

EX 1003

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Fortinet, Inc.,

Petitioner,

v.

Netskope, Inc.,

Patent Owner.

Case No. 2025-0041

U.S. Patent 8,397,282

DECLARATION OF JOHN BLACK, JR.

I, John Black, Jr. declare as follows:

I. INTRODUCTION

1. I have been retained by Fortinet, Inc. to provide analysis and opinions regarding the patentability of certain claims in U.S. Patent No. 8,397,282 ("the '282 Patent").

2. I am being compensated for my time at the rate of \$675 per hour. I have

no interest in the outcome of this proceeding and the payment of my fees is in no way contingent on my providing any particular opinions.

3. If requested, I am prepared to explain (in a deposition or at a trial before the Patent Trial and Appeal Board (the "Board")) the technology disclosed in the '282 Patent—including the state of the art in the relevant timeframe.

II. EDUCATIONAL AND PROFESSIONAL BACKGROUND

1. My qualifications can be found in my Curriculum Vitae (attached as Appendix A), which includes my detailed employment background, professional experience, and list of technical publications and patents.

2. I am an Associate Professor of Computer Science at the University of Colorado, Boulder. I received a B.S. in Mathematics and Computer Science from the California State University at Hayward (now "California State University, East Bay") in 1988. I received an M.S. in Computer Science in 1997, and a Ph.D. in Computer Science in 2000, both from the University of California at Davis.

3. I have taught more than 60 classes in computer science, on subjects including data structures, algorithms, networking, operating systems, software engineering, security, cryptography, discrete mathematics, and quantum computing. I have authored or coauthored more than 20 publications, primarily on issues relating to computer security. I have been involved with computers and networking for over 40 years in both commercial and academic capacities.

4. My earliest interest was in networks and security. My first memories in this regard were around 1975 when a group of friends and I learned about the telephone network and its security. A few years later, personal computers became available, and I spent most of my free time studying, programming, and modifying them. I worked extensively with various networking products throughout the 1980s, and developed an interest in cryptography soon thereafter. Although my undergraduate institution had no courses in cryptography or security in the 1980s, I decided to pursue self-study at the time, and opted to double major in Computer Science and Mathematics because cryptography is a blend of these two subject areas.

5. After earning my B.S. degree in 1988, I worked for six years at Ingres Corp as a software developer, writing and reviewing code written in C. My work primarily was directed at transaction logging, data type support, and security.

6. In 1995, I began my Ph.D. at UC Davis under cryptographer Phillip Rogaway. My area of focus was cryptography and security and my research involved encryption, authentication, hash functions, and network security.

7. After graduation I took a position as an Assistant Professor at the University of Nevada at Reno. In the Fall of 2001, I taught the networking class there, which included coverage of Ethernet, interior gateway protocols, exterior gateway protocols, ARP, DHCP/BOOTP, STP, IP, UDP, TCP, HTTP, SMTP, BGP and other protocols. In 2001, as a graduate student and I looked at the security of

ARP and invented a new protocol “AuthARP” to add security to the protocol.

8. In 2002, I moved to the University of Colorado at Boulder where I am currently employed. In the Fall of 2002, I co-designed and co-taught a new course called “Foundations of Computer and Network Security,” which included descriptions of security issues around both wired and wireless security challenges, mostly for public-facing network services including the world-wide web. I have taught this class seven more times since then, including modern topics such as wireless networking, the Internet of Things, and so forth.

9. In my career at the University of Colorado, I have published several more papers in the area of cryptography and network security. I have taught more than 30 courses in network security and cryptography, and have graduated several PhD students in these areas.

10. I have also worked for a consulting company at times, writing software on contract basis. Although most projects are covered by NDAs, many involved networks and/or aspects of computer security.

11. In 2011, I began technical consulting for a local company called Cardinal Peak, which focuses primarily on video encoding and delivery systems. My work for Cardinal Peak has largely been directed to video encoding, transcoding, compression, encryption, and DRM, but also has included networking projects and embedded device work. For example, I designed the security system for the Pro1

smart thermostat, implemented the DRM for the Yonder Music App, worked on 802.1X code for smart dog collars, and helped design the cryptography used in Fitbit devices for wireless transfer of a Fitbit watch to a phone or laptop. My work has been adopted into various commercial protocols, including enterprise Wi-Fi and EMV tap-to-pay technology.

12. As part of my duties in the Computer Science department at the University of Colorado I served as chair of the Computing Committee. In 2014, I oversaw the construction of a new datacenter, including cooling, power conditioning, cabling, networking, and security for the research units within the department.

13. In 2016, I took a two-and-a-half year leave of absence from the University of Colorado to start a company named “SecureSet” in Denver, Colorado. The objective of SecureSet is to take reasonably proficient technical people and turn them into computer and network security specialists via five months of intensive training. SecureSet was sold to WeWork in 2019 and continues to offer computer security classes today. In 2018, I returned to my position at the University where I remain employed to the present day.

III. MATERIALS REVIEWED

In preparing this declaration, I reviewed and considered the following materials, along with any other materials referenced in the body of my declaration:

Exhibit	Description
1001	U.S. Patent No. 8,397,282 ("the '282 patent")
1002	File History of the '282 patent
1004	U.S. Patent No. 6,154,775 ("Coss")
1005	U.S. Patent Publication No. 2003/0041266 ("Ke")

14. I may use these documents and information, or other information obtained during the course of this proceeding, as well as representative charts, graphs, schematics and diagrams, animations, and models based on those documents and information, to support and to explain my testimony.

15. My opinions are based in part on a review and analysis of the above-mentioned documents and materials. I have also drawn on my education, experience, and knowledge of basic engineering and design principles in forming my opinions.

IV. RELEVANT ART AND LEVEL OF ORDINARY SKILL

16. I have been informed that the level of ordinary skill in the relevant art at the time of the invention is relevant to inquiries such as the meaning of claim terms, the meaning of disclosures found in the prior art, and the reasons one of ordinary skill in the art may have for combining references.

17. I have been informed that factors that may be considered in determining the level of ordinary skill include: (1) the type of problems encountered in the art; (2) prior art solutions to those problems; (3) rapidity with which innovations are made; (4) sophistication of the technology; and (5) education level of active workers in the relevant field. I have been further informed that a POSITA is also a person of

ordinary creativity.

18. In my opinion, A POSITA in the field of the '282 Patent would have had either (1) at least a bachelor's degree in computer science or computer engineering or an equivalent field plus at least one year of experience working on computer networking, or (2) at least 3 years of experience working on computer networking, even without a formal degree.

19. At the time of the alleged invention, I was at least a POSITA in the art relevant to the '282 Patent. Further, based on my experience, I understand and know of the capabilities of a POSITA in 2004—and I am familiar with how a POSITA would have understood and used the terminology found in the '282 Patent and the Challenged Claims at the time of filing, and with the state of the art at that time.

20. Even though my own experience and qualifications exceed that of a POSITA, the statements herein regarding the knowledge of a POSITA and what would have been obvious to a POSITA are from the perspective of POSITA at the relevant priority date.

V. LEGAL STANDARDS APPLIED

21. I am not an attorney, but I have been informed of several legal principles, which I have employed in forming my opinions in this declaration. I have been informed that each patent claim is considered separately for purposes of patentability and that a dependent claim that depends from another claim includes

all of the limitations of the claim from which it depends.

A. Burden of Proof

22. I have been informed that the petitioner in an *inter partes* review bears the burden of establishing unpatentability by "a preponderance of the evidence." I have been further informed and understand that, to prove an assertion by "a preponderance of the evidence," the party with the burden of proof must demonstrate that it is more likely than not that the assertion is true.

B. Claim Construction

23. I have been informed that the claims of the '282 Patent in this proceeding are to be construed using the same claim construction standard applied in district court. I have been informed the claim construction analysis begins with the ordinary meaning of a claim term, and there is a presumption that the term carries its plain and ordinary meaning to a person of ordinary skill in the art ("POSITA") as of the effective filing date of the patent. I have also been informed that the most important sources for determining the meaning of a claim term are the claims, the specification, and the prosecution history of the patent at issue, which collectively is "intrinsic evidence." I have also been informed that a patentee can act as their own lexicographer and provide specific definitions of claim terms that may differ from the plain and ordinary meaning.

1. "wherein the set of firewall rules is dynamically self-configurable during runtime"

24. A POSITA would have understood this term to mean "a set of firewall rules that are configured without any human operator interaction while the node is evaluating whether to accept or deny the packet." This is based on the plain meaning of "during runtime"—*i.e.*, when the firewall is in operation it is evaluating packets for passing/dropping—and in view of the prosecution history which distinguishes the prior art based on the fact that the prior art required "an advanced user to establish a firewall policy for a computer." EX1002, p. 617.

2. "chains of rules forming various paths through a hierarchical structure"

25. A POSITA would have understood this term to mean "a list of one or more linear and serialized sequence of firewall rules forming various paths through a hierarchical structure." This is consistent with the '282 patent's definition of "chains of rules." The specification explains them to be "serialized sequences of one or more rules" (EX1001, 5:5-7) and "Chain—Represents a list of one or more firewall rules. A firewall rule has the normal predicate/antecedent format, where elements of the predicate must match before the actions of the antecedent are enacted. Chains of firewall rules are linear and serial, as the name implies." EX1001, 5:24-28. Similarly, the specification explains "rule chains" "in a hierarchical name space" that follow paths using predicate/antecedent rule logic that is "supported by

the syntax of a software programming language." EX1001, 6:36-59.

3. "defined places for dynamically updating the set of firewall rules during runtime"

26. A POSITA would have understood this term to mean "one or more isolated locations within the sequence of firewall rules to add, remove, or change a rule while the node is evaluating whether to accept or deny the packet." This is based, again, on the plain meaning of "during runtime"—*i.e.*, when the firewall is in operation it is evaluating packets for passing/dropping—and the specification's explanation that the dynamic rule chains "offer isolated, well-defined places for specific behavior to be introduced." EX1001, 5:5-12. In addition, the specification explains that the rules could be added either "as part of firewalls configuration *or* runtime reconfiguration." EX1001, 7:62-8:4 (emphasis added). That the patentee claimed the "runtime" option confirms that this term is understood to refer to rules added while the firewall is in operation—*i.e.*, while it is processing packets—and not simply before it is put into service.

For all other terms in the '282 Patent, I have performed my analysis using their plain and ordinary meaning to a POSITA.

C. Anticipation

27. I have been informed that a patent claim is unpatentable if it is "anticipated" by prior art. I have been informed that a reference anticipates a claim if the reference discloses each limitation of the claim at issue, either expressly or

inherently. I have been informed that a prior art reference may anticipate a claim without disclosing a feature of the claimed invention if that missing characteristic is necessarily present, or inherent, in the single anticipating reference as would have been understood by a POSITA.

D. Obviousness

28. I have been informed that a patent claim is unpatentable if it would have been obvious to a POSITA at the time of the claimed invention. I have been further informed that a claimed invention is not patentable if differences between it and the prior art are such that the subject matter as a whole would have been obvious to a POSITA at the time of invention. I have been informed that factors relevant to the determination of obviousness include the scope and content of the prior art, the level of ordinary skill in the art at the time of the invention, differences between the claimed invention and the prior art, and "secondary considerations" or objective indicia of nonobviousness.

29. I have been informed that a single reference alone can render a patent claim obvious if any differences between the reference and the claim would have been known or obvious to a POSITA at the time of the invention. That is, if the POSITA could have adapted the reference to meet the claims of the patent by applying known concepts to achieve expected results, then the patent claims are rendered obvious by that reference.

30. I have been informed that the combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results. I have been further informed and understand that when a patent claim simply arranges old elements with each performing the same function it had been known to perform and yields no more a POSITA would expect from such an arrangement, the combination is obvious.

31. I have been informed that a patent claim composed of several limitations is not obvious merely because each limitation was independently known in the prior art. Hindsight reasoning is not an appropriate basis for combining references to form an obviousness combination. I have been further informed that it can be important to identify a reason that would have prompted a POSITA to combine multiple prior art references, or multiple teachings of the same prior art reference.

32. I have been informed that various "secondary considerations" (sometimes referred to as objective indicia of nonobviousness) may support a determination of nonobviousness and that such secondary considerations must be considered as part of an obviousness analysis. I have been informed that even strong evidence of secondary considerations may not be sufficient to rebut a strong showing of obviousness. I have been informed that the patentee bears the burden of showing the existence and effect of secondary considerations of nonobviousness, after which

the burden of coming forward with rebuttal evidence shifts to the party arguing obviousness. I have been informed that the following secondary considerations may indicate nonobviousness:

- a. Evidence that others, including the accused infringer, copied the patented invention.
- b. Evidence of a persistent problem or need in the art that was resolved by the patented invention.
- c. Evidence that others have tried and failed to solve the problem or failed to provide the need resolved by the claimed invention.
- d. The willingness of industry to license the patent at issue.
- e. Evidence that those of ordinary skill in the art were skeptical as to the merits of the invention.
- f. Evidence of praise directed to the invention by others in the field.
- g. Evidence that those of ordinary skill in the art were surprised by the capabilities of the claimed invention.

33. I have also been informed that the near-simultaneous invention by others can be evidence that a claimed invention is obvious. I have further been informed that the patentee bears the burden of showing a "nexus" between the claimed invention and the evidence proffered on secondary considerations.

VI. TECHNOLOGY BACKGROUND

34. The '282 Patent deals with basic concepts in computer networks and firewalls that were well-known and well-understood as of 2004.

Networks

35. A computer “network” is a set of two or more interconnected computers, where the connections can be formed using a variety of technologies including both wired connections (e.g., modem, serial, ethernet, fiber optic) and wireless connections (e.g., cellular, WiFi, satellite, etc.). In order for a computer to connect to a network, the computer must have a network “interface” which facilitates the intercommunication between the computer’s operating system and the network. A computer can have multiple interfaces and therefore be a member of multiple networks; for example, a smartphone might be connected to both the Internet over its cellular interface and the local network via its WiFi interface.

Routers

36. A special type of network device known as a “router” acts as a waypoint between at least two different networks in order to allow traffic from one network to be passed through the router and into another network. A router therefore has at least two network interfaces, one interface for each connected network that the router can process traffic to or from. Routers have a “routing table” that informs how network traffic should be routed through the device to the appropriate destination interface.

For example, an entry in a routing table might look like this:

```
default via 192.168.1.1 dev eth0 proto dhcp metric 100
192.168.1.0/24 dev eth0 proto kernel scope link src 192.168.1.42
metric 100
```

37. This essentially says that any arriving traffic destined for an IP address beginning with “192.168.1” should be sent to the interface called “eth0” (2nd line of table above), and all other traffic not matching that destination IP address should be sent to the gateway router at IP address 192.168.1.1 (1st line of table above). The “gateway” is a router that sits between a given network and the outside world, typically on the edge of a home or enterprise network and the Internet.

Firewalls

38. A “firewall” is a type of network security device that is typically situated at or near the edge of a protected network to filter traffic entering from the outside world, and may also filter traffic from within the protected network destined for the outside world. The first firewalls, called “traditional” or “basic” firewalls, emerged by 1990 and one of the most well-known books describing basic firewalls was published in 1994.¹

39. A basic firewall is a device, either a special-purpose appliance or a software program running on a general purpose computer, that can permit traffic to

¹ Cheswick and Bellovin, "Firewalls and Internet Security: Repelling the Wily Hacker", First Edition, ©1994, available at <https://www.wilyhacker.com/1e/>.

pass through it, or deny passage through it, according to rules within the firewall. By the year 2003, one of the most widely-used and well-known firewall was the software called “iptables” supplied with the Linux operating system.² Although iptables has a rather arcane and difficult to master syntax, security specialists and network administrators became experts in its use if they were responsible for overseeing Linux-based computer systems. An example of an iptables command is listed here:

```
iptables -A INPUT -i eth0 -p tcp -s 1.2.3.4 -j DROP
```

40. This command adds an entry to the iptables firewall ruleset saying that any traffic from IP address 1.2.3.4 arriving on network interface “eth0” and using the TCP protocol should be dropped (i.e., it should not be permitted through the firewall). This rule was added to the “INPUT” chain of the firewall ruleset. This effectively blocks all incoming traffic from the IP address 1.2.3.4, which may be done if, for instance, it were determined that this IP address was being operated by a malicious entity who was exhibiting threatening behavior.

41. The “iptables” software featured a number of different rulesets called “chains” and could track the “state” of a session for more advanced filtering features.

² Indeed, the '282 patent itself admits that "IPTables" is an "existing operating system mechanism[]" which may be used to implement the claimed concepts. EX1001, 4:66-5:4. For an example of how IPTables was used and understood in that timeframe, see, for example, "IPTables" Section (8), updated March 9, 2002, <https://people.netfilter.org/kadlec/ipset/iptables.man.html>.

Further commands could be issued to add further rules, remove a given rule, or modify/replace rules in the ruleset, meaning dynamic changes to the iptables firewall table were possible and indeed some software at that time exploited this ability to modify rulesets.

VII. THE '282 PATENT AND PROSECUTION HISTORY

A. The '282 Patent

42. The '282 patent relates generally to a *dynamically configurable firewall*. EX1001, Abstract. More specifically, it describes “network firewalls that can dynamically adapt to changing conditions and operator requirements.” EX1001, 1:35–38. The patent asserts that this approach “provides a new level of flexibility including, but not limited to, dynamically adding new network interface abstractions or groupings of interface abstractions and tailoring the behavior of those abstractions to the network client devices’ specific needs.” *Id.*, 2:46–50. However, as discussed below, the patent’s disclosures largely recite well-understood firewall concepts and conventional rule-based logic that were long known in the art.

43. The '282 patent refers to the sources and destinations of network traffic as “nodes.” *Id.*, 2:56–64. Each “node...is simultaneously a source of and destination for network packets [and p]ackets travel between nodes over intra-firewall connections within the firewall model.” *Id.*, 4:35–39. The firewall determines whether to drop or allow packets according to rule sets associated with the incoming and outgoing nodes. *See, e.g.*, EX1001, 8:5–25, 9:27–34. The specification further

notes that “rules are applied to packets following on the connection between” the nodes. *Id.*, 9:38–39. Thus, the patent merely describes the basic operation of a firewall—evaluating packets as they traverse from an input interface to an output interface—framed as “the experience of a network packet as it travels through the firewall from its arriving node to its departing node.” *Id.*, 6:36–42.

44. Within the firewall, the rules are organized as “dynamic chains of rules,” defined as “serialized sequences of one or more rules.” *Id.*, 5:5–7. These “chains of firewall rules are linear and serial, as the name implies.” *Id.*, 5:24–28. Although the patent briefly mentions representing rule chains as leaves in a tree, it quickly concedes that the underlying logic is simple: “Rule chains essentially represent predicate/antecedent rule logic and can be classified as classic production rules systems as known in the Artificial Intelligence community.... Said differently, rule chains represent classic ‘If–Then’ logic, as might be supported by the syntax of a software programming language.” *Id.*, 6:48–59. In other words, the “dynamic chains” are nothing more than standard conditional rules—conventional “if–then” logic long used in programming and firewall implementations. Indeed, the specification admits that the “representative set of firewall rules” and the disclosed “firewall functions are relatively standard and basic capabilities that can be found in most firewall implementations.” *Id.*, 6:60–65.

45. Finally, the patent suggests that, beyond static inbound and outbound

rules, additional rule chains may be “inserted and deleted dynamically.” *Id.*, 7:62–64. These so-called dynamic rule chains are reached via “taps” in the main rule set, which redirect the firewall to another chain “dynamically loaded as part of a firewall’s configuration or runtime reconfiguration.” *Id.*, 7:65–8:4. But while the patent repeatedly invokes the notion of “dynamic” configuration, it provides no technical explanation of how such dynamic rules are generated, when or under what conditions they are introduced, or by what mechanism they are incorporated into or removed from the firewall’s configuration file. *Id.* The ’282 patent thus offers little more than a high-level restatement of known firewall behavior and elementary rule-based processing, without disclosing any specific algorithm, architecture, or mechanism to achieve the claimed “dynamic” functionality.

B. The '282 Prosecution History

46. U.S. Patent Application No. 13/092,488, which ultimately issued as the ’282 patent, was filed on April 22, 2011. During prosecution, the Examiner repeatedly rejected the claims under §103, finding that the alleged core elements of the invention were already disclosed in the prior art of record. As the Examiner explained, the prior art taught “a method and device for controlling data through a firewall... comprising: defining at least one node... associated with two or more network interfaces; associating a set of firewall rules with the at least one node... and accepting or denying [a] packet based on the set of firewall rules,” as well as

“reconfiguring the firewall [] at runtime.” EX1002, p. 547.

47. In response to these sustained rejections, the Applicant amended the claims to add language requiring that the firewall be “dynamically self-configurable... during runtime without operator interaction.” *Id.*, p. 609. The Examiner, however, again found this purportedly “self-configurable” feature to be obvious in view of the cited references. *Id.*, pp. 633–635. Only after further amendment—adding that “the set of firewall rules comprises a plurality of chains of rules forming various paths through a hierarchical structure, and wherein the hierarchical structure comprises defined places for dynamically updating the set of firewall rules during runtime”—did the Examiner allow the claims. *Id.*, pp. 663.

VIII. PRIOR ART

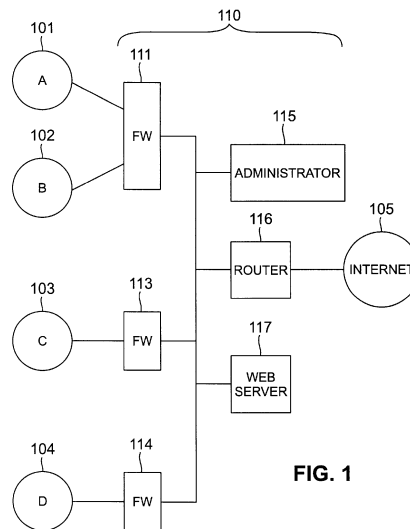
A. U.S. Patent No. 6,154,775 to Coss (EX1004, "Coss")

48. Coss, U.S. Patent No. 6,154,775, was filed on September 12, 1997, and issued on November 28, 2000. I understand Coss is prior art to the '282 patent.

49. **Coss** describes a configurable firewall that supports dynamically adaptable rule chains, similar to the system disclosed in the '282 patent. Specifically, Coss presents an "improved computer network firewall" designed to "support multiple security policies" and to do so "by applying any one of several distinct sets of access rules." Coss, Abstract. "At the firewall, packets are inspected and filtered, i.e., passed on or dropped depending on whether they conform to a set of predefined

access rules." *Id.*, 1:14-26. In addition to static, preloaded rules, Coss also supports the use of "[d]ynamic rules" that can be added at runtime, as explained more fully below. *Id.*, Abstract.

50. Like the multi-node architecture of the '282 patent, **Coss's** firewall supports "multiple security domains...each with a separate security policy." *Coss*, 3:35-40. For example, as shown in Figure 1, **Coss** illustrates "four user sites...of corporations A through D, with firewall protection in their connections to the Internet." *Id.*, 3:43-57. The "firewall facility" consists of one or more "firewall processors," each configurable to protect one or more domains (or corporate "sites"). *Id.*



Coss, FIG. 1.

51. In Coss, firewall processors can be deployed in different configurations depending on the security requirements of the sites they protect. For example,

processors 113 and 114 are each dedicated to a single site (C and D, respectively) while processor 111 simultaneously serves two separate sites (101 and 102), enforcing distinct firewall policies for each site's Internet traffic and also for communications between those two sites. As Coss discloses, " FIG. 1 shows four user sites 101-104, e.g., of corporations A through D, with firewall protection in their connections to the Internet 105. Such protection is provided by a firewall facility, here in the form of a LAN 110, including firewall processors 111, 113 and 114, an administrator processor 115, a router 116 and a web server 117. Each of firewall processors 113 and 114 is dedicated to a single site, namely respective sites 103 and 104. Firewall processor 111 is configured to serve the two sites 101 and 102. Firewall processor 111 implements separate firewall policies for each of the two sites vis-a-vis the Internet 105, as well as for communications between the two sites. A process for preferred operation of the firewall processor 111 is described below with reference to FIGS. 5A and 5B, including properly selecting among different firewall policies." Coss, 3:43-57.

52. Coss further describes partitioning a site into multiple "sub-sites," each with its own security policy enforced by the firewall. Coss, 3:58-4:3 ("FIG. 2 shows a user site 201 connected to the Internet 105 via a firewall processor 211. An administrator processor 215 and a router 216 are connected to the firewall processor 211. The router 216 is connected to additional firewall processors 212 and 213 which

are internal to the user site 201. The firewall processor 212 protects a single sub-site 223, such as Human Resources (HR). The firewall processor 213 is configured for protecting two sub-sites, such as Payroll (P) and Disbursements (D), vis-a-vis the remainder of the site 201 as well as with respect to communications between sub-sites 221 and 222. This can be achieved by employing the process illustrated by FIGS. 5A and 5B in the firewall processor 213."), FIG. 2.

53. In addition, Coss introduces the concept of host groups, which allow administrators to add or remove hosts from a group without needing to reconfigure other portions of the rule set. *Id.*, 2:41–46 ("Other types of dynamic rules include rules which define a host group, such that the host group can be modified to add or drop different hosts without altering other aspects of the access rule set.").

54. Coss also teaches rule processing using hierarchical chains of rules. For example, Coss's firewall processes information on a packet-by-packet basis. Coss, 1:65–2:6 ("In accordance with a first aspect of the invention, a computer network firewall is able to support (a) multiple security policies, (b) multiple users, or (c) multiple security policies as well as multiple users, ***by applying any one of several distinct sets of access rules for a given packet. The particular rule set that is applied for any packet*** can be determined based on information such as the incoming and outgoing network interfaces as well as the network source and destination addresses.").

55. Access rules are organized into distinct tabular sets, with each entry defining conditions and the corresponding action for packets that meet those conditions. Coss, 4:4–11 ("*The security policies can be represented by sets of access rules which are represented in tabular form* and which are loaded into the firewall by a firewall administrator. As illustrated in FIG. 3, such a table can provide for categories including rule number, designations of source and destination hosts, a designation of a special service which can be called for in a packet, and a specification of an action to be taken on a packet."). Rules are evaluated in sequence until a match is found; if no match occurs, the packet is dropped. Coss, 4:26–30. ("In rule processing for a packet, *the rules are applied sequentially* until a rule is found which is satisfied by the packet (or until the rule table is exhausted, in which case the packet is dropped)."). This structure parallels the predicate–action or “if–then” format described in the ’282 patent: a packet satisfies a rule only if every condition within that rule is met. Coss, 4:30–31 ("For a packet to satisfy a rule, each condition included in the rule must be met."). Figure 3 illustrates a representative rule table as implemented in Coss.

RULE NO.	SOURCE HOST	DEST. HOST	SERVICE	ACTION
10	A	B	FTP	PASS
20	A	*	*	DROP
30	B	C	TELNET	PROXY
40	*	D	MAIL	PASS

FIG. 3

56. Coss further specifies that the rules are organized into "categories" such as "Source Host," "Destination Host," and "Service," for example, and that these rules represents criteria that must be satisfied by a packet before the firewall allows it to pass. Coss, 4:13-16 ("In FIG. 3, the categories "Source Host," "Destination Host" and "Service" impose conditions which must be satisfied by data included in a packet for the specified action to be taken on that packet.") The specification also makes clear that these categories are processed hierarchically. For example, Figures 5A–5B illustrate that the rule engine evaluates the source-domain rules first, then proceeds to destination-domain rules, and so on. See FIGs. 5A–5B; 6:18–7:9 (step 504: searching the source domain rule set; step 507: destination domain look-up).

57. Finally, Coss describes the use of dynamic rules alongside static rules. Specifically, Coss teaches using "dynamic rules...which are included with the access rules as a need arises, for processing along with the access rules, e.g., by a rule

processing engine." Coss, 8:28-30. As Coss explains, Dynamic rules "can be loaded at any time by trusted parties" which includes "a trusted application, remote proxy, or firewall administrator." *Id.*, 8:30-35. A POSITA would have understood that insofar as Coss separately lists "trusted application" and "firewall administrator" confirms that the trusted application loads a dynamic rule without the assistance or intervention of a firewall administrator—otherwise, Coss would have just listed the administrator as loading the rules. Likewise, the fact that rules can be added "at any time," including "based on events happening in the network," confirms that Coss's rules may be added at runtime. Coss, 8:35-40. Dynamic rules in Coss also can be removed from the rule set once it has served its function and, generally, "dynamic rules allow a given rule set to be modified based on events happening in the network without requiring that the entire rule set be reloaded." *Id.*, 8:35-40.

B. U.S. Patent Publication No. 2003/0041266 to Ke (EX1005, "Ke")

58. Similar to the '282 patent, Ke describes a configurable firewall that secures multiple virtual local area networks (VLANs) using a packet-based firewall model. Ke, [0005]. Ke explains that its system includes a firewall engine that receives firewall policies and applies those policies to individual packets: "The data processing system includes *a firewall engine that can receive a set of firewall policies and apply the firewall policies to a data packet...* one or more virtual private networks that each have an associated destination address and policies and a

controller that can detect an incoming data packet, *examine the incoming data packet for a virtual private network destination address and identify the policies associated with the virtual private network destination*. If the policies include firewall policies, then the controller can call the firewall engine and apply the set of firewall policies corresponding to the virtual private network destination to the data packet....” Ke, [0005]; *see also* [0032]–[0035].

59. Ke further discloses a network security system that partitions firewall services and other security resources into multiple configurable domains. Ke, [0016] (“security system resources including firewall services and *a controller that can partition the security system resources into a plurality of separate security domains. Each security domain can be configurable to enforce one or more policies relating to a specific subsystem*, and to allocate security system resources to the one or more security domains”). These domains support Ke’s “multi-customer, multi-domain architecture,” where each domain acts as an independent system with its own policy set. Ke, [0031] (“provides a multi-customer, multi-domain architecture...to create and manage separate security domains, each domain acting as a stand alone system and having its own set of policies”).

60. Ke implements this architecture using “Virtual Systems,” or VLANs. Ke, [0031]–[0033]. On the secure side of the firewall, VLAN traffic is carried over a trunk line and separated by switches to reach each customer’s servers. As Ke

explains, a VLAN “is a Layer 2 multiplexing technique that allows several streams of data to share the same physical medium, such as a trunk cable, while enjoying total segregation.” Ke, [0033].

61. Thus, Ke’s Virtual Systems operate in the same way Coss’s firewall processors are used to manage multiple domains for different enterprises. Indeed, Ke’s “multi-customer, multi-domain architecture” (Ke, [0031]) is described similarly (as protecting customer sites) to Coss’s “multiple security domains...four user sites...of corporations A through D.” Coss, 3:35–56. While Coss establishes that domains can be defined and managed independently (and even grouped dynamically), Ke provides the explicit configuration mechanisms for implementing those domains in practice.

62. Ke also describes how such domains are configured in real time or at system startup (Ke, [0058], “configuration of the Internet security system in real time or at start up with a saved configuration script”). For example, the system may be configured so that “VLAN1 = Customer A” and “VLAN2 = Customer B,” with each VLAN treated as its own virtual system enforcing corresponding firewall policies for packet traffic. *Id.*, [0059]-[0082] (describing the configuration of virtual systems), [0115]-[0122] (describing "defin[ing] virtual LANs and setting "private polic[ies]" for a particular customer). Ke even discloses creating a named virtual system such as “marketing” and adding interfaces to it (*id.*, [0069], “a new virtual

system named 'marketing' and configur[ing] that system...adding two virtual interfaces for the 'marketing' system"). These virtual interfaces provide the ingress and egress points for VLAN traffic, enabling multiple isolated data streams to share the same physical trunk while remaining logically segregated (*id.*, [0033]).

IX. SUMMARY OF OPINIONS

63. I have been asked to compare the references cited herein to claims 1-35 of the '282 patent (the "Challenged Claims").

a. Grounds 1 and 2: In my opinion, claims 1-35 of the '282 patent are anticipated by, and at minimum obvious over, Coss.

b. Ground 3: In my opinion, claims 1-35 of the '282 patent are obvious over Coss in view of Ke.

X. GROUNDS 1-2 – CLAIMS 1-35 ARE ANTICIPATED BY OR OBVIOUS OVER COSS

A. Claim 1

1. Claim 1[pre]: "A method for controlling data through a firewall performed on at least one data controlling computer having computer instructions stored on at least one non-transitory computer readable medium, comprising:"

64. In my opinion, Coss discloses this limitation. Coss discloses a method for controlling data through a firewall (e.g., "information...transmitted in the form of packets" is "filter[ed] at a...firewall") performed on at least one data controlling computer (e.g., "firewall for controlling the flow of data") having computer instructions stored on at least one non-transitory computer readable medium (e.g.,

"computer system software...on general-purpose PC hardware").

65. For example, **Coss** discloses a multi-domain computer network firewall implemented on a general-purpose computer. Coss, Abstract ("The invention provides *improved computer network firewalls* which include one or more features for increased processing efficiency. A firewall in accordance with the invention *can support multiple security policies, multiple users or both, by applying any one of several distinct sets of access rules...*"), 1:9-11 ("This invention relates to the prevention of unauthorized access in computer networks and, more particularly, to *firewall protection within computer networks.*"), Coss, 3:25-35 ("The preferred techniques can be implemented at a firewall for controlling the flow of data between, for example, separate local area networks (LANs) or subnets of a LAN. Exemplary embodiments of the invention are described herein in terms of processes. Efficient prototypes of such processes have been implemented as computer system software, using the "C" programming language for implementation on general-purpose PC hardware. Efficiency can be enhanced further, as is known, by special-purpose firmware or hardware computer system implementations."). Coss's firewall applies "a set of predefined access rules" to packets moving through the firewall and passes or drops (*i.e.*, accepts or denies) the packets accordingly. Coss, 1:14-26.

2. Claim 1[a]: "defining at least one node, wherein the at least one node is associated with two or more network interfaces;"

66. In my opinion, Coss discloses, and at minimum renders obvious this limitation. Coss discloses, and at minimum renders obvious, defining at least one node (e.g., "domain", "[sub-]site", "LAN", "subnets"), wherein the at least one node is associated with two or more network interfaces (e.g., "each domain is associated with one or more network interfaces", "[incoming/outgoing] network interface").

67. As **Coss** explains, the firewall "support[s] multiple security domains...each with a separate security policy" and "in the firewall, each domain is associated with one or more network interfaces." Coss, 3:36-4:3 ("With a capability for supporting multiple security domains, a single firewall can support multiple users, each with a separate security policy. Also, as different security policies can apply for communications between sub-sites, such a capability can be used within a site. Respective configurations are illustrated by FIGS. 1 and 2. ¶ FIG. 1 shows four user sites 101-104, e.g., of corporations A through D, with firewall protection in their connections to the Internet 105.... ¶ FIG. 2 shows a user site 201 connected to the Internet 105 via a firewall processor 211."), 6:18-28 ("FIGS. 5A and 5B illustrate over-all flow for packet processing by a firewall which supports multiple domains. Such processing includes determining the domains which the packet is to cross, examining the applicable rules to ascertain whether the packet may pass, and determining whether any special processing is required. In the firewall, each domain

is associated with one or more network interfaces. Interfaces that support more than one domain are separated using an IP address range to distinguish the packets.").

68. Figure 1 of Coss "shows four user sites 101-104, e.g., of corporations A through D, with firewall protection in their connections to the Internet 105. Such protection is provided by a firewall facility, here in the form of a LAN 110, including firewall processors 111, 113 and 114, an administrator processor 115, a router 116 and a web server 117...." Coss, 3:36-4:3. Coss also explains that its firewall protects "sub-sites" with each sub-site capable of having its own security policy. *Id.* As Coss teaches, each "firewall processor" can serve one or more domains—e.g., "firewall processor 111 is configured to serve the two sites 101 and 102." *Id.* A POSITA would have understood Coss's teaching of a firewall designed to "support multiple security domains" with "separate security polic[ies]," as disclosing, or at least rendering obvious, the step of defining the domains—e.g., the firewall entry and departure networks—that the firewall serves. This is so because protecting each such site (or sub-site) would have required explicit definition steps of the domains being protected (such as loading domain configurations) to associate the site with interfaces and policies as a system typically cannot protect an undefined entity. This is further confirmed by Coss itself and the Figure 6 "domain table" which defines Domain A, Domain B, etc., to associate the respective IP range to that Domain.

INTERFACE	ADDRESS RANGE	DOMAIN
0	10.50.0.0 - 10.50.255.255	A
0	10.60.0.0 - 10.60.255.255	B
1	*	C
2	*	*

FIG. 6

Coss, FIG. 6; *see also id.*, 7:10-15 ("For convenient linking of each network interface to a domain, a domain table is used. In cases where an interface is shared by multiple domains, an address range is included. This is illustrated by FIG. 6 which shows non-overlapping address ranges."); *see also id.*, 1:61-2:6, 3:25-35, 6:33-37.

3. Claim 1[b]: "associating a set of firewall rules with the at least one node;"

69. In my opinion, Coss discloses, and at minimum renders obvious this limitation. Coss discloses, and at minimum renders obvious, associating a set of firewall rules ("distinct set of access rules";) with the at least one node ("domain").

70. Coss discloses associating a set of firewall rules with a node by teaching the use of "distinct sets of access rules" applied to a packet based on the applicable domains. *E.g.*, Coss, 1:61-2:6 ("[C]omputer network firewall is able to support (a) multiple security policies, (b) multiple users, or (c) multiple security policies as well

as multiple users, by applying any one of several distinct sets of access rules for a given packet. The particular rule set that is applied for any packet can be determined based on information such as the incoming and outgoing network interfaces as well as the network source and destination addresses."). As an example, the firewall must identify the applicable "rule set" to apply to a packet based on the source and/or destination domains of the packet, which confirms that rule set as associated with either of those domains. *Id.*

71. Coss explains that the rules or "security policies" are stored in a rule table and include multiple "categories" including "Source Host" and "Destination Host," for example, and may specify particular actions to take on a packet. Coss, 4:4-19 ("The security policies can be represented by sets of access rules which are represented in tabular form and which are loaded into the firewall by a firewall administrator. As illustrated in FIG. 3, such a table can provide for categories including rule number, designations of source and destination hosts, a designation of a special service which can be called for in a packet, and a specification of an action to be taken on a packet. Special services can include proxy services, network address translation, and encryption, for example. In FIG. 3, the categories "Source Host," "Destination Host" and "Service" impose conditions which must be satisfied by data included in a packet for the specified action to be taken on that packet.").

RULE NO.	SOURCE HOST	DEST. HOST	SERVICE	ACTION
10	A	B	FTP	PASS
20	A	*	*	DROP
30	B	C	TELNET	PROXY
40	*	D	MAIL	PASS

FIG. 3

Coss, FIG. 3. "In FIG. 3, the categories 'Source Host,' 'Destination Host' and 'Service' impose conditions which must be satisfied by data included in a packet for the specified action to be taken on that packet." Coss, 4:4-19.

72. Coss also teaches the "over-all flow for packet processing by a firewall that supports multiple domains. Such processing includes determining the domains which the packet is to cross, examining the applicable rules to ascertain whether the packet may pass, and determining whether any special processing is required." Coss, 6:18-28; *see also id.* 6:29-65 (discussing the flowchart of Figures 5A and 5B, which includes step 503 ("the source domain is determined"), step 504 ("the rule set for the source domain is searched for a match" and applied to the packet), step 506 ("the destination domain is determined"), and step 507 ("rule set look-up for the destination domain are carried out in a manner analogous to that employed for the source domain"). Based on these disclosures, a POSITA would have understood and

at least have found obvious that Coss's firewall necessarily requires associating each of the distinct set of access rules to the appropriate domains in order to "look-up" or "search[] for" a rule set based on the domain.

4. Claim 1[c]: "receiving a packet at a first node of the at least one node; and"

73. In my opinion, Coss discloses this limitation. **Coss discloses receiving a packet at a first node of the at least one node.**

74. For example, Coss expressly teaches that the "over-all flow for packet processing by a firewall that supports multiple domains" includes, for example, step "501: an IP packet is received by the firewall at an interface." Coss, 6:29-30. Further, as explained above in the prior limitation, the firewall implements and applies a particular rule set based on the domains that the packet crosses, confirming for a POSITA that the firewall receives the packet at the domain. Coss, 6:18-28 ("FIGS. 5A and 5B illustrate over-all flow for packet processing by a firewall which supports multiple domains. Such processing includes determining the domains which the packet is to cross, examining the applicable rules to ascertain whether the packet may pass, and determining whether any special processing is required. In the firewall, each domain is associated with one or more network interfaces. Interfaces that support more than one domain are separated using an IP address range to distinguish the packets."). See also, Coss at 7:64-65, 9:33, 9:52-53.

5. Claim 1[d]: "accepting or denying the packet based on the set of firewall rules, wherein the set of firewall rules is dynamically self-configurable during runtime without operator interaction, wherein the set of firewall rules comprises a plurality of chains of rules forming various paths through a hierarchical structure, and wherein the hierarchical structure comprises defined places for dynamically updating the set of firewall rules during runtime."

75. In my opinion, Coss discloses, and at minimum renders obvious this limitation. Coss discloses, and at minimum renders obvious, accepting (e.g., "pass") or denying (e.g., "drop") the packet based on the set of firewall rules, wherein the set of firewall rules is dynamically self-configurable during runtime without operator interaction (e.g., "dynamic rule[] which acts to alter the operation of the...initial set of rules under specified conditions"; "dynamic rules...can be loaded at any time by...a trusted application"), wherein the set of firewall rules comprises a plurality of chains of rules forming various paths through a hierarchical structure (e.g., "rules are applied sequentially"), and wherein the hierarchical structure comprises defined places for dynamically updating the set of firewall rules during runtime (e.g., at step 1012 of FIG 10's packet processing process).

76. Coss discloses a plurality of chains of rules forming various paths through a hierarchical structure and using these rules to accept or deny packets at the firewall.

a. For example, Coss teaches using "sets of access rules" that are maintained in a "rule table." The rules are "applied sequentially until a rule is

found," otherwise, the packet is dropped—confirming that the rules are applied in a linear and serialized manner. *E.g.*, Coss, 4:4-19 ("The security policies can be represented by sets of access rules which are represented in tabular form.... As illustrated in FIG. 3, such a table can provide for categories including rule number, designations of source and destination hosts, a designation of a special service which can be called for in a packet, and a specification of an action to be taken on a packet.... In FIG. 3, the categories "Source Host," "Destination Host" and "Service" impose conditions which must be satisfied by data included in a packet for the specified action to be taken on that packet...."), 4:27-37 ("In rule processing for a packet, the rules are applied sequentially until a rule is found which is satisfied by the packet (or until the rule table is exhausted, in which case the packet is dropped). For a packet to satisfy a rule, each condition included in the rule must be met. For example, with reference to FIG. 3, a packet from source host A to destination host D and representing mail will be dropped under Rule 20...").

b. As illustrated in Figures 3, the rule table includes the rule in a hierarchical structure as a chain of rules:

RULE NO.	SOURCE HOST	DEST. HOST	SERVICE	ACTION
10	A	B	FTP	PASS
20	A	*	*	DROP
30	B	C	TELNET	PROXY
40	*	D	MAIL	PASS

FIG. 3

Looking at Figure 3, for example, a POSITA would have understood that the rules are maintained as groups based on the categories and the rules would be ordered based on the hierarchical structure applied. For example, rules for Source Host A, followed by rules for Source Host B, etc. Further, as the above quotes confirm, a packet arriving from Domain A is checked against a different set of rules than a packet arriving from Domain B, confirming that the rules comprise a plurality of chains forming various paths through this hierarchical structure. This is further confirmed by the flows of figures 5A and 5B which illustrate that first rules associated with the source domain are checked; then rules associated with the destination domain, etc., and based on these rules the firewall will either pass or drop the packet.

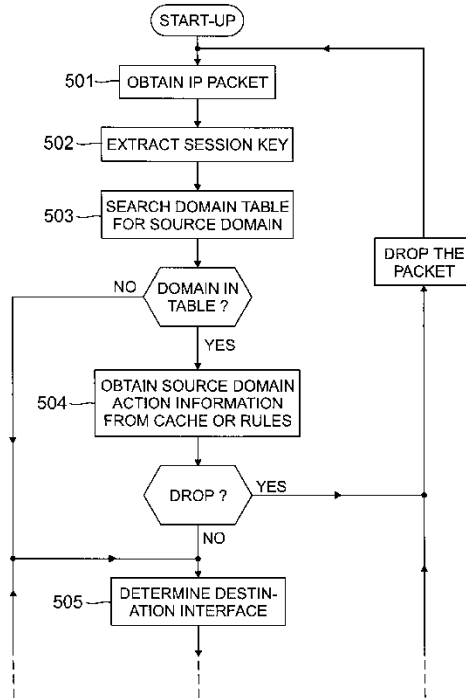


FIG. 5A

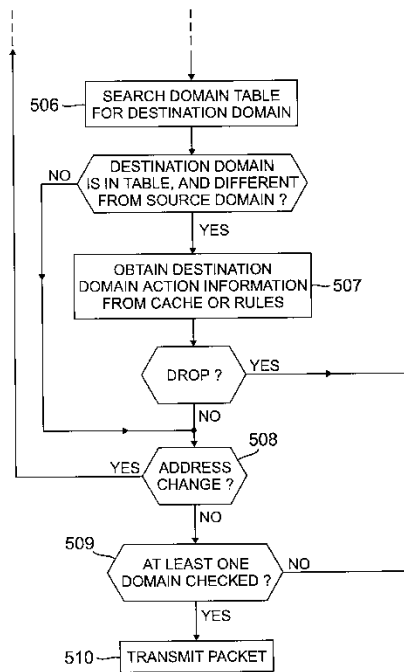


FIG. 5B

c. As Coss explains, "FIGS. 5A and 5B illustrate over-all flow for

packet processing by a firewall which supports multiple domains. ...503: on the basis of which interface received the packet and the source IP address of the received packet, the source domain is determined...; 504: ...the rule set for the source domain is searched for a match; if a match is found in the rules and if the action is not "drop," the process continues with step 505; if a match is found in the rules and the action is "drop," ..., the packet is dropped, and the process returns to step 501; if no match is found in the rules, the packet is dropped and the process returns to step 501;...506: using the destination interface and the destination address of the packet, the destination domain is determined; if the destination domain is not found, or if the destination domain matches the domain just checked, the process skips to step 508; 507: ..., rule set look-up for the destination domain are carried out in a manner analogous to that employed for the source domain in step 504;...509: if the packet was not processed with respect to any domain, the packet can be dropped, as a firewall owner has no interest in supporting communications between interfaces which are not subject to any access rules; 510: with all actions having resulted in "pass," the packet is sent out the appropriate network interface.". Coss, 6:18-7:9.

d. Further, as Coss explains, "[t]he particular rule set that is applied for any packet can be determined based on information such as the incoming

and outgoing network interfaces as well as the network source and destination addresses." E.g., Coss, 1:61-2:6. Coss thus expressly discloses applying a particular set of rules based on, e.g., the source domain, which confirms the firewall rules as a plurality of chains with a hierarchical structure.

77. Coss further discloses the set of firewall rules is dynamically self-configurable during runtime without operator interaction.

a. Coss discloses implementing a set of dynamically adaptable rules "which acts to alter the operation of [an]...initial set of rules under specified conditions." E.g., Coss, Claim 35 ("A method for providing a firewall service in a computer network, comprising the steps of: forming an augmented set of rules by including, in an already-loaded initial set of access rules, at least one dynamic rule which acts to alter the operation of the already-loaded initial set of rules under specified conditions without reloading at least one unaltered rule of the already-loaded set of access rules; and using the augmented set of rules in validating a packet; wherein the at least one rule is a dynamic rule....").

b. Coss explains that its rules may be added at runtime (*e.g.*, "as a need arises" and "can be loaded at any time") and, in particular, at particular places during the processing of a packet's traversal of the firewall. E.g., Coss, 8:27-59 ("Dynamic rules are rules which are included with the access rules as

a need arises, for processing along with the access rules, e.g., by a rule processing engine.... They can be loaded at any time by trusted parties, e.g., a trusted application, remote proxy or firewall administrator, to authorize specific network sessions.... Once a dynamic rule has served its function, it can be removed from the rule set. The dynamic rules allow a given rule set to be modified based on events happening in the network without requiring that the entire rule set be reloaded..."); see also *id.*, 2:33-46.

c. Further, Coss teaches that a dynamic rule can be loaded by "a trusted application" *or* by a "firewall administrator." Coss, 8:27-59. If Coss intended the "trusted application" to require human intervention, it need not have separately identified an application and an administrator. Thus, a POSITA would have understood that a dynamic rule could be loaded with operator intervention (i.e., by a "firewall administrator") or without operator intervention (i.e., by a "trusted application"). At minimum, based on this disclosure and the distinction between an application and an administrator, a POSITA would have found it to be an obvious design choice to allow the trusted application to load the rule without requiring an administrator to intervene.

d. Coss further confirms that these rules are dynamically self-configurable during runtime by teaching that they "allow a given rule set to

be modified based on events happening in the network" (*id.*) and by including steps for loading a dynamic rule during the processing of a given packet in the flowchart of Figure 10. *E.g.*, Coss, 9:28-10:21 (disclosure of Figures 10A-10B teaches that "dynamic rules can be used as described below" and step "1012: the firewall loads a dynamic rule to perform this action; 1013: the remote proxy sends the packet to the firewall; based on the dynamic rule loaded in step 1012").

e. Finally, Coss discloses defined places where the dynamic rules may be loaded in the process. For example, while following the rule path which sends the packet to the firewall proxy, and then following the flow of figures 10A and 10B, Coss includes an exemplary defined place—at step 1012—where a dynamic rule may be added. Coss, 9:28-10:21. This exemplary dynamic rule is loaded *after* determining the specific set of rules associated with this packet and applying the action specified in the rule (steps 1001, 1002, 1004 of FIGs 10A-10B) and a particular location—*i.e.* step 1012—within the packet processing flow. Coss, 9:28-10:21

6. Claim 2: "A method according to claim 1, wherein dynamically updating the set of firewall rules during runtime further comprises, during runtime, adding a rule to the set of firewall rules, deleting a rule from the set of firewall rules, or modifying a rule in the set of firewall rules without operator interaction."

78. In my opinion, Coss discloses, and at minimum renders obvious this

limitation. Coss discloses, and at minimum renders obvious, a method according to claim 1, wherein dynamically updating the set of firewall rules during runtime further comprises, during runtime (e.g., when a packet is being processed by a proxy application), adding a rule to the set of firewall rules (e.g., "load[ing]"/"add[ing]" a rule), deleting a rule from the set of firewall rules (e.g., "remov[ing]" a rule), or modifying a rule in the set of firewall rules without operator interaction.

79. As discussed with respect to element [1D], Coss's rules "can be loaded at any time", they "can be removed from the rule set," and the "dynamic rules allow a given rule set to be modified based on events happening in the network." Coss, 8:28-40 ("Dynamic rules are rules which are included with the access rules as a need arises,.... They *can be loaded at any time* by trusted parties, e.g., a trusted application, remote proxy or firewall administrator, to authorize specific network sessions. A dynamic rule can be set for single-session use, or its use can be limited as to time. *Once a dynamic rule has served its function, it can be removed from the rule set. The dynamic rules allow a given rule set to be modified based on events happening in the network* without requiring that the entire rule set be reloaded.")

7. Claim 3: "A method according to claim 1, wherein associating the set of firewall rules with the at least one node further comprises associating a first subset of the set of firewall rules with the first node."

80. In my opinion, Coss discloses, and at minimum renders obvious this limitation. Coss discloses, and at minimum renders obvious, a method according to

claim 1, wherein associating the set of firewall rules with the at least one node further comprises associating a first subset of the set of firewall rules with the first node.

81. As discussed above with respect to claim 1, Coss teaches that each node can have its own security policy and, accordingly, Coss implicitly teaches that the firewall must associate the subset of rules that apply to that node in order to implement the applicable security policy. Coss, 3:36-4:3 ("With a capability for supporting multiple security domains, a single firewall can support multiple users, each with a separate security policy. Also, as different security policies can apply for communications between sub-sites, such a capability can be used within a site. Respective configurations are illustrated by FIGS. 1 and 2. ¶ FIG. 1 shows four user sites 101-104, e.g., of corporations A through D, with firewall protection in their connections to the Internet 105.... ¶ FIG. 2 shows a user site 201 connected to the Internet 105 via a firewall processor 211."). For example, as reflected in Figure 3 below, the subset of rules that apply to packets originating in Domain A would be applied to the Domain A node.

RULE NO.	SOURCE HOST	DEST. HOST	SERVICE	ACTION
10	A	B	FTP	PASS
20	A	*	*	DROP
30	B	C	TELNET	PROXY
40	*	D	MAIL	PASS

FIG. 3

8. Claim 4: "A method according to claim 3, wherein the at least one node further comprises at least two nodes including a second node, further comprising associating a second subset of the set of firewall rules with the second node."

82. In my opinion, Coss discloses, and at minimum renders obvious this limitation. Coss discloses, and at minimum renders obvious, a method according to claim 3, wherein the at least one node further comprises at least two nodes including a second node, further comprising associating a second subset of the set of firewall rules with the second node.

83. As discussed above with respect to claims 1 and 3, Coss discloses a firewall protecting multiple nodes (for example, Domains A-D) with each domain being protected using its own security policies. Coss, 3:36-4:3 ("With a capability for supporting multiple security domains, a single firewall can support multiple users, each with a separate security policy. Also, as different security policies can

apply for communications between sub-sites, such a capability can be used within a site. Respective configurations are illustrated by FIGS. 1 and 2. ¶ FIG. 1 shows four user sites 101-104, e.g., of corporations A through D, with firewall protection in their connections to the Internet 105.... ¶ FIG. 2 shows a user site 201 connected to the Internet 105 via a firewall processor 211."). Thus Coss teaches at least two nodes—based on Domain A and Domain B—and a POSITA would have implicitly understood that this architecture requires associating rules for Domain A to the Domain A node and rules for Domain B to the Domain B node as reflected in Figure 3.

RULE NO.	SOURCE HOST	DEST. HOST	SERVICE	ACTION
10	A	B	FTP	PASS
20	A	*	*	DROP
30	B	C	TELNET	PROXY
40	*	D	MAIL	PASS

FIG. 3

9. Claim 5: " A method according to claim 4, wherein the first subset of firewall rules is the same as or at least partially different from the second subset of firewall rules."

84. In my opinion, Coss discloses, and at minimum renders obvious this limitation. Coss discloses, and at minimum renders obvious, a method according to

claim 4, wherein the first subset of firewall rules is the same as or at least partially different from the second subset of firewall rules.

85. As a matter of basic logic, any two sets of rules are either the same, or at least partially (if not completely) different. Thus, Coss discloses this limitation. In particular, Coss teaches having rules applicable both to domains A and B (e.g., rule 10), rules applicable to only domain A (rule 20), rules applicable to B and C (rule 30), and rules applicable to only D (rule 40). A POSITA thus would have understood Coss to disclose wherein the first subset of rules (rules applicable to domain A) are at least partially different from the second set of firewall rules—i.e., rules 10 and 20 are associated with domain A, while rules 10 and 30 are associated with domain B.

10. Claim 6: "A method according to claim 4, wherein one of the first subset of firewall rules or second subset of firewall rules equals the entire set of firewall rules."

86. In my opinion, Coss discloses, and at minimum renders obvious this limitation. Coss discloses, and at minimum renders obvious, a method according to claim 4, wherein one of the first subset of firewall rules or second subset of firewall rules equals the entire set of firewall rules.

87. In particular, Coss teaches that the firewall acts as a corporate boundary to the Internet while protecting internal communications between "sub-sites" in the corporate network as illustrated in Fig. 2:

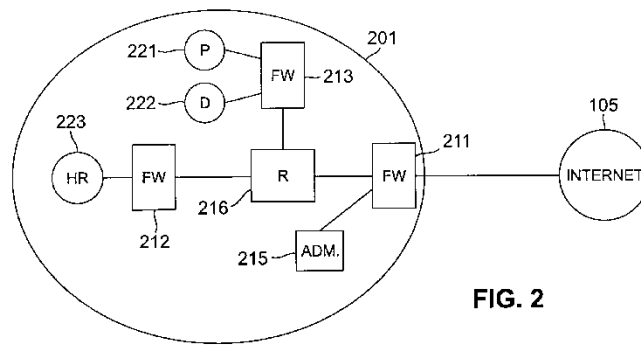


FIG. 2

Coss, FIG. 2. As Coss explains: "FIG. 2 shows a user site 201 connected to the Internet 105 via a firewall processor 211. An administrator processor 215 and a router 216 are connected to the firewall processor 211. The router 216 is connected to additional firewall processors 212 and 213 which are internal to the user site 201. The firewall processor 212 protects a single sub-site 223, such as Human Resources (HR). The firewall processor 213 is configured for protecting two sub-sites, such as Payroll (P) and Disbursements (D), vis-a-vis the remainder of the site 201 as well as with respect to communications between sub-sites 221 and 222..." Coss, 3:58-4:3.

88. According to Coss, the "firewall is able to support...multiple security policies as well as multiple users, by applying any one of several distinct sets of access rules for a given packet. The particular rule set that is applied for any packet can be determined based on information such as the incoming and outgoing network interfaces as well as the network source and destination addresses." Coss, 1:61-2:6.

89. Based on these disclosures, a POSITA would have understood or at least

found obvious that the boundary "firewall processor" (211) should be associated with the entire set of firewall rules. This would ensure that communications between each of these subsites and the Internet are properly protected and that internal communications are not inadvertently sent across the boundary to unprotected networks.

11. Claim 7: "A method according to claim 1, further comprising, after receiving the packet and prior to accepting or denying the packet, conditioning the packet based on the set of firewall rules."

90. In my opinion, Coss discloses, and at minimum renders obvious this limitation. Coss discloses a method according to claim 1, further comprising, after receiving the packet and prior to accepting or denying the packet, conditioning the packet (e.g., "processing"; "address change"; "network address translation", "encryption") based on the set of firewall rules.

91. Coss teaches that packets may be conditioned during rule processing by, for example, performing network address translation (NAT) or encryption. Coss, 4:4-19 ("As illustrated in FIG. 3, such a table can provide for categories including rule number, designations of source and destination hosts, a designation of a special service which can be called for in a packet, and a specification of an action to be taken on a packet. Special services can include proxy services, network address translation, and encryption, for example."). I note that the '282 patent admits that "conditioning" a packet includes NAT. EX1001, 6:18-28, 6:60-7:12.

92. Moreover, a POSITA would have understood NAT to take place before applying rules to accept/drop a packet because NAT translates the packet's source or destination IP address to their internally-routable forms, and that translation would therefore ensure that the appropriate domain (or sub-domain) rules are applied to the packet. Likewise, a POSITA would have understood that encryption (for example, to transmit packets across an encrypted channel to a proxy application for filtering, Coss at 9:31-10:15) would have been performed before the proxy application applies the applicable rule set to the packets.

93. Coss also teaches that rules may "call[] for an address change, e.g., to a proxy or for insertion of one packet into another ("tunnel option")," in which case, "the packet's destination address is replaced with the address of the remote proxy [and] the destination port can be changed as well." Coss, 6:66-7:3. In these cases, "the original packet header data is recorded in the session cache along with any changed values." *Id.* Again, for the same reasons stated above, a POSITA would have understood these initial address changes to take place before the firewall decides whether to allow the packet to its final destination.

12. Claim 8: "A method according to claim 7, wherein conditioning the packet based on the set of firewall rules further comprises rewriting a portion of a network packet header associated with the packet."

94. In my opinion, Coss discloses, and at minimum renders obvious this limitation. Coss discloses a method according to claim 7, wherein conditioning the

packet based on the set of firewall rules further comprises rewriting (e.g., "changing") a portion of a network packet header associated with the packet (e.g., "source port", "destination port", "destination address", "source address").

95. As discussed with respect to Claim 7, Coss's teachings of performing "address change[s]" and "Network Address Translation" (i.e., NAT), each requires rewriting a portion of the network packet header associated with that packet. Coss, 6:66-7:3 ("if a rule that applies to the packet calls for an address change"), 8:61-9:9 (disclosing proxy reflection in which "the firewall replaces the destination address in the packet with the host address of the proxy application..."), 9:42-48 ("the packet's destination address is replaced with the address of the remote proxy; if configured, the destination port can be changed as well" and "the original packet header data is recorded...along with any changed values.").

96. Moreover, a POSITA would have understood Coss's teachings that "IP header values are changed back to the original values" *after* passing through the firewall as proof that the initial rewriting of the header takes place before the packet has finished moving through the firewall. Coss, 10:10-17. In other words, when the address changes were initially made, the firewall rules had not yet determined whether the packet should be allowed or dropped. After moving through the firewall, "accepted" (or "allowed") packets are then "changed back to their original values."

13. Claim 9: "A method according to claim 1, wherein each of the at least one node is associated with at least two network interfaces."

97. In my opinion, Coss discloses, and at minimum renders obvious this limitation. Coss renders obvious a method according to claim 1, wherein each of the at least one node is associated with at least two network interfaces.

98. As discussed above with respect to claim 1, the independent claim already recites and requires that a "node is associated with two or more network interfaces." As a matter of basic logic, a node associated with "two or more interfaces" is necessarily associated with "at least two" interfaces. Accordingly, this claim is met for the same reasons claim 1 is met.

14. Claim 10: "A method according to claim 1, wherein each of the two or more network interfaces is connected with at least one physical device."

99. In my opinion, Coss discloses, and at minimum renders obvious this limitation. Coss renders obvious a method according to claim 1, wherein each of the two or more network interfaces is connected with at least one physical device.

100. As explained with respect to element [1a] above, Coss's firewall is configured to protect multiple domains (i.e., nodes). E.g., Coss, 3:43-57 ("FIG. 1 shows four user sites 101-104, e.g., of corporations A through D, with firewall protection in their connections to the Internet 105...."), 3:58-4:3 ("FIG. 2 shows a user site 201 connected to the Internet 105 via a firewall processor 211. An

administrator processor 215 and a router 216 are connected to the firewall processor 211. The router 216 is connected to additional firewall processors 212 and 213 which are internal to the user site 201. The firewall processor 212 protects a single sub-site 223, such as Human Resources (HR)...").

101. Further, Coss teaches that the network interfaces may be implemented as physical devices—network interface cards (NICs) and that "each domain is associated with one or more network interfaces" and includes both "incoming" and "outgoing" interfaces. Coss, 6:3-25 ("In the firewall, a decision module or engine, here called a "domain support engine" (DSE) determines which security policy to use for a new network session. Each new session must be approved by the security policies of the source domain and the destination domain(s). For connections going to the Internet, it is likely that only a single domain check is performed. The ***DSE makes the domain selection based on the incoming or outgoing network interface***, as well as on the source or destination network address of each packet. Inclusion, in packets, of source or destination addresses allows for multiple users to be supported by a single network interface. The ***incoming or outgoing network interface may be in the form of a network interface card (NIC), e.g., an Intel EtherExpress Pro 100B card available from Intel Corporation.*** ¶ FIGS. 5A and 5B illustrate over-all flow for packet processing by a firewall which supports multiple domains. Such processing includes determining the domains which the packet is to cross, examining

the applicable rules to ascertain whether the packet may pass, and determining whether any special processing is required. In the firewall, *each domain is associated with one or more network interfaces*. Interfaces that support more than one domain are separated using an IP address range to distinguish the packets.").

15. Claim 11: "A method according to claim 1, wherein the set of firewall rules being dynamically self-configurable further comprises dynamically updating the set of firewall rules during runtime without operator interaction."

102. In my opinion, Coss discloses, and at minimum renders obvious this limitation. Coss discloses, and at minimum renders obvious, a method according to claim 1, wherein the set of firewall rules being dynamically self-configurable further comprises dynamically updating the set of firewall rules during runtime without operator interaction.

103. As discussed above with respect to elements [1D] and claim 2, Coss teaches the use of dynamic rules that can be loaded "at any time by trusted parties," including by "a trusted application" or a "firewall administrator." Coss, 8:28-40. As I explain above in those limitations, a POSITA would have therefore understood Coss as disclosing dynamically updating the set of firewall rules during runtime (e.g., while processing a packet through a proxy) and without operator intervention (e.g., by a trusted application in contrast to a firewall administrator). *See also* Coss at, e.g., Claim 35, 2:33-46, 9:66-10:21.

16. Claim 12[pre]: "A device for controlling data through a firewall, comprising:"

104. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Element [1PRE], above.

17. Claim 12[a]: " a plurality of network interfaces, wherein each of the plurality of network interfaces is operable to utilize one or more physical devices;"

105. In my opinion, Coss discloses, and at minimum renders obvious this limitation. Coss discloses, and at minimum renders obvious, a plurality of network interfaces, wherein each of the plurality of network interfaces is operable to utilize one or more physical devices.

106. As discussed above in Element [1a] and claim 10, Coss's firewall protects multiple domains (i.e., nodes). Coss at, e.g., 3:36-4:3. In addition, Coss discloses that "an incoming or outgoing network interface may be in the form of a network interface card (NIC) e.g., an Intel EtherExpress Pro 100B card available from Intel Corporation."

18. Claim 12[b]: "a first computer readable storage medium storing a set of firewall rules, wherein the set of firewall rules is dynamically self-configurable during runtime without operator interaction, wherein the set of firewall rules comprises a plurality of chains of rules forming various paths through a hierarchical structure, and wherein the hierarchical structure comprises defined places for dynamically updating the set of firewall rules during runtime without operator interaction;"

107. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Elements [1B] and [1D], above.

19. Claim 12[c]: "a data controlling computer program comprising data controlling computer program code stored on either the first computer readable storage medium or on a second computer readable storage medium, the data controlling computer program code being executable to:"

108. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Element [1PRE], above.

20. Claim 12[C(i)]: "define at least one node, wherein the at least one node is associated with two or more network interfaces of the plurality of network interfaces; and"

109. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Elements [1A] and [1PRE], above.

21. Claim 12[C(ii)]: "when a packet is received at one of the two or more network interfaces associated with the at least one node, accept or deny the packet based on a review of the set of firewall rules."

110. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Elements [1C] and [1D], above.

22. Claim 13: "A device according to claim 12, wherein the data controlling computer program code is further executable to, while the firewall is processing traffic through the at least one node, add a rule to the set of firewall rules, delete a rule from the set of firewall rules, or modify a rule in the set of firewall rules."

111. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Claim 2, above.

23. Claim 14[pre]: "A device according to claim 12, wherein the at least one node comprises a first node, and wherein the data controlling computer program code is further executable to:"

112. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Claim 3, above.

24. Claim 14[a]: "associate a first subset of the set of firewall rules with the first node; and"

113. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Claim 3, above.

25. Claim 14[b]: " if the packet is received at the first node, apply the first subset of the set of firewall rules associated with the first node."

114. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Claim 3, above.

26. Claim 15[pre]: "A device according to claim 14, wherein the at least one node further comprises at least two nodes including a second node, and wherein the data controlling computer program code is further executable to:"

115. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Claim 4, above.

27. Claim 15[a]: "associate a second subset of the set of firewall rules with the second node; and"

116. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Claim 4, above.

28. Claim 15[b]: "if the packet is received at the second node, apply the second subset of the set of firewall rules associated with the second node."

117. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Claim 4, above.

29. Claim 16: "A device according to claim 15, wherein the first subset of the set of firewall rules is the same as or different from the second subset of the set of firewall rules."

118. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Claim 5, above.

30. Claim 17: "A device according to claim 15, wherein either the first subset of the set of firewall rules or the second subset of the set of firewall rules equals the entire set of firewall rules."

119. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Claim 6, above.

31. Claim 18: "A device according to claim 12, wherein the data controlling computer program code is further executable to condition the packet based on the set of firewall rules."

120. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Claim 7, above.

32. Claim 19: "A device according to claim 18, wherein conditioning the packet based on the set of firewall rules further comprises rewriting a portion of a network packet header associated with the packet."

121. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Claim 8, above.

33. Claim 20: "A device according to claim 12, wherein each of the at least one node is associated with at least two network interfaces."

122. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Claim 9, above.

34. Claim 21: "A device according to claim 12, wherein each of the plurality of network interfaces is connected with at least one physical device."

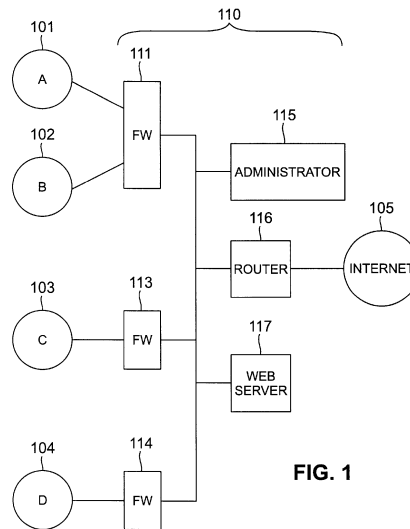
123. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Claim 10, above.

35. Claim 22: "A device according to claim 12, wherein each of the plurality of network interfaces is physically connected to every other network interface of the plurality of network interfaces and wherein physical connection between the plurality of network interfaces comprises indirect physical connection between the plurality of network interfaces."

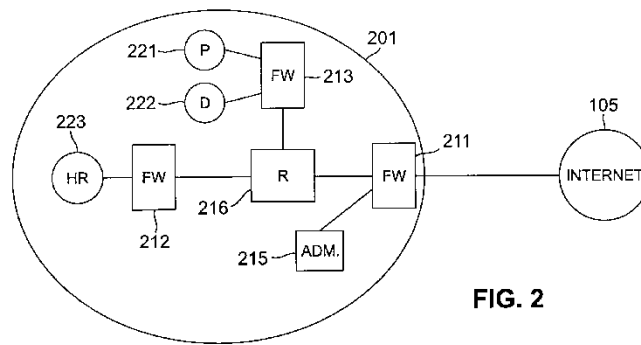
124. In my opinion, Coss discloses, and at minimum renders obvious this limitation. Coss discloses, and at minimum renders obvious, a device according to claim 12, wherein each of the plurality of network interfaces is physically connected to every other network interface of the plurality of network interfaces and wherein physical connection between the plurality of network interfaces comprises indirect physical connection between the plurality of network interfaces.

125. Coss's firewall protects "user sites" and "sub-sites" with each site connected via direct and/or indirect connections through, for example, a router. For example, with respect to Figure 1, Coss teaches: "FIG. 1 shows four user sites 101-

104, e.g., of corporations A through D, with firewall protection in their connections to the Internet 105. Such protection is provided by a firewall facility, here in the form of a LAN 110, including firewall processors 111, 113 and 114, an administrator processor 115, a router 116 and a web server 117...." Coss, 3:43-57.



126. Similarly, for Figure 2, Coss teaches: "FIG. 2 shows a user site 201 connected to the Internet 105 via a firewall processor 211. An administrator processor 215 and a router 216 are connected to the firewall processor 211. The router 216 is connected to additional firewall processors 212 and 213 which are internal to the user site 201. The firewall processor 212 protects a single sub-site 223, such as Human Resources (HR). The firewall processor 213 is configured for protecting two sub-sites, such as Payroll (P) and Disbursements (D), vis-a-vis the remainder of the site 201 as well as with respect to communications between sub-sites 221 and 222." Coss, 3:58-4:3.



127. Based on these disclosures, a POSITA would have at least found it obvious to connect, directly or indirectly, each such interface in order to properly communicate packets across the network.

36. Claim 23: "A device according to claim 12, wherein the data controlling computer program code is further executable to dynamically update the set of firewall rules during runtime without operator interaction."

128. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Claim 11, above.

37. Claim 24[pre]: "A data controlling computer program product comprising computer instructions stored on at least one non-transitory computer readable medium, wherein the computer instructions are operable when executed by at least one processor to:"

129. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Claim [1PRE], above.

38. Claim 24[a]: "define at least one node for controlling data through a firewall, wherein at least one of the at least one node is associated with two or more network interfaces;"

130. In my opinion, Coss discloses, and at minimum renders obvious this

limitation for the same reasons as discussed in Claim [1A], above.

39. Claim 24[b]: "associate a set of firewall rules with the at least one node, wherein the set of firewall rules further comprises a plurality of chains of rules forming various paths through a hierarchical structure, and wherein the hierarchical structure comprises defined places for dynamically updating the set of firewall rules during runtime without operator interaction;"

131. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Claim [1B] and [1D], above.

40. Claim 24[c]: "receive a packet at a first node of the at least one node; and"

132. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Claim [1C], above.

41. Claim 24[d]: "accept or deny the packet based on a review of the set of firewall rules, wherein the set of firewall rules is dynamically self-configurable during runtime without operator interaction."

133. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Claim [1D], above.

42. Claim 25: "A data controlling computer program product according to claim 24, wherein the computer instructions are further executable to dynamically update the set of firewall rules during runtime without operator interaction."

134. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Claim 2, above.

43. Claim 26: "A data controlling computer program product according to claim 25, wherein the computer instructions are further executable to, while the firewall is processing traffic

through the at least one node, add a rule to the set of firewall rules, delete a rule from the set of firewall rules, or modify a rule in the set of firewall rules."

135. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Claim 2, above.

44. Claim 27[pre]: "A data controlling computer program product according to claim 26, wherein the computer instructions are further executable to:"

136. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Claim 3, above.

45. Claim 27[a]: "associate a first subset of the set of firewall rules with the first node; and"

137. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Claim 3, above.

46. Claim 27[b]: "if the packet is received at the first node, apply the first subset of the set of firewall rules associated with the first node."

138. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Claim 3, above.

47. Claim 28[pre]: "A data controlling computer program product according to claim 27, wherein the at least one node further comprises at least two nodes including a second node, and wherein the computer instructions are further executable to:"

139. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Claim 4, above.

48. Claim 28[a]: "associate a second subset of the set of firewall rules with the second node; and"

140. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Claim 4, above.

49. Claim 28[b]: "if the packet is received at the second node, apply the second subset of the set of firewall rules associated with the second node."

141. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Claim 4, above.

50. Claim 29: "A data controlling computer program product according to claim 28, wherein the first subset of the set of firewall rules is the same as or different from the second subset of the set of firewall rules."

142. I In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Claim 5, above.

51. Claim 30: "A data controlling computer program product according to claim 28, wherein either the first subset of the set of firewall rules or the second subset of the set of firewall rules equals the entire set of firewall rules."

143. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Claim 6, above.

52. Claim 31: "A data controlling computer program product according to claim 24, wherein the computer instructions are further executable to condition the packet based on the set of firewall rules."

144. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Claim 7, above.

53. Claim 32: "A data controlling computer program product according to claim 31, wherein conditioning the packet based on the set of firewall rules further comprises rewriting a portion of a network packet header associated with the packet."

145. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Claim 8, above.

54. Claim 33: "A data controlling computer program product according to claim 24, wherein each of the at least one node comprises at least two network interfaces."

146. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Claim 9, above.

55. Claim 34: "A data controlling computer program product according to claim 24, wherein each of the two or more network interfaces is connected with at least one physical device."

147. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Claim 10, above.

56. Claim 35: "A data controlling computer program product according to claim 24, wherein each of the two or more network interfaces is physically connected to every other network interface of the two or more network interfaces and wherein physical connection between the two or more network interfaces comprises indirect physical connection between the two or more network interfaces."

148. In my opinion, Coss discloses, and at minimum renders obvious this limitation for the same reasons as discussed in Claim 22, above.

XI. GROUND 3 – CLAIMS 1-35 ARE OBVIOUS OVER COSS IN VIEW OF KE

149. In my opinion, Coss in view of Ke renders obvious the elements of the

independent claims and, in turn, the elements of all the dependent claims. In particular, Coss already discloses and at minimum renders obvious (as discussed above) all the elements of the claims, including "defining at least one node, wherein the at least one node is associated with two or more network interfaces." *See*, Element [1A], above. However, to the extent Patent Owner argues that Coss does not expressly disclose "defining" a node because such definition requires disclosure of the specific configuration steps, Ke includes these teachings and it would have been obvious for a POSITA to implement Coss using these teachings from Ke.

150. Specifically, Ke discloses defining at least one node (e.g., "configuring an Internet security system"; "VLAN"/"VPN") wherein the at least one node is associated with two or more network interfaces ("adding two virtual interfaces for the [] system").

151. As discussed above in the overview of Ke, Ke describes a configurable firewall that secures multiple virtual local area networks (VLANs) using a packet-based firewall model. Ke, [0005]. A POSITA would have understood Ke's teachings of using "virtual systems" to create and manage customer domains to have similar uses as Coss's disclosure of a multi-domain system to protect user sites for different corporations. *See* Coss, 3:35-56 (disclosing "multiple security domains" comprising "four user sits...of corporations A through D"), Ke, [0031] ("multi-customer, multi-domain architecture"). Thus, a POSITA would have understood that to the extent

Coss discloses a multi-domain system (and one with defined host "groups" that are dynamically modifiable), but does not provide explicit disclosure on how to configure and define such domains, **Ke** provides those implementation details.

152. **Ke** teaches "configuration of the Internet security system in real time or at start up with a saved configuration script." **Ke**, [0058]. In particular, **Ke** discloses creating "virtual systems" (each with its own firewall policies for incoming and outgoing packets) in order to configure the security system to support multiple domains. *E.g.*, **Ke**, [0005] ("***The data processing system includes a firewall engine that can receive a set of firewall policies and apply the firewall policies to a data packet***, an authentication engine that can receive a set of authentication policies and authenticate a data packet in accordance with the authentication policies, ***one or more virtual private networks that each have an associated destination address and policies*** and a controller that can detect an incoming data packet, ***examine the incoming data packet for a virtual private network destination address and identify the policies associated with the virtual private network destination***. If the policies include firewall policies, then the controller can call the firewall engine and ***apply the set of firewall policies corresponding to the virtual private network destination to the data packet.***"), [0016] (describing "security system resources including ***firewall services and a controller that can partition the security system resources into a plurality of separate security domains***. Each security domain can be

configurable to enforce one or more policies relating to a specific subsystem, and to allocate security system resources to the one or more security domains."

153. **Ke** provides an express example of defining "a new virtual system named 'marketing' and configur[ing] that system." Ke, [0069]. The configuration steps include "adding two virtual interfaces for the 'marketing' system." *Id.* As Ke further explains, the "virtual interfaces" act as the incoming and outgoing interfaces for the VLAN, "allow[ing] several streams of data to share the same physical medium, such as a trunk cable, while enjoying total segregation." *Id.* [0033].

154. A POSITA would have been motivated to implement Coss's multi-domain firewall architecture using Ke's approach of defining domains as "virtual systems" with corresponding network interfaces. Both Coss and Ke operate in the same field of computer networking—specifically, protecting multiple domains with distinct firewall policies—and both address similar challenges. For example, the '282 patent describes "network firewalls that can dynamically adapt to changing conditions and operator requirements" (EX1001, 1:35–37), while Ke highlights the "complex[ity]" and high cost of relying on physical network infrastructure that requires "a large amount of separate equipment" and must be reconfigured "every time a new customer joins the Internet data center," making it "a labor intensive and costly task" (Ke, [0004]). A POSITA would have recognized that applying Ke's virtual domain model to Coss's multi-domain system would reduce time, cost, and

complexity. Both references also stress runtime efficiency—Coss through its use of dynamic rules loaded and removed at runtime (Coss, 8:28–40), and Ke through its real-time reconfiguration capabilities (Ke, [0058]). A POSITA would have appreciated that incorporating Ke’s runtime flexibility into Coss’s framework would provide additional operational benefits. Moreover, Coss’s disclosure of “host group[s]” that can be dynamically modified without disrupting other rule sets (Coss, 2:41–46) provides an express motivation to combine with Ke’s teachings. Taken together, a POSITA would have recognized the scalability and cost advantages of implementing Ke’s virtual system approach within Coss’s multi-domain firewall.

155. A POSITA would have had a reasonable expectation of success in implementing Coss in view of Ke because the combination relies on straightforward and well-understood networking concepts. Coss’s approach to partitioning domains is conceptually aligned with Ke’s use of virtual domains, and a POSITA would have recognized that mapping Coss’s domains onto VLANs represents a natural integration of compatible technologies. For instance, Coss already identifies the use of “tunneling” as a mechanism for secure communication between domains (Coss, 6:66–7:3, *discussing address changes and “tunnel option” for packet redirection*), and a POSITA would have understood that these tunnels could be efficiently realized using Ke’s VLAN framework (Ke, [0052]–[0053]). In this way, Coss and Ke provide complementary disclosures, and a POSITA would have seen that combining them

applies established networking principles to achieve their intended purposes.

XII. SECONDARY CONSIDERATIONS OF NON-OBVIOUSNESS

156. Neither the applicant nor the Examiner identified any secondary considerations that favored a finding of non-obviousness during prosecution of the application that issued as the '282 Patent. *See generally* EX1002. Further, I am unaware of any secondary considerations that tend to show non-obviousness of the Challenged Claims. To the extent that such secondary considerations become available to me in the course of this proceeding, I reserve the right to address them in a supplemental declaration.

157. I declare under penalty of perjury that the foregoing is true and correct.

Executed this 10 day of October, 2025 in Louisville, Colorado.

A handwritten signature in black ink, appearing to be 'D. B. J.', written in a cursive style.

Appendix A

John R. Black, Jr.

Department of Computer Science
430 UCB
University of Colorado
Boulder, CO 80309-0430 USA

office: +1 303 492-0573
FAX: +1 303 492-2844
secretary: +1 303 492-7514

Email: jrblack@cs.colorado.edu
WWW: <http://www.cs.colorado.edu/~jrblack/>

Position	Assoc. Prof. Computer Science, University of Colorado at Boulder	<i>7/08–present</i>
Research	Cryptography, Network Security, Computer Security.	
Past Employment	SecureSet LLC Vice President of Education	<i>9/15–7/18</i>
	University of Colorado at Boulder Assistant Professor of Computer Science.	<i>7/02–7/08</i>
	University of Nevada, Reno Assistant Professor of Computer Science.	<i>7/00–6/02</i>
	University of California, Davis Research Assistant	<i>7/97–7/00</i>
	University of California, Davis Teaching Assistant	<i>8/95–6/97</i>
	Ingres Corporation Senior Member of Technical Staff	<i>6/88–4/94</i>
Education	University of California, Davis Ph.D. in Computer Science. Thesis: Message Authentication Codes. Advisor: Phillip Rogaway.	<i>9/95–9/00</i>
	California State University at Hayward B.S. in Computer Science and Mathematics, 1988. Honors: Summa Cum Laude	<i>9/84–6/88</i>
Awards	NSF CAREER Award, 2002 Chancellor’s Teaching Fellowship, UC Davis, 1998 Outstanding Teaching Assistant, UC Davis, 1998 Outstanding Teaching Assistant, UC Davis, 1997 A Check for \$2.56 from Don Knuth, 1996	

Journal Publications

1. P. Rogaway, M. Bellare and J. Black, “OCB: A Block-Cipher Mode of Operation for Efficient Authenticated Encryption.” *ACM Transactions on Information and System Security (TISSEC)*, Volume 6, Issue 3, pp. 365–403, August, 2003.
2. J. Black, and P. Rogaway, “CBC MACs for Arbitrary Length Messages: The Three-Key Constructions.” *Journal of Cryptology*, Volume 18, Number 2, pp. 111–132, Spring, 2005.
3. J. Black, “The Impossibility of Technology-Based DRM and a Modest Suggestion.” *Journal of Telecommunications and High-Technology Law —JTHTL*, Volume 3, Number 2, pp. 387–396, Spring, 2005.
4. J. Black, M. Cochran and R. Gardner, “An Analysis of the Internet Chess Club.” *IEEE Security and Privacy*, Volume 4, Number 1, pp. 46–52, January, 2006.
5. J. Black, M. Cochran and T. Shrimpton, “On the Impossibility of Highly-Efficient Blockcipher-Based Hash Functions.” *Journal of Cryptology*, Volume 22, Number 3, pp. 311–329, Fall, 2009.
6. J. Black, P. Rogaway, T. Shrimpton, and M. Stam “An Analysis of the Blockcipher-Based Hash Functions from PGV.” *Journal of Cryptology*, Volume 23, Number 4, pp. 519–545, Fall, 2010.
7. C. Wilks, M. Cline, E. Weiler, M. Diehkans, B. Craft, C. Martin, D. Murphy, H. Pierce, J. Black, D. Nelson, B. Litzinger, T. Hatton, L. Maltbie, M. Ainsworth, P. Allen, L. Rosewood, E. Mitchell, B. Smith, J. Warner, J. Groboske, H. Telc, D. Wilson, B. Sanford, H. Schmidt, D. Haussler, D. Maltbie “The Cancer Genomics Hub (CGHub): overcoming cancer through the power of torrential data.” *Database*, Volume 2014, doi: 10.1093/database/bau093.

Book Chapters

1. J. Black, “Cryptography.” Invited article for the Encyclopedia of Life Support Systems under the auspices of UNESCO. See <http://www.eolss.net>. 14 pages, March, 2004.
2. J. Black, “Authenticated Encryption.” Invited article for the *Encyclopedia of Cryptography and Security*, Springer-Verlag. 12 pages. August, 2005.

**Conference
Publications
(Refereed)**

1. J. Black, C. Martel, and H. Qi, “Graph and Hashing Algorithms for Modern Architectures: Design and Performance.” *Workshop on Algorithm Engineering — WAE ’98*, Saarbrücken, Germany. Second Workshop on Algorithm Engineering, proceedings, pp. 37–48. Full version of this paper available at theory.cs.ucdavis.edu.
2. J. Black, S. Halevi, H. Krawczyk, T. Krovetz, and P. Rogaway, “UMAC: Fast and Secure Message Authentication.” *Advances in Cryptology — CRYPTO ’99*, Lecture Notes in Computer Science, Vol. 1666, Springer-Verlag, pp. 216–233, 1999. Full version and updated version of this paper available at www.cs.ucdavis.edu/~rogaway/umac.
3. J. Black and P. Rogaway, “CBC MACs for Arbitrary-Length Messages: The Three-Key Constructions.” *Advances in Cryptology — CRYPTO 2000*, Lecture Notes in Computer Science, Vol. 1880, Springer-Verlag, pp. 197–215, 2000.
4. P. Rogaway, M. Bellare, J. Black and T. Krovetz, “OCB: A Block-Cipher Mode of Operation for Efficient Authenticated Encryption.” *Eighth ACM Conference on Computer and Communications Security (CCS-8)*, ACM Press, pp. 196–205, 2001.
5. J. Black and P. Rogaway, “Enciphering Finite Sets of Arbitrary Size.” *RSA Data Security Conference, Cryptographer’s Track (RSA-CT)*, Lecture Notes in Computer Science, Vol. 2271, Springer-Verlag, pp. 114–130, 2002.
6. J. Black and P. Rogaway, “A Block-Cipher Mode of Operation for Parallelizable Message Authentication.” *Advances in Cryptology — EUROCRYPT 2002*, Lecture Notes in Computer Science, Vol. 2332, Springer-Verlag, pp. 384–397, 2002.
7. J. Black and H. Urtubia, “Side-Channel Attacks on Symmetric Encryption Schemes: The Case for Authenticated Encryption.” *USENIX Security Symposium — Security ’02*. 10 pages, 2002.
8. J. Black, P. Rogaway, and T. Shrimpton, “Black-Box Analysis of the Block-Cipher-Based Hash-Function Constructions from PGV.” *Advances in Cryptology — CRYPTO 2002*, Lecture Notes in Computer Science, Vol. 2442. 16 pages, 2002.
9. J. Black, P. Rogaway, and T. Shrimpton, “Encryption Scheme Security in the Presence of Key-Dependent Messages.” *Selected Areas in Cryptography — SAC 2002*, Lecture Notes in Computer Science, Vol. 2595, 14 pages, 2002.
10. R. Motwani, J. Breidenbach and J. Black, “Collocated Dataglyphs for Large Message Storage and Retrieval.” *Security, Steganography, and Watermarking of Multimedia Contents VI*, Society for Imaging Science and Technology (I&ST) jointly with International Society for Optical Engineering (SPIE), Vol. 5306, 19 pages, 2004.
11. J. Black, M. Cochran and T. Shrimpton, “On the Impossibility of Highly-Efficient Blockcipher-Based Hash Functions.” *Advances in Cryptology — EUROCRYPT 2005*, Lecture Notes in Computer Science, Vol. 3494, Springer-Verlag, pp. 526–541, 2005.
12. J. Black, M. Cochran and R. Gardner, “Lessons Learned: A Security Analysis of the Internet Chess Club.”, *Annual Computer Security Applications Conference — ACSAC 2005*, Tucson AZ, USA, pp. 220–228, 2005.
13. J. Black and M. Cochran and T. Highland, “A Study of the MD5 Attacks: Insights and Improvements”, *Fast Software Encryption — FSE 2006*, Lecture Notes in Computer Science, Vol. 4047, Springer-Verlag, pp. 262–277, 2006.

**Conference
Publications
(cont.)**

14. J. Black, “The Ideal-Cipher Model, Revisited: An Uninstantiable Blockcipher-Based Hash Function.”, *Fast Software Encryption — FSE 2006*, Lecture Notes in Computer Science, Vol. 4047, Springer-Verlag, pp. 328–340, 2006.
15. J. Black, “Compare-by-Hash: A Reasoned Analysis”, *USENIX Annual Technical Conference — USENIX 2006*, 8 pages, 2006.
16. J. Black and M. Cochran, “MAC Reforgeability”, *Fast Software Encryption — FSE 2009*, Lecture Notes in Computer Science, Vol. 5665, Springer-Verlag, pp. 345–362, 2009.
17. J.H. Huang, J. Black, and S. Mishra, “Security and Privacy in a Sensor-Based Search and Rescue System,” *1st ICST/CREATE-NET International Conference on Ad Hoc Networks — ADHOCNETS 2009*, Vol. 28, Springer.
18. A. Sayler, D. Grunwald, J. Black, E. White, M. Monaco, “Supporting CS education via virtualization and packages: tools for successfully accommodating “bring-your-own-device” at scale,” *SIGCSE 2014*, pp. 313-318, 2014.

**Workshop
Publications
(Non-
Refereed)**

1. J. Black and P. Rogaway, “A Suggestion for Handling Arbitrary-Length Messages with the CBC MAC.” *NIST Symmetric Key Block Cipher Modes of Operation Workshop—2000*, 4 pages, Sep 2000.
2. J. Black and P. Rogaway, “OCB: Proposal to NIST.” *2nd NIST Symmetric Key Block Cipher Modes of Operation Workshop—2001*, 36 pages, Aug 2001.
3. J. Black and P. Rogaway, “PMAC: Proposal to NIST.” *2nd NIST Symmetric Key Block Cipher Modes of Operation Workshop—2001*, 27 pages, Aug 2001.

Selected Talks

1. Data Structures for Fast Graph Algorithms. Presented at the 1997 UC Davis Workshop on Computing, Davis, USA, October 1997. (See Conference Publication #1.)
2. UMAC: Fast and Secure Message Authentication. Presented at CRYPTO '99, Santa Barbara, USA, August 1999. (See Conference Publication #2)
3. CBC MACs for Arbitrary-Length Messages: The Three-Key Constructions. Presented at CRYPTO 2000, Santa Barbara, USA, August 2000. (See Conference Publication #3)
4. A Suggestion for Handling Arbitrary-Length Messages with the CBC MAC. Presented at NIST Symmetric Key Block Cipher Modes of Operation Workshop—2000, October, 2000; also presented at the 2nd NIST Modes Workshop in Santa Barbara, USA, August 2001.
5. Enciphering Finite Sets of Arbitrary Size. Presented at RSA-CT '02, San Jose, USA, February 2002. (See Conference Publication #5)
6. A Block-Cipher Mode of Operation for Parallelizable Message Authentication. Presented at EUROCRYPT 2002, Amsterdam, The Netherlands, May 2002. (See Conference Publication #6)
7. Side-Channel Attacks on Symmetric Encryption Schemes. Presented at USENIX Security 2002, San Francisco, USA, August 2002. (See Conference Publication #7)
8. Practical Cryptography and Autonomic Web Computing. Invited talk at the 47th meeting of the IFIP Working Group 10.4. Rincon, Puerto Rico, January 2005.
9. On the Impossibility of Highly-Efficient Blockcipher-Based Hash Functions. Presented at EUROCRYPT 2005, Aarhus, Denmark, May 2005. (See Conference Publication #11)
10. The Ideal-Cipher Model, Revisited: An Uninstantiable Blockcipher-Based Hash Function. Presented at FSE 2006, Graz, Austria, March 2006. (See Conference Publication #14)
11. Compare-by-Hash: A Reasoned Analysis. Presented at USENIX Technical Conference 2007, Boston, MA, June 2006. (See Conference Publication #15)

Patents

1. T. McSheery, J. Black, S. Nollet, J. Johnson, and V. Jivan. Distributed-Processing Motion Tracking System for Tracking Individually Modulated Light Points. US Patent 6,324,296 B1. November 2001.

Funding

1. NCHIA E-Team Grant. “Entrepreneurship for Undergraduates.” PI: John Black. Period: 2000-2001. Amount: \$6,000.
2. University of Nevada Junior Faculty Research Grant. “Fast, Provably-Secure Cryptography.” PI: John Black. Period: 2001-2002. Amount: \$10,000.
3. NSF CAREER Award. “Highly-Optimized Provably-Secure Cryptography.” PI: John Black. Period: 2002-2007. Amount: \$469,925.
4. NSF NeTS Grant. “NeTS ProWIN: Topology And Routing With Steerable Antennas.” PI: Dirk Grunwald. Co-PIs: John Black, Douglas Sicker. Period: 2005–2008. Amount: \$745,215.
5. NSF Cybertrust Grant. “Cryptography for Constrained Environments.” PI: John Black. Period: 2005–2008. Amount: \$294,887.

Teaching History

ECS 122A — Design and Analysis of Algorithms (UC Davis). Co-taught once with Professor Rogaway; subsequently taught the course independently.

CMPSC 290G — Intro to Cryptoanalysis (UCSB).

CS 365 — Discrete Mathematics (UNR).

CS 425 — Software Engineering (UNR).

CS 426 — Senior Projects (UNR). Supervised 11 group projects in topics ranging from fingerprint recognition to audio editing to GUI design.

CS 432 — Computer Networks (UNR). Introduction to low-level networking concepts with an emphasis on network security.

CS 665 — Graduate Analysis of Algorithms (UNR). A typical algorithms course with emphasis on complexity theory.

CS 709 — Modern Cryptography (UNR). A graduate course introducing cryptography and visiting some of the research front.

CS 791G — Computer Network Security (UNR). A seminar course covering various topics related to network security.

CSCI 2270 — Data Structures (CU); Program design, Object orientation, Java, Linked lists, Arrays, Stacks and Queues, Hash tables, Trees, Balanced Binary Trees, Multi-core programming.

CSCI 3104 — Algorithms (CU); Divide-and-conquer, Greedy, Graph Algorithms, NP-Completeness, Quantum Algorithms.

CSCI 3753 — Operating Systems (CU); Scheduling, Virtual Memory, Filesystems, Multi-core systems, Pthreads, Kernel data structures, virtualization, Security.

CSCI 4830 — Network Security (CU); a new course developed to introduce basics of cryptography and network security. Covers SSL, PKI, DDOS attacks, wireless security, buffer overruns, and more.

CSCI 4900 — Solving Puzzles with Computers (CU); a one-unit undergraduate course describing some hard combinatorial puzzles and how computers can be used to attack them.

CSCI 5413 — Ethical Hacking (CU); Network security, nmap, netcat, Kali Linux, Buffer overruns, Format string vulnerabilities, Race Conditions, Web Security, SQL Injections, XSS, CSRF, Wireless.

CSCI 6268 — Foundations of Computer and Network Security (CU); an introductory course covering basic cryptography, cryptographic protocols, attacks, and principles, as well as core network security attack and defense.

CSCI 7000 — Cryptography Seminar (CU); A graduate course introducing basic cryptographic definitions and then making some forays to the research front.

CSCI 7000 — Cryptanalysis Seminar (CU); A graduate course introducing students to cryptanalysis. Differential and linear cryptanalysis, square attack, RSA basics, factoring, protocol errors, lattices, Coppersmith's algorithm.

CSCI 7000 — Quantum Computing (CU); Introduction to quantum circuits, number theory, Shor's Algorithm, Grover's Algorithm.

Graduate Students

Rakhi Motwani, M.S., Completed: Spring 2002.

Scott Fritzinger, M.S., Completed: Summer 2002.

Hector Urtubia, M.S., Completed: Spring 2003.

Hiba Fayoumi, M.S., Completed: Summer 2004.

Mary Hedges, M.S., Completed: Spring 2007.

Joeseph Dunn, Ph.D., co-advisor with John Bennett, Completed: Summer 2007.

Martin Cochran, Ph.D., Completed: Spring 2008.

Jared Nishikawa, Ph.D., Completed: Spring 2016.

Undergraduate Students Troy Trimble, University of California at San Diego. REU Student, Summer 2003.
Gagan Sekhon, California State University at Hayward. REU Student, Summer 2003.
Ryan Gardner, University of Colorado at Boulder. REU Student, Summer 2004.
Trevor Highland, University of Texas at Austin. REU Student, Summer 2005.

**External
Service**

Secretary, International Association for Cryptologic Research, 2005–2007.
Program Committee, ACNS 2015.
Program Committee, CT-RSA 2015.
Program Committee, FSE 2014.
Program Committee, CT-RSA 2014.
Program Committee, CT-RSA 2013.
Program Committee, Eurocrypt 2012.
Program Committee, FSE 2011.
Program Committee, PKC 2011.
Program Committee, CANS 2010.
General Chair, CRYPTO 2009.
Program Committee, CRYPTO 2008.
Program Committee, ACNS 2008.
Program Committee, RSA-CT 2007.
Program Committee, ISC 2007.
Program Committee, ACNS 2007.
Program Committee, ACM CCS 2006.
Program Committee, CANS 2006.
Program Committee, ICISC 2006.
Program Committee, SECURITY 2006.
Program Committee, ACSAC 2006.
Program Committee, CRYPTO 2005.
Program Committee, SAC 2005.
Program Committee, ICISC 2005.
Program Committee, CANS 2005.
Program Committee, IEEE SISW 2005.
Program Committee, CRYPTO 2004.
Program Committee, EUROCRYPT 2004.
Program Committee, RSA-CT 2003.
NSF CISE Panelist, 2001, 2003, 2005, 2006, 2007, 2009.
Referee for Journal of Cryptography, 1999-2006.
Referee for Software: Practice and Experience, 2005.
Referee for IEEE Communications Magazine, 2005.
Referee for IEEE Transactions on Circuits and Systems I, 2005.
Referee for IEEE Computer, 2005.
Referee for Journal of Computer Security, 2004.
Referee for IEEE Transactions on Information Theory, 2003.
Referee for IEEE Transactions on Computers, 2002.
Reviewer for CRYPTO 1999–2002, SODA 1998, SPAA 2002, Asiacrypt 2004, EURO-
CRYPT 2006.

Developed **CryptoStats** web site: an application which tracks publication rates by year, by author, by conference in the two main cryptography conferences. It was heavily used in my community (on average 240 hits per month), 2003-2009.

ACM Programming Contest problem composer, 2003–2007.
ACM Programming Contest site administrator, 2005.

Graduate Student Mixer organizer, CRYPTO 2005.

**Internal
Service**

Chair, Computing Committee, 2010–2011, 2012–2015.

Chair, Space Committee, 2012–2015.

Liaison, Casey Feldman Foundation, 2010–2015.

Member, Departmental Executive Committee, 2003–2005, 2007–2009, 2013–2015.

Member, Executive Committee, Computer and Communications Security Center, 2003–2006.

Member, Departmental Search Committee, 2003–2006, 2012–2014.

Chair, Departmental Search Committee, 2008–2009.

Member, Graduate Committee, 2005–2006.

Developed departmental voting software, now used for all departmental votes and college votes.