dt:** 03/06/2007 **Application #:** 10702486 **Filing Dt:** 11/07/2003

Publication #: 20040107285 **Pub Dt:** 06/03/2004

Inventors: Victor Larson, Robert Dunham Short III, Edmund Colby Munger, Michael Williamson

Title: METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN COMPUTERS OF VIRTUAL PRIVATE NETWORK

Assignment: 1

Reel/Frame: 014679/0947 **Recorded:** 11/07/2003 **Pages:** 3

Conveyance: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS).

Assignors: LARSON, VICTOR **Exec Dt:** 11/06/2003

SHORT, ROBERT DUNHAM III **Exec Dt:** 10/27/2003

MUNGER, EDMUND COLBY **Exec Dt:** 11/05/2003

WILLIAMSON, MICHAEL **Exec Dt:** 11/05/2003

Assignee: SCIENCE APPLICATIONS INTERNATIONAL CORPORATION
10260 CAMPUS POINT DRIVE
SAN DIEGO, CALIFORNIA 92121

Correspondent: BANNER & WITCOFF, LTD.
ROSS A. DANNENBERG
1001 G STREET, N.W., 11TH FLOOR
WASHINGTON, DC 20001

Assignment: 2

Reel/Frame: 018757/0326 **Recorded:** 01/10/2007 **Pages:** 5

Conveyance: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS).

Assignor: SCIENCE APPLICATIONS INTERNATIONAL CORPORATION **Exec Dt:** 12/21/2006

Assignee: VIRNETX INC.
5615 SCOTTS VALLEY DRIVE, SUITE 110
SCOTTS VALLEY DRIVE, CALIFORNIA 95066

Correspondent: BANNER & WITCOFF, LTD.
1001 G STREET, N.W. - 11TH FLOOR
WASHINGTON, D.C. 20001-4597

Assignment: 3

Reel/Frame: 027558/0281 **Recorded:** 01/19/2012 **Pages:** 3

Conveyance: CHANGE OF ADDRESS OF ASSIGNEE

Assignor: VIRNETX INC. **Exec Dt:** 01/19/2012

Assignee: VIRNETX INC.
P.O. BOX 439
ZEPHYR COVE, NEVADA 89448

Correspondent: MCDERMOTT WILL & EMERY LLP
600 13TH STREET NW
WASHINGTON, DC 20005

UNITED STATES PATENT AND TRADEMARK OFFICE
UNDER SECRETARY OF COMMERCE FOR INTELLECTUAL PROPERTY AND
DIRECTOR OF THE UNITED STATES PATENT AND TRADEMARK OFFICE

JANUARY 20, 2012

PTAS

MCDERMOTT WILL & EMERY LLP
600 13TH STREET NW
WASHINGTON, DC 20005

501790945

UNITED STATES PATENT AND TRADEMARK OFFICE
NOTICE OF RECORDATION OF ASSIGNMENT DOCUMENT

THE ENCLOSED DOCUMENT HAS BEEN RECORDED BY THE ASSIGNMENT RECORDATION BRANCH OF THE U.S. PATENT AND TRADEMARK OFFICE. A COMPLETE COPY IS AVAILABLE AT THE ASSIGNMENT SEARCH ROOM ON THE REEL AND FRAME NUMBER REFERENCED BELOW.

PLEASE REVIEW ALL INFORMATION CONTAINED ON THIS NOTICE. THE INFORMATION CONTAINED ON THIS RECORDATION NOTICE REFLECTS THE DATA PRESENT IN THE PATENT AND TRADEMARK ASSIGNMENT SYSTEM. IF YOU SHOULD FIND ANY ERRORS OR HAVE QUESTIONS CONCERNING THIS NOTICE, YOU MAY CONTACT THE ASSIGNMENT RECORDATION BRANCH AT 571-272-3350. PLEASE SEND REQUEST FOR CORRECTION TO: U.S. PATENT AND TRADEMARK OFFICE, MAIL STOP: ASSIGNMENT RECORDATION BRANCH, P.O. BOX 1450, ALEXANDRIA, VA 22313.

RECORDATION DATE: 01/19/2012

REEL/FRAME: 027558/0281
NUMBER OF PAGES: 3

BRIEF: CHANGE OF ADDRESS OF ASSIGNEE

DOCKET NUMBER: 077580-0010

ASSIGNOR:
VIRNETX INC.

DOC DATE: 01/19/2012

ASSIGNEE:
VIRNETX INC.
P.O. BOX 439
ZEPHYR COVE, NEVADA 89448

APPLICATION NUMBER: 09429643
PATENT NUMBER: 7010604
TITLE: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

FILING DATE: 10/29/1999
ISSUE DATE: 03/07/2006

APPLICATION NUMBER: 09504783
PATENT NUMBER: 6502135
TITLE: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILTY

FILING DATE: 02/15/2000
ISSUE DATE: 12/31/2002

APPLICATION NUMBER: 09874258
PATENT NUMBER: 7209479
TITLE: THIRD PARTY VPN CERTIFICATION

FILING DATE: 06/06/2001
ISSUE DATE: 04/24/2007

APPLICATION NUMBER: 10082164 FILING DATE: 02/26/2002
PATENT NUMBER: 6618761 ISSUE DATE: 09/09/2003
TITLE: AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY USING WEIGHTED TRANSMISSION PATHS

APPLICATION NUMBER: 10082285 FILING DATE: 02/26/2002
PATENT NUMBER: 6834310 ISSUE DATE: 12/21/2004
TITLE: PREVENTING PACKET FLOODING OF A COMPUTER ON A COMPUTER NETWORK

APPLICATION NUMBER: 10259494 FILING DATE: 09/30/2002
PATENT NUMBER: 7490151 ISSUE DATE: 02/10/2009
TITLE: ESTABLISHMENT OF A SECURE COMMUNICATION LINK BASED ON A DOMAIN NAME SERVICE (DNS) REQUEST

APPLICATION NUMBER: 10401551 FILING DATE: 03/31/2003
PATENT NUMBER: 7133930 ISSUE DATE: 11/07/2006
TITLE: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

APPLICATION NUMBER: 10401888 FILING DATE: 03/31/2003
PATENT NUMBER: 6907473 ISSUE DATE: 06/14/2005
TITLE: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

APPLICATION NUMBER: 10702486 FILING DATE: 11/07/2003
PATENT NUMBER: 7188180 ISSUE DATE: 03/06/2007
TITLE: METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN COMPUTERS OF VIRTUAL PRIVATE NETWORK

APPLICATION NUMBER: 10702522 FILING DATE: 11/07/2003
PATENT NUMBER: 6839759 ISSUE DATE: 01/04/2005
TITLE: METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN COMPUTERS OF VIRTUAL PRIVATE NETWORK WITHOUT USER ENTERING ANY CRYPTOGRAPHIC INFORMATION

APPLICATION NUMBER: 10702580 FILING DATE: 11/07/2003
PATENT NUMBER: 6826616 ISSUE DATE: 11/30/2004
TITLE: METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN COMPUTERS OF VIRTUAL PRIVATE NETWORK

APPLICATION NUMBER: 10714849 FILING DATE: 11/18/2003
PATENT NUMBER: 7418504 ISSUE DATE: 08/26/2008
TITLE: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS USING SECURE DOMAIN NAMES

APPLICATION NUMBER: 11301022 FILING DATE: 12/13/2005
PATENT NUMBER: 7996539 ISSUE DATE: 08/09/2011
TITLE: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

APPLICATION NUMBER: 11679416 FILING DATE: 02/27/2007
PATENT NUMBER: 8051181 ISSUE DATE: 11/01/2011
TITLE: METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN COMPUTERS OF VIRTUAL PRIVATE NETWORK

APPLICATION NUMBER: 11839969 FILING DATE: 08/16/2007
PATENT NUMBER: 7933990 ISSUE DATE: 04/26/2011
TITLE: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

APPLICATION NUMBER: 11839987 FILING DATE: 08/16/2007
PATENT NUMBER: 7987274 ISSUE DATE: 07/26/2011
TITLE: METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN COMPUTERS OF VIRTUAL PRIVATE NETWORK

APPLICATION NUMBER: 11840508 FILING DATE: 08/17/2007
PATENT NUMBER: 7945654 ISSUE DATE: 05/17/2011
TITLE: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS USING SECURE DOMAIN NAMES

APPLICATION NUMBER: 11840560 FILING DATE: 08/17/2007
PATENT NUMBER: 7921211 ISSUE DATE: 04/05/2011
TITLE: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS USING SECURE DOMAIN NAMES

APPLICATION NUMBER: 11840579 FILING DATE: 08/17/2007
PATENT NUMBER: 7944915 ISSUE DATE: 05/17/2011
TITLE: THIRD PARTY VPN CERTIFICATION

APPLICATION NUMBER: 11924460 FILING DATE: 10/25/2007
PATENT NUMBER: ISSUE DATE:
TITLE: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

APPLICATION NUMBER: 13080684 FILING DATE: 04/06/2011
PATENT NUMBER: ISSUE DATE:
TITLE: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

ASSIGNMENT RECORDATION BRANCH
PUBLIC RECORDS DIVISION

UNITED STATES PATENT AND TRADEMARK OFFICE

UNDER SECRETARY OF COMMERCE FOR INTELLECTUAL PROPERTY AND
DIRECTOR OF THE UNITED STATES PATENT AND TRADEMARK OFFICE

JANUARY 16, 2007

PTAS

700305353A

700305353A

BANNER & WITCOFF, LTD.
1001 G STREET, N.W. - 11TH FLOOR
WASHINGTON, D.C. 20001-4597

UNITED STATES PATENT AND TRADEMARK OFFICE
NOTICE OF RECORDATION OF ASSIGNMENT DOCUMENT

THE ENCLOSED DOCUMENT HAS BEEN RECORDED BY THE ASSIGNMENT DIVISION OF THE U.S. PATENT AND TRADEMARK OFFICE. A COMPLETE MICROFILM COPY IS AVAILABLE AT THE ASSIGNMENT SEARCH ROOM ON THE REEL AND FRAME NUMBER REFERENCED BELOW.

PLEASE REVIEW ALL INFORMATION CONTAINED ON THIS NOTICE. THE INFORMATION CONTAINED ON THIS RECORDATION NOTICE REFLECTS THE DATA PRESENT IN THE PATENT AND TRADEMARK ASSIGNMENT SYSTEM. IF YOU SHOULD FIND ANY ERRORS OR HAVE QUESTIONS CONCERNING THIS NOTICE, YOU MAY CONTACT THE EMPLOYEE WHOSE NAME APPEARS ON THIS NOTICE AT 571-272-3350. PLEASE SEND REQUEST FOR CORRECTION TO: U.S. PATENT AND TRADEMARK OFFICE, MAIL STOP: ASSIGNMENT SERVICES BRANCH, P.O. BOX 1450, ALEXANDRIA, VA 22313.

RECORDATION DATE: 01/10/2007

REEL/FRAME: 018757/0326
NUMBER OF PAGES: 5

BRIEF: ASSIGNMENT OF ASSIGNOR'S INTEREST (SEE DOCUMENT FOR DETAILS).
DOCKET NUMBER: 007170.00019

ASSIGNOR:
SCIENCE APPLICATIONS INTERNATIONAL CORPORATION

DOC DATE: 12/21/2006

ASSIGNEE:
VIRNETX INC.
5615 SCOTTS VALLEY DRIVE, SUITE 110
SCOTTS VALLEY DRIVE, CALIFORNIA

95066

SERIAL NUMBER: 10259494 FILING DATE: 09/30/2002
PATENT NUMBER: ISSUE DATE:
TITLE: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

SERIAL NUMBER: 10714849 FILING DATE: 11/18/2003
PATENT NUMBER: ISSUE DATE:
TITLE: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS USING SECURE DOMAIN NAMES

SERIAL NUMBER: 09874258 FILING DATE: 06/06/2001
PATENT NUMBER: ISSUE DATE:
TITLE: THIRD PARTY VPN CERTIFICATION

SERIAL NUMBER: 10702486 FILING DATE: 11/07/2003
PATENT NUMBER: ISSUE DATE:
TITLE: METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN COMPUTERS OF VIRTUAL PRIVATE NETWORK

SERIAL NUMBER: 10401551 FILING DATE: 03/31/2003
PATENT NUMBER: 7133930 ISSUE DATE: 11/07/2006
TITLE: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

SERIAL NUMBER: 09429643 FILING DATE: 10/29/1999
PATENT NUMBER: 7010604 ISSUE DATE: 03/07/2006
TITLE: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

SERIAL NUMBER: 09504783 FILING DATE: 02/15/2000
PATENT NUMBER: 6502135 ISSUE DATE: 12/31/2002
TITLE: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILTY

SERIAL NUMBER: 10082164 FILING DATE: 02/26/2002
PATENT NUMBER: 6618761 ISSUE DATE: 09/09/2003
TITLE: AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY USING WEIGHTED TRANSMISSION PATHS

SERIAL NUMBER: 10401888 FILING DATE: 03/31/2003
PATENT NUMBER: 6907473 ISSUE DATE: 06/14/2005
TITLE: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

SERIAL NUMBER: 10082285 FILING DATE: 02/26/2002
PATENT NUMBER: 6834310 ISSUE DATE: 12/21/2004
TITLE: PREVENTING PACKET FLOODING OF A COMPUTER ON A COMPUTER NETWORK

SERIAL NUMBER: 10702522 FILING DATE: 11/07/2003
PATENT NUMBER: 6839759 ISSUE DATE: 01/04/2005
TITLE: METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN COMPUTERS

INFORMATION

018757/0326 PAGE 3

SERIAL NUMBER: 10702580
PATENT NUMBER: 6826616
TITLE: METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN COMPUTERS OF VIRTUAL PRIVATE NETWORK

FILING DATE: 11/07/2003
ISSUE DATE: 11/30/2004

ANTIONE ROYALL, EXAMINER
ASSIGNMENT SERVICES BRANCH
PUBLIC RECORDS DIVISION

PTO/SB/44 (09-07)
Approved for use through 08/31/2013. OMB 0651-0033
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

(Also Form PTO-1050)

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 7,188,180

APPLICATION NO.: 10/702,486

ISSUE DATE : March 6, 2007

INVENTOR(S) : Victor Larson et al.

It is certified that an error appears or errors appear in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

IN PATENT TITLE PAGE

Item (73), Assignee:, delete the assignee name listed, and insert the following name:

--VIRNETX, INC., Zephyr Cove, NV (US)--

MAILING ADDRESS OF SENDER (Please do not use customer number below):

Atabak R. Royaee, Reg. No. 59,037
McDERMOTT WILL & EMERY LLP
28 State Street, Boston, MA, 02109

PTO/SB/44 (09-07)
Approved for use through 08/31/2013. OMB 0651-0033
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

(Also Form PTO-1050)

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 7,188,180

APPLICATION NO.: 10/702,486

ISSUE DATE : March 6, 2007

INVENTOR(S) : Victor Larson et al.

It is certified that an error appears or errors appear in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

IN PATENT TITLE PAGE

Item (73), Assignee:, delete the assignee name listed, and insert the following name:

--VIRNETX, INC., Zephyr Cove, NV (US)--

MAILING ADDRESS OF SENDER (Please do not use customer number below):

Atabak R. Royaee, Reg. No. 59,037
McDERMOTT WILL & EMERY LLP
28 State Street, Boston, MA, 02109

Electronic Acknowledgement Receipt

EFS ID:	12458493
Application Number:	10702486
International Application Number:	
Confirmation Number:	8949
Title of Invention:	METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN COMPUTERS OF VIRTUAL PRIVATE NETWORK
First Named Inventor/Applicant Name:	Victor Larson
Customer Number:	23630
Filer:	Atabak R Royaee/Jacqueline Andreu
Filer Authorized By:	Atabak R Royaee
Attorney Docket Number:	77580-039 (VRNK-1CP2DV)
Receipt Date:	03-APR-2012
Filing Date:	07-NOV-2003
Time Stamp:	15:34:52
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Request for Certificate of Correction	Request_For_Certificate_of_Correction.pdf	304690 1c8cd7ae108b39efa71b847454c506406e22f8cb	no	10

Warnings:

The page size in the PDF is too large. The pages should be 8.5 x 11 or A4. If this PDF is submitted, the pages will be resized upon entry into the Image File Wrapper and may affect subsequent processing

Information:

2	Miscellaneous Incoming Letter	Certificate_of_Correction.pdf	54456 a2a6727acca33134bea5a34f7c60267977b53283	no	2

Warnings:

The page size in the PDF is too large. The pages should be 8.5 x 11 or A4. If this PDF is submitted, the pages will be resized upon entry into the Image File Wrapper and may affect subsequent processing

Information:

Total Files Size (in bytes):	359146

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

UNITED STATES PATENT AND TRADEMARK OFFICE

4-10-2012

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

Patent No : 7,188,180 B2
Ser. No. : 10702,486
Inventor(s) : Victor Larson, et. al.
Issued : March 6, 2007

Re: Request for Certificate of Correction

Consideration has been given your request for the issuance of a certificate of correction for the above-identified patent under the provisions of Rule(s) 1.322.

Assignees' names and addresses (assignment data) printed in a patent, are based *solely* on information supplied in the appropriate space for identifying the assignment data, i.e., item 3 of the Issue Fee Transmittal Form PTOL-85B. Granting of a request under 37 CFR 3.81(b) is required to correct applicant's error providing ***incorrect or erroneous*** assignment data, *before* issuance of a Certificate of Correction, under 37 CFR 1.323 *(see Manual of Patent Examining Procedures (M.P.E.P) Chp.1400, sect. 1481).* This procedure is required *at any time after the issue fee is paid,* including after issuance of the patent.

In view of the foregoing, your request is hereby denied.

A request to correct the Assignee under 37 CFR 3.81(b) should include:

A. **the processing fee set forth in 37 CFR 1.17(i) (currently $130);**
B. a statement that the failure to include the correct assignee name on the PTOL-85B was inadvertent; and
C. a copy of the Notice of Recordation of Assignment Document, reflecting the reel and frame number where the assignment(s) is recorded and/or reflecting proof of *the date* the assignment was submitted for recordation.

In the Request, Applicant(s) may request that the file be forwarded to Certificates of Correction Branch, for issuance of a Certificate of Correction, if the Request is granted.

Any request under 37 CFR 3.81(b) should be directed to the following address or facsimile number:

By mail: Mail Stop PETITIONS
Commissioner for Patents
Post Office Box 1450
Alexandria, VA 22313-1450

By hand: Customer Service Window
Mail Stop Petitions
Randolph Building
401 Dulany Street
Alexandria, VA 22314

By fax: (703) 872-9306
ATTN: Office of Petitions

If a fee (currently $100) was previously submitted for consideration of a Request for Certificate of Correction, under CFR 1.323, to correct assignment data, , no additional fee is required.

Eva James
For Mary Diggs
Decisions & Certificates
of Correction Branch
(571) 272-3422 or 703- 756 -1580

Atabak R. Royaee
Mcdermott Will & Emery LLP
28 State Street
Boston, MA 02109

ej

PATENT
Attorney Docket No. 77580-0039 (VRNK-1CP2DV)
Customer Number 23630

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Larson et al.
Patent No.	7,188,180
Application Serial No.:	10/702,486
Filing Date:	November 7, 2003
Title:	Method For Establishing Secure Communication Link Between Computers Of Virtual Private Network
Examiner:	Lim, Krisna
Art Unit:	2153
Confirmation No.:	8949
Atty. Docket No.:	77580-0039 (VRNK-1CP2DV)

CERTIFICATE OF MAILING OR TRANSMISSION

I hereby certify that this correspondence is being facsimile transmitted (571) 273-8300 to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Mail Stop Certificate of Correction, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 or filed via EFS-Web on the date shown below:

Date: July 12, 2012

Jacqueline J. Andreu

Mail Stop PETITIONS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

REQUEST FOR CERTIFICATE OF CORRECTION

UNDER 37 CFR §§1.322 & 1.323

Sir:

In reviewing the above-identified patent, a printing error was discovered therein requiring correction in order to conform with the Official Record in the application.

The error noted is set forth below and on the attached copy of form PTO/SB/44 in the manner required by the Commissioner's Notice.

IN PATENT TITLE PAGE

Item (73), Assignee:, delete the assignee name and address listed, and insert the following:

--VIRNETX, INC., Zephyr Cove, NV (US)--

The error in the Assignee Name identified in the appended PTO/SB/44 form occurred through the fault of the Patent Office, as disclosed by the records of the application which matured into this patent. See, for example, the attached copy of the Issue Fee Transmittal Form PTOL-85B (filed in this case on Jan. 16, 2007), which shows the Assignee Name as follows:

VirnetX, Inc.

The Office is requested to open the electronic copy of the Issue Fee Transmittal Form PTOL-85B, and zoom-in on the Assignee Name typed in that form to see that it has been correctly spelled as VIRNETX, INC. Please note that the letters "r" and "n" must have incorrectly been read as an "m" by the Office, which is incorrect.

A copy of the Notice of Recordation of Assignment and a copy of the Assignment records as appears on the USPTO's Assignment website are attached which reflect (1) VIRNETX, INC., to be the assignee of the above-referenced patent, (2) the reel and frame numbers where the assignments are recorded, and (3) the proof of the date of the recordation of the assignments.

Since the error in the spelling of the Assignee Name occurred through the fault of the Patent Office, it is respectfully requested that a Certificate of Correction be issued without expense to the Applicant under Rule 37 CFR § 1.322.

As provided above, the error in the Assignee Names was as the result of the Office's mistake. The Assignee Address was correct.

However, since the issuance of the patent, the Assignee has moved to a new address. As indicated by the attached Notice of Recordation of Assignment, this change of address has been recorded with the Patent Office.

Accordingly, and pursuant to 37 CFR § 1.323, the Applicant respectfully request the Office to issue a Certification of Correction that reflects the Assignee's current address.

The Applicant submits that the failure to include the "**Zephyr Cove, NV (US)**" address with the Assignee Name on the PTOL-85B form was inadvertent.

As indicated above, a copy of the Notice of Recordation of Assignment and the Assignment records as appears on the USPTO's Assignment website are attached for the Office's review.

With respect to correcting the Assignee Address, the Office is authorized to charge the fee set forth in **37 CFR § 1.20(a)** and the processing fee set forth in **37 CFR § 1.17(i)** to Deposit Account 50-1133.

Two (2) copies of PTO Form PTO/SB/44 are appended. The complete Certificate of Correction involves one (1) page. Issuance of the Certificate of Correction containing the corrections is earnestly requested.

Kindly forward the file to the Certificate of Correction branch, for issuance of a Certificate of Correction, if this Request is granted. The Office is also authorized to charge any shortage in fees pursuant to any applicable rule, or credit any overpayment, in connection with this filing to Deposit Account 50-1133.

Respectfully submitted,

McDERMOTT WILL & EMERY LLP

Dated: July 12, 2012

By: _____

Atabak R. Royaee, Ph.D., Reg. No. 59,037
McDERMOTT WILL & EMERY LLP
28 State Street
Boston, MA 02109-1775
Tel. (617) 535-4108
Fax: (617) 535-3800

DM_US 32659767-1.077580.0039

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail Mail Stop ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

22907 7590 11/21/2006

BANNER & WITCOFF
1001 G STREET N W
SUITE 1100
WASHINGTON, DC 20001

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission
I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/702,486	11/07/2003	Victor Larson	000479.00112	8949

TITLE OF INVENTION: METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN COMPUTERS OF VIRTUAL PRIVATE NETWORK

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1400	$300	$0	$1700	02/21/2007

EXAMINER	ART UNIT	CLASS-SUBCLASS
LIM, KRISNA	2153	709-227000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).
☐ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. Use of a Customer Number is required.

2. For printing on the patent front page, list
(1) the names of up to 3 registered patent attorneys or agents OR, alternatively,
(2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.
1 Banner & Witcoff, Ltd.
2
3

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE: VirnetX, Inc.

(B) RESIDENCE: (CITY and STATE OR COUNTRY): Scotts Valley, CA

Please check the appropriate assignee category or categories (will not be printed on the patent): ☐ Individual ☒ Corporation or other private group entity ☐ Government

4a. The following fee(s) are submitted:
☒ Issue Fee
☒ Publication Fee (No small entity discount permitted)
☐ Advance Order - # of Copies _____

4b. Payment of Fee(s): (Please first reapply any previously paid issue fee shown above)
☐ A check is enclosed.
☐ Payment by credit card. Form PTO-2038 is attached.
☒ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment, to Deposit Account Number 19-0733 (enclose an extra copy of this form).

5. Change in Entity Status (from status indicated above)
☐ a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27. ☐ b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature _____ Date January 16, 2007

Typed or printed name Ross A. Dannenberg Registration No. 49,024

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Patent Assignment Abstract of Title

NOTE:Results display only for issued patents and published applications. For pending or abandoned applications please consult USPTO staff.

Total Assignments: 3

Patent #: 7188180 **Issue Dt:** 03/06/2007 **Application #:** 10702486 **Filing Dt:** 11/07/2003

Publication #: 20040107285 **Pub Dt:** 06/03/2004

Inventors: Victor Larson, Robert Dunham Short III, Edmund Colby Munger, Michael Williamson

Title: METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN COMPUTERS OF VIRTUAL PRIVATE NETWORK

Assignment: 1

Reel/Frame: 014679/0947 **Recorded:** 11/07/2003 **Pages:** 3

Conveyance: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS).

Assignors: LARSON, VICTOR — **Exec Dt:** 11/06/2003

SHORT, ROBERT DUNHAM III — **Exec Dt:** 10/27/2003

MUNGER, EDMUND COLBY — **Exec Dt:** 11/05/2003

WILLIAMSON, MICHAEL — **Exec Dt:** 11/05/2003

Assignee: SCIENCE APPLICATIONS INTERNATIONAL CORPORATION
10260 CAMPUS POINT DRIVE
SAN DIEGO, CALIFORNIA 92121

Correspondent: BANNER & WITCOFF, LTD.
ROSS A. DANNENBERG
1001 G STREET, N.W., 11TH FLOOR
WASHINGTON, DC 20001

Assignment: 2

Reel/Frame: 018757/0326 **Recorded:** 01/10/2007 **Pages:** 5

Conveyance: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS).

Assignor: SCIENCE APPLICATIONS INTERNATIONAL CORPORATION **Exec Dt:** 12/21/2006

Assignee: VIRNETX INC.
5615 SCOTTS VALLEY DRIVE, SUITE 110
SCOTTS VALLEY DRIVE, CALIFORNIA 95066

Correspondent: BANNER & WITCOFF, LTD.
1001 G STREET, N.W. - 11TH FLOOR
WASHINGTON, D.C. 20001-4597

Assignment: 3

Reel/Frame: 027558/0281 **Recorded:** 01/19/2012 **Pages:** 3

Conveyance: CHANGE OF ADDRESS OF ASSIGNEE

Assignor: VIRNETX INC. **Exec Dt:** 01/19/2012

Assignee: VIRNETX INC.
P.O. BOX 439
ZEPHYR COVE, NEVADA 89448

Correspondent: MCDERMOTT WILL & EMERY LLP
600 13TH STREET NW
WASHINGTON, DC 20005

UNITED STATES PATENT AND TRADEMARK OFFICE
UNDER SECRETARY OF COMMERCE FOR INTELLECTUAL PROPERTY AND
DIRECTOR OF THE UNITED STATES PATENT AND TRADEMARK OFFICE

JANUARY 20, 2012

PTAS

MCDERMOTT WILL & EMERY LLP
600 13TH STREET NW
WASHINGTON, DC 20005

501790945

UNITED STATES PATENT AND TRADEMARK OFFICE
NOTICE OF RECORDATION OF ASSIGNMENT DOCUMENT

THE ENCLOSED DOCUMENT HAS BEEN RECORDED BY THE ASSIGNMENT RECORDATION BRANCH OF THE U.S. PATENT AND TRADEMARK OFFICE. A COMPLETE COPY IS AVAILABLE AT THE ASSIGNMENT SEARCH ROOM ON THE REEL AND FRAME NUMBER REFERENCED BELOW.

PLEASE REVIEW ALL INFORMATION CONTAINED ON THIS NOTICE. THE INFORMATION CONTAINED ON THIS RECORDATION NOTICE REFLECTS THE DATA PRESENT IN THE PATENT AND TRADEMARK ASSIGNMENT SYSTEM. IF YOU SHOULD FIND ANY ERRORS OR HAVE QUESTIONS CONCERNING THIS NOTICE, YOU MAY CONTACT THE ASSIGNMENT RECORDATION BRANCH AT 571-272-3350. PLEASE SEND REQUEST FOR CORRECTION TO: U.S. PATENT AND TRADEMARK OFFICE, MAIL STOP: ASSIGNMENT RECORDATION BRANCH, P.O. BOX 1450, ALEXANDRIA, VA 22313.

RECORDATION DATE: 01/19/2012

REEL/FRAME: 027558/0281
NUMBER OF PAGES: 3

BRIEF: CHANGE OF ADDRESS OF ASSIGNEE

DOCKET NUMBER: 077580-0010

ASSIGNOR:
VIRNETX INC.

DOC DATE: 01/19/2012

ASSIGNEE:
VIRNETX INC.
P.O. BOX 439
ZEPHYR COVE, NEVADA 89448

APPLICATION NUMBER: 09429643
PATENT NUMBER: 7010604
TITLE: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

FILING DATE: 10/29/1999
ISSUE DATE: 03/07/2006

APPLICATION NUMBER: 09504783
PATENT NUMBER: 6502135
TITLE: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILTY

FILING DATE: 02/15/2000
ISSUE DATE: 12/31/2002

APPLICATION NUMBER: 09874258
PATENT NUMBER: 7209479
TITLE: THIRD PARTY VPN CERTIFICATION

FILING DATE: 06/06/2001
ISSUE DATE: 04/24/2007

APPLICATION NUMBER: 10082164 FILING DATE: 02/26/2002
PATENT NUMBER: 6618761 ISSUE DATE: 09/09/2003
TITLE: AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY USING WEIGHTED TRANSMISSION PATHS

APPLICATION NUMBER: 10082285 FILING DATE: 02/26/2002
PATENT NUMBER: 6834310 ISSUE DATE: 12/21/2004
TITLE: PREVENTING PACKET FLOODING OF A COMPUTER ON A COMPUTER NETWORK

APPLICATION NUMBER: 10259494 FILING DATE: 09/30/2002
PATENT NUMBER: 7490151 ISSUE DATE: 02/10/2009
TITLE: ESTABLISHMENT OF A SECURE COMMUNICATION LINK BASED ON A DOMAIN NAME SERVICE (DNS) REQUEST

APPLICATION NUMBER: 10401551 FILING DATE: 03/31/2003
PATENT NUMBER: 7133930 ISSUE DATE: 11/07/2006
TITLE: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

APPLICATION NUMBER: 10401888 FILING DATE: 03/31/2003
PATENT NUMBER: 6907473 ISSUE DATE: 06/14/2005
TITLE: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

APPLICATION NUMBER: 10702486 FILING DATE: 11/07/2003
PATENT NUMBER: 7188180 ISSUE DATE: 03/06/2007
TITLE: METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN COMPUTERS OF VIRTUAL PRIVATE NETWORK

APPLICATION NUMBER: 10702522 FILING DATE: 11/07/2003
PATENT NUMBER: 6839759 ISSUE DATE: 01/04/2005
TITLE: METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN COMPUTERS OF VIRTUAL PRIVATE NETWORK WITHOUT USER ENTERING ANY CRYPTOGRAPHIC INFORMATION

APPLICATION NUMBER: 10702580 FILING DATE: 11/07/2003
PATENT NUMBER: 6826616 ISSUE DATE: 11/30/2004
TITLE: METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN COMPUTERS OF VIRTUAL PRIVATE NETWORK

APPLICATION NUMBER: 10714849 FILING DATE: 11/18/2003
PATENT NUMBER: 7418504 ISSUE DATE: 08/26/2008
TITLE: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS USING SECURE DOMAIN NAMES

APPLICATION NUMBER: 11301022 FILING DATE: 12/13/2005
PATENT NUMBER: 7996539 ISSUE DATE: 08/09/2011
TITLE: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

APPLICATION NUMBER: 11679416 FILING DATE: 02/27/2007
PATENT NUMBER: 8051181 ISSUE DATE: 11/01/2011
TITLE: METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN COMPUTERS OF VIRTUAL PRIVATE NETWORK

APPLICATION NUMBER: 11839969 FILING DATE: 08/16/2007
PATENT NUMBER: 7933990 ISSUE DATE: 04/26/2011
TITLE: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

APPLICATION NUMBER: 11839987 FILING DATE: 08/16/2007
PATENT NUMBER: 7987274 ISSUE DATE: 07/26/2011
TITLE: METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN COMPUTERS OF VIRTUAL PRIVATE NETWORK

APPLICATION NUMBER: 11840508 FILING DATE: 08/17/2007
PATENT NUMBER: 7945654 ISSUE DATE: 05/17/2011
TITLE: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS USING SECURE DOMAIN NAMES

APPLICATION NUMBER: 11840560 FILING DATE: 08/17/2007
PATENT NUMBER: 7921211 ISSUE DATE: 04/05/2011
TITLE: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS USING SECURE DOMAIN NAMES

APPLICATION NUMBER: 11840579 FILING DATE: 08/17/2007
PATENT NUMBER: 7944915 ISSUE DATE: 05/17/2011
TITLE: THIRD PARTY VPN CERTIFICATION

APPLICATION NUMBER: 11924460 FILING DATE: 10/25/2007
PATENT NUMBER: ISSUE DATE:
TITLE: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

APPLICATION NUMBER: 13080684 FILING DATE: 04/06/2011
PATENT NUMBER: ISSUE DATE:
TITLE: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

ASSIGNMENT RECORDATION BRANCH
PUBLIC RECORDS DIVISION

PTO/SB/44 (09-07)
Approved for use through 08/31/2013. OMB 0651-0033
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

(Also Form PTO-1050)

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 7,188,180

APPLICATION NO.: 10/702,486

ISSUE DATE : March 6, 2007

INVENTOR(S) : Victor Larson et al.

It is certified that an error appears or errors appear in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

IN PATENT TITLE PAGE

Item (73), Assignee:, delete the assignee name and address listed, and insert the following:

-- VIRNETX, INC., Zephyr Cove, NV (US) --

MAILING ADDRESS OF SENDER (Please do not use customer number below):

Atabak R. Royaee, Reg. No. 59,037
McDERMOTT WILL & EMERY LLP
28 State Street, Boston, MA, 02109

PTO/SB/44 (09-07)
Approved for use through 08/31/2013. OMB 0651-0033
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

(Also Form PTO-1050)

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 7,188,180

APPLICATION NO.: 10/702,486

ISSUE DATE : March 6, 2007

INVENTOR(S) : Victor Larson et al.

It is certified that an error appears or errors appear in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

IN PATENT TITLE PAGE

Item (73), Assignee:, delete the assignee name and address listed, and insert the following:

-- VIRNETX, INC., Zephyr Cove, NV (US) --

MAILING ADDRESS OF SENDER (Please do not use customer number below):

Atabak R. Royaee, Reg. No. 59,037
McDERMOTT WILL & EMERY LLP
28 State Street, Boston, MA, 02109

Electronic Acknowledgement Receipt

EFS ID:	13233614
Application Number:	10702486
International Application Number:	
Confirmation Number:	8949
Title of Invention:	METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN COMPUTERS OF VIRTUAL PRIVATE NETWORK
First Named Inventor/Applicant Name:	Victor Larson
Customer Number:	23630
Filer:	Atabak R Royaee/Jacqueline Andreu
Filer Authorized By:	Atabak R Royaee
Attorney Docket Number:	77580-039 (VRNK-1CP2DV)
Receipt Date:	12-JUL-2012
Filing Date:	07-NOV-2003
Time Stamp:	13:27:48
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Request for Certificate of Correction	Request_For_Certificate_of_Corrections.pdf	263549 11ec72ab2ee4c2b59e585b3c9fbebaa51f63a3b2	no	8

Warnings:

The page size in the PDF is too large. The pages should be 8.5 x 11 or A4. If this PDF is submitted, the pages will be resized upon entry into the Image File Wrapper and may affect subsequent processing
Information:

2	Miscellaneous Incoming Letter	Certificate_of_Correction.pdf	57230 bca7713eaf5415d0ecbb47bf5fe83bd74351309f	no	2

Warnings:
The page size in the PDF is too large. The pages should be 8.5 x 11 or A4. If this PDF is submitted, the pages will be resized upon entry into the Image File Wrapper and may affect subsequent processing
Information:

Total Files Size (in bytes):	320779

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PATENT
Customer No. 22,852
Attorney Docket No. 11798.0005

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re *Inter Partes* Reexamination of:	)	
	)	
Victor LARSON et al.	)	Control No.: 95/001,792
	)	
U. S. Patent No. 7,188,180	)	Group Art Unit: 3992
	)	
Issued: March 7, 2007	)	Examiner: Deandra M. Hughes
	)	
For: METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN COMPUTERS OF VIRTUAL PRIVATE NETWORK	)	Confirmation No. 1972

Mail Stop *Inter Partes* Reexam
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Commissioner:

REVOCATION OF POWER OF ATTORNEY,
STATEMENT UNDER 37 C.F.R. § 3.73(b),
AND GRANT OF NEW POWER OF ATTORNEY

The undersigned, a representative authorized to sign on behalf of the assignee owning all of the interest in U.S. Patent No. 7,188,180 ("the '180 patent"), hereby revokes all previous powers of attorney or authorization of agent granted in the '180 patent before the date of execution hereof.

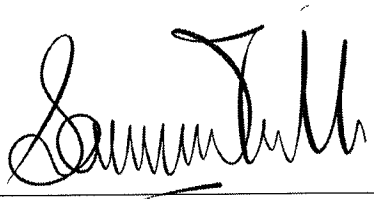
In compliance with 37 C.F.R. § 3.73(b), the undersigned verifies that VirnetX Inc. is the assignee of the entire right, title, and interest in the '180 patent by virtue of an assignment recorded in the U.S. Patent and Trademark Office at Reel 018757, Frame 0326 on January 10, 2007.

The undersigned representative of the assignee hereby grants its power of attorney to the patent practitioners associated with **Finnegan, Henderson, Farabow, Garrett & Dunner,**

L.L.P., Customer Number 22,852, to transact all business in the Patent and Trademark Office connected with the '180 patent, including the reexamination proceedings assigned control no. 95/001,792, and in any other proceedings involving the '180 patent.

Please also send all future correspondence concerning the '180 patent to the address associated with **Finnegan, Henderson, Farabow, Garrett & Dunner, L.L.P., Customer Number 22,852.**

Dated: 11/30/12

By: 

Sameer Mathur
Vice President, Corporate Development and Product Marketing
VirnetX Inc.

Electronic Acknowledgement Receipt

EFS ID:	14369533
Application Number:	10702486
International Application Number:	
Confirmation Number:	8949
Title of Invention:	METHOD FOR ESTABLISHING SECURE COMMUNICATION LINK BETWEEN COMPUTERS OF VIRTUAL PRIVATE NETWORK
First Named Inventor/Applicant Name:	Victor Larson
Customer Number:	23630
Filer:	Joseph Edwin Palys./connie sisk
Filer Authorized By:	Joseph Edwin Palys.
Attorney Docket Number:	77580-039 (VRNK-1CP2DV)
Receipt Date:	03-DEC-2012
Filing Date:	07-NOV-2003
Time Stamp:	16:33:58
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Power of Attorney	Patent_POA_180.pdf	55065 ab2fda08717c5150903d7faac453b25c851c a11c	no	2

Warnings:	
Information:	

Total Files Size (in bytes):	55065

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
10/702,486	11/07/2003	Victor Larson	77580-039 (VRNK-1CP2DV)

CONFIRMATION NO. 8949

POWER OF ATTORNEY NOTICE

23630
McDermott Will & Emery
The McDermott Building
500 North Capitol Street, N.W.
Washington, DC 20001

OC000000058101937

Date Mailed: 12/14/2012

NOTICE REGARDING CHANGE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 12/03/2012.

- The Power of Attorney to you in this application has been revoked by the assignee who has intervened as provided by 37 CFR 3.71. Future correspondence will be mailed to the new address of record(37 CFR 1.33).

/zabraha/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
10/702,486	11/07/2003	Victor Larson	11798.0005

CONFIRMATION NO. 8949

POA ACCEPTANCE LETTER

22852
FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER LLP
901 NEW YORK AVENUE, NW
WASHINGTON, DC 20001-4413

OC000000058101960

Date Mailed: 12/14/2012

NOTICE OF ACCEPTANCE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 12/03/2012.

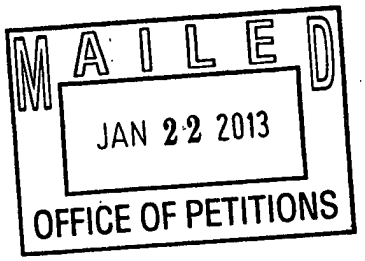
The Power of Attorney in this application is accepted. Correspondence in this application will be mailed to the above address as provided by 37 CFR 1.33.

/zabraha/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov



FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER LLP
901 NEW YORK AVENUE, NW
WASHINGTON DC 20001-4413

In re Patent No. 7,188,180 :
Issue Date: March 6, 2007 :
Application No. 10/702,486 : DECISION ON PETITION
Filed: November 7, 2003 :
Attorney Docket No. 11798.0005 :

This is a decision on the Request For Certificate Of Correction Under 37 CFR §§1.322 & 1.323, filed July 12, 2012, requesting correction on the Title Page of the subject patent, to correct assignee's name. The Request is being treated as a Petition under 37 CFR §1.322(a) for which no fee is required. A completed Certificate of Correction Form (PTO/SB/44) was submitted with the petition.

The petition under 37 CFR §1.322(a) is **GRANTED**.

Petitioner requests that the present Petition was submitted to correct assignee's name on the previously submitted PTOL-85b, filed January 16, 2007, and that such error occurred through the fault of the Patent Office. Accordingly, petitioner requests, in effect, that the Title Page of the above-identified patent be corrected, via issuance of a Certificate of Correction, to correct the spelling of the assignee's name identified thereon from:

"VimetX, Inc.

to

--VIRNETX, INC.--

Telephone inquiries related this communication should be directed to the undersigned at (571)272-3213. Inquiries regarding this issuance of a certificate of correction should be directed to the Certificate of Correction Branch at (703)756-1814.

The Certificates of Correction Branch will be notified of this decision granted the petition under 37 CFR §1.322(a) and directing issuance of the requested Certificate of Correction.

Cheryl Gibson-Baylor
Petitions Examiner
Office of Petitions

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 7,188,180 B2
APPLICATION NO. : 10/702486
DATED : March 6, 2007
INVENTOR(S) : Victor Larson et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

TITLE PAGE

Item (73), Assignee:, delete the assignee name and address listed, and insert the following:

-- VIRNETX, INC., Zephyr Cove, NV (US) --

Signed and Sealed this
Twenty-sixth Day of February, 2013

Teresa Stanek Rea
Acting Director of the United States Patent and Trademark Office

AO 120 (Rev. 08/10)

TO:	REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK
Mail Stop 8 Director of the U.S. Patent and Trademark Office P.O. Box 1450 Alexandria, VA 22313-1450	

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court Eastern District of Texas Tyler Division on the following

☐ Trademarks or ☑ Patents. (☐ the patent action involves 35 U.S.C. § 292.):

DOCKET NO.	DATE FILED	U.S. DISTRICT COURT
6:13-cv-00351	4/22/2013	Eastern District of Texas Tyler Division

PLAINTIFF	DEFENDANT
VirnetX Inc. and Science Applications International Corporation	Microsoft Corporation

	PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1	6,502,135	12/31/2002	VirnetX Inc.
2	7,188,180	3/6/2007	VirnetX Inc.
3	7,418,504	8/26/2008	VirnetX Inc.
4	7,490,151	2/10/2009	VirnetX Inc.
5	7,921,211	4/5/2011	VirnetX Inc.

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY ☐ Amendment ☐ Answer ☐ Cross Bill ☐ Other Pleading	
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 7,188,274		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director **Copy 3—Upon termination of action, mail this copy to Director**
Copy 2—Upon filing document adding patent(s), mail this copy to Director **Copy 4—Case file copy**

AO 120 (Rev. 08/10)

TO:	REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK
Mail Stop 8 Director of the U.S. Patent and Trademark Office P.O. Box 1450 Alexandria, VA 22313-1450	

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court Eastern District of Texas Tyler Division on the following

☐ Trademarks or ☑ Patents. (☐ the patent action involves 35 U.S.C. § 292.):

DOCKET NO.	DATE FILED	U.S. DISTRICT COURT
6:13-cv-00351	4/22/2013	Eastern District of Texas Tyler Division

PLAINTIFF	DEFENDANT
VirnetX Inc. and Science Applications International Corporation	Microsoft Corporation

	PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1	6,502,135	12/31/2002	VirnetX Inc.
2	7,188,180	3/6/2007	VirnetX Inc.
3	7,418,504	8/26/2008	VirnetX Inc.
4	7,490,151	2/10/2009	VirnetX Inc.
5	7,921,211	4/5/2011	VirnetX Inc.

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY ☐ Amendment ☐ Answer ☐ Cross Bill ☐ Other Pleading	
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 7,987,274		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director **Copy 3—Upon termination of action, mail this copy to Director**
Copy 2—Upon filing document adding patent(s), mail this copy to Director **Copy 4—Case file copy**