

UNITED STATES PATENT AND TRADEMARK OFFICE

---

BEFORE THE PATENT TRIAL AND APPEAL BOARD

---

APPLE INC.  
Petitioner,

v.

VIRNETX, INC. AND SCIENCE APPLICATIONS INTERNATIONAL  
CORPORATION,  
Patent Owners

Patent No. 8,051,181

Issued: November 1, 2011

Filed: February 27, 2007

Inventors: Victor Larson, *et al.*

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

---

*Inter Partes* Review No. IPR2014-00485 and IPR2014-00486

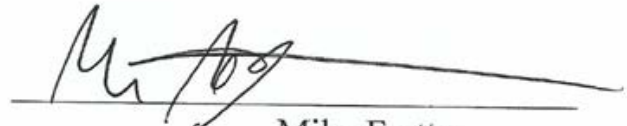
---

**Declaration of Michael Fratto Regarding**

**U.S. Patent No. 8,051,181**

I, Michael Fratto, do hereby declare and state, that all statements made herein of my 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.

Dated: March 9, 2014



Mike Fratto

## TABLE OF CONTENTS

|      |  |    |
|------|--|----|
| I.   | INTRODUCTION .....   | 1  |
| A.   | Engagement .....   | 1  |
| B.   | Background and Qualifications .....  | 1  |
| C.   | Compensation and Prior Testimony .....                                       | 4  |
| D.   | Information Considered.....  | 5  |
| II.  | LEGAL STANDARDS FOR PATENTABILITY .....                                      | 6  |
| A.   | Anticipation .....   | 7  |
| B.   | Obviousness.....   | 8  |
| III. | THE '181 PATENT .....  | 14 |
| A.   | Effective Filing Date of the '181 Patent .....                               | 14 |
| B.   | Prosecution History of the '181 Patent .....                                 | 16 |
| C.   | The Person of Ordinary Skill in the Art .....                                | 16 |
| D.   | Overview of the Claims of the '181 Patent.....                               | 17 |
| E.   | Description of the Background Technology Relevant to the<br>'181 Patent..... | 17 |
| 1.   | The Open Internet .....  | 17 |
| 2.   | Domain Names and the Domain Name System (DNS).....                           | 19 |
| a.   | The DNS System .....   | 19 |
| b.   | Domain Names and URIs .....  | 22 |
| c.   | Domain Name Resolution .....   | 24 |
| d.   | DNS SRV Records .....  | 32 |
| 3.   | Communications and Name Resolution on Windows-<br>Based Networks .....       | 32 |
| 4.   | Encryption, Tunnels, and Data Security .....                                 | 40 |
| a.   | Encryption.....  | 41 |
| b.   | Authentication.....  | 43 |
| c.   | IPsec – RFC 2401 .....   | 44 |
| d.   | Tunneling.....   | 45 |
| 5.   | VPNs .....   | 48 |

|     |   |     |
|-----|---|-----|
| 6.  | Multimedia Transmission over the Internet.....  | 51  |
| F.  | Common Design Perspectives in April 2000.....   | 54  |
| IV. | GENERAL ISSUES RELATED TO MY PATENTABILITY ANALYSIS.....  | 56  |
| A.  | The Claims of the '181 Patent I Am Addressing in this Report .....  | 57  |
| B.  | Comparison with Patent Family Members.....  | 65  |
| C.  | Interpretation of Certain Claim Terms .....   | 67  |
| 1.  | Secure Name .....   | 68  |
| 2.  | Secure Name Service .....   | 71  |
| 3.  | Secure Domain Name .....  | 78  |
| 4.  | Unsecured Name .....  | 81  |
| 5.  | Secure Communication Link (Claims 1-23, 25, and 28)<br>and Securely Communicate (Claims 24, 27, and 29) .....               | 82  |
| D.  | Prior Art References.....   | 86  |
| 1.  | Exhibit 1031 – U.S. Patent 6,496,867 to Beser .....   | 86  |
| 2.  | Exhibit 1003 – U.S. Patent 6,557,037 to <u>Provino</u> .....  | 87  |
| 3.  | Exhibit 1004 – Kiuchi et al., “C-HTTP - The<br>Development of a Secure, Closed HTTP-based Network<br>on the Internet” ..... | 87  |
| 4.  | Request for Comment (RFC) Publications .....  | 87  |
| V.  | PATENTABILITY ANALYSIS OF CLAIMS 1-29 OF THE '181<br>PATENT .....   | 90  |
| A.  | U.S. Patent No. 6,496,867 to Beser.....   | 90  |
| 1.  | Overview of Beser .....   | 90  |
| 2.  | Overview of RFC 2401 .....  | 122 |
| 3.  | Comparison of Beser to Claims 1-29 of the '181 Patent.....  | 133 |
| a.  | Claim 1.....  | 133 |
| b.  | Claim 2.....  | 141 |
| c.  | Claim 3.....  | 149 |
| d.  | Claim 4.....  | 151 |

|    |  |     |
|----|--|-----|
| e. | Claim 5.....   | 152 |
| f. | Claim 6.....   | 153 |
| g. | Claim 7.....   | 153 |
| h. | Claim 8.....   | 154 |
| i. | Claim 9.....   | 155 |
| j. | Claims 10-11.....  | 156 |
| k. | Claim 12.....  | 156 |
| l. | Claim 13.....  | 157 |
| m. | Claims 14-17.....  | 158 |
| n. | Claim 18.....  | 160 |
| o. | Claim 19.....  | 160 |
| p. | Claim 20.....  | 161 |
| q. | Claim 21.....  | 161 |
| r. | Claim 22.....  | 163 |
| s. | Claim 23.....  | 166 |
| t. | Claim 24.....  | 166 |
| u. | Claim 25.....  | 175 |
| v. | Claim 26.....  | 177 |
| w. | Claim 27.....  | 186 |
| x. | Claim 28.....  | 189 |
| y. | Claim 29.....  | 196 |
| B. | RFC 2543 – Session Initiation Protocol.....                      | 202 |
| 1. | Overview of RFC 2543.....  | 202 |
| 2. | Comparison of RFC 2543 to Claims 1-29 of the '181<br>Patent..... | 234 |
| a. | Claim 1.....   | 234 |
| b. | Claim 2.....   | 242 |
| c. | Claims 3-4.....  | 250 |
| d. | Claim 5.....   | 252 |

|    |   |     |
|----|---|-----|
| e. | Claim 6.....  | 252 |
| f. | Claim 7.....  | 253 |
| g. | Claim 8.....  | 254 |
| h. | Claim 9.....  | 255 |
| i. | Claim 10.....   | 256 |
| j. | Claim 11.....   | 256 |
| k. | Claim 12.....   | 257 |
| l. | Claim 13.....   | 257 |
| m. | Claims 14-17.....   | 258 |
| n. | Claim 18.....   | 259 |
| o. | Claims 19-20.....   | 259 |
| p. | Claim 21.....   | 260 |
| q. | Claim 22.....   | 261 |
| r. | Claim 23.....   | 262 |
| s. | Claim 24.....   | 263 |
| t. | Claim 25.....   | 269 |
| u. | Claim 26.....   | 270 |
| v. | Claim 27.....   | 278 |
| w. | Claim 28.....   | 279 |
| x. | Claim 29.....   | 287 |
| C. | Kiuchi et al., “C-HTTP - The Development of a Secure, Closed<br>HTTP-based Network on the Internet” ..... | 292 |
| 1. | Overview of Kiuchi.....   | 292 |
| 2. | Comparison of Kiuchi to Claims 1-6, 8-9 and 13-19, 21-<br>29 of the ’181 Patent .....                     | 303 |
| a. | Claim 1.....  | 303 |
| b. | Claim 2.....  | 306 |
| c. | Claim 3.....  | 310 |
| d. | Claim 4.....  | 310 |
| e. | Claim 5.....  | 311 |

|    |   |     |
|----|---|-----|
| f. | Claim 6.....  | 312 |
| g. | Claim 8.....  | 312 |
| h. | Claim 9.....  | 312 |
| i. | Claim 13.....   | 313 |
| j. | Claims 14-17.....   | 313 |
| k. | Claim 18.....   | 315 |
| l. | Claim 19.....   | 315 |
| m. | Claim 21.....   | 316 |
| n. | Claim 22.....   | 316 |
| o. | Claim 23.....   | 317 |
| p. | Claim 24.....   | 317 |
| q. | Claim 25.....   | 320 |
| r. | Claim 26.....   | 321 |
| s. | Claim 27.....   | 325 |
| t. | Claim 28.....   | 326 |
| u. | Claim 29.....   | 330 |
| D. | U.S. Patent No. 6,557,037 to Provino .....                  | 333 |
| 1. | Overview of Provino .....                                   | 334 |
| 2. | Overview of Kosiur.....                                     | 355 |
| 3. | Comparison of Provino to Claims 1-29 of the '181 Patent ... | 358 |
| a. | Claim 1.....  | 358 |
| b. | Claim 2.....  | 362 |
| c. | Claim 3.....  | 366 |
| d. | Claim 4.....  | 367 |
| e. | Claim 5.....  | 368 |
| f. | Claim 6.....  | 368 |
| g. | Claim 7.....  | 369 |
| h. | Claim 8.....  | 370 |
| i. | Claim 9.....  | 371 |

|     |  |     |
|-----|--|-----|
| j.  | Claim 10.....  | 372 |
| k.  | Claim 11.....  | 373 |
| l.  | Claim 12.....  | 374 |
| m.  | Claim 13.....  | 375 |
| n.  | Claim 14.....  | 376 |
| o.  | Claim 15.....  | 377 |
| p.  | Claim 16.....  | 378 |
| q.  | Claim 17.....  | 379 |
| r.  | Claim 18.....  | 380 |
| s.  | Claim 19.....  | 380 |
| t.  | Claim 20.....  | 381 |
| u.  | Claim 21.....  | 381 |
| v.  | Claim 22.....  | 382 |
| w.  | Claim 23.....  | 385 |
| x.  | Claim 24.....  | 385 |
| y.  | Claim 25.....  | 392 |
| z.  | Claim 26.....  | 395 |
| aa. | Claim 27.....  | 403 |
| bb. | Claim 28.....  | 408 |
| cc. | Claim 29.....  | 412 |
| E.  | Provino Renders Obvious Claims 21, 26, and 27 .....          | 416 |
| F.  | Provino in View of Beser Renders Obvious Claims 12-17 .....  | 418 |
| a.  | Claim 12.....  | 418 |
| b.  | Claim 13.....  | 420 |
| c.  | Claim 14.....  | 421 |
| d.  | Claim 15.....  | 422 |
| e.  | Claim 16.....  | 423 |
| f.  | Claim 17.....  | 424 |
| G.  | Provino In View of Kosiur Renders Obvious Claims 12-17 ..... | 425 |



|      |  |     |
|------|--|-----|
| a.   | Claim 12.....  | 425 |
| b.   | Claim 13.....  | 427 |
| c.   | Claim 14.....  | 428 |
| d.   | Claim 15.....  | 429 |
| e.   | Claim 16.....  | 430 |
| f.   | Claim 17.....  | 432 |
| VI.  | CONCLUSION.....                                      | 433 |
| VII. | APPENDIX A: MATERIALS CONSIDERED BY MIKE FRATTO..... | 434 |

## TABLE OF APPENDICES

Appendix A: List of Materials Considered

## **I. INTRODUCTION**

### **A. Engagement**

1. I have been retained by counsel for Apple Inc. as an expert witness in the above-captioned proceeding. I have been asked to provide my opinion about the state of the art of the technology described in U.S. Patent No. 8,051,181 (“the ’181 patent”) and on the patentability of claims 1-29 of the ’181 patent. The following is my written report on these topics.

### **B. Background and Qualifications**

2. My Curriculum Vitae is submitted herewith as Exhibit 1004.
3. I received a Bachelor’s of Science Degree from Syracuse University in 2001 in Information Science and Technology.
4. I began working in the field of computer science in 1987. Between 1987 and 1992, I was a consultant where I wrote software that would connect remote offices and collect/process data for input into other programs.
5. Between 1994 and 1997, I was a freelance editor for Network Computing, where I covered remote access and telecommunications products and technologies. I held this position while earning my B.S. in Information Science and Technology from Syracuse University.
6. Between June 1997 and June of 2004, I was a permanent employee of Network Computing, starting as an Associate Technology Editor and rising to Senior Technology Editor. In these capacities, I focused on network security

solutions and products, which was a rapidly evolving field at that time. I have subsequently held a number of other positions with Network Computing, as well as affiliated publications. For example, from July 2004 to April 2006, I served as Editor of Secure Enterprise Magazine – which focused on security solutions specific to enterprise systems. After April 2006, I returned to Network Computing, where I served in a progression of senior positions, ultimately serving as Editor of Network Computing between August 2009 and October of 2012.

7. I am presently a Senior Analyst with Current Analysis. In this respect I manage the Enterprise Networking practice within the business and technology and software group. I also provide competitive analysis on trends and changes in the enterprise networking markets as well as consult with network equipment vendors on product messaging, marketing, and outreach.

8. I presently serve as an adjunct faculty member of the School of Information Studies at Syracuse University.

9. I have been studying, evaluating, testing and describing networking, networking security and related technologies for more than 15 years. Since well before 1999, I have had an extensive background and experience in network systems, software and related technologies, with a particular focus on network security.

10. I also have extensive hands-on experience with wide range of networking and networking security products developed and sold in the 1993 to 2002 time frame. This came from my various positions with Network Computing, where I reviewed, tested and described these products in a technical publication devoted to this field. I also wrote articles about network infrastructure, data center, and network access control items that were published by Network Computing. I also am very familiar with Internet standards governing networking and security, which I discuss below.

11. A substantial amount of my extensive experience with networking and security systems used in the late 1990s came from my work in reviewing these products for Network Computing. When I performed a typical review of a new product, I would first define a “problem set” the product was designed to address. I would interview IT administrators and executives and speak with the developers of the product. I also would study the standards relevant to the product or its implementation, and would evaluate the technical documentation accompanying the product. I would then create a set of comparative measures to assess the ability of the product to execute. I would then test the product under a variety of situations designed to evaluate these comparative measures. To do this, I would set up a test network, verify its operation, conduct the tests, and ensure the results

were accurate. I would also often compare the product to other products in the same class or category.

12. In the 1997 to 2000 time frame, I was particularly focused on remote access products including modems, ISDN, and virtual private networking products, technologies, and standards as well as network and host-based firewalls.

13. I am the author of several books and publications devoted to secure networking technologies. I also have taught a number of courses on systems and technologies used in networking security, particularly focused on Internet security solutions. These books and courses are listed on my C.V. (Ex. 1030).

### **C. Compensation and Prior Testimony**

14. I am being compensated at a rate of \$250 per hour for my study and testimony in this matter. I am also being reimbursed for reasonable and customary expenses associated with my work and testimony in this investigation. My compensation is not contingent on the outcome of this matter or the specifics of my testimony.

15. I have never testified in Federal District Court.

16. I have provided declarations that were submitted in inter partes reexamination proceedings based on patents owned by Patent Owner. These proceedings include Control No. 95/001,682, 95/001,679, 95/001,788 and 95/001,789.

17. I have provided declarations that were submitted in *inter partes* review proceedings based on patents owned by Patent Owner. These proceedings include IPR2013-00348, -00349, -00354, -00393, -00394, -00397, and -00398 filed by Petitioner Apple, IPR2014-00237 and -00238 filed by Petitioner Apple, and IPR2014-00171, -00172, -00172, -00174, -00175, -00176, and -00177 filed by another petitioner.

**D. Information Considered**

18. My opinions are based on my years of education, research and experience, as well as my investigation and study of relevant materials. In forming my opinions, I have considered the materials I identify in this report and those listed in Appendix A.

19. I may rely upon these materials and/or additional materials to respond to arguments raised by the Patent Owner. I may also consider additional documents and information in forming any necessary opinions — including documents that may not yet have been provided to me.

20. My analysis of the materials produced in this investigation is ongoing and I will continue to review any new material as it is provided. This report represents only those opinions I have formed to date. I reserve the right to revise, supplement, and/or amend my opinions stated herein based on new information and on my continuing analysis of the materials already provided.

## **II. LEGAL STANDARDS FOR PATENTABILITY**

21. In expressing my opinions and considering the subject matter of the claims of the '181 patent, I am relying upon certain basic legal principles that counsel has explained to me.

22. First, I understand that for an invention claimed in a patent to be found patentable, it must be, among other things, new and not obvious from what was known before the invention was made.

23. I understand the information that is used to evaluate whether an invention is new and not obvious is generally referred to as “prior art” and generally includes patents and printed publications (e.g., books, journal publications, articles on websites, product manuals, etc.).

24. I understand in this proceeding Apple has the burden of proving that the '181 patent are anticipated by or obvious from the prior art by a preponderance of the evidence. I understand that “a preponderance of the evidence” is evidence sufficient to show that a fact is more likely true than it is not.

25. I understand that in this proceeding, the claims must be given their broadest reasonable interpretation consistent with the specification. The claims after being construed in this manner are then to be compared to the information in the prior art.



26. I understand that in this proceeding, the information that may be evaluated is limited to patents and printed publications. My analysis below compares the claims to patents and printed publications that are prior art to the claims.

27. I understand that there are two ways in which prior art may render a patent claim unpatentable. First, the prior art can be shown to “anticipate” the claim. Second, the prior art can be shown to have made the claim “obvious” to a person of ordinary skill in the art. My understanding of the two legal standards is set forth below.

**A. Anticipation**

28. I understand that the following standards govern the determination of whether a patent claim is “anticipated” by the prior art.

29. I have applied these standards in my evaluation of whether claims 1-29 of the ’181 patent would have been anticipated by the prior art.

30. I understand that the “prior art” includes patents and printed publications that existed before the earliest filing date (the “effective filing date”) of the claim in the patent. I also understand that a patent will be prior art if it was filed before the effective filing date of the claimed invention, while a printed publication will be prior art if it was publicly available before that date.

31. I understand that, for a patent claim to be “anticipated” by the prior art, each and every requirement of the claim must be found, expressly or inherently, in a single prior art reference as recited in the claim. I understand that claim limitations that are not expressly described in a prior art reference may still be there if they are “inherent” to the thing or process being described in the prior art. For example, an indication in a prior art reference that a particular process complies with a published standard would indicate that the process must inherently perform certain steps or use certain data structures that are necessary to comply with the published standard.

32. I understand that it is acceptable to consider evidence other the information in a particular prior art document to determine if a feature is necessarily present in or inherently described by that reference.

**B. Obviousness**

33. I understand that a claimed invention is not patentable if it would have been obvious to a person of ordinary skill in the field of the invention at the time the invention was made.

34. I understand that the obviousness standard is defined in the patent statute (35 U.S.C. § 103(a)) as follows:

A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the

prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

35. I understand that the following standards govern the determination of whether a claim in a patent is obvious. I have applied these standards in my evaluation of whether claims 1-29 of the '181 patent would have been considered obvious in April of 2000.

36. I understand that to find a claim in a patent obvious, one must make certain findings regarding the claimed invention and the prior art. Specifically, I understand that the obviousness question requires consideration of four factors (although not necessarily in the following order):

- The scope and content of the prior art;
- The differences between the prior art and the claims at issue;
- The knowledge of a person of ordinary skill in the pertinent art; and
- Whatever objective factors indicating obviousness or non-obviousness may be present in any particular case.

37. In addition, I understand that the obviousness inquiry should not be done in hindsight, but must be done using the perspective of a person of ordinary skill in the relevant art as of the effective filing date of the patent claim.

38. I understand the objective factors indicating obviousness or non-obviousness may include: commercial success of products covered by the patent claims; a long-felt need for the invention; failed attempts by others to make the invention; copying of the invention by others in the field; unexpected results achieved by the invention; praise of the invention by the infringer or others in the field; the taking of licenses under the patent by others; expressions of surprise by experts and those skilled in the art at the making of the invention; and the patentee proceeded contrary to the accepted wisdom of the prior art.

39. I understand the combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results. I also understand that an example of a solution in one field of endeavor may make that solution obvious in another related field. I also understand that market demands or design considerations may prompt variations of a prior art system or process, either in the same field or a different one, and that these variations will ordinarily be considered obvious variations of what has been described in the prior art.

40. I also understand that if a person of ordinary skill can implement a predictable variation, that variation would have been considered obvious. I understand that for similar reasons, if a technique has been used to improve one device, and a person of ordinary skill in the art would recognize that it would

improve similar devices in the same way, using that technique to improve the other device would have been obvious unless its actual application yields unexpected results or challenges in implementation.

41. I understand that the obviousness analysis need not seek out precise teachings directed to the specific subject matter of the challenged claim, but instead can take account of the “ordinary innovation” and experimentation that does no more than yield predictable results, which are inferences and creative steps that a person of ordinary skill in the art would employ.

42. I understand that sometimes it will be necessary to look to interrelated teachings of multiple patents; the effects of demands known to the design community or present in the marketplace; and the background knowledge possessed by a person having ordinary skill in the art. I understand that all these issues may be considered to determine whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue.

43. I understand that the obviousness analysis cannot be confined by a formalistic conception of the words “teaching, suggestion, and motivation.” I understand that in 2007, the Supreme Court issued its decision in *KSR Int’l Co. v. Teleflex, Inc.* where the Court rejected the previous requirement of a “teaching, suggestion, or motivation to combine” known elements of prior art for purposes of an obviousness analysis as a precondition for finding obviousness. It is my

understanding that *KSR* confirms that any motivation that would have been known to a person of skill in the art, including common sense, or derived from the nature of the problem to be solved, is sufficient to explain why references would have been combined.

44. I understand that a person of ordinary skill attempting to solve a problem will not be led only to those elements of prior art designed to solve the same problem. I understand that under the *KSR* standard, steps suggested by common sense are important and should be considered. Common sense teaches that familiar items may have obvious uses beyond the particular application being described in a reference, that if something can be done once it is obvious to do it multiple times, and in many cases a person of ordinary skill will be able to fit the teachings of multiple patents together like pieces of a puzzle. As such, the prior art considered can be directed to any need or problem known in the field of endeavor in April of 2000 and can provide a reason for combining the elements of the prior art in the manner claimed. In other words, the prior art does not need to be directed towards solving the same problem that is addressed in the patent. Further, the individual prior art references themselves need not all be directed towards solving the same problem.

45. I understand that an invention that might be considered an obvious variation or modification of the prior art may be considered non-obvious if one or

more prior art references discourages or lead away from the line of inquiry disclosed in the reference(s). A reference does not “teach away” from an invention simply because the reference suggests that another embodiment of the invention is better or preferred. My understanding of the doctrine of teaching away requires a clear indication that the combination should not be attempted (*e.g.*, because it would not work or explicit statements saying the combination should not be made.

46. I understand that a person of ordinary skill is also a person of ordinary creativity.

47. I further understand that in many fields, it may be that there is little discussion of obvious techniques or combination, and it often may be the case that market demand, rather than scientific literature or knowledge, will drive design trends. When there is such a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions, a person of ordinary skill has good reason to pursue the known options within their technical grasp. If this leads to the anticipated success, it is likely the product not of innovation but of ordinary skill and common sense. In that instance the fact that a combination was obvious to try might show that it was obvious. The fact that a particular combination of prior art elements was “obvious to try” may indicate that the combination was obvious even if no one attempted the combination. If the combination was obvious to try (regardless of whether it was actually tried) or

leads to anticipated success, then it is likely the result of ordinary skill and common sense rather than innovation.

### **III. THE '181 PATENT**

#### **A. Effective Filing Date of the '181 Patent**

48. I understand that the '181 patent issued from U.S. Application No. 11/679,416, filed February 27, 2007.

49. I understand that the '416 application is a “continuation” of U.S. Application No. 10/702,486, filed November 7, 2003. I understand that the '486 application became U.S. Patent No. 7,188,180.

50. I understand that the '486 application is a “division” of U.S. Application No. 09/558,209, filed April 26, 2000, now abandoned.

51. I understand that the '209 application is a “continuation-in-part” of U.S. Application No. 09/504,783, filed February 15, 2000. I understand the '783 application became U.S. Patent No. 6,502,135. I understand that the '783 application is a “continuation-in-part” of U.S. Application No. 09/429,643, filed on October 29, 1999. The '783 and '643 applications each claim priority under 35 U.S.C. 119(e) to Provisional Application Nos. 60/106,261, filed October 30, 1998 and 60/137,704, filed June 7, 1999.



52. I understand that a “continuation-in-part” application is a patent application that is related to an earlier filed patent application, but which adds to, changes or deletes information relative to what was in the previous application.

53. I understand that a “continuation-in-part” application may describe an invention that is not described in the earlier application, and that this may result in a patent claim being evaluated using the filing date of the continuation-in-part application, rather than the earlier related application.

54. I understand that claims 1, 2, 24, 26, 28, and 29 of the ’181 patent are independent claims. I also understand that claims 3-23 depend from claim 2, claim 25 depends from claim 24, and claim 27 depends from claim 26.

55. I understand that a dependent claim incorporates all of the requirements of the independent claim from which it depends. So, in the case of the ’181 patent, claims 2-23, claim 25, and claim 27 cannot enjoy an effective filing date earlier than that of claims 2, 24, and 26, respectively, from which they depend.

56. I understand that during an *inter partes* reexamination of the ’180 patent, Control No. 95/001678, VirnetX acknowledged that the priority date of the ’180 patent was April 26, 2000. Ex. 1023 at 232 (Further, the closeness of proximity of the alleged publication date of Aventail to the April 26, 2000 priority

date of the '180 Patent raises further doubt as to the availability of the Aventail as prior art.”). The '180 patent was issued from a parent of the '181 patent.

57. I have therefore used April 26, 2000, as the earliest effective filing date of the '181 patent claims in my analysis.

**B. Prosecution History of the '181 Patent**

58. The application from which the '181 patent issued was filed on February 27, 2007 as Application No. 11/679,416 and listed Victor Larson, among others, as an inventor.

**C. The Person of Ordinary Skill in the Art**

59. I believe a person of ordinary skill in the art in the field of the '181 patent would be someone who had a good working knowledge of networking protocols, including those employing security techniques, as well as computer systems that support these protocols and techniques. The person would have gained this knowledge either through several years of practical working experience or through education and training, or through a combination of these. The person also would be very familiar with Internet standards related to communications and security, with software development techniques, and with a variety of client-server systems and technologies.

#### **D. Overview of the Claims of the '181 Patent**

60. The '181 patent claims generally concern processes and systems for a Domain Name Service ("DNS") or name service used to establish a secure communication link between devices.

#### **E. Description of the Background Technology Relevant to the '181 Patent**

61. In April of 2000, all the technologies addressed in the '181 patents were well known to those in the art. I will address some of the more pertinent technologies below.

##### **1. The Open Internet**

62. The Internet is a series of public, interconnected networks that connects millions of computers. The Internet functions by routing communications and data between an initiating computer and a destination computer. *See* Ex. 1064 (Hoke) at 1:53-59.

63. Both in April of 2000 and now, the predominant standard that governs how information is formatted and transmitted over the Internet is the Transmission Control Protocol / Internet Protocol (TCP/IP). *See generally* Ex. 1047 (RFC 1180) at 9-16.

64. Under the TCP/IP protocol, computers on the Internet have a unique identifier, referred to as an Internet Protocol (IP) address, that consists of a string of numbers (*e.g.*, "199.181.132.250"). This unique IP address provides an

identifier that allows any given computer to know which computer connected to the Internet to send information to. Ex. 1047 (RFC 1180) at 9-11.

65. Information is transmitted over the Internet in the form of datagrams commonly called “packets.” Packets are bundles of information arranged in a defined way. Ex. 1020 (RFC 791) at 7, 17-22; Ex. 1047 (RFC 1180) at 3-5.

66. Every packet contains “routing” information including the IP addresses of the originating and destination computers. *See* Ex. 1020 (RFC 791) at 17-22.

67. Packets can be sent via intermediary computers (called routers) to the destination specified in each packet. Ex. 1047 (RFC 1180) at 3-5; Ex. 1039 (NT for Dummies) at 10.

68. Routers can automatically forward packets using the routing information contained in the packet. This “automatic forwarding” concept is one of the main reasons why the Internet is considered an “open” network. Ex. 1039 (NT for Dummies) at 10.

69. Typically, on each local network or subnet, one router will act as a gateway. Ex. 1039 (NT for Dummies) at 10. A gateway router is able to send packets from the local network to the public network. *Id.*; Ex. 1020 (RFC 791) at 15; Ex. 1040 (Win98 Resource Kit) at 37-39.

70. A “firewall” and a “proxy server” are essentially specialized routers that implement additional functionality, such as packet filtering and security services. Ex. 1039 (NT for Dummies) at 13; Ex. 1040 (Win98 Resource Kit) at 37-39.

71. The open nature of the Internet can be a problem for users seeking privacy and security. Because messages are passed through many computers, it is possible for messages sent over the Internet to be intercepted and read or altered by intermediaries. *E.g.*, Ex. 1045 (RFC 1123) at 5; Ex. 1047 (RFC 1180) at 15; Ex. 1064 (Hoke) at 2:6-33, 46-55. As will be discussed more below, several technologies have been created to provide data security.

## **2. Domain Names and the Domain Name System (DNS)**

72. My comments below describe the DNS system and protocols as they existed in April 2000. The DNS systems used today largely follow the same structure and conventions, particularly for communications using the IPv4 protocol that was in place in April of 2000.

### **a. The DNS System**

73. The domain name system (DNS) correlates domain names with IP addresses of computers on the Internet. *See generally* Ex. 1036 (RFC 1034).

74. The domain name system allows a user to type in an easily understandable phrase (*e.g.*, “www.apple.com”), which will then be parsed and

translated into the IP address (“12.34.56.78”) of the computers hosting that website (*i.e.*, Apple’s Website). Ex. 1047 (RFC 1180) at 11-12; Ex. 1037 (RFC 1035) at 2.

75. DNS was a critically important step in the development of the web because it enabled users to navigate the web using understandable words, rather than complex strings of numbers. *See* Ex. 1039 (NT for Dummies) at 6.

76. Like the routing system of the Internet, the DNS system also has an open and “automatic” design. Specifically, DNS servers perform “name resolution” – converting a human understandable domain name to a corresponding Internet address – using a standardized process which is summarized in two “Requests for Comment” documents; namely, RFC 1034 (Ex. 1036) and RFC 1035 (Ex. 1037).

77. The name resolution process is a relatively simple process. Generally, a domain name requiring name resolution is sent to a DNS server which, in a hierarchical manner, will search a sequence of databases to find the IP address for that domain name, and will then return the numerical address corresponding to that domain name to the requesting computer. Ex. 1036 (RFC 1034) at 4-5. The databases store data in a form that correlates IP addresses to domain names. *Id.*

78. The Internet Corporation for Assigned Names and Numbers (ICANN) is responsible for the registration and management of Internet domain names. Domain names that require resolution by public Internet users must be registered

and the resource records associated with such domain names must be made publicly available on a registered authoritative DNS server. *See* Ex. 1047 (RFC 1180) at 11; Ex. 1039 (NT for Dummies) at 6-7; Ex. 1036 (RFC 1034) at 18-21, 27-28 (describing adding resource records to a name server); Ex. 1065 (RFC 2136) at 1, 6-10, 21 (describing a mechanism for adding, modifying, deleting records from a name server). People working in this field use the terms “register” or “registration” to refer to the process of storing data representing an association between a network address (e.g., an IP address) and name (e.g., a domain name) that can be used to find the computer at that network address. So, for example, a domain name will be “registered” by providing data correlating an IP address with the domain name, which a DNS server will later use to resolve domain name resolution requests.

79. For a domain name to be processed by conventional domain name system servers, the top level domain name (e.g., .com, .edu, .gov, and .org) must be registered with ICANN. *See* Ex. 1039 (NT for Dummies) at 6-7.

80. Not all DNS servers are publicly available. Ex. 1039 (NT for Dummies) at 7. DNS servers may be implemented on private networks and intranets with access limited to only authorized users of the network. *See* Ex. 1037 (RFC 1035) at 6.

81. Domain names registered with private DNS servers do not need to conform to ICANN's standards for domain names. Ex. 1037 (RFC 1035) at 6; Ex. 1039 (NT for Dummies) at 7; Ex. 1036 (RFC 1034) at 18-21, 27-28.

**b. Domain Names and URIs**

82. Most Internet users are familiar with domain names as they are used in Uniform Resource Locators or "URLs," which are a type of Uniform Resource Identifier (URI). See Ex. 1046 (RFC 2068) at 13.

83. Every URL has essentially three parts: the transfer format, the domain name or host name of the server where the resource resides, and the path to the specific resource. An example of a URL is shown below:

|                              |                          |  |
|------------------------------|--------------------------|--|
| <u>http://</u>               | <u>images.apple.com/</u> | <u>iphone.jpg</u>                        |
| Transfer<br>format<br>(http) | Domain name              | Resource ( <i>e.g.</i> , a picture file) |

84. A conventional domain name typically has two parts: (1) a network domain name; and (2) a host name. The two components are joined by a period separator. In the above example, "apple.com" is the network domain name and "images" is the host name. Although this example shows a single network domain field, the network domain name field may be further divided into sub domain names by means of additional periods (*e.g.*, "iphone.images.apple.com"). Ex. 1036 (RFC 1034) at 6-10; Ex. 1037 (RFC 1035) at 6-7; Ex. 1039 (NT for Dummies) at 6.



85. The traditional concept of a Domain Name centers on the definition of a hierarchical name space, which uses a “dot” convention to separate fields representing different levels in the hierarchy. Ex. 1036 (RFC 1034) at 1-3.

86. A standard domain name can be any string that starts with a letter, ends with a letter or digit, and has letters, digits, or hyphens as internal characters. Segments of the domain name can be divided into fields or file locations by the DNS system (*i.e.*, the characters “.” and “/”). Ex. 1036 (RFC 1034) at 6-10; Ex. 1037 (RFC 1035) at 6-7.

87. The fields of a domain name are resolved from right to left. Thus, the top level domain (*e.g.*, “.com”) is represented by the text field at the right hand end of the domain name. Ex. 1036 (RFC 1034) at 5-6; Ex. 1037 (RFC 1035) at 6-7.

88. The essential characteristic of a domain name is that it enables a user to use the DNS system to identify an IP address corresponding to a “host name” (*e.g.*, 192.168.1.101) or an IP network number corresponding to a network domain assignment of a block of IP addresses (*e.g.*, 192.168.1.0). Ex. 1036 (RFC 1034) at 20-21, 28-29; Ex. 1045 (RFC 1123) at 50, 56.

89. The DNS system also can be used to resolve URIs that contain domain names. For example, URIs containing domain names are used as identification strings for mail resources, such as bob@smtp.roadrunner.com. Ex. 1036 (RFC 1034) at 6-8.

90. URIs also occur in Internet telephony systems as identifiers of subscribers, for example as sip:alice@atlanta.example.com. Ex. 1036 (RFC 1034) at 6-8; Ex. 1045 (RFC 1123) at 33; *see* Ex. 1031 (Beser) at 10:55-67; Ex. 1033 (RFC 2543) at 20-24.

**c. Domain Name Resolution**

91. The public DNS system is a distributed database. No single server contains an entry for every domain name. Instead, domain names are divided up hierarchically into various zones, with each zone having a DNS server or servers responsible for resolving addresses within the zone. Ex. 1036 (RFC 1034) at 6-9, 13; Ex. 1037 (RFC 1035) at 2-3; Ex. 1045 (RFC 1123) at 50, 56-59.

92. The ICANN top level domains, such as the .com domain, are root zones. Each root zone is divided into sub-zones, and each sub-zone typically is further divided into additional zones. Ex. 1036 (RFC 1034) at 6-9.

93. Each zone may have its own DNS server that maps domain names to the DNS server of the appropriate the sub-zone, or if there are no sub-zones, it maps the names to the IP address of the host computer or to another host name. Ex. 1036 (RFC 1034) at 17-21, 27-28.

94. A DNS server within a zone is an authoritative server only for domain names contained within that zone. If it receives a query for a domain name in a

sub-zone, it will forward the requester to the DNS server for the sub-zone. Ex. 1036 (RFC 1034) at 17-21.

95. Typically, the owner of a domain name (*e.g.*, apple.com) will maintain a DNS server that is authoritative for all addresses within that domain. Ex. 1036 (RFC 1034) at 17-21, 27-28; Ex. 1045 (RFC 1123) at 50. The owner is responsible for managing the resource records (*e.g.*, records mapping a domain name to the corresponding IP addresses) on that name server, and can easily add new entries identifying host names within its zone. Ex. 1036 (RFC 1034) at 17-21, 27-28. For example, if Apple manages the apple.com zone, Apple could add entries for ipad.apple.com, air.apple.com, or itunes.apple.com to its name server without having to get permission from the owner of a name server higher up the heirarchy.

96. The owner of a zone may manage the resources listed on its name server by manually adding records to the server. Ex. 1036 (RFC 1034) at 17-21, 27-28. Alternatively, the owner could configure the name server to receive electronic messages that automatically would add, modify, or delete records. Ex. 1065 (RFC 2136) at 1, 6-10. Such a configuration was particularly useful on private DNS servers (*e.g.*, on a corporate intranet) where the IP addresses of computers might change due to use of DHCP or similar protocols. *See* Ex. 1065

(RFC 2136) at 21; Ex. 1039 (NT for Dummies) at 4-5, 6-7; *see also* ¶¶ 127-133, *below* (other protocols such as WINS can be used for a similar purpose).

97. A conventional DNS generally relies on the use of a recursive DNS server that responds to a client computer's request for resolution of a domain name. A recursive DNS server acts as a DNS resolver and does the work of traversing through DNS servers to find an authoritative answer. Ex. 1036 (RFC 1034) at 15-16, 20-21; Ex. 1037 (RFC 1035) at 3-5. A DNS resolver on a server is a proxy server that resolves requests on behalf of a client computer. Ex. 1036 (RFC 1034) at 28-30; Ex. 1037 (RFC 1035) at 3-5, 42-47.

98. The following example illustrates the standard domain name resolution process. The domain name resolution process starts when a user of the client computer types a DNS name, such as `www.apple.com`, into a web browser or any other DNS-enabled software application. Ex. 1036 (RFC 1034) at 11, 15.

99. To resolve the domain name, the browser contacts a recursive DNS server. The DNS resolver component of the server will contact the root DNS server, which will see that the domain name is a “.com” name, and will tell the resolver to contact the .com DNS server. The resolver will then contact the .com DNS server, which will see that the domain name is “apple.com” and will tell the resolver to contact the apple.com DNS server. Ex. 1036 (RFC 1034) at 11-12, 16-17.

100. The resolver will then contact the apple.com DNS server, which will see that the resolver is requesting the “www” host, and will return the IP address of the server at “www.apple.com”. Ex. 1036 (RFC 1034) at 16-17.

101. The resolver returns the address to the recursive DNS server, which returns the answer (the IP address associated with the domain name) to the client on the requesting computer (*e.g.*, the web browser).

102. The client can then use the IP address to open a socket to the remote host and start network communications. *See* Ex. 1036 (RFC 1034) at 11, 15-16, 26-27.

103. In practice, it is rare that the root server is actually contacted. ISPs and other service providers typically maintain DNS proxy servers that contain cached copies of the entries of the root server to improve the efficiency of the process. Ex. 1036 (RFC 1034) at 4-6, 20; *see* Ex. 1037 (RFC 1035) at 3-6; Ex. 1039 (NT for Dummies) at 13.

104. As noted above, a domain name is represented in a canonical form using dot separated fields. However, even if a domain name satisfies that canonical form, it will not be resolved by conventional DNS servers unless (1) its top level domain is a registered domain; and, (2) the remainder of the domain name below the top level domain, including the host name, is not registered with an authoritative DNS server.

105. I note that domain name servers do not simply or always return an IP address in response to a request. For example, if a domain name resolution request finds there is no IP address associated with a supplied domain name, the response returned to the client will not include an IP address. There still will be data in the response, but the data will include an error code (an “RCODE”). Ex. 1036 (RFC 1034) at 15-16, 43-44; Ex. 1037 (RFC 1035) at 25-27.

106. This is explained in RFC 1034 (Ex. 1036) and RFC 1035 (Ex. 1037). For example, RFC 1034 explains that a certain RCODE is returned if the domain name supplied in the request does not exist. *See* Ex. 1036 (RFC 1034) at 43-44 (“This response states that the name does not exist. This condition is signaled in the response code (RCODE) section of the header.”).

107. The RCODES that were in use before April of 2000 are documented in RFC 1035, which I have reproduced from the RFC below.

| RCODE | Response code - this 4 bit field is set as part of responses. The values have the following interpretation: |   |
|-------|---|---|
|       | 0   | No error condition  |
|       | 1   | Format error - The name server was unable to interpret the query. |
|       | 2   | Server failure - The name   |

|  |      |  |
|--|------|--|
|  |      | server was unable to process this query due to a problem with the name server.   |
|  | 3    | Name Error - Meaningful only for responses from an authoritative name server, this code signifies that the domain name referenced in the query does not exist.   |
|  | 4    | Not Implemented - The name server does not support the requested kind of query.  |
|  | 5    | Refused - The name server refuses to perform the specified operation for policy reasons. For example, a name server may not wish to provide the information to the particular requester, or a name server may not wish to perform a particular operation ( <i>e.g.</i> , zone transfer) for particular data. |
|  | 6-15 | Reserved for future use.   |

Ex. 1037 (RFC 1035) at 25-27.

108. Thus, if a DNS server cannot resolve a domain name, it will return data to the requester that **will not include an IP address but will include some form of an error message or code**. Ex. 1036 (RFC 1034) at 15-16, 43-446; Ex. 1037 (RFC 1035) at 25-27. The calling application (*e.g.*, a web browser) typically would report a DNS error to a user via a standard error message, such as “host unknown,” “host unavailable” or “connection refused” message.

109. For example, the domain name “[www.xyz.gov](#)” cannot be resolved by conventional DNS servers if the domain “xyz.gov” is not registered with an authoritative server. This can happen if the entry is not in any name database, or if the DNS request times out (*i.e.*, no response is received within the period specified for receiving a response).

110. Similarly, the domain name “[www.microsoft.scom](#)” (replacing .com with .scom) will not be resolved by conventional DNS servers because the top level domain “.scom” is not a valid registered top level domain, and as a consequence [www.microsoft.scom](#) also cannot be registered with ICANN.

111. A DNS server also may return an RCODE without an IP address for policy reasons. Ex. 1037 (RFC 1035) at 26-27 (“The name server refuses to perform the specified operation for policy reasons. For example, a name server may not wish to provide the information to the particular requester, or a name



server may not wish to perform a particular operation (*e.g.*, zone transfer) for particular data.”).

112. For example, a system including a DNS server could be configured to return only an RCODE if a user is found not to be authorized to access the DNS server. Another example would be that the DNS resolution would violate policies the administrator has established for providing access to network resources (*e.g.*, a request to access a particular server that is restricted to a particular zone or region of the network, or which only is to serve internal users of a private network).

113. The standardized DNS name resolution process also uses “timeout” periods that will return an error to the client if no response is returned within a specified period. *See* Ex. 1037 at 43-44 (“The key algorithm uses the state information of the request to select the next name server address to query, and also computes a timeout which will cause the next action should a response not arrive. The next action will usually be a transmission to some other server, but may be a temporary error to the client.”). A timeout of a name resolution request could also result in an RCODE being returned (*e.g.*, RCODE 2 or “server failure”).

114. Importantly, these access control concepts are built into the standards that govern DNS name resolution, and thus were very well known and established before April of 2000. For example, RFC 1034 and RFC 1035 were released in the 1987 – more than 13 years before the effective filing date of the ’181 patent.

#### **d. DNS SRV Records**

115. Since RFC 1034 and RFC 1035 were released in the 1987, many extensions to the DNS process have been proposed and adopted.

116. One extension is the SRV resource record specified in RFC 2052.

117. RFC 2052 was published in October 1996 as an experimental standard and was ratified as a standard in February 2000. Ex. 1082 (RFC 2052) at 1.

118. RFC 2052 specifies a standard for resolving the address of a particular service, as opposed to a particular host, on a domain. As RFC 2052 explains:

Clients ask for a specific service/protocol for a specific domain (the word domain is used here in the strict RFC 1034 sense), and get back the names of any available servers.

Ex. 1082 (RFC 2052) at 1.

119. The SRV resource record allows a client to ask a name server for a domain for the names of the computers in that domain that support certain protocols.

### **3. Communications and Name Resolution on Windows-Based Networks**

120. One of the most commonly used operating systems in April 2000 was Microsoft Windows, and in particular, Windows 98.

121. In April 2000, Microsoft created a suite of products, from server and client operating systems and software to business applications, that enabled an

organization to install and run Windows products on its backend server systems, its networks, and the computers of its employees.

122. Many organizations used Windows and other Microsoft products to power their employees' computers as well as to power the servers that ran their networks. Networks powered by and designed for Windows machines often were called Microsoft-based or Windows-based networks.

123. Microsoft incorporated into Windows 98 many features and tools that allowed it to easily connect to Microsoft-based networks. When a Windows 98 computer connected to such a network, the user would be able to access and use many standard and proprietary network features implemented by Microsoft.

124. Windows 98 contained components that allowed a user who was not in the office to connect remotely to the company's network. For example, a mobile employee using a laptop could connect to the company's network in at least two different ways. One was by dialing up a direct connection to a computer on the network using a modem. Another was by establishing a secure tunnel over the Internet. Ex. 1040 (Win98 Resource Kit) at 35-36, 183-187.

125. In Windows 98, the VPN functionality was "implemented using Point-to-Point Tunneling Protocol (PPTP). Because PPTP is a secure protocol, only authenticated users can gain access to your dial-up server. Also, you can encrypt data transfer to prevent Internet intruders from listening in." Ex. 1040

(Win98 Resource Kit) at 150-151; *see id.* at 37, 183-187. Here, it is recognized that a VPN could be created using PPTP and that using encryption with that technique is optional.

126. If a remote user established a VPN connection with a network, that user would be able to connect to the other computers and servers on the network and access and shared resources and files. Ex. 1040 (Win98 Resource Kit) at 149; *see* ¶ 134, *below*. Windows 98 contained many built-in and optional components that supported various networking protocols, such as TCP/IP, IPX/SPX, NetBIOS, and NetWare. Ex. 1040 (Win98 Resource Kit) at 46. Many networks employed TCP/IP protocol because it offered many benefits, such as allowing the network to easily route packets to and from Internet, which also was an IP-based network. Ex. 1040 (Win98 Resource Kit) at 38, 48-49; *see id.* at 51-55.

127. Computers connected to a TCP/IP network would be assigned IP addresses. *Id.* at 53-54. Windows 98 supported use of globally unique IP addresses, which could be used on the public Internet, as well as private IP addresses, which are only usable on the local network. Ex. 1040 (Win98 Resource Kit) at 53-54 (“Depending on your needs, you can use either private IP addresses or globally unique IP addresses for each workstation. You could also use both a private IP address and a globally unique IP address if you have a multihomed computer (a computer that has two different adapters, each connected to a different

network). For example, a workstation could be connected both to the corporate network (using an adapter that is configured with a private IP address) and to the Internet (using a dial-up adapter that is configured with a globally unique IP address.”).

128. Windows 98 allowed a user to connect to other computers on the same network. To enable a user friendly addressing scheme, Windows 98 supported both DNS and WINS, which are systems that translate or resolve human-understandable names into computer-understandable network addresses. Ex. 1040 (Win98 Resource Kit) at 73; *see id.* at 73-83.

129. Windows Internet Naming Service (“WINS”) was Microsoft’s implementation of the NetBIOS name system. Under the WINS protocol, each computer on a network was assigned a name. When the computer connected to the network, it would register its name and a network address with a WINS server. Ex. 1040 (Win98 Resource Kit) at 81-82 (WINS “is a service that runs on Windows NT Server to optimize NetBIOS name resolution. It provides a distributed database for **registering and querying dynamic computer name-to-IP address mappings** in a routed network environment. You can use WINS either alone or in conjunction with DNS.” (emphasis added)). Once the name was registered, a user on another computer on the network could send messages to that computer by asking the WINS server for address associated with the computer’s

name. *See* Ex. 1040 (Win98 Resource Kit) at 80-83.

130. Windows 98 supported DHCP, the Dynamic Host Configuration Protocol. Ex. 1040 (Win98 Resource Kit) at 60-61. Under DHCP, when a computer connected to a network, the DHCP server automatically would assign a dynamic IP address to the computer. On such networks, a computer would have a different address each time it connected to the network (such as after a system restart).

131. Even with DHCP, WINS enabled a computer to be accessed by its name because each time the computer connected to the network, it would register its name and address with the WINS server. Ex. 1040 (Win98 Resource Kit) at 81-82; Ex. 1039 (NT for Dummies) at 4-5, 6-7.

132. A Microsoft-based network could be configured to enable a computer running Windows 98 to be identifiable under both the WINS and TCP/IP addressing schemes. Ex. 1040 (Win98 Resource Kit) at 77. When configured in this way, the local DNS server would return the IP address of a Windows 98 computer using the computer's WINS name. To enable this functionality, the mapping of the WINS name of the Windows 98 computer to the IP address of the computer is registered (i.e., stored) in a distributed database, which can be used to handle queries for dynamic name-to-IP address mappings in a routed network environment. This is described, for example, in Ex. 1040 (Win98 Resource Kit) at

81 as follows:

## **Using WINS for Name Resolution**

Windows Internet Naming Service (WINS) is a service that runs on Windows NT Server to optimize NetBIOS name resolution. It provides a distributed database for registering and querying dynamic computer name-to-IP address mappings in a routed network environment. You can use WINS either alone or in conjunction with DNS.

WINS reduces the use of local broadcasts for name resolution and allows users to locate computers on remote networks automatically. Furthermore, when dynamic addressing through DHCP results in new IP addresses for computers that move between subnetworks, the changes are updated automatically in the WINS database. Neither the user nor the network administrator needs to make manual accommodations for name resolution in such a case.

WINS consists of two components: the WINS server, which handles name queries and registrations, and the client software (NetBIOS over TCP/IP), which queries for computer name resolution. A WINS server is a Windows NT Server 3.5 or later computer with WINS server software installed. When Microsoft TCP/IP is installed under Windows 98, WINS client software is installed automatically.

On a Windows-based network, users can browse transparently across routers. To allow browsing without WINS, you must ensure that the users' primary domain has Windows NT Server computers on both sides of the router to act as master browsers. These computers need to contain correctly configured LMHosts files with entries for the domain controllers across the subnet.

With WINS, such strategies are not necessary, because the WINS servers and proxies provide the support necessary for browsing Windows NT domains across routers. For a technical discussion of how WINS works and how it can be set up on the network, see *Windows NT Server 4.0 TCP/IP* in the Windows NT Server 4.0 documentation set.

If there are WINS servers installed on your network, you can use WINS in combination with broadcast name queries to resolve NetBIOS computer names to IP addresses. If you do not use this option, Windows 98 can use name query broadcasts (b-node mode of NetBIOS over TCP/IP) plus the local LMHosts file to resolve computer names to IP addresses. Broadcast resolution is limited to the local network, as described earlier in this section.

If DHCP is used for automatic configuration, these parameters can be provided by the DHCP server. Otherwise, you must configure information about WINS servers manually. WINS configuration is global for all network adapters on a computer.

133. Windows also allowed a configuration that permitted name resolution to be performed through a WINS Lookup configuration. In that configuration, a user on the organization's network could access another computer running Windows 98 by using its WINS name, and a user outside of the organization's network could access the computer through the Internet using its DNS name. This is explained, for example, at Ex. 1040 (Win98 Resource Kit) at 77 as follows:

#### **Using DNS with WINS Lookup**

If you are already using WINS servers for local name resolution but you want to make those servers visible to the global Internet, you can do so without configuring DNS on each computer. Windows NT Server 4.0 and later includes a feature called DNS with WINS Lookup. If you configure a Windows NT Server 4.0 and later to use DNS with WINS Lookup, that Windows NT Server can query the WINS servers for the "friendly names" of your computers, then use the information to construct a name that can be used on the global Internet.

For example, suppose the network **terrafirminc.tld** contains a Windows 98 computer named Annuals. On the internal network, the Windows NT Server is using DHCP to automatically assign IP addresses and WINS to resolve those IP addresses to the computer's "friendly names." You want Internet users to be able to connect to Annuals using the fully qualified domain name **annuals.terrafirminc.tld**. If you have Windows NT Server 4.0 or later, you can configure Microsoft DNS server on that computer to query WINS for the IP address for Annuals. WINS returns the IP address, and DNS then successfully resolves the IP address to the FQDN **annuals.terrafirminc.tld**.

For more information about this configuration, see the *Microsoft Windows NT Resource Kit* (for Microsoft Windows NT Server version 4.0).

134. Windows 98 contained components that allowed a user who was not

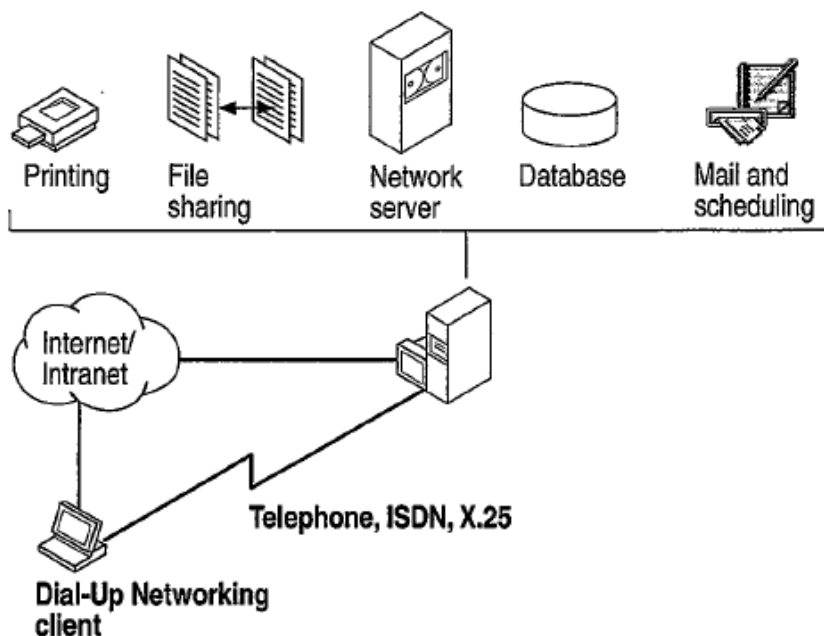


in the office to connect remotely to the company's network. For example, a mobile employee using a laptop could connect to the company's network in at least two different ways. One was by dialing up a direct connection to a computer on the network using a modem. Another was by establishing a secure tunnel over the Internet. Ex. 1040 (Win98 Resource Kit) at 150-151, 183-187.

135. In Windows 98, the VPN functionality was "implemented using Point-to-Point Tunneling Protocol (PPTP). Because PPTP is a secure protocol, only authenticated users can gain access to your dial-up server. Also, you can encrypt data transfer to prevent Internet intruders from listening in." Ex. 1040 (Win98 Resource Kit) at 150-151; *see id.* at 183-187.

136. If a remote user established a VPN connection with a network, that user would be able to connect to the other computers and servers on the network and access and shared resources and files. For example:

With Dial-Up Networking and virtual private networking, you can connect from a remote site to a computer that has been configured as a remote access server, or connect to a network through the remote access server. For example, as Figure 19.1 shows, if you connect to a Windows NT Remote Access Server, you can access its shared resources (if the Microsoft File and Printer Sharing service has been enabled), or you can use it as a gateway to a network that is running the TCP/IP, IPX/SPX, and NetBEUI network protocols.



**Figure 19.1 Connecting to a remote access server**

Figure 19.1 illustrates two types of connections: a dial-up connection and a virtual private network connection through the Internet. You would use either the dial-up connection or the virtual private network connection to access those resources.

Ex. 1040 (Win98 Resource Kit) at 144-149. As the figure shows, a user connecting via a VPN would be able to access files that were shared on the network, servers, and printers.

137. Windows 98 natively supported encryption of VPN traffic.

Encryption could be “required by either the server or the client” and Windows could be configured to “refuse to connect to a server that does not support data encryption.” Ex. 1040 (Win98 Resource Kit) at 150-151, 155; *see id.* at 183-187.

#### **4. Encryption, Tunnels, and Data Security**

138. In April of 2000, messages sent across the Internet were handled by many intermediaries, and consequently, could be intercepted and read or modified. This remains true today. As a result, many techniques for authenticating users and for providing data security have been developed.

**a. Encryption**

139. One way to prevent the intermediaries from reading or altering messages is through encryption. Encryption involves using a key to transform data into “cipher text” before sending it on a network and then using a key to transform the cipher text back into the original data.

140. There are two basic types of keys that can be used to encrypt data: symmetric and asymmetric.

141. In symmetric encryption, a single key is used to both encrypt and decrypt protected content. Symmetric encryption is fast, relatively simple, and straightforward to implement, which allows it to be more easily implemented in a wide range of hardware configurations. However, one problem with symmetric key encryption is that, because both the sender and receiver must know the key, the key must be transmitted in unencrypted form before use (or otherwise exchanged).

142. In contrast, asymmetric or Public Key Cryptography (“PK” or “PKI”) is based on a mathematical algorithm that allows the creation of two interlocking keys, one called the “Private Key” that is held in secret by a person or computer,

and a second “Public Key” that is distributed freely. *See generally* Ex. 1048 (Rivest) (filed December 14, 1977). Data encrypted with the private key can only be decrypted with the corresponding public key, and *vice versa*.

143. Many prior art systems used both methods of encryption to provide data security.

144. For example, during the 1990s the Secure Sockets Layer (“SSL”) transport protocol was conceived by Netscape Communications (to be superseded by the “Transport Layer Security” or TLS protocol in January 1999). Ex. 1052 (RFC 2246) at 5.

145. SSL uses asymmetric encryption to securely exchange a symmetric key, and then uses the symmetric key to encrypt subsequent communications within a particular HTTP session. Ex. 1052 (RFC 2246) at 3-5.

146. With the advent of SSL/TLS, the application data exchanged between a browser user and a secure web site could be kept confidential.

147. Similarly, SHTTP, a standard that was competing with SSL, also used both types of encryption. SHTTP incorporated many security features such as a challenge-response mechanism and the ability to ask a user to enter a username and password. Ex. 1049 (RFC 2660) at 8-12.

148. By far the most common implementation of such security features is based on the use of SSL/TLS protocols at the transport level.

## **b. Authentication**

149. Another way of preventing unauthorized entities from accessing a system is authentication. Many authentication techniques were well known in April of 2000. *E.g.*, Ex. 1040 (Win98 Resource Kit) at 5, 24-29; Ex. 1046 (RFC 2068) at 46, 88 (discussing HTTP proxy-authenticate and proxy-authorization headers).

150. Perhaps the most basic authentication technique is user-password authentication. Ex. 1040 (Win98 Resource Kit) at 5; Ex. 1046 (RFC 2068) at 46. In one implementation of this technique in Windows 98, an operating system that was popular in April 2000, a user provides a user name and password to a server or other security service, “which grants or denies the request by checking the requestor’s user name and password against a network-wide or server-wide stored list.” Ex. 1040 (Win98 Resource Kit) at 5; *see also id.* at 11. I note that this technique has been used for decades as a basic technique in client-server and other contexts. *See* Ex. 1046 (RFC 2068) at 46.

151. Other widely used techniques in April of 2000 include: the password authentication protocol (PAP), Ex. 1050 (RFC 1334) at 3-4; *see* Ex. 1046 (RFC 2068) at 46, Ex. 1062 (RFC 1928) at 2-3 (explaining that SOCKs supports user/password authentication); the challenge handshake protocol (CHAP), which is used to periodically verify the identity of the peer using a 3-way handshake, Ex.

1051 (RFC 1994) at 3-4; and digital “certificates” distributed by trusted third parties, Ex. 1040 (Win98 Resource Kit) at 24-29, Ex. 1060 (RFC 2510) at 3-6; amongst other methods.

**c. IPsec – RFC 2401**

152. Security has been a concern on the Internet since its inception. Not surprisingly, there were many mature, robust and well documented security protocols that were in widespread use before April of 2000.

153. One of the most widely known documents in April of 2000 devoted to security techniques suitable for Internet communications was RFC 2401, entitled Security Architecture for the Internet Protocol (IPsec), which was first published by the Internet Engineering Task Force (IETF) in November of 1998. Ex. 1032 (RFC 2401).

154. IPsec is a protocol for providing security services for traffic at the IP layer of Internet communications. Ex. 1032 (RFC 2401) at 4. Generally speaking, IPsec does this by enabling a system to select required security protocols, determine the algorithm(s) to use for the service(s), and put in place any cryptographic keys required to provide the requested services. Ex. 1032 (RFC 2401) at 7-8.

155. IPsec can be used to protect one or more “paths” between a pair of hosts, between a pair of security gateways, or between a security gateway and a host. Ex. 1032 (RFC 2401) at 7-8.

156. IPsec can use a variety of encryption techniques. Ex. 1032 (RFC 2401) at 8-9. As I explained earlier, encryption uses a secret key to encrypt the data before transmission, and this hides the actual contents of the packet from eavesdroppers.

157. Examples of encryption algorithms that were known in April 2000 and suitable for use in IPsec included DES, 3DES, Blowfish and AES.

158. IPsec provides secure communications between nodes and can be implemented in either a “transport” or “tunnel” model. Ex. 1032 (RFC 2401) at 7.

159. The IPsec Transport Mode provides a secure connection between two endpoints by encapsulating the payload of the IP packet. Ex. 1032 (RFC 2401) at 9. Tunnel Mode encapsulates the entire IP packet to provide a virtual “secure hop” between two gateways. Ex. 1032 (RFC 2401) at 9-10.

160. Many other methods of encrypting communications links were well known in the art, and had been documented in Internet standards publications. *See generally* Ex. 1054 (RFC 1968); Ex. 1055 (RFC 2420).

#### **d. Tunneling**

161. In April of 2000, many tunneling protocols were well known in the art.

162. Tunneling protocols generally involve creating a data packet intended for transmission on one network and encapsulating that packet within another data structure for transmission on another network. *See, e.g.*, Ex. 1064 (Hoke) at 7:47-53; Ex. 1032 (RFC 2401) at 32-35.

163. One commonly known tunneling technique was the Point-to-Point Protocol (“PPP”). PPP uses encapsulation to create a link between computers on two different systems. Ex. 1053 (RFC 1661) at 1.

164. For example, under PPP, a client would create a data packet intended for transmission on one network, *e.g.*, an IP network, and then encapsulate that packet within another packet intended for transmission on another network, *e.g.*, an IPX network. Ex. 1053 (RFC 1661) at 2, 6-7. This would allow a computer connected to a local network that used a propriety networking standard (like IPX) to send IP packets intended for transmission on the open Internet. *See* Ex. 1053 (RFC 1661) at 6-7.

165. Another tunneling protocol was the Layer 2 Tunneling Protocol (“L2TP”). Ex. 1058 (RFC 2661). Like PPP, L2TP is a technique for tunneling across different network types, but it operates at a lower level of the protocol stack. Ex. 1058 (RFC 2661) at 2-3. L2TP itself did not provide for encryption of packets,



but it was commonly combined with IPsec for additional security. Ex. 1058 (RFC 2661) at 48-49 (“The tunnel endpoints may optionally perform an authentication procedure of one another during tunnel establishment.”).

166. In April of 2000, these and many other similar tunneling methods were well known in the art. *E.g.*, Ex. 1057 (RFC 2364); Ex. 1059 (RFC 1483); Ex. 1056 (RFC 2118).

167. Another technique involves transporting packets containing private address information across a network using public addresses. One example of this is the Network Address Translation (“NAT”) protocol, which is the standard protocol used for transmitting packets between external networks and internal networks. Ex. 1038 (RFC 2663).

168. Internal networks often use private IP addresses. To enable computers connected to private networks to communicate across the Internet, the NAT protocol translates private IP addresses be translated into public ones. Ex. 1038 (RFC 2663).

169. The NAT methodology was introduced in the early 1990s. In April of 2000, it would be a routine matter for someone to use NAT to automatically modify packets transmitted from a computer trying to communicate beyond a NAT router. It was also well known in April of 2000 that NAT could be combined with other tunneling techniques. *See* Ex. 1064 (Hoke) at Figure 1.

170. NAT operates by presenting a single or small range of addresses to the Internet (the router “external” address). Behind the NAT router will be a much larger group of computers that use one of the reserved IP address blocks. A computer behind the NAT router is assigned an “inside” IP address by the NAT router, and typically also a port number. When that computer makes a request that must be sent out on the Internet, the NAT router takes the “inside” source IP address of the computer and replaces it with the NAT router's own “outside” IP address, and the assigned port number. Ex. 1038 (RFC 2663).

## **5. VPNs**

171. In April of 2000, the term “Virtual Private Network,” or VPN, referred to technology that enabled the creation of a private data “network” over a public telecommunications infrastructure.

172. The essence of a VPN is that it enables private communications to be transmitted over the insecure pathways of the open Internet. Ex. 1064 (Hoke) at 2:28-55. VPNs did this, and still do this, by using encryption, authentication, or tunneling, or combinations of these security techniques. *See* Ex. 1044 (Ferguson) at 16.

173. In April of 2000, VPNs were well known in the art. *See* Ex. 1043 (Ferguson); Ex. 1044 (Ferguson) (in a September 1998 article, explaining that VPNs were well known); Ex. 1041 (Aventail AutoSOCKS); Ex. 1042 (Aventail

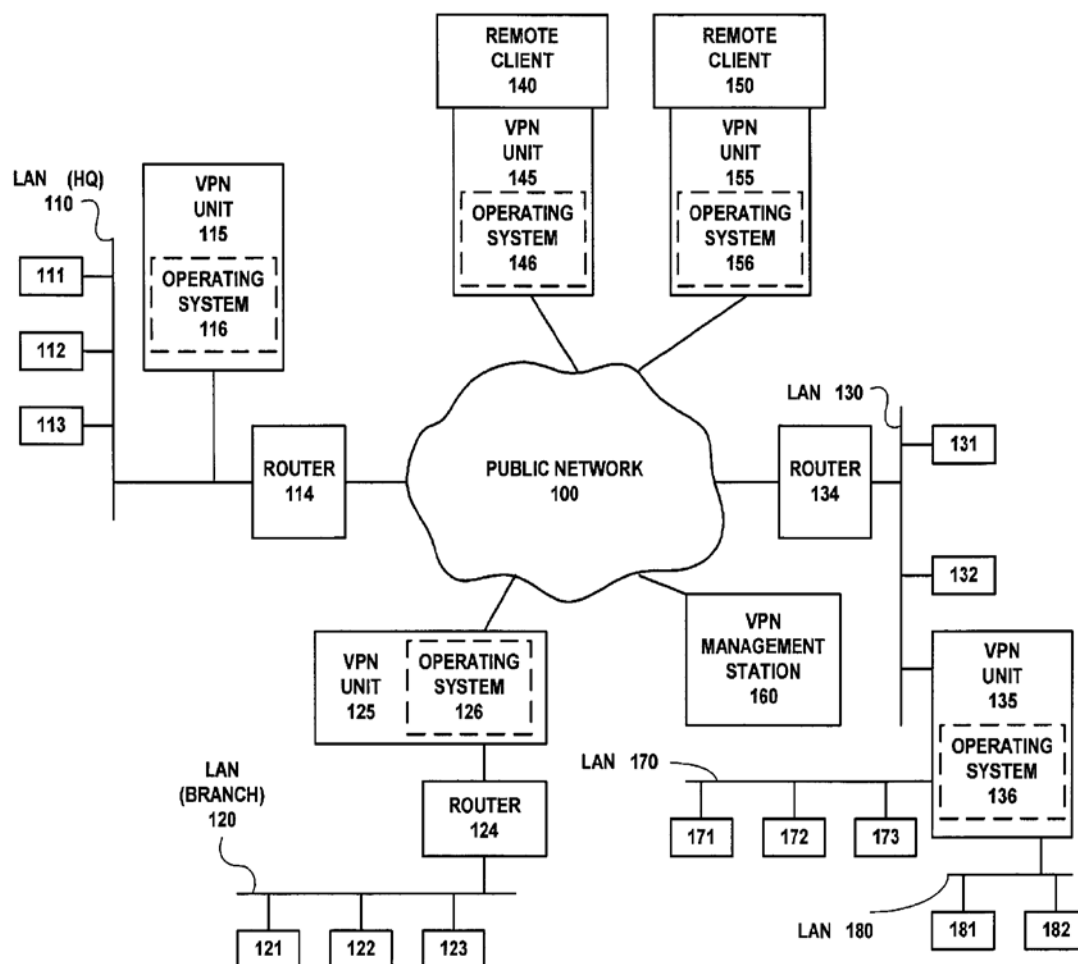
Connect v3.1); Ex. 1065 (Provino); Ex. 1053 (RFC 1661); Ex. 1058 (RFC 2661); Ex. 1064 (Hoke); Ex. 1040 (Win98 Resource Kit) at 150-187.

174. Several examples of prior art VPN systems are described in U.S. Patent No. 6,701,437 to Hoke. Ex. 1074 (Hoke). These examples show that many of the elements of the '181 claims were well known in the art prior to invention.

175. For example, Figure 3 of Hoke (Ex. 1074) shows a decisional process for evaluating packets sent on a network which is used to determine if the packets should be routed through the VPN server or through the Internet. If a packet were to be routed through the VPN server, the server would automatically encrypt the packet and apply other security rules to it. This process would be a routine exercise well known to people working in the field of secure networking in April of 2000 and for many years prior to that date.

176. The VPN systems described by Hoke also show the various types of VPN connections that can be made.

177. For example, Hoke shows that VPN connections can be made directly between individual computers, between an individual computer and a gateway on a LAN (*e.g.*, a remote user connecting to a corporate network), and between two gateways on different LANs (*e.g.*, a company using VPN as a WAN connection). *See* Ex. 1074 at Figure 1:



**FIG. 1**

178. Hoke thus shows a variety of VPN configurations that I recall were in use before April of 2000.

179. In one of the configurations illustrated in Hoke, a private network (e.g., corporate intranet) is shown having a VPN gateway computer situated between the private network and the Internet. A client outside of the network connects to this gateway server over the Internet. After being authenticated, the

client creates IP packets intended to be sent over the private network to a destination on that network. Hoke shows those packets being encapsulated inside IP packets sent to the gateway over the Internet. The gateway receives those encapsulated packets, removes the IP header information, and then transmits the packets on the private network to their intended destination. Ex. 1074 (Hoke) at 2:28-55.

180. Well before April of 2000, this type of system was widely used to enable employees to use a home computer or a laptop while travelling to connect via the Internet to a private network resource (*e.g.*, a corporate file or mail server to access files, download email and retrieve other resources from a corporate intranet). *See* Ex. 1040 (Win98 Resource Kit) at 149; Ex. 1074 (Hoke) at Figure 1.

181. By April of 2000, many commercial products were available that enabled companies to create secure communication over the Internet. One example were edge routers like the Beser network devices. *See* Ex. 1031 (Beser). Another example was the Aventail Extranet server, which was widely commercialized and had Fortune 100 companies as customers as early as January 1999. In fact, the Aventail VPN products was so well known they were mentioned in an introductory book for configuring and using Windows NT Servers. Ex. 1039 (Windows NT Server 4 for Dummies) at 13-14.

## **6. Multimedia Transmission over the Internet**

182. In April of 2000, several suites of standards for the transmission of multimedia data over the Internet or other networks existed.

183. Because of the complex nature of multimedia transmission, no single standard governed it. Instead, such transmissions were governed by suites of standards. A distinct standard would be used for each aspect of the transmission process, such as: for initiating and terminating a transmission, for regulating a transmission, for describing a transmission, for encoding the data to be transmitted, and for transporting a transmission.

184. The ITU had developed a suite of standards for multimedia transmission that included H.323, H.225, H.235, and H.245. H.323 is an umbrella standard that describes how the other standards are used within the multimedia call process. *See* Ex. 1077 (H.323) at 3 (“Products claiming compliance with Version 1 of H.323 shall comply with all of the mandatory requirements of H.323 (1996) which references Recommendations H.225.0 (1996) and H.245 (1996).”), 13 (citing H.235), 91 (“Authentication and security for H.323 systems is optional; however, if it is provided, it shall be provided in accordance with Recommendation H.235.”). H.225 specifies the call signaling channel which is used to establish and terminate calls. H.245 is a control protocol used to carry control messages and define the data to be transferred. H.235 is a security protocol that specifies a secure mode of operation for H.323 terminals.

185. H.323 calls can be used to transmit voice and video. A call or communication between parties would include multiple channels or sessions. Different types of media were sent over different channels.

186. The IETF also had developed a suite of standards. In April 2000, the approved standards in the IETF suite included SIP, SDP, and RTP. Like the ITU's protocol, a call or communication between parties would include multiple channels or sessions, with different types of media being send over each one.

187. Both H.323 and SIP are able to interface with Public Switched Telephone Networks (PSTN or GSTN for Global Switched Telephone Networks). Ex. 1033 (RFC 2543) at 8 ("SIP can also initiate multi-party calls using a multipoint control unit (MCU) or fully-meshed interconnection instead of multicast. Internet telephony gateways that connect **Public Switched Telephone Network (PSTN)** parties can also use SIP to set up calls between them."); *id.* ("SIP can also be used in conjunction with other call setup and signaling protocols. . . . In another example, SIP might be used to determine that the callee is reachable via the PSTN and indicate the phone number to be called, possibly suggesting an **Internet-to-PSTN gateway** to be used."); *see id.* at 91; Ex. 1077 (H.323) at 3 ("H.323 entities may be used in point-to-point, multipoint, or broadcast (as described in Recommendation H.332) configurations. They may interwork with H.310 terminals on B-ISDN, H.320 terminals on N-ISDN, H.321

terminals on B-ISDN, H.322 terminals on Guaranteed Quality of Service LANs, H.324 terminals on **GSTN and wireless networks**, V.70 terminals on **GSTN**, and voice terminals on **GSTN** or ISDN through the use of Gateways.”).

188. It was common for companies or large organizations to configure their communications systems so that they could integrate their data networks and phone network. For example, by connecting their intranet to both the Internet and a Private Branch Exchange (PBX), an organization could enable its VOIP phone system to route calls over the PSTN to a traditional phone, or to route them over the Internet or intranet to other VOIP phones. This allowed for easy integration of multimedia services, such as those provided by H.323. *See, e.g.*, Ex. 1078 (Christie) at 7:18-59.

189. I discuss these protocols as they relate to SIP in more detail below.

#### **F. Common Design Perspectives in April 2000**

190. I believe, based on my experiences prior to April of 2000, that there were a number of well-accepted principles that would influence the design and implementation of secure networking systems. In this section, I will summarize some of these principles.

191. One well-accepted principle was it was possible to deploy a set of functional capabilities of a system in either a consolidated design (*e.g.*, a single server) or by distributing those functions (*e.g.*, multiple servers that can



communicate with each other). For example, to someone working in the field of the '181 patent, it would have been routine to deploy a set of functionalities (*e.g.*, DNS server, encryption layer, packet routing schemes) either as services on a single server, or to distribute those functionalities across several servers that could communicate with each other. Distributing or consolidating a set of functions would have been considered a routine design choice and could be implemented with only routine effort by someone working in the field of the '181 patent.

192. It also was common to refer to RFPs and other standards documents to explain how to configure and deploy services and systems. So, for example, instead of spelling out the requirements and implementation of a particular networking protocol or security scheme (*e.g.*, IPSec), you would just refer to the particular standards document (*e.g.*, an RFP). By doing that, it was understood that the referenced standard or protocol would be consulted, as needed, to understand the technical requirements and configuration issues associated with that standard or protocol. In other words, referring to a particular RFP or standards document would be a shorthand way of directing people to the technical information in that RFP or standards document.

193. I believe another well-accepted design concept was to use equivalent functional services when designing and deploying systems. For example, if a system had used a particular encryption technique or protocol, it would be simple

and routine for a person to replace that encryption technique or protocol with another one that provided it served the same functional purpose (*e.g.*, encrypting of routed traffic). The focus when considering design options was more on the functional role of the component or service, rather than on the particular features of the component or particular protocol or standard being used.

194. I also believe people working in the field of secure networking would generally consider it a good practice to combine or add techniques or components that would yield a more secure network or communication process. So, for example, a person would not think twice of adding a secure IP traffic routing protocol into an already secure VPN networking platform. While the VPN networking platform might be fine as is, the addition of other security techniques would be immediately recognized as being useful and desirable, and would be considered as part of the routine process of designing and implementing a secure networking environment.

#### **IV. General Issues Related to My Patentability Analysis**

195. As I explain in more detail below, I believe claims 1-29 of the '181 patent are either anticipated or would have been considered obvious by a person of ordinary skill in the art based on a number of prior art references, particularly when the claims are given their broadest reasonable interpretation consistent with the specification.

**A. The Claims of the '181 Patent I Am Addressing in this Report**

196. The claims of the '181 patent that I am addressing in this report (*i.e.*, 1-29) are reproduced below.

Claim 1 of the '181 patent reads:

1. A non-transitory machine-readable medium comprising instructions for a method of communicating with a first device associated with a secure name and an unsecured name, the method comprising:

receiving, at a network address corresponding to the secure name associated with the first device, a message from a second device of the desired to securely communicate with the first device; and

sending a message over a secure communication link from the first device to the second device.

Claim 2 of the '181 patent reads:

2. A method of using a first device to communicate with a second device having a secure name, the method comprising:

from the first device, sending a message to a secure name service, the message requesting a network address associated with the secure name of the second device;

at the first device, receiving a message containing the network address associated with the secure name of the second

device; and

from the first device, sending a message to the network address associated with the secure name of the second device using a secure communication link.

Claim 3 of the '181 patent reads:

3. The method according to claim 2, wherein the secure name of the second device is a secure domain name.

Claim 4 of the '181 patent reads:

4. The method according to claim 2, wherein the secure name indicates security.

Claim 5 of the '181 patent reads:

5. The method according to claim 2, wherein receiving the message containing the network address associated with the secure name of the second device includes receiving the message in encrypted form.

Claim 6 of the '181 patent reads:

6. The method according to claim 5, further including decrypting the message.

Claim 7 of the '181 patent reads:

7. The method according to claim 2, wherein the second device is capable of supporting a secure communication link as well as

a non-secure communication link, the method further including establishing a non-secure communication link with the second device when needed.

Claim 8 of the '181 patent reads:

8. The method according to claim 2, wherein receiving a message containing the network address associated with the secure name of the device includes receiving the network address as an IP address associated with the secure name of the device.

Claim 9 of the '181 patent reads:

9. The method according to claim 2, further including automatically initiating the secure communication link after it is enabled.

Claim 10 of the '181 patent reads:

10. The method according to claim 2, wherein receiving a message containing the network address associated with the secure name of the device includes receiving the message at the first device through tunneling within the secure communication link.

Claim 11 of the '181 patent reads:

11. The method according to claim 2, wherein receiving a message containing the network address associated with the secure name of the device includes receiving the message in the

form of at least one tunneled packet.

Claim 12 of the '181 patent reads:

12. The method according to claim 2, wherein the receiving and sending of messages includes receiving and sending the messages in accordance with any one of a plurality of communication protocols.

Claim 13 of the '181 patent reads:

13. The method according to claim 2, wherein the receiving and sending of messages through the secure communication link includes multiple sessions.

Claim 14 of the '181 patent reads:

14. The method according to claim 2, further including supporting a plurality of services over the secure communication link.

Claim 15 of the '181 patent reads:

15. The method according to claim 14, wherein the plurality of services comprises a plurality of communication protocols, a plurality of application programs, multiple sessions, or a combination thereof.

Claim 16 of the '181 patent reads:

16. The method according to claim 15, wherein the plurality of application programs comprises video conferencing, e-mail, a

word processing program, telephony or a combination thereof.

Claim 17 of the '181 patent reads:

17. The method according to claim 15, wherein the plurality of services comprises audio, video or a combination thereof.

Claim 18 of the '181 patent reads:

18. The method according to claim 2, wherein the secure communication link is an authenticated link.

Claim 19 of the '181 patent reads:

19. The method according to claim 2, wherein the first device is a computer, and the steps are performed on the computer.

Claim 20 of the '181 patent reads:

20. The method according to claim 2, wherein the first device is a client computer connected to a communication network, and the method is performed by the client computer on the communication network.

Claim 21 of the '181 patent reads:

21. The method according to claim 2, further including providing an unsecured name associated with the device.

Claim 22 of the '181 patent reads:

22. The method according to claim 2, wherein the secure name is registered prior to the step of sending a message to a secure

name service.

Claim 23 of the '181 patent reads:

23. The method according to claim 2, wherein the secure name of the second device is a secure, non-standard domain name.

Claim 24 of the '181 patent reads:

24. A method of using a first device to securely communicate with a second device over a communication network, the method comprising:

at the first device requesting and obtaining registration of a secure name for the first device, the secure name being associated with a network address;

receiving at the network address associated with the secure name of the first device a message from a second device of the desire to securely communicate with the first device; and

sending a message securely from the first device to the second device.

Claim 25 of the '181 patent reads:

25. The method according to claim 24, wherein requesting and obtaining registration of a secure name for the first device comprises using the first device to obtain a registration of the secure name for the first device, and wherein sending a message securely comprises sending the message from the first device to



the second device using a secure communication link.

Claim 26 of the '181 patent reads:

26. A method of using a first device to communicate with a second device over a communication network, the method comprising:

from the first device requesting and obtaining registration of an unsecured name associated with the first device;

from the first device requesting and obtaining registration of a secure name associated with the first device, wherein a unique network address corresponds to the secure name associated with the first device;

receiving at the unique network address associated with the secure name a message from a second device requesting the desire to securely communicate with the first device; and

from the first device sending a message securely from the first device to the second device.

Claim 27 of the '181 patent reads:

27. The method according to claim 26,

wherein requesting and obtaining registration of an unsecured name associated with the first device comprises using the first device to obtain a registration of the unsecured name associated with the first device, and

wherein requesting and obtaining registration of a secure name associated with the first device comprises using the first device to obtain a registration of the secure name associated with the first device.

Claim 28 of the '181 patent reads:

28. A non-transitory machine-readable medium comprising instructions for:

    sending a message to a secure name service, the message requesting a network address associated with a secure name of a device;

    receiving a message containing the network address associated with the secure name of the device; and

    sending a message to the network address associated with the secure name of the device using a secure communication link.

Claim 29 of the '181 patent reads:

29. A non-transitory machine-readable medium comprising instructions for a method of communicating with a device having a secure name, the method comprising:

    receiving at a network address associated with a secure name of a first device a message from a second device requesting the desired to securely communicate with the first device, wherein the secure name of the first device is registered;

and

sending a message securely from the first device to the second device.

## **B. Comparison with Patent Family Members**

197. I note that the claims in the '181 Patent very similar to the claims in the '274 and 180 patents. The independent claims are substantially similar to each other, and each patent has a similar set of dependent claims.

198. The table below compares claim 2 of the '181 patent, claim 1 of the '274 patent, and claim 1 of the '180 patent. As shown in the table, each claim can be conceptualized as specifying the performance of four steps.

| <b>181 – Claim 2</b>   | <b>274 – Claim 1</b>  | <b>180 – Claim 1</b>  |
|--|---|---|
| A method of using a first device to communicate with a second device having a secure name, the method comprising:  | A method of accessing a secure network address, comprising:   | A method for accessing a secure computer network address, comprising steps of:  |
| from the first device, <b>sending a message to a secure name service</b> , the message <b>requesting a network address associated with the secure name</b> of the second device; | <b>sending a query message</b> from a first network device <b>to a secure domain service</b> , the query message <b>requesting from the secure domain service a secure network address</b> for a second network device; | receiving a secure domain name; <b>sending a query message to a secure domain name service</b> , the query message <b>requesting from the secure domain name service a secure computer network address corresponding to the secure domain</b> |

|  |   |  |
|--|---|--|
|  |   | <b>name;</b>   |
| at the first device,<br><b>receiving a message containing the network address associated with the secure name</b> of the second device; and                              | <b>receiving</b> at the first network device a <b>response message from the secure domain name service containing the secure network address</b> for the second network device; and | <b>receiving from the secure domain name service a response message containing the secure computer network address corresponding to the secure domain name;</b><br>and |
| from the first device,<br><b>sending a message to the network address associated with the secure name</b> of the second device using a <b>secure communication link.</b> | <b>sending an access request message</b> from the first network device <b>to the secure network address</b> using a <b>virtual private network communication link.</b>              | <b>sending an access request message to the secure computer network address</b> using a <b>virtual private network communication link.</b>                             |

199. First, each claim speaks of “communicat[ing] with” or “accessing” a secure location having a secure “network address” corresponding to a “secure name” or “secure domain name.” A “secure domain name” is a type of “secure name” and therefore any reference that shows use of a “secure domain name” would show a “secure name”.

200. Second, each claim refers to querying a secure “name service” or “domain name service” for a “network address” associated with a secure location. A “secure domain name service” is one type of “name service” and therefore any reference that discloses the former also discloses the latter.

201. Third, each claim indicates that the requested secure “network address” is received by a requester.

202. Fourth, each claim indicates that a “message” or an “access request message” is sent to the returned secure “network address” over a “secure” or “virtual private” “communication link.” An “access request message” is one type of a “message” – any reference that shows transmission of an “access request message” also shows transmission of a “message.” As explained in the claim construction sections, there is no relevant distinction between a “secure communication link” or a “virtual private network communication link” as both terms refer to secure links.

203. A person of ordinary skill in the art would consider the minor differences in language to be immaterial to the three processes.

### **C. Interpretation of Certain Claim Terms**

204. I understand that in an *inter partes* review proceedings, claims are to be given their broadest reasonable interpretation in view of the specification.

205. I also understand that, where a patent applicant explicitly defines a term to mean something in the patent disclosure, that definition should be used when evaluating the claims. An explicit definition will be something like “a ‘foo’ means ‘a widget that is 4 inches wide by 6 inches long.’”

206. I understand that if no explicit definition is provided for a term in the patent specification, the claim terms must be given their plain meaning unless that would be plainly inconsistent with how the term is being used in the claim or the patent specification.

207. I also understand that the Patent Owner has argued that certain terms in the claims have a particular meaning in district court litigation and in other PTO proceedings. I have considered those interpretations in reaching my conclusions about what the terms in the claims mean.

208. I understand that the standards used by a district court to interpret the claims are different than those used by the PTO in this proceeding. The main difference, as I understand it, is that in this proceeding, the claims are to be read as broad as is reasonable based on the specification. This may cause the claims to cover certain things that a Court might find are not within the scope of the claims in its proceeding.

# **1. Secure Name**

209. Each of the independent claims of the '181 patent uses the phrase **“secure name.”**

210. I did not find the phrase “secure name” in the '181 patent disclosure. There is nothing in the '181 patent that tries to define this term.

211. I understand that the claim 3 of the '181 patent says that “the secure name of the second device is a secure domain name.” I also understand this means that a “secure name” would logically include a secure **domain** name but can include other things as well.

212. I understand that during prosecution of the '181 patent, Patent Owner made statements to the Patent Office about the phrase “**secure name.**” For example, I understand that in a final rejection of the claims of the '181 patent, the Patent Office explained that “it is unclear what kind of ‘secure name’ that the applicants are talking about (i.e., is it a secure domain name?).” Ex. 1026 (181 File History) at 782-783 (Final Rejection at 3). In response, Patent Owner stated:

**[T]he Applicant submits that a “secure name” is a name associated with a network address of a first device. The name can be registered such that a second device can obtain the network address associated with the first device from a secure name registry and send a message to the first device. The first device can then send a secure message to the second device. The claimed “secure name” includes, but is not limited to, a secure domain name. For example, a “secure name” can be a secure non-standard domain name, such as a secure non-standard top-level domain name (e.g., .scom) or a telephone number.**

Ex. 1026 (181 FH) at 813 (Remarks to Office Action Oct. 8, 2010) (emphasis added); *see also* Ex. 1025 at 52:14-27 (providing additional examples of secure domain names).

213. I also understand that during the reexamination of the '181 patent, Patent Owner made a number of statements about the meaning of the phrase “**secure name**.” For example, Patent Owner stated: “‘secure names’ are those names used to communicate securely that are resolved by a secure name service.” Ex. 1072 ('949 reexam FH) at 688-689 (Response to Office Action, Sept. 4, 2012 at 45-46).

214. As I explain in greater detail below, I understand that the Patent Owner in litigation argued that a “secure domain name” is a “non-standard domain name that corresponds to a secure computer network address and cannot be resolved by a conventional DNS.” Ex. 1017 (Jt Claim Construction Chart) at 19-20. These statements suggest the term “secure name” is both related to and broader than the meaning of a secure domain name. Ex. 1026 (181 FH) at 813 (Remarks to Office Action Oct. 8, 2010). So, for example, it could include a telephone number or a non-standard top level domain.

215. I believe the broadest reasonable construction of this “secure name” thus should include “*a name that corresponds to a secure computer network*”



*address that cannot be resolved by a conventional name service.”* Ex. 1017 (Jt Claim Construction Chart) at 19-20.

## 2. Secure Name Service

216. Independent claims 2 and 28 of the ’181 patent uses the phrase “**secure name service.**”

217. I did not find the phrase “secure name service” in the ’181 patent disclosure. I also did not find an explicit definition of this phrase.

218. I understand that during examination of the ’181 patent, the Patent Owner amended its claims to include the phrase “**secure name service.**” Ex. 1026 (181 FH) at 806 (Claims Oct. 8, 2010). Patent Owner explained to the Patent Office that “[t]his secure name service is a service for **resolving** secure names into network addresses.” Ex. 1026 (181 FH) at 812 (Remarks Oct. 8, 2010) (emphasis in original).

219. I understand that during examination of the ’181 patent, Patent Owner stated that “a ‘secure name’ is a name associated with a network address of a first device. The name can be registered such that a second device can obtain the network address associated with the first device from a secure name registry and send a message to the first device.” Ex. 1026 (181 FH) at 813 (Remarks Oct. 8, 2010).

220. I understand that during the *inter partes* reexamination of the '181 patent, Patent Owner stated that: “‘**secure names**’ are those names used to communicate securely that are resolved by a **secure name service** (*i.e.*, a service that both resolves a name into a network address and further supports establishing a secure communication link). Ex. 1072 ('949 reexam FH) at 688-689 (Response to Office Action, Sept. 4, 2012).

221. There is no discussion in the '181 patent stating or suggesting that a secure name service must “provide any further support for establishing a secure communication link.”

222. I understand that during litigation before the International Trade Commission, Investigation No. 337-TA-858, Patent Owner stated that “**secure name service**” should be understood to mean “[a] lookup service that returns a network address for a requested secure name and facilitates establishing a secure communication link based on a secure name.” Ex. 1071 (ITC Jt Claim Construction) at 2.

223. There is no mention in the '181 patent of a “secure name service” – the phrase only shows up in the claims. Therefore it is not clear what the Patent Owner wanted this phrase to mean – or whether it was supposed to be the same thing as a secure **domain** name service.

224. A person of ordinary skill would understand a “name service” to be a conventional domain name service that provides network addresses corresponding to network names in response to queries.

225. I understand that during examination of the ’181 patent, Patent Owner stated that a “**secure name**” can include a “secure . . . domain name.” Ex. 1026 (181 FH) at 813 (Remarks Oct. 8, 2010). That comment suggests that a “**secure name service**” may be a broader type of a “secure domain name service.” Thus, discussion in the ’181 patent about the “secure **domain** name service” is relevant to deciphering the broadest reasonable interpretation of “**secure name service**.”

226. The secure domain name service discussed in the ’181 patent would appear to function in an analogous manner to a domain name service, but only acts on secure domain names (*i.e.*, by providing the the network address corresponding to the secure domain name). For example, the ’181 patent states at col. 6:22-31:

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.**

227. The '181 patent also states at 7:20-37:

**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.

228. The '181 patent also discusses operation of its secure domain name service feature at 50:19-35:

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.

229. From these discussions of the secure domain name service, it is clear the secure domain name service is discrete component or service that is distinct from other components and services in the network. For example, another component – the VPN gatekeeper – handles VPN creation and communications. The '181 patent explains this relationship at 51:16-29 as follows:

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.

230. Based on these passages in the '181 patent, it is also clear that the “secure domain name service” is not something that “provides further support” or “facilitates” establishing a secure communication link beyond providing name resolution services for a particular type of domain name – the unique secure domain names that are discussed in the patent.

231. The secure domain name service discussed in the '181 patent would appear to function in an analogous manner to a domain name service, but only acts on secure domain names (*i.e.*, by providing the the network address corresponding to the secure domain name).

232. For example, the '181 patent explains that, “for each secure domain name, SDNS 3313 stores a computer network address corresponding to the secure domain name.” Ex. 1025 at 50:60-64. The patent further explains that “[a]n 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.” Ex. 1025 (181 patent) at 50:64-67.

233. I understand that in a reexamination proceeding involving the '180 patent which is a parent of the '181 patent, Patent Owner argued that the claim

term “secure domain name service” should be understood to refer to something “different from a conventional domain name service.” Ex. 1023, p. 232 (“Response to Office Action in Reexamination” filed Apr. 19, 2010 at 8). In making that argument, VirnetX stated that “the ’180 Patent explicitly states that a secure domain name service can resolve addresses for a secure domain name; whereas, a conventional domain name service cannot resolve addresses for a secure domain name.” *Id.* at 7. In response to that argument, the Patent Office stated “the ’180 patent explains that a secure domain name service can resolve addresses for a secure domain name whereas a conventional domain name service cannot resolve addresses for a secure domain name.” Ex. 1023 (270 Reexam FH) at 119 (Action Closing Prosecution of Jun. 16, 2010 at 3).

234. I understand that in an earlier litigation involving the ’180 patent, Patent Owner asserted that a “secure domain name service” is “a service that receives requests for secure computer network addresses corresponding to secure domain names, and is capable of providing trustworthy responses.” Ex 1016 at 31. Finding no support for “capable of providing trustworthy responses,” the District Court adopted a broader construction and found the phrase means “a lookup service that returns a secure network address for a requested secure domain name.” Ex. 1016 at 31.

235. Considering all of these points, I believe the broadest reasonable construction of the term “secure name service” would at least include “*a service that can resolve network addresses for a secure name for which a conventional name service cannot resolve addresses.*” It is unclear what else the phrase might cover given that it was not used in the ’181 patent other than in the claims.

### 3. Secure Domain Name

236. Claim 3 includes the phrase “**secure domain name.**”

237. The ’181 patent explains that a “secure domain name” is a domain name that corresponds to a secure computer. Ex. 1025 (181 patent) at 50:60-51:15. For example, it states:

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.

238. The '181 patent also explains that “[b]ecause 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 indication that the universal resource locator (URL) is unknown.” Ex. 1025 (181 patent) at 50:19-22. This explanation — *i.e.*, a secure domain name that is not capable of resolution by conventional DNS servers—is used consistently throughout the specification.

239. As I explained above, the Patent Owner stated the following during examination of the '181 patent:

**[T]he Applicant submits that a “secure name” is a name associated with a network address of a first device.** The name can be registered such that a second device can obtain the network address associated with the first device from a secure name registry and send a message to the first device. The first device can then send a secure message to the second device. **The claimed “secure name” includes, but is not limited to, a secure domain name. For example, a “secure name” can be a secure non-standard domain**

**name, such as a secure non-standard top-level domain name (e.g., .scom) or a telephone number.**

Ex. 1026 (181 FH) at 813 (Remarks to Office Action Oct. 8, 2010) (emphasis added). *See also* Ex. 1025 (181 patent) at 52:16-35 (providing additional examples of secure domain names).

240. In this explanation of the term “**secure name**” to the Patent Office, the Patent Owner made it clear what it considered a “secure domain name” to be – it is a “secure non-standard top-level domain name” such as “.scom.”

241. A person of ordinary skill in April of 2000 would have thus understood that a “**secure domain name**” to be a non-standard domain name that could not be resolved by a conventional DNS server.

242. This is consistent with the Patent Owner’s statements in *Inter Partes* Reexamination No. 95/001,270 (“‘270 Reexamination”), which involves the parent to the ’181 patent. I understand in that proceeding, the Patent Owner argued that that a “secure domain name” is a name that “cannot be resolved by a conventional domain name service.” Ex. 1023 (’270 Reexam FH) at 229-31 (“Response to Office Action in Reexamination” filed Apr. 19, 2010 at 6). In response, the Patent Office found that a “a secure domain name is a non-standard domain name and that querying a convention domain name server using a secure domain name will result

in a return message indicating that the URL is unknown.” Ex. 1023 (270 Reexam FH) at 119, 129-30 (Action Closing Prosecution of Jun. 16, 2010, p. 3).

243. Finally, I understand that the Patent Owner asserted in litigation that a “*secure domain name*” means a “*non-standard domain name that corresponds to a secure computer network address and cannot be resolved by a conventional DNS.*” Ex. 1017 (Jt Claim Construction) at 19-20.

244. Based on all of these considerations, I believe the broadest reasonable interpretation of “secure domain name” would be a “non-standard domain name that corresponds to a secure computer network address and cannot be resolved by a conventional DNS.”

245. I note that while the ’181 patent explains that a “secure domain name” cannot be resolved by a conventional DNS, this would appear to be simply the consequence of the conventional domain name service not containing entries for these secure domain names, and being unable to communicate with the secure domain name server that contains those names. The descriptions in the ’181 patent do make it clear that the secure domain name server functions in the analogous manner to a conventional domain name server – it receives queries, looks up network addresses associated with the name in its database or table, and returns that network address if it is present.

#### **4. Unsecured Name**

246. Several claims of the '181 patent use the phrase “**unsecured name.**”

247. The only place this phrase appears is in the claims of the '181 patent; the phrase “unsecured name” is not used anywhere else in the '181 patent specification.

248. The '181 patent specification does describe an “‘unsecure’ target site 2611” as being “accessible via conventional IP protocols.” Ex. 1025 (181 patent) at 39:49-52.

249. A person of ordinary skill would thus understand that an “unsecured name” would be a name referring to a computer that would be accessible using conventional IP protocols. Thus, an “unsecured name” would therefore encompass a conventional domain name—*i.e.*, a name that can be resolved by a conventional domain name service.

250. I also understand that during litigation before the International Trade Commission, Investigation No. 337-TA-858, Patent Owner contended that an “*unsecured name*” should be understood to mean “*a name that can be resolved by a conventional name service.*” Ex. 1071 (Joint list of disputed claims re ITC) at 2.

## **5. Secure Communication Link (Claims 1-23, 25, and 28) and Securely Communicate (Claims 24, 27, and 29)**

251. Independent claims 1, 2, and 28, and dependent claim 25 (from 24) of the '181 patent use the phrase “**secure communication link.**” I understand this

means claims 1-23, 25, and 28 of the patent therefore concern a secure communication link.

252. The '181 patent states that “The secure communication link is a virtual private network communication link over the computer network.” Ex. 1025 (181 patent) at 6:57-59. The '181 patent thus appears to equate a “secure communication link” with a VPN, and a “secure communication link” must therefore encompass the same features as a virtual private network.

253. Independent claims 24, 27, and 29 of the '181 patent use the phrase “securely communicate.” There is no definition of this phrase anywhere in the patent. It is only used once, and that is simply to indicate that computers can securely communicate. *See* Ex. 1025 at 7:38-43 (“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.”). This suggests “securely communicating” means that the communications are being handled via a VPN/secure communication link.

254. I understand that Patent Owner asserted in litigation that a “secure communication link” is “an encrypted communication link.” Ex. 1066 at 10-11.

255. The '181 patent specification does not explicitly define “virtual private network.” I did not find anything in the '181 patent specification that

attempts to give this phrase a special meaning. I therefore believe the ordinary meaning of a “virtual private network” existing in April of 2000 would be what a person of ordinary skill would interpret a “secure communication link” to mean.

256. I understand that Patent Owner has asserted in litigation that a “virtual private network” is “a network of computers which privately communicate with each other by encrypting traffic on insecure communication paths between the computers.” Ex. 1066 at 3-8.

257. The broadest reasonable construction of “secure communication link” is broader than what the Patent Owner says it is. To a person of ordinary skill in the art in April of 2000, a secure communication link would be understood to include networks of computers that can privately communicate with each other over an insecure communication path between the computers, provided the communications are anonymous and secure. Encryption is typically used to provide this anonymity and security, but is not essential because other techniques can be used to ensure the anonymity and security of the secure communications.

258. I am familiar with several publications from before April 2000 that explain what a “VPN” is. For example, in a paper by Paul Ferguson and Geoff Huston published in September of 1998, these authors describe a VPN as follows:

So what is a Virtual Private Network? As we have discussed, a VPN can take several forms. A VPN can be between two end-systems, or it can be between two or more networks. **A VPN can be built using**

**tunnels or encryption** (at essentially any layer of the protocol stack), **or both, or alternatively constructed using MPLS or one of the “virtual router” methods.** A VPN can consist of networks connected to a service provider’s network by leased lines, Frame Relay, or ATM, or a VPN can consist of dial-up subscribers connecting to centralized services, or other dial-up subscribers.

The pertinent conclusion here is that while a VPN can take many forms, there are some basic common problems that a VPN is built to solve, which can be listed as virtualization of services and segregation of communications to a closed community of interest, while simultaneously exploiting the financial opportunity of economies of scale of the underlying common host communications system.

Ex. 1044 at 16-17; *see also* Ex. 1043 at 2.

259. As I noted above in ¶ 217, the main difference between Patent Owner’s definition and the ordinary meaning of “virtual private network” is that Patent Owner’s definition requires encrypting all of the network traffic. This is not what people in the field of the ’181 patent would have believed was necessary. For example, a secure communication link could be viably established using other security techniques, such as obfuscation or IP hopping schemes.

260. In fact, the ’181 patent itself recognizes that “Data security is **usually** tackled using some form of data encryption.” Ex. 1025 (181 patent) at 1:50-51. The ’181 patent also uses an example of an established “hiding” or “obfuscation” technique that does not use encryption to protect the anonymity of the secure

communication link or VPN. Ex. 1025 (181 patent) at 2:37-60. I believe a person of ordinary skill would conclude that if those non-encryption techniques prevented the discovery or interception of the traffic being sent between the computers, the communication link would be considered “private” and secure, and therefore would be considered a secure communication link.

#### **D. Prior Art References**

261. I understand that a U.S. patent is a formally published document, and that I may rely on the dates in the patent as to when the patent was filed and when it was granted.

262. I understand that for printed publications, questions can arise whether it was publicly disseminated and when that occurred.

263. I believe based on my personal experiences that each of the publications I refer to below was publicly disseminated without any restrictions before April of 2000, and often a year or more before that date.

##### **1. Exhibit 1031 – U.S. Patent 6,496,867 to Beser**

264. Exhibit 1031 (Beser) is a copy of United States Patent No. 6,496,867 to Beser et al.

265. I understand that because Beser was filed on August 27, 1999, several months before the April 2000 effective filing date of the '181 patent, it is prior art to the claims of the '181 patent under 35 U.S.C. § 102(e).



**2. Exhibit 1003 – U.S. Patent 6,557,037 to Provino**

266. Exhibit 1003 (Provino) is a copy of United States Patent No. 6,557,037 to Provino et al.

267. I understand that because Provino was filed on August 27, 1999, several months before the April 2000 effective filing date of the '181 patent, it is prior art the claims of the '181 patent under 35 U.S.C. § 102(e).

**3. Exhibit 1004 – Kiuchi et al., “C-HTTP - The Development of a Secure, Closed HTTP-based Network on the Internet”**

268. “C-HTTP - The Development of a Secure, Closed HTTP-based Network on the Internet,” authored by Takahiro Kiuchi and Shigekoto Kaihara (“Kiuchi”) (Ex. 1004), is a printed publication that was presented at the 1996 Symposium on Network and Distributed Systems Security (SNDSS) on February 22 & 23, 1996, and published by IEEE in the Proceedings of SNDSS 1996. Kiuchi is a printed publication that was distributed publicly without restriction no later than **February 1996**.

269. I understand that because Kiuchi was published and available more than year before the April 2000 effective filing date of the '181 patent, it is prior art the claims of the '181 patent under 35 U.S.C. § 102(b).

**4. Request for Comment (RFC) Publications**

270. There are a number of publications that I discuss in this report that are “Request For Comment” documents or “RFCs.” These publications are prepared

and distributed under a formalized publication process overseen by one of several Internet standards or governing bodies, such as the Internet Engineering Task Force (IETF) or the W3C Consortium.

271. The way IETF RFC publications are prepared and released to the public in a formalized and structured process. In fact, the RFC development and publication process itself is described in an RFC – RFC 2026, dated October 1996. That RFC explains that that RFC publications and “Internet-Drafts” are widely disseminated on the Internet. For example, § 2.1 of RFC 2026 explains:

Each distinct version of an Internet standards-related specification is published as part of the “Request for Comments” (RFC) document series. This archival series is the official publication channel for Internet standards documents and other publications of the IESG, IAB, and Internet community. RFCs can be obtained from a number of Internet hosts using anonymous FTP, gopher, World Wide Web, and other Internet document-retrieval systems.

Ex. 1061 at 5.

272. RFC 2026 explains that once a standard is adopted, it will be formally published. *See* Ex. 1061 at § 6.1.3 (“If a standards action is approved, notification is sent to the RFC Editor and copied to the IETF with instructions to publish the specification as an RFC.”). Draft IETF standards-track and other documents are also formally published and widely distributed. *See also id.* at 7 (“During the development of a specification, draft versions of the document are made available

for informal review and comment by placing them in the IETF's "Internet-Drafts" directory, which is replicated on a number of Internet hosts. This makes an evolving working document readily available to a wide audience, facilitating the process of review and revision.").

273. RFC documents are published on a specific date, which starts a period for others to provide comments on the document. *See* Ex. 1061 at 19-20 ("These minimum periods are intended to ensure adequate opportunity for community review without severely impacting timeliness. These intervals shall be measured from the date of publication of the corresponding RFC(s)..."). The publication date of each RFC is contained in the RFC, typically in the top right corner of the first page of the document. This is the date it was released for public distribution on the Internet.

274. Each RFC document is uniquely numbered. If it becomes necessary to make a substantive change (*e.g.*, other than a minor typographical error), the RFC will be republished with a different number. *See, e.g.*, Ex. 1061 at 19-20 ("... the IESG or RFC Editor may be asked to republish the RFC (with a new number) with corrections...").

275. The formalized process of preparing, publishing and widely distributing RFC documents is a very important part of the Internet culture, which works to develop standards in an open and transparent process. It is also important

to the adoption of these standards, and the stability and functionality of the Internet for developers to adhere to standards and evolving “best practices.”

276. Because RFC documents are formally prepared and published, and are then widely distributed without any restrictions, they are printed publications as I understand that concept in the patent law.

## **V. Patentability Analysis of Claims 1-29 of the '181 Patent**

### **A. U.S. Patent No. 6,496,867 to Beser**

#### **1. Overview of Beser**

277. Beser describes methods and systems for initiating a tunneling association over a public data network such as the Internet. Ex. 1031 (Beser) at 1:8-10 (“The present invention relates to communications in data networks. More specifically, it relates to a method for initiating a tunneling association in a data network.”); *id.* at 3:62-64 (“The data network 10 includes a public network 12 (e.g. the Internet or a campus network) . . . .”); *id.* at 4:5-7 (“the [] devices may be assigned public network addresses on the Internet”); *id.* at 11:12-13.

278. Beser describes a process in which an IP tunnel is securely and transparently established between two network devices with the aid of a third trusted network device on a public network. *See generally* Ex. 1031 (Beser).

279. Beser explains that its methods involve “negotiating private addresses, such as private Internet Address, for the ends of the tunneling association.” Ex.

1031 (Beser) at Abstract; 2:46-50 (“One aspect of the invention includes a method for initiating a tunneling association between an originating end of the tunneling association and a terminating end of the tunneling association.”). This allows a first network device (*e.g.*, one Voice-Over-IP or “VOIP” client) to communicate directly with a second network device (*e.g.*, another VOIP client) on a different private network. For example, Beser explains:

One aspect of the invention includes a method for initiating a tunneling association between an originating end of the tunneling association and a terminating end of the tunneling association. The method includes receiving a request to initiate the tunneling association on a first network device. The first network device is associated with the originating end of the tunneling association, and the request includes a unique identifier for the terminating end of the tunneling association. A trusted-third-party network device is informed of the request on a public network. A public network address for a second network device is associated with the unique identifier for the terminating end of the tunneling association on the trusted-third-party network device. The second network device is associated with the terminating end of the tunneling association. A first private network address on the first network device and a second private network address on the second network device are negotiated through the public network. The first private network address is assigned to the originating end of the tunneling association and the second private network address is assigned to the terminating end of the tunneling association.

Ex. 1031 (Beser) at 2:46-67; *see also* Ex. 1031 (Beser) at 7:62-8:15.

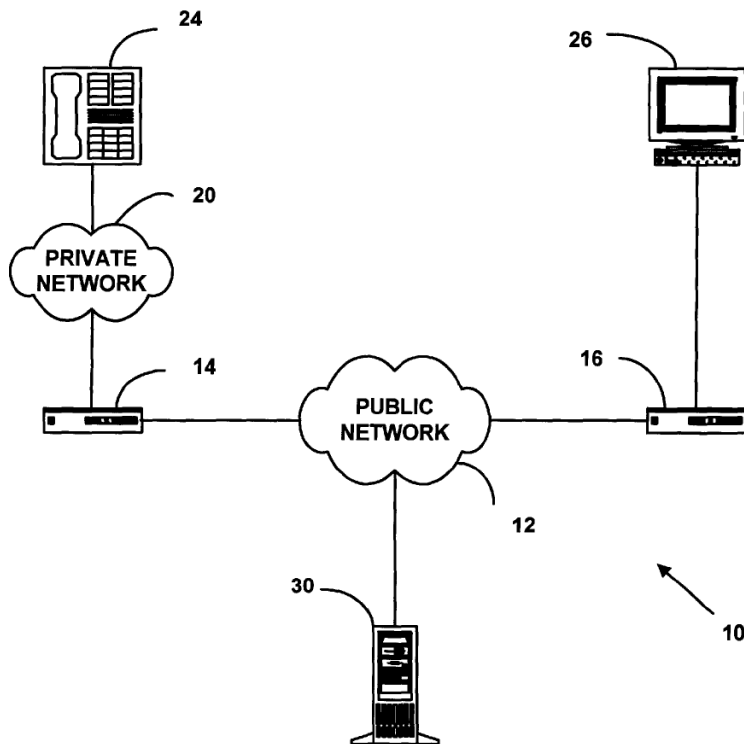
280. Beser explains that negotiation of the tunneling association occurs automatically upon receipt of a request by the trusted-third-party network device. *See, e.g.*, Ex. 1031 (Beser) at Figure 4; *id.* at 11:9-25.

281. Beser shows that the negotiation of the tunneling association starts automatically when a user initiates a request by entering the website destination of a WebTV source or picking up a handset of a VOIP device. *See, e.g.*, Ex. 1031 (Beser) at 4:43-54; 10:22-36 (“**if the originating telephony device 24 is a phone** that is physically connected to the first network device 14 the request may include an electrical signal measured by an interface to an application on the first network device 14 as a result of **the phone going ‘off-hook’**. . . . Alternatively, if the originating **telephony device 24** is a data network device on a local area network or private network 20 which includes the first network device 14, the request may include an IP 58, ICMP 56, or MAC 54 message that indicates the initiation of a VoIP association.”). The negotiation process is also transparent to the user – the user just requests the connection by taking an action.

### **Basic Components of the Beser Scheme**

282. Beser explains that a trusted-third-party network device resident on the public network is used to automatically broker tunneling associations between the first and second network devices. *See, e.g.*, Ex. 1031 (Beser) at 9:26-34 (“...a

first private network address and a second private network address are negotiated on the first network device and the second network device. In one exemplary preferred embodiment, the negotiation is carried out through the trusted-third-party network device...”; *id.* at 12:6-19 (“negotiation may occur through the trusted-third-party network device 30 to further ensure the anonymity of the telephony devices (24, 26).”). Beser shows that the originating end device, terminating end device, first network device, second network device, and trusted-third-party network devices each are separate devices. Ex. 1031 (Beser) at Fig. 1 (reproduced below):



283. Beser explains that “network devices (24, 26) [] are originating and terminating ends of data flow.” Ex. 1031 (Beser) at 4:43-44. Devices 14 and 16 in this example can be “edge routers.” Ex. 1031 (Beser) at 4:19-29.

284. Beser indicates the trusted-third-party network device (30) can be a DNS server. Ex. 1031 (Beser) at 4:9-11 (“The **trusted-third-party** 30 may be a back-end service, a **domain name server**, or the owner/manager of database or directory services. Moreover, the trusted-third-party network device 30 may not be located in one physical location but may be distributed over several locations and the information may be replicated over the several locations.” (emphasis added)).

285. Beser explains that the trusted-third-party network device can store and maintain a database that maps each registered unique identifier to the associated IP address or addresses. For example, Beser explains the trusted-third-party network device can include a directory service for subscribers that associates E.164 numbers with IP addresses. Ex. 1031 (Beser) at 11:45-58 (“For example, the trusted-third-party network device 30 may be a directory service, owned and operated by a telephone company, that retains a list of E.164 numbers of its subscribers. **Associated with a E.164 number in the directory database is the IP 58 address of a particular second network device 16. The database entry may also include a public IP 58 addresses for the terminating telephony device 26.** Many data structures that are known to those skilled in the art are possible for the



association of the unique identifiers and IP 58 addresses for the second network devices 16.” (emphasis added)).

286. Beser explains that its systems are designed to work with standards-compliant processes and techniques. *See, e.g.*, Ex. 1031 (Beser) at 4:55-5:2 (“Network devices and routers for preferred embodiments of the present invention include network devices that can interact with network system 10 based on standards proposed by the Institute of Electrical and Electronic Engineers (‘IEEE’), International Telecommunications Union- 60 Telecommunication Standardization Sector (‘ITU’), Inter- net Engineering Task Force (‘IETF’), or Wireless Application Protocol (‘WAP’) Forum. However, network devices based on other standards could also be used. IEEE standards can be found on the World Wide Web at the Universal 65 Resource Locator (‘URL’) ‘www.ieee.org.’ The ITU, (formerly known as the CCITT) standards can be found at the URL ..www.itu.ch... IETF standards can be found at the URL ‘www.ietf.org.’ The WAP standards can be found at the URL ‘www.wapforum.org.’”); Ex. 1031 (Beser) at 9:64-10:2 (“FIG. 5 is a flow diagram illustrating a Method 110 for initiating a VoIP association between an originating telephony device 24 and a terminating telephony device 26. VoIP is described in ITU standard H.323. As is known in the art, H.323 is a protocol used for multimedia communications including voice.”).

287. Beser further explains its processes could be implemented using any standards-compliant pre-existing (legacy) equipment and systems. Ex. 1031 (Beser) at 4:44-5:2.

288. Although Figure 1 only shows two end devices, Beser shows that its scheme can be used to initiate tunneling associations among multiple different pairs of end devices. Ex. 1031 (Beser) at Figure 17. This is how a person working in the field of the '181 patent would read this type of description.

289. Beser explains that tunneling associations provide security by hiding (obfuscating) the terminating addresses of the sending and receiving computers or devices. This makes the tunneling associations “private” or “anonymous” because they cannot be discovered on the public network. For example, as Beser states:

The negotiation is performed on a public network, such as the Internet, through a trusted-third party without revealing the private addresses. The method provides for **hiding the identity of the originating and terminating ends of the tunneling association from the other users of the public network. Hiding the identities may prevent interception of media flow between the ends of the tunneling association or eavesdropping** on Voice-over-Internet-Protocol calls. The method **increases the security of communication** on the data network without imposing a computational burden on the devices in the data network.

Ex. 1031 (Beser) at Abstract (emphasis added).

290. This explanation is repeated throughout the Beser patent. *See, e.g.*, Ex. 1031 (Beser) at 3:4-9 (“The method and system described herein may help ensure that the addresses of the **ends of the tunneling association are hidden on the public network** and may increase the security of communication without an increased computational burden.”); Ex. 1031 (Beser) at 8:15-20 (“Method 100 may result in the establishment of a virtual tunneling association between the originating end and the terminating end of the tunneling association **without revealing the identities of both ends of the tunneling association on the public network.**” (emphasis added)); Ex. 1031 (Beser) at 12:6-19 (“The negotiation ensures that neither the private nor any public IP 58 addresses for the ends of the VoIP association appear in the source 88 or destination 90 address”).

291. Beser explains that network traffic in an IP tunnel is ordinarily encrypted to increase the security of the tunnel. For example, it explains:

One method of thwarting the hacker is to establish a Virtual Private Network (“VPN”) by initiating a tunneling connection between edge routers on the public network. For example, tunneling packets between two end-points over a public network is accomplished by encapsulating the IP packet to be tunneled within the payload field for another packet that is transmitted on the public network. **The tunneled IP packets, however, may need to be encrypted before the encapsulation in order to hide the source IP address.**

Ex. 1031 (Beser) at 2:6-14.

292. Beser explains that traffic within an IP tunnel is usually encrypted using an industry standard technique called IP Security or “IPsec.” Ex. 1031 (Beser) at 1:54-55 (“Of course, the sender may encrypt the information inside the IP packets before transmission, *e.g.* with IP Security (‘IPsec’)”).

293. This description of use of encryption techniques in conventional IP tunneling approaches is consistent with the guidance in the IPsec standard, RFC 2401 (Ex. 1032). For example, RFC 2401 explains:

IPsec allows the user (or system administrator) to control the granularity at which a security service is offered. For example, **one can create a single encrypted tunnel to carry all the traffic between two security gateways or a separate encrypted tunnel can be created for each TCP connection between each pair of hosts communicating across these gateways.**

Ex. 1032 (RFC 2401) at 7.

294. Beser indicates that terminating network devices can include telephony devices, personal computers, and multimedia devices. As Beser explains:

Multimedia devices include Web-TV sets and decoders, interactive video-game players, or personal computers running multimedia applications. Telephony devices include VoIP devices (portable or stationary) or personal computers running facsimile or audio applications. However, the ends of the data flow may be other types

of network devices and the present invention is not restricted to telephony or multimedia devices.

Ex. 1031 (Beser) at 4:47-50.

295. Beser explains possible end devices include “VoIP devices (portable or stationary) or personal computers running facsimile or audio applications.” Ex. 1031 (Beser) at 4:47-50. A person of ordinary skill in the art would have understood that a “personal computer” would include both stationary desktop computers and portable laptop computers.

296. Beser explains that end devices can include portable VOIP devices. Ex. 1031 (Beser) at 4:47-50. Beser also indicates that data may be sent to and from an end device (*e.g.*, a portable VOIP device) using data transmission standards that were developed by the “Wireless Application Protocol (‘WAP’) Forum.” Ex. 1031 (Beser) at 4:55-62.

297. It was well known that the WAP Forum was an industry association that has developed the de-facto world standard for wireless information and telephony services on digital mobile phones and other wireless terminals. Handset manufacturers representing 95 percent of the world market across all technologies have committed to shipping WAP-enabled devices.

Ex. 1080 (W3C WAP) at 2-3.

298. By April 2000, the WAP Forum had developed a standardized architecture and protocol to simplify the process of building data applications for mobile phones. As the Wireless Application Protocol Specification explains:

The Wireless Application Protocol (WAP) . . . specifies an application framework and network protocols for wireless devices such as **mobile telephones**, pagers, and personal digital assistants (PDAs). **The specifications extend and leverage mobile networking technologies (such as digital data networking standards) and Internet technologies (such as XML, URLs, scripting, and various content formats).** The effort is aimed at enabling operators, manufacturers, and content developers to meet the challenges in building advanced differentiated services and implementations in a fast and flexible manner.

Ex. 1079 (WAP Architecture Spec) at 3.

299. A person of ordinary skill in the art would have understood that the Beser end devices could be mobile phones or other mobile devices that communicate over a cellular network. *See* ¶¶ 295-298, *above*.

300. The data that is to be transmitted through a Beser IP tunnel can represent VOIP traffic, “multimedia” content (*e.g.*, video, audio), or content for web pages (*e.g.*, delivered to WebTV devices or decoders or personal computers).

Ex. 1031 (Beser) at 4:47-50. The data also can represent messages of many different types under different protocols. *See id.* at 7:9-17 (“The payload field 84

of the IP 58 packet 80 typically comprises the data that is sent from one network device to another. However, the payload field 84 may also comprise network management messages, such as ICMP 56 messages, or data packets of another protocol such as UDP 60, SNMP 62, TFTP 64, or DHCP 66.”); *id.* at 6:25-57.

301. The data that can be transmitted through a Beser IP tunnel can represent video conference traffic. Ex. 1031 (Beser) at 9:67-10:2 (“FIG. 5 is a flow diagram illustrating a Method 110 for initiating a VoIP association between an originating telephony device 24 and a terminating telephony device 26. VoIP is described in **ITU standard H.323**. As is known in the art, **H.323 is a protocol used for multimedia communications including voice**.”). In this example, Beser points out the multimedia data can be compliant with the H.323 standard. A person of ordinary skill in the art would have understood that the multimedia communications specified in H.323 may include video conference communications. Ex. 1077 (H.323) at 12 (“H.323 terminals provide audio **and optionally video** and data communications capability in point-to-point or multipoint conferences”).

302. Beser indicates the first and second network device “may be modified routers or modified gateways.” Ex. 1031 (Beser) at 4:7-11. Beser also explains that in a “preferred embodiment,” the first or second network devices are each an “edge router,” which Beser explains “routes data packets between one or more

networks such as a backbone network (*e.g.*, a public network 12) and Local Area Networks (*e.g.*, private network 20).” Ex. 1031 (Beser) at 4:19-24. This is an illustration of my general observation that people working in the field of the ’181 patent can deploy desired functionalities in different ways, such as by consolidating functions in one component, or by distributing those functions across several components of the network.

303. An edge router is a computer, as it has a CPU, memory and storage. *See* Ex. 1031 (Beser) at 5:15-45.

304. The first and second network devices are also called “gateway” computers because they are connected to both a private network and the public Internet. Ex. 1031 (Beser) at 4:7-8 (“The first network device 14 and the second network device 16 may be modified routers or **modified gateways**”); *id.* at 4:19-24.

305. Beser explains the first and second network devices receive and evaluate requests from end devices and route traffic to destinations on the private network. Ex. 1031 (Beser) at 4:55-5:2; *id.* at 8:21-47 (“the first network device receives a request to initiate the tunneling connection” and the request will “indicate[] to the tunneling application that it should examine the request message for its content and not ignore the datagram.”); *id.* at 22:8-22 (“The first network device 14 recognizes that the packet has come from the originating network device



24 and is destined for the terminating network device 26 by determining that the packet includes a private network address for the terminating network device 26.”).

306. Beser explains its systems are deployed on the public Internet, and use conventional domain name navigational techniques. *See* Ex. 1031 (Beser) at 2:43-68; *id.* at 3:62-66. For example, in its VOIP (“Voice-Over-IP”) examples, the Beser methods must employ IP addresses of the origination and destination devices that are communicating with each other via a secure IP tunnel.

307. It was well known before April 2000 that the function of a “DNS server” was to return the IP address associated with a domain name. *See, e.g.*, Ex. 1025 (’181 patent) at 38:54-56 (“Conventional Domain Name Servers (DNSs) provide a look-up function that returns the IP address of a requested computer or host.”). If an address was unknown, it would not be resolved by the DNS server, and an error message would be returned. *See* ¶¶ 105-114, *above*.

308. It was also known before April 2000 that a DNS name server responds to a standard name query by providing a standard response. *See also* Ex. 1036 (RFC 1034) at 21 (“The principal activity of name servers is to answer standard queries. Both the query and its response are carried in a standard message format which is described in [RFC-1035]. The query contains a QTYPE, QCLASS, and QNAME, which describe the types and classes of desired information and the name of interest.”).

309. Beser explains that the trusted-third-party network device “may be distributed over several physical locations.” Ex. 1031 (Beser) at 11:32-36. This means that the functions that are to be performed by the “trusted-third-party” network device do not have to be performed by one device or computer, but can be distributed among several computers or devices, such as a server system.

310. Transparent proxy servers that intercept and evaluate DNS requests were known in the prior art. *See, e.g.*, Ex. 1073 (Blum) at Figures 1 and 4; *see also* Ex. 1073 (Blum) at 3:19-26 (“A request for communication or communications request as the terms are used herein may be a connection request directed to a particular server, either local or remote, as identified by a server name or Internet Protocol (IP) address, or the communications request may be an address resolution request such as a Domain Name Services (DNS) request to determine an IP address from a given server name provided in a Uniform Resource Locator (URL) for example.”); Ex. 1039 (NT for Dummies) at 13-14 (“Firewalls and proxy servers are networking tools that are little more than special-purpose routers. A firewall may be used to filter traffic, both inbound and outbound. Firewall filters can be based on source or destination address, a specific protocol, or port address . . . A proxy server is an enhanced firewall, and its primary purpose is to manage communications between an in-house network and external networks such as the

Internet. Proxies hide the identity of internal clients and can keep local copies of resources that are accessed frequently (this is called *caching* . . .).”).

311. Beser states that IP tunneling techniques were well known, and were used to create “virtual tunnels” between nodes:

One service that can be performed in the application layer is that of initiating and maintaining a virtual tunnel. Virtual tunneling techniques are known to those skilled in the art. Examples of virtual tunneling include Internet Encapsulation Protocol tunneling, Generic Route Encapsulation (“GRE”), and IP in IP tunneling. All three tunneling techniques are based on encapsulating a data packet of one protocol inside a data packet of another protocol.

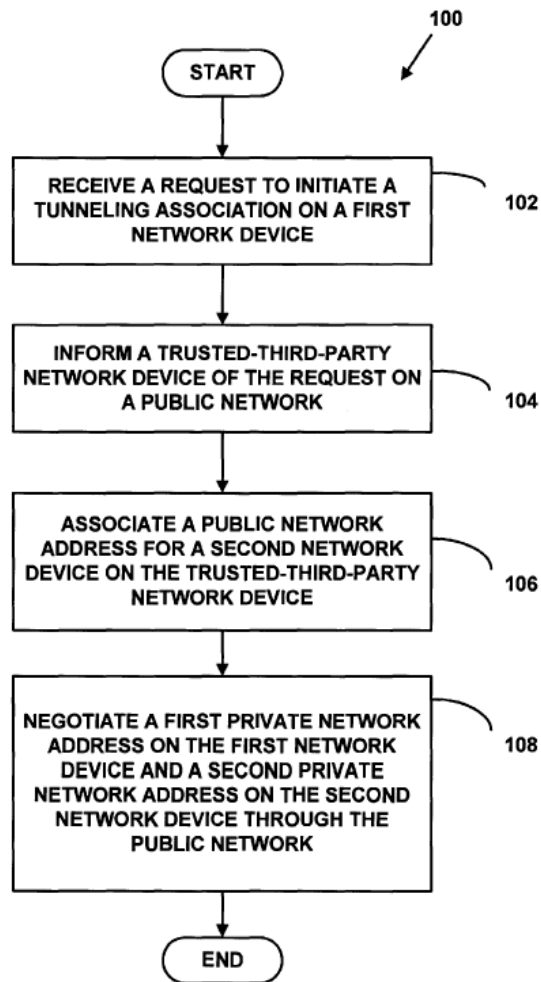
Ex. 1031 (Beser) at 6:58-7:5. This is indicating that the types of tunneling techniques that the Beser method is suitable for implementing are those that establish “virtual” tunnels.

312. Virtual IP “tunnels” can be thought of as being constituent elements of a virtual private network.

### **The Process for Initiating a Beser IP Tunnel**

313. Beser generally illustrates how its tunneling-establishment method works in Figure 4, shown below:

**FIG. 4**



314. Beser explains that its processes start with “receiving a request to initiate the tunneling connection on a first network device at step 102.” Ex. 1031 (Beser) at 7:63-66.

315. Beser shows that an originating device, or an application running on the device, will generate the requests. Ex. 1031 (Beser) at 9:64-10:41, steps 102 & 104; *see* Ex. 1031 (Beser) at 8:28.

316. Beser shows that the first network device evaluates all of the data packets passed through it. If a packet contains a request to initiate an IP tunnel, it

will contain an indicator, such as a distinctive sequence of bits, to inform the first network device that the packet needs special processing. Ex. 1031 (Beser) at 8:33-43.

317. If the first network device determines a data packet contains a request to initiate an IP tunnel, it sends the request to a trusted-third-party device. Ex. 1031 (Beser) at 8:21-50 (“... At Step 104 of Method 100, the trusted-third party network device is informed of the request.”). Otherwise it processes the packet normally. *Id.*

318. Beser explains that the “request includes **a unique identifier for the terminating end** of the tunneling association.” Ex. 1031 (Beser) at 8:1-3 (emphasis added). This unique identifier specifies the destination (the “target”) and not the trusted third party network device or DNS server.

319. Beser explains that requests containing the unique identifier can take many forms. Examples given in Beser for forms of the unique identifier include “any of **a dial-up number**, an electronic mail address or **a domain name**.” Ex. 1031 (Beser) at 10:37-42; *see also id.* at 10:55-11:2; *id.* at 10:60-66 (“[A] company may move premises while still retaining **the same domain name** and it may be more appropriate to **identify the user by the static domain name**. There are many other possibilities for the unique identifier, e.g. employee number, social

security number, driver's license number, or even a previously assigned public IP 58 address.”).

320. Beser thus shows a unique identifier can be a domain name. *Id.* So, for example, a web browser that made a request for a website using a domain name (*i.e.*, a DNS request) would be a request in the meaning of the Beser process.

321. Beser also shows that a unique identifier could be a **dial-up number** or an unconventional identifier such as an employee number. *Id.* So, for example, a VOIP application that made a request to connect to another party using a telephone number (*i.e.*, a secure name) would be a request in the meaning of the Beser process.

322. Beser shows that an end device can be identified by multiple identifiers. For example, Beser shows a scenario where an end device is associated with a phone number, a domain name, and an email address. Ex. 1031 (Beser) at 10:37-42; *id.* at 10:55-67 (“For example, the user of the terminating telephony device 26 may have moved from one office to another office while still retaining the same electronic mail address. **Rather than identifying the terminating user by the number assigned to a physical device in the office, it may be more appropriate to identify the user by the static electronic mail address.** Similarly, a company may move premises while still retaining the same domain name and **it may be more appropriate to identify the user by the static domain name.**”).

Beser explains that any one of those identifiers can be the unique identifier. *Id.* So in one configuration the phone number is used as a unique identifier that is registered with the trusted-third-party network device, while the domain name simply would be registered with a public DNS server.

323. It was common to associate a computer or device with multiple names, where one name was used for internal communications and the other name used for external communications. For example, such a configuration was a standard option available in Windows 98. *See* ¶¶ 127-133, *above*.

324. Beser explains that the request containing the unique identifier is sent to the trusted-third-party network device, and then, in the next step of the process (step 104 in Figure 4), the trusted third party network device receives the request, evaluates it (including the unique information in the request) and takes action based on that evaluation. *See, e.g.*, Ex. 1031 (Beser) at 11:45-58; *id.* at 17:45-49.

325. Beser explains that the transmission of the request containing the unique identifier “may require encryption or authentication to ensure that the unique identifier cannot be read on the public network.” Ex. 1031 (Beser) at 11:22-25. Beser thus points out that a trusted-third-party network device can be configured to require that requests be encrypted and clients be authenticated before it will process a request. *Id.*

326. In evaluating the request, the trusted-third-party network device compares the unique identifier to a database of entries (*e.g.*, a database of subscribers) to determine whether the unique identifier correlates to an IP address associated with a user or end device. *See, e.g.*, Ex. 1031 (Beser) at 11:45-59; Ex. 1031 (Beser) at 7:67-8:9; *id.* at 17:45-49; *see* ¶ 285, *above*.

327. Beser also explains that as part of the actions taken in response to a request, the trusted-third-party network device may be configured to return the IP address associated with the unique identifier. Ex. 1031 (Beser) at 17:45-63; *see also* Ex. 1031 (Beser) at 21:48-22:22.

328. As I explained, in one example, the trusted-third-party network device can function as a DNS server. *See* ¶ 284, *above*. If the identifier specified a non-secure destination, the trusted-third-party network device would resolve the domain name and return the IP address – this is simply what DNS servers do. *See, e.g.*, Ex. 1025 ('181 patent) at 38:54-56 (“Conventional Domain Name Servers (DNSs) provide a look-up function that returns the IP address of a requested computer or host.”); *see also* ¶¶ 76-77, *above*.

329. If the unique identifier in the request specifies a secure destination (*e.g.*, the unique identifier corresponds to a second network device), the trusted-third-party network device will automatically establish a tunneling association by



negotiating with the first and second network devices. *See, e.g.*, Ex. 1031 (Beser) at Figure 4; *id.* at 11:9-44; *id.* at 11:58-12:19.

330. Beser shows that the trusted-third-party network device negotiates private network addresses to be associated with end devices (*e.g.*, telephony devices in the VOIP example). *See* Ex. 1031 (Beser) at 11:58-62 (“a first private IP 58 address on the first network device 14 and a second private IP address on the second network device are negotiated through the public network.”). The private IP addresses are allocated and established by the trusted-third-party network device. *See, e.g.*, Ex. 1031 (Beser) at 11:9-24; 11:33-37; 11:45-58; 12:5-19.

331. As part of the negotiation process, the third-party-network device transmits some of the information from the DNS request to the second network device. *See* Ex. 1031 (Beser) at 11:9-21. The trusted-third-party network device will send the second network device a message containing the private IP address of the originating device and the public IP address of the first network device. Ex. 1031 (Beser) at 13:30-41 (“The trusted-third-party network device 30 constructs a second IP 58 packet 164 with the public IP 58 address of the trusted-third-party network device 30 in the source address field 88 and the public IP 58 address of the second network device 16 in the destination address field 90. Included in the payload field 84 of the second IP 58 packet 164 are **the first private IP 58 address** and **the public IP 58 address of the first network device 14**. The second

IP 58 packet 164 is sent to the second network device 16 on the public network 12.”).

332. After it receives the message containing the private IP address, the second network device will select a second private IP address that will be associated with the terminating device. Ex. 1031 (Beser) at 13:49-64 (“The second private IP 58 address is selected to be a different address than the first private IP 58 address . . . . The selected second private IP 58 address is assigned to the terminating end of the tunneling association 26 on the second network device 16.”); *see* 13:41-47.

333. The second network device will transmit the second private IP address back to the trusted-third-party network device which will pass the address along to the first network device. Ex. 1031 (Beser) at 13:66-14:2 (“At Step 158, the second private IP 58 address is communicated from the second network device 16 to the first network device 14 through the trusted-third-party network device 30 on the public network 12.”).

334. After the private IP addresses have been negotiated, the first network device may inform the originating end device of the private IP addresses. Ex. 1031 (Beser) at 21:48-55 (“[t]he assignment of private network addresses to the ends of the tunneling association may also include transmitting the private network addresses to the network devices at the ends of the tunneling association . . . .”).

Similarly, the second network device may inform the terminating end device of the private IP addresses. *Id.*

335. Receipt of a private IP address would inform the end device that the trusted-third-party network device had established an IP tunnel.

336. Beser explains that the negotiation process combined with the placement of the first and second network devices ensures that the tunnel would remain private and only the IP addresses of the first and second network device would be observable on the public network. The Beser scheme thus hides the IP addresses of the source and destination devices by modifying each IP packet's source and destination fields when transmitted across the public network. For example, it explains:

The negotiation ensures that neither the private nor any public IP 58 addresses for the ends of the VoIP association appear in the source 88 or destination 90 address fields of the IP 58 packets that comprise the negotiation. The IP 58 packets of the negotiation step 118 will only have source 88 or destination 90 address fields containing the IP 58 addresses of the first 14, second 16, or trusted-third-party 30 network device. **In this manner the identities of the originating 24 and terminating 26 telephony devices are inside the payload fields 84 of the IP 58 packets and may be hidden from hackers on the public network 12.** The negotiation may occur through the trusted-third-party network device 30 to further ensure the anonymity of the telephony devices (24, 26).

Ex. 1031 (Beser) at 12:6-19.

337. The private network IP addresses of the end devices are used in conjunction with the public IP addresses of the first and second network devices to send data through the IP tunnel. Ex. 1031 (Beser) at 12:28-37. On packets to be sent between the end devices, the first and second network devices will modify the headers for transmission on the public network. Ex. 1031 (Beser) at 12:28-37; *id.* at 21:63-22:28.

338. After the tunneling association is established, data may be securely transmitted between the end devices. *E.g.*, Ex. 1031 (Beser) at 23:15-29.

### **Features of the Beser Process**

339. Beser explains that network devices maintain network address tables to allow the translation of public and private IP addresses. Ex. 1031 (Beser) at 21:63-22: 28; *id.* at Tables 1-4c. The addresses are translated by comparing the addresses in the IP packet's header to a table associating private IP addresses with public IP addresses. *Id.* at 22:8-22 ("The **first network device** 14 examines the entry in its network address table that contains the private network address for the terminating network device 26 and determines that this private network address is associated with the public network address for the second network device 16. In this manner, the first network device 14 knows where to route the packet on the public network 12 by translating the private network address for the terminating

network device 26 to the public network address for the second network device 16.”); *id.* at 22:23-47 (“For example, **the second network device 16** receives a packet from the public network 12 and recognizes it, from an indicator included in the packet or otherwise, that the payload should be passed up to a higher layer of the protocol stack 50 for examination. Upon examination, the second network device 16 recognizes that the packet is destined for the terminating end of the tunneling association 26 by determining that the packet includes the private network address for the terminating end 26.”).

340. Beser explains that the transmission of the request containing the unique identifier “may require **encryption** or **authentication** to ensure that the unique identifier cannot be read on the public network.” Ex. 1031 (Beser) at 11:22-25. This is explaining that authentication with the trusted-third-party device must precede delivery and evaluation of the request by the trusted-third-party network device.

341. Although Beser does not specifically describe the authentication process, many authentication techniques were well known in April of 2000. *See* ¶¶149-151, *above*. The decision of which authentication technique to use would have been a simple design choice.

342. As explained above, Beser specifies that encryption ordinarily should be used to protect IP tunnels. *See, e.g.*, Ex. 1031 (Beser) at 1:54-58. Beser also

specifically states that queries and other communications involving the unique identifier (*e.g.*, a domain name or phone number) may be encrypted. Ex. 1031 (Beser) at 11:22-25 (“The IP 58 packets may require **encryption** or authentication to ensure that the unique identifier cannot be read on the public network 12.”), 17:50-52 (“With reference to FIG. 14, the trusted-third-party network device 30 constructs a first **IP 58 packet 232...**”), 18:2-5 (“The first **IP 58 packet 232 may require encryption** or authentication to ensure that the public IP 58 address of the second network device 16 cannot be read on the public network 12.”).

343. I believe a person of ordinary skill in the art would have read Beser in April of 2000 as saying that, with certain limited exceptions, IP tunneling methods ordinarily will encrypt all traffic sent between the nodes of the tunnel.

344. For example, the way Beser refers to use of encryption in IP tunneling schemes indicates it is a **conventional technique that is ordinarily used**. Ex. 1031 (Beser) at 1:54-56 (“Of course, the sender may encrypt the information inside the IP packets before transmission, *e.g.*, with IP Security (‘IPsec’).”). In fact, Beser points out that encryption is normally used in IP tunneling schemes, such as to hide the source IP address. Ex. 1031 (Beser) at 2:12-14 (“The tunneled IP packets, however, **may need to be encrypted before the encapsulation in order to hide the source IP address.**”).

345. Beser also cautions against implementing certain tunneling techniques because they can “cause[] security problems by preventing certain types of encryption from being used.” Ex. 1031 (Beser) at 2:22-27. This is a clear statement that the ordinary thing to do is to encrypt the traffic in an IP tunnel.

346. Beser also refers to RFC 2401 (Ex. 1032) describing the IPsec protocol to explain how encryption is conventionally incorporated into IP tunneling schemes. Ex. 1031 (Beser) at 1:54-56. A person of ordinary skill in the art would immediately recognize that, under the IPsec protocol, communications within an IP tunnel (*i.e.*, between a first and second network device) to initiate the tunnel and following establishment of the IP tunnel will be encrypted. *See* Ex. 1031 (Beser) at 1:56-2:15.

347. Beser identifies some practical concerns related to transmission of high volumes of data (*e.g.*, for VOIP and multimedia settings) over an IP tunnel. Beser indicates in these applications, a person might choose to not use encryption in an IP tunnel for the network traffic being sent through the tunnel. As Beser states:

Moreover, encryption at the source and decryption at the destination **may** be infeasible for certain data formats. For example, streaming data flows, such as multimedia or Voice-over-Internet-Protocol (“VoIP”), **may** require a great deal of computing power to encrypt or decrypt the IP packets on the fly. The increased strain on computer

power **may** result in jitter, delay, or the loss of some packets. The expense of added computer power **might** also dampen the customer's desire to invest in VoIP equipment.

Ex. 1031 (Beser) at 1:56-67.

348. These practical concerns are plainly linked to two particular types of high volume data transfer situations (*i.e.*, a where because of the high volume of data being transferred, the equipment being used to establish the IP tunnel and carry its traffic may not be able to handle the volume of the traffic).

349. A person of ordinary skill would recognize from the way these implementations are being discussed that the concerns are both limited and could be easily addressed.

350. For example, the person would immediately recognize that the Beser IP tunneling scheme would work perfectly fine in settings other than these two high data volume scenarios. For example, data communications other than those carrying video data would be recognized to not present these concerns.

351. Even in these two high data volume applications, Beser would be read by a person of ordinary skill as saying that encryption ordinarily should be used.

Ex. 1031 (Beser) at 2:7-13. Beser is just saying that it *may* not be possible to encrypt every packet and maintain transmission quality due to computer power limitation (*e.g.*, during times of high network traffic). Ex. 1031 (Beser) at 2:7-13. Based on the way Beser is describing these high volume data transfer situations, I



believe a person of ordinary skill would have read the Beser patent as indicating that encryption should be used in its IP tunneling systems other than in specific resource constrained situations.

352. The person of ordinary skill also would recognize from the discussion in Beser that the concerns about encrypting high volumes of network traffic can be easily resolved – just use more powerful equipment that can handle the encryption/decryption processing demands. Or, alternatively, configure the systems to transmit the video or multimedia content using less data (*e.g.*, by using lower resolution video) that is compatible with the processing capabilities of the equipment being used. These are obvious, commonsense solutions to the issues that Beser flags as being a *potential* concern with these two types of implementations.

353. It is also clear that Beser does use encryption in establishing its IP tunnels – it explains that queries involving the unique identifier (*e.g.*, a domain name or phone number) may be encrypted. Ex. 1031 (Beser) at 11:22-25 (“The IP 58 packets may require encryption or authentication to ensure that the unique identifier cannot be read on the public network 12.”). Beser thus is saying that encryption should be used during establishment of its secure IP tunnels.

354. Beser explains that its processes are flexible and many different configurations of systems implementing its processes are possible. Ex. 1031

(Beser) at 5:3-14. I believe that a person of ordinary skill in the art would have found it obvious to modify Beser to incorporate other features that were well known in the art.

355. Further, a person of ordinary skill in the art would also recognize that Beser is describing systems that are to be used in combination with legacy techniques. Ex. 1031 (Beser) at 3:4-9 (“The method and system described herein **may help ensure** that the addresses of the ends of the tunneling association are hidden on the public network and **may increase the security** of communications without an increased computational burden”).

356. Beser identifies certain limitations of existing security techniques, such as VPNs, and points out that its scheme can be used to enhance those known techniques. Ex. 1031 (Beser) at 2:1-13. For example, Beser states that its technique is useful for connecting devices on separate Local Area Networks via a tunnel over the Internet. Ex. 1031 (Beser) at 4:5-29. Similarly, Beser explains its technique is useful in the business setting for identifying employees in different office locations and establishing communications across offices. Ex. 1031 (Beser) at 10:55-67 (“For example, the user of the terminating telephony device 26 may have moved from one office to another office while still retaining the same electronic mail address. Rather than identifying the terminating user by the number assigned to a physical device in the office, it may be more appropriate to identify

the user by the static electronic mail address. Similarly, a company may move premises while still retaining the same domain name and it may be more appropriate to identify the user by the static domain name.”).

357. Because Beser is describing a process that can provide additional security on top of traditional secure communications processes, a person of ordinary skill in the art would have thought it obvious and common sense to use the Beser schemes with existing VPN systems, such as the systems described in RFC 2401 or Hoke. See ¶¶ 152-160, 171-176, *above*.

358. Beser, RFC 2401, and Hoke all use standards-based DNS and other Internet communication protocols.

359. Based on the closely analogous methods and applications of the systems and processes described in Beser, RFC 2401, and Hoke, I believe a person of ordinary skill would have considered modifying the Beser scheme to use features of the schemes described in either RFC 2401 or Hoke.

360. For example, a person of ordinary skill in the art would have been motivated to consider adapting these systems shown in Beser to work with the schemes shown in RFC 2401 because Beser explicitly directs that person to the IPsec standard (which is defined in RFC 2401). See ¶¶ 292-293, 356-359, *above*. Also, as I explain in more detail below, both Beser and RFC 2401 are both describing IP tunneling techniques and systems. See ¶¶ 281-282, *above* and

¶¶ 369, 371, *below*. Below, I provide a more detailed description of the systems and processes disclosed by RFC 2401.

## 2. Overview of RFC 2401

361. RFC 2401 describes the IPsec protocol promulgated by the Internet Engineering Task Force (IETF). The IPsec protocol was developed through extensive deliberations, and reflects well-known and accepted principles for designing security mechanisms applicable to network communications. As RFC 2401 explains:

This memo specifies the base architecture for IPsec compliant systems. The goal of the architecture is to provide various security services for traffic at the IP layer, in both the IPv4 and IPv6 environments. This document describes the goals of such systems, their components and how they fit together with each other and into the IP environment. It also describes the security services offered by the IPsec protocols, and how these services can be employed in the IP environment.

Ex. 1032 (RFC 2401) at 3.

362. RFC 2401 explains that:

IPsec provides security services at the IP layer by enabling a system to select required security protocols, determine the algorithm(s) to use for the service(s), and put in place any cryptographic keys required to provide the requested services. **IPsec can be used to protect one or more “paths” between a pair of hosts, between a pair of security**

**gateways, or between a security gateway and a host.** (The term “security gateway” is used throughout the IPsec documents to refer to an intermediate system that implements IPsec protocols. For example, a router or a firewall implementing IPsec is a security gateway.)

Ex. 1032 (RFC 2401) at 5.

363. RFC 2401 also explains that IPsec can be implemented in several manners to provide secure communications between nodes:

- a. Integration of IPsec into the native IP implementation. This requires access to the IP source code and is applicable to both hosts and security gateways.
- b. “Bump-in-the-stack” (BITS) implementations, where IPsec is implemented “underneath” an existing implementation of an IP protocol stack, between the native IP and the local network drivers. Source code access for the IP stack is not required in this context, making this implementation approach appropriate for use with legacy systems. **This approach, when it is adopted, is usually employed in hosts.**
- c. The use of an outboard crypto processor is a common design feature of network security systems used by the military, and of some commercial systems as well. It is sometimes referred to as a “Bump-in-the-wire” (BITW) implementation. **Such implementations may be designed to serve either a host or a gateway (or both).** Usually the BITW device is IP addressable. When supporting a single host, it

may be quite analogous to a BITS implementation, **but in supporting a router or firewall, it must operate like a security gateway.**

Ex. 1032 (RFC 2401) at 8. In other words, RFC 2401 is explaining that IPsec can be implemented into existing schemes for security, such as via edge routers or gateway network devices.

364. RFC 2401 further explains that IPsec can be implemented in “transport” or “tunnel” models. In the latter, RFC 2401 explains:

A tunnel mode SA is essentially an SA [security association] applied to an IP tunnel. Whenever either end of a security association is a security gateway, the SA **MUST** be tunnel mode. Thus an SA between two security gateways is always a tunnel mode SA, as is an SA between a host and a security gateway. Note that for the case where traffic is destined for a security gateway, *e.g.*, SNMP commands, the security gateway is acting as a host and transport mode is allowed. But in that case, the security gateway is not acting as a gateway, *i.e.*, not transiting traffic. Two hosts **MAY** establish a tunnel mode SA between themselves. The requirement for any (transit traffic) SA involving a security gateway to be a tunnel SA arises due to the need to avoid potential problems with regard to fragmentation and reassembly of IPsec packets, and in circumstances where multiple paths (*e.g.*, via different security gateways) exist to the same destination behind the security gateways.

For a tunnel mode SA, there is an “outer” IP header that specifies the IPsec processing destination, plus an “inner” IP header that specifies the (apparently) ultimate destination for the packet. The security

protocol header appears after the outer IP header, and before the inner IP header. If AH is employed in tunnel mode, portions of the outer IP header are afforded protection (as above), as well as all of the tunneled IP packet (*i.e.*, all of the inner IP header is protected, as well as higher layer protocols). If ESP is employed, the protection is afforded only to the tunneled packet, not to the outer header.

Ex. 1032 (RFC 2401) at 9-10. RFC 2401 thus is explaining that IPsec can be readily implemented to protect a tunnel established between two nodes on a network using the IP addresses of the two nodes.

365. For example, as RFC 2401 explains, the IPsec protocol calls for encryption of all IP traffic being sent between nodes of the VPN network, and that encryption happens automatically. RFC 2401 shows that, in the IPsec protocol, outbound and inbound IP packets are examined and afforded the specified protection based on the IP and transport layer header information (*e.g.*, the outbound packet is analyzed and encrypted according to a specified method) and that the encryption and tunneling mechanisms of IPsec work automatically. See generally *id.* at 13 (describing handling of inbound and outbound IPsec traffic); *Id.* at 29-34 (describing the protocols for handling outbound and inbound IP packets).

366. RFC 2401 explains that IPsec automatically evaluates IP packets to determine whether encryption and authentication is required. Specifically, IP packets are to be evaluated as follows:

In a security gateway or BITW implementation (and in many BITS implementations), **each outbound packet is compared against the SPD to determine what processing is required for the packet.** If the packet is to be discarded, this is an auditable event. If the traffic is allowed to bypass IPsec processing, the packet continues through “normal” processing for the environment in which the IPsec processing is taking place. If IPsec processing is required, the packet is either mapped to an existing SA (or SA bundle), or a new SA (or SA bundle) is created for the packet. Since a packet's selectors might match multiple policies or multiple extant SAs and since the SPD is ordered, but the SAD is not, IPsec MUST:

1. Match the packet's selector fields against the outbound policies in the SPD to locate the first appropriate policy, which will point to zero or more SA bundles in the SAD.
2. Match the packet's selector fields against those in the SA bundles found in (1) to locate the first SA bundle that matches. If no SAs were found or none match, create an appropriate SA bundle and link the SPD entry to the SAD entry. If no key management entity is found, drop the packet.
3. Use the SA bundle found/created in (2) to do the required IPsec processing, *e.g.*, authenticate and encrypt.

In a host IPsec implementation based on sockets, the SPD will be consulted whenever a new socket is created, to determine what, if any, IPsec processing will be applied to the traffic that will flow on that socket.



Ex. 1032 (RFC 2401) at 32.

367. Under the IPsec scheme, the IP packet is evaluated as it is leaving the starting node. Ex. 1032 (RFC 2401) at 32. If a policy dictates that the packet requires IPsec handling, then authentication and encryption are applied automatically to create the IPsec relationship with the destination. *Id.* at 6, 32. If the applicable policy does not require IPsec handling, then the packet is passed through for normal handling by the involved network devices. *Id.* at 32. This handling is automatic, and does not require user intervention, other than as needed to satisfy authentication requirements. *Id.*; *id.* at 4-5.

368. RFC 2401 provides specific guidance on using IPsec to establish VPNs. For example, RFC 2401 observes that “For example, a company could create a Virtual Private Network (VPN) using IPsec in security gateways at several sites.” *Id.* at 26. RFC 2401 then provides specific models for implementing VPNs using IPsec:

#### 4.5 Basic Combinations of Security Associations

This section describes four examples of combinations of security associations that **MUST** be supported by compliant IPsec hosts or security gateways. Additional combinations of AH and/or ESP in tunnel and/or transport modes **MAY** be supported at the discretion of the implementor. Compliant implementations **MUST** be capable of generating these four combinations and on receipt, of processing them, but **SHOULD** be able to receive and process any combination.

The diagrams and text below describe the basic cases. The legend for the diagrams is:

===== = one or more security associations (AH or ESP, transport or tunnel)

---- = connectivity (or if so labeled, administrative boundary)

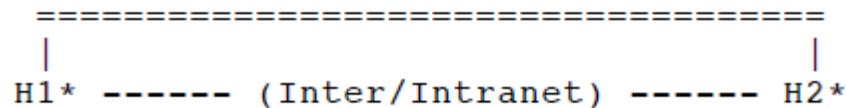
Hx = host x

SGx = security gateway x

X\* = X supports IPsec

NOTE: The security associations below can be either AH or ESP. The mode (tunnel vs transport) is determined by the nature of the endpoints. For host-to-host SAs, the mode can be either transport or tunnel.

Case 1. The case of providing end-to-end security between 2 hosts across the Internet (or an Intranet).

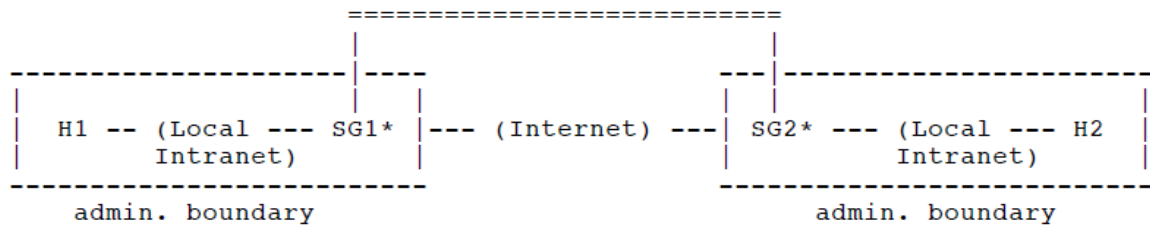


Note that either transport or tunnel mode can be selected by the hosts. So the headers in a packet between H1 and H2 could look like any of the following:

| Transport                | Tunnel                    |
|--------------------------|---------------------------|
| 1. [IP1][AH][upper]      | 4. [IP2][AH][IP1][upper]  |
| 2. [IP1][ESP][upper]     | 5. [IP2][ESP][IP1][upper] |
| 3. [IP1][AH][ESP][upper] |                           |

Note that there is no requirement to support general nesting, but in transport mode, both AH and ESP can be applied to the packet. In this event, the SA establishment procedure MUST ensure that first ESP, then AH are applied to the packet.

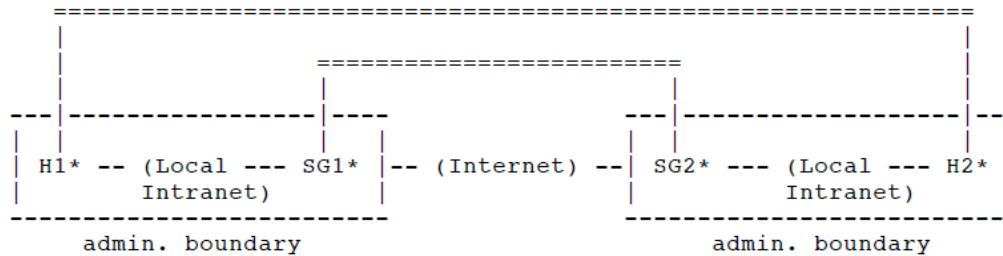
Case 2. This case illustrates simple virtual private networks support.



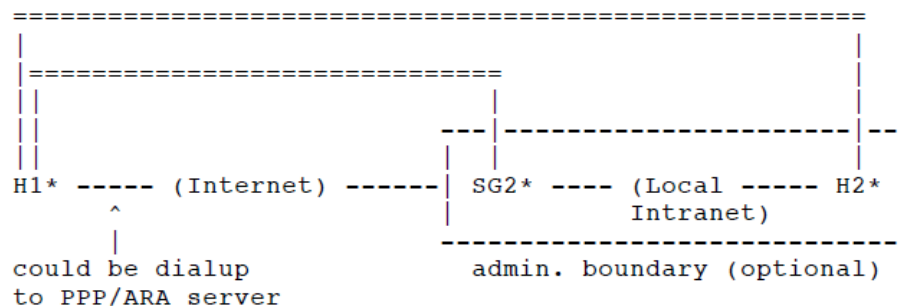
Only tunnel mode is required here. So the headers in a packet between SG1 and SG2 could look like either of the following:

| Tunnel                    |
|---------------------------|
| 4. [IP2][AH][IP1][upper]  |
| 5. [IP2][ESP][IP1][upper] |

Case 3. This case combines cases 1 and 2, adding end-to-end security between the sending and receiving hosts. It imposes no new requirements on the hosts or security gateways, other than a requirement for a security gateway to be configurable to pass IPsec traffic (including ISAKMP traffic) for hosts behind it.



Case 4. This covers the situation where a remote host (H1) uses the Internet to reach an organization's firewall (SG2) and to then gain access to some server or other machine (H2). The remote host could be a mobile host (H1) dialing up to a local PPP/ARA server (not shown) on the Internet and then crossing the Internet to the home organization's firewall (SG2), etc. The details of support for this case, (how H1 locates SG2, authenticates it, and verifies its authorization to represent H2) are discussed in Section 4.6.3, "Locating a Security Gateway".



Only tunnel mode is required between H1 and SG2. So the choices for the SA between H1 and SG2 would be one of the ones in case 2. The choices for the SA between H1 and H2 would be one of the ones in case 1.

Note that in this case, the sender MUST apply the transport header before the tunnel header. Therefore the management interface to the

IPsec implementation **MUST** support configuration of the SPD and SAD to ensure this ordering of IPsec header application.

As noted above, support for additional combinations of AH and ESP is optional. Use of other, optional combinations may adversely affect interoperability.

Ex. 1032 (RFC 2401) at 25-27.

369. I note that in its “Case 3” VPN implementation, RFC 2401 is suggesting use of a design which uses edge routers on two different networks to establish the encrypted IP tunnel through which the network devices (*i.e.*, the “first” and “second” network devices of Beser) behind those routers will communicate. This is the precise network design shown in Figure 1 of Beser (*i.e.*, in an IP tunneling deployments in which edge routers are connected to private networks, and where the edge routers communicate over an insecure pathway such as the Internet). *Compare* Figure 1 of Beser and Case 3 of RFC 2401, at 26-27.

370. RFC 2401 also shows that the version of IPsec being described in RFC 2401 employs a highly flexible design that allows users to select a wide variety of encryption techniques that are suitable for different types of IP applications. In particular, RFC 2401 explains:

IPsec is designed to provide interoperable, high quality, cryptographically-based security for IPv4 and IPv6. The set of security services offered includes access control, connectionless integrity, data origin authentication, protection against replays (a form

of partial sequence integrity), confidentiality (encryption), and limited traffic flow confidentiality. These services are provided at the IP layer, offering protection for IP and/or upper layer protocols.

...

When these mechanisms are correctly implemented and deployed, they ought not to adversely affect users, hosts, and other Internet components that do not employ these security mechanisms for protection of their traffic. These mechanisms also are designed to be algorithm-independent. This modularity permits selection of different sets of algorithms without affecting the other parts of the implementation. For example, different user communities may select different sets of algorithms (creating cliques) if required.

Ex. 1032 (RFC 2401) at 4-5.

371. RFC 2401 thus (i) proposes the precise network design used by Beser, and (ii) explains how to address the practical concerns identified in Beser about encryption of high volumes of IP traffic in certain situations (*e.g.*, in VOIP or multimedia settings). *See, e.g.*, RFC 2401 at 4 (“This modularity permits selection of different sets of algorithms without affecting the other parts of the implementation. For example, different user communities may select different sets of algorithms (creating cliques) if required.”); *Id.* at 6 (“IPsec allows the user (or system administrator) to control the granularity at which a security service is offered. For example, one can create a single encrypted tunnel to carry all the traffic between two security gateways or a separate encrypted tunnel can be created

for each TCP connection between each pair of hosts communicating across these gateways.”); *id.* at 43-44 (IPsec allows for “multi-level secure networking,” which allows different tunnels with different security policies to be used for data of differing sensitivity).

### **3. Comparison of Beser to Claims 1-29 of the '181 Patent**

#### **a. Claim 1**

372. Beser describes methods and systems that create IP tunnels between an originating device and a terminating device using a trusted-third-party network device. IP tunnels are secure tunnels established over the Internet. *See* ¶¶ 277-280, *above*. The Beser IP tunnel provides security and anonymity by hiding the end devices’ public IP addresses when data are transmitted over a public network. *See* ¶¶ 289-291, *above*.

373. Beser explains the originating and terminating devices can be VOIP devices such as phones or personal computers. *See* ¶¶ 294-295, *above*. Beser explains that the originating device (*e.g.*, telephony device 24 in Fig. 1) is associated with a first network device (*e.g.*, edge router 14 in Figure 1) and the terminating network device (*e.g.*, computer 26 in Figure 1) is associated with a second network device (*e.g.*, edge router 16). *See* ¶¶ 282-283, 285, 302-305, 329-339, *above*.

374. Beser explains that the originating and terminating network devices each have a public IP address. *See* ¶¶ 281-282, 285, 306, 339, *above*.

375. Beser explains that the originating and terminating network devices each have a unique identifier. The unique identifier could be one of many different types of identifiers, such as a domain name or a phone number. *See* ¶¶ 318-327, *above*. Beser explains that the “unique identifier is any of a dial-up number, an electronic mail address, or a domain name.” *See* ¶ 319, *above*; Ex. 1031 (Beser) at 10:37-42.

376. Beser shows that the originating and terminating network devices each can be associated with more than one identifier. For example, a terminating network device could be associated with both a phone number and a domain name. *See* ¶¶ 322-323, *above*.

377. During its prosecution of the '181 patent, Patent Owner represented a “‘secure name’ can be a secure non-standard domain name, such as a secure non-standard top-level domain name (e.g., .scom) or a telephone number.” *See* ¶ 212, *above*.

378. Therefore, Beser shows that the terminating device is associated with a secure identifier, such as a phone number. *See* ¶¶ 319-323, *above*.

379. Beser also shows the terminating device is associated with unsecure identifiers, such as a domain name and a public IP address. *Id*.



380. As I explained above (¶¶ 322-323), where a phone number is used as the unique identifier for establishing IP tunnels, the domain name could be used for standard, non-secure communications.

381. In describing the functionality of its systems, Beser defines methods of enabling the IP tunneling functionality that could be implemented on a range of standards-compliant systems. *See* ¶¶ 277, 286, 306, 354, 358, *above*.

382. The Beser systems are implemented using programmed computers and other programmable devices that have storage media containing code that configures how those devices function.

383. Beser thus shows “*A non-transitory machine-readable medium comprising instructions for a method of communicating with a first device associated with a secure name and an unsecured name,*” as specified by claim 1.

384. Beser explains that, to initiate an IP tunnel, an originating device will send out a request to initiate a tunneling association with a terminating device. Beser explains that the request will contain a unique identifier associated with the terminating device. *See* ¶¶ 318-324, *above*. Beser explains that the “request includes a unique identifier” that identifies “the terminating end of the tunneling association.” Ex. 1031 (Beser) at 8:1-3.

385. In the Beser scheme, a request is received by the first network device. Beser explains that the first network device can be a router or other device, and it

will evaluate all IP traffic passing through it. *See* ¶¶ 283, 302-305, 314-315, *above*. If the first network device determines that a packet contains a request for initiating an IP tunnel, it will send the request to the trusted-third-party network device. *See* ¶¶ 302-305, 314-317, *above*. Otherwise, it would process the packet normally. *See* ¶ 317, *above*.

386. Beser explains that the request containing the unique identifier (*e.g.*, phone number) will be received by the trusted-third-party network device. *See* ¶¶ 318-324, *above*. The trusted-third-party network device can be a domain name server, and it also can include a phone number directory service. *See* ¶¶ 282, 284-286, 305-309, 324-331, *above*.

387. Beser explains that after the trusted-third-party network device receives the request, it evaluates the request to determine whether it contains a unique identifier that is associated with a secure terminating device. *See* ¶¶ 324-331, *above*. For example, if the request contains an identifier (*e.g.*, a phone number) that corresponds to an entry in its directory, the trusted-third-party network device determines the unique identifier corresponds to a secure destination (*i.e.*, one that requires negotiation of an IP tunnel). *See* ¶¶ 284-286, 305-309, 324-331, *above*. The trusted-third-party network device will automatically negotiate an IP tunnel between the originating and terminating devices by negotiating private IP addresses. *See* ¶¶ 278, 280-281, *above*; *see also* ¶¶ 326-338, *above*.

388. To negotiate private IP addresses, the trusted-third-party network device will send a message to the second network device requesting to negotiate private IP addresses. *See* ¶¶ 329-335, *above*. The second network device is associated with the terminating device. *See* ¶¶ 282-283, 285, 302-305, 329-339, *above*. Ex. 1031 (Beser) at 13:30-34 (“The trusted-third-party network device 30 has already determined that **the terminating end of this tunneling association 26 is associated with the second network device 16** during the associating Step 106 of Method 100 (FIG. 4).”). *Beser* shows that a message indicating that the originating device desires to communicate securely with the terminating device is received at an address associated with the terminating device.

389. The private IP address for the originating device will be transmitted from the first network device to the second network device through the trusted-third-party network device. *See* ¶¶ 329-335, *above*; Ex. 1031 (Beser) at 12:45-48 (“At Step 144, the first private network address is communicated from the first network device to the second network device through the public network”). This message also is a message indicating that the originating device desires to communicate securely with the terminating device.

390. The private IP address for the terminating device will be transmitted from the second network device to the first network device through the trusted-third-party network device. *See* ¶¶ 329-335, *above*; Ex. 1031 (Beser) at 12:52-54

(“At Step 148, the second private network address is communicated from the second network device to the first network device through the public network.”).

391. Beser thus shows “*receiving, at a network address corresponding to the secure name associated with the first device, a message from a second device of the desired to securely communicate with the first device,*” as specified by claim 1.

392. After the private network addresses are negotiated, the first network device will inform the originating device of the private IP address associated with the terminating device. *See ¶¶ 330-331, above.* The originating device can then securely transmit data to the terminating device by sending IP packets to the private IP address. *See ¶¶ 336-339, above.* Similarly, the terminating device can then securely transmit data to the originating device by sending IP packets to the private IP address. *See ¶¶ 336-339, above.*

393. Beser shows that its IP tunnels allow end devices to directly communicate over a secure and anonymous channel. *See ¶¶ 278-281, 289-293, 336-338, above.*

394. Beser indicates that IP traffic within an IP tunnel ordinarily will be encrypted utilizing the techniques described in RFC 2401 (*i.e.*, under the IPsec protocol). Encryption of the tunneling connection therefore occurs automatically. *See ¶¶ 291-293, 325, 342-353, above.*

395. Beser thus shows “*sending a message over a secure communication link from the first device to the second device*,” as specified by claim 1.

396. Accordingly, Beser anticipates claim 1 of the ’181 patent under 35 U.S.C. § 102(e).

397. If a secure communication link is found to require all network traffic to be encrypted, and it also is found that Beser does not show a tunneling association in which all of the network traffic is encrypted, I believe a person of ordinary skill in April of 2000 would have found it obvious to encrypt all traffic sent over the IP tunnels in the Beser scheme based on the guidance in Beser and RFC 2401. See ¶¶ 361-371, *above*.

398. A person of ordinary skill would have considered Beser with RFC 2401 because Beser specifically refers to the IPsec protocol defined in RFC 2401 as being a conventional security scheme used in establishing IP tunnels. See ¶¶ 292-293, 356-359, *above*. Also, Beser and RFC 2401 both are describing IP tunneling techniques and systems. See ¶¶ 281-282, 369, 371, *above*. Both also are describing methods of transparently creating a tunneling association between an originating device and a terminating device. See ¶¶ 281-282, 361-364, 368-371, *above*.

399. The guidance in RFC 2401 would have specifically suggested to a person of ordinary skill that one should encrypt all of the IP traffic being sent over

a secure IP tunnel. *See* ¶¶ 369-371, *above*. This is also specifically suggested by Beser. *See* ¶¶ 292-293, *above*. A person of ordinary skill would have recognized that it would have been common and routine to integrate IPsec functionality into the edge routers and gateways shown in Beser, and that such a configuration is shown in RFC 2401 in the “case 3” design with network devices on private networks communicating through a tunnel established between edge routers. *See* ¶¶ 282-283, 302-305, 362-371, *above*. It was common practice to add extra security functionality to existing systems by integrating protocols such as IPsec into edge routers and proxy servers. *See* ¶¶ 190-194, *above*.

400. Another obvious design choice that is reflected in the IPsec standard itself is to set the IPsec parameters to accommodate the particular features of the network implementation being used. *See* ¶¶ 370-371, *above*. So, a person of ordinary skill reading Beser and IPsec would have immediately recognized that one could alter the IPsec parameters (*e.g.*, encryption strength) when implementing Beser to accommodate a high data volume implementation when using equipment with limited processing capacity. *See* ¶¶ 293, 371, *above*.

401. A person concerned with the capacity of the equipment in a particular implementation to carry the volume of encrypted traffic also would have considered it an obvious design choice to use more powerful edge routers that can accommodate the higher volume of encrypted network traffic for VOIP or

multimedia applications. The issue flagged in Beser is that with certain high volume implementations, it may be difficult for the equipment to handle the on-the-fly encryption of the VOIP or multimedia data traffic – the issue Beser flags is not linked to what the data represents, but simply the capacity of the devices to encrypt on the fly the volume of data being transferred. *See* ¶¶ 347-354, *above*. Another obvious design choice would be to reduce the quality of the captured signal (*e.g.*, by sampling the voice call at a lower rate or by using a lower video quality), which would reduce the volume of data that must be transferred to be compatible with the capacity of the devices being used in the implementation. These would be practical, common sense solutions to the problem that Beser identifies of limited capacity of the network devices.

**b. Claim 2**

402. As I explained above with respect to claim 1 (¶¶ 372-381), Beser describes methods and systems that create secure and anonymous IP tunnels between an originating device and a terminating device using a trusted-third-party network device. *See* ¶¶ 277-280, 289-291, *above*. The originating device and terminating device can be VOIP devices such as phones or personal computers, and they are associated with a first network device and a second network device, respectively. *See* ¶¶ 282-283, 285, 294-295, 302-305, 329-339, *above*.

403. Beser explains that the originating and terminating network devices each have a public IP address. *See* ¶¶ 281-282, 285, 306, 339, *above*.

404. Beser explains that the originating and terminating network devices each have a unique identifier. The unique identifier could be one of many different types of identifier, such as a domain name or a phone number. *See* ¶¶ 318-327, *above*. Beser explains that the “unique identifier is any of a dial-up number, an electronic mail address, or a domain name.” *See* ¶ 319, *above*; Ex. 1031 (Beser) at 10:37-42.

405. Beser shows that the originating and terminating network devices each can be associated with more than one identifier. For example, a terminating network device could be associated with both a phone number and a domain name. *See* ¶¶ 322-323, *above*.

406. During its prosecution of the '181 patent, Patent Owner represented a “‘secure name’ can be a secure non-standard domain name, such as a secure non-standard top-level domain name (e.g., .scom) or a telephone number.” *See* ¶ 212, *above*.

407. Beser shows that the terminating device is associated with a secure identifier, such as a phone number. *See* ¶¶ 319-323, *above*. Beser also shows the terminating device is associated with unsecure identifiers, such as a domain name and a public IP address. *Id.* As I explained above (¶¶ 322-323), where a phone



number is used as the unique identifier for establishing IP tunnels, the domain name could be used for standard, non-secure communications.

408. In describing the functionality of its systems, Beser defines methods of enabling the IP tunneling functionality that could be implemented on a range of standards-compliant system. *See ¶¶ 277, 286, 306, 354, 358, above.*

409. Beser thus shows “*A method of using a first device to communicate with a second device having a secure name*” as specified by the preamble of claim 2.

410. Beser explains that, to initiate an IP tunnel, an originating device will send out a request to initiate a tunneling association with a terminating device. Beser explains that the request will contain a unique identifier associated with the terminating device. *See ¶¶ 318-324, above.* Beser explains that the “request includes a unique identifier” that identifies “the terminating end of the tunneling association.” Ex. 1031 (Beser) at 8:1-3.

411. The request is received by the first network device. Beser explains that the first network device can be a router or other device, and it will evaluate all IP traffic passing through it. *See ¶¶ 283, 302-305, 314-315, above.* If the first network device determines that a packet contains a request for initiating an IP tunnel, it will send the request to the trusted-third-party network device. *See*

¶¶ 302-305, 314-317, *above*. Otherwise, it would process the packet normally.

*See* ¶ 317, *above*.

412. Beser explains that the request containing the unique identifier (*e.g.*, phone number) will be received by the trusted-third-party network device. *See* ¶¶ 318-324, *above*.

413. Beser explains the trusted-third-party network device can be a standard DNS server. Beser also explains the trusted-third-party network device can provide a directory service that translates non-standard names, such as E.164 phone numbers, into IP addresses. *See* ¶¶ 284-285, 326-331, *above*.

414. The trusted-third-party network device is a secure name service within the meaning of the claims because it can resolve non-standard domain names such as phone numbers.

415. Beser shows that the trusted-third-party network device can require “encryption or authentication.” *See* ¶¶ 291-293, 325, 340-353, *above*. A person of ordinary skill in the art would understand the trusted-third party network device to be a secure name service because it is described as a “trusted” device and it can require users to authenticate before communicating with them.

416. Beser explains that after the trusted-third-party network device receives the request, it evaluates it to determine whether it contains a unique identifier that is associated with a secure terminating device. *See* ¶¶ 324-331,

*above*. For example, if the request contains an identifier (*e.g.*, a phone number) that corresponds to an entry in its directory, the trusted-third-party network device determines the unique identifier corresponds to a secure destination (*i.e.*, one that requires negotiation of an IP tunnel). *See* ¶¶ 284-286, 305-309, 324-331, *above*. The trusted-third-party network device will automatically negotiate an IP tunnel between the originating and terminating devices by negotiating private IP addresses. *See* ¶¶ 278, 280-281, *above*.

417. Beser thus shows “*from the first device, sending a message to a secure name service, the message requesting a network address associated with the secure name of the second device,*” as specified in claim 2.

418. If the trusted-third-party network device determines the unique identifier corresponds to a secure destination (*e.g.*, a phone number listed in its directory), it will automatically create an IP tunnel between the originating and terminating devices by negotiating private IP addresses. *See* ¶¶ 326-338, *above*. *see also* ¶¶ 278, 280-281, *above*.

419. To negotiate private IP addresses, the trusted-third-party network device will send a message to the second network device requesting to negotiate private IP addresses. *See* ¶¶ 329-335, *above*. The second network device is associated with the terminating device. *See* ¶¶ 282-283, 285, 302-305, 329-339, *above*; Ex. 1031 (Beser) at 13:30-34 (“The trusted-third-party network device 30

has already determined that **the terminating end of this tunneling association 26 is associated with the second network device 16** during the associating Step 106 of Method 100 (FIG. 4).”). The trusted-third-party network device also will inform the first network of the second network device’s IP address. *See ¶¶ 330-337, above.* Therefore, the first network device receives a message containing a network address associated with the secure name of the terminating device.

420. The private IP address for the originating device will be transmitted from the first network device to the second network device through the trusted-third-party network device. *See ¶¶ 329-335, above.* Ex. 1031 (Beser) at 12:45-48 (“At Step 144, the first private network address is communicated from the first network device to the second network device through the public network”).

421. The private IP address for the terminating device will be transmitted from the second network device to the first network device through the trusted-third-party network device. *See ¶¶ 329-335, above.* Ex. 1031 (Beser) at 12:52-54 (“At Step 148, the second private network address is communicated from the second network device to the first network device through the public network.”). Therefore, the first network device receives a message containing a network address associated with the secure name of the terminating device.

422. After the private network addresses are negotiated, the first network device will inform the originating device of the private IP address associated with

the terminating device. *See* ¶¶ 330-334, *above*. The originating device can then securely transmit data to the terminating device. *See* ¶¶ 295, 296, *above*.

Therefore, the originating device receives a message containing a network address associated with the secure name of the terminating device.

423. Beser thus shows “*at the first device, receiving a message containing the network address associated with the secure name of the second device,*” as specified in claim 2.

424. Beser shows that the trusted-third-party network device can require “encryption or authentication” of any communications exchanged with it. *See* ¶¶ 291-293, 325, 340-353, *above*. Therefore, during the negotiation process, the first network device sends a message to the address associated with the terminating device using a secure communication link.

425. Beser explains the originating device will transmit data to the terminating device by addressing IP packets to the private IP address of the terminating device. *See* ¶¶ 336-339, *above*. Beser explains the first network device will receive the packet, recognize it is destined for the terminating end device, and transmit it to the second network device. *See* ¶¶ 336-339, *above*. The second network device will route the packet to the terminating device. *See* ¶¶ 336-339, *above*.

426. Beser shows that its IP tunnels allow end devices to directly communicate over a secure and anonymous channel. See ¶¶ 278-281, 289-293, 336-338, *above*. Beser indicates that IP traffic within an IP tunnel ordinarily will be encrypted utilizing the techniques described in RFC 2401 (*i.e.*, under the IPsec protocol). Encryption of the tunneling connection therefore occurs automatically. See ¶¶ 278, 280-281, 291-293, 325, 342-353, *above*.

427. Beser thus shows “*from the first device, sending a message to the network address associated with the secure name of the second device using a secure communication link,*” as specified in claim 2.

428. Accordingly, Beser anticipates claim 2 of the ’181 patent under 35 U.S.C. § 102(e).

429. If a secure communication link is found to require all network traffic to be encrypted, and it also is found that Beser does not describe a tunneling association in which the network traffic is encrypted, I believe a person of ordinary skill in April of 2000 would have found encrypting all traffic to have been obvious based on the guidance in Beser and RFC 2401 for the same reasons as I explained above with respect to claim 1. See ¶¶ 397-401, *above*.

430. I also note that a person of ordinary skill would have considered it a simple design choice to distribute the functions being performed in the Beser scheme in different ways. For example, the lookup function performed by the

trusted third party network device could be distributed among multiple servers or devices. Similarly, one could configure the edge router or equivalent edge device to handle part of the lookup function. These variations would have no impact in the operation of the system – the IP tunnels will be established and will function in the same way.

**c. Claim 3**

431. Beser shows that many different types of unique identifiers can be used to identify the devices available for an IP tunnel. Beser shows the unique identifier can be a domain name. *See ¶¶ 318-327 above.*

432. Beser shows that the trusted third-party network device can be a domain name server. *See ¶¶ 282, 284-286, 305-309, 324-331, above.*

433. Beser shows that after receiving a request, the trusted-third-party network device compares the unique identifier in the request (*e.g.*, a domain name) to an internal database of users or end devices. *See ¶¶ 285, 326 above.*

434. If the unique identifier (*e.g.*, domain name) is one that will cause negotiation of a tunneling association (*i.e.*, if it matched an entry in the database), the trusted-third-party network device will negotiate with the first and second network devices to establish private IP addresses for the end devices. *See ¶¶ 329-336, above.*

435. Beser shows that if a user attempts to connect to certain domain names, those domain names will cause an IP tunnel to be established. *See* ¶¶315-338, *above*. Therefore, those domain names are secure domain names.

436. Beser thus shows “*The method according to claim 2, wherein the secure name of the second device is a secure domain name,*” as specified by claim 3.

437. If it is found that Beser does not describe use of a secure domain name, I believe a person of ordinary skill in April of 2000 would have found using a secure domain name as a unique identifier to have been obvious based on the guidance in Beser and Kiuchi. Kiuchi shows the use of non-standard domain names to identify target computers that are located within a closed network. *See* ¶¶ 979-981, *below*.

438. A person would have considered Beser with Kiuchi because both are describing IP tunneling techniques and systems. *See* ¶¶ 281-282, 369, 371, *above*, ¶¶ 972-977, *below*. Both also are describing methods of transparently creating a tunneling association between an originating device and a terminating device. *See* ¶¶ 281-282, 361-364, 368-371, *above*, ¶¶ 994-998, *below*.

439. The architecture of the Kiuchi system is analogous to the Beser system. For example, the CHTTP name server corresponds to the trusted-third-



party network device, and the client and server side proxies correspond to the first and second network devices, respectively.

440. A person considering the Beser and Kiuchi schemes would recognize that one could readily integrate the Kiuchi secure domain name server functionality within the trusted-third-party network device in the Beser scheme. *See* ¶¶ 286-287, 354-360, *above*, ¶ 998, *below*. Both perform the same basic function of receiving a request containing a non-standard network location identifier (*e.g.*, a telephone number or a non-standard secure domain name), evaluating that non-standard name, and providing corresponding private IP addresses to use to establish a secure communication link (*e.g.*, a VPN such as a secure IP tunnel).

**d. Claim 4**

441. Beser shows that many different types of unique identifiers can be used to identify the devices available for an IP tunnel. Beser shows the unique identifier can be a phone number or another non-standard name. *See* ¶¶ 318-327, *above*.

442. Therefore a unique identifier that was a dial-up number or non-standard name would indicate security.

443. Beser thus shows “*The method according to claim 2, wherein the secure name indicates security,*” as specified by claim 4.

444. If it is found that Beser does not describe use of a secure name that indicates security, I believe a person of ordinary skill in April of 2000 would have found using a such a name as a unique identifier to have been obvious based on the guidance in Beser and Kiuchi. Kiuchi shows the use of non-standard domain names to identify target computers that are located within a closed network. *See* ¶¶ 979-981, *below*. Use of a secure domain name would indicate security. A person would have considered Beser with Kiuchi for the same reasons I explained above with respect to claim 3. *See* ¶¶ 437-440, *above*.

**e. Claim 5**

445. Beser shows systems and processes in which a trusted-third-party network device can be configured to require that communications with it be encrypted. *See* ¶¶ 291-293, 325, 340-353, *above*. Any messages sent from the trusted-third-party network device to first network device would be encrypted. *See* ¶ 342, *above*; Ex. 1031 (Beser) at 17:50-52 (“With reference to FIG. 14, the trusted-third-party network device 30 constructs a first **IP 58 packet 232...**”), 18:2-5 (“The first **IP 58 packet 232 may require encryption** or authentication to ensure that the public IP 58 address of the second network device 16 cannot be read on the public network 12.”).

446. Beser thus shows “*The method according to claim 2, wherein receiving the message containing the network address associated with the secure*

*name of the second device includes receiving the message in encrypted form,” as specified by claim 5.*

**f. Claim 6**

447. Beser shows systems and processes in which a trusted-third-party network device can be configured to require that requests be encrypted and clients be authenticated before it will process a request. *See* ¶¶ 291-293, 325, 340-353, *above*.

448. If the trusted-third-party network device received an encrypted message, it necessarily would decrypt the message before processing it because the unique identifiers are contained in the body of the packets. Ex. 1031 (Beser) at 11:20-25.

449. Similarly, if the first network device received an encrypted message from the trusted-third-party network device, it necessarily would decrypt the message before processing it because the IP addresses are transmitted in the body of the packets. Ex. 1031 (Beser) at 12:6-19; *see* ¶ 340, *above*.

450. Beser thus shows “*The method according to claim 5, further including decrypting the message,*” as specified by claim 6.

**g. Claim 7**

451. Beser shows the terminating device has a private IP address and a public IP address. *See* ¶¶ 322, 336-339, *above*; Ex. 1031 (Beser) at 12:6-13 (“The

negotiation ensures that neither the private nor any public IP 58 addresses for the ends of the VoIP association appear in the source 88 or destination 90 address”).

452. Packets sent to the public IP address of the terminating device would be routed to the terminating device. *See* ¶ 339, *above*.

453. Beser thus shows “*The method according to claim 2, wherein the second device is capable of supporting a secure communication link as well as a non-secure communication link, the method further including establishing a non-secure communication link with the second device when needed,*” as specified by claim 7.

#### **h. Claim 8**

454. Beser shows that the trusted-third-party network device will negotiate private IP addresses for the originating and terminating devices. *See* ¶¶ 329-336, *above*.

455. Beser shows that the first network device receives a message containing the IP address of the second network device. *See* ¶¶ 330-333, *above*.

456. Beser shows that the first network device receives a message containing the private IP address for the terminating device. *See* ¶¶ 333-334, *above*. The first network device then informs the originating device of the private IP address to be used to communicate with the terminating device. *See* ¶¶ 333-334, *above*.

457. Beser thus shows “*The method according to claim 2, wherein receiving a message containing the network address associated with the secure name of the device includes receiving the network address as an IP address associated with the secure name of the device,*” as specified by claim 8.

**i. Claim 9**

458. Where a unique identifier corresponds to a secure terminating device, the trusted-third-party network device will automatically initiate an IP tunnel between the originating and terminating devices by negotiating private IP addresses. *See ¶¶ 329-336, above.*

459. Negotiation of the IP tunnel is transparent to the user, who simply initiates a request by taking an action (*e.g.*, entering the website destination of a WebTV source, picking up a handset of a VOIP device). *See ¶¶ 280-282, above.* They are also automatically established. *See ¶¶ 280-282, 329, above.*

460. Beser indicates that IP traffic within an IP tunnel ordinarily will be encrypted utilizing the techniques described in RFC 2401 (*i.e.*, under the IPsec protocol). Encryption of the tunneling connection therefore occurs automatically. *See ¶¶ 291-293, 325, 342-353, above.*

461. Beser thus shows “*The method according to claim 2, further including automatically initiating the secure communication link after it is enabled,*” as specified by claim 9.

**j. Claims 10-11**

462. Beser shows systems and processes in which communications involving a trusted-third-party network device can be required to be encrypted. *See ¶¶ 291-293, 325, 340-353, above.*

463. Beser indicates that IP traffic can be encrypted utilizing the techniques described in RFC 2401 (*i.e.*, under the IPsec protocol). Communications sent to and from the trusted-third-party network device could be encrypted and sent via an IPsec tunnel. *See ¶¶ 278, 280-281, 291-293, 325, 342-353, above.* Where IPsec was used to secure those communications, the first network device would receive the public and private IP addresses associated with the terminating end device in tunneled packets. *See ¶¶ 363-369, above.*

464. Beser thus shows “*The method according to claim 2, wherein receiving a message containing the network address associated with the secure name of the device includes receiving the message at the first device through tunneling within the secure communication link,*” as specified by claim 10.

465. Beser thus shows “*The method according to claim 2, wherein receiving a message containing the network address associated with the secure name of the device includes receiving the message in the form of at least one tunneled packet,*” as specified by claim 11.

**k. Claim 12**

466. Beser indicates that terminating network devices can include telephony devices, personal computers, and multimedia devices. *See* ¶ 294, *above*.

467. Beser explains that the data that can be transmitted through an IP tunnel can represent VOIP traffic, “multimedia” content (*e.g.*, video, audio), or content for web pages (*e.g.*, delivered to WebTV devices or decoders or personal computers). *See* ¶¶ 295-301, *above*. The data that is transmitted through a Beser IP tunnel also can represent video conference traffic or other multimedia content as specified in H.323. *See* ¶¶ 286, 301, *above*.

468. Data transmitted through a Beser IP tunnel can represent data packets of another protocol such as UDP, TCP, SNMP, or TFTP. *See* ¶ 296, *above*.

469. Beser thus shows “*The method according to claim 2, wherein the receiving and sending of messages includes receiving and sending the messages in accordance with any one of a plurality of communication protocols,*” as specified by claim 12.

### **I. Claim 13**

470. Beser indicates that terminating network devices can include telephony devices, personal computers, and multimedia devices. *See* ¶ 294, *above*.

471. Beser explains that the data that can be transmitted through an IP tunnel can represent VOIP traffic, “multimedia” content (*e.g.*, video, audio), or content for web pages (*e.g.*, delivered to WebTV devices or decoders or personal

computers). *See* ¶¶ 295-301, *above*. The data that is transmitted through a Beser IP tunnel also can represent video conference traffic or other multimedia content as specified in H.323. *See* ¶¶ 286, 301, *above*.

472. When transmitting multimedia content, each media type (*e.g.*, audio and video) would be transmitted in its own session.

473. Beser thus shows “*The method according to claim 2, wherein the receiving and sending of messages through the secure communication link includes multiple sessions,*” as specified by claim 13.

**m. Claims 14-17**

474. Beser indicates that terminating network devices can include telephony devices, personal computers, and multimedia devices. *See* ¶ 294, *above*.

475. Beser explains that the data that can be transmitted through an IP tunnel can represent VOIP traffic, “multimedia” content (*e.g.*, video, audio), or content for web pages (*e.g.*, delivered to WebTV devices or decoders or personal computers). *See* ¶¶ 295-301, *above*.

476. The data that is transmitted through a Beser IP tunnel also can represent video conference traffic or other multimedia content as specified in H.323. *See* ¶¶ 286, 301, *above*.



477. Data transmitted through a Beser IP tunnel can represent data packets of another service or protocol such as UDP, TCP, SNMP, or TFTP. *See* ¶ 296, *above*.

478. Beser thus shows “*The method according to claim 2, further including supporting a plurality of services over the secure communication link,*” as specified by claim 14.

479. Beser also thus shows “*The method according to claim 14, wherein the plurality of services comprises a plurality of communication protocols, a plurality of application programs, multiple sessions, or a combination thereof,*” as specified by claim 15.

480. Beser explains that the data that can be transmitted through an IP tunnel can represent VOIP traffic or “multimedia” content (*e.g.*, video, audio). *See* ¶¶ 295-301, *above*. The data that is transmitted through a Beser IP tunnel also can represent video conference traffic or other multimedia content as specified in H.323. *See* ¶¶ 286, 301, *above*.

481. Beser thus shows “*The method according to claim 15, wherein the plurality of application programs comprises video conferencing, e-mail, a word processing program, telephony or a combination thereof,*” as specified by claim 16.

482. Beser also thus shows “*The method according to claim 15, wherein the plurality of services comprises audio, video or a combination thereof,*” as specified by claim 17.

**n. Claim 18**

483. Beser shows systems and processes in which a trusted-third-party network device can be configured to require that requests be encrypted and clients be authenticated before it will process a request. *See* ¶¶ 291-293, 325, 340-353, *above*.

484. Beser thus shows “*The method according to claim 2, wherein the secure communication link is an authenticated link,*” as specified by claim 18.

485. If it is found that Beser does not describe a tunneling association in which the tunnel is an authenticated link, I believe a person of ordinary skill in April of 2000 would have found authenticating the IP tunnel to have been obvious based on the guidance in Beser and RFC 2401. A person of ordinary skill would have considered Beser and RFC 2401 together for the same reasons as I explained above with respect to claim 1. *See* ¶¶ 397-401, *above*.

486. RFC 2401 shows that IPsec transmissions can be authenticated. Using IPsec to protect the transmissions would have created an authenticated link.

**o. Claim 19**

487. Beser explains the originating device and first network device each can be a computer. *See* ¶¶ 282-283, 294-296, 302-305, *above*.

488. Beser thus shows “*The method according to claim 2, wherein the first device is a computer, and the steps are performed on the computer,*” as specified by claim 19.

**p. Claim 20**

489. Beser explains the originating device can be connected to the first network device. *See* ¶ 281, *above*; Ex. 1031 (Beser) at 10:23-28 (“[I]f the originating telephony device 24 is a phone that is physically connected to the first network device 14 . . .”). Beser explains the first network device can be a computer. *See* ¶¶ 302-305, 282-283, 294-296, *above*.

490. Beser shows that the originating device and first network device are connected to a communication network. *See* ¶¶ 277, 282, *above*.

491. Beser thus shows “*The method according to claim 2, wherein the first device is a client computer connected to a communication network, and the method is performed by the client computer on the communication network,*” as specified by claim 20.

**q. Claim 21**

492. Beser explains that the originating and terminating end devices each have a public IP address. *See* ¶¶ 281-282, 285, 306, 339, *above*.

493. Beser explains that the originating and terminating network devices each have a unique identifier. The unique identifier could be one of many different types of identifier, such as a domain name or a phone number. *See* ¶¶ 318-325, *above*. Beser explains that the “unique identifier is any of a dial-up number, an electronic mail address, or a domain name.” *See* ¶¶ 321, 318-323, *above*.

494. Beser shows that the originating and terminating network devices each can be associated with more than one identifier. For example, a terminating network device could be associated with both a phone number and a domain name. *See* ¶¶ 318-323, *above*.

495. During its prosecution of the '181 patent, Patent Owner represented a “‘secure name’ can be a secure non-standard domain name, such as a secure non-standard top-level domain name (e.g., .scom) or a telephone number.” *See* ¶ 212, *above*.

496. Therefore, Beser shows that the terminating device is associated with a secure identifier, such as a phone number. *See* ¶¶ 319-323, *above*. Beser also shows the terminating device is associated with unsecure identifiers, such as a domain name and a public IP address. *Id.* As I explained above (¶¶ 322-323), where a phone number is used as the unique identifier for establishing IP tunnels, the domain name could be used for standard, non-secure communications.

497. Beser thus shows “*The method according to claim 2, further including providing an unsecured name associated with the device,*” as specified by claim 21.

**r. Claim 22**

498. Beser explains the trusted-third-party network device can be a standard DNS server. Beser also explains the trusted-third-party network device can provide a directory service that translates non-standard names, such as E.164 phone numbers, into IP addresses. *See* ¶¶ 284-285, 326-331, *above*.

499. The trusted-third-party network device’s directory of subscribers can associate each subscriber’s phone number to the IP address of that subscriber’s second network device and the subscriber’s IP address. *See* ¶ 285, *above*; Ex. 1031 (Beser) at 11:45-52 (“the trusted-third-party network device 30 may be a directory service, owned and operated by a telephone company, that retains a list of E.164 numbers of its **subscribers**. Associated with a E.164 number in the directory database is **the IP 58 address of a particular second network device** 16. The database entry may also include **a public IP 58 addresses for the terminating telephony device** 26.”). This association necessarily requires the subscriber to have registered its phone number and IP addresses with the trusted-third-party network device. *See* ¶¶ 306-309, *above*.

500. Beser explains that, by identifying a user with a static unique identifier, the user can still be located even if the user's IP addresses change because, for example, the user has moved from one office to another. Ex. 1031 (Beser) at 10:56-63 (“the user of the terminating telephony device 26 may have moved from one office to another office while still retaining the same electronic mail address. **Rather than identifying the terminating user by the number assigned to a physical device in the office, it may be more appropriate to identify the user by the static electronic mail address.**”), 10:66-11:2 (“There are many other possibilities for the unique identifier, e.g. . . . a **previously assigned public IP 58 address.**”).

501. Beser therefore describes a process where users can change IP addresses. For the trusted-third-party network device to be able to associate a unique identifier with the correct IP addresses, each user must register with the trusted-third-party network device whenever the user's IP addresses change. *See* ¶¶ 127-133, 306-309, *above*.

502. The claim specifies that a secure name is registered. I interpret the claim using the plain and ordinary meaning of the term “register” as it relates to domain names and name services. When an entity registers a domain name with a name server, it specifies the name and one or more addresses to be associated with that name. *See* ¶¶ 91-110, 127-133, *above*.

503. Beser thus shows “*The method according to claim 2, wherein the secure name is registered prior to the step of sending a message to a secure name service,*” as specified by claim 22.

504. If the Board finds that Beser does not adequately describe the step of registering a secure name, I believe a person of ordinary skill in April of 2000 would have found that step to have been obvious based on the guidance in Beser and RFC 2543. RFC 2543 describes the SIP protocol, which is another technique for initiating VOIP calls. See ¶¶ 640-644, 660-662, *above*. Beser discloses that its systems can be used with H.323, which is a call signaling protocol that is analogous to SIP. See ¶¶ 286, 301, *above*; see ¶¶ 182-189, *above*. RFC 2543 describes use of a location server, which allows each user to move between devices and to register its present location so that phone calls always are routed to the correct destination. See ¶¶ 666-672, *above*. A person of ordinary skill in the art would have found it obvious to include the registration functionality in Beser’s trusted-third-party network device as it would allow that device to more efficiently maintain its directory service. It was well-known in the art that it is beneficial to configure a device to update a directory service when its network address changed. See ¶¶ 127-133, *above*. Thus, the combination of Beser and RFC 2543 would be a simple design choice, and it would be no more than uniting known technologies with no change in their respective functions to yield predictable results.

**s. Claim 23**

505. As I explained above with respect to claim 3 (¶¶ 431-435), Beser shows that if a user attempts to connect to certain domain names, those domain names will cause an IP tunnel to be established. *See* ¶¶315-338, *above*. Therefore, those domain names are secure domain names.

506. Beser thus shows “*The method according to claim 2, wherein the secure name of the second device is a secure, non-standard domain name,*” as specified by claim 23.

507. As I explained above (¶¶ 437-440), if it is found that Beser does not describe use of a secure domain name, I believe a person of ordinary skill in April of 2000 would have found using a secure domain name as a unique identifier to have been obvious based on the guidance in Beser and Kiuchi. A person would have considered Beser with Kiuchi because both are describing IP tunneling techniques and systems. *See* ¶¶ 281-282, 369, 371, *above*, ¶¶ 972-977, 979-981, *below*. Both also are describing methods of transparently creating a tunneling association between an originating device and a terminating device. *See* ¶¶ 281-282, 361-364, 368-371, *above*, ¶¶ 994-998, *below*.

**t. Claim 24**

508. As I explained above with respect to claim 1 (¶¶ 372-381), Beser describes methods and systems that create secure and anonymous IP tunnels



between an originating device and a terminating device using a trusted-third-party network device. *See* ¶¶ 277-280, 289-291, *above*. The originating device and terminating device can be VOIP devices such as phones or personal computers, and they are associated with a first network device and a second network device, respectively. *See* ¶¶ 282-283, 285, 294-295, 302-305, 329-339, *above*.

509. Beser explains that the originating and terminating network devices each have a public IP address. *See* ¶¶ 281-282, 285, 306, 339, *above*.

510. Beser explains that the originating and terminating network devices each have a unique identifier. The unique identifier could be one of many different types of identifier, such as a domain name or a phone number. *See* ¶¶ 318-327, *above*. Beser explains that the “unique identifier is any of a dial-up number, an electronic mail address, or a domain name.” *See* ¶ 319, *above*; Ex. 1031 (Beser) at 10:37-42.

511. Beser shows that the originating and terminating network devices each can be associated with more than one identifier. For example, a terminating network device could be associated with both a phone number and a domain name. *See* ¶¶ 322-323, *above*.

512. During its prosecution of the '181 patent, Patent Owner represented a “‘secure name’ can be a secure non-standard domain name, such as a secure non-

standard top-level domain name (e.g., .scom) or a telephone number.” See ¶ 212, *above*.

513. Therefore, Beser shows that the terminating device is associated with a secure identifier, such as a phone number. See ¶¶ 319-323, *above*. Beser also shows the terminating device is associated with unsecure identifiers, such as a domain name and a public IP address. *Id.* As I explained above (¶¶ 322-323), where a phone number is used as the unique identifier for establishing IP tunnels, the domain name could be used for standard, non-secure communications.

514. In describing the functionality of its systems, Beser defines methods of enabling the IP tunneling functionality that could be implemented on a range of standards-compliant system. See ¶¶ 277, 286, 306, 354, 358, *above*.

515. Beser thus shows “*A method of using a first device to securely communicate with a second device over a communication network,*” as specified in the preamble of claim 24.

516. Beser explains the trusted-third-party network device can be a standard DNS server. Beser also explains the trusted-third-party network device can provide a directory service that translates non-standard names, such as E.164 phone numbers, into IP addresses. See ¶¶ 284-285, 326-331, *above*.

517. The trusted-third-party network device’s directory of subscribers can associate each subscriber’s phone number to the IP address of that subscriber’s

second network device and the subscriber's IP address. *See* ¶ 285, *above*; Ex. 1031 (Beser) at 11:45-52 (“the trusted-third-party network device 30 may be a directory service, owned and operated by a telephone company, that retains a list of E.164 numbers of its **subscribers**. Associated with a E.164 number in the directory database is **the IP 58 address of a particular second network device 16**. The database entry may also include **a public IP 58 addresses for the terminating telephony device 26**.”). This association necessarily requires the subscriber to have registered its phone number and IP addresses with the trusted-third-party network device. *See* ¶¶ 306-309, *above*.

518. Beser explains that, by identifying a user with a static unique identifier, the user can still be located even if the user's IP addresses change because, for example, the user has moved from one office to another. Ex. 1031 (Beser) at 10:56-63 (“the user of the terminating telephony device 26 may have moved from one office to another office while still retaining the same electronic mail address. **Rather than identifying the terminating user by the number assigned to a physical device in the office, it may be more appropriate to identify the user by the static electronic mail address.**”), 10:66-11:2 (“There are many other possibilities for the unique identifier, e.g. . . . a **previously assigned public IP 58 address.**”).

519. Beser therefore describes a process where users can change IP addresses. For the trusted-third-party network device to be able to associate a unique identifier with the correct IP addresses, each user must register with the trusted-third-party network device whenever the user's IP addresses change. *See* ¶¶ 127-133, 306-309, *above*.

520. The claim specifies that a secure name is registered. I interpret the claim using the plain and ordinary meaning of the term “register” as it relates to domain names and name services. When an entity registers a domain name with a name server, it specifies the name and one or more addresses to be associated with that name. *See* ¶¶ 91-110, 127-133, *above*.

521. Beser thus shows “*at the first device requesting and obtaining registration of a secure name for the first device, the secure name being associated with a network address*” as specified by claim 24.

522. Beser explains that, to initiate an IP tunnel, an originating device will send out a request to initiate a tunneling association with a terminating device. Beser explains that the request will contain a unique identifier associated with the terminating device. *See* ¶¶ 318-324, *above*. Beser explains that the “request includes a unique identifier” that identifies “the terminating end of the tunneling association.” Ex. 1031 (Beser) at 8:1-3.

523. The request is received by the first network device. Beser explains that the first network device can be a router or other device, and it will evaluate all IP traffic passing through it. *See ¶¶ 283, 302-305, 314-315, above.* If the first network device determines that a packet contains a request for initiating an IP tunnel, it will send the request to the trusted-third-party network device. *See ¶¶ 302-305, 314-317, above.* Otherwise, it would process the packet normally. *See ¶ 317, above.*

524. Beser explains that the request containing the unique identifier (*e.g.*, phone number) will be received by the trusted-third-party network device. *See ¶¶ 318-324, above.* The trusted-third-party network device can be a domain name server, and it also can include a phone number directory service. *See ¶¶ 282, 284-286, 305-309, 324-331, above.*

525. Beser explains that after the trusted-third-party network device receives the request, it evaluates it to determine whether it contains a unique identifier that is associated with a secure terminating device. *See ¶¶ 324-331, above.* For example, if the request contains an identifier (*e.g.*, a phone number) that corresponds to an entry in its directory, the trusted-third-party network device determines the unique identifier corresponds to a secure destination (*i.e.*, one that requires negotiation of an IP tunnel). *See ¶¶ 284-286, 305-309, 324-331, above.* The trusted-third-party network device will automatically negotiate an IP tunnel

between the originating and terminating devices by negotiating private IP addresses. *See* ¶¶ 278, 280-281, *above*; *see also* ¶¶ 326-338, *above*.

526. To negotiate private IP addresses, the trusted-third-party network device will send a message to the second network device requesting to negotiate private IP addresses. *See* ¶¶ 329-335, *above*. The second network device is associated with the terminating device. *See* ¶¶ 282-283, 285, 302-305, 329-339, *above*. Ex. 1031 (Beser) at 13:30-34 (“The trusted-third-party network device 30 has already determined that **the terminating end of this tunneling association 26 is associated with the second network device 16** during the associating Step 106 of Method 100 (FIG. 4).”). Therefore, a message indicating that the originating device desires to communicate securely with the terminating device is received at an address associated with the terminating device.

527. The private IP address for the originating device will be transmitted from the first network device to the second network device through the trusted-third-party network device. *See* ¶¶ 329-335, *above*. Ex. 1031 (Beser) at 12:45-48 (“At Step 144, the first private network address is communicated from the first network device to the second network device through the public network”). This message also is a message indicating that the originating device desires to communicate securely with the terminating device.

528. The private IP address for the terminating device will be transmitted from the second network device to the first network device through the trusted-third-party network device. *See* ¶¶ 329-335, *above*. Ex. 1031 (Beser) at 12:52-54 (“At Step 148, the second private network address is communicated from the second network device to the first network device through the public network.”).

529. Beser thus shows “*receiving at the network address associated with the secure name of the first device a message from a second device of the desire to securely communicate with the first device*” as specified by claim 24.

530. After the private network addresses are negotiated, the first network device will inform the originating device of the private IP address associated with the terminating device. *See* ¶¶ 330-331, *above*. The originating device can then securely transmit data to the terminating device by sending IP packets to the private IP address. *See* ¶¶ 295, 296, *above*. Similarly, the terminating device can then securely transmit data to the originating device by sending IP packets to the private IP address. *See* ¶¶ 336-339, *above*.

531. Beser shows that its IP tunnels allow end devices to directly communicate over a secure and anonymous channel. *See* ¶¶ 278-281, 289-293, 336-338, *above*.

532. Beser indicates that IP traffic within an IP tunnel ordinarily will be encrypted utilizing the techniques described in RFC 2401 (*i.e.*, under the IPsec

protocol). Encryption of the tunneling connection therefore occurs automatically. See ¶¶291-293, 325, 342-353, *above*.

533. Beser thus shows “*sending a message securely from the first device to the second device*” as specified by claim 24.

534. Accordingly, Beser anticipates claim 24 of the ’181 patent under 35 U.S.C. § 102(e).

535. If a sending a message “securely” is found to require all network traffic to be encrypted, and it also is found that Beser does not describe a tunneling association in which the network traffic is encrypted, I believe a person of ordinary skill in April of 2000 would have found encrypting all traffic to have been obvious based on the guidance in Beser and RFC 2401 for the same reasons as I explained above with respect to claim 1. See ¶¶ 397-401, *above*.

536. If the Board finds that Beser does not adequately describe the step of registering a secure name, I believe a person of ordinary skill in April of 2000 would have found that step to have been obvious based on the guidance in Beser and RFC 2543. RFC 2543 describes the SIP protocol, which is another technique for initiating VOIP calls. See ¶¶ 640-644, 660-662, *above*. Beser discloses that its systems can be used with H.323, which is a call signaling protocol that is analogous to SIP. See ¶¶ 286, 301, *above*; see ¶¶ 182-189, *above*. RFC 2543 describes use of a location server, which allows each user to move between devices



and to register its present location so that phone calls always are routed to the correct destination. *See* ¶¶ 666-672, *above*. A person of ordinary skill in the art would have found it obvious to include that functionality in Beser's trusted-third-party network device as it would allow that device to more efficiently maintain its directory service. It was well-known in the art that it is beneficial to configure a device to update a directory service when its network address changed. *See* ¶¶ 127-133, *above*. Thus, the combination of Beser and RFC 2543 would be a simple design choice, and it would be no more than uniting known technologies with no change in their respective functions to yield predictable results.

**u. Claim 25**

537. As I explained above (¶¶ 516-520), Beser explains the trusted-third-party network device can be a standard DNS server that also provides a subscriber directory service that translates non-standard names, such as E.164 phone numbers, into IP addresses. *See* ¶¶ 284-285, 326-331, *above*. Beser also explains that a user can be identified through the trusted-third-party network device even if the user's IP addresses change because, for example, the user switches offices. *See* ¶ 518, *above*; Ex. 1031 (Beser) at 10:56-11:2. Another reason a user's IP address might change is because the local network uses DHCP to allocate IP addresses. *See* Ex. 1031 (Beser) at 6:41-54 & 7:7-15 (discussing DHCP).

538. To maintain the directory, each user would need to register its current IP addresses each time it switched devices or its IP addresses changed. This registration takes place from the user's end device because the end device is in the best position to know when its IP addresses change and an update is needed. Configuring end devices to perform such registrations was well known in the art. See ¶¶ 127-133, *above* (WINS and DNS in Windows 98).

539. The claim specifies that a secure name is registered. I interpret the claim using the plain and ordinary meaning of the term “register” as it relates to domain names and name services. When an entity registers a domain name with a name server, it specifies the name and one or more addresses to be associated with that name. See ¶¶ 91-110, 127-133, *above*.

540. Beser thus shows “*The method according to claim 24, wherein requesting and obtaining registration of a secure name for the first device comprises using the first device to obtain a registration of the secure name for the first device*” as specified in claim 25.

541. As I explained above with respect to claim 24 (¶¶ 530-532), after the private network addresses are negotiated, the originating device can securely transmit data to the terminating device by sending IP packets to the private IP address. See ¶¶ 295, 296, *above*.

542. Beser thus shows “*The method according to claim 24 . . . wherein sending a message securely comprises sending the message from the first device to the second device using a secure communication link*” as specified in claim 25.

**v. Claim 26**

543. As I explained above with respect to claim 1 (¶¶ 372-381), Beser describes methods and systems that create secure and anonymous IP tunnels between an originating device and a terminating device using a trusted-third-party network device. *See* ¶¶ 277-280, 289-291, *above*. The originating device and terminating device can be VOIP devices such as phones or personal computers, and they are associated with a first network device and a second network device, respectively. *See* ¶¶ 282-283, 285, 294-295, 302-305, 329-339, *above*.

544. In describing the functionality of its systems, Beser defines methods of enabling the IP tunneling functionality that could be implemented on a range of standards-compliant system. *See* ¶¶ 277, 286, 306, 354, 358, *above*.

545. Beser thus shows “*A method of using a first device to communicate with a second device over a communication network,*” as specified in the preamble of claim 26.

546. Beser explains that the originating and terminating network devices each have a public IP address. *See* ¶¶ 281-282, 285, 306, 339, *above*.

547. Beser explains that the originating and terminating network devices each have a unique identifier. The unique identifier could be one of many different types of identifier, such as a domain name or a phone number. *See* ¶¶ 318-327, *above*. Beser explains that the “unique identifier is any of a dial-up number, an electronic mail address, or a domain name.” *See* ¶ 319, *above*; Ex. 1031 (Beser) at 10:37-42.

548. Beser shows that the originating and terminating network devices each can be associated with more than one identifier. For example, a terminating network device could be associated with both a phone number and a domain name. *See* ¶¶ 322-323, *above*.

549. During its prosecution of the '181 patent, Patent Owner represented a “‘secure name’ can be a secure non-standard domain name, such as a secure non-standard top-level domain name (e.g., .scom) or a telephone number.” *See* ¶ 212, *above*.

550. Therefore, Beser shows that the terminating device is associated with a secure identifier, such as a phone number. *See* ¶¶ 319-323, *above*. Beser also shows the terminating device is associated with unsecure identifiers, such as a domain name and a public IP address. *Id.* As I explained above (¶¶ 322-323), where a phone number is used as the unique identifier for establishing IP tunnels, the domain name could be used for standard, non-secure communications.

551. Where the device has a domain name listed in a public DNS server, the device necessarily must have registered its domain name and IP address with a DNS server. *See* ¶¶ 91-110, 127-133, *above*.

552. The claim specifies that an unsecured name is registered. I interpret the claim using the plain and ordinary meaning of the term “register” as it relates to domain names and name services. When an entity registers a domain name with a name server, it specifies the name and one or more addresses to be associated with that name. *See* ¶¶ 91-110, 127-133, *above*.

553. Beser thus shows “*from the first device requesting and obtaining registration of an unsecured name associated with the first device*” as specified by claim 26.

554. Beser shows that the trusted-third-party network device can include a directory of subscribers to a telephone service that associates each subscriber’s phone number to the IP address of that subscriber’s second network device and the subscriber’s IP address. *See* ¶¶ 284-285, 326-331, *above*; Ex. 1031 (Beser) at 11:45-52 (“the trusted-third-party network device 30 may be a directory service, owned and operated by a telephone company, that retains a list of E.164 numbers of its **subscribers**. Associated with a E.164 number in the directory database is **the IP 58 address of a particular second network device** 16. The database entry may

also include **a public IP 58 addresses for the terminating telephony device 26.**”).

555. The directory associates each phone number with the correct IP addresses, which necessarily requires the subscriber to have registered its phone number and IP addresses with the trusted-third-party network device. *See ¶¶ 306-309, above.*

556. Beser explains that, by identifying a user with a static unique identifier, the user can still be located even if the user’s IP addresses change because, for example, the user has moved from one office to another. Ex. 1031 (Beser) at 10:56-63 (“the user of the terminating telephony device 26 may have moved from one office to another office while still retaining the same electronic mail address. **Rather than identifying the terminating user by the number assigned to a physical device in the office, it may be more appropriate to identify the user by the static electronic mail address.**”), 10:66-11:2 (“There are many other possibilities for the unique identifier, e.g. . . . **a previously assigned public IP 58 address.**”).

557. Beser therefore describes a process where users can change IP addresses and where the trusted-third-party network device is able to associate a unique identifier with the correct IP addresses. This necessarily requires each user to register with the trusted-third-party network device its unique identifier and IP

addresses whenever the user's IP addresses change. This registration would take place from the user's end device because the end device is in the best position to know when its IP addresses change and an update is needed. Configuring end devices to perform such registrations was well known in the art. *See* ¶¶ 127-133, *above* (WINS and DNS in Windows 98); *see* ¶¶ 91-110, *above*.

558. The claim specifies that a secure name is registered. I interpret the claim using the plain and ordinary meaning of the term "register" as it relates to domain names and name services. When an entity registers a domain name with a name server, it specifies the name and one or more addresses to be associated with that name. *See* ¶¶ 91-110, 127-133, *above*.

559. Beser thus shows "*from the first device requesting and obtaining registration of a secure name associated with the first device, wherein a unique network address corresponds to the secure name associated with the first device*" as specified by claim 26.

560. Beser explains that, to initiate an IP tunnel, an originating device will send out a request to initiate a tunneling association with a terminating device. Beser explains that the request will contain a unique identifier associated with the terminating device. *See* ¶¶ 318-324, *above*. Beser explains that the "request includes a unique identifier" that identifies "the terminating end of the tunneling association." Ex. 1031 (Beser) at 8:1-3.

561. The request is received by the first network device. Beser explains that the first network device can be a router or other device, and it will evaluate all IP traffic passing through it. *See ¶¶ 283, 302-305, 314-315, above.* If the first network device determines that a packet contains a request for initiating an IP tunnel, it will send the request to the trusted-third-party network device. *See ¶¶ 302-305, 314-317, above.* Otherwise, it would process the packet normally. *See ¶ 317, above.*

562. Beser explains that the request containing the unique identifier (*e.g.*, phone number) will be received by the trusted-third-party network device. *See ¶¶ 318-324, above.* The trusted-third-party network device can be a domain name server, and it also can include a phone number directory service. *See ¶¶ 282, 284-286, 305-309, 324-331, above.*

563. Beser explains that after the trusted-third-party network device receives the request, it evaluates it to determine whether it contains a unique identifier that is associated with a secure terminating device. *See ¶¶ 324-331, above.* For example, if the request contains an identifier (*e.g.*, a phone number) that corresponds to an entry in its directory, the trusted-third-party network device determines the unique identifier corresponds to a secure destination (*i.e.*, one that requires negotiation of an IP tunnel). *See ¶¶ 284-286, 305-309, 324-331, above.* The trusted-third-party network device will automatically negotiate an IP tunnel



between the originating and terminating devices by negotiating private IP addresses. *See* ¶¶ 278, 280-281, *above*; *see also* ¶¶ 326-338, *above*.

564. To negotiate private IP addresses, the trusted-third-party network device will send a message to the second network device requesting to negotiate private IP addresses. *See* ¶¶ 329-335, *above*. The second network device is associated with the terminating device. *See* ¶¶ 282-283, 285, 302-305, 329-339, *above*. Ex. 1031 (Beser) at 13:30-34 (“The trusted-third-party network device 30 has already determined that **the terminating end of this tunneling association 26 is associated with the second network device 16** during the associating Step 106 of Method 100 (FIG. 4).”). Therefore, a message indicating that the originating device desires to communicate securely with the terminating device is received at an address associated with the terminating device.

565. The private IP address for the originating device will be transmitted from the first network device to the second network device through the trusted-third-party network device. *See* ¶¶ 329-335, *above*. Ex. 1031 (Beser) at 12:45-48 (“At Step 144, the first private network address is communicated from the first network device to the second network device through the public network”). This message also is a message indicating that the originating device desires to communicate securely with the terminating device.

566. The private IP address for the terminating device will be transmitted from the second network device to the first network device through the trusted-third-party network device. *See* ¶¶ 329-335, *above*. Ex. 1031 (Beser) at 12:52-54 (“At Step 148, the second private network address is communicated from the second network device to the first network device through the public network.”).

567. After the private network addresses are negotiated, the second network device will inform the terminating device of the private IP address associated with the originating device. *See* ¶¶ 330-334, *above*.

568. Beser thus shows “*receiving at the unique network address associated with the secure name a message from a second device requesting the desire to securely communicate with the first device*” as specified by claim 26.

569. Beser explains the originating device will transmit data to the terminating device by addressing IP packets to the private IP address of the terminating device. *See* ¶¶ 336-339, *above*. Beser explains the first network device will receive the packet, recognize it is destined for the terminating end device, and transmit it to the second network device. *See* ¶¶ 336-339, *above*. The second network device will route the packet to the terminating device. *See* ¶¶ 336-339, *above*.

570. Similarly, Beser explains the terminating device will transmit data to the originating device by addressing IP packets to the private IP address of the

originating device. *See* ¶¶ 336-339, *above*. Beser explains the second network device will receive the packet, recognize it is destined for the originating end device, and transmit it to the first network device. *See* ¶¶ 336-339, *above*. The first network device will route the packet to the originating device. *See* ¶¶ 336-339, *above*.

571. Beser shows that its IP tunnels allow end devices to directly communicate over a secure and anonymous channel. *See* ¶¶ 278-281, 289-293, 336-338, *above*.

572. Beser indicates that IP traffic within an IP tunnel ordinarily will be encrypted utilizing the techniques described in RFC 2401 (*i.e.*, under the IPsec protocol). Encryption of the tunneling connection therefore occurs automatically. *See* ¶¶ 291-293, 325, 342-353, *above*.

573. Beser thus shows “*from the first device sending a message securely from the first device to the second device*” as specified by claim 26.

574. Accordingly, Beser anticipates claim 26 of the ’181 patent under 35 U.S.C. § 102(e).

575. If a sending a message “securely” is found to require all network traffic to be encrypted, and it also is found that Beser does not describe a tunneling association in which the network traffic is encrypted, I believe a person of ordinary skill in April of 2000 would have found encrypting all traffic to have been obvious

based on the guidance in Beser and RFC 2401 for the same reasons as I explained above with respect to claim 1. *See* ¶¶ 397-401, *above*.

576. If the Board finds that Beser does not adequately describe the step of registering a secure or unsecured name, I believe a person of ordinary skill in April of 2000 would have found that step to have been obvious based on the guidance in Beser and RFC 2543 for the same reasons I presented above with respect to claims 22, 24, and 25. *See* ¶¶ 504, 536, 538-539, *above*.

**w. Claim 27**

577. Therefore, Beser shows that the terminating device is associated with a secure identifier, such as a phone number. *See* ¶¶ 319-323, *above*. Beser also shows the terminating device is associated with unsecure identifiers, such as a domain name and a public IP address. *Id.* As I explained above (¶¶ 322-323), where a phone number is used as the unique identifier for establishing IP tunnels, the domain name could be used for standard, non-secure communications.

578. Where the device has a domain name listed in a public DNS server, the device necessarily must have registered its domain name and IP address with a DNS server. Configuring end devices to perform such registrations was well known in the art. *See* ¶¶ 127-133, *above* (WINS and DNS in Windows 98).

579. Beser thus shows “*wherein requesting and obtaining registration of an unsecured name associated with the first device comprises using the first device*

*to obtain a registration of the unsecured name associated with the first device,” as specified by claim 27.*

580. Beser explains the trusted-third-party network device can be a standard DNS server. Beser also explains the trusted-third-party network device can provide a directory service that translates non-standard names, such as E.164 phone numbers, into IP addresses. *See ¶¶ 284-285, 326-331, above.*

581. The trusted-third-party network device’s directory of subscribers can associate each subscriber’s phone number to the IP address of that subscriber’s second network device and the subscriber’s IP address. *See ¶ 285, above; Ex. 1031 (Beser) at 11:45-52 (“the trusted-third-party network device 30 may be a directory service, owned and operated by a telephone company, that retains a list of E.164 numbers of its **subscribers**. Associated with a E.164 number in the directory database is **the IP 58 address of a particular second network device 16**. The database entry may also include **a public IP 58 addresses for the terminating telephony device 26**.”).* This association necessarily requires the subscriber to have registered its phone number and IP addresses with the trusted-third-party network device.

582. Beser explains that, by identifying a user with a static unique identifier, the user can still be located even if the user’s IP addresses change because, for example, the user has moved from one office to another. Ex. 1031

(Beser) at 10:56-63 (“the user of the terminating telephony device 26 may have moved from one office to another office while still retaining the same electronic mail address. **Rather than identifying the terminating user by the number assigned to a physical device in the office, it may be more appropriate to identify the user by the static electronic mail address.**”), 10:66-11:2 (“There are many other possibilities for the unique identifier, e.g. . . . a **previously assigned public IP 58 address.**”). Another reason a user’s IP address might change is because the local network uses DHCP to allocate IP addresses. *See* Ex. 1031 (Beser) at 6:41-54 & 7:7-15 (discussing DHCP).

583. Beser therefore describes a process where users can change IP addresses. For the trusted-third-party network device to be able to associate a unique identifier with the correct IP addresses, each user must register with the trusted-third-party network device whenever the user’s IP addresses change. This registration would take place from the user’s end device because the end device is in the best position to know when its IP addresses change and an update is needed. Configuring end devices to perform such registrations was well known in the art. *See* ¶¶ 127-133, *above* (WINS and DNS in Windows 98).

584. Beser thus shows “*wherein requesting and obtaining registration of a secure name associated with the first device comprises using the first device to*

*obtain a registration of the secure name associated with the first device,” as specified by claim 27.*

585. Accordingly, Beser anticipates claim 27 of the ’181 patent under 35 U.S.C. § 102(e).

**x. Claim 28**

586. As I explained above with respect to claim 1 (¶¶ 372-381), Beser describes methods and systems that create secure and anonymous IP tunnels between an originating device and a terminating device using a trusted-third-party network device. *See* ¶¶ 277-280, 289-291, *above*. The originating device and terminating device can be VOIP devices such as phones or personal computers, and they are associated with a first network device and a second network device, respectively. *See* ¶¶ 282-283, 285, 294-295, 302-305, 329-339, *above*.

587. Beser explains that the originating and terminating network devices each have a public IP address. *See* ¶¶ 281-282, 285, 306, 339, *above*.

588. Beser explains that the originating and terminating network devices each have a unique identifier. The unique identifier could be one of many different types of identifier, such as a domain name or a phone number. *See* ¶¶ 318-327, *above*. Beser explains that the “unique identifier is any of a dial-up number, an electronic mail address, or a domain name.” *See* ¶ 319, *above*; Ex. 1031 (Beser) at 10:37-42.

589. Beser shows that the originating and terminating network devices each can be associated with more than one identifier. For example, a terminating network device could be associated with both a phone number and a domain name. *See* ¶¶ 322-323, *above*.

590. During its prosecution of the '181 patent, Patent Owner represented a “‘secure name’ can be a secure non-standard domain name, such as a secure non-standard top-level domain name (e.g., .scom) or a telephone number.” *See* ¶ 212, *above*.

591. Therefore, Beser shows that the terminating device is associated with a secure identifier, such as a phone number. *See* ¶¶ 319-323, *above*. Beser also shows the terminating device is associated with unsecure identifiers, such as a domain name and a public IP address. *Id.* As I explained above (¶¶ 322-323), where a phone number is used as the unique identifier for establishing IP tunnels, the domain name could be used for standard, non-secure communications.

592. In describing the functionality of its systems, Beser defines methods of enabling the IP tunneling functionality that could be implemented on a range of standards-compliant system. *See* ¶¶ 277, 286, 306, 354, 358, *above*.

593. The Beser systems are implemented using programmed computers and other programmable devices that have storage media containing code that configures how those devices function.



594. Beser thus shows “*A non-transitory machine-readable medium comprising instructions,*” as specified by the preamble of claim 28.

595. Beser explains that, to initiate an IP tunnel, an originating device will send out a request to initiate a tunneling association with a terminating device. Beser explains that the request will contain a unique identifier associated with the terminating device. *See* ¶¶ 318-324, *above*. Beser explains that the “request includes a unique identifier” that identifies “the terminating end of the tunneling association.” Ex. 1031 (Beser) at 8:1-3.

596. The request is received by the first network device. Beser explains that the first network device can be a router or other device, and it will evaluate all IP traffic passing through it. *See* ¶¶ 283, 302-305, 314-315, *above*. If the first network device determines that a packet contains a request for initiating an IP tunnel, it will send the request to the trusted-third-party network device. *See* ¶¶ 302-305, 314-317, *above*. Otherwise, it would process the packet normally. *See* ¶ 317, *above*.

597. Beser explains that the request containing the unique identifier (*e.g.*, phone number) will be received by the trusted-third-party network device. *See* ¶¶ 318-324, *above*.

598. Beser explains the trusted-third-party network device can be a standard DNS server. Beser also explains the trusted-third-party network device

can provide a directory service that translates non-standard names, such as E.164 phone numbers, into IP addresses. *See ¶¶ 284-285, 326-331, above.*

599. The trusted-third-party network device is a secure name service within the meaning of the claims because it can resolve non-standard domain names such as phone numbers.

600. Beser shows that the trusted-third-party network device can require “encryption or authentication.” *See ¶¶ 291-293, 325, 340-353, above.* A person of ordinary skill in the art would understand the trusted-third party network device to be a secure name service because it is described as a “trusted” device and it can require users to authenticate before communicating with them.

601. Beser explains that after the trusted-third-party network device receives the request, it evaluates it to determine whether it contains a unique identifier that is associated with a secure terminating device. *See ¶¶ 324-331, above.* For example, if the request contains an identifier (*e.g.*, a phone number) that corresponds to an entry in its directory, the trusted-third-party network device determines the unique identifier corresponds to a secure destination (*i.e.*, one that requires negotiation of an IP tunnel). *See ¶¶ 284-286, 305-309, 324-331, above.* The trusted-third-party network device will automatically negotiate an IP tunnel between the originating and terminating devices by negotiating private IP addresses. *See ¶¶ 278, 280-281, above; see also ¶¶ 326-338, above.*

602. Beser thus shows “*sending a message to a secure name service, the message requesting a network address associated with a secure name of a device*” as specified by claim 28.

603. If the trusted-third-party network device determines the unique identifier corresponds to a secure destination (*e.g.*, the destination of a VOIP request), it will negotiate private IP addresses for the originating and terminating devices. *See* ¶¶ 284-286, 305-309, 324-338, *above*.

604. To negotiate private IP addresses, the trusted-third-party network device will send a message to the second network device requesting to negotiate private IP addresses. *See* ¶¶ 329-335, *above*. The second network device is associated with the terminating device. *See* ¶¶ 282-283, 285, 302-305, 329-339, *above*; Ex. 1031 (Beser) at 13:30-34 (“The trusted-third-party network device 30 has already determined that **the terminating end of this tunneling association 26 is associated with the second network device 16** during the associating Step 106 of Method 100 (FIG. 4).”). The trusted-third-party network device also will inform the first network of the second network device’s IP address. *See* ¶¶ 330-337, *above*.

605. The private IP address for the originating device will be transmitted from the first network device to the second network device through the trusted-third-party network device. *See* ¶¶ 329-335, *above*; Ex. 1031 (Beser) at 12:45-48

(“At Step 144, the first private network address is communicated from the first network device to the second network device through the public network”).

606. The private IP address for the terminating device will be transmitted from the second network device to the first network device through the trusted-third-party network device. *See* ¶¶ 329-335, *above*; Ex. 1031 (Beser) at 12:52-54 (“At Step 148, the second private network address is communicated from the second network device to the first network device through the public network.”).

607. After the private network addresses are negotiated, the second network device will inform the terminating device of the private IP address associated with the originating device. *See* ¶¶ 330-334, *above*.

608. Beser thus shows “*receiving a message containing the network address associated with the secure name of the device*” as specified by claim 28.

609. Beser shows that the trusted-third-party network device can require “encryption or authentication” of any communications exchanged with it. *See* ¶¶ 291-293, 325, 340-353, *above*. Therefore, during the negotiation process, the first network device sends a message to the address associated with the terminating device using a secure communication link.

610. Beser explains the originating device will transmit data to the terminating device by addressing IP packets to the private IP address of the terminating device. *See* ¶¶ 336-339, *above*. Beser explains the first network

device will receive the packet, recognize it is destined for the terminating end device, and transmit it to the second network device. *See* ¶¶ 336-339, *above*. The second network device will route the packet to the terminating device. *See* ¶¶ 336-339, *above*.

611. Beser shows that its IP tunnels allow end devices to directly communicate over a secure and anonymous channel. *See* ¶¶ 278-281, 289-293, 336-338, *above*. Beser indicates that IP traffic within an IP tunnel ordinarily will be encrypted utilizing the techniques described in RFC 2401 (*i.e.*, under the IPsec protocol). Encryption of the tunneling connection therefore occurs automatically. *See* ¶¶ 291-293, 325, 342-353, *above*.

612. Beser thus shows “*sending a message to the network address associated with the secure name of the device using a secure communication link*” as specified by claim 28.

613. Accordingly, Beser anticipates claim 28 of the ’181 patent under 35 U.S.C. § 102(e).

614. If a secure communication link is found to require all network traffic to be encrypted, and it also is found that Beser does not describe a tunneling association in which the network traffic is encrypted, I believe a person of ordinary skill in April of 2000 would have found encrypting all traffic to have been obvious

based on the guidance in Beser and RFC 2401 for the same reasons as I explained above with respect to claim 1. *See* ¶¶ 397-401, *above*.

**y. Claim 29**

615. As I explained above with respect to claim 1 (¶¶ 372-381), Beser describes methods and systems that create secure and anonymous IP tunnels between an originating device and a terminating device using a trusted-third-party network device. *See* ¶¶ 277-280, 289-291, *above*. The originating device and terminating device can be VOIP devices such as phones or personal computers, and they are associated with a first network device and a second network device, respectively. *See* ¶¶ 282-283, 285, 294-295, 302-305, 329-339, *above*.

616. Beser explains that the originating and terminating network devices each have a public IP address. *See* ¶¶ 281-282, 285, 306, 339, *above*.

617. Beser explains that the originating and terminating network devices each have a unique identifier. The unique identifier could be one of many different types of identifier, such as a domain name or a phone number. *See* ¶¶ 318-327, *above*. Beser explains that the “unique identifier is any of a dial-up number, an electronic mail address, or a domain name.” *See* ¶ 319, *above*; Ex. 1031 (Beser) at 10:37-42.

618. Beser shows that the originating and terminating network devices each can be associated with more than one identifier. For example, a terminating

network device could be associated with both a phone number and a domain name. *See* ¶¶ 322-323, *above*.

619. During its prosecution of the '181 patent, Patent Owner represented a “‘secure name’ can be a secure non-standard domain name, such as a secure non-standard top-level domain name (e.g., .scom) or a telephone number.” *See* ¶ 212, *above*.

620. Therefore, Beser shows that the terminating device is associated with a secure identifier, such as a phone number. *See* ¶¶ 319-323, *above*. Beser also shows the terminating device is associated with unsecure identifiers, such as a domain name and a public IP address. *Id.* As I explained above (¶¶ 322-323), where a phone number is used as the unique identifier for establishing IP tunnels, the domain name could be used for standard, non-secure communications.

621. In describing the functionality of its systems, Beser defines methods of enabling the IP tunneling functionality that could be implemented on a range of standards-compliant system. *See* ¶¶ 277, 286, 306, 354, 358, *above*.

622. The Beser systems are implemented using programmed computers and other programmable devices that have storage media containing code that configures how those devices function.

623. Beser thus shows “*A non-transitory machine-readable medium comprising instructions for a method of communicating with a device having a secure name,*” as specified by the preamble of claim 29.

624. Beser explains that, to initiate an IP tunnel, an originating device will send out a request to initiate a tunneling association with a terminating device. Beser explains that the request will contain a unique identifier associated with the terminating device. *See* ¶¶ 318-324, *above*. Beser explains that the “request includes a unique identifier” that identifies “the terminating end of the tunneling association.” Ex. 1031 (Beser) at 8:1-3.

625. The request is received by the first network device. Beser explains that the first network device can be a router or other device, and it will evaluate all IP traffic passing through it. *See* ¶¶ 283, 302-305, 314-315, *above*. If the first network device determines that a packet contains a request for initiating an IP tunnel, it will send the request to the trusted-third-party network device. *See* ¶¶ 302-305, 314-317, *above*. Otherwise, it would process the packet normally. *See* ¶ 317, *above*.

626. Beser explains that the request containing the unique identifier (*e.g.*, phone number) will be received by the trusted-third-party network device. *See* ¶¶ 318-324, *above*. The trusted-third-party network device can be a domain name



server, and it also can include a phone number directory service. *See ¶¶ 282, 284-286, 305-309, 324-331, above.*

627. Beser explains that after the trusted-third-party network device receives the request, it evaluates it to determine whether it contains a unique identifier that is associated with a secure terminating device. *See ¶¶ 324-331, above.* For example, if the request contains an identifier (*e.g.*, a phone number) that corresponds to an entry in its directory, the trusted-third-party network device determines the unique identifier corresponds to a secure destination (*i.e.*, one that requires negotiation of an IP tunnel). *See ¶¶ 284-286, 305-309, 324-331, above.* The trusted-third-party network device will automatically negotiate an IP tunnel between the originating and terminating devices by negotiating private IP addresses. *See ¶¶ 278, 280-281, above; see also ¶¶ 326-338, above.*

628. After the private network addresses are negotiated, the second network device will inform the terminating device of the private IP address associated with the originating device. *See ¶¶ 330-334, above.*

629. As I explained above with respect to claim 24 (¶¶ 516-519), Beser describes a process where the trusted-third-party network device keeps its subscriber directory up to date, even where users change IP addresses. For the trusted-third-party network device to be able to associate a unique identifier with the correct IP addresses, each user necessarily must register with the trusted-third-

party network device whenever the user's IP addresses change. *See ¶¶ 91-110, 127-133, above.*

630. The claim specifies that a secure name is registered. I interpret the claim using the plain and ordinary meaning of the term “register” as it relates to domain names and name services. When an entity registers a domain name with a name server, it specifies the name and one or more addresses to be associated with that name. *See ¶¶ 91-110, 127-133, above.*

631. Beser thus shows “*receiving at a network address associated with a secure name of a first device a message from a second device requesting the desired to securely communicate with the first device, wherein the secure name of the first device is registered*” as specified by claim 29.

632. Beser explains the originating device will transmit data to the terminating device by addressing IP packets to the private IP address of the terminating device. *See ¶¶ 295, 296, above.* Beser explains the first network device will receive the packet, recognize it is destined for the terminating end device, and transmit it to the second network device. *See ¶¶ 336-339, above.* The second network device will route the packet to the terminating device. *See ¶¶ 336-339, above.*

633. Similarly, Beser explains the terminating device will transmit data to the originating device by addressing IP packets to the private IP address of the

originating device. *See* ¶¶ 336-339, *above*. Beser explains the second network device will receive the packet, recognize it is destined for the originating end device, and transmit it to the first network device. *See* ¶¶ 336-339, *above*. The first network device will route the packet to the originating device. *See* ¶¶ 336-339, *above*.

634. Beser shows that its IP tunnels allow end devices to directly communicate over a secure and anonymous channel. *See* ¶¶ 278-281, 289-293, 336-338, *above*.

635. Beser indicates that IP traffic within an IP tunnel ordinarily will be encrypted utilizing the techniques described in RFC 2401 (*i.e.*, under the IPsec protocol). Encryption of the tunneling connection therefore occurs automatically. *See* ¶¶ 291-293, 325, 342-353, *above*.

636. Beser thus shows “*sending a message securely from the first device to the second device*” as specified by claim 29.

637. Accordingly, Beser anticipates claim 29 of the ’181 patent under 35 U.S.C. § 102(e).

638. If a sending a message “securely” is found to require all network traffic to be encrypted, and it also is found that Beser does not describe a tunneling association in which the network traffic is encrypted, I believe a person of ordinary skill in April of 2000 would have found encrypting all traffic to have been obvious

based on the guidance in Beser and RFC 2401 for the same reasons as I explained above with respect to claim 1. *See* ¶¶ 397-401, *above*.

## **B. RFC 2543 – Session Initiation Protocol**

639. As explained in more detail below, the processes, systems, and code described in RFC 2543 anticipate and would have made obvious to a person of ordinary skill in the art claims 1-11, 14-25, and 28-30 of the '181 patent.

### **1. Overview of RFC 2543**

640. RFC 2543 describes the Session Initiation Protocol (SIP), a standard promulgated by the IETF. The SIP protocol was developed through extensive deliberations, and reflects well-known and accepted principles for designing call signaling mechanisms applicable to establishing multimedia communications between devices over a network.

641. SIP is “an application-layer control (signaling) protocol for creating, modifying and terminating **sessions** with one or more participants.” Ex. 1033 (RFC 2543) at 1, 7. RFC 2543 explains that a “session” “include[s] Internet multimedia conferences, Internet telephone calls and multimedia distribution.” Ex. 1033 (RFC 2543) at 1, 7; *see* Ex. 1033 (RFC 2543) at 11.

642. SIP was designed as part of a suite of protocols developed by the IETF for transmitting multimedia data. As RFC 2543 explains:

SIP is designed as part of the overall IETF multimedia data and control architecture currently incorporating protocols such as RSVP

(RFC 2205 [2]) for reserving network resources, **the real-time transport protocol (RTP) (RFC 1889 [3]) for transporting real-time data** and providing QOS feedback, the real-time streaming protocol (RTSP) (RFC 2326 [4]) for controlling delivery of streaming media, the session announcement protocol (SAP) [5] for advertising multimedia sessions via multicast and **the session description protocol (SDP) (RFC 2327 [6]) for describing multimedia sessions**. However, the functionality and operation of SIP does not depend on any of these protocols.

Ex. 1033 (RFC 2543) at 8. While other standards govern the description and packet format of multimedia transmissions, SIP governs the processes for exchanging messages for initiating, maintaining, and ending multimedia transmissions (*e.g.*, voice and video conference calls).

643. RFC 2543 defines a SIP “session” as:

Session: . . . ‘A multimedia session is a set of multimedia senders and receivers and the data streams flowing from senders to receivers. A multimedia conference is an example of a multimedia session.’ (RFC 2327) (A session as defined for SDP can comprise one or more RTP sessions.) As defined, a callee can be invited several times, by different calls, to the same session. If SDP is used, a session is defined by the concatenation of the user name, session id, network type, address type and address elements in the origin field.

Ex. 1033 (RFC 2543) at 11. The referenced RFC is RFC 2327, the Session Description Protocol (SDP), another standard in the IETF multimedia data and control architecture. *See* Ex. 1033 (RFC 2543) at 8, 11.

644. RFC 2543 explains that a “call” is a set of sessions between two or more parties communicating in a “conference.” A “conference” is defined as a “multimedia session.” For example, “[a] point-to-point Internet telephony conversation maps into a single SIP call.” Ex. 1033 (RFC 2543) at 9. A video conference between two parties also would be an example of a call. *See* Ex. 1033 (RFC 2543) at 136-137.

645. RFC 2543 shows SIP uses session descriptions written in SDP format to communicate information about the multimedia streams to be exchanged between parties. Ex. 1033 (RFC 2543) at 15 (the INVITE message “typically contains a session description, for example written in SDP (RFC 2327) format, that provides the called party with enough information to join the session.”), 84 (for certain SIP messages, “the message body is always a session description”), 136-40.

646. RFC 2327 explains that “The purpose of SDP is to convey information about media streams in multimedia sessions to allow the recipients of a session description to participate in the session.” Ex. 1035 (RFC 2327) at 4. It further explains that an SDP description will include data fields that specify:

- o The type of media (video, audio, etc)

- o The transport protocol (RTP/UDP/IP, H.320, etc)
- o The format of the media (H.261 video, MPEG video, etc)

Ex. 1035 (RFC 2327) at 5.

647. Each SDP description can specify one or more media streams with different parameters for each stream. Ex. 1035 (RFC 2327) at 4-5 (“A multimedia session, for these purposes, is defined as a set of media streams that exist for some duration of time”); *see* Ex. 1033 (RFC 2543) at 11 (“A session as defined for SDP can comprise one or more RTP sessions”); Ex. 1034 (RFC 1889) at 8 (“In a multimedia session, each medium is carried in a separate RTP session . . .”). For example, the description may include information about the bandwidth for each stream and other session attributes. Ex. 1035 (RFC 2327) at 7-8.

648. RFC 2543 explains that encryption keys for the session may be specified in an SDP description sent in a SIP message. Ex. 1033 (RFC 2543) at 104-05 (“The SIP message body MAY also contain encryption keys for the session itself.”).

649. RFC 2327 also explains that an SDP description may include one or more encryption keys to be used to encrypt the streams described. Ex. 1035 (RFC 2327) at 17 (an SDP description “may be used to convey encryption keys. A key field is permitted before the first media entry (in which case it applies to all media in the session), or for each media entry as required.”). The SDP description may

specify one key to use for all the streams, or it may specify that a key be used for only a particular stream. Ex. 1035 (RFC 2327) at 17.

650. RFC 2543 explains that the contents of SIP communications may contain sensitive information. Ex. 1033 (RFC 2543) at 104 (“SIP requests and responses can contain sensitive information about the communication patterns and communication content of individuals. The SIP message body MAY also contain **encryption keys for the session itself.**”). RFC 2543 then explains that that the SIP scheme incorporates a robust use of encryption, stating that “SIP supports three complementary forms of encryption to protect privacy....” Ex. 1033 (RFC 2543) at 104. Those three encryption techniques are (i) “end-to-end encryption of the SIP message body and certain sensitive header fields”; (ii) “hop-by-hop encryption to prevent eavesdropping that tracks who is calling whom;” and (iii) “hop-by-hop encryption of Via fields to hide the route a request has taken.” Ex. 1033 (RFC 2543) at 104-05.

651. Each message sent in the SIP scheme will carry information indicating whether encryption is being used in the communication. Specifically, the SIP general-header field contains an “Encryption” field, which indicates whether the message has been encrypted. As RFC 2543 explains:

The Encryption general-header field specifies that the content has been encrypted. Section 13 describes the overall SIP security architecture and algorithms. This header field is intended for end-to-



end encryption of requests and responses. Requests are encrypted based on the public key belonging to the entity named in the To header field. Responses are encrypted based on the public key conveyed in the Response-Key header field. Note that the public keys themselves may not be used for the encryption. This depends on the particular algorithms used.

Ex. 1035 (RFC 2543) at 54; *see also* Ex. 1035 (RFC 2543) at 25-26 (describing form of messages, including the “general-header” field), and at 27 (showing that “Encryption” is one of the data elements in the “general-header” field of each message).

652. The inclusion of an “Encryption” field in the general-header of each SIP message enables the sender or receiver to determine if the message being transmitted will be encrypted. In other words, the callee or the caller, upon receiving a SIP message, will parse the general-field and determine based on the encryption parameter whether encryption is being used for the communication.

653. RFC 2543 shows that the use of encryption can be specified from the very start of the exchange that sets up a call. This is illustrated in the examples provided in RFC 2543. For example, on page 55, RFC 2543 shows an example of an “INVITE” message that has been encrypted. One reason the message would be encrypted is because it specifies the encryption key or keys to be used to encrypt session data. Ex. 1033 (RFC 2543) at 104 (“The SIP message body MAY also

contain **encryption keys for the session itself.**”). The INVITE message, of course, is the first communication sent by the caller to the callee, and starts the “handshake” process that sets up a call between the caller and the callee. Ex. 1033 (RFC 2543) at 17. So, in this example, the caller is indicating from the inception of the calling process whether encryption should be used for the call.

654. RFC 2543 shows that each stream in a session or call would be handled by a separate “RTP” session. Ex. 1033 (RFC 2543) at 136-140 (showing SDP descriptions specifying an RTP session for each data stream); Ex. 1033 (RFC 2543) at 11 (“A session as defined for SDP can comprise one or more RTP sessions”); Ex. 1034 (RFC 1889) at 8 (“In a multimedia session, each medium is carried in a separate RTP session . . .”). The “RTP” standard is the Real Time Transport Protocol (RTP), which is defined in RFC 1889. RFC 1889 is another standard in the IETF multimedia data and control architecture.

655. RFC 1889 explains that RTP defines a standard for the transmission of multimedia data. “RTP provides end-to-end network transport functions suitable for applications transmitting real-time data, such as audio, video or simulation data, over multicast or unicast network services.” Ex. 1034 (RFC 1889) at 1, 3.

656. RFC 2543 shows that SIP utilizes both the SDP and RTP standards to establish multimedia sessions or calls. Each of the three protocols regulates

different aspects of a call. SIP is used to establish, maintain, and terminate a multimedia call. SDP is used by SIP to exchange among call participants a session description that contains the parameters for the call. Ex. 1033 (RFC 2543) at 84 (for certain SIP messages, “the message body is always a session description”). The session description may specify that RTP (or another protocol) be used to packetize and transmit the multimedia data exchanged during the call. Ex. 1033 (RFC 2543) at 8, 11, 136-140. RTP would govern the format of packets containing multimedia data that are exchanged during a call. Therefore, RFC 2543 incorporates RFC 2327 and RFC 1889 by reference.

657. RFC 2543 explains that SIP can be used with other call setup and signaling protocols, such as those developed by the ITU. As RFC 2543 explains:

SIP can also be used in conjunction with other call setup and signaling protocols. In that mode, an end system uses SIP exchanges to determine the appropriate end system address and protocol from a given address that is protocol-independent. For example, **SIP could be used to determine that the party can be reached via H.323 [7], obtain the H.245 [8] gateway and user address and then use H.225.0 [9] to establish the call.**

Ex. 1033 (RFC 2543) at 8.

### **SIP Basics**

658. A summary of the basic process for establishing a call using SIP is as follows:

Callers and callees are identified by SIP addresses, described in Section 1.4.1. When making a SIP call, a caller first locates the appropriate server (Section 1.4.2) and then sends a SIP request (Section 1.4.3). The most common SIP operation is the invitation (Section 1.4.4). Instead of directly reaching the intended callee, a SIP request may be redirected or may trigger a chain of new SIP requests by proxies (Section 1.4.5). Users can register their location(s) with SIP servers (Section 4.2.6).

Ex. 1033 (RFC 2543) at 12.

659. The SIP architecture includes several distinct devices, including: at least two SIP user agents (*e.g.*, software for a SIP client), one or more SIP servers (such as proxy servers or redirection servers), and one or more location servers.

Ex. 1033 (RFC 2543) at 9-11. The architecture contemplates the existence of at least two different parties: a caller (or calling party) and a callee (or called party).

Ex. 1033 (RFC 2543) at 9-11.

660. SIP supports point-to-point calls between two parties. Ex. 1033 (RFC 2543) at 1, 7. It also supports conferences among multiple parties. Ex. 1033 (RFC 2543) at 1, 7.

661. SIP supports Internet telephony calls. Ex. 1033 (RFC 2543) at 9 (“A point-to-point Internet telephony conversation maps into a single SIP call.”).

662. SIP also can be used in conjunction with other protocols. For example, SIP supports Internet-to-Public Switched Telephone Network (PSTN) calls. Ex. 1033 (RFC 2543) at 8; *see id.* at 91.

663. A party is identified by a SIP address also called a “SIP URL.” SIP URLs take the form of: “user at domain name” or “user at IP address”. Ex. 1033 (RFC 2543) at 12 (“The SIP URL takes a form similar to a mailto or telnet URL, i.e., user@host. The user part is a user name or a telephone number. The host part is either a **domain name** or a numeric network address.”).

664. The user portion of a SIP URL can be a user name or a phone number. For example, “sip:j.doe@big.com” and “sip:alice@example.com” would be SIP URLs where “j.doe” and “alice” represent a user name. Ex. 1033 (RFC 2543) at 24. An example of a SIP URL with a phone number for a user name is “sip:+1-212-555-1212@gateway.com;user=phone”. Ex. 1033 (RFC 2543) at 24.

665. Alternatively, the user’s host location can be identified by using an IP address. For example, “sip:alice@10.1.2.3” would be a valid SIP URL. Ex. 1033 (RFC 2543) at 12, 24.

666. Under SIP, users wishing to be able to receive calls register their availability with a SIP server by sending a REGISTER request message to the server. In the REGISTER message, the user will specify its SIP address and the current address at which it can be reached. The registration process allows a user

to move between a number of different locations or end systems while still being able to receive calls addressed to its SIP address. Ex. 1033 (RFC 2543) at 17-18.

667. The SIP server will record the user's location in a location server. The location server will store the "address-of-record" for the user, which "typically [is] of the form 'user@domain' rather than 'user@host'." Ex. 1033 (RFC 2543) at 32. However, either SIP URL format can be used.

668. If a user changes location, it can send another REGISTER message to the SIP server to change the user's address to reflect the user's new location. Ex. 1033 (RFC 2543) at 31-34. If a user signs off, it can cancel a registration by sending a message to the SIP server. Ex. 1033 (RFC 2543) at 33.

669. Registrations are not permanent, and they expire after a period set by the SIP server. If no period is specified, the default expiration period for a registration is one hour. Ex. 1033 (RFC 2543) at 33.

670. When a user registers with a server, the user may specify special call handling procedures to be used. Ex. 1033 (RFC 2543) at 18 ("The REGISTER request allows a client to let a proxy or redirect server know at which address(es) it can be reached. A client MAY also use it to install call handling features at the server.").

671. The registration process enables a user to be identifiable by a “unique personal identity (i.e., personal number)” and switch devices and still receive calls.

As RFC 2543 explains:

**These facilities also enable personal mobility.** In the parlance of telecommunications intelligent network services, this is defined as: "Personal mobility is the ability of end users to originate and receive calls and access subscribed telecommunication services on any terminal in any location, and the ability of the network to identify end users as they move. **Personal mobility is based on the use of a unique personal identity (i.e., personal number).**" Personal mobility complements terminal mobility, i.e., the ability to maintain communications when moving a single end system from one subnet to another.

Ex. 1033 (RFC 2543) at 7.

672. RFC 2543 provides that SIP could be used on many types of devices, including mobile devices such as mobile phones and laptops.

The "tag" parameter serves as a general mechanism to distinguish multiple instances of a user identified by a single SIP URL. As proxies can fork requests, the same request can reach multiple instances of a user (**mobile and home phones**, for example).

Ex. 1033 (RFC 2543) at 65-66.

673. Although SIP could be used on mobile devices, it did not support terminal mobility (*i.e.*, allowing a user on a mobile device actively involved in a

call to switch networks without the call dropping). *See* Ex. 1033 (RFC 2543) at 7.

However, Dr. Henning Schulzrinne, one of the authors of RFC 2543, proposed modifications to the SIP standard to enable mobile users to move between networks while using SIP. As Dr. Schulzrinne explained:

The application layer protocol Session Initiation Protocol (SIP) already supports personal mobility, and the changes needed to support device mobility are minor. **In this document, we will discuss how mobility support in SIP can improve the performance for realtime services in wireless networks**, and propose an architecture for how this can be done.

Ex. 1075 (Mobility Support Using SIP) at 1.

674. Dr. Schulzrinne further explained that:

As was stated in the previous section, SIP supports personal mobility, i.e., a user can be found independent of location and network device (PC, laptop, IP phone, etc.). The step from personal mobility to IP mobility support is basically the roaming frequency, and that a user can change location (IP address) during a traffic flow. Therefore, in order to support IP mobility, we need to add the ability to move while a session is active. It is assumed that the mobile host belongs to a home network, on which there is a SIP server (in this example, a SIP redirect server), which receives registrations from the mobile host each time it changes location. This is similar to home agent registration in Mobile IP.



Ex. 1075 (Mobility Support Using SIP) at 3 (78); *see also* Ex. 1033 (RFC 2543) at 2 (“SIP is designed to be independent of the lower-layer transport protocol and can be extended with additional capabilities”). As Dr. Schulzrinne explains, the SIP protocol already can be implemented on PCs, laptops, and IP phones, even without modifying it to support IP mobility.

### **Initiating a SIP Call**

675. A call between a caller and a callee is initiated by the transmission of an INVITE message by the caller. Ex. 1033 (RFC 2543) at 20, 96.

676. An INVITE message is structured similar to an HTTP message, and it contains a number of various header fields, including a “To”, “From”, “Request-URI”, and “Via” field. Ex. 1033 (RFC 2543) at 24-28. The message also includes a “message body” that contains a session description. Ex. 1033 (RFC 2543) at 26, 84-85.

677. A caller addresses an INVITE message to a callee by putting the callee’s SIP address in the “To” and “Request-URI” fields of the INVITE message. Ex. 1033 (RFC 2543) at 20, 96.

678. The caller will put its own address in the “From” field of the INVITE message. It also must put its own address into a “Via” field:

The client originating the request **MUST** insert into the request a Via field containing its **host name** or network address and, if not the default port number, the port number at which it wishes to receive

responses. (Note that this port number can differ from the UDP source port number of the request.) **A fully-qualified domain name is RECOMMENDED.**

Ex. 1033 (RFC 2543) at 68.

679. Any proxy that handles and forwards an INVITE message must add its address to the Via field. Ex. 1033 (RFC 2543) at 17. The Via field is used to route responses back through all the appropriate servers, which ensures the responses can be sent through compliant firewalls. Ex. 1033 (RFC 2543) at 17.

680. The caller transmits the INVITE message by either sending it to a local proxy server that will handle locating the callee or by sending it to the SIP server associated with the callee's domain. Ex. 1033 (RFC 2543) at 13.

681. SIP servers must be able to use an IP-based protocol (*i.e.*, UDP and TCP) to transport messages. Ex. 1033 (RFC 2543) at 20 ("Proxy, registrar, and redirect servers **MUST** implement both UDP and TCP transport"). The caller or the server can choose to use either UDP or TCP based on the particular needs of the call:

In an Internet context, SIP is able to utilize both UDP and TCP as transport protocols, among others. UDP allows the application to more carefully control the timing of messages and their retransmission, to perform parallel searches without requiring TCP connection state for each outstanding request, and to use multicast.

Routers can more readily snoop SIP UDP packets. TCP allows easier passage through existing firewalls.

Ex. 1033 (RFC 2543) at 18; *see also id.* at 20 (SIP can be used with “ATM AAL5, IPX, frame relay or X.25”).

682. SIP servers also can be implemented on top of other transport protocols. Ex. 1033 (RFC 2543) at 18 (“SIP makes minimal assumptions about the underlying transport and network-layer protocols. The lower-layer can provide either a packet or a byte stream service, with reliable or unreliable service.”).

683. Where a caller initiates a call by sending the INVITE message to the SIP server associated with the callee’s domain, it first must determine the IP address associated with the callee’s SIP server. To locate the callee’s SIP server, the caller will query a “DNS server for address records for the host portion of the Request-URI. If the DNS server returns no address records, the client stops, as it has been unable to locate a server.” Ex. 1033 (RFC 2543) at 13-14.

684. In requesting the IP address for the callee’s SIP server, RFC 2543 describes two different options that the client can use. The client can send a message to a DNS server for that domain requesting the IP address of the SIP server (*e.g.*, requesting the IP address of sip.domainname or sip.example.com). Ex. 1033 (RFC 2543) at 13-14.

685. RFC 2543 also discloses that the SIP server can be located using the process specified in RFC 2052, which involves sending a message to the DNS server that requests the IP address of the server for that domain's SIP service. Ex. 1033 (RFC 2543) at 14, 146-148. RFC 2543 (which was adopted in March 1999) notes that RFC 2052 was not a fully ratified standard yet, but that it was expected that "better DNS techniques" like those specified in RFC 2052 would become widely available soon. Ex. 1033 (RFC 2543) at 14. I note that the IETF adopted RFC 2052 as a ratified standard in February 2000. Ex. 1082 (RFC 2052) at 1.

686. After receiving the IP address returned by the DNS server, the caller will use the address to send the INVITE message to the callee's SIP server. Ex. 1033 (RFC 2543) at 13.

687. Where a caller initiates a call by sending the INVITE message to a local SIP proxy server, the proxy server uses the same process as the caller to obtain an IP address for the callee. Ex. 1033 (RFC 2543) at 10 (defining a proxy server as "[a]n intermediary program **that acts as both a server and a client** for the purpose of making requests on behalf of other clients"), 98 ("A stateful proxy acts as a virtual [user agent]"). Because the proxy server must send the INVITE message using an IP protocol (*i.e.*, UDP or TCP), the proxy server must resolve the SIP URL into an IP address. *See* Ex. 1033 (RFC 2543) at 14-15 ("Once the host part has been resolved to a SIP server, the client sends one or more SIP requests to

that server and receives one or more responses from the server . . . If TCP is used, request and responses within a single SIP transaction are carried over the same TCP connection (see Section 10).”).

688. After the callee’s SIP server receives the INVITE message, it will determine whether the callee is available by querying a location server. If the callee has registered with the server, the location server will return the callee’s current address to the SIP server. Ex. 1033 (RFC 2543) at 15-17, 19.

689. If the callee’s SIP server is a proxy server, it will forward the INVITE message to the callee at its present address. Ex. 1033 (RFC 2543) at 15-17, 19. If the address returned by the location server contains a domain name (and not an IP address), the SIP server first must obtain the callee’s IP address by querying a DNS server before sending the message to the callee.

690. If the callee’s SIP server is a redirection server, it will return the address of the callee to the caller (or the caller’s local proxy server). The caller (or the caller’s local proxy server) then will send another INVITE message directly to the callee. Ex. 1033 (RFC 2543) at 15-17, 19.

691. When the callee receives the INVITE message, it will return one or more response messages that contains a status indicator. Ex. 1033 (RFC 2543) at 15-17, 19. The response will be routed back through each of the proxies identified in the Via header of the INVITE message. Ex. 1033 (RFC 2543) at 17.

692. A status indicator is a 3-digit code. There are several categories of status indicators including: Informational (1xx), *e.g.*, “Trying”, “Ringing”, “Call Is Being Forwarded”; Success (2xx); Redirect (3xx), *e.g.*, “Moved Temporarily” or “Moved Permanently”; and Error (4xx, 5xx, or 6xx). Ex. 1033 (RFC 2543) at 37-38; *see id.* at 75-84.

693. The callee can accept the call by returning a response indicating “200 OK”. Ex. 1033 (RFC 2543) at 76. If it accepts the call, the callee will return a session description with the response message that specifies the domain name or IP address where data can be sent, and the port numbers for any media streams it wishes to accept. Ex. 1033 (RFC 2543) at 123 (“the response must contain a description of where to send the data”), 140 (in the session description, the “‘c’ destination” field lists the address where media can be sent), 15 (“The INVITE request typically contains a session description, for example written in SDP (RFC 2327 [6]) format, that provides the called party with enough information to join the session. . . . For a unicast session, **the session description enumerates the media types and formats that the caller is willing to use and where it wishes the media data to be sent.** In either case, **if the callee wishes to accept the call,** it responds to the invitation **by returning a similar description listing the media it wishes to use.”); *see also* Ex. 1035 (RFC 2327) at 7-9 (showing the “c=” or “connection”**

attribute as specifying the IP address or domain name where data can be sent), 12-14.

694. For example, consider the scenario where a user named Bell calls a user named Watson and specifies a single audio stream. One valid response to the INVITE message would be the following:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP kton.bell-tel.com
From: A. Bell <sip:a.g.bell@bell-tel.com>
To: <sip:watson@bell-tel.com> ;tag=37462311
Call-ID: 3298420296@kton.bell-tel.com
CSeq: 1 INVITE
Contact: sip:watson@boston.bell-tel.com
Content-Type: application/sdp
Content-Length: ...

v=0
o=watson 4858949 4858949 IN IP4 192.1.2.3
s=I'm on my way
c=IN IP4 boston.bell-tel.com
m=audio 5004 RTP/AVP 0 3
```

Ex. 1033 (RFC 2543) at 124. Here, Watson is accepting the call by listing in the first line the success status indicator (“200 OK”). In the session description at the bottom of the message, Watson is specifying that he can receive data at “boston.bell-tel.com” and that audio data should be sent to port 5004. Ex. 1033 (RFC 2543) at 125. Instead of listing a domain name, Watson could have listed an IP address instead. *See* Ex. 1033 (RFC 2543) at 106 (showing an example where “c=IN IP4 135.180.144.94”). If Watson wished to specify additional media types

that he could receive, he could have specified additional media streams by including additional “m=” entries. *See* Ex. 1033 (RFC 2543) at 15, 132-33 (showing examples of messages with multiple media streams).

695. If the caller listed multiple media streams, the callee would indicate for each one whether it accepted or rejected the stream. To indicate it accepts a stream, the callee will specify a valid port number where it can receive data of that type; to reject a stream, the callee will specify a port number of zero. Ex. 1033 (RFC 2543) at 137. After the call setup process is complete, the caller will send any session data to the network address (*e.g.*, the IP address or domain name plus port number) or addresses specified in the callee’s response.

696. The callee also can reject the call. For example, the callee can decline to answer by returning a response indicating “603 Decline”. As another example, if the callee cannot support the call parameters specified in the INVITE, it will decline to answer by returning a response indicating “606 Not Acceptable”. Ex. 1033 (RFC 2543) at 84, 131-132.

697. If the SIP server cannot locate a callee, it will determine the callee is unavailable for communications and return a response indicating “404 Not Found”. Ex. 1033 (RFC 2543) at 78.

698. If the original caller receives a response message containing a success or redirection status code, the original caller will respond with an ACK message



acknowledging receipt of the response. Ex. 1033 (RFC 2543) at 15-17, 19, 29. If the response message indicated that the callee accepted the call (*i.e.*, it contained a “200 (OK)” status indicator), the ACK message may contain a final SDP description to be used for the call. Ex. 1033 (RFC 2543) at 29 (“The ACK request MAY contain a message body with the final session description to be used by the callee. If the ACK message body is empty, the callee uses the session description in the INVITE request.”).

699. After the ACK message has been received, the call initiation process is complete and the parties can exchange data. *See* Ex. 1033 (RFC 2543) at 15. Data streams will be sent through the channels specified in the final SDP description. Ex. 1033 (RFC 2543) at 123-25, 136-38 (data streams will be sent to the addresses and ports specified the SDP description, and will be formatted and processed as specified in the SDP description).

700. The call will continue until it is terminated. For a two-party call, either caller or callee can terminate the call by transmitting a BYE message. Ex. 1033 (RFC 2543) at 30 (“A party receiving a BYE request MUST cease transmitting media streams specifically directed at the party issuing the BYE request”). For a multi-party call, any party can leave the call by transmitting a BYE message. Ex. 1033 (RFC 2543) at 30.

701. The BYE message is transmitted via SIP servers using the same forwarding process used for transmitting INVITE messages. Ex. 1033 ([RFC 2543](#)) at 30.

702. [RFC 2543](#) has two figures which depict examples of the message flow associated with transmitting an INVITE message.

703. The first figure, Figure 1, depicts the process where the caller transmits the message to the callee's SIP server and where that server is a proxy server.

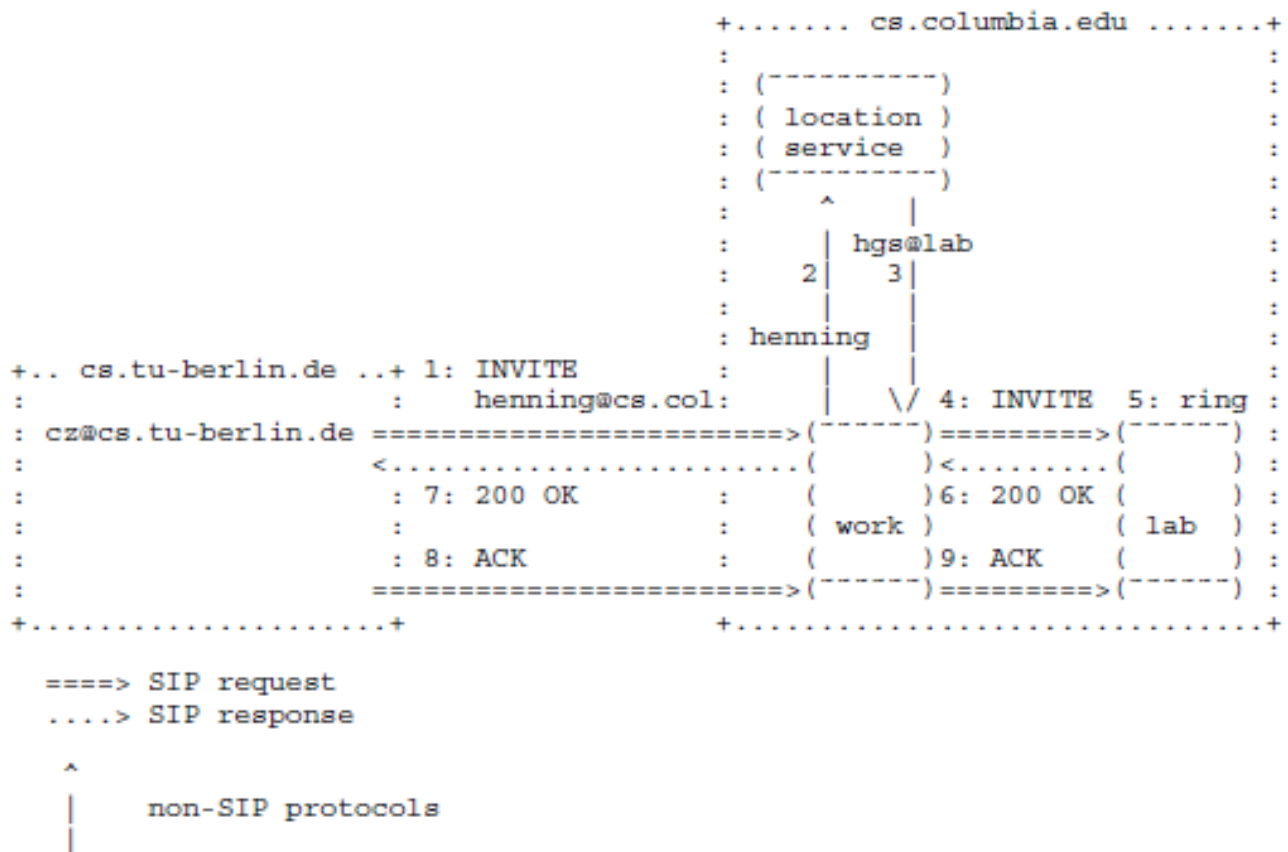


Figure 1: Example of SIP proxy server

Ex. 1033 (RFC 2543) at 16.

704. In step 1, the caller sends an INVITE message to SIP proxy server at cs.columbia.edu requesting to initiate a call with a user named “henning.”

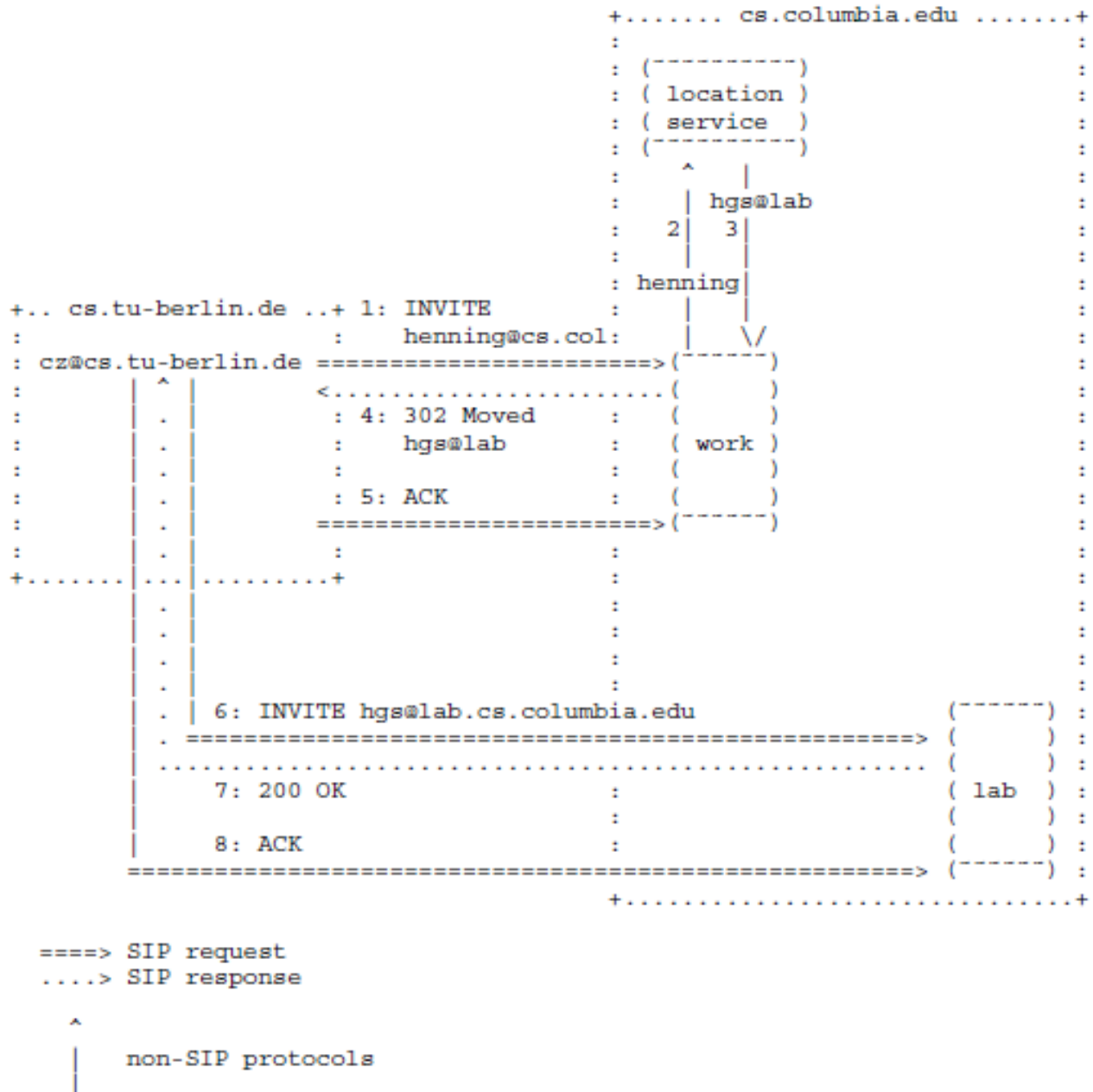
Although not depicted, the caller includes in the INVITE message a session description which specifies the types of media to be transmitted, and optionally, the sessions keys to be used for transmission. In step 2, the SIP server queries a location server to determine whether henning is available, and where henning is located.

705. In step 3, the location server informs the proxy that henning is available and can be found by sending a message to the address “hgs@lab”. Because the location service returned a URL and not an IP address, the proxy server inherently must resolve the URL into an IP address before transmitting the INVITE to the callee in step 4 because the proxy server transmits data using an IP protocol (*i.e.*, UDP or TCP). After the proxy obtains the IP address for hgs@lab, it transmits the INVITE message.

706. In step 5, the phone rings and the call is accepted. In step 6, the callee returns a SIP response indicating “200 OK”, which means the call has been accepted. Although not depicted, the callee can include in the SIP response a session description specifying the call parameters it is willing to accept. In step 7, the SIP proxy server forwards the response back to the caller. In step 8, the caller

transmits an ACK message accepting the call. In step 9, the callee receives the ACK message, and the call has been established.

707. In Figure 2, RFC 2543 shows the same process as in Figure 1 except the SIP server associated with the callee's domain is a redirect server.

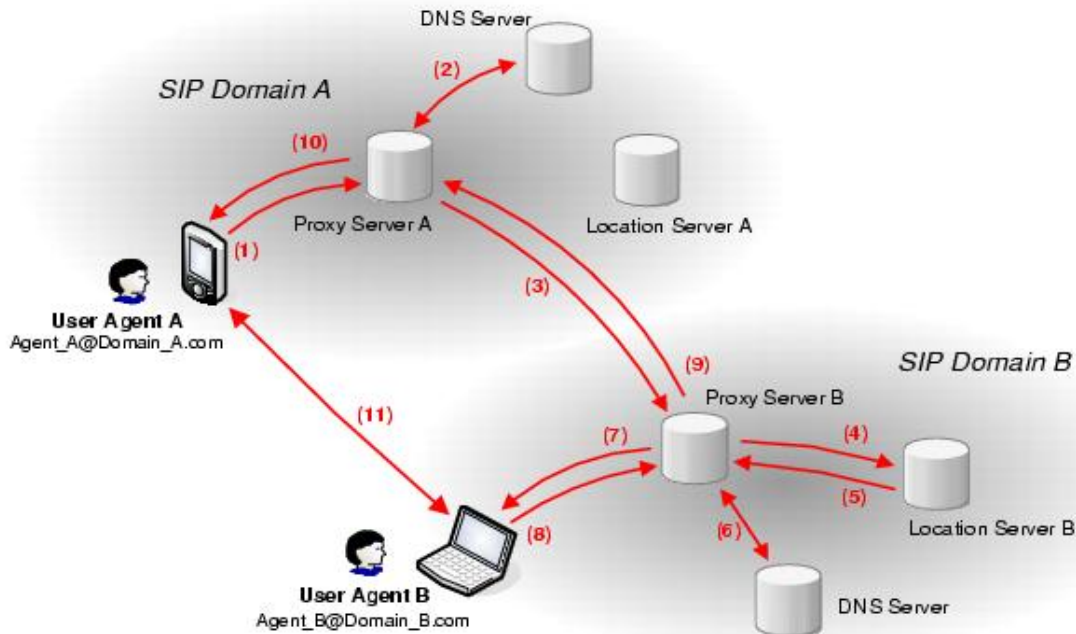


Ex. 1033 (RFC 2543) at 19.

708. Where the callee's SIP server is a redirect server, the message exchange is similar to where it is a proxy server. Except, instead of forwarding the INVITE message to the callee after the callee's location is determined, the redirect server informs the caller of the callee's location by sending a SIP response containing a "302 Moved" status code along with the callee's present address. Although not depicted, if the callee was not registered with the location server (in other words, the callee was offline), the redirect server would have returned a "404 Not Found" status code instead.

709. Although both figures depict the scenario where the caller handles establishing the call, the process would be similar where the caller uses a local SIP proxy server. The main difference is that within the box labeled "cs.tu-berlin.de" would be a proxy server, just like the "work" proxy server in the "cs.columbia.edu" box of figure 1, that handled transmitting the messages for the caller (*i.e.*, "cz@cs.tu-berlin.de").

710. The figure below shows how the process would work where the user used a local SIP proxy server. Although the figure is from a recent publication by IBM, it depicts one of the SIP protocol's basic call setup processes, which is the same process specified in RFC 2543.



Ex. 1081 (IBM SIP) at 4.

711. In step 1, the caller transmits an INVITE message to a local proxy server. In step 2, the proxy server identifies the SIP server associated with the callee's domain by querying a DNS server. In step 3, the caller's proxy server uses the IP address to transmit the INVITE message to the callee's SIP server.

712. In step 4, the callee's SIP server asks a location server whether the callee is available. If the callee is available, the location server will return the SIP address registered by the callee. If the response does not contain an IP address, in step 6 the callee's SIP server the resolves the SIP address by querying a DNS server. In step 7, the proxy server sends the INVITE message to the callee.

713. In step 8, the callee accepts the call, and returns a SIP response to its SIP server. In steps 9 and 10, the callee's SIP server sends the response to the

caller's SIP server, which sends the response to caller. Although not depicted, the caller would acknowledge the response by sending an ACK message to the callee via the two proxy servers. In step 11, the caller and the callee exchange data.

### **Encryption and Call Handling Features**

714. The INVITE message “typically contains a session description, for example written in SDP (RFC 2327) format, that provides the called party with enough information to join the session.” Ex. 1033 (RFC 2543) at 15. The description will contain information such as the different types of media to be transmitted, the format of the media, and the minimum bandwidth necessary to participate in the call. Ex. 1033 (RFC 2543) at 123-124. This description will be transmitted in the “message body” of the INVITE message. Ex. 1033 (RFC 2543) at 15.

715. The callee will include in its response message a session description listing its capabilities. Ex. 1033 (RFC 2543) at 36, 76, 84 (“2xx responses to INVITE requests contain session descriptions”), 136-137. If the callee can support all of the features specified in the INVITE message, it will return the same session description. If it cannot, it will return a session description identifying the features specified in the INVITE message that it does support. *See* Ex. 1033 (RFC 2543) at 123-124, 136-137.

716. If the callee informed the caller that it cannot support all of the specified features, when the caller returns the ACK message, it will contain a session description listing the capabilities to be used during the call. Ex. 1033 (RFC 2543) at 29.

717. If the callee cannot support any of the capabilities specified by the caller, it can reject the call, and ask the caller to propose different session parameters by sending a response with a particular error status indicator code (*e.g.*, 606 Not Acceptable). Ex. 1033 (RFC 2543) at 84, 131-132; *see also* ¶ 723, *below*.

718. Another example of a session parameter that can be specified in the INVITE message is the encryption key for the session content. RFC 2543 explains “[t]he SIP message body MAY also contain encryption keys for the session itself.” Ex. 1033 (RFC 2543) at 104-05. Any multimedia transferred during a subsequent call would be encrypted with those keys. *See* Ex. 1034 (RFC 1889) at 9 (“Protocols and mechanisms that may be needed in addition to RTP to provide a usable service. In particular, for multimedia conferences, a control protocol may distribute multicast addresses and keys for encryption, [and] negotiate the encryption algorithm to be used . . .”), 49-50 (“Confidentiality means that only the intended receiver(s) can decode the received packets . . . Confidentiality of the content is achieved by encryption.”).



719. RFC 2543 explains that the SIP messages themselves also may be encrypted. SIP supports three types of encryption for use in call signaling:

- o End-to-end encryption of the SIP message body and certain sensitive header fields;
- o hop-by-hop encryption to prevent eavesdropping that tracks who is calling whom;
- o hop-by-hop encryption of Via fields to hide the route a request has taken.

Ex. 1033 (RFC 2543) at 104-05.

720. If part of a SIP message has been encrypted, the “Encryption” header field will specify the type of encryption that has been used. Ex. 1033 (RFC 2543) at 26-27, 54-55. If a SIP message is encrypted, the message body and certain sensitive header fields will be encrypted. Ex. 1033 (RFC 2543) at 104-05; *id.* at 40-42 (showing that most header fields can be encrypted). If an INVITE message is encrypted by the caller or a proxy, it would be encrypted with the public key of the entity specified in the “To” field of the message. Ex. 1033 (RFC 2543) at 54.

721. The INVITE message can contain an encryption key to be used to encrypt subsequent SIP messages. The key would be transmitted in a header field called “Response-Key.” Ex. 1033 (RFC 2543) at 54, 63-64.

722. To ensure the confidentiality of the SIP messages and call parameters, the caller could insist that the callee return an encrypted response. Ex. 1033 (RFC

2543) at 63-64. All SIP messages sent during the duration of the call would be encrypted.

723. If the callee cannot support encryption and an INVITE message specifies that encryption must be used, the callee must return a response indicating an error code. Ex. 1033 (RFC 2543) at 63-64.

If the client insists that the server return an encrypted response, it includes a 'Require: org.ietf.sip.encrypt-response' header field in its request. If the server cannot encrypt for whatever reason, it **MUST** follow normal Require header field procedures and return a 420 (Bad Extension) response. If this Require header field is not present, a server **SHOULD** still encrypt if it can.

Ex. 1033 (RFC 2543) at 64. If a SIP server determines that the callee cannot support encryption, it will return an error code.

724. If the session description sent in the INVITE message specifies the encryption key to be used to encrypt the session data, the message would need to be encrypted to ensure the key remained confidential. Ex. 1033 (RFC 2543) at 104-05.

725. RFC 2543 also explains that a proxy server may be configured to automatically make changes to the SIP messages it processes to ensure that all communications are encrypted:

### 13.1.3 Encryption by Proxies

Normally, proxies are not allowed to alter end-to-end header fields and message bodies. Proxies MAY, however, encrypt an unsigned request or response with the key of the call recipient. **Proxies need to encrypt a SIP request if the end system cannot perform encryption or to enforce organizational security policies.**

Ex. 1033 (RFC 2543) at 108.

726. RFC 2543 also explains that a proxy server may ensure security by encrypting messages using IPsec:

#### 13.1.4 Hop-by-Hop Encryption

SIP requests and responses MAY also be protected by security mechanisms at the transport or network layer. No particular mechanism is defined or recommended here. Two possibilities are **IPSEC** or TLS. The use of a particular mechanism will generally need to be specified out of band, through manual configuration, for example.

Ex. 1033 (RFC 2543) at 108.

727. RFC 2543 explains that IPsec, which is defined in RFC 2401, may be used to encrypt SIP message or other data transmitted between the caller and the callee. Ex. 1033 (RFC 2543) at 108; *see id.* at 109 (“hop-by-hop authentication can be provided, for example, by the IPSEC Authentication Header”). I described IPsec in more detail in ¶¶ 361-371, above.

728. RFC 2543 also explains that hop-by-hop encryption hides the identity of the caller and the callee, enabling calls to be established anonymously. Ex.

1033 (RFC 2543) at 105 (“Hop-by-hop encryption encrypts the entire SIP request or response on the wire so that packet sniffers or other eavesdroppers cannot see who is calling whom”).

729. A clients or SIP proxy server may request that its address be hidden from future recipients of a message by adding a “Hide” request header to a message. Ex. 1033 (RFC 2543) at 57-58, 71.

730. SIP provides for authentication. Ex. 1033 (RFC 2543) at 109-119. RFC 2543 shows that the callee’s proxy server can require a caller to authenticate before it will process an INVITE message. Ex. 1033 (RFC 2543) at 45, 39 (proxy server can return a “401 Unauthorized” error code). Another way SIP provides authentication is through the use of IPsec, which allows for both end-to-end authentication and hop-by-hop authentication. Ex. 1033 (RFC 2543) at 109 (“hop-by-hop authentication can be provided, for example, by the IPSEC Authentication Header”).

## **2. Comparison of RFC 2543 to Claims 1-29 of the ’181 Patent**

### **a. Claim 1**

731. RFC 2543 describes methods and systems for establishing multimedia sessions between a caller and callee. *See* ¶¶ 640-644, 656-660, 675, *above*.

732. RFC 2543 explains that SIP users are identified using SIP URLs. A SIP URL may be of the form “user@domain name”. *See* ¶¶ 663-665, *above*. The

“user” can be a user name or a phone number. *See* ¶¶ 663-664, *above*; *see* Ex. 1033 (RFC 2543) at 20-24.

733. RFC 2543 shows that each SIP user also is associated with an IP address or a domain name. *See* ¶¶ 693-694, *above*. RFC 2543 shows that, when establishing a call, each SIP user must specify the IP address or domain name, as well as the port number, that should be used to send it session (*e.g.*, multimedia) data. *See* ¶¶ 693-695, *above*. Therefore, RFC 2543 shows that a callee is associated with an unsecure name such as domain name and an IP address.

734. To start the process for initiating a call, the caller creates an INVITE message that contains the callee’s SIP URL. *See* ¶¶ 663, 675-678, *above*. The caller or the caller’s SIP proxy server will transmit the INVITE message to a SIP server associated with the callee’s domain. *See* ¶¶ 683-687, *above*.

735. RFC 2543 explains that the callee’s SIP server will receive the INVITE message, and then determine whether the callee is available by querying a location server using the callee’s SIP URL. *See* ¶¶ 666-671, 688, *above*. If the callee has registered its availability, the location server will return the IP address or domain name where the user can be reached.

736. The callee’s SIP URL and user name are a secure names because they are non-standard names that can only be resolved by a SIP server or a SIP location server.

737. RFC 2543 explains that a SIP user name can be a phone number. During its prosecution of the '181 patent, Patent Owner represented a “‘secure name’ can be a secure non-standard domain name, such as a secure non-standard top-level domain name (e.g., .scom) or a telephone number.” See ¶ 212, *above*. Therefore, a SIP user name is a secure name.

738. If the callee has registered its location with the server, it will receive the INVITE message. The callee will return a response message indicating whether it accepts the call. If the caller receives a response message indicating that callee accepted the call, the caller will confirm establishment of the call by returning an ACK message to the callee. See ¶¶ 698-699, *above*.

739. The caller and callee then can exchange session (*e.g.*, multimedia) data. See ¶¶ 699-700, *above*. The result of a SIP transaction is the establishment of a connection between a caller and a callee. See ¶¶ 699-700, *above*.

740. RFC 2543 thus shows “*A non-transitory machine-readable medium comprising instructions for a method of communicating with a first device associated with a secure name and an unsecured name,*” as specified by claim 1.

741. RFC 2543 explains that to start the process for initiating a call, the caller creates an INVITE message. The caller addresses the message to a callee by putting the callee’s SIP address in the “To” and the “Request-URI” header fields. See ¶¶ 663, 675-678, *above*.

742. In the INVITE message, the caller will specify the parameters for the call by including a session description in the message body of the INVITE message. *See ¶¶ 644-656, 676, 714, 718, 724, above.* For example, the caller can request that the call include both audio and video data, or just one or other. *See ¶¶ 645-647, above.*

743. Another parameter that can be specified in the session description is the encryption key to be used to encrypt session data. *See ¶¶ 644-656, 676, 714, 718, 724, above.* If the session description contains an encryption key, the INVITE message will be encrypted with a public key associated with callee. *See ¶¶ 714-724, above.* To indicate that the SIP message is encrypted, the caller will set the “Encryption” header field of the INVITE message. *See ¶¶ 645-653, above.*

744. Depending on how the SIP environments are configured, there are several ways the INVITE message could be transmitted to the callee.

745. If the caller does not have a local SIP server, the caller will locate the SIP server associated with the callee’s domain by querying a DNS server. *See ¶¶ 681-687, above.* The caller then will send the INVITE message to an IP address returned by the DNS server. *See ¶¶ 683-687, above.*

746. If the caller has a local SIP server, the INVITE message would be sent through the local server, which will handle setting up the call for the caller. *See ¶¶ 677, 680, above.* The local SIP proxy server will locate the SIP server

associated with the callee's domain by querying a DNS server. *See ¶¶ 681-687, above.* The proxy server then will send the INVITE message to an IP address returned by the DNS server. *See ¶¶ 683-687, above.*

747. Next, the callee's SIP server will receive the INVITE message. The callee's SIP server will determine whether the callee is available by querying a location server using callee's SIP URL. *See ¶¶ 688-697, above.* If the callee has registered, the location server will return the callee's address.

748. If the callee is available and the callee's SIP server is a proxy server, it will forward the INVITE message to the callee. *See ¶¶ 688-690, 697, above.* If it is a SIP redirect server, it will return the address of the callee to the caller. The caller then will send the INVITE message to the callee. *See ¶¶ 688-697, 703-706, 710-713, above.*

749. The callee will receive the INVITE message at the address it registered with the location server. *See ¶¶ 688-691, 693, above.* The callee then can decide whether it would like to accept the call. Therefore, a message indicating a desire to securely communicate is received at an address associated with the callee.

750. RFC 2543 thus shows “*receiving, at a network address corresponding to the secure name associated with the first device, a message from a second*



*device of the desired to securely communicate with the first device,” as specified by claim 1.*

751. After the callee receives the INVITE message, it can decide whether it wants to accept the call. If the callee accepts the call, it will return a response message to the caller containing a success status indicator code. *See ¶ 693, above.* If it accepts, the callee also will specify an IP address or domain name at which it can receive data. *See ¶¶ 693-695, above.*

752. If the INVITE message specified that encryption must be used, the callee will encrypt the response. *See ¶¶ 714-724, above.* The transmission of the encrypted response satisfies sending a message over a secure communication link from the first to the second device.

753. The caller receives the response message from the callee. If the message indicates the callee has accepted the call, the caller returns an ACK message to the callee to confirm acceptance of the call. *See ¶¶ 698-699, above.* If the caller specified that SIP messages should be encrypted, the caller will encrypt the ACK message. *See ¶¶ 714-724, above.*

754. After the callee receives the ACK message, the call is established and the parties can exchange session data according the parameters specified during this process. *See ¶¶ 699-700, above.* If the session description specified an encryption key to use to encrypt session data, the caller and callee will encrypt the

audio and video data streams. *See* ¶¶ 645-649, 699, 714-730, *above*. The transmission of the encrypted session data satisfies sending a message over a secure communication link from the first to the second device.

755. RFC 2543 thus shows “*sending a message over a secure communication link from the first device to the second device*,” as specified by claim 1.

756. Accordingly, RFC 2543 anticipates claim 1 of the ’181 patent under 35 U.S.C. § 102(e).

757. If it is found both that RFC 2543 does not completely describe the encryption process and that RFC 2543, RFC 1889, and RFC 2327 (which are incorporated by reference into RFC 2543) cannot be considered as a single disclosure, it would have been obvious to a person of ordinary skill in the art in April 2000 to combine the teachings of these RFCs. As RFC 2543 explains: “SIP is designed as part of the overall IETF multimedia data and control architecture currently incorporating protocols such as . . . **the real-time transport protocol (RTP) (RFC 1889 [3]) for transporting real-time data** and providing QOS feedback . . ., and **the session description protocol (SDP) (RFC 2327 [6]) for describing multimedia sessions**. Ex. 1033 (RFC 2543) at 8. RFC 2543 thus explicitly discloses that it is designed to be used in conjunction with the protocols described in RFC 1889 and RFC 2327.

758. RFC 2543 also explicitly discloses that SIP messages include session description data structures that are compliant with the SDP protocol specified in RFC 2327. RFC 2543 even includes an appendix that summarizes how to use the SDP protocol in SIP messages. Ex. 1033 (RFC 2543) at 136-140. RFC 2327 shows that it can be used to distribute encryption keys to encrypt session data. *See* ¶¶ 645-649, *above*. RFC 2543 also shows SIP being used to initiate a call involving RTP data streams, which are defined by RFC 1889. Ex. 1033 (RFC 2543) at 137. RFC 1889 shows that RTP data can be encrypted. *See* ¶ 718, *above*. I believe a person of ordinary skill in February of 2000 would have found it obvious to consider RFC 2543, RFC 1889, and RFC 2327 together.

759. A person would have considered RFC 2543 with RFC 2401 because RFC 2543 refers to the IPsec standard defined in RFC 2401. *See* ¶¶ 726-727, 730, *above*. In addition, it was common practice to add extra security functionality to existing systems by integrating protocols such as IPsec into edge routers and proxy servers. *See* ¶¶ 190-194, *above*.

760. The guidance in RFC 2401 would have specifically suggested utilizing a transparent proxy server to ensure that organizational security policies were enforced. *See* ¶¶ 369-371, *above*. This is also specifically suggested by RFC 2543. *See* ¶¶ 725-727, 730, *above*. A person of ordinary skill in the art could have configured a local firewall to automatically apply IPsec to any network traffic

being transmitted between corporate networks, or to any outbound network traffic. In this configuration, IPsec could be used to encrypt both SIP messages and session data transmitted between the caller and the callee.

**b. Claim 2**

761. RFC 2543 describes methods and systems for establishing multimedia sessions between a caller and callee. *See ¶¶ 640-644, 656-660, 675, above.*

762. RFC 2543 explains that SIP users are identified using SIP URLs. A SIP URL may be of the form “user@domain name”. *See ¶¶ 663-665, above.* The “user” can be a user name or a phone number. *See ¶¶ 663-664, above; see Ex. 1033 (RFC 2543) at 20-24.*

763. RFC 2543 shows that each SIP user also is associated with an IP address or a domain name. *See ¶¶ 693-694, above.* RFC 2543 shows that, when establishing a call, each SIP user must specify the IP address or domain name, as well as the port number, that should be used to send it session or multimedia data. *See ¶¶ 693-695, above.* Therefore, RFC 2543 shows that a callee is associated with an unsecure name such as domain name and an IP address.

764. To start the process for initiating a call, the caller creates an INVITE message that contains the callee’s SIP URL. *See ¶¶ 663, 675-678, above.* The caller or the caller’s SIP proxy server will transmit the INVITE message to a SIP server associated with the callee’s domain. *See ¶¶ 683-687, above.*

765. RFC 2543 explains that the callee's SIP server will receive the INVITE message, and then determine whether the callee is available by querying a location server using the callee's SIP URL. *See ¶¶ 666-671, 688, above.* If the callee has registered its availability, the location server will return the IP address or domain name where the user can be reached.

766. The callee's SIP URL and user name are a secure names because they are non-standard names that can only be resolved by a SIP server.

767. RFC 2543 explains that a SIP user name can be a phone number. During its prosecution of the '181 patent, Patent Owner represented a "secure name" can be a secure non-standard domain name, such as a secure non-standard top-level domain name (e.g., .scom) or a telephone number." *See ¶ 212, above.* Therefore, a SIP user name is a secure name.

768. If the callee is available, it will receive the INVITE message. The callee will return a response message indicating whether it accepts the call. If the caller receives a response message indicating that callee accepted the call, the caller will confirm establishment of the call by returning an ACK message to the callee. *See ¶¶ 698-699, above.*

769. The caller and callee then can exchange session (e.g., multimedia) data. *See ¶¶ 699-700, above.* The result of a SIP transaction is the establishment of a connection between a caller and a callee. *See ¶¶ 699-700, above.*

770. RFC 2543 thus shows “*A method of using a first device to communicate with a second device having a secure name*” as specified by the preamble of claim 2.

771. RFC 2543 explains that to start the process for initiating a call, the caller creates an INVITE message. The caller addresses the message to a callee by putting the callee’s SIP address in the “To” and the “Request-URI” header fields. *See ¶¶ 663, 675-678, above.*

772. In the INVITE message, the caller will specify the parameters for the call by including a session description in the message body of the INVITE message. *See ¶¶ 644-656, 676, 714, 718, 724, above.* For example, the caller can request that the call include both audio and video data, or just one or other. *See ¶¶ 645-647, above.*

773. Another parameter that can be specified in the session description is the encryption key to be used to encrypt session data. *See ¶¶ 644-656, 676, 714, 718, 724, above.* If the session description contains an encryption key, the INVITE message will be encrypted with a public key associated with callee. *See ¶¶ 714-724, above.* To indicate that the SIP message is encrypted, the caller will set the “Encryption” header field of the INVITE message. *See ¶¶ 645-653, above.*

774. SIP can be configured so the caller does not have a local SIP proxy server. In this configuration, the caller will locate the callee and transmit the

INVITE message directly to the callee. The caller will locate the SIP server associated with the callee's domain by querying a DNS server. *See ¶¶ 681-687, above.* The caller then will send the INVITE message to the callee's SIP server by sending it to an IP address returned by the DNS server. *See ¶¶ 683-687, above.*

775. The callee's SIP server will receive the INVITE message. The callee's SIP server will determine whether the callee is available by querying a location server using the callee's SIP URL. *See ¶¶ 688-697, above.* If the callee's SIP server is a redirect server and the location server returns an address for the callee, the SIP server will return the address to the caller. *See ¶¶ 688-697, 703-706, 710-713, above.* If the callee's SIP server is a proxy server and the location server returns an address for the callee, the SIP server will forward the INVITE message to the callee. *See ¶¶ 688-697, 703-710, above.* If the callee decides to accept the call, it will return a response message via the SIP server that includes a session description that contains a network address (*e.g.*, a domain name or IP address), along with a port number, where the callee can receive session data. *See ¶¶ 693-695, above.* RFC 2543 shows that authentication can be required before the callee's proxy server will process an INVITE message. *See ¶ 730, above.* In this configuration, the callee's SIP server is a secure name service that resolves the callee's SIP URL for the caller.

776. RFC 2543 thus shows “*from the first device, sending a message to a secure name service, the message requesting a network address associated with the secure name of the second device,*” as specified in claim 2.

777. If the callee’s SIP server is a redirect server, after the callee’s SIP server receives the INVITE message, it returns the address of the callee to the caller. *See ¶¶ 688-697, 703-706, 710-713, above.* Therefore, the caller receives a message containing the network address associated with the SIP URL. The caller then will send the INVITE message to the callee. *See ¶¶ 688-697, 703-706, 710-713, above.*

778. If the callee’s SIP server is a proxy server and the location server returns an address for the callee, the SIP server will forward the INVITE message to the callee. *See ¶¶ 688-697, 703-710, above.*

779. In either scenario, the callee will receive the INVITE message at the address it registered with the location server. *See ¶¶ 688-691, 693, above.* The callee then can decide whether it would like to accept the call. If the callee accepts the call, it will return a response message that contains a success status indicator code. *See ¶ 693, above.* The response message also will include a session description that contains a network address (*e.g.*, a domain name or IP address), along with a port number, where the callee can receive session data. *See ¶¶ 693-695, above.*



780. The response message containing the domain name or IP address where the callee can receive session data also is a message containing the network address associated with the SIP URL.

781. The caller will receive the response message. If the message indicates the callee has accepted the call, the caller returns an ACK message to the callee to confirm acceptance of the call. *See ¶¶ 698-699, above.*

782. RFC 2543 thus shows “*at the first device, receiving a message containing the network address associated with the secure name of the second device,*” as specified in claim 2.

783. If the caller specifies in the INVITE message that all SIP messages should be encrypted, the INVITE message, the callee’s response, and the ACK message each would be encrypted. Where a SIP message is encrypted, the message body and most header fields will be encrypted. *See ¶¶ 719-724, above.*

784. If the callee accepts the call, it will return a response message that contains a success status indicator code. *See ¶ 693, above.* The caller will receive the response message. If the message indicates the callee has accepted the call, the caller returns an ACK message to the callee to confirm acceptance of the call. *See ¶¶ 698-699, above.* If the caller specified that SIP messages should be encrypted, the ACK would be encrypted. Therefore, the transmission of the encrypted INVITE message or ACK message from the caller to the callee satisfies sending a

message to the the network address associated with the secure name of the second device using a secure communication link.

785. After the callee receives the ACK message, the call is established and the parties can exchange session (*e.g.*, multimedia) data according the parameters specified during this process. *See ¶¶ 699-700, above.*

786. The caller will send session data to the address that the callee listed in its response message. *See ¶¶ 693-695, above.*

787. If the session description specified an encryption key to use to encrypt session data, the caller and callee will encrypt the audio and video data streams. *See ¶¶ 645-649, 699, 714-730, above.* The transmission of encrypted session data from the caller to the callee also satisfies sending a message to the the network address associated with the secure name of the second device using a secure communication link.

788. RFC 2543 thus shows “*from the first device, sending a message to the network address associated with the secure name of the second device using a secure communication link,*” as specified in claim 2.

789. Accordingly, RFC 2543 anticipates claim 2 of the ’181 patent under 35 U.S.C. § 102(e).

790. If it is found both that RFC 2543 does not completely describe the encryption process and that RFC 2543, RFC 1889, and RFC 2327 (which are

incorporated by reference into RFC 2543) cannot be considered as a single disclosure, it would have been obvious to a person of ordinary skill in the art in April 2000 to combine the teachings of these RFCs. As RFC 2543 explains: “SIP is designed as part of the overall IETF multimedia data and control architecture currently incorporating protocols such as . . . **the real-time transport protocol (RTP) (RFC 1889 [3]) for transporting real-time data** and providing QOS feedback . . . , and **the session description protocol (SDP) (RFC 2327 [6]) for describing multimedia sessions**. Ex. 1033 (RFC 2543) at 8. RFC 2543 thus explicitly discloses that it is designed to be used in conjunction with the protocols described in RFC 1889 and RFC 2327.

791. RFC 2543 also explicitly discloses that SIP messages include session description data structures that are compliant with the SDP protocol specified in RFC 2327. RFC 2543 even includes an appendix that summarizes how to use the SDP protocol in SIP messages. Ex. 1033 (RFC 2543) at 136-140. RFC 2327 shows that it can be used to distribute encryption keys to encrypt session data. *See* ¶¶ 645-649, *above*. RFC 2543 also shows SIP being used to initiate a call involving RTP data streams, which are defined by RFC 1889. Ex. 1033 (RFC 2543) at 137. RFC 1889 shows that RTP data can be encrypted. *See* ¶ 718, *above*. I believe a person of ordinary skill in February of 2000 would have found it obvious to consider RFC 2543, RFC 1889, and RFC 2327 together.

792. A person would have considered RFC 2543 with RFC 2401 because RFC 2543 refers to the IPsec standard defined in RFC 2401. See ¶¶ 726-727, 730, *above*. In addition, it was common practice to add extra security functionality to existing systems by integrating protocols such as IPsec into edge routers and proxy servers. See ¶¶ 190-194, *above*.

793. The guidance in RFC 2401 would have specifically suggested utilizing a transparent proxy server to ensure that organizational security policies were enforced. See ¶¶ 369-371, *above*. This is also specifically suggested by RFC 2543. See ¶¶ 725-727, 730, *above*. A person of ordinary skill in the art could have configured a local firewall to automatically apply IPsec to any network traffic being transmitted between corporate networks, or to any outbound network traffic. In this configuration, IPsec could be used to encrypt both SIP messages and session data transmitted between the caller and the callee.

**c. Claims 3-4**

794. RFC 2543 explains that SIP users are identified using SIP URLs. A SIP URL may be of the form “user@domain name”. See ¶ 663, *above*. The “user” can be a user name or a phone number. See ¶¶ 663-664, *above*; see Ex. 1033 (RFC 2543) at 20-24.

795. The callee’s SIP URL is a secure domain name because it is a specialized, non-standard domain name that can only be resolved by a SIP server.

796. RFC 2543 thus shows “*The method according to claim 2, wherein the secure name of the second device is a secure domain name,*” as specified by claim 3.

797. Use of a SIP URL indicates that a secure, non-standard name is being used to establish a connection and therefore indicates security.

798. RFC 2543 thus shows “*The method according to claim 2, wherein the secure name indicates security,*” as specified by claim 4.

799. If it is found that RFC 2543 does not describe use of a secure domain name, I believe a person of ordinary skill in April of 2000 would have found using a secure domain name as part of a SIP URL to have been obvious based on the guidance in RFC 2543 and Kiuchi. Kiuchi shows the use of non-standard domain names to identify target computers that are located within a closed network. *See* ¶¶ 979-981, *below*.

800. A person would have considered RFC 2543 with Kiuchi because both are describing secure communication techniques and systems. *See* ¶¶ 645-649, 699, 714-724, *above*, ¶¶ 972-977, *below*. Both also are describing methods that can utilize specialized proxy servers to establish a connection between devices on two different local networks. *See* ¶¶ 659, 680-682, 687-691, *above*, ¶¶ 975-982, 997, *below*. Both describe processes where an entity that desires to participate in

the network need to register its name and address with a server. *See* ¶¶ 666-671, *above*, ¶¶ 978, 985, *below*.

801. Use of a non-standard domain name would indicate security. Kiuchi shows that an origin server and its server-side proxy are associated with a secure hostname resolvable by the C-HTTP name server. *See* ¶¶ 978-980, *above*. The hostname can be an unconventional domain name such as “Coordinating.Center.CSCRG,” which includes the non-standard top-level domain name “CSCRG.” *See* ¶ 979, *above*. Use of a non-standard domain name as part of a SIP URL would indicate security.

**d. Claim 5**

802. RFC 2543 shows that SIP messages can be encrypted. *See* ¶¶ 650-653, 719-728, *above*. If the caller specified that SIP messages must be encrypted, any response to an INVITE message returned by the callee’s SIP server or the callee will be encrypted. The response message contains the network address where the callee can receive session data. *See* ¶¶ 693-695, *above*.

803. RFC 2543 thus shows “*The method according to claim 2, wherein receiving the message containing the network address associated with the secure name of the second device includes receiving the message in encrypted form,*” as specified by claim 5.

**e. Claim 6**

804. RFC 2543 shows that SIP messages can be encrypted. If the caller or the callee receives an encrypted SIP message or data packet, it necessarily will decrypt it so it can read the encrypted parts of the message.

805. RFC 2543 thus shows “*The method according to claim 5, further including decrypting the message,*” as specified by claim 6.

**f. Claim 7**

806. As I explained above, the call initiation process starts with the caller transmitting an INVITE message. In the INVITE message, the caller will specify the parameters for the call by including a session description in the message body of the INVITE message. *See ¶¶ 644-656, 676, 714, 718, 724, above.* For example, the caller can request that the call include both audio and video data, or just one or other. *See ¶¶ 645-647, above.*

807. Another parameter that can be specified in the session description is the encryption key to be used to encrypt session data. *See ¶¶ 644-656, 676, 714, 718, 724, above.* If the session description contains an encryption key, the INVITE message will be encrypted with a public key associated with callee. *See ¶¶ 714-724, above.* To indicate that the SIP message is encrypted, the caller will set the “Encryption” header field of the INVITE message. *See ¶¶ 645-653, above.*

808. If the callee accepts the call and does not specify alternative call parameters, the parties can exchange data according the parameters specified in the INVITE message. *See ¶¶ 699-700, above.*

809. The caller will send session data to the network address (*e.g.*, domain name or IP address) that the callee listed in its response message. *See ¶¶ 693-695, above.*

810. If the session description specified individual encryption keys to use to encrypt session data, the caller and callee will encrypt the audio and video data streams using the appropriate key. *See ¶¶ 645-649, 699, 714-724, above.*

811. If the session description did not specify an encryption key to be used, the audio and video data streams will be sent in the clear. Therefore, a call could include data transmitted over a secure, encrypted link and also data transmitted in the clear.

812. RFC 2543 thus shows “*The method according to claim 2, wherein the second device is capable of supporting a secure communication link as well as a non-secure communication link, the method further including establishing a non-secure communication link with the second device when needed,*” as specified by claim 7.

**g. Claim 8**



813. After a call has been established, the caller will send multimedia data to the callee by sending it to the network address and port number the callee specified in its response message. See ¶¶ 693-695, *above*. RFC 2543 shows that the callee can specify its address as either a domain name or an IP address. See ¶¶ 693-695, *above*.

814. RFC 2543 thus shows “*The method according to claim 2, wherein receiving a message containing the network address associated with the secure name of the device includes receiving the network address as an IP address associated with the secure name of the device,*” as specified by claim 8.

#### **h. Claim 9**

815. After the callee receives the ACK message, the call initiation process is complete, and the call parameters have been established. See ¶¶ 698-699, *above*. The caller then will send session (*e.g.*, multimedia) data to the callee by sending it to the network address and port number the callee specified in its response message. See ¶¶ 693-695, *above*.

816. If the call parameters included an encryption key to use to encrypt session data, all data sent between the caller and callee will be encrypted. This encryption happens automatically, without any additional input from the user. See ¶¶ 645-649, 699, 714-724, *above*.

817. RFC 2543 thus shows “*The method according to claim 2, further including automatically initiating the secure communication link after it is enabled,*” as specified by claim 9.

**i. Claim 10**

818. RFC 2543 explains that SIP messages can be protected by using the IPsec standard defined in RFC 2401. See ¶¶ 726-727, 730, *above*.

819. RFC 2401 explains that IPsec is a tunneling protocol, and packets transmitted via IPsec are tunneled. See ¶¶ 363-369, *above*.

820. RFC 2543 therefore shows that where IPsec is used to protect SIP messages, the caller and callee would receive messages via a tunnel.

821. RFC 2543 thus shows “*The method according to claim 2, wherein receiving a message containing the network address associated with the secure name of the device includes receiving the message at the first device through tunneling within the secure communication link,*” as specified by claim 10.

**j. Claim 11**

822. As I explained with respect to claim 10, RFC 2543 explains that SIP messages can be protected by using the IPsec standard defined in RFC 2401. See ¶¶ 726-727, 730, *above*. IPsec is a tunneling protocol, and packets transmitted via IPsec are tunneled. See ¶¶ 363-369, *above*.

823. RFC 2543 therefore shows that where IPsec is used to protect SIP messages, the caller and callee would receive packets via a tunnel.

824. RFC 2543 thus shows “*The method according to claim 2, wherein receiving a message containing the network address associated with the secure name of the device includes receiving the message in the form of at least one tunneled packet,*” as specified by claim 11.

**k. Claim 12**

825. RFC 2543 explains that SIP messages can be transmitted using TCP or UDP, and that SIP can be implemented using many other protocols. *See* ¶ 681, *above*; Ex. 1033 (RFC 2543) at 20 (SIP also can be used with “ATM AAL5, IPX, frame relay or X.25”).

826. RFC 2543 thus shows “*The method according to claim 2, wherein the receiving and sending of messages includes receiving and sending the messages in accordance with any one of a plurality of communication protocols,*” as specified by claim 12.

**l. Claim 13**

827. RFC 2543 explains that one call between parties can include multiple session data streams. *See* ¶¶ 647-649, 654-657, 660-662, 681-683, 694-696, 721, *above*. For example, a call could include both an audio and video stream, and both

sessions could be encrypted with the same key. *See* ¶¶ 647-649, 654-657, 660-662, 681-683, 694-696, 721, *above*.

828. RFC 2543 thus shows “*The method according to claim 2, wherein the receiving and sending of messages through the secure communication link includes multiple sessions,*” as specified by claim 13.

**m. Claims 14-17**

829. RFC 2543 explains that one call between parties can include multiple session data streams. *See* ¶¶ 647-649, 654, 694, 695-661, *above*. For example, a call could include both an audio and video stream, and both sessions could be encrypted with the same or different keys. *See* ¶¶ 647-649, 654, 694, 695-661, 721, *above*.

830. RFC 2543 thus shows “*The method according to claim 2, further including supporting a plurality of services over the secure communication link,*” as specified by claim 14.

831. RFC 2543 thus shows “*The method according to claim 14, wherein the plurality of services comprises a plurality of communication protocols, a plurality of application programs, multiple sessions, or a combination thereof,*” as specified by claim 15.

832. RFC 2543 thus shows “*The method according to claim 15, wherein the plurality of application programs comprises video conferencing, e-mail, a*

*word processing program, telephony or a combination thereof,”* as specified by claim 16.

833. RFC 2543 thus shows “*The method according to claim 15, wherein the plurality of services comprises audio, video or a combination thereof,*” as specified by claim 17.

**n. Claim 18**

834. RFC 2543 explains that SIP messages can be protected by using the IPsec standard defined in RFC 2401. See ¶¶ 726-727, 730, *above*.

835. RFC 2543 shows that authentication can be required before a connection will be established. For example, it shows that IPsec can be used to authenticate a message either hop-by-hop or end-to-end. See ¶ 730, *above*.

836. RFC 2543 thus shows “*The method according to claim 2, wherein the secure communication link is an authenticated link,*” as specified by claim 18.

**o. Claims 19-20**

837. RFC 2543 shows the caller can be a computer that is connected to a communication network. See ¶¶ 672-674, *above*. Where the caller does not have a local SIP proxy server, all the steps of claim 2 are performed by the caller. See ¶¶ 680-686, 707-709, *above*.

838. RFC 2543 thus shows “*The method according to claim 2, wherein the first device is a computer, and the steps are performed on the computer,*” as specified by claim 19.

839. RFC 2543 thus shows “*The method according to claim 2, wherein the first device is a client computer connected to a communication network, and the method is performed by the client computer on the communication network,*” as specified by claim 20.

**p. Claim 21**

840. RFC 2543 explains that SIP users are identified using SIP URLs. A SIP URL may be of the form “user@domain name”. See ¶ 663, *above*. The “user” can be a user name or a phone number. See ¶¶ 663-664, *above*; see Ex. 1033 (RFC 2543) at 20-24.

841. RFC 2543 shows that each SIP user also is associated with an IP address or a domain name. See ¶¶ 693-694, *above*. RFC 2543 shows that, when establishing a call, each SIP user must specify the IP address or domain name, as well as the port number, that should be used to send it session or multimedia data. See ¶¶ 693-695, *above*. Therefore, RFC 2543 shows that a callee is associated with an unsecure name such as domain name and an IP address.

842. RFC 2543 thus shows “*The method according to claim 2, further including providing an unsecured name associated with the device,*” as specified by claim 21.

**q. Claim 22**

843. The SIP protocol includes a registration process that allows a user to move between a number of different locations or end systems while still being able to receive calls addressed to its SIP address. *See ¶¶ 666-671, above.*

844. Users wishing to be able to receive calls register their availability with a SIP server by sending a REGISTER request message to the server. In the REGISTER message, the user will specify its SIP address and the current address at which it can be reached. *See ¶¶ 666-671, above.*

845. The SIP server will record the user’s location in a location server. The location server will store the “address-of-record” for the user, which “typically [is] of the form ‘user@domain’ rather than ‘user@host’.” Ex. 1033 (RFC 2543) at 32. The “user” can be a user name or a phone number. *See ¶¶ 663-664, above; see Ex. 1033 (RFC 2543) at 20-24.*

846. If a user changes location, it can send another REGISTER message to the SIP server to change the user’s address to reflect the user’s new location. Ex. 1033 (RFC 2543) at 31-34. If a user signs off, it can cancel a registration by sending a message to the SIP server. Ex. 1033 (RFC 2543) at 33.

847. For a callee to be able to receive calls, the callee necessarily must have registered its location with a location server. Where a callee successfully receives an INVITE message transmitted by a caller, the callee would have registered its location before the caller transmitted the INVITE message. See ¶¶ 692, 708, *above*.

848. RFC 2543 thus shows “*The method according to claim 2, wherein the secure name is registered prior to the step of sending a message to a secure name service,*” as specified by claim 22.

**r. Claim 23**

849. RFC 2543 explains that SIP users are identified using SIP URLs. A SIP URL may be of the form “user@domain name”. See ¶¶ 663-665, *above*. The “user” can be a user name or a phone number. See ¶¶ 663-664, *above*; see Ex. 1033 (RFC 2543) at 20-24.

850. The callee’s SIP URL is a secure domain name because it is a specialized, non-standard domain name that can only be resolved by a SIP server.

851. RFC 2543 thus shows “*The method according to claim 2, wherein the secure name of the second device is a secure, non-standard domain name,*” as specified by claim 23.

852. If it is found that RFC 2543 does not describe use of a secure, non-standard domain name, I believe a person of ordinary skill in April of 2000 would



have found using a secure, non-standard domain name as part of a SIP URL to have been obvious based on the guidance in RFC 2543 and Kiuchi for the same reasons as with respect to claims 3 and 4. *See* ¶¶ 799-801, *above*.

**s. Claim 24**

853. RFC 2543 describes methods and systems for establishing multimedia sessions between a caller and callee. *See* ¶¶ 640-644, 656-660, 675, *above*.

854. RFC 2543 explains that sessions and SIP communications may be encrypted. *See* ¶¶ 645-649, 699, 714-724, *above*.

855. RFC 2543 thus shows “*A method of using a first device to securely communicate with a second device over a communication network,*” as specified in the preamble of claim 24.

856. The SIP protocol includes a registration process that allows a user to move between a number of different locations or end systems while still being able to receive calls sent to its SIP address. *See* ¶¶ 666-671, *above*.

857. Users wishing to be able to receive calls register their availability with a SIP server by sending a REGISTER request message to the server. In the REGISTER message, the user will specify its SIP address and the current address at which it can be reached. *See* ¶¶ 666-671, *above*.

858. The SIP server will record the user’s location in a location server. The location server will store the “address-of-record” for the user, which “typically

[is] of the form ‘user@domain’ rather than ‘user@host’.” Ex. 1033 (RFC 2543) at 32. The “user” portion of a SIP URL can be a user name or a phone number. *See* ¶¶ 663-664, *above*; *see* Ex. 1033 (RFC 2543) at 20-24.

859. If a user changes location, it can send another REGISTER message to the SIP server to change the user’s address to reflect the user’s new location. Ex. 1033 (RFC 2543) at 31-34. If a user signs off, it can cancel a registration by sending a message to the SIP server. Ex. 1033 (RFC 2543) at 33.

860. RFC 2543 thus shows “*at the first device requesting and obtaining registration of a secure name for the first device, the secure name being associated with a network address*” as specified by claim 24.

861. RFC 2543 explains that to start the process for initiating a call, the caller creates an INVITE message. The caller addresses the message to a callee by putting the callee’s SIP address in the “To” and the “Request-URI” header fields. *See* ¶¶ 663, 675-678, *above*.

862. In the INVITE message, the caller will specify the parameters for the call by including a session description in the message body of the INVITE message. *See* ¶¶ 644-656, 676, 714, 718, 724, *above*. For example, the caller can request that the call include both audio and video data, or just one or other. *See* ¶¶ 645-647, *above*.

863. Another parameter that can be specified in the session description is the encryption key to be used to encrypt session data. *See ¶¶ 644-656, 676, 714, 718, 724, above.* If the session description contains an encryption key, the INVITE message will be encrypted with a public key associated with callee. *See ¶¶ 714-724, above.* To indicate that the SIP message is encrypted, the caller will set the “Encryption” header field of the INVITE message. *See ¶¶ 645-653, above.*

864. SIP can be configured so the caller does not have a local SIP proxy server and the callee’s SIP proxy server is a redirect server. In this configuration, the caller will locate the callee and transmit the INVITE message directly to the callee. The caller will locate the SIP server associated with the callee’s domain by querying a DNS server. *See ¶¶ 681-687, above.* The caller then will send the INVITE message to the callee’s SIP server by sending it to an IP address returned by the DNS server. *See ¶¶ 683-687, above.*

865. The callee’s SIP server will receive the INVITE message. The callee’s SIP server will determine whether the callee is available by querying a location server for the callee’s user name. *See ¶¶ 688-697, above.* Because the callee’s SIP server is a redirect server, if the location server returns an address for the callee, the SIP server will return the address to the caller. *See ¶¶ 688-697, 703-706, 710-713, above.*

866. After the caller receives the callee's address, the caller sends the INVITE message directly to the callee. *See ¶¶ 688-697, 703-706, 710-713, above.*

867. The callee will receive the INVITE message at the address it registered with the location server. *See ¶¶ 688-691, 693, above.* The callee then can decide whether it would like to accept the call. Therefore, a message indicating a desire to securely communicate is received at an address associated with the callee.

868. RFC 2543 thus shows “*receiving at the network address associated with the secure name of the first device a message from a second device of the desire to securely communicate with the first device*” as specified by claim 24.

869. If the caller specifies in the INVITE message that all SIP messages should be encrypted, the INVITE message, the callee's response, and the ACK message each would be encrypted.

870. If the callee accepts the call, it will return a response message that contains a success status indicator code. *See ¶ 693, above.* The transmission of the encrypted reply message from the callee to the caller satisfies sending a message securely from the first to the second device.

871. The caller will receive the response message. If the message indicates the callee has accepted the call, the caller returns an ACK message to the callee to

confirm acceptance of the call. *See* ¶¶ 698-699, *above*. If the caller specified that SIP message should be encrypted, the ACK message would be encrypted.

872. After the callee receives the ACK message, the call is established and the parties can exchange session data according the parameters specified during this process. *See* ¶¶ 699-700, *above*.

873. The caller will send session data to the network address and port number that the callee listed in its response message. *See* ¶¶ 693-695, *above*. If the session description specified an encryption key to use to encrypt session data, the caller and callee will encrypt the audio and video data streams. *See* ¶¶ 645-649, 699, 714-730, *above*. The transmission of the encrypted session data from the callee to the caller satisfies sending a message securely from the first to the second device.

874. RFC 2543 thus shows “*sending a message securely from the first device to the second device*” as specified by claim 24.

875. Accordingly, RFC 2543 anticipates claim 24 of the ’181 patent under 35 U.S.C. § 102(e).

876. If it is found both that RFC 2543 does not completely describe the encryption process and that RFC 2543, RFC 1889, and RFC 2327 (which are incorporated by reference into RFC 2543) cannot be considered as a single disclosure, it would have been obvious to a person of ordinary skill in the art in

April 2000 to combine the teachings of these RFCs. As RFC 2543 explains: “SIP is designed as part of the overall IETF multimedia data and control architecture currently incorporating protocols such as . . . **the real-time transport protocol (RTP) (RFC 1889 [3]) for transporting real-time data** and providing QOS feedback . . ., and **the session description protocol (SDP) (RFC 2327 [6]) for describing multimedia sessions**. Ex. 1033 (RFC 2543) at 8. RFC 2543 thus explicitly discloses that it is designed to be used in conjunction with the protocols described in RFC 1889 and RFC 2327.

877. RFC 2543 also explicitly discloses that SIP messages include session description data structures that are compliant with the SDP protocol specified in RFC 2327. RFC 2543 even includes an appendix that summarizes how to use the SDP protocol in SIP messages. Ex. 1033 (RFC 2543) at 136-140. RFC 2327 shows that it can be used to distribute encryption keys to encrypt session data. *See* ¶¶ 645-649, *above*. RFC 2543 also shows SIP being used to initiate a call involving RTP data streams, which are defined by RFC 1889. Ex. 1033 (RFC 2543) at 137. RFC 1889 shows that RTP data can be encrypted. *See* ¶ 718, *above*. I believe a person of ordinary skill in February of 2000 would have found it obvious to consider RFC 2543, RFC 1889, and RFC 2327 together.

878. A person would have considered RFC 2543 with RFC 2401 because RFC 2543 refers to the IPsec standard defined in RFC 2401. *See* ¶¶ 726-727, 730,

*above*. In addition it was common practice to add extra security functionality to existing systems by integrating protocols such as IPsec into edge routers and proxy servers. *See* ¶¶ 190-194, *above*.

879. The guidance in RFC 2401 would have specifically suggested utilizing a transparent proxy server to ensure that organizational security policies were enforced. *See* ¶¶ 369-371, *above*. This is also specifically suggested by RFC 2543. *See* ¶¶ 725-727, 730, *above*. A person of ordinary skill in the art could have configured a local firewall to automatically apply IPsec to any network traffic being transmitted between corporate networks, or to any outbound network traffic. In this configuration, IPsec could be used to encrypt both SIP messages and session data transmitted between the caller and the callee.

**t. Claim 25**

880. As I explained above, the SIP protocol includes a registration process that allows a user to move between a number of different locations or end systems while still being able to receive calls addressed to its SIP address. *See* ¶¶ 666-671, *above*. When a user registers its location, the REGISTER message is sent from the user's computer or device.

881. RFC 2543 thus shows “*The method according to claim 24, wherein requesting and obtaining registration of a secure name for the first device*

*comprises using the first device to obtain a registration of the secure name for the first device.”*

882. After the callee receives the ACK message, the call is established and the parties can exchange session data according the parameters specified during this process. *See ¶¶ 699-700, above.*

883. The caller will send session data to the network address and port number that the callee listed in its response message. *See ¶¶ 693-695, above.*

884. If the session description specified an encryption key to use to encrypt session data, the caller and callee will encrypt the audio and video data streams. *See ¶¶ 645-649, 699, 714-724, above.*

885. RFC 2543 thus shows “*The method according to claim 24 . . . wherein sending a message securely comprises sending the message from the first device to the second device using a secure communication link*” as specified in claim 25.

886. Accordingly, RFC 2543 anticipates claim 25 of the ’181 patent under 35 U.S.C. § 102(e).

**u. Claim 26**

887. RFC 2543 describes methods and systems for establishing multimedia sessions between a caller and callee. *See ¶¶ 640-644, 656-660, 675, above.*



888. RFC 2543 thus shows “*A method of using a first device to communicate with a second device over a communication network,*” as specified in the preamble of claim 26.

889. RFC 2543 explains that SIP users are identified using SIP URLs. A SIP URL may be of the form “user@domain name”. See ¶¶ 663-665, *above*. The “user” can be a user name or a phone number. See ¶¶ 663-664, *above*; see Ex. 1033 (RFC 2543) at 20-24.

890. RFC 2543 shows that each SIP user also is associated with an IP address or a domain name. See ¶¶ 693-694, *above*. RFC 2543 shows that, when establishing a call, each SIP user must specify the IP address or domain name, as well as the port number, that should be used to send it session (*e.g.*, multimedia) data. See ¶¶ 693-695, *above*. Therefore, RFC 2543 shows that a callee is associated with an unsecure name such as domain name and an IP address.

891. RFC 2543 allows a SIP user to use a domain name as the address at which it can receive session data. For example, a callee may provide a domain name as the address at which it can receive session data. In order to be able to receive messages at that address, the callee necessarily must have registered that domain name with a public DNS server. See ¶¶ 91-110, 127-133, *above*. Otherwise, the caller would not be able to resolve the domain name into an IP

address and would be unable to send IP packets to the callee. *See* ¶¶ 693-695, *above*; *see also* ¶¶ 105-111, 91-103, *above* (background).

892. The claim specifies that an unsecured name is registered. I interpret the claim using the plain and ordinary meaning of the term “register” as it relates to domain names and name services. When an entity registers a domain name with a name server, it specifies the name and one or more addresses to be associated with that name. *See* ¶¶ 91-110, 127-133, *above*.

893. RFC 2543 thus shows “*from the first device requesting and obtaining registration of an unsecured name associated with the first device*” as specified by claim 26.

894. The SIP protocol includes a registration process that allows a user to move between a number of different locations or end systems while still being able to receive calls addressed to its SIP address. *See* ¶¶ 666-671, *above*.

895. Users wishing to be able to receive calls register their availability with a SIP server by sending a REGISTER request message to the server. In the REGISTER message, the user will specify its SIP address and the current address at which it can be reached. *See* ¶¶ 666-671, *above*.

896. The SIP server will record the user’s location in a location server. The location server will store the “address-of-record” for the user, which “typically [is] of the form ‘user@domain’ rather than ‘user@host’.” Ex. 1033 (RFC 2543) at

32. The “user” can be a user name or a phone number. *See* ¶¶ 663-664, *above*; *see* Ex. 1033 (RFC 2543) at 20-24.

897. If a user changes location, it can send another REGISTER message to the SIP server to change the user’s address to reflect the user’s new location. Ex. 1033 (RFC 2543) at 31-34. If a user signs off, it can cancel a registration by sending a message to the SIP server. Ex. 1033 (RFC 2543) at 33.

898. RFC 2543 thus shows “*from the first device requesting and obtaining registration of a secure name associated with the first device, wherein a unique network address corresponds to the secure name associated with the first device*” as specified by claim 26.

899. RFC 2543 explains that to start the process for initiating a call, the caller creates an INVITE message. The caller addresses the message to a callee by putting the callee’s SIP address in the “To” and the “Request-URI” header fields. *See* ¶¶ 663, 675-678, *above*.

900. In the INVITE message, the caller will specify the parameters for the call by including a session description in the message body of the INVITE message. *See* ¶¶ 644-656, 676, 714, 718, 724, *above*. For example, the caller can request that the call include both audio and video data, or just one or other. *See* ¶¶ 645-647, *above*.

901. Another parameter that can be specified in the session description is the encryption key to be used to encrypt session data. *See ¶¶ 644-656, 676, 714, 718, 724, above.* If the session description contains an encryption key, the INVITE message will be encrypted with a public key associated with callee. *See ¶¶ 714-724, above.* To indicate that the SIP message is encrypted, the caller will set the “Encryption” header field of the INVITE message. *See ¶¶ 645-653, above.*

902. After a caller (or the caller’s SIP proxy server) transmits an INVITE message to the callee’s SIP server, the server will attempt to locate the callee by querying a location server using the callee’s SIP URL. *See ¶¶ 688-697, above.* If the callee has registered with the server as being available, the location server will return the address the callee registered. *See ¶¶ 666-671, 688, above.*

903. The callee’s SIP server then will either forward the INVITE message to the callee at the registered address or return the address to the caller. *See ¶¶ 666-671, 688, above.*

904. The callee will receive the INVITE message at the address it registered with the location server. *See ¶¶ 688-691, 693, above.* The callee then can decide whether it would like to accept the call. Therefore, a message indicating a desire to securely communicate is received at an address associated with the callee.

905. RFC 2543 thus shows “*receiving at the unique network address associated with the secure name a message from a second device requesting the desire to securely communicate with the first device*” as specified by claim 26.

906. If the callee accepts the call, it will return a response message that contains a success status indicator code. *See ¶ 693, above.* The transmission of the encrypted reply message from the callee to the caller satisfies sending a message securely from the first to the second device.

907. The caller will receive the response message. If the message indicates the callee has accepted the call, the caller returns an ACK message to the callee to confirm acceptance of the call. *See ¶¶ 698-699, above.* If the caller specified that SIP message should be encrypted, the ACK message would be encrypted. Where a SIP message is encrypted, the message body and most header fields will be encrypted. *See ¶¶ 719-720, above.*

908. After the callee receives the ACK message, the call is established and the parties can exchange session data according the parameters specified during this process. *See ¶¶ 699-700, above.*

909. The caller will send session data to the address that the callee listed in its response message. *See ¶¶ 693-695, above.* If the session description specified an encryption key to use to encrypt session data, the caller and callee will encrypt the audio and video data streams. *See ¶¶ 645-649, 699, 714-730, above.* The

transmission of the encrypted session data from the callee to the caller satisfies sending a message securely from the first to the second device.

910. RFC 2543 thus shows “*from the first device sending a message securely from the first device to the second device*” as specified by claim 26.

911. Accordingly, RFC 2543 anticipates claim 26 of the ’181 patent under 35 U.S.C. § 102(e).

912. If it is found both that RFC 2543 does not completely describe the encryption process and that RFC 2543, RFC 1889, and RFC 2327 (which are incorporated by reference into RFC 2543) cannot be considered as a single disclosure, it would have been obvious to a person of ordinary skill in the art in April 2000 to combine the teachings of these RFCs. As RFC 2543 explains: “SIP is designed as part of the overall IETF multimedia data and control architecture currently incorporating protocols such as . . . **the real-time transport protocol (RTP) (RFC 1889 [3]) for transporting real-time data** and providing QOS feedback . . . , and **the session description protocol (SDP) (RFC 2327 [6]) for describing multimedia sessions**. Ex. 1033 (RFC 2543) at 8. RFC 2543 thus explicitly discloses that it is designed to be used in conjunction with the protocols described in RFC 1889 and RFC 2327.

913. RFC 2543 also explicitly discloses that SIP messages include session description data structures that are compliant with the SDP protocol specified in

RFC 2327. RFC 2543 even includes an appendix that summarizes how to use the SDP protocol in SIP messages. Ex. 1033 (RFC 2543) at 136-140. RFC 2327 shows that it can be used to distribute encryption keys to encrypt session data. *See* ¶¶ 645-649, *above*. RFC 2543 also shows SIP being used to initiate a call involving RTP data streams, which are defined by RFC 1889. Ex. 1033 (RFC 2543) at 137. RFC 1889 shows that RTP data can be encrypted. *See* ¶ 718, *above*. I believe a person of ordinary skill in February of 2000 would have found it obvious to consider RFC 2543, RFC 1889, and RFC 2327 together.

914. A person would have considered RFC 2543 with RFC 2401 because RFC 2543 refers to the IPsec standard defined in RFC 2401. *See* ¶¶ 726-727, 730, *above*. In addition it was common practice to add extra security functionality to existing systems by integrating protocols such as IPsec into edge routers and proxy servers. *See* ¶¶ 190-194, *above*.

915. The guidance in RFC 2401 would have specifically suggested utilizing a transparent proxy server to ensure that organizational security policies were enforced. *See* ¶¶ 369-371, *above*. This is also specifically suggested by RFC 2543. *See* ¶¶ 725-727, 730, *above*. A person of ordinary skill in the art could have configured a local firewall to automatically apply IPsec to any network traffic being transmitted between corporate networks, or to any outbound network traffic.

In this configuration, IPsec could be used to encrypt both SIP messages and session data transmitted between the caller and the callee.

**v. Claim 27**

916. RFC 2543 explains that SIP users are identified using SIP URLs. A SIP URL may be of the form “user@domain name”. *See* ¶ 663, *above*. The “user” can be a user name or a phone number. *See* ¶¶ 663-664, *above*; *see* Ex. 1033 (RFC 2543) at 20-24.

917. RFC 2543 shows that each SIP user also is associated with an IP address or a domain name. *See* ¶¶ 693-694, *above*. RFC 2543 shows that, when establishing a call, each SIP user must specify the IP address or domain name, as well as the port number, that should be used to send it session or multimedia data. *See* ¶¶ 693-695, *above*. Therefore, RFC 2543 shows that a callee is associated with an unsecure name such as domain name and an IP address.

918. RFC 2543 allows a SIP user to use a domain name as the address at which it can receive session data. For example, a callee may provide a domain name as the address at which it can receive session data. In order to be able to receive messages at that address, the callee necessarily must have registered that domain name with a public DNS server. *See* ¶¶ 91-110, 127-133, *above*. Otherwise, the caller would not be able to resolve the domain name into an IP



address and would be unable to send IP packets to the callee. *See* ¶¶ 693-695, *above*; *see also* ¶¶ 105-111, 91-103, *above* (background).

919. RFC 2543 thus shows “*wherein requesting and obtaining registration of an unsecured name associated with the first device comprises using the first device to obtain a registration of the unsecured name associated with the first device,*” as specified by claim 27.

920. As I explained above, the SIP protocol includes a registration process that allows a user to move between a number of different locations or end systems while still being able to receive calls addressed to its SIP address. *See* ¶¶ 666-671, *above*. When a user registers its location, the REGISTER message is sent from the user’s computer or device.

921. RFC 2543 thus shows “*wherein requesting and obtaining registration of a secure name associated with the first device comprises using the first device to obtain a registration of the secure name associated with the first device,*” as specified by claim 27.

922. Accordingly, RFC 2543 anticipates claim 27 of the ’181 patent under 35 U.S.C. § 102(e).

**w. Claim 28**

923. RFC 2543 describes methods and systems for establishing multimedia sessions between a caller and callee. *See* ¶¶ 640-644, 656-660, 675, *above*.

924. RFC 2543 thus shows “*A non-transitory machine-readable medium comprising instructions,*” as specified by the preamble of claim 28.

925. RFC 2543 explains that to start the process for initiating a call, the caller creates an INVITE message. The caller addresses the message to a callee by putting the callee’s SIP address in the “To” and the “Request-URI” header fields. *See ¶¶ 663, 675-678, above.*

926. In the INVITE message, the caller will specify the parameters for the call by including a session description in the message body of the INVITE message. *See ¶¶ 644-656, 676, 714, 718, 724, above.* For example, the caller can request that the call include both audio and video data, or just one or other. *See ¶¶ 645-647, above.*

927. Another parameter that can be specified in the session description is the encryption key to be used to encrypt session data. *See ¶¶ 644-656, 676, 714, 718, 724, above.* If the session description contains an encryption key, the INVITE message will be encrypted with a public key associated with callee. *See ¶¶ 714-724, above.* To indicate that the SIP message is encrypted, the caller will set the “Encryption” header field of the INVITE message. *See ¶¶ 645-653, above.*

928. SIP can be configured so the caller does not have a local SIP proxy server. In this configuration, the caller will locate the callee and transmit the INVITE message directly to the callee. The caller will locate the SIP server

associated with the callee's domain by querying a DNS server. *See* ¶¶ 681-687, *above*. The caller then will send the INVITE message to the callee's SIP server by sending it to an IP address returned by the DNS server. *See* ¶¶ 683-687, *above*.

929. The callee's SIP server will receive the INVITE message. The callee's SIP server will determine whether the callee is available by querying a location server using the callee's SIP URL. *See* ¶¶ 688-697, *above*. If the callee's SIP server is a redirect server and the location server returns an address for the callee, the SIP server will return the address to the caller. *See* ¶¶ 688-697, 703-706, 710-713, *above*. If the callee's SIP server is a proxy server and the location server returns an address for the callee, the SIP server will forward the INVITE message to the callee. *See* ¶¶ 688-697, 703-710, *above*. If the callee decides to accept the call, it will return a response message via the SIP server that includes a session description that contains a network address (*e.g.*, a domain name or IP address), along with a port number, where the callee can receive session data. *See* ¶¶ 693-695, *above*. RFC 2543 shows that authentication can be required before the callee's proxy server will process an INVITE message. *See* ¶ 730, *above*. In this configuration, the callee's SIP server is a secure name service that resolves the callee's SIP URL for the caller.

930. RFC 2543 thus shows “*sending a message to a secure name service, the message requesting a network address associated with a secure name of a device*” as specified by claim 28.

931. If the callee’s SIP server is a redirect server, after the callee’s SIP server receives the INVITE message, it returns the address of the callee to the caller. *See ¶¶ 688-697, 703-706, 710-713, above.* Therefore, the caller receives a message containing the network address associated with the SIP URL. The caller then will send the INVITE message to the callee. *See ¶¶ 688-697, 703-706, 710-713, above.*

932. If the callee’s SIP server is a proxy server and the location server returns an address for the callee, the SIP server will forward the INVITE message to the callee. *See ¶¶ 688-697, 703-710, above.*

933. In either scenario, the callee will receive the INVITE message at the address it registered with the location server. *See ¶¶ 688-691, 693, above.* The callee then can decide whether it would like to accept the call. Therefore, a message indicating a desire to securely communicate is received at an address associated with the callee.

934. After the callee receives the INVITE message, it can decide whether it wants to accept the call. If the callee accepts the call, it will return a response message that contains a success status indicator code. *See ¶ 693, above.* The

response message also will include a session description that contains a network address (*e.g.*, a domain name or IP address) and port number where the callee can receive session data. *See ¶¶ 693-695, above.*

935. The response message containing the domain name or IP address where the callee can receive session data also is a message containing the network address associated with the SIP URL.

936. The caller will receive the response message. If the message indicates the callee has accepted the call, the caller returns an ACK message to the callee to confirm acceptance of the call. *See ¶¶ 698-699, above.*

937. RFC 2543 thus shows “*receiving a message containing the network address associated with the secure name of the device*” as specified by claim 28.

938. If the caller specifies in the INVITE message that all SIP messages should be encrypted, the INVITE message, the callee’s response, and the ACK message each would be encrypted. Where a SIP message is encrypted, the message body and most header fields will be encrypted. *See ¶¶ 719-720, above.*

939. If the callee accepts the call, it will return a response message that contains a success status indicator code. *See ¶ 693, above.* The response message also will include a session description that contains a domain name or IP address where the callee can receive session data. *See ¶¶ 693-695, above.*

940. The caller will receive the response message. If the message indicates the callee has accepted the call, the caller returns an ACK message to the callee to confirm acceptance of the call. *See ¶¶ 698-699, above.* This message would be encrypted. Therefore, the transmission of the encrypted INVITE message or ACK message from the caller to the callee satisfies sending a message to the the network address associated with the secure name of the second device using a secure communication link.

941. After the callee receives the ACK message, the call is established and the parties can exchange session data according the parameters specified during this process. *See ¶¶ 699-700, above.*

942. The caller will send session data to the network address that the callee listed in its response message. *See ¶¶ 693-695, above.*

943. If the session description specified an encryption key to use to encrypt session data, the caller and callee will encrypt the audio and video data streams. *See ¶¶ 645-649, 699, 714-730, above.* The transmission of encrypted session data from the caller to the callee also satisfies sending a message to the the network address associated with the secure name of the second device using a secure communication link.

944. RFC 2543 thus shows “*sending a message to the network address associated with the secure name of the device using a secure communication link*” as specified by claim 28.

945. Accordingly, RFC 2543 anticipates claim 28 of the ’181 patent under 35 U.S.C. § 102(e).

946. If it is found both that RFC 2543 does not completely describe the encryption process and that RFC 2543, RFC 1889, and RFC 2327 (which are incorporated by reference into RFC 2543) cannot be considered as a single disclosure, it would have been obvious to a person of ordinary skill in the art in April 2000 to combine the teachings of these RFCs. As RFC 2543 explains: “SIP is designed as part of the overall IETF multimedia data and control architecture currently incorporating protocols such as . . . **the real-time transport protocol (RTP) (RFC 1889 [3]) for transporting real-time data** and providing QOS feedback . . ., and **the session description protocol (SDP) (RFC 2327 [6]) for describing multimedia sessions**. Ex. 1033 (RFC 2543) at 8. RFC 2543 thus explicitly discloses that it is designed to be used in conjunction with the protocols described in RFC 1889 and RFC 2327.

947. RFC 2543 also explicitly discloses that SIP messages include session description data structures that are compliant with the SDP protocol specified in RFC 2327. RFC 2543 even includes an appendix that summarizes how to use the

SDP protocol in SIP messages. Ex. 1033 (RFC 2543) at 136-140. RFC 2327 shows that it can be used to distribute encryption keys to encrypt session data. *See* ¶¶ 645-649, *above*. RFC 2543 also shows SIP being used to initiate a call involving RTP data streams, which are defined by RFC 1889. Ex. 1033 (RFC 2543) at 137. RFC 1889 shows that RTP data can be encrypted. *See* ¶ 718, *above*. I believe a person of ordinary skill in February of 2000 would have found it obvious to consider RFC 2543, RFC 1889, and RFC 2327 together.

948. A person would have considered RFC 2543 with RFC 2401 because RFC 2543 refers to the IPsec standard defined in RFC 2401. *See* ¶¶ 726-727, 730, *above*. In addition it was common practice to add extra security functionality to existing systems by integrating protocols such as IPsec into edge routers and proxy servers. *See* ¶¶ 190-194, *above*.

949. The guidance in RFC 2401 would have specifically suggested utilizing a transparent proxy server to ensure that organizational security policies were enforced. *See* ¶¶ 369-371, *above*. This is also specifically suggested by RFC 2543. *See* ¶¶ 725-727, 730, *above*. A person of ordinary skill in the art could have configured a local firewall to automatically apply IPsec to any network traffic being transmitted between corporate networks, or to any outbound network traffic. In this configuration, IPsec could be used to encrypt both SIP messages and session data transmitted between the caller and the callee.



**x. Claim 29**

950. RFC 2543 describes methods and systems for establishing multimedia sessions between a caller and callee. *See* ¶¶ 640-644, 656-660, 675, *above*.

951. RFC 2543 thus shows “A *non-transitory machine-readable medium comprising instructions for a method of communicating with a device having a secure name,*” as specified by the preamble of claim 29.

952. RFC 2543 explains that to start the process for initiating a call, the caller creates an INVITE message. The caller addresses the message to a callee by putting the callee’s SIP address in the “To” and the “Request-URI” header fields. *See* ¶¶ 663, 675-678, *above*.

953. In the INVITE message, the caller will specify the parameters for the call by including a session description in the message body of the INVITE message. *See* ¶¶ 644-656, 676, 714, 718, 724, *above*. For example, the caller can request that the call include both audio and video data, or just one or other. *See* ¶¶ 645-647, *above*.

954. Another parameters that can be specified in the session description is the encryption key to be used to encrypt session data. *See* ¶¶ 644-656, 676, 714, 718, 724, *above*. If the session description contains an encryption key, the INVITE message will be encrypted with a public key associated with callee. *See* ¶¶ 714-

724, *above*. To indicate that the SIP message is encrypted, the caller will set the “Encryption” header field of the INVITE message. *See ¶¶ 645-653, above.*

955. After a caller (or the caller’s SIP proxy server) transmits an INVITE message to the callee’s SIP server, the server will attempt to locate the callee by querying a location server using the callee’s SIP URL. *See ¶¶ 688-697, above.* If the callee has registered with the server as being available, the location server will return the address the callee registered. *See ¶¶ 666-671, 688, above.*

956. The callee’s SIP server then will either forward the INVITE message to the callee at the registered address or return the address to the caller. *See ¶¶ 666-671, 688, above.*

957. The callee will receive the INVITE message at the address it registered with the location server. *See ¶¶ 688-691, 693, above.* The callee then can decide whether it would like to accept the call. Therefore, a message indicating a desire to securely communicate is received at an address associated with the callee.

958. RFC 2543 thus shows “*receiving at a network address associated with a secure name of a first device a message from a second device requesting the desired to securely communicate with the first device, wherein the secure name of the first device is registered*” as specified by claim 29.

959. If the callee accepts the call, it will return a response message that contains a success status indicator code. *See ¶ 693, above.* The response message also will include a session description that contains a network address (*e.g.*, a domain name or IP address) and port number where the callee can receive session data. *See ¶¶ 693-695, above.*

960. If the caller specifies in the INVITE message that all SIP messages should be encrypted, the INVITE message, the callee's response, and the ACK message each would be encrypted. Where a SIP message is encrypted, the message body and most header fields will be encrypted. *See ¶¶ 719-720, above.*

961. If all SIP messages are to be encrypted, the callee's transmission of the encrypted response message to the caller satisfies sending a message securely from the first device to the second device.

962. The caller will receive the response message. If the message indicates the callee has accepted the call, the caller returns an ACK message to the callee to confirm acceptance of the call. *See ¶¶ 698-699, above.*

963. After the callee receives the ACK message, the call is established and the parties can exchange session data according the parameters specified during this process. *See ¶¶ 699-700, above.*

964. The caller will send session (*e.g.*, multimedia) data to the address that the callee listed in its response message. *See ¶¶ 693-695, above.* If the session

description specified an encryption key to use to encrypt session data, the caller and callee will encrypt the audio and video data streams. *See* ¶¶ 645-649, 699, 714-730, *above*. The transmission of the encrypted session from the callee to the caller satisfies sending a message securely from the first to the second device.

965. RFC 2543 thus shows “*sending a message securely from the first device to the second device*” as specified by claim 29.

966. Accordingly, RFC 2543 anticipates claim 29 of the ’181 patent under 35 U.S.C. § 102(e).

967. If it is found both that RFC 2543 does not completely describe the encryption process and that RFC 2543, RFC 1889, and RFC 2327 (which are incorporated by reference into RFC 2543) cannot be considered as a single disclosure, it would have been obvious to a person of ordinary skill in the art in April 2000 to combine the teachings of these RFCs. As RFC 2543 explains: “SIP is designed as part of the overall IETF multimedia data and control architecture currently incorporating protocols such as . . . **the real-time transport protocol (RTP) (RFC 1889 [3]) for transporting real-time data** and providing QOS feedback . . ., and **the session description protocol (SDP) (RFC 2327 [6]) for describing multimedia sessions**. Ex. 1033 (RFC 2543) at 8. RFC 2543 thus explicitly discloses that it is designed to be used in conjunction with the protocols described in RFC 1889 and RFC 2327.”

968. RFC 2543 also explicitly discloses that SIP messages include session description data structures that are compliant with the SDP protocol specified in RFC 2327. RFC 2543 even includes an appendix that summarizes how to use the SDP protocol in SIP messages. Ex. 1033 (RFC 2543) at 136-140. RFC 2327 shows that it can be used to distribute encryption keys to encrypt session data. *See* ¶¶ 645-649, *above*. RFC 2543 also shows SIP being used to initiate a call involving RTP data streams, which are defined by RFC 1889. Ex. 1033 (RFC 2543) at 137. RFC 1889 shows that RTP data can be encrypted. *See* ¶ 718, *above*. I believe a person of ordinary skill in February of 2000 would have found it obvious to consider RFC 2543, RFC 1889, and RFC 2327 together.

969. A person would have considered RFC 2543 with RFC 2401 because RFC 2543 refers to the IPsec standard defined in RFC 2401. *See* ¶¶ 726-727, 730, *above*. In addition it was common practice to add extra security functionality to existing systems by integrating protocols such as IPsec into edge routers and proxy servers. *See* ¶¶ 190-194, *above*.

970. The guidance in RFC 2401 would have specifically suggested utilizing a transparent proxy server to ensure that organizational security policies were enforced. *See* ¶¶ 369-371, *above*. This is also specifically suggested by RFC 2543. *See* ¶¶ 725-727, 730, *above*. A person of ordinary skill in the art could have configured a local firewall to automatically apply IPsec to any network traffic

being transmitted between corporate networks, or to any outbound network traffic.

In this configuration, IPsec could be used to encrypt both SIP messages and session data transmitted between the caller and the callee.

**C. Kiuchi et al., “C-HTTP - The Development of a Secure, Closed HTTP-based Network on the Internet”**

971. “C-HTTP - The Development of a Secure, Closed HTTP-based Network on the Internet,” authored by Takahiro Kiuchi and Shigekoto Kaihara (“Kiuchi”) (Ex. 1004), is a printed publication that was presented at the 1996 Symposium on Network and Distributed Systems Security (SNDSS) on February 22 & 23, 1996, and published by IEEE in the Proceedings of SNDSS 1996. Kiuchi is a printed publication that was distributed publicly without restriction no later than **February 1996**.

**1. Overview of Kiuchi**

972. Kiuchi describes techniques for creating a closed network over the Internet (referred to as the “C-HTTP” system). Kiuchi explains that the C-HTTP protocol enables the creation of a secure communication link over the Internet: “Using C-HTTP, a closed HTTP-based virtual network . . . for closed groups; for example, the headquarters and branches of a given corporation.” Ex. 1004 (Kiuchi) at 69.

973. In the Kiuchi system, a user agent (*e.g.*, a web browser) running on a computer in one private network is able to connect to and then access private web

pages (*e.g.*, HTML documents) stored on an origin server located in a different private network. Ex. 1004 (Kiuchi) at 64, 69. The user agent and origin server become members of a closed network. Ex. 1004 (Kiuchi) at 64.

974. The user agent and the origin server are conventional HTTP/1.0 compatible devices. Ex. 1004 (Kiuchi) at 64; *id.* at 65 (“**An origin server which is compatible with HTTP/1.0** is responsible for user authentication if necessary. It uses the built-in HTTP/1.0 authentication mechanism. Information concerning a user's ID, password and security realm (HTTP/1.0) are encrypted by proxies and are transferred only in encrypted form through the Internet.”). This means the user agent and server communicate in the standard client-server Internet model (*e.g.*, using TCP/IP and HTTP). An HTTP “user agent” is a web browser. A web browser would have been run on a personal computer, a PDA, a WAP-enabled mobile phone, or other device.

975. Kiuchi shows that its closed network is constructed over the Internet by using a client-side proxy and a server-side proxy that communicate with each other using encrypted communications. Ex. 1004 (Kiuchi) at 64-65 (“A client-side proxy and server-side proxy communicate with each other using a secure, encrypted protocol (C-HTTP).”). The proxies transparently intercept network traffic and perform proxy functions for the user agent and the origin server. Ex. 1004 (Kiuchi) at 64-65; *see also* Ex. 1004 (Kiuchi) at 68 (“Negotiations

concerning type and representation of objects are done between an origin server and user agent, using HTTP/1.0. As for these negotiations, C-HTTP is **transparent** to both of them. ... 3) Easy manipulation by end-users. End-users do not have to employ security protection procedures. **They do not even have to be conscious of using C-HTTP based communications.** This makes the design and implementation of C-HTTP simple.”). The proxies are installed in firewall devices situated between the user agent and the origin server, which are unaware of these proxies. Ex. 1004 (Kiuchi) at 64, 68.

976. The proxy servers are modified HTTP proxy servers, and they can behave as conventional HTTP/1.0 compatible devices. Ex. 1004 (Kiuchi) at 64; *id.* at 65 (“A client-side proxy behaves as an HTTP/1.0 compatible proxy . . .”), 68. Kiuchi shows that its proxy servers support standard domain names that can be resolved by public DNS servers on the Internet. Ex. 1004 (Kiuchi) at 65 (“If a client-side proxy receives an error status, **then it performs DNS lookup**, behaving like an ordinary HTTP/1.0 proxy.”); *see id.* (each proxy server has a “hostname (which does not have to be the same as its **DNS name**)”).

977. The client-side proxy and server-side proxy work in conjunction with a C-HTTP name server located on the Internet. Ex. 1004 (Kiuchi) at 64 (“In this paper, we discuss the design and implementation of a closed HTTP (Hypertext Transfer Protocol)-based network (C-HTTP) which can be built on the **Internet**.”



(emphasis added)). Each proxy that is part of the closed network must register with the C-HTTP name server. Ex. 1004 (Kiuchi) at 64-65. The C-HTTP name server stores information about each registered proxy including a hostname, IP address, and public key for the proxy. Ex. 1004 (Kiuchi) at 65, 67.

978. Kiuchi explains that for an organization “to participate in a closed network, it must [] install a client-side and/or server-side proxy on its firewall [and] register an IP address . . . and hostname (which does not have to be the same as its DNS name).” Ex. 1004 (Kiuchi) at 65 (emphasis added). In that passage, Kiuchi shows that its proxy servers each have a domain name that is registered with public DNS name and a “hostname” that is registered with the C-HTTP name server. *Id.* Kiuchi also shows that an organization must register its hostname with the C-HTTP name server before it can participate in the closed network. *Id.*

979. Kiuchi teaches that web servers and other secure destinations located within the C-HTTP closed network can be identified by non-standard domain names. For example, as shown in Kiuchi Appendix 3, a client-side proxy is identified by the domain name “University.of.Tokyo.Branch.Hospital,” which includes a non-standard top level domain name of “Hospital.” Ex. 1004 (Kiuchi) at 73. Similarly, a secure server-side proxy is identified by the domain name “Coordinating.Center.CSCRG,” which includes the non-standard top-level domain name “CSCRG.” Ex. 1004 (Kiuchi) at 73.

980. Kiuchi explains that “[i]n a C-HTTP-based network, instead of a DNS, a C-HTTP-based secure, encrypted name and certification service is used” to resolve host names into addresses. Ex. 1004 (Kiuchi) at 64. As Kiuchi explains, “C-HTTP includes its own **secure name service**, which contains a certification mechanism . . . [and the] C-HTTP name service is efficient because it can do **name resolution** and host certification simultaneously.” Ex. 1004 (Kiuchi) at 68 (emphasis added). The C-HTTP name server in the Kiuchi system thus acts as a secure domain name service for the closed network. *See also id.* at 64-65 (“When a server-side proxy accepts a request for connection from a client-side proxy, it asks the C-HTTP name server whether the client-side proxy is an appropriate member of the closed network. If the name server confirms that the query is legitimate, **it then examines whether the client-side proxy is permitted to access to the server-side proxy. If access is permitted, the C-HTTP name server sends the IP address and public key** of the client-side proxy . . .”).

981. Requests sent to and responses received from the C-HTTP server are authenticated with digital signatures. Ex. 1004 (Kiuchi) at 64; *see also id.* at 66 (the C-HTTP name server determines whether a query is “legitimate,” and then it “examines whether the client-side proxy is permitted to access to the server-side proxy”); *id.* at 67 (“Trial implementation is under way using 1) RSA as the asymmetric key encryption method (OSISEC RSA library) [2], 2) DES as the

symmetric key encryption method (GNU DES library) [3], 3) **RSA as the electronic signature method** (OSISEC RSA library) and 4) **a one-way has function based on MD5[4]).**”).

982. When a user agent makes an HTTP request to connect to a host, the request (which contains a URL specifying the destination) is received by the client-side proxy. Ex. 1004 (Kiuchi) at 65. After the client-side proxy receives the request, it determines whether the hostname specified in the URL corresponds to a secure destination by sending to a C-HTTP name server a request to resolve the hostname in the URL into an IP address. Ex. 1004 (Kiuchi) at 65 (the client-side proxy strips out the hostname from the URL and then “asks the C-HTTP name server whether it can communicate with the host specified in [the] URL.”); *see also* Ex. 1004 (Kiuchi) at 70-75. The hostname in a URL is a domain name. So for example, a URL in the Kiuchi scheme could be something like www.securehospital.com/desiredfile.htm, with “securehospital.com” being the hostname in the request.

983. After the C-HTTP name server receives the request from the client-side proxy, it determines whether the requested hostname is within the closed network and whether the client-side proxy is permitted to connect to the server-side proxy. Ex. 1004 (Kiuchi) at 65; *see also* Ex. 1004 (Kiuchi) at 64-65 (“When a server-side proxy accepts a request for connection from a client-side proxy, **it asks**

the C-HTTP name server whether the client-side proxy is an appropriate member of the closed network. If the name server confirms that the query is legitimate, it then examines whether the client-side proxy is permitted to access to the server-side proxy. If access is permitted, the C-HTTP name server sends the IP address and public key of the client-side proxy and both request and response Nonce values, which are the same as those sent to the client-side proxy.”). The C-HTTP server in the Kiuchi system thus evaluates and resolves the hostname in the request, and if access to the destination is permitted, returns the IP address of that hostname along with other information. Ex. 1004 (Kiuchi) at 65.

984. If the C-HTTP name server determines the connection is not permitted, the C-HTTP server returns a status code that indicates an error. Ex. 1004 (Kiuchi) at 65 (“**If it is not permitted, it sends a status code which indicates an error.** If a client-side proxy receives an error status, then it performs DNS lookup . . .); *see id.* at 65-66 (the server-side proxy “asks the C-HTTP name server whether the client-side proxy is an appropriate member of the close network. . . . If not, it [the C-HTTP name server] **sends a status code which indicates an error** and the server-side proxy refuses the connection from the client-side proxy.”); *see also* Ex. 1004 (Kiuchi) at 70-75.

985. If the C-HTTP name server determines the requested hostname is not within the closed network, the C-HTTP server returns a status code that indicates

an error. Ex. 1004 (Kiuchi) at 65 (“If the name server confirms that the query is legitimate, **it examines whether the requested server-side proxy is registered in the closed network** and is permitted to accept the connection from the client-side proxy. If the connection is permitted, the C-HTTP name server sends the IP address and public key of the server-side proxy and both request and response Nonce values. **If it is not permitted, it sends a status code which indicates an error.**”); *see also* Ex. 1004 (Kiuchi) at 70-75.

986. If the client-side proxy receives an error code from the C-HTTP name server, it means that the request has specified a non-secure destination, and should be handled like any conventional, non-secure connection request. Ex. 1004 (Kiuchi) at 65 (“If a client-side proxy receives an error status, then it performs DNS lookup, behaving like an ordinary HTTP/1.0 proxy.”).

987. If the connection request specifies a non-secure destination (*i.e.*, a website that is not part of the closed network), the client-side proxy resolves the hostname in the URL into an IP address by forwarding the request to a standard DNS server. Ex. 1004 (Kiuchi) at 65. The client-side proxy then acts as an ordinary HTTP proxy, and it will contact the server (the hostname) specified by the URL and return the requested content to the user agent. *Id.*

988. If the C-HTTP name server determines that the hostname corresponds to a secure destination and that a connection is permitted, the C-HTTP name server

resolves the hostname into an IP address and “sends the IP address and public key of the server-side proxy,” a nonce (a random number to prevent replay attacks), and other information to the client-side proxy. Ex. 1004 (Kiuchi) at 65; *see also* Ex. 1004 (Kiuchi) at 70-75. The response from the C-HTTP name server is encrypted. Ex. 1004 (Kiuchi) at 73-74.

989. If the client-side proxy receives an IP address and key from the C-HTTP name server, it determines the hostname in the URL corresponds to a secure destination. Ex. 1004 (Kiuchi) at 65. The client-side proxy then automatically sends an encrypted request to the server-side proxy requesting to establish a secure connection. Ex. 1004 (Kiuchi) at 65; *see also* Ex. 1004 (Kiuchi) at 70-75. The request includes the client-side proxy’s hostname and a symmetric data exchange key. *Id.*

990. After the server-side proxy receives the request, it sends a message to the C-HTTP name server containing the hostname of the client-side proxy to verify it is part of the closed network. Ex. 1004 (Kiuchi) at 65-66 (“the server-side proxy . . . authenticates the client-side proxy [and] checks the integrity of the request . . .”). Kiuchi also shows that the C-HTTP name server will evaluate each request it receives and determine whether “the requested server-side proxy . . . is permitted to accept the connection from the client-side proxy.” Ex. 1004 (Kiuchi) at 65; *see also* Ex. 1004 (Kiuchi) at 70-75.

991. If the C-HTTP name server confirms that the connection is permitted, it returns the client-side proxy's IP address, public key, a nonce, and other information. The server-side proxy uses that information to send to the client-side proxy an encrypted message containing a connection identifier and a second symmetric data exchange key. Ex. 1004 (Kiuchi) at 66, 74. The connection identifier will be included in the URL of any future HTTP requests so the proxies can determine that a secure session already has been established. Ex. 1004 (Kiuchi) at 65-66; *see also* Ex. 1004 (Kiuchi) at 70-75.

992. After the client-side proxy accepts and validates the message, the secure connection is established. Ex. 1004 (Kiuchi) at 66.

993. The client-side proxy will forward to the user agent any data it receives from the origin server. Ex. 1004 (Kiuchi) at 67. For example, Kiuchi explains it will forward HTML documents, as well as image and sound data. Ex. 1004 (Kiuchi) at 67.

994. Once the secure communication connection is established, when the user agent sends an HTTP request to access a secure web page on the origin server, the client-side proxy forwards the request to the server-side proxy in encrypted form using a C-HTTP format. Ex. 1004 (Kiuchi) at 66. The server-side proxy communicates with the origin server using HTTP/1.0. Ex. 1004 (Kiuchi) at 66; *see also* Ex. 1004 (Kiuchi) at 70-75.

995. The proxies automatically ensure that data exchanged between the user agent and the origin server is encrypted before being sent over the Internet. Ex. 1004 (Kiuchi) at 68. “C-HTTP is transparent to both of them [an origin server and user agent].” Ex. 1004 (Kiuchi) at 68.

996. Thus, the C-HTTP system sets up a secure connection automatically, and the process is completely transparent to the user. Ex. 1004 (Kiuchi) at 68 (End-users “do not even have to be conscious of using C-HTTP based communications.”).

997. Kiuchi explains that C-HTTP is built on HTTP because of HTTP’s flexibility. Ex. 1004 (Kiuchi) at 67. “HTTP has the flexibility to be able to provide services similar to those which have been provided by [other] protocols,” such as “FTP, SMTP, NNTP, GOPHER.” Ex. 1004 (Kiuchi) at 67. Kiuchi also explains that HTTP allows “distributed multimedia information systems with user-friendly graphical interfaces based on hypertext can be easily developed.” Ex. 1004 (Kiuchi) at 67.

998. Kiuchi provides a detailed specification of its C-HTTP specification. *See* Ex. 1004 (Kiuchi) at 70-75. This detailed specification, combined with the description in the paper, provides extensive guidance for implementing the Kiuchi system, and would be used by a person of ordinary skill to implement or adapt the Kiuchi system to work in different settings or applications.



## 2. Comparison of Kiuchi to Claims 1-6, 8-9 and 13-19, 21-29 of the '181 Patent

### a. Claim 1

999. Kiuchi describes systems and processes in which a secure connection between a user agent and an origin server is automatically established by proxy servers and a C-HTTP name server in response to a request from a user specifying a secure destination in the closed network. *See* ¶¶ 975, 978, 995-996, *above*.

1000. Kiuchi shows that the process is started when a user agent makes an HTTP request to connect to a hostname that is specified within a URL. *See* ¶¶ 978-980, *above*. The HTTP request is a request to resolve the hostname in the URL into an IP address so the user agent can connect to the destination. *See* ¶¶ 976, 980, 982-989, *above*.

1001. A client computer (*e.g.*, the origin server) and its proxy server (*e.g.*, server-side proxy) each can be considered a “device” within the meaning of the claims. *See, e.g.*, Ex. 1025 ('181 patent) at 51:15-29 (explaining that during the process of “establishing a VPN communication link between software module 3309 and secure server 3320” the VPN gatekeeper “provisions computer 3301 and secure web server computer 3320, **or a secure edge router for server computer 3320**, thereby creating the VPN.”).

1002. Kiuchi shows that an origin server and its server-side proxy can be associated with a domain name resolvable by a public DNS and a secure hostname

resolvable by the C-HTTP name server. *See* ¶¶ 978-980, *above*. The hostname can be an unconventional domain name such as “Coordinating.Center.CSCRG,” which includes the non-standard top-level domain name “CSCRG.” *See* ¶ 979, *above*.

1003. When the client-side proxy receives the request from the user agent, it sends a request to a C-HTTP name server. *See* ¶¶ 982-983, *above*. The C-HTTP name server evaluates the request to determine whether the hostname corresponds to a destination that is part of the closed network. *See* ¶¶ 982-987, *above*. The C-HTTP name server also evaluates whether the connection between the user agent and the origin server is permitted. *See* ¶ 984, *above*.

1004. If the C-HTTP name server determines the destination is not part of the closed network, it returns an error code. *See* ¶ 984, *above*. If the client-side proxy receives an error code, it determines the URL specifies a non-secure destination and it forwards the hostname in the URL to a conventional DNS server. *See* ¶ 985, *above*.

1005. If the C-HTTP name server determines the hostname in the URL corresponds to a secure destination and that the connection is permitted, it will return an IP address corresponding to the hostname along with other information. *See* ¶ 988, *above*. This information allows the client-side proxy to send a message to the server-side proxy.

1006. Kiuchi thus shows “*A non-transitory machine-readable medium comprising instructions for a method of communicating with a first device associated with a secure name and an unsecured name,*” as specified by claim 1.

1007. If the hostname corresponded to a secure destination, the client-side proxy receives from the C-HTTP name server the IP address, public key, and other information corresponding to the hostname. *See ¶¶ 988-989, above.* The client-side proxy then uses the IP address and key to send an encrypted message to the server-side proxy requesting to make a connection. *See ¶¶ 988-989, above.*

1008. The server-side proxy receives the connection request, and verifies the secure connection is permitted by sending a message to the C-HTTP name server to determine whether the client-side proxy is part of the closed network. *See ¶ 990, above.* If the C-HTTP name server verifies the connection is permissible, the name server will return the IP address and key associated with the client-side proxy, and the server-side proxy accepts the connection. *See ¶¶ 991-992, above.* Otherwise, the server-side proxy rejects the connection. *See ¶¶ 991-992, above.*

1009. Kiuchi thus shows “*receiving, at a network address corresponding to the secure name associated with the first device, a message from a second device of the desired to securely communicate with the first device,*” as specified by claim 1.

1010. If the C-HTTP name server verifies the connection is permissible, the server-side proxy accepts the connection. *See ¶¶ 990-992, above.* The server-side proxy then sends an encrypted message to the client-side proxy containing a data exchange key. *See ¶ 991, above.*

1011. After the client-side proxy receives the message from the server-side proxy, the connection is established. *See ¶ 992, above.* Data can then be securely transmitted between the user agent and the origin server because the proxy servers will automatically encrypt any traffic sent between them. *See ¶¶ 992-996, above.*

1012. Kiuchi thus shows “*sending a message over a secure communication link from the first device to the second device,*” as specified by claim 1.

1013. Accordingly, Kiuchi anticipates claim 1 of the ’181 patent under 35 U.S.C. § 102(e).

**b. Claim 2**

1014. Kiuchi describes systems and processes in which a secure connection between a user agent and an origin server is automatically established by proxy servers and a C-HTTP name server in response to a request from a user specifying a secure destination in the closed network. *See ¶¶ 975, 978, 995-996, above.*

1015. Kiuchi shows that the process is started when a user agent makes an HTTP request to connect to a host that is specified within a URL. *See ¶¶ 978-980, above.* The HTTP request is a request to resolve the hostname in the URL into an

IP address so the user agent can connect to the destination. *See* ¶¶ 976, 980, 982-989, *above*.

1016. Kiuchi shows that an origin server and its server-side proxy can be associated with a domain name resolvable by a public DNS and a secure hostname resolvable by the C-HTTP name server. *See* ¶¶ 978-980, *above*. The hostname can be an unconventional domain name such as “Coordinating.Center.CSCRG,” which includes the non-standard top-level domain name “CSCRG.” *See* ¶ 979, *above*.

1017. Kiuchi thus shows “*A method of using a first device to communicate with a second device having a secure name*” as specified by the preamble of claim 2.

1018. When the client-side proxy receives a request from the user agent, it sends a request to a C-HTTP name server. *See* ¶¶ 982-983, *above*. As Kiuchi explains, the C-HTTP name server is a “secure name service, which contains a certification mechanism . . . [and the] C-HTTP name service is efficient because it can do name resolution and host certification simultaneously.” Ex. 1004 (Kiuchi) at 68 (emphasis added); *see* ¶ 980, *above*.

1019. The C-HTTP name server evaluates the request to determine whether the hostname corresponds to a destination that is part of the closed network. *See* ¶¶ 982-987, *above*. The C-HTTP name server also evaluates whether the

connection between the user agent and the origin server is permitted. *See ¶ 984, above.*

1020. If the C-HTTP name server determines the destination is not part of the closed network, it returns an error code. *See ¶ 984, above.* If the client-side proxy receives an error code, it determines the URL specifies a non-secure destination and it forwards the hostname in the URL to a conventional DNS server. *See ¶ 985, above.*

1021. If the C-HTTP name server determines the hostname in the URL corresponds to a secure destination and that the connection is permitted, it will return an IP address corresponding to the hostname along with other information. *See ¶ 988, above.*

1022. Kiuchi thus shows “*from the first device, sending a message to a secure name service, the message requesting a network address associated with the secure name of the second device,*” as specified in claim 2.

1023. If the hostname corresponds to a secure destination, the client-side proxy receives from the C-HTTP name server the IP address, public key, and other information corresponding to the hostname. *See ¶¶ 988-989, above.* Therefore the client-side proxy receives the network address associated with the hostname.

1024. Kiuchi thus shows “*at the first device, receiving a message containing the network address associated with the secure name of the second device,*” as specified in claim 2.

1025. The client-side proxy then uses the IP address and key to send an encrypted message to the server-side proxy requesting to make a connection. *See ¶¶ 988-989, above.* Therefore the client-side proxy sends a secure, encrypted message to the network address associated with the hostname.

1026. The server-side proxy receives the encrypted connection request, and verifies the secure connection is permitted by sending a message to the C-HTTP name server to determine whether the client-side proxy is part of the closed network. *See ¶ 990, above.* If the C-HTTP name server verifies the connection is permissible, the name server will return the IP address and key associated with the client-side proxy, and the server-side proxy accepts the connection. *See ¶¶ 991-992, above.* Otherwise, the server-side proxy rejects the connection. *See ¶¶ 991-992, above.*

1027. If the C-HTTP name server verifies the connection is permissible, the server-side proxy accepts the connection. *See ¶¶ 990-992, above.* The server-side proxy then sends an encrypted message to the client-side proxy containing a data exchange key. *See ¶ 991, above.*

1028. After the client-side proxy receives the message from the server-side proxy, the connection is established. *See* ¶ 992, *above*. Data can then be securely transmitted between the user agent and the origin server because the proxy servers will automatically encrypt any traffic sent between them. *See* ¶¶ 992-996, *above*.

1029. Kiuchi thus shows “*from the first device, sending a message to the network address associated with the secure name of the second device using a secure communication link,*” as specified in claim 2.

1030. Accordingly, Kiuchi anticipates claim 2 of the ’181 patent under 35 U.S.C. § 102(e).

**c. Claim 3**

1031. Kiuchi shows that an origin server and its server-side proxy are associated with a secure hostname resolvable by the C-HTTP name server. *See* ¶¶ 978-980, *above*. The hostname can be an unconventional domain name such as “Coordinating.Center.CSCRG,” which includes the non-standard top-level domain name “CSCRG.” *See* ¶ 979, *above*.

1032. Kiuchi thus shows “*The method according to claim 2, wherein the secure name of the second device is a secure domain name,*” as specified by claim 3.

**d. Claim 4**



1033. Kiuchi shows that an origin server and its server-side proxy are associated with a secure hostname resolvable by the C-HTTP name server. *See* ¶¶ 978-980, *above*. The hostname can be an unconventional domain name such as “Coordinating.Center.CSCRG,” which includes the non-standard top-level domain name “CSCRG.” *See* ¶ 979, *above*. A non-standard domain name would indicate security.

1034. Kiuchi thus shows “*The method according to claim 2, wherein the secure name indicates security,*” as specified by claim 4.

**e. Claim 5**

1035. When the client-side proxy receives a request from the user agent, it sends a request to a C-HTTP name server. *See* ¶¶ 982-983, *above*.

1036. If the C-HTTP name server determines the hostname in the URL corresponds to a secure destination and that the connection is permitted, it will return an IP address corresponding to the hostname along with other information. *See* ¶ 988, *above*. The response from the C-HTTP name server is encrypted. *See* ¶ 988, *above*.

1037. Kiuchi thus shows “*The method according to claim 2, wherein receiving the message containing the network address associated with the secure name of the second device includes receiving the message in encrypted form,*” as specified by claim 5.

**f. Claim 6**

1038. If the client-side proxy receives an encrypted response from the C-HTTP name server, it necessarily decrypts it to obtain the IP address, public key and other information for the server-side proxy. *See* ¶¶ 988-989, *above*; Ex. 1004 (Kiuchi) at 72-73 (showing those fields are received in encrypted form).

1039. Kiuchi thus shows “*The method according to claim 5, further including decrypting the message,*” as specified by claim 6.

**g. Claim 8**

1040. If the hostname corresponds to a secure destination, the client-side proxy receives from the C-HTTP name server the IP address, public key, and other information corresponding to the hostname. *See* ¶¶ 988-989, *above*. Therefore the client-side proxy receives an IP address associated with the hostname.

1041. Kiuchi thus shows “*The method according to claim 2, wherein receiving a message containing the network address associated with the secure name of the device includes receiving the network address as an IP address associated with the secure name of the device,*” as specified by claim 8.

**h. Claim 9**

1042. Kiuchi shows that after the client-side proxy receives the server-side proxy’s key, it automatically sends an encrypted message to the server-side proxy. *See* ¶¶ 989, 991, 995-996, *above*.

1043. Kiuchi shows that after the server-side proxy verifies the connection is permitted, it creates a connection identifier for the session and all communications between the client-side and server-side proxy will be encrypted.

1044. Kiuchi thus shows “*The method according to claim 2, further including automatically initiating the secure communication link after it is enabled,*” as specified by claim 9.

**i. Claim 13**

1045. Kiuchi shows that the proxy servers can be configured to close the TCP session after each request and response. During the course of a secure session, data will be sent using multiple TCP sessions. Ex. 1004 (Kiuchi) at 67 (“C-HTTP itself is stateful but **the TCP connection is closed after each transaction** (request and response pair) in order to reduce the possibility of it being intercepted by attackers.”).

1046. This is consistent with the ’181 patent specification, which similarly shows use of multiple TCP sessions during a secure session. Ex. 1025 (181 patent) (181 patent) at 18:42-46.

1047. Kiuchi thus shows “*The method according to claim 2, wherein the receiving and sending of messages through the secure communication link includes multiple sessions,*” as specified by claim 13.

**j. Claims 14-17**

1048. Kiuchi explains that its system is built on HTTP because of its flexibility in permitting “distributed multimedia information systems with user-friendly graphical interfaces.” *See* ¶ 997, *above*. This flexibility allows the system to “provide services similar to those which have been provided by [other] protocols,” such as “FTP, SMTP, NNTP, GOPHER.” *Id.*; Ex. 1004 (Kiuchi) at 67.

1049. Kiuchi thus shows “*The method according to claim 2, further including supporting a plurality of services over the secure communication link,*” as specified by claim 14.

1050. Kiuchi explains that any type of data that transmitted via HTTP can be sent through its systems, such as electronic mail, HTML documents, and multimedia. *See* ¶ 997, *above*.

1051. Kiuchi thus shows “*The method according to claim 14, wherein the plurality of services comprises a plurality of communication protocols, a plurality of application programs, multiple sessions, or a combination thereof,*” as specified by claim 15.

1052. Kiuchi thus shows “*The method according to claim 15, wherein the plurality of application programs comprises video conferencing, e-mail, a word processing program, telephony or a combination thereof,*” as specified by claim 16.

1053. Kiuchi thus shows “*The method according to claim 15, wherein the plurality of services comprises audio, video or a combination thereof,*” as specified by claim 17.

**k. Claim 18**

1054. The C-HTTP name server authenticates the request using a digital signature and then evaluates it to determine whether the connection is permitted. *See* ¶¶ 981, 983-984, *above*; Ex. 1004 (Kiuchi) at 64-65 (“the name server . . . examines whether the client-side proxy is permitted to access to the server-side proxy.”).

1055. Kiuchi shows that the server-side proxy also authenticates the connection by asking the C-HTTP name server to confirm that a client-side proxy is part of the closed network. *See* ¶¶ 981, 990, *above*.

1056. Kiuchi thus shows “*The method according to claim 2, wherein the secure communication link is an authenticated link,*” as specified by claim 18.

**l. Claim 19**

1057. As I explained above, the steps of the method are performed by the client-side proxy on behalf of the user agent. *See* ¶¶ 1014-1029, *above*.

1058. The client-side proxy is a computer connected to a communication network. *See* ¶¶ 975-978, *above*.

1059. Kiuchi thus shows “*The method according to claim 2, wherein the first device is a computer, and the steps are performed on the computer,*” as specified by claim 19.

**m. Claim 21**

1060. Kiuchi shows that each host and proxy sever is associated with a domain name resolvable by a public DNS and a secure hostname resolvable by the C-HTTP name server. *See ¶¶ 978-980, above.* Therefore, each device is provided with a public domain name.

1061. Kiuchi thus shows “*The method according to claim 2, further including providing an unsecured name associated with the device,*” as specified by claim 21.

**n. Claim 22**

1062. Kiuchi shows that before an organization can participate in the closed network, it must register a hostname with C-HTTP name server. *See ¶¶ 978, 985, above; Ex. 1004 (Kiuchi) at 65 (“If the name server confirms that the query is legitimate, **it examines whether the requested server-side proxy is registered** in the closed network . . .”).*

1063. Kiuchi thus shows “*The method according to claim 2, wherein the secure name is registered prior to the step of sending a message to a secure name service,*” as specified by claim 22.

**o. Claim 23**

1064. Kiuchi shows that an origin server and its server-side proxy are associated with a secure hostname resolvable by the C-HTTP name server. *See* ¶¶ 978-980, *above*. The hostname can be an unconventional domain name such as “Coordinating.Center.CSCRG,” which includes the non-standard top-level domain name “CSCRG.” *See* ¶ 979, *above*.

1065. Kiuchi thus shows “*The method according to claim 2, wherein the secure name of the second device is a secure, non-standard domain name,*” as specified by claim 23.

**p. Claim 24**

1066. Kiuchi describes systems and processes in which a secure connection between a user agent and an origin server is automatically established by proxy servers and a C-HTTP name server in response to a request from a user specifying a secure destination in the closed network. *See* ¶¶ 975, 978, 995-996, *above*.

1067. Kiuchi shows that the process is started when a user agent makes an HTTP request to connect to a hostname that is specified within a URL. *See* ¶¶ 978-980, *above*. The HTTP request is a request to resolve the hostname in the URL into an IP address so the user agent can connect to the destination. *See* ¶¶ 976, 980, 982-989, *above*.

1068. Kiuchi shows that an origin server and its server-side proxy can be associated with a domain name resolvable by a public DNS and a secure hostname resolvable by the C-HTTP name server. *See ¶¶ 978-980, above.* The hostname can be an unconventional domain name such as “Coordinating.Center.CSCRG,” which includes the non-standard top-level domain name “CSCRG.” *See ¶ 979, above.*

1069. When the client-side proxy receives the request from the user agent, it sends a request to a C-HTTP name server. *See ¶¶ 982-983, above.* The C-HTTP name server evaluates the request to determine whether the hostname corresponds to a destination that is part of the closed network. *See ¶¶ 982-987, above.* The C-HTTP name server also evaluates whether the connection between the user agent and the origin server is permitted. *See ¶ 984, above.*

1070. If the C-HTTP name server determines the destination is not part of the closed network, it returns an error code. *See ¶ 984, above.* If the client-side proxy receives an error code, it determines the URL specifies a non-secure destination and it forwards the hostname in the URL to a conventional DNS server. *See ¶ 985, above.*

1071. If the C-HTTP name server determines the hostname in the URL corresponds to a secure destination and that the connection is permitted, it will return an IP address corresponding to the hostname along with other information.



See ¶ 988, *above*. This information allows the client-side proxy to send a message to the server-side proxy.

1072. Kiuchi thus shows “*A method of using a first device to securely communicate with a second device over a communication network,*” as specified in the preamble of claim 24.

1073. Kiuchi shows that before an organization can participate in the closed network, it must install a proxy server and register that proxy server’s hostname and IP address with C-HTTP name server. See ¶ 978, *above*.

1074. Kiuchi thus shows “*at the first device requesting and obtaining registration of a secure name for the first device, the secure name being associated with a network address*” as specified by claim 24.

1075. If the hostname corresponds to a secure destination, the client-side proxy receives from the C-HTTP name server the IP address, public key, and other information corresponding to the hostname. See ¶¶ 988-989, *above*. The client-side proxy then uses the IP address and key to send an encrypted message to the server-side proxy requesting to make a connection. See ¶¶ 988-989, *above*.

1076. The server-side proxy receives the connection request, and verifies the secure connection is permitted by sending a message to the C-HTTP name server to determine whether the client-side proxy is part of the closed network. See ¶ 990, *above*. If the C-HTTP name server verifies the connection is permissible, the name

server will return the IP address and key associated with the client-side proxy, and the server-side proxy accepts the connection. *See* ¶¶ 991-992, *above*. Otherwise, the server-side proxy rejects the connection. *See* ¶¶ 991-992, *above*.

1077. Kiuchi thus shows “*receiving at the network address associated with the secure name of the first device a message from a second device of the desire to securely communicate with the first device*” as specified by claim 24.

1078. If the C-HTTP name server verifies the connection is permissible, the server-side proxy accepts the connection. *See* ¶¶ 990-992, *above*. The server-side proxy then sends an encrypted message to the client-side proxy containing a data exchange key. *See* ¶ 991, *above*. Therefore, the server-side proxy sends a message securely to the client-side proxy.

1079. After the client-side proxy receives the message from the server-side proxy, the connection is established. *See* ¶ 992, *above*. Data can then be securely transmitted between the user agent and the origin server because the proxy servers will automatically encrypt any traffic sent between them. *See* ¶¶ 992-996, *above*.

1080. Kiuchi thus shows “*sending a message securely from the first device to the second device*” as specified by claim 24.

1081. Accordingly, Kiuchi anticipates claim 24 of the ’181 patent under 35 U.S.C. § 102(e).

**q. Claim 25**

1082. Kiuchi shows that before an organization can participate in the closed network, it must install a proxy server and register that proxy server's hostname and IP address with C-HTTP name server. *See ¶ 978, above.*

1083. Kiuchi thus shows “*The method according to claim 24, wherein requesting and obtaining registration of a secure name for the first device comprises using the first device to obtain a registration of the secure name for the first device*” as specified in claim 25.

1084. If the C-HTTP name server verifies the connection is permissible, the server-side proxy accepts the connection. *See ¶¶ 990-992, above.* The server-side proxy then sends an encrypted message to the client-side proxy containing a data exchange key. *See ¶ 991, above.*

1085. After the client-side proxy receives the message from the server-side proxy, the connection is established. *See ¶ 992, above.* Data can then be securely transmitted between the user agent and the origin server because the proxy servers will automatically encrypt any traffic sent between them. *See ¶¶ 992-996, above.*

1086. Kiuchi thus shows “*The method according to claim 24 . . . wherein sending a message securely comprises sending the message from the first device to the second device using a secure communication link*” as specified in claim 25.

**r. Claim 26**

1087. Kiuchi describes systems and processes in which a secure connection between a user agent and an origin server is automatically established by proxy servers and a C-HTTP name server in response to a request from a user specifying a secure destination in the closed network. *See ¶¶ 975, 978, 995-996, above.*

1088. Kiuchi shows that the process is started when a user agent makes an HTTP request to connect to a hostname that is specified within a URL. *See ¶¶ 978-980, above.* The HTTP request is a request to resolve the hostname in the URL into an IP address so the user agent can connect to the destination. *See ¶¶ 976, 980, 982-989, above.*

1089. Kiuchi shows that an origin server and its server-side proxy can be associated with a domain name resolvable by a public DNS and a secure hostname resolvable by the C-HTTP name server. *See ¶¶ 978-980, above.* The hostname can be an unconventional domain name such as “Coordinating.Center.CSCRG,” which includes the non-standard top-level domain name “CSCRG.” *See ¶ 979, above.*

1090. When the client-side proxy receives the request from the user agent, it sends a request to a C-HTTP name server. *See ¶¶ 982-983, above.* The C-HTTP name server evaluates the request to determine whether the hostname corresponds to a destination that is part of the closed network. *See ¶¶ 982-987, above.* The C-

HTTP name server also evaluates whether the connection between the user agent and the origin server is permitted. *See ¶ 984, above.*

1091. If the C-HTTP name server determines the destination is not part of the closed network, it returns an error code. *See ¶ 984, above.* If the client-side proxy receives an error code, it determines the URL specifies a non-secure destination and it forwards the hostname in the URL to a conventional DNS server. *See ¶ 985, above.*

1092. If the C-HTTP name server determines the hostname in the URL corresponds to a secure destination and that the connection is permitted, it will return an IP address corresponding to the hostname along with other information. *See ¶ 988, above.* This information allows the client-side proxy to send a message to the server-side proxy.

1093. Kiuchi thus shows “*A method of using a first device to communicate with a second device over a communication network,*” as specified in the preamble of claim 26.

1094. Kiuchi shows that each host and proxy sever is associated with a domain name resolvable by a public DNS and a secure hostname resolvable by the C-HTTP name server. *See ¶¶ 978-980, above.* For the domain name to be listed in a public DNS server, it necessarily must have been registered. *See ¶¶ 91-110, 127-133, above.*

1095. Kiuchi thus shows “*from the first device requesting and obtaining registration of an unsecured name associated with the first device*” as specified by claim 26.

1096. Kiuchi shows that before an organization can participate in the closed network, it must install a proxy server and register that proxy server’s hostname and IP address with C-HTTP name server. *See ¶ 978, above.*

1097. Kiuchi thus shows “*from the first device requesting and obtaining registration of a secure name associated with the first device, wherein a unique network address corresponds to the secure name associated with the first device*” as specified by claim 26.

1098. If the hostname corresponds to a secure destination, the client-side proxy receives from the C-HTTP name server the IP address, public key, and other information corresponding to the hostname. *See ¶¶ 988-989, above.* The client-side proxy then uses the IP address and key to send an encrypted message to the server-side proxy requesting to make a connection. *See ¶¶ 988-989, above.*

1099. The server-side proxy receives the connection request, and verifies the secure connection is permitted by sending a message to the C-HTTP name server to determine whether the client-side proxy is part of the closed network. *See ¶ 990, above.* If the C-HTTP name server verifies the connection is permissible, the name server will return the IP address and key associated with the client-side proxy, and

the server-side proxy accepts the connection. *See ¶¶ 991-992, above.* Otherwise, the server-side proxy rejects the connection. *See ¶¶ 991-992, above.*

1100. Kiuchi thus shows “*receiving at the unique network address associated with the secure name a message from a second device requesting the desire to securely communicate with the first device*” as specified by claim 26.

1101. If the C-HTTP name server verifies the connection is permissible, the server-side proxy accepts the connection. *See ¶¶ 990-992, above.* The server-side proxy then sends an encrypted message to the client-side proxy containing a data exchange key. *See ¶ 991, above.*

1102. After the client-side proxy receives the message from the server-side proxy, the connection is established. *See ¶ 992, above.* Data can then be securely transmitted between the user agent and the origin server because the proxy servers will automatically encrypt any traffic sent between them. *See ¶¶ 992-996, above.*

1103. Kiuchi thus shows “*from the first device sending a message securely from the first device to the second device*” as specified by claim 26.

1104. Accordingly, Kiuchi anticipates claim 26 of the ’181 patent under 35 U.S.C. § 102(e).

**s. Claim 27**

1105. Kiuchi shows that each host and proxy sever is associated with a domain name resolvable by a public DNS and a secure hostname resolvable by the

C-HTTP name server. *See* ¶¶ 978-980, *above*. For the domain name to be listed in a public DNS server, it necessarily must have been registered. *See* ¶¶ 91-110, 127-133, *above*.

1106. Kiuchi thus shows “*wherein requesting and obtaining registration of an unsecured name associated with the first device comprises using the first device to obtain a registration of the unsecured name associated with the first device,*” as specified by claim 27.

1107. Kiuchi shows that before an organization can participate in the closed network, it must install a proxy server and register that proxy server’s hostname and IP address with C-HTTP name server. *See* ¶ 978, *above*.

1108. Kiuchi thus shows “*wherein requesting and obtaining registration of a secure name associated with the first device comprises using the first device to obtain a registration of the secure name associated with the first device,*” as specified by claim 27.

1109. Accordingly, Kiuchi anticipates claim 27 of the ’181 patent under 35 U.S.C. § 102(e).

**t. Claim 28**

1110. Kiuchi describes systems and processes in which a secure connection between a user agent and an origin server is automatically established by proxy



servers and a C-HTTP name server in response to a request from a user specifying a secure destination in the closed network. *See ¶¶ 975, 978, 995-996, above.*

1111. Kiuchi shows that the process is started when a user agent makes an HTTP request to connect to a hostname that is specified within a URL. *See ¶¶ 978-980, above.* The HTTP request is a request to resolve the hostname in the URL into an IP address so the user agent can connect to the destination. *See ¶¶ 976, 980, 982-989, above.*

1112. Kiuchi shows that an origin server and its server-side proxy can be associated with a domain name resolvable by a public DNS and a secure hostname resolvable by the C-HTTP name server. *See ¶¶ 978-980, above.* The hostname can be an unconventional domain name such as “Coordinating.Center.CSCRG,” which includes the non-standard top-level domain name “CSCRG.” *See ¶ 979, above.*

1113. Kiuchi thus shows “*A non-transitory machine-readable medium comprising instructions,*” as specified by the preamble of claim 28.

1114. When the client-side proxy receives a request from the user agent, it sends a request to a C-HTTP name server. *See ¶¶ 982-983, above.* As Kiuchi explains, the C-HTTP name server is a “secure name service, which contains a certification mechanism . . . [and the] C-HTTP name service is efficient because it

can do name resolution and host certification simultaneously.” Ex. 1004 (Kiuchi) at 68 (emphasis added); *see* ¶ 980, *above*.

1115. The C-HTTP name server evaluates the request to determine whether the hostname corresponds to a destination that is part of the closed network. *See* ¶¶ 982-987, *above*. The C-HTTP name server also evaluates whether the connection between the user agent and the origin server is permitted. *See* ¶ 984, *above*.

1116. If the C-HTTP name server determines the destination is not part of the closed network, it returns an error code. *See* ¶ 984, *above*. If the client-side proxy receives an error code, it determines the URL specifies a non-secure destination and it forwards the hostname in the URL to a conventional DNS server. *See* ¶ 985, *above*.

1117. If the C-HTTP name server determines the hostname in the URL corresponds to a secure destination and that the connection is permitted, it will return an IP address corresponding to the hostname along with other information. *See* ¶ 988, *above*.

1118. Kiuchi thus shows “*sending a message to a secure name service, the message requesting a network address associated with a secure name of a device*” as specified by claim 28.

1119. If the hostname corresponds to a secure destination, the client-side proxy receives from the C-HTTP name server the IP address, public key, and other information corresponding to the hostname. *See ¶¶ 988-989, above.* Therefore the client-side proxy receives the network address associated with the hostname.

1120. Kiuchi thus shows “*receiving a message containing the network address associated with the secure name of the device*” as specified by claim 28.

1121. The client-side proxy then uses the IP address and key to send an encrypted message to the server-side proxy requesting to make a connection. *See ¶¶ 988-989, above.* Therefore the client-side proxy sends a secure, encrypted message to the network address associated with the hostname.

1122. The server-side proxy receives the encrypted connection request, and verifies the secure connection is permitted by sending a message to the C-HTTP name server to determine whether the client-side proxy is part of the closed network. *See ¶ 990, above.* If the C-HTTP name server verifies the connection is permissible, the name server will return the IP address and key associated with the client-side proxy, and the server-side proxy accepts the connection. *See ¶¶ 991-992, above.* Otherwise, the server-side proxy rejects the connection. *See ¶¶ 991-992, above.*

1123. If the C-HTTP name server verifies the connection is permissible, the server-side proxy accepts the connection. *See ¶¶ 990-992, above.* The server-side

proxy then sends an encrypted message to the client-side proxy containing a data exchange key. *See* ¶ 991, *above*.

1124. After the client-side proxy receives the message from the server-side proxy, the connection is established. *See* ¶ 992, *above*. Data can then be securely transmitted between the user agent and the origin server because the proxy servers will automatically encrypt any traffic sent between them. *See* ¶¶ 992-996, *above*.

1125. Kiuchi thus shows “*sending a message to the network address associated with the secure name of the device using a secure communication link*” as specified by claim 28.

1126. Accordingly, Kiuchi anticipates claim 28 of the ’181 patent under 35 U.S.C. § 102(e).

**u. Claim 29**

1127. Kiuchi describes systems and processes in which a secure connection between a user agent and an origin server is automatically established by proxy servers and a C-HTTP name server in response to a request from a user specifying a secure destination in the closed network. *See* ¶¶ 975, 978, 995-996, *above*.

1128. Kiuchi shows that the process is started when a user agent makes an HTTP request to connect to a hostname that is specified within a URL. *See* ¶¶ 978-980, *above*. The HTTP request is a request to resolve the hostname in the

URL into an IP address so the user agent can connect to the destination. *See* ¶¶ 976, 980, 982-989, *above*.

1129. Kiuchi shows that an origin server and its server-side proxy can be associated with a domain name resolvable by a public DNS and a secure hostname resolvable by the C-HTTP name server. *See* ¶¶ 978-980, *above*. The hostname can be an unconventional domain name such as “Coordinating.Center.CSCRG,” which includes the non-standard top-level domain name “CSCRG.” *See* ¶ 979, *above*.

1130. When the client-side proxy receives the request from the user agent, it sends a request to a C-HTTP name server. *See* ¶¶ 982-983, *above*. The C-HTTP name server evaluates the request to determine whether the hostname corresponds to a destination that is part of the closed network. *See* ¶¶ 982-987, *above*. The C-HTTP name server also evaluates whether the connection between the user agent and the origin server is permitted. *See* ¶ 984, *above*.

1131. If the C-HTTP name server determines the destination is not part of the closed network, it returns an error code. *See* ¶ 984, *above*. If the client-side proxy receives an error code, it determines the URL specifies a non-secure destination and it forwards the hostname in the URL to a conventional DNS server. *See* ¶ 985, *above*.

1132. If the C-HTTP name server determines the hostname in the URL corresponds to a secure destination and that the connection is permitted, it will return an IP address corresponding to the hostname along with other information. *See* ¶ 988, *above*. This information allows the client-side proxy to send a message to the server-side proxy.

1133. Kiuchi thus shows “*A non-transitory machine-readable medium comprising instructions for a method of communicating with a device having a secure name,*” as specified by the preamble of claim 29.

1134. If the hostname corresponds to a secure destination, the client-side proxy receives from the C-HTTP name server the IP address, public key, and other information corresponding to the hostname. *See* ¶¶ 988-989, *above*. The client-side proxy then uses the IP address and key to send an encrypted message to the server-side proxy requesting to make a connection. *See* ¶¶ 988-989, *above*.

1135. The server-side proxy receives the connection request, and verifies the secure connection is permitted by sending a message to the C-HTTP name server to determine whether the client-side proxy is part of the closed network. *See* ¶ 990, *above*. If the C-HTTP name server verifies the connection is permissible, the name server will return the IP address and key associated with the client-side proxy, and the server-side proxy accepts the connection. *See* ¶¶ 991-992, *above*. Otherwise, the server-side proxy rejects the connection. *See* ¶¶ 991-992, *above*.

1136. Kiuchi shows that before an organization can participate in the closed network, it must install a proxy server and register that proxy server's hostname and IP address with C-HTTP name server. *See ¶ 978, above.*

1137. Kiuchi thus shows “*receiving at a network address associated with a secure name of a first device a message from a second device requesting the desired to securely communicate with the first device, wherein the secure name of the first device is registered*” as specified by claim 29.

1138. If the C-HTTP name server verifies the connection is permissible, the server-side proxy accepts the connection. *See ¶¶ 990-992, above.* The server-side proxy then sends an encrypted message to the client-side proxy containing a data exchange key. *See ¶ 991, above.*

1139. After the client-side proxy receives the message from the server-side proxy, the connection is established. *See ¶ 992, above.* Data can then be securely transmitted between the user agent and the origin server because the proxy servers will automatically encrypt any traffic sent between them. *See ¶¶ 992-996, above.*

1140. Kiuchi thus shows “*sending a message securely from the first device to the second device*” as specified by claim 29.

1141. Accordingly, Kiuchi anticipates claim 29 of the '181 patent under 35 U.S.C. § 102(e).

**D. U.S. Patent No. 6,557,037 to Provino**

1142. As explained in more detail below, the processes and systems shown in Provino anticipate and would have made obvious to a person of ordinary skill in the art claims 1-29 of the '181 patent.

### **1. Overview of Provino**

1143. Provino describes systems and methods for establishing secure communication between devices connected to public networks, such as the Internet, and devices connected to private networks. Ex. 1003 (Provino) at Abstract, 1:14-16, 2:59-65, 8:14-26, 9:6-1, 9:32-10:33. Provino explains that this is accomplished by facilitating the resolution of “secondary addresses, such as the Internet’s human-readable addresses, to network addresses by nameservers connected to the private networks.” Ex. 1003 (Provino) at 2:59-65. Provino summarizes its invention as follows:

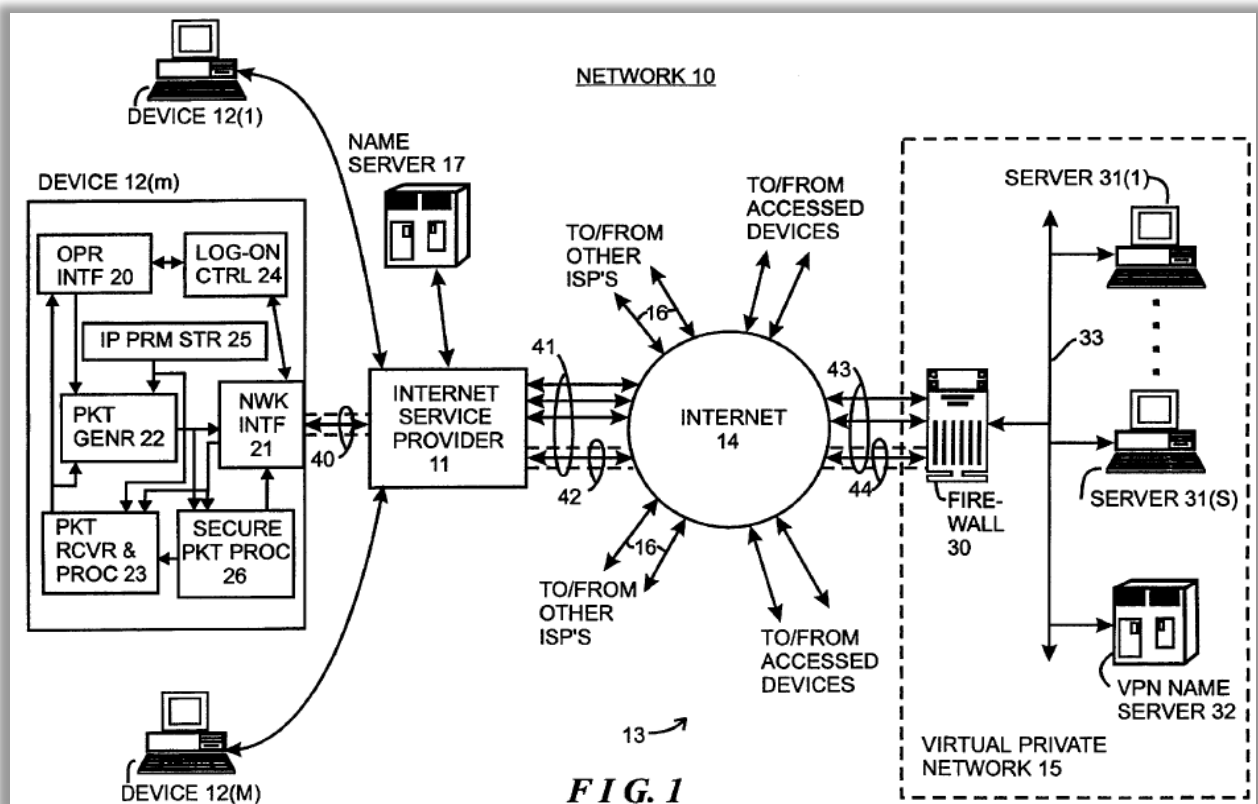
[T]he invention provides a system comprising a virtual private network and an external device interconnected by a digital network. The virtual private network has a firewall, at least one internal device and a nameserver each having a network address. The internal device also has a secondary address, and the nameserver is configured to provide an association between the secondary address and the network address. The firewall, in response to a request from the external device to establish a connection therebetween, provides the external device with the network address of the nameserver. The external device, in response to a request from an operator or the like, including the internal device's secondary address, requesting access to the



internal device, generates a network address request message for transmission over the connection to the firewall requesting resolution of the network address associated with the secondary address. The firewall provides the address resolution request to the nameserver, and the nameserver provides the network address associated with the secondary address to the firewall. The firewall, in turn, provides the network address in a network address response message for transmission over the connection to the external device. The external device can thereafter use the network address so provided in subsequent communications with the firewall.

Ex. 1003 (Provino) at 2:66-3:24.

1144. The architecture of Provino's system is illustrated in Figure 1:



1145. Figure 1 shows network 10 of the Provino system as including: (i) various devices (12(1), 12(m), and 13) interconnected through the Internet (14); (ii) device(s) 13, at least one of which is a Virtual Private Network (15) (which includes servers 31(s), VPN Name Server 32, and Firewall 30), and which is capable of securely communicating with other devices (*e.g.*, 12(1) and 12(m)); and (iii) name servers (17 and 32) that are each capable of resolving certain human readable domain names for integer Internet addresses and configured, together with firewall (30), to establish secure communication links between the devices 12(1), 12(m) and device(s)13, *i.e.*, VPN 15. Ex. 1003 (Provino) at Figure 1; Ex. 1003 (Provino) at 8:58-62.

1146. Provino shows that its systems are connected to a communication network. Ex. 1003 (Provino) at Figure 1. Provino explains that the communication network can include the Internet. Ex. 1003 (Provino) at 1:67-2:5.

1147. The “devices 12(m)” described in Provino that are connected to the Internet (through ISP 11) include personal computers and computer workstations, and communicate using any convenient protocol, such as “point-to-point protocol” (*i.e.*, “PPP”), or any “conventional multi-drop network protocol.” Ex. 1003 (Provino) at 4:23-35; *see also id.* at 4:50-5:17. The devices 12(m) include “network and/or telephony interface devices,” as well as application programs for processing data. Ex. 1003 (Provino) at 4:35-49, 5:18-59 (describing

communication sessions and applications, such as Internet browsers, for processing data received in communication session).

1148. The “devices 13” described in Provino, which are accessed and communicated with by devices 12(m), can be “personal computers, computer workstations, and the like, and also including mini- and mainframe computers, mass storage systems, compute[r] servers, local area networks (“LAN’s”) and wide area networks (“WAN’s”), including such devices . . . which may be directly or indirectly connected to the networks.” Ex. 1003 (Provino) at 5:65-6:6.

1149. Provino explains that at least one of devices “will include at least one private network 15.” Ex. 1003 (Provino) at 6:6-29. The virtual private network 15 is comprised of a plurality of devices, including “a firewall 30, a plurality of servers 31(1) through 31(S) (generally identified by reference numeral 31(s)) and a nameserver 32, all interconnected by a communication link 33.” Ex. 1003 (Provino) at 6:15-29. Provino describes the firewall 30 and servers 31(s) as “similar to any of the various types of devices 12(m) and 13 described herein, and thus may include, for example, personal computers, computer workstations, and the like, and also including mini-and mainframe computers, mass storage systems, compute servers, local area networks (“LAN’s”) and wide area networks (“WAN’s”) including such devices and numerous other types of devices which may be connected directly or indirectly to the networks.” *Id.*

1150. The virtual private network (VPN 15) is described in Provino as, for example, “maintained by a company, governmental agency, organization or the like, which desires to allow the servers 31(a) to access other devices outside of the virtual private network 15 and transfer information thereto over the Internet 14 . . . in a controlled manner.” Ex. 1003 (Provino) at 9:6-13.

1151. The systems described in Provino include nameservers. Ex. 1003 (Provino) at Figure 1; Ex. 1003 (Provino) at 7:25-8:14, 8:67-9:5. Provino explains that the nameservers are able to resolve certain human-readable names (*e.g.*, domain names) into Internet addresses. *Id.*; Ex. 1003 (Provino) at 8:67-9:5. For example, at 1:54-60, Provino states:

Internet domains ... which are interconnected in the Internet accommodate the use of human readable names. To accommodate the use of human-readable names, **nameservers, also referred to as DNS servers, are provided to resolve the human –readable names to Internet addresses.**

Ex. 1003 (Provino) at 1:54-60 (emphasis added). Provino explains that the Internet provides “a second addressing mechanism which is more easily utilized by human operators of the respective devices,” that includes human-readable domain names that can be resolved by certain nameservers. Ex. 1003 (Provino) at 7:22-8:14. Provino shows that a request to resolve a name associated with an internal device is handled by the nameserver 17, firewall 30, and VPN nameserver 32 working

together. Ex. 1003 (Provino) at 10:56-67, 12:37-61.

1152. What Provino is explaining is that its systems use the standardized domain name resolution process, which is defined by RFC 1034 and RFC 1035. I described this process in detail above. See ¶¶ 73-114, *above*. The domain names resolved by the Provino system therefore would include standard domain names, which necessarily include a top-level domain.

1153. The nameservers described in Provino that are external to the virtual private network 15, such as nameserver 17, are maintained by an ISP or another entity unaffiliated with the virtual private network 15. Ex. 1003 (Provino) at 7:37-46, 8:34-57.

1154. Provino explains that its systems also include internal nameservers, such as vpn nameserver 32, that resolve certain domain names for devices internal to the virtual private network, such as for server 31(s). Ex. 1003 (Provino) at 8:67-9:5.

1155. Through an authentication process, firewall 30 in the Provino scheme regulates access to devices that are external to virtual private network 15, but desire to establish secure communications with a device internal to the virtual private network 15. Ex. 1003 (Provino) at 9:13-15 (“The firewall 30 serves to control access by devices external to the virtual private network 15 to servers 31(s) within the virtual private network 15”); *see also* Ex. 1003 (Provino) at 10:56-67.

Provino describes the way in which firewall 30 facilitates this service to the virtual private network 15:

Firewall 30 serves to control access by devices external to the virtual private network 15 to servers 31(s) within the virtual private network 15. In that operation, the firewall 30 also connects to the Internet 14, receives message packets therefrom for transfer to a server 31(s). If the message packet indicates that the source of the message packet is requesting access to the particular server 31(s), and if the source is authorized to access the server 31(s), the firewall 30 will forward the message packet over the communication link 33 to the server 31(s). On the other hand if the source is not authorized to access the server 31(s), the firewall 30 will not forward the message packet to the server 31(s), and may, instead, transmit a response message packet to the source device indicating that the source was not authorized to access the server 31(s). The firewall may be similar to other devices 31(s) in the virtual private network 15, with the addition of one or more connections to the Internet, which are generally identified by reference numeral 43.

Ex. 1003 (Provino) at 9:5-31.

1156. Provino shows that its authentication processes for secure communication requests, as described above, use cryptographic techniques.

Provino shows systems that receive a request for a network address from device 12(m). Ex. 1003 (Provino) at 9:46-67. The firewall will determine whether device 12(m) is authorized to access server 31(s), and if so, the device and firewall will

engage in a dialog where they exchange encryption algorithms and keys. Ex. 1003 (Provino) at 9:46-67. For example, at 9:56 to 10:12, Provino explains:

During the dialog, the firewall 30 may provide the device 12(m) with the identification of a decryption algorithm and associated decryption key which the device 12(m) is to use in decrypting the encrypted portions of message packets which the virtual private network transmits to the device 12(m). In addition, the firewall 30 may also provide the device 12(m) with the identification of an encryption algorithm and associated encryption key which the device 12(m) is to use in encrypting the portions of message packets which the device 12(m) transmits to the virtual private network 15 which are to be encrypted; alternatively, the device 12(m) can provide the identification of the encryption algorithm and key that it (that is device 12(m)) will use to the firewall 30 during the dialog. The device 12(m) can store in its IP parameter store 25 information concerning the secure tunnel, including information associating the identification of the firewall 30 and the identifications of the encryption and decryption algorithms and associated keys for message packets to be transferred over the secure tunnel.

1157. Provino describes its systems as supporting the establishment of a “secure tunnel” to facilitate secure communication links between a device external to the virtual private network 15, such as device 12(m), and a device internal to that network, such as server 31(s). Ex. 1003 (Provino) at 9:32-45. As Provino explains:

**Communications between devices external to the virtual private network 15, such as device 12(m), and a device, such as a server 31(s), inside the virtual private network 15, may be maintained over a secure tunnel between the firewall 30 and the external device as described above to maintain the information transferred therebetween secret while being transferred over the Internet 14 and through the ISP 11.** A secure tunnel between device 12(m) and virtual private network 15 is represented in FIG. 1 by logical connections identified by reference numerals 40, 42, and 44; it will be appreciated that the logical connection 42 comprises one of the logical connections 41 between ISP 11 and Internet 14, and logical connection 44 comprises one of the logical connections 43 between the Internet 14 and the firewall 30.

Ex. 1003 (Provino) at 9:32-44 (emphasis added).

1158. Provino explains that data sent over a secure tunnel is encrypted. Ex. 1003 (Provino) at 5:42-52 (“In connection with the invention, as noted above the device 12(m) also includes a secure message packet processor 26. The secure message packet processor 26 facilitates the establishment and use of a ‘secure tunnel,’ which will be described below, between the device 12(m) and another device 12 (m') (m' = m) or 13. **Generally, in a secure tunnel, information in at least the data portion of message packets transferred between device 12(m) and a specific other device 12(m') (m' = m) or 13 is maintained in secret by, for example, encrypting the data portion** prior to transmission by the source



device.”).

1159. Provino describes its processes as using using tunneled packets (or tunnel packaging) over the secure communication link between device 12(m) and, for example, server 31(s). In particular, Provino describes:

[T]he device 12(m) and firewall 30 can transfer message packets over the secure tunnel. The device 12(m), 15 in generating message packets for transfer over the secure tunnel, makes use of the secure packet processor 26 to encrypt the portions of the message packets which are to be encrypted prior to transmission by the network interface 21 to the ISP 11 for transfer over the Internet 14 to the firewall 30, and to decrypt the encrypted portions of the message packets received by the device 12(m) which are encrypted.

Ex. 1003 (Provino) at 10:13-21. The individual packets described in Provino are encapsulated by firewall 30 by “generat[ing] a message packet for transmission to the device 12(m) including the encrypted response message packet.” Ex. 1003 (Provino) at 14:61-63. Further, Provino explains that “depending on the protocol for transmission of message packets over the communication link 33, the firewall 30 may need to process and/or modify the message packet to conform to the protocol of Internet 14.” Ex. 1008 (Provino) at 15:16-19.

1160. Provino explains that its systems can establish secure tunnels between multiple devices (*e.g.*, device 12(m) and devices 13, *i.e.*, devices in VPN 15) over the Internet. Ex. 1003 (Provino) at 5:66-6:28. Provino also describes systems in

which a request to establish a secure tunnel with a second location is initiated from a first location (*e.g.*, device 12(*m*)), and wherein the second location (*e.g.*, server 31(*s*) or device 13) comprises, for example, a computer. Ex. 1003 (Provino) 6:19-23 (“The firewall 30 and servers 31(*s*) maybe similar to any of the various devices of devices 12(*m*) and 13 and may include, for example, personal computers, computer workstations, and the like”).

1161. Provino describes the process of establishing a secure communication link between two devices as “proceed[ing] in two phases.” Ex. 1003 (Provino) at 12:1-2. Provino describes the first phase as follows:

In the first phase, the device 12(*m*) and virtual private network 15 cooperate to establish a secure tunnel through the Internet 14. In that first phase, the virtual private network 15, in particular the firewall 30 provide the identification of a nameserver 32, and may also provide the encryption and decryption algorithm and key information, as described above.

Ex. 1003 (Provino) at 12:1-8. *See also* Ex. 1003 (Provino) at 12:17-13:25, which describes the first phase actions in more detail.

1162. Provino then explains the second phase as follows:

In the second phase, after the secure tunnel has been established, the device 12(*m*) can use the information provided during the first phase in connection with generating and transferring message packets to one or more servers 31(*s*) in the virtual private network 15, in the process obtaining resolution human-readable Internet addresses to integer

Internet addresses as necessary from the nameserver 32 that was identified by the firewall 30 during the first phase.

Ex. 1003 (Provino) at 12:8-16. *See also* Ex. 1003 (Provino) at 13:26-15:30, which provides additional details on the second phase actions and related actions.

### **Phase 1 – Secure Tunnel Establishment**

1163. Provino explains that in the first phase of establishing a secure communication link between two devices, the device 12(m) requests establishment of a secure tunnel for transfer of data with firewall 30. Ex. 1003 (Provino) at 9:46-47, 12:17-21. The integer Internet address for the firewall can be provided by the operator or by nameserver 17 by resolving a human-readable Internet address for the firewall in the request. Provino explains that nameserver 17 does not have the Internet addresses for devices internal to the virtual private network 15, because it is external to the virtual private network 15. Instead, nameserver 17 has the Internet address for the firewall 30. Ex. 1003 (Provino) at 10:45-55 (“[N]ameserver 17 is not provided with integer Internet addresses for servers 31(s) and other devices which are in the virtual private network 15, except for integer Internet addresses associated with the firewall 30. Thus, the device 12(m), after the operator has entered the human-readable Internet address, will not be able to obtain the integer Internet address of the server 31(s) which is to be accessed from that nameserver 17.”); *see also* Ex. 1003 (Provino) at 8:65-67, 9:52-56.

1164. The firewall 30 of VPN 15 authenticates device 12(m) by determining whether it is authorized to access a server 31(s) residing within the VPN 15. *See* Ex. 1003 (Provino) at 9:56-60, 12:25-36. If the firewall 30 determines that device 12(m) is authorized, the firewall 30 provides the encryption and decryption algorithms and keys used for establishing a secure tunnel with the device 12(m), and thus communication between the device 12(m) devices within and the VPN 15. *See* Ex. 1003 (Provino) at 9:56-10:12, 12:25-36.

1165. Provino thus shows that its secure tunnels will be established automatically—i.e., the VPN 15 will be extended to the device 12(m)—when an authorized user initiates a request to resolve a human-readable domain name associated with a device on the private network via the virtual private network 15. Ex. 1003 (Provino) at 10:56-11:45, 13:3-25, 14:1-38. The firewall 30 will also provide device 12(m) with the address of the internal nameserver, *i.e.*, VPN nameserver 32. Ex. 1003 (Provino) at 10:56-67, 12:37-13:21.

## **Phase 2 –Secure Communication Link**

1166. Provino shows that, after establishment of the secure tunnel, an operator of a device 12(m) enters a human-readable domain name associated with server 31(s). Ex. 1003 (Provino) at 7:37-43, 8:14-26, 9:46-60, 13:26-40. The external device 12(m) will first contact the external nameserver, *i.e.*, nameserver 17, which is a conventional DNS server, in an “attempt to obtain the integer

Internet address associated with the human-readable Internet address.” Ex. 1003 (Provino) at 8:34-57, 10:56-67, 11:5-11. Because the domain name for server 31(s) maps to a device internal to the virtual private network, nameserver 17 will not have the private Internet address associated with server 31(s), and cannot resolve the request, which will cause nameserver 17 to send a response message packet to device 12(m). Ex. 1003 (Provino) at 10:56-11:13. Because it cannot resolve the requested human-readable address associated with server 31(a), nameserver 17 provides the Internet address for firewall 30. See Ex. 1003 (Provino) at 9:52-56 (“The message packet may be directed to a predetermined integer Internet address associated with the firewall 30 which is reserved for secure tunnel establishment requests, and which is known to and provided to the device 12(m) by the nameserver 17.”); 10:45-67; 11:10-25; 12:17-266.

1167. The device 12(m) then sends an encrypted request message to the firewall 30 through the established secure tunnel. See Ex. 1003 (Provino) at 13:63-14:26. The request message includes address information for the VPN nameserver 32—transmitted previously by firewall 30 during establishment of the secure tunnel—and the human-readable domain name of the server 31(s). *Id.* Firewall 30 decrypts the request message and transmits it over the communication link 33 to VPN nameserver 32. The message includes the human-readable domain name associated with server 31(s) and it sent for resolution by VPN nameserver 32. Ex.

1003 (Provino) at 14:27-34; *see also id. at* 10:13-33. Ex. 1003 (Provino) at 14:27-34

1168. If the VPN nameserver can resolve the name, it will return the integer Internet address of server 31(s) to firewall 30, which transmits it to device 12(m) over the secure tunnel. Ex. 1003 (Provino) at 11:13-25; 14:42-46. Device 12(m) and server 31(s) can then communicate directly through the secure tunnel established by firewall 30. Ex. 1003 (Provino) at 13:26-31. Provino shows that the integer Internet address of the server 31(s) is an IP address. As explained above, prior to creation of the secure tunnel, the firewall 30 must authenticate the device 12(m). Ex. 1003 (Provino) at 9:56-60. Thus, in order for the device 12(m) to access the server 31(s), Provino explains that the device 12(m) must be authorized to do so. Accordingly, the integer Internet address of the server 31(s) is a network address that requires authorization for access and is associated with a computer (i.e., server 31(s)) configured to be accessed through a virtual private network. Provino also explains that the integer Internet address of the server 31(s) is “in the form of an ‘n’-bit integer (where ‘n’ may be thirty two or 128),” which are the number of bits in Internet Protocol Version 4 and 6 IP addresses, respectively. Ex. 1003 (Provino) at 1:45-47. Therefore, the integer Internet address of the server 31(s) is an IP address.

1169. Provino explains that the human-readable domain name of the server 31(s) may be a non-standard domain name that corresponds to a secure computer network address (i.e., integer Internet address of the server 31(s)) that cannot be resolved by a conventional name service. In particular, Provino describes that the human-readable domain name of the server 31(s) may be “any form of secondary or informal network address arrangement.” Ex. 1008 (Provino) at 16:12-17. In other words, the virtual private network 15 described in Provino is configured to support non-standard human readable domain names. Also, as described above, conventional nameserver 17 cannot resolve the human-readable domain names of the servers 31(s) internal to the VPN 15. Ex. 1003 (Provino) at 11:11-14.

1170. Provino shows that the VPN nameserver 32 is a service that can resolve secure computer network addresses for a secure name for which a conventional name service cannot resolve addresses. In particular, VPN nameserver 32 is behind firewall 30 on the private network, within VPN 15 and specifically configured to resolve the human-readable domain names of the servers 31(s) internal to the VPN 15. Ex. 1003 (Provino) at 8:67-9:5.

1171. Provino shows that certain names are associated with devices in the private network 15 (e.g., server 31(s)) which require authentication of requesting device 12(m) and which are reserved for secure tunnel establishment requests, and

therefore those secure names indicate security. Ex. 1003 (Provino) at 9:53-55 (“The message packet may be directed to a predetermined integer Internet address associated with the firewall 30 which is reserved for secure tunnel establishment requests.”)

1172. The human-readable names described in Provino that are associated with devices in the private network 15—*e.g.*, the human-readable Internet address of server 31(s)—would be considered “secure name(s)” because they can only be resolved by a “secure name service,” (*e.g.* nameserver 32) and are associated with and/or can be used in establishing a secure communication link. Ex. 1003 (Provino) at 7:37-43, 8:64-9:5; *see also* Ex. 1003 (Provino) at 9:32-10:13 (describing creation of “secure tunnel” between device 12(m) and devices within VPN 15 through firewall 30); 16:12-17. The human readable domain names of server 31(s) may be “any form of secondary or informal network address arrangement.” Ex. 1003 (Provino) at 16:12-17. Thus, the VPN 15 described in Provino is configured to support non-standard human readable domain names. In addition, the human-readable names associated with devices in the virtual private network described in Provino cannot be resolved by a conventional DNS server. Ex. 1003 (Provino) at 11:11-14.

1173. Provino describes a system that includes a domain name database (*e.g.*, nameservers 17 and 32 which necessarily include a list of domain names).



See ¶¶ 73, 91-110, 127-133. Each nameserver stores a list of domain names and corresponding network addresses for communication for use in resolving requests from clients. *Id.*; Ex. 1003 (Provino) at Figure 1; *id.* at 1:60-2:5; *id.* at 10:56-11:45. Ex. 1003 (Provino) at 8:57-9:5 (describing use of nameserver 32 to provide network addresses of secure domain names corresponding to servers on the virtual private network).

1174. Provino describes secure DNS systems comprising a domain name database that stores human-readable domain names and corresponding network addresses. Ex. 1003 (Provino) at 1:67-2:5 (nameservers 17 and 32 each store domain names and network addresses as part of a domain name database); *id.* at 8:34-52; 11:5-25. For example, Provino explains:

**If the nameserver 32 has an integer Internet address associated with the human-readable Internet address in the request message packet provided by the device 12(m), it will provide the integer Internet address in a manner that is generally similar to that described above in connection with nameserver 18.**

Ex. 1003 (Provino) at 11:5-25 (emphasis added). One of ordinary skill in the art would also recognize that because nameservers 17 and 32 store domain names and associated integer Internet addresses as part of a domain name database, those stored domain names and associated integer Internet addresses are necessarily registered prior to any query. *See generally* ¶¶ 73-137.

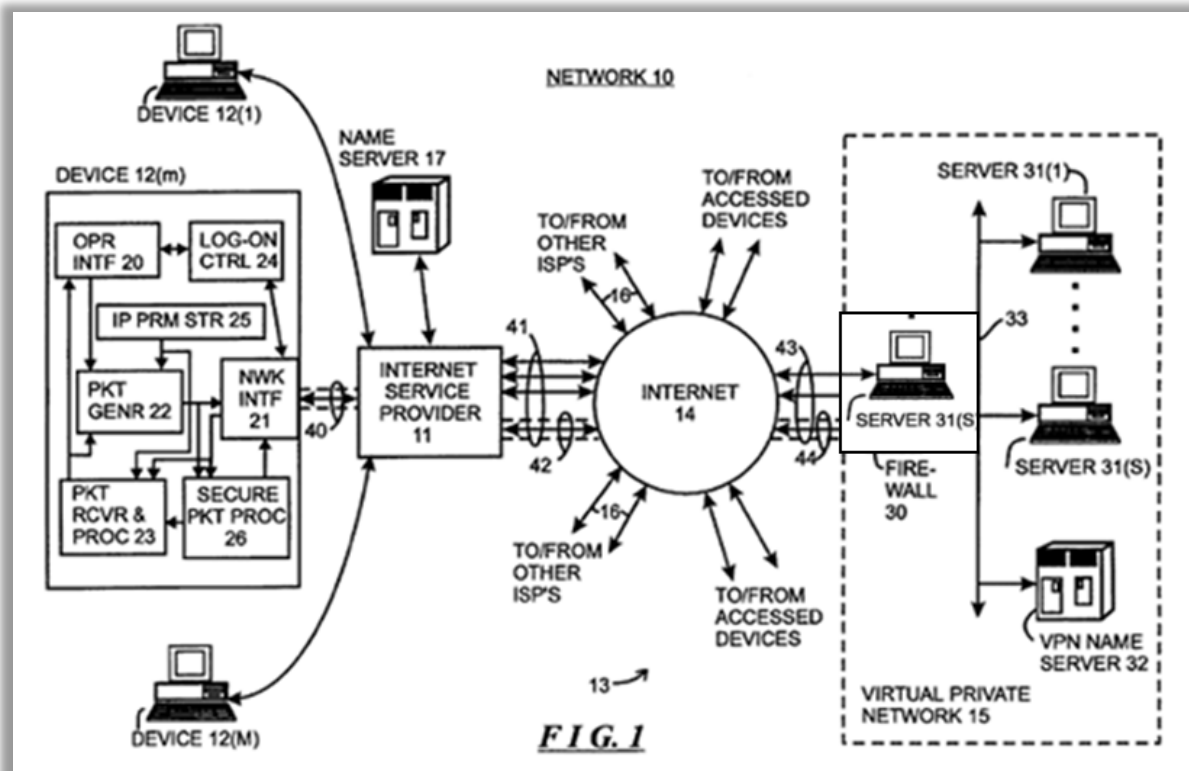
1175. The device server 31(s) is associated with a secure name. The human-readable name associated with integer Internet address of server 31(s) would be considered a “secure name ” because it corresponds to a secure network address and cannot be resolved by a conventional domain name service. Only VPN nameserver 32 can resolve human-readable names associated with server 31(s).

1176. The device server 31(s) is also associated with an integer Internet address that is “in the form of an ‘n’-bit integer (where ‘n’ may be thirty two or 128),” which are the number of bits in Internet Protocol Version 4 and 6 IP addresses, respectively. Ex. 1003 (Provino) at 1:45-47. Therefore, the integer Internet address of the server 31(s) is an IP address, which is accessible by conventional IP protocols.

1177. The human-readable name associated with firewall 30—which facilitates communications with and is associated with server 31(s)—would be considered an “unsecure name” because it can be resolved by a conventional domain name service, such as nameserver 17. I also note that the firewall 30 and server 31(s) may, in fact, be the same device. Ex. 1003 (Provino) at 9:26-31(“The firewall may be similar to other devices 31(s) in the virtual private network 15, with the addition of one or more connections to the Internet.”).

1178. Provino explains that its systems can be configured in any manner that performs the various functions described in the patent. *See generally* Ex. 1003 at

15:65-16:60. For example, Provino explains that “numerous modifications may be made to the arrangement” of the system presented “in connection with FIG 1.” *Id.* As I explain above in ¶ 1177, one modification that is expressly suggested by Provino is combining the functionality of firewall 30 and device 31(s) into a single device, “with the addition of one or more connections to the Internet.” Ex. 1003 at 9:27-31; *see also* Ex. 1003 at 6:19-28. The modification suggested in Provino, i.e., including both firewall functions (i.e., firewall 30 in Provino) and common network server functions (e.g., file server or an intranet web server) into a single device, would also be an obvious design choice for a person of ordinary skill in the art, as it was well known before April 2000 that server computers often hosted multiple services. *See* ¶¶ 190-194. The architecture for this configuration is reproduced below in Annotation “A” of Figure 1:



### ANNOTATION “A” – FIGURE 1

In this configuration, the combined firewall 30/server 31(s) computer in Provino will have (i) registered its human-readable name with the external nameserver, which is a conventional nameserver, in order to facilitate the resolution of requests from external devices that include that human-readable domain name (*i.e.*, the unsecured name) of the firewall, and (ii) registered its internal network name (“secondary address”) with an internal or private nameserver, which is a “secure name” according to the ‘181 patent. Moreover, an access request from an external device that specifies the “secure name” of server 31(s) in this configuration would still require (i) authentication of an external device by the “firewall 30” functions,

(ii) decryption and forwarding of the request by firewall 30 to vpn nameserver 32 over the private network for resolution, and (iii) return of the network address associated with the “secure name” of server 31(s) to the external device.

## **2. Overview of Kosiur**

1179. Kosiur provides information regarding the capabilities of VPNs at or before the time of the '181 patent. In particular, Kosiur is “a book [that] aims to provide you with the background on VPN technologies and products that you need to make appropriate business decisions about the design of a VPN and expectations for its use.” Ex. 1012 (Kosiur) at 9. Chapter 15 of Kosiur “covers the basics of network performance and related application requirements as well as methods for offering network services to your customers that can be differentiated on the basis of those application and/or user requirements.” Ex. 1012 (Kosiur) at 9.

1180. Kosiur explains that “a wide variety of applications can run on networks.” Ex. 1012 (Kosiur) at 244. In particular, Kosiur describes that, if they’re configured for differentiated services, virtual private networks can support “newer applications, such as interactive multimedia and videoconferencing.” Ex. 1012 (Kosiur) at 254. Kosiur explains that VPN traffic may include “file transfers, Web browsing, and e-mail,” in which case the VPN “won't need to be concerned with QoS.” Ex. 1012 (Kosiur) at 249. Conversely, Kosiur explains that VPN traffic may also include “transactional traffic, interactive multimedia, and IP telephony,”

in which case “you’ll need to track developments in QoS technologies.” *Id.* For example, Kosiur describes that a VPN can be configured to “utiliz[e] streaming multimedia,” which would include audio and/or video. Ex. 1012 (Kosiur) at 244. Therefore, what Kosiur is explaining is that VPNs could be configured to support a plurality of application programs, including application programs that support interactive multimedia, file transfers, Web browsing, and e-mail.

1181. Provino does not provide extensive details about the types of applications and services that are supported by VPN 15. However, Provino notes that the device 12(m) may be operated by an “employee of a particular company or governmental agency.” Ex. 1003 (Provino) at 3:54-56. Provino also explains that the devices 12(m) include “network and/or telephony interface devices,” as well as application programs for processing data. Ex. 1003 (Provino) at 4:35-49, 5:18-59; *see also* Ex. 1003 (Provino) at 5:10-13 (describing transfer of information over the Internet through “Web pages or the like”) 28-32 (describing communication sessions and applications, such as Internet browsers, for processing data received in communication session).

1182. In April of 2000, a person of ordinary skill would have considered the guidance in Provino and Kosiur together because configuring the VPN 15 described by Provino to support the explicit applications and services described by Kosiur would have facilitated the as the productivity of the employees operating

devices 12(m) and the goals of the system to extend the security of the VPN 15 to remote locations, such as for an “employee of a particular company or government agency.” Ex. 1003 (Provino) at 3:54-56.

1183. In particular, Kosiur recognizes the increasing mobility of the modern workforce in which businesspersons require productivity applications, such as videoconferencing. *See* Ex. 1012 (Kosiur) at 13. As of 1998, Kosiur tells us that VPNs were commonly configured to support various services and applications, such as videoconferencing. *See* Ex. 1012 (Kosiur) at 254. Indeed, Provino describes that the devices in its system are capable of supporting numerous applications and services using various transfer methods. *See* Ex. 1003 (Provino) at 6:10-50. Moreover, Provino describes the maintenance of its VPN 15 “by a company, governmental agency, organization or the like.” Ex. 1003 (Provino) at 9:6-7. Prior to 2000, it was common for companies or similar organizations to have their employees use interactive multimedia, videoconferencing, file transfers, Web browsing, and e-mail as part of their daily routine to increase the employees’ productivity and mobility. *See* Ex. 1012 (Kosiur) at p. 13. A person of ordinary skill in the art in April of 2000 would have thus found it obvious or combine Provino with Kosiur because that person would recognized that a company or other similar organization would find it desirable to make those resources available to

remotely located employees through the VPN 15 so as to similarly increase the productivity of those remote employees.

### **3. Comparison of Provino to Claims 1-29 of the '181 Patent**

#### **a. Claim 1**

1184. Provino describes systems and methods that establish virtual private networks between devices connected to public networks, such as the Internet, and devices on a private network. *See ¶¶ 1143-1146, 1159-1168, above.*

1185. The devices connected to the Internet include an external device 12(m), and device(s)13, which can include firewall 30 and server 31(s), devices that are located on the private network. *See ¶¶ 1143-1146, above.*

1186. Provino explains that its systems include nameservers—e.g., Nameserver 17 and VPN Nameserver 32—that each resolve certain human-readable names (*e.g.*, domain names) associated with devices in the Provino system into integer Internet addresses. *See ¶¶ 1143-1146, 1151-1154, 1166-1170, 1173-1174, above.*

1187. Provino shows that server 31(s) is associated with both a secure name, the human-readable name associated with the integer Internet address of server 31(s), and an unsecured name(s), the human-readable name associated with firewall 30. *See ¶¶ 1169-1178, above.*

1188. I note that when firewall 30 and server 31(s) operate as separate



devices, the unsecured name of firewall 30 will not be directly related to server 31(s). *See* ¶¶ 1176-1178, *above*. However, firewall 30 is associated with server 31(s), as firewall 30 facilitates all external communications to devices that are internal to VPN 15, such as server 31(s). *See* ¶¶ 1177-1178, *above*. I also note that Provino contemplates that functionalities in firewall 30 and server 31(s) may, in fact, be incorporated into the same device. *Id.* Ex. 1003 (Provino) at 9:26-31 (“The firewall may be similar to other devices 31(s) in the virtual private network 15, with the addition of one or more connections to the Internet.”).

1189. Provino thus shows “*A non-transitory machine-readable medium comprising instructions for a method of communicating with a first device associated with a secure name and an unsecured name,*” as specified by claim 1.

1190. Provino describes a system that uses nameservers (e.g., nameserver 17 and VPN nameserver 32) working in conjunction with firewall 30 to establish an authenticated VPN between a device 12(m), an external device on the public network, and devices on the internal private network protected by the firewall 30. *See* ¶¶ 1143-1146, 1155-1156, 1159-1168, *above*.

1191. Provino shows that firewall 30 regulates access by the external devices (*i.e.*, devices connected to a public network, such as device 12(m)) to internal devices (*i.e.*, devices on the private network, such as server 31(s)), *see* ¶¶ 1155-1156, 1164, *above*, and also facilitates the establishment of the VPN. *See*

¶¶1155-1168, *above*.

1192. Provino shows that the nameservers in its scheme resolve human-readable names into network addresses (e.g., IP addresses). See ¶¶ 1151-1154, 1166-1170, 1173-1174, *above*. Certain of the nameservers reside on the public network (e.g., “nameserver 17”) and resolve conventional domain names, while others (e.g., “nameserver 32”) are accessible only on the private network and resolve names of internal devices on the private network. See ¶¶1151-1154, 1166-1170, 1173-1174, *above*.

1193. In a first phase of the Provino process, a VPN is established between an external device 12(m) and a firewall 30, which includes a step where the external device is authenticated. See ¶¶ 1155-1156, 1157-1168, *above*. If the external device is authenticated, the VPN is extended to the external device 12(m) and firewall 30 returns the network address of the internal VPN nameserver. See ¶¶ 1155-1156, 1157-1168, *above*.

1194. Provino explains that a request from an external device 12(m) to securely communicate with server 31(s) on the private network includes an encrypted request message to firewall 30 through the VPN that specifies address information for the nameserver 32 and the human-readable domain name of the server 31(s). See ¶¶ 1162, 1166-1168, *above*. Firewall 30 decrypts the request message and transmits it over the communication link 33 to nameserver 32. See ¶¶

1155-1156, 1162, 1166-1168, *above*. If server 31(s) had previously registered its human-readable domain name with nameserver 32, the integer Internet address (i.e., the network address) associated with the human-readable domain name will be sent to the device 12(m). *See* ¶¶ 1162, 1166-1168, 1173-1174, *above*.

1195. Provino shows that after the device 12(m) obtains the network address of server 31(s) from nameserver 32, the device 12(m) may send an access request message to securely communicate with server 31(s) using the VPN that is established by firewall 30. *See* ¶ 1155-1168, *above*. The access request message is routed to and received by server 31(s) on the private network.

1196. As I explain above in ¶¶ 1169-1178, Provino also explains that internal devices on the private network are associated with two names: (i) a “secondary address” which is a human-readable name associated with the internal network address of the device (a “secure name”) and (ii) a human-readable name associated with the firewall (an “unsecure name”).

1197. Provino thus shows “*receiving, at a network address corresponding to the secure name associated with the first device, a message from a second device of the desired to securely communicate with the first device,*” as specified by claim 1.

1198. Provino explains that its systems support sending messages from device 12 (m) to server 31(s) over a secure communication link. *See* ¶¶ 1155-

1168, *above*; *see also* Ex. 1003 (Provino) at 9:32-44 (**Communications between devices external to the virtual private network 15, such as device 12(m), and a device, such as a server 31(s), inside the virtual private network 15, may be maintained over a secure tunnel between the firewall 30 and the external device as described above to maintain the information transferred therebetween secret while being transferred over the Internet 14 and through the ISP 11.**) (emphasis added).

1199. Provino thus shows “*sending a message over a secure communication link from the first device to the second device,*” as specified by claim 1.

1200. Accordingly, Provino anticipates claim 1 of the ’181 patent under 35 U.S.C. § 102(e).

#### **b. Claim 2**

1201. Provino describes systems and methods that establish virtual private networks between devices connected to public networks, such as the Internet, and devices on a private network. *See* ¶¶ 1143-1162, *above*.

1202. The devices connected to the Internet include an external device 12(m), and device(s)13, which can include firewall 30 and server 31(s), devices that are located on the private network. *See* ¶¶ 1143-1146, *above*.

1203. Provino explains that its systems include nameservers—e.g., Nameserver 17 and VPN Nameserver 32—that each resolve certain human-

readable names (*e.g.*, domain names) associated with devices in the Provino system into integer Internet addresses. *See ¶¶ 1143-1146, 1151-1154, 1162, above.*

1204. Provino shows that server 31(s) is associated with both a secure name, the human-readable name associated with the integer Internet address of server 31(s), and an unsecured name(s), the human-readable name associated with firewall 30. *See ¶¶ 1169-1178, above.*

1205. Provino thus shows “*A method of using a first device to communicate with a second device having a secure name*” as specified by the preamble of claim 2.

1206. Provino describes a system that uses nameservers (*e.g.*, nameserver 17 and VPN nameserver 32) working in conjunction with firewall 30 to establish an authenticated VPN between a device 12(m), an external device on the public network, and devices on the internal private network protected by the firewall 30. *See ¶¶ 1143-1146, 1155-1156, 1159-1168, above.*

1207. Provino shows that firewall 30 regulates access by the external devices (*i.e.*, devices connected to a public network, such as device 12(m)) to internal devices (*i.e.*, devices on the private network, such as server 31(s)), *see ¶¶ 1155-1156, 1164, above*, and also facilitates the establishment of the VPN. *See ¶¶ 1155-1168, above.*

1208. Provino shows that the nameservers in its scheme resolve human-readable names into network addresses (e.g., IP addresses). See ¶¶ 1151-1154, 1166-1170, 1173-1174, *above*. Certain of the nameservers reside on the public network (e.g., “nameserver 17”) and resolve conventional domain names, while others (e.g., “nameserver 32”) are accessible only on the private network and resolve names of internal devices on the private network. See ¶¶ 1151-1154, 1166-1170, 1173-1174, *above*.

1209. In a first phase of the Provino process, a VPN is established between an external device 12(m) and a firewall 30, which includes a step where the external device is authenticated. See ¶¶ 1155-1156, 1157-1168, *above*. If the external device is authenticated, the VPN is extended to the external device 12(m) and firewall 30 returns the network address of the internal VPN nameserver. See ¶¶ 1155-1156, 1157-1168, *above*.

1210. Provino explains that a request from an external device 12(m) to securely communicate with server 31(s) on the private network includes an encrypted request message to firewall 30 through the VPN that specifies address information for the nameserver 32 and the human-readable domain name of the server 31(s). See ¶¶ 1162, 1166-1168, *above*. Firewall 30 decrypts the request message and transmits it over the communication link 33 to nameserver 32. *Id.* See ¶¶ 1155-1156, 1162, 1166-1168, *above*.

1211. Provino thus shows “*from the first device, sending a message to a secure name service, the message requesting a network address associated with the secure name of the second device,*” as specified in **claim 2**.

1212. Provino shows that If server 31(s) had previously registered its human-readable domain name with nameserver 32, the integer Internet address (i.e., the network address) associated with the human-readable name (i.e., the “secure name”) will be sent to the device 12(m). See ¶ 1155-1168, 1169-1176, *above*.

1213. Provino thus shows “*at the first device, receiving a message containing the network address associated with the secure name of the second device,*” as specified in claim 2.

1214. Provino shows that after the device 12(m) obtains the network address associated with the secure name of server 31(s) from nameserver 32, the device 12(m) may send an access request message to securely communicate with server 31(s) using the VPN that is established by firewall 30. See ¶ 1155-1168, *above*. The access request message is routed to and received by server 31(s) on the private network.

1215. As I explain above in ¶¶ 1169-1178, Provino explains that internal devices on the private network are associated with two names: (i) a “secondary address” which is a human-readable name associated with the internal network

address of the device (a “secure name”) and (ii) a human-readable name associated with the firewall (an “unsecure name”).

1216. Provino shows that its systems supports sending messages between device 12 (m) to the network address associated with the human-readable domain name of server 31(s) over a secure communication link. *See* ¶¶ 1155-1168, *above*; *see also* Ex. 1003 (Provino) at 9:32-44 (**Communications between devices external to the virtual private network 15, such as device 12(m), and a device, such as a server 31(s), inside the virtual private network 15, may be maintained over a secure tunnel between the firewall 30 and the external device as described above to maintain the information transferred therebetween secret while being transferred over the Internet 14 and through the ISP 11.**) (emphasis added).

1217. Provino thus shows “*from the first device, sending a message to the network address associated with the secure name of the second device using a secure communication link,*” as specified in **claim 2**.

1218. Accordingly, Provino anticipates claim 2 of the ’181 patent under 35 U.S.C. § 102(e).

### **c. Claim 3**

1219. Provino shows that devices internal to the virtual private network are associated with a “human-readable domain names” that function as “secondary



addresses.” See ¶¶ 1169-1172, 1175, *above*. These secondary addresses are “secure domain names” as they are non-standard domain names corresponding to the network address of the internal device that cannot be resolved by a conventional name service. *Id.*

1220. Provino thus shows “*The method according to claim 2, wherein the secure name of the second device is a secure domain name,*” as specified by claim 3.

**d. Claim 4**

1221. Provino shows that communications with “secondary addresses” (i.e., destined for internal devices) require authentication through a firewall that is associated with a network address that has been reserved for secure tunnel establishment requests. See ¶¶ 1169-1172, 1175, *above*; see also Ex. 1003 (Provino) at 9:32-26, 53-55 (“The message packet may be directed to a predetermined integer Internet address associated with the firewall 30 **which is reserved for secure tunnel establishment requests, and which is known to and provided to the device 12(m) by the nameserver 17.**”) (emphasis added). The secure name that is associated with a reserved network address thus indicates security. See ¶ 1171, *above*.

1222. Provino thus shows “*The method according to claim 2, wherein the secure name indicates security,*” as specified by claim 4.

**e. Claim 5**

1223. Provino explains that a request from device 12(m) to establish a secure communication link with server 31(s) includes an encrypted request message to firewall 30 through a secure tunnel that specifies address information for the nameserver 32 and the human-readable domain name of the server 31(s). *See ¶¶ 1155-1158, 1161-1168, above.*

1224. The firewall 30 described in the Provino scheme then decrypts the request message and transmits it over the communication link 33 to VPN nameserver 32. *See ¶¶ 1155-1158, 1161-1168, above.*

1225. Provino shows that the message sent to device 12(m) will be encrypted and comprises responses from firewall 30 and VPN nameserver 32 containing the network address of server 31(s). *See ¶¶ 1155-1158, 1161-1168, above; see also Ex. 1003 (Provino) at 14:26-15:30.*

1226. Provino thus shows “*The method according to claim 2, wherein receiving the message containing the network address associated with the secure name of the second device includes receiving the message in encrypted form,*” as specified by claim 5.

**f. Claim 6**

1227. Provino shows that data sent over a secure tunnel from server 31(s) to device 12 (m) is decrypted, including encrypted message packets from firewall 30.

See ¶¶ 1155-1158, 1161-1168, *above*; Ex. 1003 (Provino) at 15:20-30 (“After the device 12(m) receives the message from the firewall 30, it (that is, the message packet) will be provided to the secure packet processor 26. The secure packet processor 26, in turn, will decrypt the encrypted portion of the message packet to obtain the integer Internet address associated with the human-readable Internet address.”).

1228. Provino thus shows “*The method according to claim 5, further including decrypting the message,*” as specified by claim 6.

**g. Claim 7**

1229. Provino explains that its systems supports sending messages between device 12 (m) to server 31(s) over a secure communication link. See ¶¶ 1143-1146, 1155-1168, *above*; *see also* Ex. 1003 (Provino) at 9:32-44

(“Communications between devices external to the virtual private network 15, such as device 12(m), and a device, such as a server 31(s), inside the virtual private network 15, may be maintained over a secure tunnel between the firewall 30 and the external device as described above to maintain the information transferred therebetween secret while being transferred over the Internet 14 and through the ISP 11.”)

1230. Provino explains that internal network devices are typically personal computers, servers, and other computer workstations. See ¶¶ 1148-1149, *above*.

While Provino explains these internal network devices can communicate securely via a VPN with an external device, the devices can also communicate with and be accessed by other internal devices on the private network. Such communications will be unsecured as they are within a trusted (secure) network environment.

1231. Provino also shows that the internal network devices also are able to communicate outside the private network via the firewall. See ¶¶ 1146-1149, *above*. In both instances, the internal network devices (the “second device” in claim 7) are able to support both a secure communication link (i.e., communications over a VPN with the external network device) and ordinary communications with other internal devices or public network resources (e.g., public websites).

1232. Provino thus shows “*The method according to claim 2, wherein the second device is capable of supporting a secure communication link as well as a non-secure communication link, the method further including establishing a non-secure communication link with the second device when needed,*” as specified by claim 7.

#### **h. Claim 8**

1233. Provino explains if server 31(s) had previously registered its human-readable domain name with VPN nameserver 32, the integer Internet address

associated with the human readable domain name will be sent to the device 12(m).

See ¶¶ 1155-1168, 1169-1176, *above*.

1234. Provino shows that the message sent to device 12(m) will be encrypted and comprises responses from firewall 30 and VPN nameserver 32 containing the network address of server 31(s). See ¶¶ 1161-1168, *above*; see also Ex. 1003 (Provino) at 14:26-15:30.

1235. The integer Internet address of the server 31(s) (*i.e.*, the network address) described in Provino can be an IP address. See ¶ 1168, *above*.

1236. Provino thus shows “*The method according to claim 2, wherein receiving a message containing the network address associated with the secure name of the device includes receiving the network address as an IP address associated with the secure name of the device,*” as specified by claim 8.

#### **i. Claim 9**

1237. Provino shows that its secure tunnels will be established automatically—*i.e.*, the VPN 15 will be extended to the device 12(m)—when an authorized user initiates a request to resolve a human-readable domain name associated with a device on the virtual private network. See ¶¶ 1161-1165, *above*.

1238. Provino thus shows “*The method according to claim 2, further including automatically initiating the secure communication link after it is enabled,*” as specified by claim 9.

**j. Claim 10**

1239. Provino describes device 12(m) as sending an encrypted request message to the firewall 30 through the established secure tunnel. *See* Ex. 1003 (Provino) at 13:63-14:26. The request message includes address information for the nameserver 32 and the human-readable domain name of the server 31(s). *Id.* *See also* ¶¶ 1155-1168, *above*.

1240. Firewall 30 thereafter decrypts the request message and transmits the request from device 12(m) over the communication link 33, which includes the human-readable domain name associated with server 31(s), for resolution by VPN nameserver 32. *See* ¶¶ 1155-1161, 1163-1165, *above*; Ex. 1003 (Provino) at 14:27-34; *see also id.* at 10:13-33.

1241. Provino explains if server 31(s) had previously registered its human-readable domain name with nameserver 32, the integer Internet address associated with the human readable domain name will be sent to the device 12(m). *See* ¶¶ 1162, 1166-1168, 1173-1174, *above*.

1242. Provino shows that the message sent to device 12(m) will be encrypted and comprises responses from firewall 30 and VPN nameserver 32 containing the network address of server 31(s). *See* ¶ 1155-1156, 1166-1168, *above*; *see also* Ex. 1003 (Provino) at 14:26-15:30.

1243. Provino thus shows “*The method according to claim 2, wherein receiving a message containing the network address associated with the secure name of the device includes receiving the message at the first device through tunneling within the secure communication link,*” as specified by claim 10.

**k. Claim 11**

1244. Provino describes device 12(m) as sending an encrypted request message to the firewall 30 through the established secure tunnel. *See* Ex. 1003 (Provino) at 13:63-14:26. The request message includes address information for the nameserver 32 and the human-readable domain name of the server 31(s) . *Id.*

1245. Firewall 30 thereafter decrypts the request message and transmits the request from device 12(m) over the communication link 33, which includes the human-readable domain name associated with server 31(s), for resolution by VPN nameserver 32. Ex. 1003 (Provino) at 14:27-34; *see also id. at* 10:13-33.

1246. Provino explains if server 31(s) had previously registered its human-readable domain name with nameserver 32, the integer Internet address associated with the human readable domain name will be sent to the device 12(m). *See ¶¶ 1162, 1166-1168, 1173-1174, above*

1247. Provino shows that the message sent to device 12(m) will be encrypted and comprises responses from firewall 30 and VPN nameserver 32

containing the network address of server 31(s). *See* ¶¶ 1155-1161, 1163-1165, *above*; *see also* Ex. 1003 (Provino) at 14:26-15:30.

1248. Provino describes the message sent from firewall 30 to device 12(m) as a “message packet.” *See* ¶¶ 11565-1160, *above*, *see also* Ex. 1003 (Provino) at 15:20-30.

1249. Provino thus shows “*The method according to claim 2, wherein receiving a message containing the network address associated with the secure name of the device includes receiving the message in the form of at least one tunneled packet,*” as specified by claim 11.

#### **I. Claim 12**

1250. Provino shows that its system can operation using point-to-point protocol or conventional multi-drop network protocol. *See, above*. Ex. 1003 (Provino) at 4:23-49. *See* ¶ 1147, *above*.

1251. The systems described in Provino support many different services and application programs (such as Web browsers) via Internet communications. *See, e.g.,* Ex. 1003 (Provino) at 5:10-13, 28-32 (transfer of information over the Internet through “Web pages or the like”). Web pages inherently support a plurality of services, including audio applications and multimedia applications, are typically transferred via multiple sessions, and can be sent using a number of different communication protocols. *See* ¶¶ 1146-1149, *above*.



1252. Provino also explains that the devices 12(m) include “network and/or telephony interface devices,” as well as application programs for processing data. *See*, 1146-1149, *above*; *see also* Ex. 1003 (Provino) at 4:35-49, 5:18-59 (describing communication sessions and applications, such as Internet browsers, for processing data received in communication session). *See* ¶¶ 1146-1149, *above*.

1253. Provino thus shows “*The method according to claim 2, wherein the receiving and sending of messages includes receiving and sending the messages in accordance with any one of a plurality of communication protocols,*” as specified by claim 12.

**m. Claim 13**

1254. Provino shows that its system can operation using point-to-point protocol or conventional multi-drop network protocol. *See, above*. Ex. 1003 (Provino) at 4:23-49. *See* ¶ 1147, *above*.

1255. The systems described in Provino support many different services and application programs (such as Web browsers) via Internet communications. *See, e.g.*, Ex. 1003 (Provino) at 5:10-13, 28-32 (transfer of information over the Internet through “Web pages or the like”). Web pages inherently support a plurality of services, including audio applications and multimedia applications, are typically transferred via multiple sessions, and can be sent using a number of different communication protocols. *See* ¶¶ 1146-1149, *above*.

1256. Provino also explains that the devices 12(m) include “network and/or telephony interface devices,” as well as application programs for processing data. *See*, 1146-1149, *above*; *see also* Ex. 1003 (Provino) at 4:35-49, 5:18-59 (describing communication sessions and applications, such as Internet browsers, for processing data received in communication session). *See* ¶¶ 1146-1149, *above*.

1257. Provino thus shows “*The method according to claim 2, wherein the receiving and sending of messages through the secure communication link includes multiple sessions,*” as specified by claim 13.

**n. Claim 14**

1258. Provino shows that its system can operation using point-to-point protocol or conventional multi-drop network protocol. *See, above*. Ex. 1003 (Provino) at 4:23-49. *See* ¶ 1147, *above*.

1259. The systems described in Provino support many different services and application programs (such as Web browsers) via Internet communications. *See, e.g.*, Ex. 1003 (Provino) at 5:10-13, 28-32 (transfer of information over the Internet through “Web pages or the like”). Web pages inherently support a plurality of services, including audio applications and multimedia applications, are typically transferred via multiple sessions, and can be sent using a number of different communication protocols. *See* ¶¶ 1146-1149, *above*.

1260. Provino also explains that the devices 12(m) include “network and/or telephony interface devices,” as well as application programs for processing data. *See*, 1146-1149, *above*; *see also* Ex. 1003 (Provino) at 4:35-49, 5:18-59 (describing communication sessions and applications, such as Internet browsers, for processing data received in communication session). *See* ¶¶ 1146-1149, *above*.

1261. Provino thus shows “*The method according to claim 2, further including supporting a plurality of services over the secure communication link,*” as specified by claim 14.

**o. Claim 15**

1262. Provino shows that its system can operation using point-to-point protocol or conventional multi-drop network protocol. *See, above*. Ex. 1003 (Provino) at 4:23-49. *See* ¶ 1147, *above*.

1263. The systems described in Provino support many different services and application programs (such as Web browsers) via Internet communications. *See, e.g.*, Ex. 1003 (Provino) at 5:10-13, 28-32 (transfer of information over the Internet through “Web pages or the like”). Web pages inherently support a plurality of services, including audio applications and multimedia applications, are typically transferred via multiple sessions, and can be sent using a number of different communication protocols. *See* ¶¶ 1146-1149, *above*.

1264. Provino also explains that the devices 12(m) include “network and/or telephony interface devices,” as well as application programs for processing data. *See*, 1146-1149, *above*; *see also* Ex. 1003 (Provino) at 4:35-49, 5:18-59 (describing communication sessions and applications, such as Internet browsers, for processing data received in communication session). *See* ¶¶ 1146-1149, *above*.

1265. Provino thus shows “*The method according to claim 14, wherein the plurality of services comprises a plurality of communication protocols, a plurality of application programs, multiple sessions, or a combination thereof,*” as specified by claim 15.

**p. Claim 16**

1266. Provino shows that its system can operation using point-to-point protocol or conventional multi-drop network protocol. *See, above*. Ex. 1003 (Provino) at 4:23-49. *See* ¶ 1147, *above*.

1267. The systems described in Provino support many different services and application programs (such as Web browsers) via Internet communications. *See, e.g.*, Ex. 1003 (Provino) at 5:10-13, 28-32 (transfer of information over the Internet through “Web pages or the like”). Web pages inherently support a plurality of services, including audio applications and multimedia applications, are typically transferred via multiple sessions, and can be sent using a number of different communication protocols. *See* ¶¶ 1146-1149, *above*.

1268. Provino also explains that the devices 12(m) include “network and/or telephony interface devices,” as well as application programs for processing data. *See*, 1146-1149, *above*; *see also* Ex. 1003 (Provino) at 4:35-49, 5:18-59 (describing communication sessions and applications, such as Internet browsers, for processing data received in communication session). *See* ¶¶ 1146-1149, *above*.

1269. Provino thus shows “*The method according to claim 15, wherein the plurality of application programs comprises video conferencing, e-mail, a word processing program, telephony or a combination thereof,*” as specified by claim 16.

**q. Claim 17**

1270. Provino shows that its system can operation using point-to-point protocol or conventional multi-drop network protocol. *See, above*. Ex. 1003 (Provino) at 4:23-49. *See* ¶ 1147, *above*.

1271. The systems described in Provino support many different services and application programs (such as Web browsers) via Internet communications. *See, e.g.,* Ex. 1003 (Provino) at 5:10-13, 28-32 (transfer of information over the Internet through “Web pages or the like”). Web pages inherently support a plurality of services, including audio applications and multimedia applications, are typically transferred via multiple sessions, and can be sent using a number of different communication protocols. *See* ¶¶ 1146-1149, *above*.

1272. Provino also explains that the devices 12(m) include “network and/or telephony interface devices,” as well as application programs for processing data. *See*, 1146-1149, *above*; *see also* Ex. 1003 (Provino) at 4:35-49, 5:18-59 (describing communication sessions and applications, such as Internet browsers, for processing data received in communication session). *See* ¶¶ 1146-1149, *above*.

1273. Provino thus shows “*The method according to claim 15, wherein the plurality of services comprises audio, video or a combination thereof,*” as specified by claim 17.

**r. Claim 18**

1274. Provino shows that certain names are associated with devices in the private network (*e.g.*, server 31(s)) which, prior to gaining access, require authentication of requesting external devices (*e.g.*, device 12(m)) and which are reserved for secure tunnel establishment requests. *See* ¶¶ 1155-1156, *above*; *see also* Ex. 1003 (Provino) at 9:32-26, 53-55 (“The message packet may be directed to a predetermined integer Internet address associated with the firewall 30 which is reserved for secure tunnel establishment requests, and which is ***known to and provided to the device 12(m) by the nameserver 17.***”) (emphasis added)

1275. Provino thus shows “*The method according to claim 2, wherein the secure communication link is an authenticated link,*” as specified by claim 18.

**s. Claim 19**

1276. The “devices 12(m)” described in Provino that are connected to the Internet (through ISP 11) include personal computers and computer workstations.”  
*See ¶¶1146-1149, above; see also Ex. 1003 (Provino) at 4:23-35, 4:50-5:17.*

1277. Provino thus shows “*The method according to claim 2, wherein the first device is a computer, and the steps are performed on the computer,*” as specified by claim 19.

**t. Claim 20**

1278. The “devices 12(m)” described in Provino that are connected to the Internet (through ISP 11) include personal computers and computer workstations, and communicate using any convenient protocol, such as “point-to-point protocol” (i.e., “PPP”), or any “conventional multi-drop network protocol.” *See ¶¶ 1146-1149, above; See also Ex. 1003 (Provino) at 4:23-35, 4:50-5:17.*

1279. Provino thus shows “*The method according to claim 2, wherein the first device is a client computer connected to a communication network, and the method is performed by the client computer on the communication network,*” as specified by claim 20.

**u. Claim 21**

1280. Provino shows that its scheme is facilitated by resolution of “secondary addresses, such as the Internet’s human-readable addresses, to network

addresses by nameservers connected to the private networks.” Ex. 1003 (Provino) at 2:59-65. See ¶¶ 1169-1178, *above*.

1281. Provino shows that internal network devices are associated with both a secure name (a human-readable name associated with the integer Internet address of the device) and one or more unsecure names (e.g., the human-readable name associated with the firewall 30). See ¶¶ 1169-1178, *above*. Provino also shows that the functionalities of firewall 30 and internal device (server 31(s)) may reside in a single computer, in which case the computer would be associated with both a secure name (the internal network name) and a conventional domain name (an “unsecured” name). See ¶¶ 1177-1178, *above*.

1282. Provino thus shows “*The method according to claim 2, further including providing an unsecured name associated with the device,*” as specified by claim 21.

#### **v. Claim 22**

1283. Provino shows that requests to securely communicate are routed from an external device 12(m) to an internal device (e.g., server 31(s)) through a firewall and an internal nameserver (“nameserver 32”). See ¶¶ 1155-1168, *above*.

1284. Provino explains that the internal nameserver (“nameserver 32”) can use an internal network device name (e.g., a “secondary address”) provided in the request from the external device 12 (m) to determine the network address of the



internal device (e.g., server 31(s)), and will return that network address via the firewall to the requesting external device. *See* 1151-1154, 1166-1170, 1173-1174, *above*.

1285. In the Provino system, the internal nameserver (nameserver 32) establishes associations between the name (e.g., the “secondary address”) of each internal device on the private network and the private network address of that internal device. *See* Ex. 1003 at 3:3-6 (“The internal device also has a secondary address, and the nameserver is configured to provide an association between the secondary address and the network address.”)(emphasis added). The internal nameserver 32 stores these associations in a database and uses that database to respond to name resolution requests— that is how nameservers inherently function (i.e., the nameservers resolve human-readable names (*e.g.*, domain names) into integer Internet addresses). *See* ¶¶ 91-110, 127-133, 1151-1154, 1166-1170, 1173-1174, *above*.

1286. Provino also shows that secondary addresses associated with internal devices can take any form. Ex. 1003 (“Further, although the invention has been described in connection with a network which provides for human-readable network addresses, it will be appreciated that the invention can be used in connection with any network which provides for any form of secondary or informal network address arrangements.”). A person of ordinary skill in April of

2000 would recognize from this that the secondary names of internal devices may be standard or non-standard domain names, a WINS name or another form of a name assigned via the internal network to the internal network device. *See ¶¶ 91-110, 127-133, 1151-1154, 1166-1170, 1173-1174, above.*

1287. Provino explains that internal network devices can be conventional personal computers, servers (e.g., including firewall computers) and other devices. *See ¶¶ 1148-1149, above.* Under conventional networking techniques known well before 2000, when an internal device is connected to an internal network, it will request and be assigned both a unique network address (e.g., an IP address) and a “secondary address” (e.g., a domain name or a WINS name) by a device that regulates the internal network (e.g., a server). Under these techniques, other devices on the internal network can route traffic to this internal device using either of the secondary or network addresses. *See ¶¶ 73-137, 1151-1154, 1166-1170, 1173-1174, above*

1288. Provino also shows that an internal device (e.g., server 31(s))—*i.e.*, the first device of claim 24—registers its human-readable domain name with the internal nameserver (“nameserver 32”). *See ¶¶ 91-110, 127-133, 1151-1154, 1166-1170, 1173-1174, above.* Through this process, the “secondary address” (“secure name”) of the internal device is associated with its internal network address, which enables resolution of requests from external devices that include the human-

readable “secondary address” of the internal device. *See ¶¶ 91-110, 127-133, 1151-1154, 1166-1170, 1173-1174, above.*

1289. Provino thus shows “*The method according to claim 2, wherein the secure name is registered prior to the step of sending a message to a secure name service,*” as specified by claim 22.

**w. Claim 23**

1290. Provino shows that the human-readable name of server 31(s) may be a non-standard domain name that corresponds to a secure computer network address (i.e., integer Internet address of the server 31(s)). Provino describes that the human-readable domain name of the server 31(s) may be “any form of secondary or informal network address arrangement.” *See ¶¶ 1176-1178, above.* Ex. 1008 (Provino) at 16:12-17.

1291. Provino shows that the human-readable name of server 31(s) (i.e., the “secure name”) cannot be resolved by a conventional name service, such as nameserver 17. Ex. 1003 (Provino) at 11:11-14. *See ¶¶ 1176-1178, above*

1292. Provino thus shows “*The method according to claim 2, wherein the secure name of the second device is a secure, non-standard domain name,*” as specified by claim 23.

**x. Claim 24**

1293. Provino describes systems and methods that establish virtual private

networks between devices connected to public networks, such as the Internet, and devices on a private network. *See ¶¶ 1143-1162, above.*

1294. The devices connected to the Internet include an external device 12(m), and device(s)13, which can include firewall 30 and server 31(s), devices that are located on the private network. *See ¶¶ 1143-1146, above.*

1295. Provino explains that its systems include nameservers—e.g., Nameserver 17 and VPN Nameserver 32—that each resolve certain human-readable names (*e.g.*, domain names) associated with devices in the Provino system into integer Internet addresses. *See ¶¶ 1143-1146, 1151-1154, 1162, above.*

1296. Provino shows that server 31(s) is associated with both a secure name, the human-readable name associated with the integer Internet address of server 31(s), and an unsecured name(s), the human-readable name associated with firewall 30. *See ¶¶ 1169-1178, above.*

1297. I note that when firewall 30 and server 31(s) operate as separate devices, the unsecured name of firewall 30 will not be directly related to server 31(s). *See ¶¶ 1177-1178, above.* However, firewall 30 is associated with server 31(s), as firewall 30 facilitates all external communications to devices that are internal to VPN 15, such as server 31(s). *See ¶¶ 1177-1178, above.* I also note that Provino contemplates that functionalities in firewall 30 and server 31(s) may, in fact, be incorporated into the same device. Ex. 1003 (Provino) at 9:26-31 (“The

firewall may be similar to other devices 31(s) in the virtual private network 15, with the addition of one or more connections to the Internet.”).

1298. Provino thus shows “*A method of using a first device to securely communicate with a second device over a communication network,*” as specified in the preamble of claim 24.

1299. Provino shows that requests to securely communicate are routed from an external device 12(m) to an internal device (e.g., server 31(s)) through a firewall and an internal nameserver (“nameserver 32”). See ¶¶ 1143-1146, 1159-1168, *above*.

1300. Provino explains that the internal nameserver (“nameserver 32”) can use an internal network device name (e.g., a “secondary address”) provided in the request from the external device 12 (m) to determine the network address of the internal device (e.g., server 31(s)), and will return that network address via the firewall to the requesting external device. See ¶¶ 1143-1146, 1151-1154, 1166-1170, 1173-1174, *above*.

1301. In the Provino system, the internal nameserver (nameserver 32) establishes associations between the name (e.g., the “secondary address”) of each internal device on the private network and the private network address of that internal device. See Ex. 1003 at 3:3-6 (“The internal device also has a secondary address, and the nameserver is configured to provide an association between the

secondary address and the network address.”)(emphasis added). The internal nameserver 32 stores these associations in a database and uses that database to respond to name resolution requests– that is how nameservers inherently function (i.e., the nameservers resolve human-readable names (*e.g.*, domain names) into integer Internet addresses). See ¶¶ 91-110, 127-133, 1151-1154, 1166-1170, 1173-1174, *above*.

1302. Provino also shows that secondary addresses associated with internal devices can take any form. Ex. 1003 (“Further, although the invention has been described in connection with a network which provides for human-readable network addresses, it will be appreciated that the invention can be used in connection with any network which provides for any form of secondary or informal network address arrangements.”). A person of ordinary skill in April of 2000 would recognize from this that the secondary names of internal devices may be standard or non-standard domain names, a WINS name or another form of a name assigned via the internal network to the internal network device. See ¶¶ 91-110, 127-133, 1151-1154, 1166-1170, 1173-1174, *above*.

1303. Provino explains that internal network devices can be conventional personal computers, servers (*e.g.*, including firewall computers) and other devices. See ¶¶ 1148-1149, *above*. Under conventional networking techniques known well before 2000, when an internal device is connected to an internal network, it will

request and be assigned both a unique network address (e.g., an IP address) and a “secondary address” (e.g., a domain name or a WINS name) by a device that regulates the internal network (e.g., a server). Under these techniques, other devices on the internal network can route traffic to this internal device using either of the secondary or network addresses. See ¶¶ 91-110, 127-133, 1151-1154, 1166-1170, 1173-1174, *above*.

1304. Provino also shows that an internal device (e.g., server 31(s))—*i.e.*, the first device of claim 24—registers its human-readable domain name with the internal nameserver (“nameserver 32”). Through this process, the “secondary address” (“secure name”) of the internal device is associated with its internal network address, which enables resolution of requests from external devices that include the human-readable “secondary address” of the internal device. See ¶¶ 91-110, 127-133, 1151-1154, 1166-1170, 1173-1174, *above*.

1305. Provino thus shows “*at the first device requesting and obtaining registration of a secure name for the first device, the secure name being associated with a network address*” as specified by **claim 24**.

1306. Provino describes a system that uses nameservers (e.g., nameserver 17 and VPN nameserver 32) working in conjunction with firewall 30 to establish an authenticated VPN between a device 12(m), an external device on the public network, and devices on the internal private network protected by the firewall 30.

1307. Provino shows that firewall 30 regulates access by the external devices (*i.e.*, devices connected to a public network, such as device 12(m)) to internal devices (*i.e.*, devices on the private network, such as server 31(s)), *see* ¶¶ 1155-1156, 1164, *above*, and also facilitates the establishment of the VPN. *See* ¶¶ 1155-1168, *above*.

1308. Provino shows that the nameservers in its scheme resolve human-readable names into network addresses (e.g., IP addresses). *See* ¶¶ 1151-1154, 1166-1170, 1173-1174, *above*. Certain of the nameservers reside on the public network (e.g., “nameserver 17”) and resolve conventional domain names, while others (e.g., “nameserver 32”) are accessible only on the private network and resolve names of internal devices on the private network. *See* ¶¶ 1151-1154, 1166-1170, 1173-1174, *above*.

1309. In a first phase of the Provino process, a VPN is established between an external device 12(m) and a firewall 30, which includes a step where the external device is authenticated. *See* ¶¶ 1155-1156, 1157-1168, *above*. If the external device is authenticated, the VPN is extended to the external device 12(m) and firewall 30 returns the network address of the internal VPN nameserver. *See* ¶¶ 1155-1156, 1157-1168, *above*.

1310. Provino explains that a request from an external device 12(m) to securely communicate with server 31(s) on the private network includes an



encrypted request message to firewall 30 through the VPN that specifies address information for the nameserver 32 and the human-readable domain name of the server 31(s). *See ¶¶ 1162, 1166-1168, above.* Firewall 30 decrypts the request message and transmits it over the communication link 33 to nameserver 32. *See ¶¶ 1155-1156, 1162, 1166-1168, above.* If server 31(s) had previously registered its human-readable domain name with nameserver 32, the integer Internet address (i.e., the network address) associated with the human-readable domain name will be sent to the device 12(m). *See ¶¶ 1162, 1166-1168, 1173-1174, above.*

1311. Provino shows that after the device 12(m) obtains the network address of server 31(s) from nameserver 32, the device 12(m) may send an access request message to securely communicate with server 31(s) using the VPN that is established by firewall 30. *See ¶ 1155-1168, above.* The access request message is routed to and received by server 31(s) on the private network.

1312. As I explain above in ¶¶ 1169-1178, Provino also explains that internal devices on the private network are associated with two names: (i) a “secondary address” which is a human-readable name associated with the internal network address of the device (a “secure name”) and (ii) a human-readable name associated with the firewall (an “unsecure name”).

1313. Provino thus shows “*receiving at the network address associated with the secure name of the first device a message from a second device of the desire to securely communicate with the first device*” as specified by claim 24.

1314. Provino explains that its systems supports sending messages between device 12 (m) to server 31(s) over a secure communication link. *See ¶¶ 1155-1168, above; see also Ex. 1003 (Provino) at 9:32-44 (Communications between devices external to the virtual private network 15, such as device 12(m), and a device, such as a server 31(s), inside the virtual private network 15, may be maintained over a secure tunnel between the firewall 30 and the external device as described above to maintain the information transferred therebetween secret while being transferred over the Internet 14 and through the ISP 11.)* (emphasis added).

1315. Provino thus shows “*sending a message securely from the first device to the second device*” as specified by claim 24.

1316. Accordingly, Provino anticipates claim 24 of the ’181 patent under 35 U.S.C. § 102(e).

**y. Claim 25**

1317. Provino shows that requests to securely communicate are routed from an external device 12(m) to an internal device (e.g., server 31(s)) through a firewall and an internal nameserver (“nameserver 32”). *See ¶¶ 1155-1168, above.*

1318. Provino explains that the internal nameserver (“nameserver 32”) can use an internal network device name (e.g., a “secondary address”) provided in the request from the external device 12 (m) to determine the network address of the internal device (e.g., server 31(s)), and will return that network address via the firewall to the requesting external device. *See ¶¶ 1151-1154, 1166-1170, 1173-1174, above.*

1319. In the Provino system, the internal nameserver (nameserver 32) establishes associations between the name (e.g., the “secondary address”) of each internal device on the private network and the private network address of that internal device. *See* Ex. 1003 at 3:3-6 (“The internal device also has a secondary address, and the nameserver is configured to provide an association between the secondary address and the network address.”)(emphasis added). The internal nameserver 32 stores these associations in a database and uses that database to respond to name resolution requests– that is how nameservers inherently function (i.e., the nameservers resolve human-readable names (e.g., domain names) into integer Internet addresses). *See ¶¶ 91-110, 127-133, 1151-1154, 1166-1170, 1173-1174, above.*

1320. Provino also shows that secondary addresses associated with internal devices can take any form. Ex. 1003 (“Further, although the invention has been described in connection with a network which provides for human-readable

network addresses, it will be appreciated that the invention can be used in connection with any network which provides for any form of secondary or informal network address arrangements.”). A person of ordinary skill in April of 2000 would recognize from this that the secondary names of internal devices may be standard or non-standard domain names, a WINS name or another form of a name assigned via the internal network to the internal network device. *See* ¶¶ 91-110, 127-133, 1151-1154, 1166-1170, 1173-1174, *above*.

1321. Provino explains that internal network devices can be conventional personal computers, servers (e.g., including firewall computers) and other devices. *See* ¶¶ 1148-1149, *above*. Under conventional networking techniques known well before 2000, when an internal device is connected to an internal network, it will request and be assigned both a unique network address (e.g., an IP address) and a “secondary address” (e.g., a domain name or a WINS name) by a device that regulates the internal network (e.g., a server). Under these techniques, other devices on the internal network can route traffic to this internal device using either of the secondary or network addresses. *See* ¶¶ 91-110, 127-133.

1322. Provino also shows that an internal device (e.g., server 31(s))—*i.e.*, the first device of claim 24—registers its human-readable domain name with the internal nameserver (“nameserver 32”). *See* ¶¶ 91-110, 127-133, 1151-1154, 1166-1170, 1173-1174, *above*. Through this process, the “secondary address” (“secure

name”) of the internal device is associated with its internal network address, which enables resolution of requests from external devices that include the human-readable “secondary address” of the internal device. *See* ¶¶ 91-110, 127-133, 1151-1154, 1166-1170, 1173-1174, *above*.

1323. Provino explains that its systems supports sending messages between device 12 (m) to server 31(s) over a secure communication link. *See* ¶¶ 1155-1168, *above*; *see also* Ex. 1003 (Provino) at 9:32-44 (**Communications between devices external to the virtual private network 15, such as device 12(m), and a device, such as a server 31(s), inside the virtual private network 15, may be maintained over a secure tunnel between the firewall 30 and the external device as described above to maintain the information transferred therebetween secret while being transferred over the Internet 14 and through the ISP 11.**) (emphasis added).

1324. Provino thus shows “*The method according to claim 24, wherein requesting and obtaining registration of a secure name for the first device comprises using the first device to obtain a registration of the secure name for the first device, and wherein sending a message securely comprises sending the message from the first device to the second device using a secure communication link*” as specified in claim 25.

**z. Claim 26**

1325. Provino describes systems and methods that establish virtual private networks between devices connected to public networks, such as the Internet, and devices on a private network. See ¶¶ 1143-1146, 1159-1168, *above*.

1326. The devices connected to the Internet include an external device 12(m), and device(s)13, which can include firewall 30 and server 31(s), devices that are located on the private network. See ¶¶ 1143-1146, *above*.

1327. Provino explains that its systems include nameservers—e.g., Nameserver 17 and VPN Nameserver 32—that each resolve certain human-readable names (*e.g.*, domain names) associated with devices in the Provino system into integer Internet addresses. See ¶¶ 1151-1154, 1166-1170, 1173-1174, *above*.

1328. Provino thus shows “*A method of using a first device to communicate with a second device over a communication network,*” as specified in the preamble of claim 26.

1329. Provino shows that a computer functioning as firewall 30 will be connected to both the private internal network and the public network (Internet). See ¶¶ 1177-1178, *above*. Because it is connected to the private network, it will request and be assigned, like any other internal network device, a private network address and a “secondary address” using the procedures followed by other internal network devices. In addition, because it is directly connected to the Internet, it will

be assigned a public Internet address and domain name. *See* Ex. 1003 (Provino) at 9:26-31 (“The firewall may be similar to other devices 31(s) in the virtual private network 15, with the addition of one or more connections to the Internet.”).

1330. It was well known before 2000 that server computers often hosted multiple services, including both firewall functions (i.e., firewall 30 in Provino) and common network server functions (e.g., file server or an intranet web server). *See* ¶¶ 190-194, *above*. Provino likewise explains its systems can be configured in any manner that performs the various functions described in the patent. *See* ¶¶ 1177-1178, *above*. This would be understood by a person of ordinary skill as indicating that firewall 30 function could reside on an internal device that is also functioning as a server 31(s) that would be a target of communications from an external device. *Id.*

1331. It was also well known before 2000 that requests for internal network resources hosted on a server connected to both a public network and a private network would be made by other internal network devices using the internal network address of the server, rather be routed through the public address of the firewall computer – doing so avoids the necessity of exposing the internal network traffic to the public network. In this configuration, the firewall 30/server 31(s) computer in Provino will have (i) registered its human-readable name with the external nameserver, which is a conventional nameserver, in order to facilitate the

resolution of requests from external devices that include that human-readable domain name (*i.e.*, the unsecured name) of the firewall, and (ii) registered its internal network name (“secondary address”), which is a “secure name” according to the ’181 patent. *See* ¶¶ 1177-1178, *above*

1332. Provino shows that device server 31(s) is associated an unsecure name(s), the human-readable name associated with firewall 30. *See* ¶ 1177-1178, *above*. *See also* ¶¶ 190-194. I note that when firewall 30 and server 31(s) operate as separate devices, the unsecure name of firewall 30 will not be directly related to server 31(s). *Id.* However, firewall 30 is associated with server 31(s), as firewall 30 facilitates all external communications to devices that are internal to VPN 15, such as server 31(s). I also note that Provino contemplates that functionalities in firewall 30 and server 31(s) may, in fact, be incorporated into the same device. *Id.* Ex. 1003 (Provino) at 9:26-31 (“The firewall may be similar to other devices 31(s) in the virtual private network 15, with the addition of one or more connections to the Internet.”).

1333. Provino thus shows “*from the first device requesting and obtaining registration of an unsecured name associated with the first device*” as specified by claim 26.



1334. Provino shows that requests to securely communicate are routed from an external device 12(m) to an internal device (e.g., server 31(s)) through a firewall and an internal nameserver (“nameserver 32”). *See ¶¶ 1155-1168, above.*

1335. Provino explains that the internal nameserver (“nameserver 32”) can use an internal network device name (e.g., a “secondary address”) provided in the request from the external device 12 (m) to determine the network address of the internal device (e.g., server 31(s)), and will return that network address via the firewall to the requesting external device. *See ¶¶ 1166-1168, above.*

1336. In the Provino system, the internal nameserver (nameserver 32) establishes associations between the name (e.g., the “secondary address”) of each internal device on the private network and the private network address of that internal device. *See Ex. 1003 at 3:3-6 (“The internal device also has a secondary address, and the nameserver is configured to provide an association between the secondary address and the network address.”)(emphasis added).* The internal nameserver 32 stores these associations in a database and uses that database to respond to name resolution requests— that is how nameservers inherently function (i.e., the nameservers resolve human-readable names (e.g., domain names) into integer Internet addresses). *See ¶¶ 91-110, 127-133, 1151-1154, 1166-1170, 1173-1174, above.*

1337. Provino also shows that secondary addresses associated with internal devices can take any form. Ex. 1003 (“Further, although the invention has been described in connection with a network which provides for human-readable network addresses, it will be appreciated that the invention can be used in connection with any network which provides for any form of secondary or informal network address arrangements.”). A person of ordinary skill in April of 2000 would recognize from this that the secondary names of internal devices may be standard or non-standard domain names, a WINS name or another form of a name assigned via the internal network to the internal network device. *See ¶¶ 91-110, 127-133, 1151-1154, 1166-1170, 1173-1174, above.*

1338. Provino explains that internal network devices can be conventional personal computers, servers (e.g., including firewall computers) and other devices. *See ¶¶ 1148-1149, above.* Under conventional networking techniques known well before 2000, when an internal device is connected to an internal network, it will request and be assigned both a unique network address (e.g., an IP address) and a “secondary address” (e.g., a domain name or a WINS name) by a device that regulates the internal network (e.g., a server). Under these techniques, other devices on the internal network can route traffic to this internal device using either of the secondary or network addresses. *See ¶¶ 91-110, 127-133, 1151-1154, 1166-1170, 1173-1174, above.*

1339. Provino also shows that an internal device (e.g., server 31(s))—*i.e.*, the first device of claim 24—registers its human-readable domain name with the internal nameserver (“nameserver 32”). Through this process, the “secondary address” (“secure name”) of the internal device is associated with its internal network address, which enables resolution of requests from external devices that include the human-readable “secondary address” of the internal device. *See* ¶¶ 91-110, 127-133, 1151-1154, 1166-1170, 1173-1174, *above*.

1340. Provino thus shows “*from the first device requesting and obtaining registration of a secure name associated with the first device, wherein a unique network address corresponds to the secure name associated with the first device*” as specified by **claim 26**.

1341. Provino describes systems configured to receive a request initiated from a device, such as device 12(m), where that that request comprises a message packet that is transferred over the Internet to the firewall 30 requesting establishment of a secure tunnel. *See* ¶ 1155-1165, *above*.

1342. Provino explains that a request from device 12(m) to establish a secure communication link with server 31(s) includes an encrypted request message to the firewall 30 through the secure tunnel that specifies address information for the nameserver 32 and the human-readable domain name of the server 31(s). *See* ¶ 1155-1168, *above*. The firewall 30 decrypts the request message

and transmits it over the communication link 33 to nameserver 32. If server 31(s) had previously registered its human-readable domain name with nameserver 32, the integer Internet address associated with the human readable domain name will be sent to the device 12(m). *See ¶¶ 1155-1168, 1169-1176, above.*

1343. Provino shows that after the device 12(m) obtains the integer Internet address of server 31(s) from nameserver 32, the device 12(m) may send an access request message to securely communicate with server 31(s) using the secure tunnel established by firewall 30, where it is received by server 31(s). *See ¶¶ 1162, 1166-1168, above.*

1344. Provino thus shows “*receiving at the unique network address associated with the secure name a message from a second device requesting the desire to securely communicate with the first device*” as specified by **claim 26**.

1345. Provino explains that its systems supports sending messages between device 12 (m) to server 31(s) over a secure communication link. *See ¶¶ 1155-1168, above; see also Ex. 1003 (Provino) at 9:32-44 (Communications between devices external to the virtual private network 15, such as device 12(m), and a device, such as a server 31(s), inside the virtual private network 15, may be maintained over a secure tunnel between the firewall 30 and the external device as described above to maintain the information transferred*

**therebetween secret while being transferred over the Internet 14 and through the ISP 11.)** (emphasis added).

1346. Provino thus shows “*from the first device sending a message securely from the first device to the second device*” as specified by claim 26.

1347. Accordingly, Provino anticipates claim 26 of the ’181 patent under 35 U.S.C. § 102(e).

**aa. Claim 27**

1348. Provino shows that a computer functioning as firewall 30 will be connected to both the private internal network and the public network (Internet). *See* ¶¶ 1177-1178, *above*. Because it is connected to the private network, it will request and be assigned, like any other internal network device, a private network address and a “secondary address” using the procedures followed by other internal network devices. In addition, because it is directly connected to the Internet, it will be assigned a public Internet address and domain name. *See* Ex. 1003 (Provino) at 9:26-31 (“The firewall may be similar to other devices 31(s) in the virtual private network 15, with the addition of one or more connections to the Internet.”).

1349. It was well known before 2000 that server computers often hosted multiple services, including both firewall functions (i.e., firewall 30 in Provino) and common network server functions (e.g., file server or an intranet web server). *See, e.g.,* ¶¶ *above*. Provino likewise explains its systems can be configured in

any manner that performs the various functions described in the patent. *See ¶¶ 1177-1178, above.* This would be understood by a person of ordinary skill as indicating that firewall 30 function could reside on an internal device that is also functioning as a server 31(s) that would be a target of communications from an external device. *Id.*

1350. It was also well known before 2000 that requests for internal network resources hosted on a server connected to both a public network and a private network would be made by other internal network devices using the internal network address of the server, rather be routed through the public address of the firewall computer – doing so avoids the necessity of exposing the internal network traffic to the public network. In this configuration, the firewall 30/server 31(s) computer in Provino will have (i) registered its human-readable name with the external nameserver, which is a conventional nameserver, in order to facilitate the resolution of requests from external devices that include that human-readable domain name (*i.e.*, the unsecured name) of the firewall, and (ii) registered its internal network name (“secondary address”), which is a “secure name” according to the ’181 patent. *See ¶¶ 1178, above; see also ¶¶ 91-110, 127-133*

1351. Provino shows that device server 31(s) is associated an unsecure name(s), the human-readable name associated with firewall 30. *See ¶¶ 1177-1178, above; See also ¶¶ 190-194, above.* I note that when firewall 30 and server 31(s)

operate as separate devices, the unsecure name of firewall 30 will not be directly related to server 31(s). *Id.* However, firewall 30 is associated with server 31(s), as firewall 30 facilitates all external communications to devices that are internal to VPN 15, such as server 31(s). I also note that Provino contemplates that functionalities in firewall 30 and server 31(s) may, in fact, be incorporated into the same device. *Id.* Ex. 1003 (Provino) at 9:26-31 (“The firewall may be similar to other devices 31(s) in the virtual private network 15, with the addition of one or more connections to the Internet.”).

1352. Provino thus shows the method of claim 26 “*wherein requesting and obtaining registration of an unsecured name associated with the first device comprises using the first device to obtain a registration of the unsecured name associated with the first device,*” as specified by claim 27.

1353. Provino shows that requests to securely communicate are routed from an external device 12(m) to an internal device (e.g., server 31(s)) through a firewall and an internal nameserver (“nameserver 32”). *See ¶¶ 1155-1168, above.*

1354. Provino explains that the internal nameserver (“nameserver 32”) can use an internal network device name (e.g., a “secondary address”) provided in the request from the external device 12 (m) to determine the network address of the internal device (e.g., server 31(s)), and will return that network address via the

firewall to the requesting external device. *See* 1151-1154, 1166-1170, 1173-1174, *above*.

1355. In the Provino system, the internal nameserver (nameserver 32) establishes associations between the name (e.g., the “secondary address”) of each internal device on the private network and the private network address of that internal device. *See* Ex. 1003 at 3:3-6 (“The internal device also has a secondary address, and the nameserver is configured to provide an association between the secondary address and the network address.”)(emphasis added). The internal nameserver 32 stores these associations in a database and uses that database to respond to name resolution requests— that is how nameservers inherently function (i.e., the nameservers resolve human-readable names (*e.g.*, domain names) into integer Internet addresses). *See* ¶¶ 91-110, 127-133, 1151-1154, 1166-1170, 1173-1174, *above*.

1356. Provino also shows that secondary addresses associated with internal devices can take any form. Ex. 1003 (“Further, although the invention has been described in connection with a network which provides for human-readable network addresses, it will be appreciated that the invention can be used in connection with any network which provides for any form of secondary or informal network address arrangements.”). A person of ordinary skill in April of 2000 would recognize from this that the secondary names of internal devices may



be standard or non-standard domain names, a WINS name or another form of a name assigned via the internal network to the internal network device. *See* ¶¶ 91-110, 127-133, 1151-1154, 1166-1170, 1173-1174, *above*.

1357. Provino explains that internal network devices can be conventional personal computers, servers (e.g., including firewall computers) and other devices. *See* ¶¶ 1148-1149, *above*. Under conventional networking techniques known well before 2000, when an internal device is connected to an internal network, it will request and be assigned both a unique network address (e.g., an IP address) and a “secondary address” (e.g., a domain name or a WINS name) by a device that regulates the internal network (e.g., a server). Under these techniques, other devices on the internal network can route traffic to this internal device using either of the secondary or network addresses. *See* ¶¶ 91-110, 127-133, 1151-1154, 1166-1170, 1173-1174, *above*.

1358. Provino also shows that an internal device (e.g., server 31(s))—*i.e.*, the first device of claim 24—registers its human-readable domain name with the internal nameserver (“nameserver 32”). Through this process, the “secondary address” (“secure name”) of the internal device is associated with its internal network address, which enables resolution of requests from external devices that include the human-readable “secondary address” of the internal device. *See* ¶¶ 91-110, 127-133, 1151-1154, 1166-1170, 1173-1174, *above*.

1359. Provino thus also shows “*wherein requesting and obtaining registration of a secure name associated with the first device comprises using the first device to obtain a registration of the secure name associated with the first device,*” as specified by claim 27.

1360. Accordingly, Provino anticipates claim 27 of the ’181 patent under 35 U.S.C. § 102(e).

**bb. Claim 28**

1361. Provino describes systems and methods that establish virtual private networks between devices connected to public networks, such as the Internet, and devices on a private network. *See* ¶¶ 1143-1146, 1155-1168, *above*.

1362. The devices connected to the Internet in Provino include an external device 12(m), and device(s)13, which includes server 31(s), a device that is located on the private network. *See* ¶¶ 1143-1146, *above*.

1363. Provino explains that its systems include nameservers—e.g., Nameserver 17 and VPN Nameserver 32—that each resolve certain human-readable names (*e.g.*, domain names) associated with devices in the Provino system into integer Internet addresses. *See* ¶¶ 1143-1146, 1151-1154, 1166-1170, 1173-1174, *above*.

1364. Provino shows that server 31(s) is associated with both a secure name, the human-readable name associated with the integer Internet address of

server 31(s), and an unsecured name(s), the human-readable name associated with firewall 30. *See* ¶¶ 1169-1178, *above*.

1365. Provino thus shows “A *non-transitory machine-readable medium comprising instructions*,” as specified by claim 28.

1366. Provino describes a system that uses nameservers (e.g., nameserver 17 and VPN nameserver 32) working in conjunction with firewall 30 to establish an authenticated VPN between a device 12(m), an external device on the public network, and devices on the internal private network protected by the firewall 30. *See* ¶¶ 1143-1146, 1155-1156, 1159-1168, *above*.

1367. Provino shows that firewall 30 regulates access by the external devices (*i.e.*, devices connected to a public network, such as device 12(m)) to internal devices (*i.e.*, devices on the private network, such as server 31(s)), *see* ¶¶ 1155-1156, 1164, *above*, and also facilitates the establishment of the VPN. *See* ¶¶ 1155-1168, *above*.

1368. Provino shows that the nameservers in its scheme resolve human-readable names into network addresses (e.g., IP addresses). *See* ¶¶ 1151-1154, 1166-1170, 1173-1174, *above*. Certain of the nameservers reside on the public network (e.g., “nameserver 17”) and resolve conventional domain names, while others (e.g., “nameserver 32”) are accessible only on the private network and

resolve names of internal devices on the private network. *See ¶¶ 1151-1154, 1166-1170, 1173-1174, above.*

1369. In a first phase of the Provino process, a VPN is established between an external device 12(m) and a firewall 30, which includes a step where the external device is authenticated. *See ¶¶ 1155-1156, 1157-1168, above.* If the external device is authenticated, the VPN is extended to the external device 12(m) and firewall 30 returns the network address of the internal VPN nameserver. *See ¶¶ 1155-1156, 1157-1168, above.*

1370. Provino explains that a request from an external device 12(m) to securely communicate with server 31(s) on the private network includes an encrypted request message to firewall 30 through the VPN that specifies address information for the nameserver 32 and the human-readable domain name of the server 31(s). *See ¶¶ 1162, 1166-1168, above.* Firewall 30 decrypts the request message and transmits it over the communication link 33 to nameserver 32. *See ¶¶ 1155-1156, 1162, 1166-1168, above.*

1371. Provino thus shows “*sending a message to a secure name service, the message requesting a network address associated with a secure name of a device*” as specified by **claim 28**.

1372. If server 31(s) had previously registered its human-readable domain name with nameserver 32, the integer Internet address (i.e., the network address)

associated with the human-readable domain name will be sent to the device 12(m).

See ¶¶ 1162, 1166-1168, 1173-1174, *above*.

1373. Provino thus shows “*receiving a message containing the network address associated with the secure name of the device*” as specified by **claim 28**.

1374. Provino shows that after the device 12(m) obtains the network address associated with the secure name of server 31(s) from nameserver 32, the device 12(m) may send an access request message to securely communicate with server 31(s) using the VPN that is established by firewall 30. See ¶ 1155-1168, *above*. The access request message is routed to and received by server 31(s) on the private network. *Id.*

1375. As I explain above in ¶¶ 1169-1178, Provino explains that internal devices on the private network are associated with two names: (i) a “secondary address” which is a human-readable name associated with the internal network address of the device (a “secure name”) and (ii) a human-readable name associated with the firewall (an “unsecure name”).

1376. Provino shows that its systems supports sending messages between device 12 (m) to the network address associated with the human-readable domain name of server 31(s) over a secure communication link. See ¶ 1155-1168, *above*; *see also* Ex. 1003 (Provino) at 9:32-44 (**Communications between devices external to the virtual private network 15, such as device 12(m), and a device,**

such as a server 31(s), inside the virtual private network 15, may be maintained over a secure tunnel between the firewall 30 and the external device as described above to maintain the information transferred therebetween secret while being transferred over the Internet 14 and through the ISP 11.) (emphasis added).

1377. Provino thus shows “*sending a message to the network address associated with the secure name of the device using a secure communication link*” as specified by claim 28.

1378. Accordingly, Provino anticipates claim 28 of the ’181 patent under 35 U.S.C. § 102(e).

**cc. Claim 29**

1379. Provino describes systems and methods that establish virtual private networks between devices connected to public networks, such as the Internet, and devices on a private network. *See ¶¶ 1143-1162, above.*

1380. The devices connected to the Internet include an external device 12(m), and device(s)13, which can include firewall 30 and server 31(s), devices that are located on the private network. *See ¶¶ 1143-1146, above.*

1381. Provino explains that its systems include nameservers—e.g., Nameserver 17 and VPN Nameserver 32—that each resolve certain human-readable names (*e.g.*, domain names) associated with devices in the Provino

system into integer Internet addresses. *See* ¶¶ 1143-1146, 1151-1154, 1162, *above*.

1382. Provino shows that server 31(s) is associated with both a secure name, the human-readable name associated with the integer Internet address of server 31(s), and an unsecured name(s), the human-readable name associated with firewall 30. *See* ¶¶ 1169-1178, *above*.

1383. Provino thus shows “*A non-transitory machine-readable medium comprising instructions for a method of communicating with a device having a secure name,*” as specified by the preamble of claim 29.

1384. Provino describes a system that uses nameservers (e.g., nameserver 17 and VPN nameserver 32) working in conjunction with firewall 30 to establish an authenticated VPN between a device 12(m), an external device on the public network, and devices on the internal private network protected by the firewall 30. *See* ¶¶ 1143-1146, 1155-1156, 1159-1168, *above*.

1385. Provino shows that firewall 30 regulates access by the external devices (*i.e.*, devices connected to a public network, such as device 12(m)) to internal devices (*i.e.*, devices on the private network, such as server 31(s)), *see* ¶¶ 1155-1156, 1164, *above*, and also facilitates the establishment of the VPN. *See* ¶¶ 1155-1168, *above*.

1386. Provino shows that the nameservers in its scheme resolve human-readable names into network addresses (e.g., IP addresses). *See* ¶¶ 1151-

1154,1166-1170, 1173-1174, *above*. Certain of the nameservers reside on the public network (e.g., “nameserver 17”) and resolve conventional domain names, while others (e.g., “nameserver 32”) are accessible only on the private network and resolve names of internal devices on the private network. *See ¶¶1151-1154,1166-1170, 1173-1174, above.*

1387. In a first phase of the Provino process, a VPN is established between an external device 12(m) and a firewall 30, which includes a step where the external device is authenticated. *See ¶¶ 1155-1156, 1157-1168, above.* If the external device is authenticated, the VPN is extended to the external device 12(m) and firewall 30 returns the network address of the internal VPN nameserver. *See ¶¶ 1155-1156, 1157-1168, above.*

1388. Provino explains that a request from an external device 12(m) to securely communicate with server 31(s) on the private network includes an encrypted request message to firewall 30 through the VPN that specifies address information for the nameserver 32 and the human-readable domain name of the server 31(s). *See ¶¶ 1162, 1166-1168, above.* Firewall 30 decrypts the request message and transmits it over the communication link 33 to nameserver 32. *Id.* *See ¶¶ 1155-1156, 1162, 1166-1168, above.*

1389. Provino explains that a request from an external device 12(m) to securely communicate with server 31(s) on the private network includes an



encrypted request message to firewall 30 through the VPN that specifies address information for the nameserver 32 and the human-readable domain name of the server 31(s). *See* ¶ 1155-1168, *above*. Firewall 30 decrypts the request message and transmits it over the communication link 33 to nameserver 32. *Id.* If server 31(s) had previously registered its human-readable domain name with nameserver 32, the integer Internet address (i.e., the network address) associated with the human-readable domain name will be sent to the device 12(m). *See* ¶ 1155-1168, 1169-1178, *above*.

1390. Provino shows that after the device 12(m) obtains the network address of server 31(s) from nameserver 32, the device 12(m) may send an access request message to securely communicate with server 31(s) using the VPN that is established by firewall 30. *See* ¶ 1155-1168, *above*. The access request message is routed to and received by server 31(s) on the private network.

1391. As I explain above in ¶¶ 1169-1178, Provino also explains that internal devices on the private network are associated with two names: (i) a “secondary address” which is a human-readable name associated with the internal network address of the device (a “secure name”) and (ii) a human-readable name associated with the firewall (an “unsecure name”).

1392. Provino thus shows “*receiving at a network address associated with a secure name of a first device a message from a second device requesting the*

*desired to securely communicate with the first device, wherein the secure name of the first device is registered” as specified by claim 29.*

1393. Provino explains that its systems supports sending messages between device 12 (m) to server 31(s) over a secure communication link. *See ¶ 1155-1168, above; see also Ex. 1003 (Provino) at 9:32-44 (**Communications between devices external to the virtual private network 15, such as device 12(m), and a device, such as a server 31(s), inside the virtual private network 15, may be maintained over a secure tunnel between the firewall 30 and the external device as described above to maintain the information transferred therebetween secret while being transferred over the Internet 14 and through the ISP 11.**)* (emphasis added)

1394. Provino thus shows “*sending a message securely from the first device to the second device*” as specified by claim 29.

1395. Accordingly, Provino anticipates claim 29 of the ’181 patent under 35 U.S.C. § 102(e).

#### **E. Provino Renders Obvious Claims 21, 26, and 27**

1396. As I explain above, Provino shows systems that anticipate claims 21, 26 and 27 for the reasons provided in §§u, z, aa. If it is found that Provino teaches that firewall 30 must reside in a computer separate from an internal device on the

private network to which communications from an external device are requested, this distinction, if established, would not render claims 21, 26 and 27 patentable.

1397. A person of ordinary skill would recognize based on his or her general knowledge and experience, and from the guidance in Provino, that one could locate the server 31(s) functionality on the same computer that hosts the firewall 30 functionality. *See, e.g.*, ¶ 1178, *above*; *see also* ¶¶ 190-194, Ex. 1003 at 9:27-31 (“The firewall may be similar to other devices 31(s) in the virtual private network 15, with the addition of one or more connections to the Internet, which are generally identified by reference numeral 43.”)

1398. In such a configuration, the server 31(s) services (e.g., a file server) would be located on a computer on the private network and would be accessed by the external device through the private network using the firewall and private nameserver 32 technique described in Provino. *Id.* The server computer also would be connected to the public network to support its firewall 30 functionality. The server computer in this configuration would register and obtain both a conventional domain name and a “secondary address” on the private network. This configuration thus would satisfy the provisions specified in claims 21, 26 and 27, and would thereby render those claims obvious.

**F. Provino in View of Beser Renders Obvious Claims 12-17**

1399. Claims 12-17 are anticipated by Provino for the reasons set forth in §§ to 1 - q, *above*. Patent Owner may contend Provino does not anticipate these claims for certain reasons. For the reasons presented below, even if Patent Owner's contentions were accepted, a person of ordinary skill in the art would have considered claims 12 to 17 to have been obvious in April of 2000 based on Provino in view of Beser, which is described in detail from ¶¶ 1349-1381.

**a. Claim 12**

1400. Beser discloses that terminating network devices can include telephony devices, personal computers, and multimedia devices. *See* 297-304, *above*. The data that is to be transmitted through a Beser IP tunnel can represent content for web pages (e.g., delivered to WebTV devices or decoders or personal computers), "multimedia" content (e.g., video, audio) or VOIP traffic. *Id.*

1401. Beser describes a process that can provide additional security on top of traditional secure communications processes, and consequently, a person of ordinary skill in the art would have thought it obvious and common sense to use the Beser schemes with existing VPN systems, such as the systems described in Provino. *See* ¶¶ 357-362, *above*; Ex. 1031 (Beser) at 3:4-9 ("The method and system described herein **may help ensure** that the addresses of the ends of the tunneling association are hidden on the public network and **may increase the**

**security** of communications without an increased computational burden”).

1402. Beser teaches network designs in which devices 14 and 16 are used as intermediaries in connecting an originator and destination and in establishing a secure IP tunnel. *See* Ex. 1031 (Beser) at ¶¶286, 305-308. A person of ordinary skill in the art would have considered relevant the implementation concepts for establishing IP tunnels taught by Beser. Provino, for example, teaches that its systems use intermediary devices to connect a requesting device and a destination on a VPNs over a secure tunnel. *See* ¶¶ 1155-1168.

1403. A person of ordinary skill in the art in April 2000 would have found it simple and common sense to combine Provino, which discusses use of standard to develop a flexible, useful communication service that facilitate its goals of securely extending functionality of a corporate or government agency to remote locations.

1404. Beser explains that the data that can be transmitted through an IP tunnel can represent VOIP traffic, “multimedia” content (*e.g.*, video, audio), or content for web pages (*e.g.*, delivered to WebTV devices or decoders or personal computers). *See* ¶¶ 298-304, *above*. The data that is transmitted through a Beser IP tunnel also can represent video conference traffic or other multimedia content as specified in H.323. *See* ¶ 304, *above*. Both Beser and Provino also indicate that their VPN systems support communications sent via conventional Internet data transport protocols (*e.g.*, TCP/IP, UDP), and conventional Internet networking

protocols (e.g., web browsing or data transfers that inherently use multiple discrete HTTP sessions to transfer data). *See* ¶ 286, above; Beser at 4:55-5:14.

1405. Beser explains that the data that can be transmitted through an IP tunnel can represent VOIP traffic, “multimedia” content (e.g., video, audio), or content for web pages (e.g., delivered to WebTV devices or decoders or personal computers). *See* ¶¶ 298-304, above. The data that is transmitted through a Beser IP tunnel also can represent video conference traffic or other multimedia content as specified in H.323. *See* ¶ 304, above.

1406. Data transmitted through a Beser IP tunnel can represent data packets of another protocol such as UDP, TCP, SNMP, or TFTP. *See* ¶ 299, above.

1407. Provino in view of Beser thus shows “*The method according to claim 2, wherein the receiving and sending of messages includes receiving and sending the messages in accordance with any one of a plurality of communication protocols,*” as specified by claim 12.

**b. Claim 13**

1408. Beser indicates that terminating network devices can include telephony devices, personal computers, and multimedia devices. *See* ¶ 297, above.

1409. Beser explains that the data that can be transmitted through an IP tunnel can represent VOIP traffic, “multimedia” content (e.g., video, audio), or content for web pages (e.g., delivered to WebTV devices or decoders or personal

computers). *See* ¶¶ 298-304, *above*. The data that is transmitted through a Beser IP tunnel also can represent video conference traffic or other multimedia content as specified in H.323. *See* ¶ 304, *above*.

1410. When transmitting multimedia content, each media type (*e.g.*, audio and video) would be transmitted in its own session.

1411. For the same reasons I provided for claim 12 above, a person of ordinary skill in the art would have combined the teachings of Beser and Provino. *See* ¶¶ 1349-1356, *above*.

1412. Provino in view of Beser thus shows “*The method according to claim 2, wherein the receiving and sending of messages through the secure communication link includes multiple sessions,*” as specified by claim 13.

### **c. Claim 14**

1413. Beser indicates that terminating network devices can include telephony devices, personal computers, and multimedia devices. *See* ¶ 297, *above*.

1414. Beser explains that the data that can be transmitted through an IP tunnel can represent VOIP traffic, “multimedia” content (*e.g.*, video, audio), or content for web pages (*e.g.*, delivered to WebTV devices or decoders or personal computers). *See* ¶¶ 298-304, *above*. The data that is transmitted through a Beser IP tunnel also can represent video conference traffic or other multimedia content as specified in H.323. *See* ¶ 304, *above*.

1415. Data transmitted through a Beser IP tunnel can represent data packets of another service or protocol such as UDP, TCP, SNMP, or TFTP. *See* ¶ 299, *above*.

1416. For the same reasons I provided for claim 12 above, a person of ordinary skill in the art would have combined the teachings of Beser and Provino. *See* ¶¶ 1349-1356, *above*.

1417. Provino in view of Beser thus shows “*The method according to claim 2, further including supporting a plurality of services over the secure communication link,*” as specified by claim 14.

**d. Claim 15**

1418. Beser indicates that terminating network devices can include telephony devices, personal computers, and multimedia devices. *See* ¶ 297, *above*.

1419. Beser explains that the data that can be transmitted through an IP tunnel can represent VOIP traffic, “multimedia” content (*e.g.*, video, audio), or content for web pages (*e.g.*, delivered to WebTV devices or decoders or personal computers). *See* ¶¶ 298-304, *above*. The data that is transmitted through a Beser IP tunnel also can represent video conference traffic or other multimedia content as specified in H.323. *See* ¶ 304, *above*.



1420. Data transmitted through a Beser IP tunnel can represent data packets of another service or protocol such as UDP, TCP, SNMP, or TFTP. *See* ¶ 299, *above*.

1421. For the same reasons I provided for claim 12 above, a person of ordinary skill in the art would have combined the teachings of Beser and Provino. *See* ¶¶ 1349-1356, *above*.

1422. Provino in view of Beser thus shows “*The method according to claim 14, wherein the plurality of services comprises a plurality of communication protocols, a plurality of application programs, multiple sessions, or a combination thereof,*” as specified by claim 15.

**e. Claim 16**

1423. Beser indicates that terminating network devices can include telephony devices, personal computers, and multimedia devices. *See* ¶ 297, *above*.

1424. Beser explains that the data that can be transmitted through an IP tunnel can represent VOIP traffic, “multimedia” content (*e.g.*, video, audio), or content for web pages (*e.g.*, delivered to WebTV devices or decoders or personal computers). *See* ¶¶ 298-304, *above*. The data that is transmitted through a Beser IP tunnel also can represent video conference traffic or other multimedia content as specified in H.323. *See* ¶ 304, *above*.

1425. Data transmitted through a Beser IP tunnel can represent data packets of another service or protocol such as UDP, TCP, SNMP, or TFTP. *See* ¶ 299, *above*.

1426. For the same reasons I provided for claim 12 above, a person of ordinary skill in the art would have combined the teachings of Beser and Provino. *See* ¶¶ 1349-1356, *above*.

1427. Provino in view of Beser thus shows “*The method according to claim 15, wherein the plurality of application programs comprises video conferencing, e-mail, a word processing program, telephony or a combination thereof,*” as specified by claim 16.

**f. Claim 17**

1428. Beser indicates that terminating network devices can include telephony devices, personal computers, and multimedia devices. *See* ¶ 297, *above*.

1429. Beser explains that the data that can be transmitted through an IP tunnel can represent VOIP traffic, “multimedia” content (*e.g.*, video, audio), or content for web pages (*e.g.*, delivered to WebTV devices or decoders or personal computers). *See* ¶¶ 298-304, *above*. The data that is transmitted through a Beser IP tunnel also can represent video conference traffic or other multimedia content as specified in H.323. *See* ¶ 304, *above*.

1430. Data transmitted through a Beser IP tunnel can represent data packets of another service or protocol such as UDP, TCP, SNMP, or TFTP. *See* ¶ 299, *above*.

1431. For the same reasons I provided for claim 12 above, a person of ordinary skill in the art would have combined the teachings of Beser and Provino. *See* ¶¶ 1349-1356, *above*.

1432. Provino in view of Beser thus shows “*The method according to claim 15, wherein the plurality of services comprises audio, video or a combination thereof,*” as specified by claim 17.

#### **G. Provino In View of Kosiur Renders Obvious Claims 12-17**

Claims 12-17 are anticipated by Provino for the reasons set forth in §§ 1 - q, *above*. Patent Owner may contend Provino does not anticipate these claims for certain reasons. For the reasons presented below, even if Patent Owner’s contentions were accepted, a person of ordinary skill in the art would have considered claims 12 to 17 to have been obvious in April of 2000 based on Provino in view of Kosiur.

##### **a. Claim 12**

1433. Kosiur explains that VPNs can be configured to support “newer applications, such as interactive multimedia and videoconferencing.” Ex. 1006 (Kosiur) at 254. In particular, Kosiur shows that VPNs can “utiliz[e] streaming

multimedia,” which would encompass both audio and/or video transmissions. Ex. 1006 (Kosiur) at 244.

1434. Kosiur shows that VPN traffic may include “file transfers, Web browsing, and e-mail,” in which case the VPN “won't need to be concerned with QoS.” Ex. 1006 (Kosiur) at 249.

1435. Kosiur also explains that VPN traffic may also include “transactional traffic, interactive multimedia, and IP telephony,” in which case “you’ll need to track developments in QoS technologies.” Ex. 1006 (Kosiur) at 249.

1436. Kosiur thus shows that a VPN may support, for example, applications that provide access to streaming multimedia, file transfers, Web browsing, and e-mail, *see* Ex. 1006 (Kosiur) at 243-254, and thus a person of ordinary skill in the art in April 2000 would recognize that Kosiur shows that VPNs can inherently support a number of communication protocols—such as HTTP, FTP, or TCP/IP. *Id.*

1437. A person of ordinary skill in the art in April 2000 would have found it simple and common sense to combine Kosiur with Provino, which discusses use of standard to develop a flexible, useful communication service including a virtual private network that facilitate its goals of securely extending functionality of a corporate or government agency to remote locations.

1438. Provino in view of Kosiur thus shows “*The method according to claim 2, wherein the receiving and sending of messages includes receiving and sending the messages in accordance with any one of a plurality of communication protocols,*” as specified by claim 12.

**b. Claim 13**

1439. Kosiur explains that VPNs can be configured to support “newer applications, such as interactive multimedia and videoconferencing.” Ex. 1006 (Kosiur) at 254. In particular, Kosiur shows that VPNs can “utiliz[e] streaming multimedia,” which would encompass both audio and/or video transmissions. Ex. 1006 (Kosiur) at 244.

1440. Kosiur shows that VPN traffic may include “file transfers, Web browsing, and e-mail,” in which case the VPN “won't need to be concerned with QoS.” Ex. 1006 (Kosiur) at 249.

1441. Kosiur also explains that VPN traffic may also include “transactional traffic, interactive multimedia, and IP telephony,” in which case “you’ll need to track developments in QoS technologies.” Ex. 1006 (Kosiur) at 249.

1442. Kosiur thus shows that a VPN may support, for example, applications that provide access to streaming multimedia, file transfers, Web browsing, and e-mail, *see* Ex. 1006 (Kosiur) at 243-254, and thus Kosiur also shows that because VPNs can support a number of services—such as multimedia, file transfers, or

Web browsing—it would necessary have the capability to transmit data through the VPN over multiple sessions. *Id.*

1443. For the same reasons I provided for claim 12 above, a person of ordinary skill in the art would have combined the teachings of Kosiur and Provino. See ¶¶ 1382-1384, *above*.

1444. Provino in view of Kosiur thus shows “*The method according to claim 2, wherein the receiving and sending of messages through the secure communication link includes multiple sessions,*” as specified by claim 13.

#### **c. Claim 14**

1445. Kosiur explains that VPNs can be configured to support “newer applications, such as interactive multimedia and videoconferencing.” Ex. 1006 (Kosiur) at 254. In particular, Kosiur shows that VPNs can “utiliz[e] streaming multimedia,” which would encompass both audio and/or video transmissions. Ex. 1006 (Kosiur) at 244.

1446. Kosiur shows that VPN traffic may include “file transfers, Web browsing, and e-mail,” in which case the VPN “won't need to be concerned with QoS.” Ex 1006 (Kosiur) at 249.

1447. Kosiur also explains that VPN traffic may also include “transactional traffic, interactive multimedia, and IP telephony,” in which case “you’ll need to track developments in QoS technologies.” Ex 1006 (Kosiur) at 249.

1448. Kosiur thus shows that a VPN may support, for example, applications that provide access to streaming multimedia, file transfers, Web browsing, and e-mail, *see* Ex. 1006 (Kosiur) at 243-254, and thus Kosiur also shows that VPNs can be configured to support a plurality of services. *Id.*

1449. For the same reasons I provided for claim 12 above, a person of ordinary skill in the art would have combined the teachings of Kosiur and Provino. *See* ¶¶ 1382-1384, *above*.

1450. Provino in view of Kosiur thus shows “*The method according to claim 2, further including supporting a plurality of services over the secure communication link,*” as specified by claim 14.

**d. Claim 15**

1451. Kosiur explains that VPNs can be configured to support “newer applications, such as interactive multimedia and videoconferencing.” Ex. 1006 (Kosiur) at 254. In particular, Kosiur shows that VPNs can “utiliz[e] streaming multimedia,” which would encompass both audio and/or video transmissions. Ex. 1006 (Kosiur) at 244.

1452. Kosiur shows that VPN traffic may include “file transfers, Web browsing, and e-mail,” in which case the VPN “won't need to be concerned with QoS.” Ex. 1006 (Kosiur) at 249.

1453. Kosiur also explains that VPN traffic may also include “transactional traffic, interactive multimedia, and IP telephony,” in which case “you’ll need to track developments in QoS technologies.” Ex 1006 (Kosiur) at 249.

1454. Kosiur thus shows that a VPN may support, for example, applications that provide access to streaming multimedia, file transfers, Web browsing, and e-mail, *see* Ex. 1006 (Kosiur) at 243-254, and thus Kosiur also shows that VPNs can support a number of services or application programs (such as multimedia, file transfers, Web browsers, or e-mail), a number of communication protocols to facilitate those services or applications (such as HTTP, FTP, or TCP/IP), or multiple sessions for using those services or applications. *Id.*

1455. For the same reasons I provided for claim 12 above, a person of ordinary skill in the art would have combined the teachings of Kosiur and Provino. *See* ¶¶ 1382-1384, *above*.

1456. Provino in view of Kosiur thus shows “*The method according to claim 14, wherein the plurality of services comprises a plurality of communication protocols, a plurality of application programs, multiple sessions, or a combination thereof,*” as specified by claim 15.

**e. Claim 16**

1457. Kosiur explains that VPNs can be configured to support “newer applications, such as interactive multimedia and videoconferencing.” Ex. 1006



(Kosiur) at 254. In particular, Kosiur shows that VPNs can “utiliz[e] streaming multimedia,” which would encompass both audio and/or video transmissions. Ex. 1006 (Kosiur) at 244.

1458. Kosiur shows that VPN traffic may include “file transfers, Web browsing, and e-mail,” in which case the VPN “won't need to be concerned with QoS.” Ex 1006 (Kosiur) at 249.

1459. Kosiur also explains that VPN traffic may also include “transactional traffic, interactive multimedia, and IP telephony,” in which case “you’ll need to track developments in QoS technologies.” Ex 1006 (Kosiur) at 249.

1460. Kosiur thus shows that a VPN may support, for example, applications that provide access to streaming multimedia, file transfers, Web browsing, and e-mail, *see* Ex. 1006 (Kosiur) at 243-254, and thus Kosiur also shows that VPNs can be configured to support a number of services or application programs, such as multimedia, which would comprise “interactive multimedia and videoconferencing,” Ex. 1006(Kosiur) at 254, file transfers, Web browsers, or e-mail. Ex. 1006 (Kosiur) at 243-254.

1461. For the same reasons I provided for claim 12 above, a person of ordinary skill in the art would have combined the teachings of Kosiur and Provino. *See ¶¶ 1382-1384, above.*

1462. Provino in view of Kosiur thus shows “*The method according to claim 15, wherein the plurality of application programs comprises video conferencing, e-mail, a word processing program, telephony or a combination thereof,*” as specified by claim 16.

**f. Claim 17**

1463. Kosiur explains that VPNs can be configured to support “newer applications, such as interactive multimedia and videoconferencing.” Ex. 1006 (Kosiur) at 254. In particular, Kosiur shows that VPNs can “utiliz[e] streaming multimedia,” which would encompass both audio and/or video transmissions. Ex. 1006 (Kosiur) at 244.

1464. Kosiur shows that VPN traffic may include “file transfers, Web browsing, and e-mail,” in which case the VPN “won't need to be concerned with QoS.” Ex 1006 (Kosiur) at 249.

1465. Kosiur also explains that VPN traffic may also include “transactional traffic, interactive multimedia, and IP telephony,” in which case “you’ll need to track developments in QoS technologies.” Ex 1006 (Kosiur) at 249.

1466. Kosiur thus shows that a VPN may support, for example, applications that provide access to streaming multimedia, file transfers, Web browsing, and e-mail, *see* Ex. 1006 (Kosiur) at 243-254, and thus Kosiur also shows that VPNs can be configured to support a number of services or application programs, such as

interactive multimedia. *Id.* Interactive multimedia would comprise, for example, application programs that provides streaming audio and/or video services, such as for “video conferencing.” Ex. 1006 (Kosiur) at 254.

1467. For the same reasons I provided for claim 12 above, a person of ordinary skill in the art would have combined the teachings of Kosiur and Provino. See ¶¶ 1382-1384, *above*.

1468. Provino in view of Kosiur thus shows “*The method according to claim 15, wherein the plurality of services comprises audio, video or a combination thereof,*” as specified by claim 17.

## **VI. CONCLUSION**

1469. For the reasons set forth above, it is my opinion that the subject matter recited in claims 1-29 of the '181 patent was not novel as of the filing date of the '181 patent.

## VII. APPENDIX A: Materials Considered by Mike Fratto

| Exhibit # | Reference Name   |
|-----------|--|
| 1001      | U.S Patent No. 7,188,180   |
| 1002      | File History of U.S Patent No. 7,188,180   |
| 1003      | U.S. Patent No. 6,557,037 (“Provino”)  |
| 1004      | Takahiro Kiuchi and Shigekoto Kaihara, “C-HTTP - The Development of a Secure, Closed HTTP-based Network on the Internet,” published by IEEE in the Proceedings of SNDSS 1996 |
| 1005      | Alvaro Guillen <i>et al.</i> , 1993 International Conference on Network Protocols, <i>An Architecture for Virtual Circuit/QoS Routing</i> (Oct. 1993) (“Guillen”)            |
| 1006      | Dave Kosiur, <i>Building and Managing Virtual Private Networks</i> (1998) (“Kosiur”)   |
| 1007      | U.S. Patent No. 6,151,628 to Xu <i>et al.</i> (“Xu”)   |
| 1008      | U.S. Patent No. 6,073,175 to Tavs <i>et al.</i> (“Tavs”)   |
| 1009      | U.S. Patent No. 6,225,993 to Lindblad <i>et al.</i> (“Lindblad”)   |
| 1010      | U.S. Patent No. 8,200,837 to Bhatti <i>et al.</i> (“Bhatti”)   |
| 1011      | Declaration of Dr. Roch Guerin (“Guerin Declaration”)  |
| 1012      | [RESERVED]   |
| 1013      | Dismissal from <i>VirnetX Inc. v. Microsoft Corp.</i> , Docket No. 6:10CV94  |
| 1014      | Copy of Docket from <i>VirnetX Inc. v. Microsoft Corp.</i> , Docket No. 6:10CV94   |
| 1015      | [RESERVED]   |
| 1016      | Claim Construction Opinion and Order from <i>VirnetX, Inc. v. Microsoft Corp.</i> , Docket No. 6:07CV80  |
| 1017      | Joint Claim Construction Chart Pursuant To P.R. 4-5(d), <i>VirnetX Inc., vs. Cisco Systems, Inc.</i> , Docket No. 6:10-CV-417  |
| 1018      | Claim Construction Opinion and Order from <i>VirnetX Inc., vs. Cisco Systems, Inc.</i> , Docket No. 6:10-CV-417  |
| 1019      | E. Gavron, RFC 1535, <i>A Security Problem and Proposed Correction With Widely Deployed DNS Software</i> (Oct. 1993)   |

| <b>Exhibit #</b> | <b>Reference Name</b>  |
|------------------|--|
| 1020             | RFC 791, <i>Internet Protocol</i> (Sep. 1981)  |
| 1021             | T. Berners-Lee <i>et al.</i> , RFC 1945, <i>Hypertext Transfer Protocol -- HTTP/1.0</i> (May 1996)   |
| 1022             | Kenneth F. Alden & Edward P. Wobber, <i>The AltaVista Tunnel: Using the Internet to Extend Corporate Networks</i> , 9 Digital Technical Journal 2 (1997) |
| 1023             | Excerpts from the Prosecution History of Reexamination Control No. 95/001,270  |
| 1024             | Excerpts from the Prosecution History of Reexamination Control No. 95/001,792  |
| 1025             | U.S Patent No. 8,051,181   |
| 1026             | File History of U.S Patent No. 8,051,181   |
| 1027             | U.S Patent No. 7,987,274   |
| 1028             | File History of U.S Patent No. 7,987,274   |
| 1031             | U.S. Patent No. 6,496,867 to Beser   |
| 1032             | Kent, S., et al., RFC 2401, "Security Architecture for the Internet Protocol," November 1998   |
| 1033             | H. Schulzrinne et al., RFC 2543, "SIP: Session Initiation Protocol," March 1999  |
| 1034             | Schulzrinne, H., et al., RFC 1889, "RTP: A Transport Protocol for Real-Time Applications," January 1996  |
| 1035             | Handley, M., et al., RFC 2327, "SDP: Session Description Protocol," April 1998   |
| 1036             | Mockapetris, P., RFC 1034, "Domain Names – Concepts and Facilities," November 1987   |
| 1037             | Mockapetris, P., RFC 1035, "Domain Names – Implementation and Specification," November 1987  |
| 1038             | Srisuresh, P., et al., RFC 2663, "IP Network Address Translator (NAT) Terminology and Considerations," August 1999                                       |
| 1039             | Tittel, E., et al., Windows NT Server 4 for Dummies, Ch. 12, pp. 191-210 (1999)  |

| <b>Exhibit #</b> | <b>Reference Name</b>  |
|------------------|--|
| 1040             | Microsoft Press, "Microsoft Windows 98 Resource Kit, The Professional's Companion to Windows 98," Ch. 9, pp. 355-396, Ch. 15, and Ch. 19, pp. 849-918 (1998) |
| 1041             | Aventail AutoSOCKS v2.1 Administration and User's Guide, 1996-1997   |
| 1042             | Aventail Connect v3.1/v2.6 Administrator's Guide, 1996-1999  |
| 1043             | Ferguson, P. and Huston, G., "What Is a VPN? – Part I," The Internet Protocol Journal, Vol. 1, No. 1 (June 1998)   |
| 1044             | Ferguson, P. and Huston, G., "What Is a VPN? – Part II," The Internet Protocol Journal, Vol. 1, No. 2 (September, 1998)                                      |
| 1045             | Braden, R., RFC 1123, "Requirements for Internet Hosts – Application and Support," October 1989  |
| 1046             | Fielding, R., et al., RFC 2068, "Hypertext Transfer Protocol – HTTP/1.1," January 1997   |
| 1047             | Socolofsky, T., et al., RFC 1180, "A TCP/IP Tutorial," January 1991  |
| 1048             | U.S. Patent No. 4,405,829 to Rivest et al.   |
| 1049             | Rescorla, E., et al., RFC 2660, "The Secure HyperText Transfer Protocol," August 1999  |
| 1050             | Lloyd, B., et al., RFC 1334, "PPP Authentication Protocols," October 1992  |
| 1051             | Simpson, W., RFC 1994, "PPP Challenge Handshake Authentication Protocol (CHAP)," August 1996   |
| 1052             | Dierks, T., et al., RFC 2246, "The TLS Protocol – Version 1.0," January 1999   |
| 1053             | Simpson, W., RFC 1661, "The Point-to-Point Protocol (PPP)," July 1994  |
| 1054             | Meyer, G., RFC 1968, "The PPP Encryption Control Protocol (ECP)," June 1996  |
| 1055             | Kummert, H., RFC 2420, "The PPP Triple-DES Encryption Protocol (3DESE)," September 1998  |
| 1056             | Pall, G., RFC 2118, "Microsoft Point-To-Point Compression (MPPC) Protocol," March 1997   |
| 1057             | Gross, G., et al., RFC 2364, "PPP Over AAL5," July 1998  |
| 1058             | Townsley, W.M., et al., RFC 2661, "Layer Two Tunneling Protocol 'L2TP'," August 1999   |

| <b>Exhibit #</b> | <b>Reference Name</b>   |
|------------------|---|
| 1059             | Heinanen, J., RFC 1483, “Multiprotocol Encapsulation over ATM Adaptation Layer 5,” July 1993  |
| 1060             | Adams, C., et al., RFC 2510, “Internet X.509 Public Key Infrastructure Certificate Management Protocols,” March 1999  |
| 1061             | Bradner, S., RFC 2026, “The Internet Standards Process – Revision 3,” October 1996  |
| 1062             | Leech, M., et al., RFC 1928, “Socks Protocol Version 5,” March 1996   |
| 1063             | Malkin, G., RFC 2453, “RIP Version 2,” November 1998  |
| 1064             | Moy, J., RFC 2328, “OSPF Version 2,” April 1998   |
| 1065             | Vixie, P., et al., RFC 2136, “Dynamic Updates in the Domain Name System (DNS Update)”, April 1997   |
| 1066             | VirnetX’s Opening Claim Construction Brief in VirnetX Inc. v. Cisco Systems, Inc., et al., 6:10-CV-417 (11/4/11) (EDTX)   |
| 1067             | Defendants’ Responsive Claim Construction Brief in VirnetX Inc. v. Cisco Systems, Inc., et al., 6:10-CV-417 (12/7/11) (EDTX)  |
| 1068             | VirnetX’s Reply Claim Construction Brief in VirnetX Inc. v. Cisco Systems, Inc., et al. , 6:10-CV-417 (12/19/11) (EDTX)   |
| 1069             | [RESERVED]  |
| 1070             | Joint Claim Construction and Prehearing Statement in VirnetX Inc. v. Apple Inc., 6:12-CV-855 (2/14/14)  |
| 1071             | International Trade Commission, Investigation No. 337-TA-858, Parties’ Joint List of Disputed Claim Terms (2/25/13)   |
| 1072             | File History of <i>Inter Partes</i> Reexamination of 8,051,181, Control No. 95/000,949  |
| 1073             | U.S. Patent No. 6,182,141 to Blum   |
| 1074             | U.S. Patent No. 6,701,437 to Hoke   |
| 1075             | E. Wedlund & H. Schulzrinne, “Mobility Support Using SIP,” WOWMOM ‘99 Proceedings of the 2nd ACM international workshop on Wireless mobile multimedia                             |
| 1076             | Record of Publication of WOWMOM (Ex. 1015) on ACM Digital Library (available at <a href="http://dl.acm.org/citation.cfm?id=313281">http://dl.acm.org/citation.cfm?id=313281</a> ) |
| 1077             | ITU-T Recommendation H.323 (February 1998)  |
| 1078             | U.S. Patent No. 6,430,176 to Christie   |

| Exhibit # | Reference Name  |
|-----------|---|
| 1079      | WAP Forum, Wireless Application Protocol Architecture Specification (April 30, 1998)  |
| 1080      | W3C, “World Wide Web Consortium and Wireless Application Protocol Forum Establish Formal Liaison Relationship, ” December 8, 1999 |
| 1081      | IBM Session Initiation Protocol (SIP)   |
| 1082      | Gulbrandsen, A., et al., RFC 2052, “A DNS RR for specifying the location of services (DNS SRV),” October 1996                     |