Exacq Technologies, Inc.
Exhibit 1012

US006477648B1

(12) **United States Patent**
Schell et al.

(10) **Patent No.: US 6,477,648 B1**
(45) **Date of Patent: *Nov. 5, 2002**

(54) **TRUSTED WORKSTATION IN A NETWORKED CLIENT/SERVER COMPUTING SYSTEM**

(75) Inventors: **Roger R. Schell**, Orem, UT (US); **Douglas Lavell Hale**, Orem, UT (US); **Willard Monten Wiseman**, Pleasant Grove, UT (US); **James P. Anderson**, Ambler, PA (US)

(73) Assignee: **Novell, Inc.**, Provo, UT (US)

(*) Notice: This patent issued on a continued prosecution application filed under 37 CFR 1.53(d), and is subject to the twenty year patent term provisions of 35 U.S.C. 154(a)(2).

Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 737 days.

(21) Appl. No.: **08/828,724**

(22) Filed: **Mar. 23, 1997**

(51) **Int. Cl.**[7] **G06F 11/00**
(52) **U.S. Cl.** **713/200**
(58) **Field of Search** 713/1, 2, 176, 713/200, 201, 202

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,196,840	A		3/1993	Leith et al.
5,202,997	A		4/1993	Arato
5,204,961	A		4/1993	Barlow
5,210,795	A	*	5/1993	Lipner et al. 380/23
5,287,519	A		2/1994	Dayan et al.
5,313,637	A		5/1994	Rose
5,341,422	A		8/1994	Blackledge et al.
5,349,642	A		9/1994	Kingdon
5,355,489	A	*	10/1994	Bealkowski et al. 395/700
5,426,775	A	*	6/1995	Boccon-Gibod 395/575
5,444,850	A		8/1995	Chang
5,448,045	A	*	9/1995	Clark 380/4
5,465,357	A	*	11/1995	Bealkowski et al. 395/700
5,701,491	A	*	12/1997	Dunn et al. 395/712
5,710,817	A	*	1/1998	Sjooquist 380/25
5,740,371	A	*	4/1998	Wallis 395/200.59
5,828,888	A	*	10/1998	Kozaki et al. 395/712

OTHER PUBLICATIONS

The X–Lock Virus Checker, Chapter 7, 1992.*

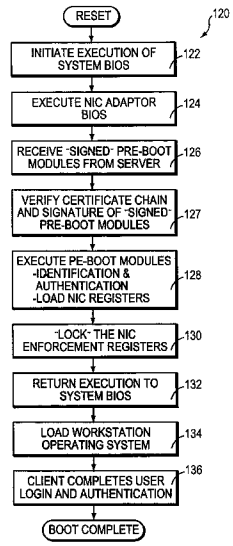
* cited by examiner

Primary Examiner—Scott Baderman
(74) *Attorney, Agent, or Firm*—Schwegman, Lundberg, Woessner & Kluth, P.A.

(57) **ABSTRACT**

A trusted workstation includes a network interface card (NIC) with trusted computing base (TCB) extensions that provide for securely booting the workstation and performing subsequent receive and transmit packet filtering in support of a network's system architecture requirements. The NIC includes a send address confirm circuit which includes a trusted source address (e.g., a MAC address) uniquely associated with the trusted workstation. For each packet to be transmitted from the trusted workstation over the network, the NIC first checks the source address inserted in the packet by the NIC driver running in the user session to be sure that the driver inserted source address is to equal to the trusted address resident. Thus, if untrusted software on the workstation attempts mischiefly transmit a forged packet with a source address other than the trusted source address, the NIC prohibits transmission of the packet with the forged source address. This prevents the trusted workstation from forging its packets with another client's source address. The NIC also includes a receive address confirmation circuit which ensures that the trusted workstation only receives packets from authorized servers.

15 Claims, 6 Drawing Sheets



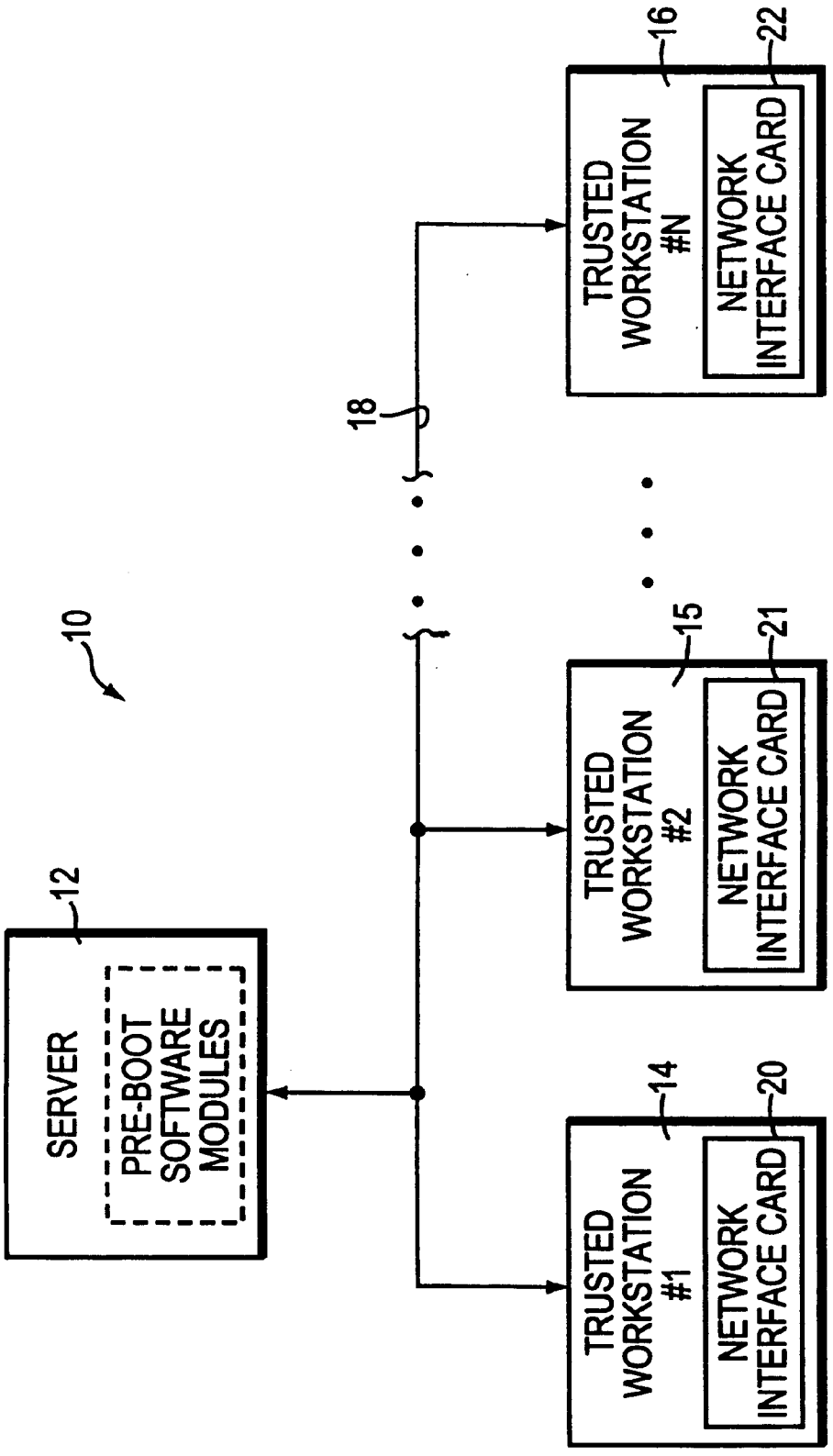

10
SERVER
PRE-BOOT SOFTWARE MODULES
12
18
TRUSTED WORKSTATION #1
14
NETWORK INTERFACE CARD
20
TRUSTED WORKSTATION #2
15
NETWORK INTERFACE CARD
21
TRUSTED WORKSTATION #N
16
NETWORK INTERFACE CARD
22

FIG. 1

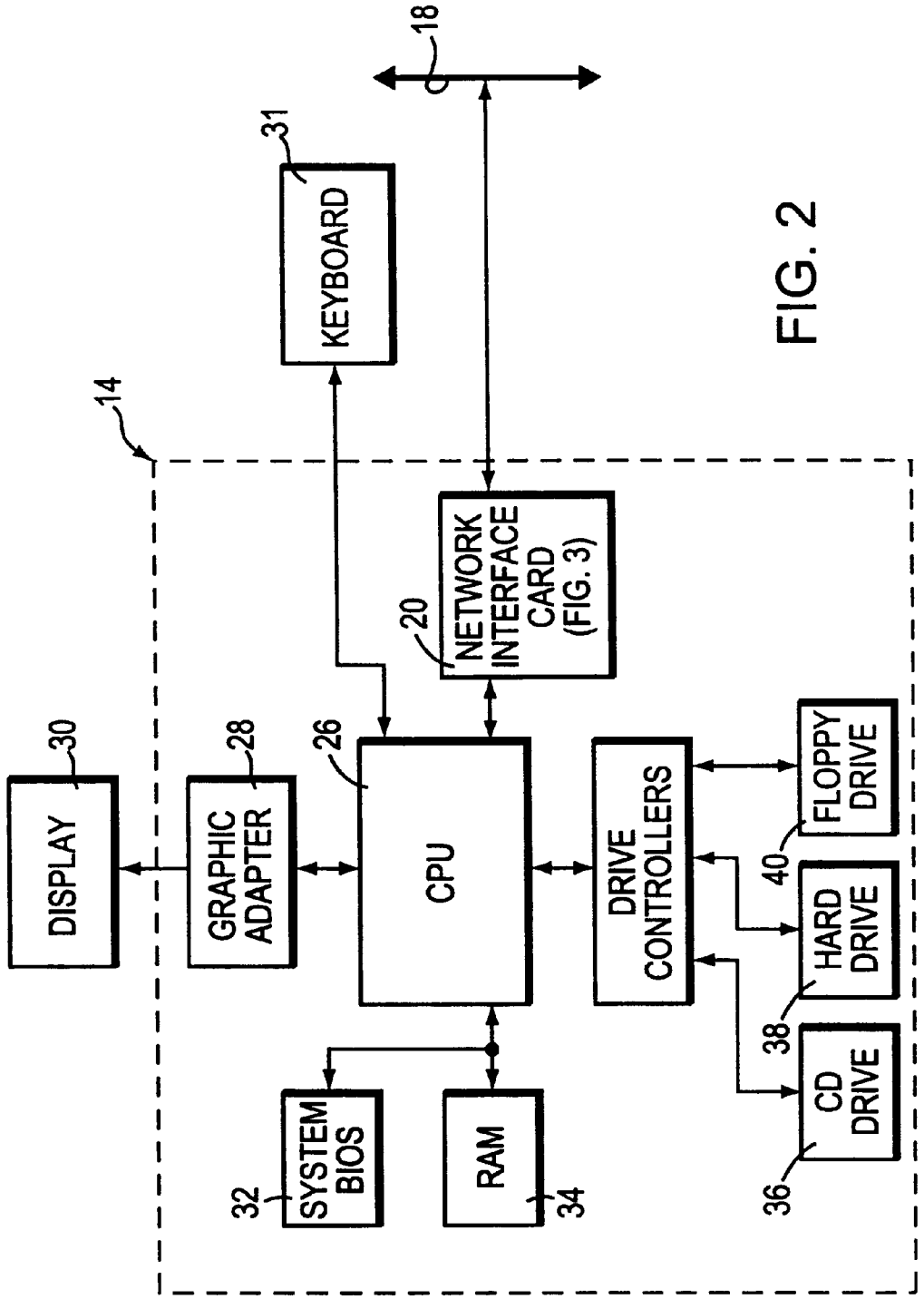

18
31
KEYBOARD
14
NETWORK INTERFACE CARD (FIG. 3)
20
30
28
26
DISPLAY
GRAPHIC ADAPTER
CPU
DRIVE CONTROLLERS
FLOPPY DRIVE
40
HARD DRIVE
38
CD DRIVE
36
SYSTEM BIOS
32
RAM
34

FIG. 2

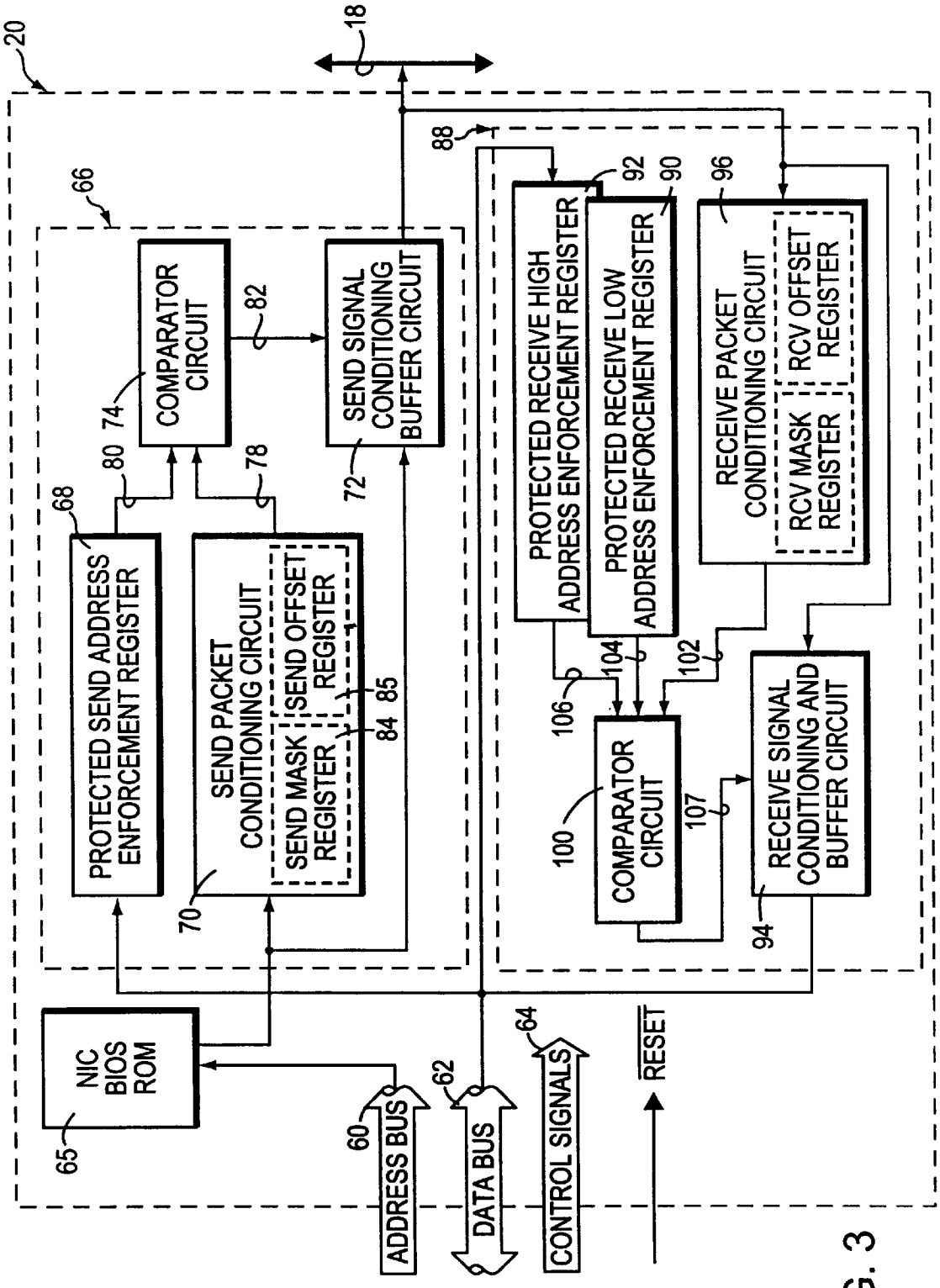

20
18
66
88
COMPARATOR CIRCUIT
74
82
SEND SIGNAL CONDITIONING BUFFER CIRCUIT
72
80
78
68
PROTECTED SEND ADDRESS ENFORCEMENT REGISTER
SEND PACKET CONDITIONING CIRCUIT
SEND OFFSET REGISTER
SEND MASK REGISTER
84
85
70
PROTECTED RECEIVE HIGH ADDRESS ENFORCEMENT REGISTER
92
PROTECTED RECEIVE LOW ADDRESS ENFORCEMENT REGISTER
90
RECEIVE PACKET CONDITIONING CIRCUIT
96
RCV MASK REGISTER
RCV OFFSET REGISTER
104
102
106
100
COMPARATOR CIRCUIT
107
RECEIVE SIGNAL CONDITIONING AND BUFFER CIRCUIT
94
NIC BIOS ROM
65
60
ADDRESS BUS
62
DATA BUS
64
CONTROL SIGNALS
RESET

FIG. 3

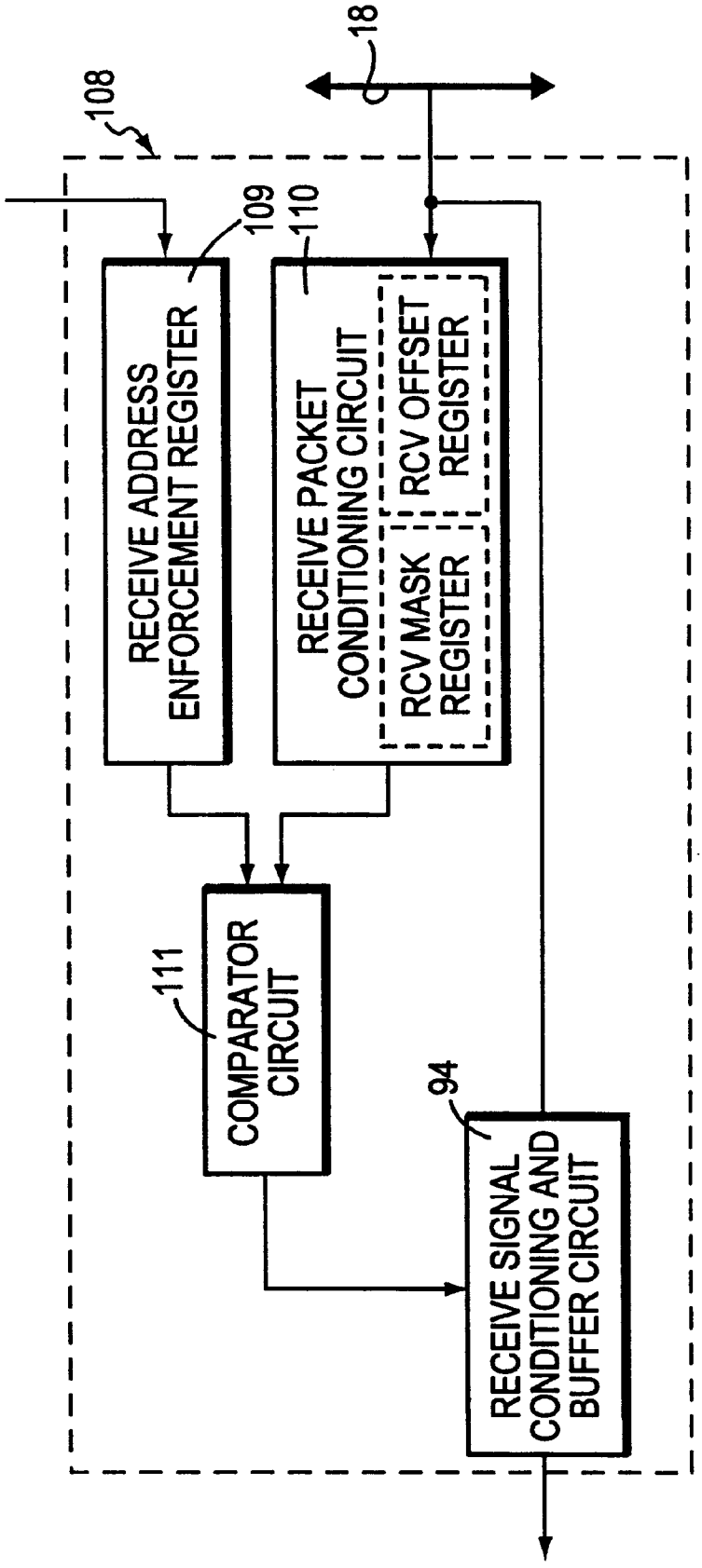

18
108
109
110
RECEIVE ADDRESS ENFORCEMENT REGISTER
RECEIVE PACKET CONDITIONING CIRCUIT
RCV MASK REGISTER
RCV OFFSET REGISTER
111
COMPARATOR CIRCUIT
94
RECEIVE SIGNAL CONDITIONING AND BUFFER CIRCUIT

FIG. 4

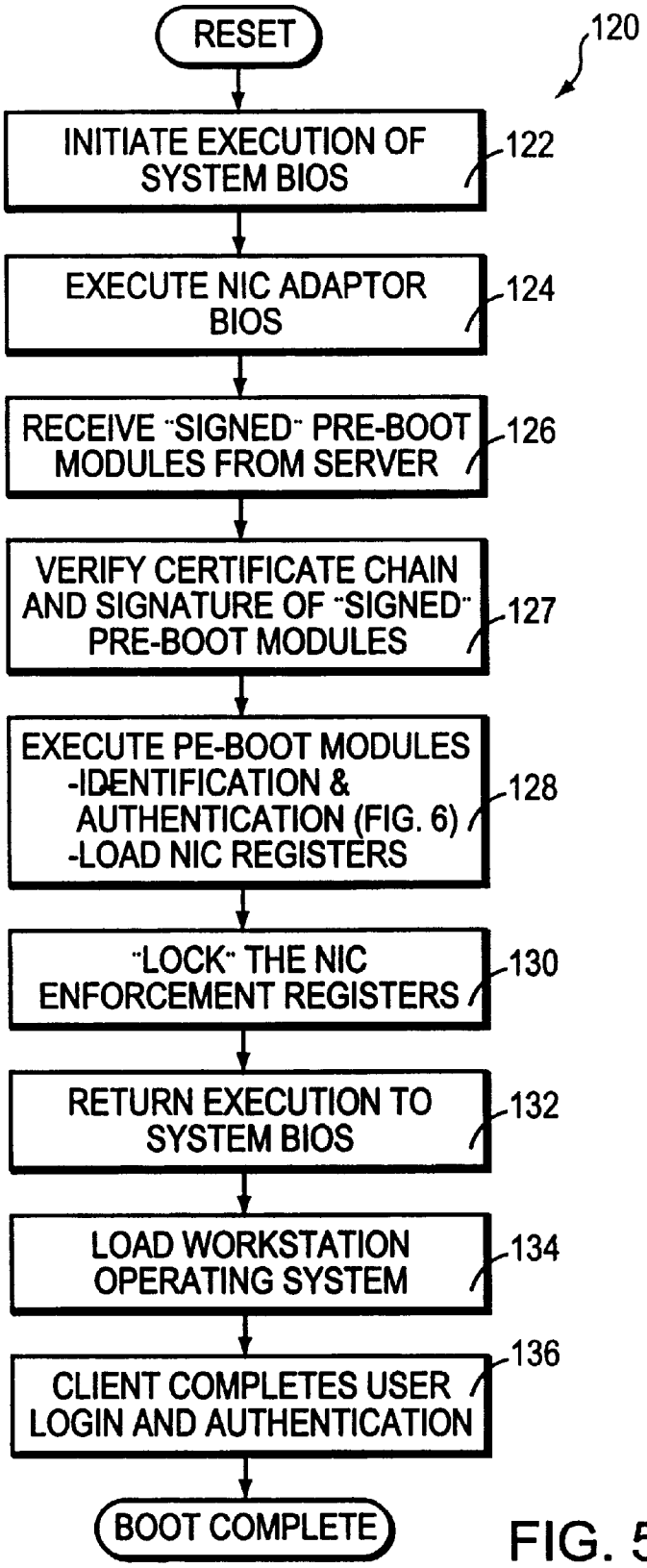

RESET
120
INITIATE EXECUTION OF SYSTEM BIOS
122
EXECUTE NIC ADAPTOR BIOS
124
RECEIVE "SIGNED" PRE-BOOT MODULES FROM SERVER
126
VERIFY CERTIFICATE CHAIN AND SIGNATURE OF "SIGNED" PRE-BOOT MODULES
127
EXECUTE PE-BOOT MODULES
-IDENTIFICATION & AUTHENTICATION (FIG. 6)
-LOAD NIC REGISTERS
128
"LOCK" THE NIC ENFORCEMENT REGISTERS
130
RETURN EXECUTION TO SYSTEM BIOS
132
LOAD WORKSTATION OPERATING SYSTEM
134
CLIENT COMPLETES USER LOGIN AND AUTHENTICATION
136
BOOT COMPLETE

FIG. 5

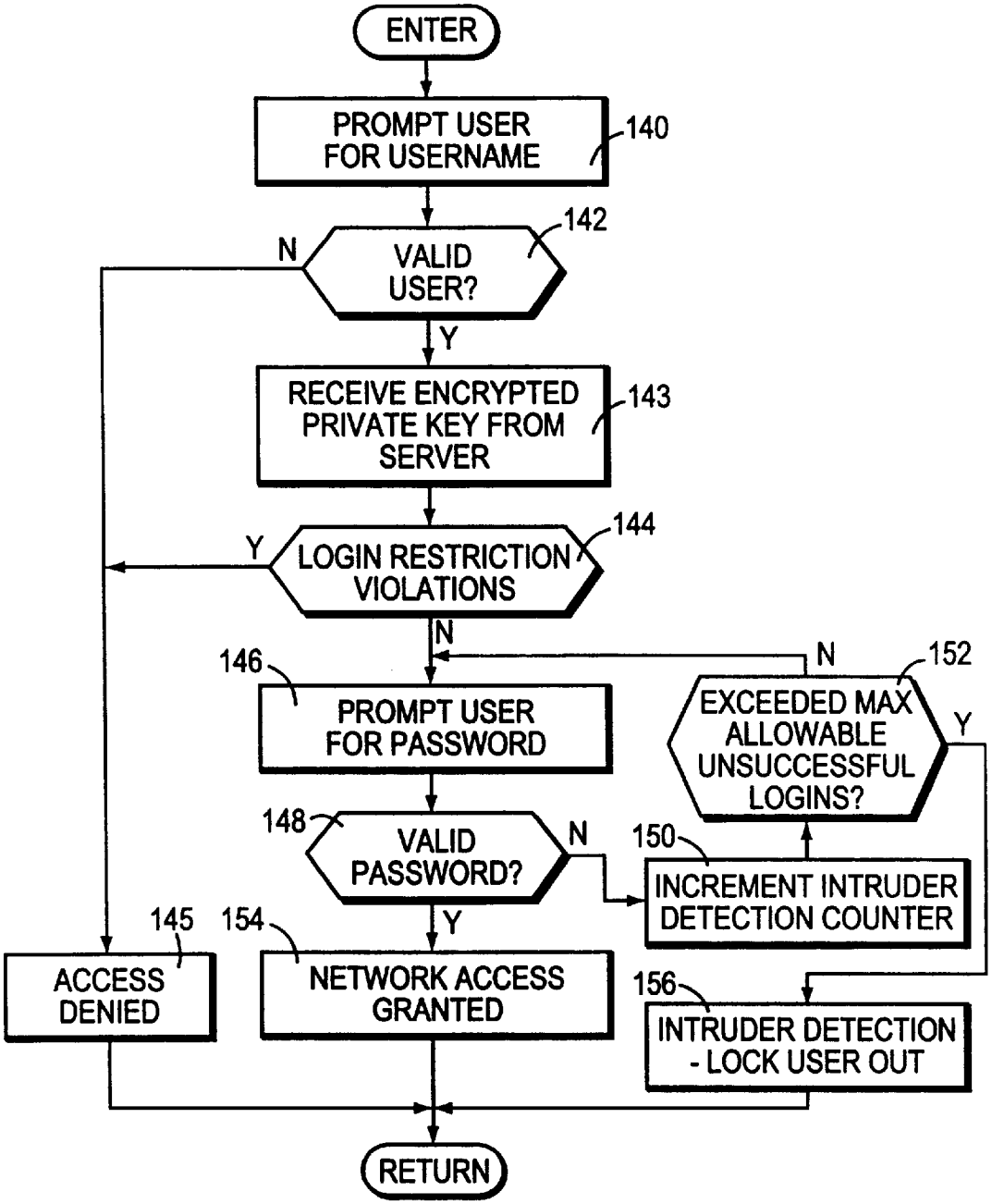

ENTER
PROMPT USER FOR USERNAME
140
VALID USER?
142
N
Y
RECEIVE ENCRYPTED PRIVATE KEY FROM SERVER
143
LOGIN RESTRICTION VIOLATIONS
144
Y
N
146
PROMPT USER FOR PASSWORD
148
VALID PASSWORD?
N
Y
150
INCREMENT INTRUDER DETECTION COUNTER
152
EXCEEDED MAX ALLOWABLE UNSUCCESSFUL LOGINS?
N
Y
145
ACCESS DENIED
154
NETWORK ACCESS GRANTED
156
INTRUDER DETECTION - LOCK USER OUT
RETURN

FIG. 6

TRUSTED WORKSTATION IN A NETWORKED CLIENT/SERVER COMPUTING SYSTEM

CROSS-REFERENCE TO RELATED APPLICATION

The subject matter of this application is related to that of copending application Ser. No. 08/907,523, filed Aug. 8, 1997, entitled "Networked Workstation Intrusion Detection System", which is assigned to the Assignee of the subject application.

TECHNICAL FIELD

The present invention relates to a computer systems, and in particular to a networked client/server computer system configured to establish a trusted workstation.

BACKGROUND OF THE INVENTION

Client/server computing has become quite a popular architecture in both small and large organizations. As known, these systems include a computer system which operates as a server for a plurality of personal computers and/or workstations, which are generally connected to the server via a network connection comprising a local area network (LAN) or a wide area network (WAN).

Client/server computing networks have dramatically increased and facilitated the access to information. However, due to ubtiqious nature of computer networks the threat to the integrity of the information stored on network resources due to "hackers"/"attackers" and malicious software components (e.g., operating system and application program viruses) has also increased. Threats include any person, place or thing which poses some danger to a network asset.

Security of the information transmitted over the network must be assured when the network is used to transmit information for businesses such as banking, brokerage, government entities and other users of highly confidential or commercially valuable information. A known threat to the security of information available on a network is a hacker/attacker who poses as an authorized user of the network by impersonating the authorized user. Passwords and other similar operating system level security features often only make it difficult for the hacker/attacker to gain access to the network. However, a patient and capable hacker/attacker can generally bypass most conventional operating system level protections to access the network.

Therefore, there is a need for a technique for ensuring the security of the information stored on a client/server networked computer system and to provide a secure, trusted workstation.

SUMMARY OF THE INVENTION

An object of the present invention is to provide a trusted workstation.

Another object is to restrict communications between a trusted workstation and a known/authorized server.

A further object is to provide a trusted distributed data processing system.

Yet another object is to prevent an unauthorized network user from impersonating an authorized network user.

Briefly, according to the present invention, a trusted workstation includes a network interface card (NIC) with trusted computing base (TCB) extensions that provide for securely booting the workstation and performing subsequent receive and transmit packet filtering operation in support of a network's system architecture requirements. The term "TCB extension" refers to extensions of the server's TCB that operate as part of the workstation's network trusted computing base (NTCB).

The NIC comprises a send address confirmation circuit which contains a trusted source address (e.g., a medium access control (MAC) address or a network layer address) uniquely associated with the trusted workstation. In general, the source address can be any address that identifies the source of a packet, including for example the MAC address, Internet address, transport layer address or session layer address, etcetera. For each packet transmitted from the trusted workstation over the network, the NIC checks the source address inserted in the packet by an NIC driver to ensure that this driver-inserted source address matches the trusted source address. Thus, if untrusted software on the workstation attempts to transmit a packet with a source address other than the trusted source address, the NIC prevents the packet from being transmitted. This prevents malicious attempts by a hacker/attacker to forge packets from a workstation with another workstation's source address.

The NIC also includes a receive address confirmation circuit that functions to ensure that the trusted workstation does not receive packets from entities other than known/authorized servers. That is, the NIC compares the source address of a packet received over the network to verify that it is from a authorized server. Significantly, if each workstation on a network is populated with a NIC, the known/authorized servers will control all packets on the network and can trust the source of all requests.

The send and receive address confirmation circuits are trusted because the contents of registers resident in these circuits are written to and modifiable only during a pre-boot state, which is the only time the untrusted elements (e.g., untrusted software on the workstation) are not accessed. That is, following a hardware reset and prior to execution of the operating or application software on the workstation, enforcement registers with the send and receive conformation circuits are written to with source address data, and then write disabled to prevent subsequent loading of unauthorized source address data.

Specifically, following a hardware reset of the workstation, the NIC is initialized and pre-boot modules are downloaded to the workstation over the network from a known server under the control of instructions resident in an adapter BIOS on the NIC. The NIC may be located on an expansion board separate (e.g., ISA or PCI compatible) from the workstation motherboard, or on the motherboard. Once the pre-boot modules are down loaded to the workstation, the pre-boot modules are executed to perform a login—identification and authentication (I & A) function for the user and to load the enforcement registers with the send and receive trusted source address information. The enforcement registers are then locked (i.e., write disable) to prevent the contents of the send and receive enforcement registers from being modified until another hardware reset occurs. Once execution of the pre-boot modules is complete, the NIC BIOS transfers code execution to a workstation system BIOS to complete the initialization of the workstation.

The pre-boot modules resident in the NIC BIOS for performing the I & A function include executable code which communicates with the server to verify the identity of the user, log the user into the network once the identity is verified, and establish a connection with the server.

The NIC can enforce as a source address (i.e., compare source addresses) associated with the data link layer address, network layer source addresses or any other address that is used to identify the source of the packet. The data link layer address is the network medium's address and is a hardware based value which is stored in the NIC. For Ethernet networks this address is often referred to as the MAC address. The network layer address is a protocol specific logical address, and therefore is understood to be above the data link layer address in the protocol stack associated with the network. To implement network layer source address enforcement, few intermediate network components (e.g., routers and bridges) need to be involved since the network layer source address is left unchanged by retransmission devices. However, with data link layer source address enforcement, at least one of the retransmission devices needs to be involved in the source address enforcement since the data link layer source address is modified as it traverses many of these devices, including routers.

An advantage of the present invention is that it provides an inexpensive technique for providing trusted workstations suitable for use in networks with heightened security requirements.

These and other objects, features and advantages of the present invention will become more apparent in light of the following detailed description of preferred embodiments thereof, as illustrated in the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a block diagram of a networked computer system which includes a plurality of trusted workstations;

FIG. **2** is a simplified block diagram of a trusted workstation from FIG. **1**;

FIG. **3** is a block diagram of a network interface card resident in the trusted workstation illustrated in FIG. **2**;

FIG. **4** is a block diagram of an alternative embodiment of the receive address confirmation circuit illustrated in FIG. **3**;

FIG. **5** is a flowchart illustration of the steps performed during system initialization; and

FIG. **6** is a flowchart illustration of the steps associated with the login process.

DESCRIPTION OF A PREFERRED EMBODIMENT

Referring to FIG. **1**, a network client server computing system **10** includes a server **12** and a plurality of trusted workstations (e.g., personal computers) **14–16**. The server **12** and workstations **14–16** are interconnected via a computer network **18** such as a local area network (LAN) or a wide area network (WAN) and communicate by exchanging discrete packets or frames of data over the computer network **18** according to known network protocols. The computer network **18** may include network intermediate connections such as a router or bridge (not shown) depending upon the span of the network. According to the present invention, each of the trusted workstations **14–16** includes an associated network interface card **20–22**, respectively, which establishes a trusted connection between the workstation and the server **12**.

FIG. **2** is a simplified block diagram illustration of the trusted workstation **14**. The workstation **14** includes conventional components such as a central processing unit (CPU) **26**, a graphics adapter **28**, system basic input/output system (BIOS) **32**, and computer readable media such as RAM **34**, CD drive **36**, hard drive **38** and a floppy drive **40**. The workstation also includes the network interface card (NIC) **20** through which the workstation communicates with the server **12** (FIG. **1**) over the computer network **18**. The workstation **14** is connected to a display **30** and a keyboard **31**.

FIG. **3** is a schematic block diagram illustration of the NIC **20**. The NIC is preferably an expansion card (e.g., ISA or PCI compatible) which is connected to the CPU **26** (FIG. **2**) via an address bus **60**, a data bus **62** and a control signal bus **64**. The signals from these buses are routed to the various components resident on the NIC in order to perform the functions discussed hereinbelow.

The NIC **20** includes send address confirmation circuitry **66** which includes a protected send address enforcement register **68**, a packet conditioning circuit **70**, send signal conditioning buffer circuitry **72**, and a comparator circuit **74**. During initialization the protected send address enforcement register **68** is loaded with a trusted source address value indicative of a link layer address (e.g., a MAC address or network layer address.) uniquely associated with the workstation **14**. The initialization steps shall be discussed hereinafter in detail. To ensure that the packet to be transmitted from the workstation has a valid source address (i.e., the source address has not been forged) the packet is input to the packet conditioning circuit **70** and the send signal conditioning buffer circuitry **72**. The packet conditioning circuit **70** isolates the source address of the packet and outputs the isolated source address on a line **78** to the comparator **74**. The send signal conditioning buffer circuitry **72** includes the necessary circuitry to receive the packet and transmit the packet as a serial signal that is adapted for transmission via the network when enabled. The comparator **74** also receives on a line **80** the trusted source address value stored in the protected send address enforcement register **68**, and compares the values of the signals on the lines **78** and **80**. If the source address values match, the comparator circuit **74** provides a signal value on a line **82** which enables the send signal conditioning buffer circuit **72** to transmit the packet onto the computer network **18**. If the addresses do not match, the NIC prevents the packet from being transmitted onto the network **18** since the packet contains a forged source address.

The send packet conditioning circuit **70** may include a mask register **84** which masks the bits in the packet not associated with the source address and an offset register **85** which shifts the source address bits, thus allowing a boolean comparison to be made with the trusted source address resident in the protected send address enforcement register **68**. One of ordinary skill will recognize that there a number of different known ways to effect the packet conditioning and comparison functions in order to isolate the source address of the packet for comparison against the trusted source address stored in the protected send address enforcement register.

The NIC **20** also includes a receive address confirmation circuit **88** which includes a protected receive low address enforcement register **90**, a protected receive high address enforcement register **92**, a receive signal conditioning and buffer circuit **94**, a receive packet conditioning circuit **96** and a receive source address comparator **100**. During initialization the protected receive address enforcement registers **90,92** are loaded with a low source address value and a high address value respectively, indicative of a range of valid source addresses associated with known/authorized servers from which the workstation may receive packets. A packet received from the network is input to the receive packet conditioning circuit **96** and the receive signal conditioning

and buffer circuit **94**. The packet conditioning circuit **96** isolates the source address of the received packet and outputs the isolated source address on a line **102** to the comparator **100**. The receive signal conditioning and buffer circuit **94** conditions the serial data stream received from the network and forms a received packet. The comparator **100** also receives the trusted low source address value on a line **104** and the trusted high source address value on a line **106**. The comparator **100** then compares the isolated source address on the line **102** with the trusted low and high source address values on lines **104, 106**, respectively, to ensure that the isolated source address value is between these two trusted address values. If it is, the comparator provides a control signal on a line **107** which enables the receive packet conditioning and buffer circuit **94** to forward the received packet onto the data bus **62**. Otherwise, the received packet is discarded since it was received from a server with which the workstation is not authorized to receive information. The receive packet and conditioning buffer circuit may include a FIFO buffer (not shown) to buffer data for transmission onto the data bus **62**.

FIG. **4** illustrates an alternative embodiment receive address confirmation circuit **108** which takes advantage of the fact that the valid receive source addresses are preferably consecutive within a predefined range. This embodiment is substantially similar to the receive address confirmation circuit **88** (FIG. **3**) with the exception of the separate address enforcement registers. Specifically, the receive address confirmation circuit **108** includes a single receive address enforcement register **109** which contains the significant bits for the range of valid receive source addresses. For example, if the range of valid source addresses is 8FFF000–8FFFFFF (hex), the receive address enforcement register **109** is loaded with the sixteen most significant bits (8FFF) since all the least significant bits are within the range of valid source addresses. Receive packet conditioning circuit **110** conditions the received packet to isolate the most significant bits representative of the range of valid source addresses and feeds the isolated/truncated source address bits into comparator **111**.

Referring to FIG. **3**, the NIC includes a BIOS ROM **65** that contains program instructions which are executed in the CPU **26** (FIG. **2**) during initialization in order to, inter alia, initiate downloading of executable pre-boot software modules resident on the server **12** (FIG. **1**). The NIC BIOS ROM **65** also includes program instructions for locking (i.e., write disabling) the protected address enforcement registers **68, 90, 92** and the mask registers on the NIC. The program instructions for locking the protected address enforcement registers ensure that in the event the pre-boot modules cannot be downloaded (e.g., if there is no physical connection between the network and the workstation), the enforcement registers may still be locked to prevent untrusted software from writing to these registers.

The initialization sequence of the NIC and workstation shall now be discussed.

FIG. **5** is a flowchart illustration of a series of initialization steps **120** performed by the trusted workstation **14** (FIG. **1**) following a hardware reset. The series of steps includes step **122** which initiates execution of the program instructions resident in the system BIOS **32** (FIG. **2**). The program instructions executed in step **122** initialize and take an inventory of the hardware resident on the main system board (i.e., the motherboard) of the workstation and the installed adapters. During this step the system BIOS calls each installed adapter board in the order of its address. The NIC BIOS generally does not have to be the first adapter BIOS called. However, in order to ensure proper system security, the NIC BIOS is preferably called prior to execution of any adapter BIOS that is software modifiable.

Initialization continues with step **124** wherein the CPU executes the program instructions resident in the NIC BIOS **65** (FIG. **3**) to initialize the hardware in the NIC. Following hardware initialization, the CPU downloads the pre-boot modules from the server in step **126** and in step **128**, executes these preboot modules to perform the identification and authorization function associated with the login process described in FIG. **6**. In addition the CPU loads the registers of the NIC's send address confirmation circuitry **66** and the receive address confirmation circuit **88** (FIG. **3**) with values stored in the NIC BIOS ROM. In an alternative embodiment, the pre-boot modules may be stored in the NIC BIOS.

To further enhance the trusted path between the workstation and the server, the pre-boot modules may be "signed". Each pre-boot module includes a different signature which the workstation uses in step **127** to verify that the module is authentic and this prevents unauthorized replacement or modification of the downloaded pre-boot modules. In general, signing of information transmitted over a network is known, see for example the text "Network Security: Private Communication in a Public World", by C. Kaufman, R. Perlman and M. Speciner, published by Prentice Hall PTR, 1995 which is hereby incorporated by reference. To verify the certificate chain and the signature of the "signed" pre-boot modules, a root master public key (i.e., the public key associated with the highest certificate authority) is stored in the NIC BIOS ROM along with the executable code required to perform the verification of step **127**.

FIG. **6** is a flow chart illustration of the login process. In step **140** a username prompt is presented to the user. In response, the user enters a username which is transmitted to the server and in step **142** the server compares the entered username against a list of authorized users. If the username is not valid, network access is denied in step **145** and the login process ends. However, if the entered username is on the list, the server returns an encrypted private key to the workstation in step **143**. The encrypted private key can only be decrypted with the user's password. In step **144** the server checks if any login restrictions, such as, time restrictions, station restrictions and account lock-out restrictions have been violated. These restrictions prevent logins from unauthorized workstations or logins during the wrong time of day. If there are violations access is denied (step **145**). However, if there are no login restrictions, the user is prompted to enter a password in step **146** and the validity of the password is determined in step **148**.

Specifically, step **148** comprises a plurality of operations between the workstation and the server to verify the validity of the password while maintaining the integrity of the password transmitted over the network **18**. Upon entry of the password, the workstation decrypts the encrypted private key with the password entered by the user. The workstation then erases the password from workstation memory to prevent a hacker from obtaining the password. The workstation then maintains an authenticator credential uniquely indicative of the user. The credential includes information identifying the user's complete name, the workstation source address and a validity period (i.e., the duration of time the authenticator is valid). The workstation then creates a signature using the authenticator credential and the decrypted private key. The signature is used for background authentication and to further assist in validating the authenticity of packets transmitted by the workstation onto the

network. In order to complete the authentication process, the workstation creates a proof using the signature, the request for authentication and a random number. The proof is then encrypted by the workstation with the user's private key and transmitted to the server which determines if the proof is valid (this verifies that the password entered by the user was the correct password). If it is, the server transmits a message to the workstation that the user properly logged onto the network and the workstation user is granted conditional access to the network (e.g., to NDS). Note, that with network centric operating system such as Novell's NetWare® 4.1, the user logs into the network rather than into individual servers resident on the network.

If the server determines that the proof is invalid, the server increments an intruder detection counter in step **150**. In step **152** the server compares the value of the counter with a predetermined maximum value to prevent logins by the user (step **156**) if there have been a number of unsuccessful attempts to enter the correct password. If a valid proof is transmitted to the server, network access is granted in step **154** (note the proof will only be valid if the user entered the correct password). In step **156**, the NIC may be disabled to prevent subsequent workstation/server communication, while still allowing the workstation to operate as a public object (i.e., as a stand alone workstation). Alternatively, the workstation may be completely disabled, for example, by not loading the operating system from the server.

The operations associated with authentication are preferably based upon the known Rivest Shamir and Adleman (RSA™) encryption technique and an independent private key algorithm. One key is public (all users of the network have access to it) while the other is kept private (only a designated user knows about it).

Referring again to FIG. **5**, in step **128** the pre-boot modules executed by the workstation also perform the step of loading the NIC registers. This step includes loading the protected send enforcement register **68** (FIG. **3**) with a trusted source address value uniquely associated with the workstation **14**. The protected receive low address enforcement register **90** and protected receive high address enforcement register **92** are also loaded with a low address value and a high address value respectively, indicative of the range of source addresses associated with servers (e.g., NetWare® servers) from which the workstation may receive packets. Preferably, each server (e.g., all NetWare® servers) has a source address within a specified address range, thus allowing the NIC to perform a source address range check to determine if the information is from a server rather than another workstation on the network.

Upon the completion of the pre-boot module execution, the workstation CPU locks (i.e., write disable) the NIC enforcement registers in step **130**. This ensures that the untrusted operating system and application software (executed after initialization) cannot mischiefly alter the contents of the NIC enforcement registers. The NIC enforcement registers can only be unlocked by a hardware reset.

In step **132** CPU execution returns to execute program instructions resident in the system BIOS **32** (FIG. **2**), and in step **134** the workstation operating system software is downloaded from the server to the workstation. The user login and authentication process is then completed in step **136**.

As noted, the NIC can enforce as a source address any address that identifies the source of a packet including the datalink layer or network layer source addresses. With network layer source address enforcement few intermediate components are involved since the network layer source address is left unchanged by retransmission devices such as routers and bridges. However, with datalink layer source address enforcement, at least one of the retransmission devices is needed because the data link layer source address is modified as it traverses many of these devices, including routers. The known servers and/or source address enforcing routers must enforce the association between the data link layer and network layer source addresses when processing packets which are either to be routed or are destined for the known server.

An API is preferably used to load the enforcement registers transparent to the implementation hardware.

Advantageously, the trusted NIC allows what would otherwise be untrusted workstations to participate in an enhanced security network (e.g., a class C2 evaluated network). The trusted NIC provides an inexpensive, hardware based network TCB within an otherwise untrusted workstation thus reducing the cost of a class C2 implementation. Even existing workstations can be allowed to participate in a trusted network by installing the NIC.

Although the present invention has been shown and described with respect to preferred embodiments thereof, it should be understood by those skilled in the art that various other changes, omissions and additions to the form and detail thereof, may be made therein without departing from the spirit and scope of the invention.

What is claimed is:

1. A BIOS computer-readable memory comprising computer-executable program instructions stored therein for being executed by a computer during initialization of the computer following a hardware reset of said computer, said computer-executable program instructions comprising identification and authentication instructions for being executed by said computer for identifying and authenticating a computer user entirely during a pre-boot state of said computer, said pre-boot state taking place after said hardware reset and before booting of any operating system in said computer in response to said hardware reset, said computer-executable program instructions also including verification instructions for being executed by said computer during said pre-boot state for verifying a signature of said identification and authentication instructions to detect whether unauthorized modification has been made to said identification and authentication instructions.

2. A BIOS computer-readable memory according to claim **1**, wherein verification of said signature comprises use of a public key stored in said memory.

3. A BIOS computer-readable memory according to claim **1**, wherein said memory is a read-only memory in said computer.

4. A BIOS computer-readable memory comprising computer-executable program instructions stored therein for being executed by a computer during initialization of the computer following a hardware reset of said computer, said program instructions comprising instructions for verifying, entirely during a pre-boot state of said computer, whether at least one pre-boot module of executable program instructions is authentic and for preventing said at least one pre-boot module from being loaded and executed by said computer during said pre-boot state unless a signature of said pre-boot module is verified during said pre-boot state, said pre-boot state taking place after said hardware reset and before booting of any operating system in said computer in response to said hardware reset.

5. A BIOS computer-readable memory according to claim **4**, wherein verification of said signature comprises use of a public key associated with a certificate authority, said key being stored in said memory.

6. A BIOS computer-readable memory according to claim 4, wherein said memory is a read-only memory in said computer.

7. A BIOS computer-readable memory comprising computer-executable program instructions stored therein for being executed by a computer during initialization of the computer following a hardware reset of said computer, said computer-executable program instructions comprising instructions for:

initiating and controlling downloading of at least one pre-boot module of executable computer program instructions from a server to the computer during a pre-boot state of said computer;

verifying a signature of said at least one pre-boot module whereby to detect whether said at least one pre-boot module has been unauthorizedly modified, during said pre-boot state; and

if said signature of said at least one module is verified, initiating execution of said at least one pre-boot module in said computer during said pre-boot state;

wherein said at least one pre-boot module includes program instructions for identifying and authenticating a user of said computer entirely during said pre-boot state, said pre-boot state taking place after said hardware reset and before booting of any operating system in said computer in response to said hardware reset.

8. A BIOS computer-readable memory according to claim 7, wherein verification of said signature comprises use of a public key stored in said memory.

9. A BIOS computer-readable memory according to claim 7, wherein said memory is read-only memory.

10. A BIOS computer-readable memory comprising computer-executable program instructions stored therein for being executed by a computer during initialization of the computer following a hardware reset of said computer, said program instructions comprising instructions for being executed during a pre-boot state of said computer to verify a signature of a pre-boot module of executable instructions to determine authenticity of said pre-boot module of executable instructions, said pre-boot module of executable instructions being for preventing the computer from booting any operating system unless a user of said computer is successfully authenticated during said pre-boot state.

11. A BIOS computer-readable memory according to claim 10, wherein verification of said signature comprises use of a master key stored in said memory.

12. A BIOS computer-readable memory according to claim 10, wherein said memory is a read-only memory.

13. A BIOS computer-readable memory comprising computer-executable program instructions stored therein for being executed by a computer during initialization of the computer following a hardware reset of said computer, said program instructions comprising instructions for being executed by said computer entirely during a pre-boot state of said computer for validating a signature of a pre-boot module of instructions whereby to determine authenticity of said module of instructions, said module of instructions being for preventing the computer from communicating with a server unless a user of said computer is successfully authenticated during said pre-boot state, said pre-boot state taking place after said hardware reset and prior to booting of any operating system in said computer in response to said hardware reset.

14. A BIOS computer-readable memory according to claim 13, wherein validation of said signature comprises use of a root master key associated with a certificate authority, said key being stored in said memory.

15. A BIOS computer-readable memory according to claim 14, wherein said memory is a read-only memory in said computer.

* * * * *