UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

MICROSOFT CORPORATION,
Petitioner,

v.

QOMPLX LLC,
Patent Owner.

IPR2026-00325
U.S. Patent No. 12,301,627
Issued: May 13, 2025
Application No. 18/892,283
Filed: September 20, 2024

Title: CORRELATING NETWORK EVENT ANOMALIES USING ACTIVE AND PASSIVE EXTERNAL RECONNAISSANCE TO IDENTIFY ATTACK INFORMATION

DECLARATION OF DR. JOHN R. BLACK, JR.

PAN Exhibit 1024

TABLE OF CONTENTS

Page

I. INTRODUCTION AND ENGAGEMENT 5

II. BACKGROUND AND QUALIFICATIONS.....5

III. MATERIALS CONSIDERED AND INFORMATION RELIED UPON REGARDING '627 PATENT 10

IV. UNDERSTANDING OF PATENT LAW 12

V. LEVEL OF SKILL IN THE ART.....16

A. Person Of Ordinary Skill In The Art.....16

B. State Of The Art 17

1. It Was Well Known To Use Graphs To Represent Computer Networks.....17

2. It Was Well Known To Use Graphs To Identify Anomalies in Computer Networks.....18

3. It Was Well Known To Use Graphs To Predict And Analyze Attack Paths 18

VI. THE '627 PATENT.....19

VII. CLAIM CONSTRUCTION 23

VIII. GROUND 1: CLAIMS 1-3, 5-6, 11-14, 16-18, 22, 24, AND 27-28 ARE OBVIOUS OVER STOKES AND CRABTREE 23

A. Stokes 23

B. Crabtree 26

C. Combining Stokes And Crabtree.....28

D. Claim 1 36

1. Element [1.1] 36

2. Element [1.2].....37

3. Element [1.3].....39
4. Element [1.4].....43
5. Element [1.5].....44
6. Element [1.6].....44
7. Element [1.7].....47
8. Element [1.8].....51
9. Element [1.9].....55
10. Element [1.10].....59
11. Element [1.11].....63
12. Element [1.12].....67
E. Claim 2 70
F. Claim 3 74
G. Claim 5 77
H. Claim 6 80
I. Claim 11 82
J. Claim 12 84
K. Claim 13 85
L. Claim 14 88
M. Claim 16 90
N. Claim 17 90
O. Claim 18 91
1. Elements [18.1]-[18.5], [18.7], and [18.9].....91
2. Element [18.6].....92
3. Element [18.8].....95
4. Element [18.10].....95

5. Element [18.11] 96

P. Claim 22 101

Q. Claim 24 105

R. Claim 27 106

S. Claim 28 109

IX. GROUND 2: CLAIM 3 IS OBVIOUS OVER STOKES IN VIEW OF CRABTREE AND SEKAR 111

A. Sekar (EX1006) 111

B. Combining Stokes, Crabtree, And Sekar 112

C. Claim 3 114

I, John R. Black, Jr., do hereby declare as follows:

I. INTRODUCTION AND ENGAGEMENT

1. I have been retained as an independent expert on behalf of Microsoft Corporation in connection with the above-captioned Petition for *Inter Partes* Review ("IPR") to provide my analyses and opinions on certain technical issues related to U.S. Patent No. 12,301,627 (hereinafter "the '627 patent").

2. I am being compensated at my usual and customary rate for the time I spent in connection with this IPR. My compensation is not affected by the outcome of this IPR.

3. Specifically, I have been asked to provide my opinions regarding whether claims 1-3, 5-6, 11-14, 16-18, 22, 24, and 27-28 (each a "Challenged Claim" and collectively the "Challenged Claims") of the '627 patent would have been obvious to a person having ordinary skill in the art ("POSITA") as of January 30, 2020. It is my opinion that each Challenged Claim would have been obvious to a POSITA after reviewing the prior art discussed herein.

II. BACKGROUND AND QUALIFICATIONS

4. My credentials are set forth in full detail in my curriculum vitae (Appendix A).

5. I hold a Bachelor of Science from the California State University at Hayward (now "California State University, East Bay") in Mathematics and

Computer Science, conferred in 1988. After my B.S. degree in 1988, I worked for 6 years for Ingres Corp. as a software developer. I received a Doctor of Philosophy in Computer Science from the University of California at Davis in 2000. My Ph.D. thesis is titled "Message Authentication Codes" (MAC), which is an area of computer security related to the authentication of messages over a network.

6. After receiving my Ph.D., I became an Assistant Professor of Computer Science at the University of Nevada, Reno, and, in 2002, I was appointed as an Assistant Professor of Computer Science at my current institution, which is the University of Colorado at Boulder. I was awarded tenure in 2008. My area of specialty is cryptography and security, including the security of computer systems and networks.

7. I have taught more than 40 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 25 journal and conference papers related to computer science. I have been involved in the fields of computer science and networking for over 35 years.

8. In addition to these activities, I have authored chapters related to cryptography and authenticated encryption for two books and have served as a referee and on the programs committee for numerous journals and conferences for

the last two decades. I also served as an advisor to numerous PhD and Masters students. In 2002, I was the recipient of the NSF Career Award and have been the recipient of over $1 million in grant money from the NSF for my research in cryptography.

9. I have published 18 papers in conferences, workshops, and journals, as well as several book chapters on topics related to my research. I have also been invited to speak at several conferences on my research. I am the recipient of an NSF CAREER award for my research as well as numerous awards for teaching. Additionally, I have worked on various industry projects as a consultant, including work involving video streaming on the server side, and on an Android-based music app called “Yonder” that streams audio content from a remote server.

10. My academic work focuses primarily on the field of computer network security. I am primarily interested in how to make networks and computers more secure. To this end, I have studied cryptography (the encoding and decoding of information) as well as ad hoc methods for attacking computer systems and for providing defenses against those attacks.

11. Some of my work has been adopted by industry groups and integrated into commercial systems. An authentication algorithm I helped develop is now used in enterprise Wi-Fi and in tap-to-pay credit card systems via chip-based credit cards or NFC-enabled smartphones.

12. My interest in computer networks began around the age of 12 when I began to learn about the only network I had ready access to: the telephone network. Friends and I started a club and began reading Bell Systems Manuals. Shortly afterwards, I taught myself to program in BASIC so that I could access the Bulletin Board Systems that were a precursor to the modern Internet.

13. Before I entered college, I began to work on IBM PCs with MS-DOS and Novell networking solutions to small businesses, which represented my first exposure to a real network, although this, again, predates mainstream use of the Internet, which was still in its nascent stages.

14. While attending CSU Hayward from 1984 to 1988, I gained my first exposure to the Internet, which was still an obscurity to most people then. After college, I joined Ingres Corp (then called "Relational Technology Inc") as a software developer. My primary responsibilities related to networking products. The company made and sold SQL database products including an SQL engine that I worked on. In addition, my work at Ingres Corp. exposed me to the early version of the modern Internet and the World Wide Web for the first time.

15. I left Ingres in 1994 and in 1995 started working toward a Ph.D. in Computer Science. My initial interest was in the area of graph algorithms and hashing algorithms, which later led to my main research focus in the area of cryptography and security. I graduated from UC Davis with a Ph.D. in Computer

Science in 2000 and took a job as an assistant professor at the University of Nevada at Reno. In Fall semester 2001, I taught the networking class there, which included coverage of Ethernet, interior gateway protocols, exterior gateway protocols, ARP, DHCP/BOOTP, IP, UDP, TCP, HTTP, SMTP and other protocols. My specialization in graduate school was authentication, which was also the focus of my Ph.D. dissertation.

16. In 2002, I moved to the University of Colorado at Boulder to take on a new faculty job. In Fall 2002 I co-designed and co-taught a new course called "Foundations of Computer and Network Security" which included descriptions of security issues around Ethernet switches, routers, DNS, BGP, TCP hijacking, HTTP, SSL, and other protocols; I have taught this class 7 times subsequently, including modern topics such as wireless networking and Internet of Things, etc.

17. In 2011, I became chair of the Computer Science Department's Computing Committee. As part of my duties, I built a new data center in the basement of the engineering center.

18. I have worked for a consulting company at times, writing software on a contract basis. Although most of those projects are covered by NDAs, there were several that involved wide-scale networks in order to move large datasets (including genomic data and video data). In one case, I was hired to examine network analysis

logs to troubleshoot an Israeli video distribution center's performance problems when delivering streaming video.

19. From 2016 to 2017 I took a leave of absence from CU Boulder to start a company called "SecureSet" in Denver Colorado. The objective of SecureSet is to take reasonably proficient technical people and turn them into computer and network security specialists in 5 months of intensive training. This training included core concepts from cryptography and security, including authentication using two-factor authentication and multi-factor authentication, as well as attacks against those login methods such as SIMjacking.

20. In 2018 I returned to the University of Colorado to resume my teaching and research duties.

III. MATERIALS CONSIDERED AND INFORMATION RELIED UPON REGARDING '627 PATENT

21. In preparing this declaration and arriving at my opinions, I have reviewed the following materials bearing Exhibit Nos. that I understand are being referenced in the IPR where my declaration may be referenced:

LIST OF EXHIBITS

No.	Description
1001	U.S. Patent No. 12,301,627 ("**'627 patent**")
1002	File History of U.S. Patent No. 12,301,627
1004	U.S. Patent App. Pub. No. 2018/0367548 ("**Stokes**")

No.	Description
1005	U.S. Patent App. Pub. No. 2018/0219919 ("**Crabtree**")
1006	U.S. Patent App. Pub. No. 2020/0059481 ("**Sekar**")
1007	U.S. Patent App. Pub. No. 2020/0296137 A1
1008	U.S. Patent App. Pub. No. 2022/0014560 A1
1009	Leman Akoglu, *et al.*, *Graph Based Anomaly Detection and Description: a Survey*, Data Mining and Knowledge Discovery, Vol. 29, Issue 3 (Springer 2014)
1010	Steven Noel & Sushil Jajodia, *Advanced Vulnerability Analysis and Intrusion Detection Through Predictive Attack Graphs*, Center for Secure Info. Sys., George Mason Univ. (2009) available at https://csis.gmu.edu/noel/pubs/2009_c4i.pdf
1011	*Managing Cyber Threats* (Vipin Kumar, et al. eds, Springer Science+Business Media, Inc., 2005)
1012	William Eberle & Lawrence Holder, *Incremental Anomaly Detection in Graphs*, 2013 IEEE 13th Int'l Conference on Data Mining Workshops (2013)
1013	Plaintiff Qomplx's Preliminary Infringement Contentions, *Qomplx LLC v. Microsoft Corp.*, Case No. 1:25-cv-01383-ADA (W.D. Tex.) (Nov. 24, 2025)
1014	Plaintiff Qomplx LLC's Preliminary Disclosure of Proposed Claim Constructions, *Qomplx LLC v. Microsoft Corp.*, Case No. 1:25-cv-01383-ADA (W.D. Tex.) (Mar. 2, 2026)
1015	Defendant's Preliminary Claim Constructions, *Qomplx LLC v. Microsoft Corp.*, Case No. 1:25-cv-01383-ADA (W.D. Tex.) (Mar. 2, 2026)
1016	Listing and Comparison of '627 Patent Challenged Claims
1017	U.S. Patent No. 9,225,730
1018	Microsoft Computer Dictionary, 5th Ed. (2002)
1019	Redline Comparison of Crabtree to '627 Patent Specification

No.	Description
1020	Complaint For Patent Infringement, *Qomplx LLC v. Microsoft Corp.*, Case No. 1:25-cv-01383-ADA (W.D. Tex. Aug. 28, 2025)
1021	U.S. Patent App. Pub. No. 2019/0081876A1
1022	Seth Webster, *et al.*, *Experience Using Active and Passive Mapping for Network Situational Awareness*, IEEE 5th Int'l Symposium on Network Computing and Applications (2006)
1023	Genevieve Bartlett, *et al.*, *Understanding Passive and Active Service Discovery*, IMC '07 (2007)

IV. UNDERSTANDING OF PATENT LAW

22. I am not an attorney. For this declaration, I have been informed about certain aspects of the law that are relevant to my opinions. The Petitioner's attorneys provided me with my understanding of the law.

23. I understand that when considering the scope of the claims of a patent, the patent claim terms should generally be given the ordinary meaning that the terms would have to a person of ordinary skill in the art in question after reading the patent as of the earliest claimed priority date.

24. I understand that the person of ordinary skill in the art is deemed to read the claim term not only in the context of the particular claim in which the term appears but in the context of the entire patent, including the other claims and the application specification as filed. I further understand that the principal considerations regarding the claims' scope and meaning are the claims' plain language (including the surrounding claim language and context), the patent

specification, and the patent's prosecution history in the Patent Office. I understand that while a claim is to be read in light of the specification, one must generally avoid importing limitations into the claim from the specification. I am also informed that the prosecution history can often inform the meaning of the claim by demonstrating how the inventor understood the invention and whether the inventor limited the invention in the course of prosecution, making the claim scope narrower than it would otherwise be. I applied these understandings when considering the scope and meaning of the claims of the '627 patent.

25. I further understand that a claim is unpatentable if it would have been obvious. Obviousness of a claim requires that the claim would have been obvious from the perspective of a POSITA at the time the alleged invention was made. I understand that a claim could have been obvious from a single prior art reference or from a combination of two or more prior art references. I understand that an obviousness analysis requires understanding the scope and content of the prior art, any differences between the alleged invention and the prior art, and the level of ordinary skill in evaluating the pertinent art.

26. I further understand that a claim would have been obvious if it unites old elements with no change to their respective functions, or merely substitutes one element for another known in the field, and that combination yields predictable results. While it may be helpful to identify a reason for this combination, I

understand there is no strict requirement to find an express teaching, suggestion, or motivation to combine within the references. When a product is available, design incentives and other market forces can prompt variations of it, either in the same field or a different one. If a POSITA can implement a predictable variation, obviousness likely prevents its patentability. For the same reason, if a technique has been used to improve one device and a POSITA would recognize that it would improve similar devices in the same way, using the technique would have been obvious. I understand that a claim would have been obvious if common sense directs one of ordinary skill, without hindsight bias caused by knowledge of the Challenged Claims, to combine multiple prior art references or add missing features to reproduce the alleged invention recited in the claims.

27. I further understand that certain factors may support or rebut the obviousness of a claim. I understand that such secondary considerations include, among other things, commercial success of the patented invention, skepticism of those having ordinary skill in the art at the time of invention, unexpected results of the invention, any long-felt but unsolved need in the art that was satisfied by the alleged invention, the failure of others to make the alleged invention, praise of the alleged invention by those having ordinary skill or expertise in the art, and copying of the alleged invention by others in the field. I understand that there must be a nexus—a connection—between any such secondary considerations and the alleged

invention. I also understand that contemporaneous and independent invention by others is a secondary consideration tending to show obviousness.

28. I am not aware of any allegations by the named inventors of the ’627 patent or any assignee of the ’627 patent that any secondary considerations would rebut the obviousness of any Challenged Claim of the ’627 patent.

29. I understand that in considering obviousness, it is important not to determine obviousness using the benefit of hindsight derived from the descriptions found in the patent being considered.

30. I understand that Petitioner has the burden of proving obviousness by a preponderance of evidence, which means that the claims are more likely than not obvious.

31. I understand that other challenges to the patentability of a patent, including patent ineligibility, enablement, written description, and definiteness or clarity of claim language, cannot be raised in IPR proceedings before the Board.

32. I understand, however, that if Patent Owner wishes to rely on the filing date of an earlier ancestor application, that ancestor application must provide a written description of the claimed invention that reasonably conveys to POSITAs that the inventor(s) had possession of the full scope of the claimed subject matter as of the filing date of that earlier ancestor application. Thus, I understand that a patent claim is not entitled to claim priority to an earlier ancestor application unless the

ancestor application provides adequate full scope written description support for the claimed invention.

33. I further understand that, to meet the written description requirement, it is not enough that the claims would have been obvious in view of the disclosure. I also understand that the written description requirement requires that the disclosure of the claimed subject matter must be: (1) express or necessarily present, (2) complete, and (3) unambiguous.

34. The analysis in this declaration is in accordance with the above-stated legal principles.

V. LEVEL OF SKILL IN THE ART

A. Person Of Ordinary Skill In The Art

35. For purposes of this declaration I have assumed an effective filing date/priority date for the Challenged Claims of the ’627 patent of January 30, 2020. To the extent Patent Owner asserts that any Challenged Claim is entitled to an earlier priority date, I reserve the right to supplement my testimony to address the Patent Owner’s arguments and evidence.

36. The person of ordinary skill in the art for purposes of the ’627 patent in January 2020 (“POSITA”) would have had a bachelor’s degree in computer science, or equivalent, with two to four years’ experience designing, implementing, or otherwise working with cybersecurity technologies. More education, e.g., a master’s

degree or Ph.D., would compensate for less work experience and vice versa. For purposes of this Declaration, the knowledge and skill of a POSITA would not change if the relevant priority date were April 22, 2021. The state of the art, immediately below, identifies certain technologies within the knowledge and skill of a POSITA.

B. State Of The Art

1. It Was Well Known To Use Graphs To Represent Computer Networks

37. Graph structures have long been used to represent and analyze data, including entities and their behaviors. *See* EX1012, 522 (2013 IEEE publication explaining that "[m]uch work has been done using *graph*-based representations of data. Using *vertices* to represent entities such as people, places and things, and *edges* to represent the relationships between the entities…, allows for a much richer expression of data than is present in the standard textual or tabular representation of information.") (emphasis original). Such techniques have been applied to computer networks since more than a decade before the '627 patent's filing date. *Id*., 521 (explaining that "in the case of computer network traffic, a graph representation of the traffic might consist of nodes representing computers and edges representing communications between the corresponding computers").

2. It Was Well Known To Use Graphs To Identify Anomalies in Computer Networks

38. These types of graphical network representations have been commonly used to identify anomalies in a computer network, such as by identifying deviations from normative behavior patterns. EX1012, 521 ("In this work, we describe methods for graph-based anomaly detection via graph partitioning and windowing, and demonstrate their ability to efficiently detect anomalies in data represented as a graph."), ("In a previous approach to graph-based anomaly detection…, we used a compression based measure to find normative patterns, and then analyzed the close matches to the normative patterns to determine if they meet the [] definition of an anomaly."). *See also* EX1009, 673-74 (2014 survey of graph-based methods for identifying "[a]nomalies in computer networks").

3. It Was Well Known To Use Graphs To Predict And Analyze Attack Paths

39. Graph-based techniques have been used to predict, represent, and analyze attack paths in computer networks since more than a decade before the '627 patent's filing date. *See* EX1010, 1 (2009 George Mason University publication explaining that "[a]ttack graphs map paths of vulnerability, showing how attackers can incrementally penetrate a network," where the authors "match the network configuration to known attacker exploits, simulating attack penetration through the network and predicting attack paths leading to compromise of mission-critical

assets," thereby obtaining a "set of all possible attack paths, organized as an attack graph"). *See also* EX1011, 47-48 (2005 book chapter entitled "Topological Analysis of Network Attack Vulnerability," describing a tool that "generates a graph of dependencies among exploits that represents all possible attack paths" and "discovers attack paths (sequences of exploits) leading to specific network targets").

VI. THE '627 PATENT

40. The '627 patent issued May 13, 2025, from U.S. Patent Application No. 18/892,283 (the "'283 application"), filed September 20, 2024, which is a continuation of Application No. 17/237,346 (the "'346 application"), filed April 22, 2021. The '346 application is a continuation-in-part of Application No. 16/777,270 (the "'270 application"), filed January 30, 2020. EX1001, Cover, 1:9-18. The '627 patent alleges priority through a series of continuation and continuation-in-part applications to an application filed October 28, 2015. However, the Challenged Claims of the '627 patent are not entitled to priority any earlier than January 30, 2020.

41. The '627 patent relates to "a system and method for… identify[ing] attack information, that can identify attack patterns and points of origin based on observed anomalies and their relationships to other observed network events." EX1001, 2:52-57. It describes "a cyber-physical system graph (CPG)" comprising "a visualization of hierarchies and relationships between devices and resources in a

security infrastructure, contextualizing security information with physical device relationships that are easily understandable for security personnel and users." *Id*., 23:55-61. The CPG "compris[es] nodes representing entities associated with the organization and edges representing relationships between entities associated with the organization." *Id*., 3:10-21.

42. The '627 patent also discloses receiving time-series information. Specifically, time-series data "may be received continuously, passively collecting events and monitoring activity over time while feeding [] collected information … for use in producing time-series graphs [] of states and changes over time", thereby "providing live monitoring" of the state of the network. *Id.*, 23:40-52. The '627 patent further discloses using that information to produce "a graph visualization of users, servers, devices, and other resources correlating physical relationships … with logical relationships (such as access privileges or database connections), to produce a meaningful and contextualized visualization of a security infrastructure that reflects the current state of the internal relationships present in the infrastructure." *Id.*, 23:34-24:13. The '627 patent further describes "updating CPG information as changes occur and events are logged." *Id.*, 24:2-5; *see also id.*, 25:22-39 (explaining that when "a new connection is detected [] a CPG may be updated [] with the new connection information").

43. The ’627 patent also discloses that the CPG includes information used to evaluate and identify security risks in the network. Specifically, the CPG may contain “behavior analytics information” concerning behavior patterns in the network that may be analyzed to recognize deviations from established behavior patterns. *Id*., 22:43-23:14, 23:62-65. The CPG may also contain “impact assessment score[s]” reflecting access capabilities of user accounts in a network, “representing the potential risk should that account be compromised.” *Id.*, 23:14-27, 23:65-24:2.

44. The ’627 patent further explains that “[p]ossible attack paths may be analyzed using the [CPG].” *Id*., 28:46-50. Figure 22, reproduced below, shows an example CPG used to analyze and rank three exemplary attack paths:

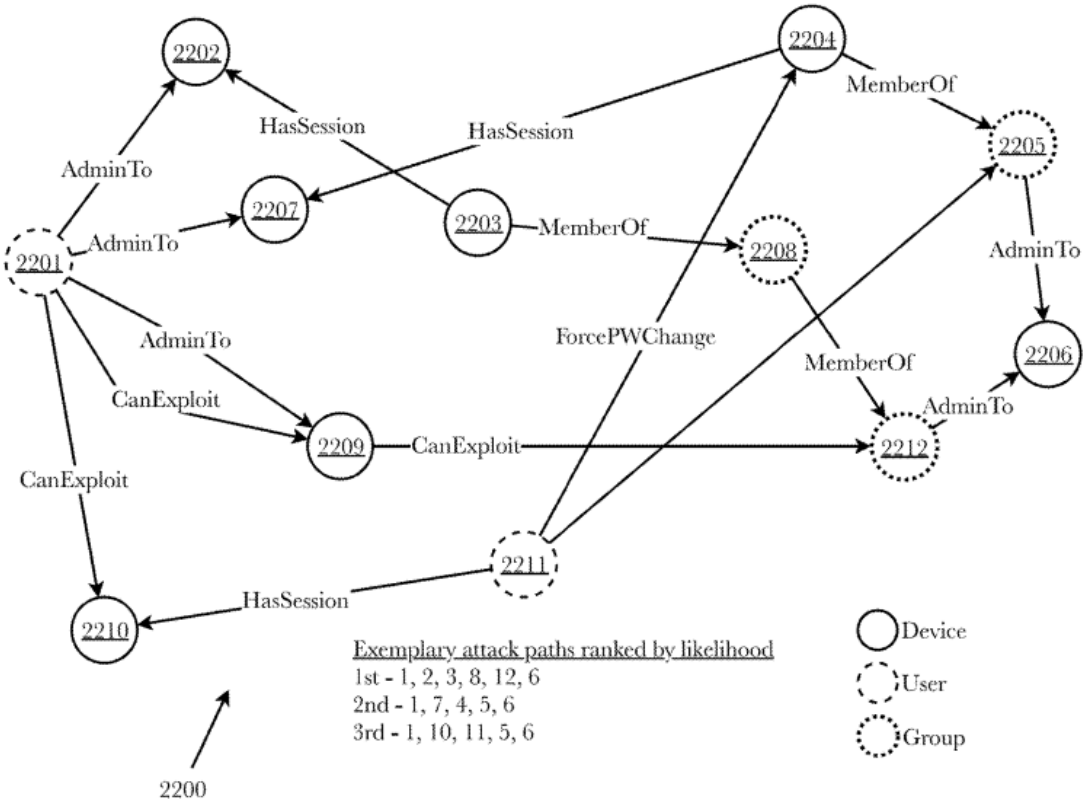


Fig. 22

Id., FIG. 22.

45. The example CPG 2200 shown in Figure 22 "contains 12 nodes (vertices) comprising: seven computers and devices designated by solid circles…, two users designated by dashed-line circles…, and three functional groups designated by dotted-line circles…," where "[t]he edges (lines) between the nodes indicate relationships between the nodes…." *Id.*, 28:34-65. The '627 patent explains that "[p]ossible attack paths may be analyzed using the [CPG] by running graph analysis algorithms…." *Id.* As a result of such analysis, "several exemplary attack paths are ranked by likelihood." *Id.*

VII. CLAIM CONSTRUCTION

46. Unless otherwise expressly discussed below, I apply the plain and ordinary meaning of all claim terms. I understand that in the district court litigation, neither Petitioner nor Patent Owner have proposed a construction for any claim term and have agreed that "account" be given its "plain and ordinary meaning." EX1014, 4; EX1015, 2.

VIII. GROUND 1: CLAIMS 1-3, 5-6, 11-14, 16-18, 22, 24, AND 27-28 ARE OBVIOUS OVER STOKES AND CRABTREE

47. As explained below, claims 1-3, 5-6, 11-14, 16-18, 22, 24, and 27-28 are obvious over Stokes and Crabtree.

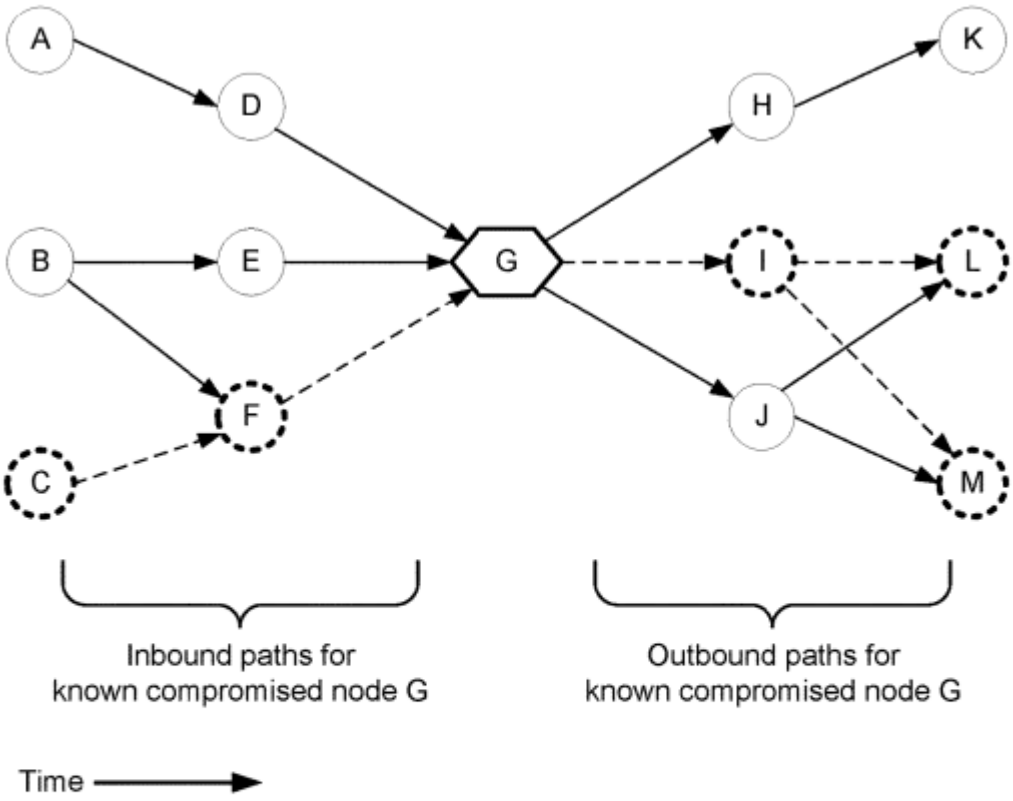
A. Stokes

48. Stokes, a U.S. patent application, was published December 20, 2018, and is prior art under at least 35 U.S.C. § 102(a)(1). EX1004, Cover. Stokes was not cited during prosecution of and is not listed on the face of the '627 patent.

49. Stokes describes using a graph to detect malicious activity in computer networks. It does so, for example, by identifying potentially malicious (or "rare") paths within a large graph of benign network connections and determining which of the rare paths to prioritize for further investigation. EX1004, [0003], [0005], [0006], [0044]. Stokes frames these teachings in two scenarios: (i) *forensic analysis*, which starts from at least one known compromised computer or user account and identifies inbound/outbound lateral movement paths; and (ii) *general detection*, which seeks

previously unknown compromised nodes using corroborating detections (e.g., remote file execution) combined with rare-path scoring. *Id.*, [0005], [0028], [0029], [0048]. Stokes further explains that events may be processed "in real-time" and that the system "may be implemented … in a streaming environment." *Id.*, [0023], [0038].

50. The Stokes system constructs a network connection graph in which computers and user accounts are modeled as nodes, and computer connections and user logon events are modeled as directed edges. EX1004, [0024], [0025], [0037]-[0038]. An example network connection graph "useful for describing forensic analysis with respect to tracking malicious lateral movement across a computer network" is shown in Figure 3 of Stokes (reproduced below). *Id.*, FIG. 3, [0063].



300

FIG. 3

51. Stokes explains that "in FIG. 3, a known compromised node (denoted as nod G [sic]) is discovered and the task is to identify the unknown compromised computers and user accounts (dashed nodes C, F, I, L, and M) along the malicious access path (dashed lines)." *Id.*, [0066].

52. A "path-rate score module" assigns each directed edge of the graph a weight based on the connection's frequency over an X-day history and computes a

path-rate score for a path by multiplying the weights of its constituent edges. *Id.*, [0044], [0058]. Stokes also describes filtering/eliminating paths as part of this scoring/ranking process, including eliminating infeasible paths based on timestamp ordering and eliminating paths outside a threshold time window. *Id.*, [0045], [0060]. Stokes further contemplates that the graph may reflect additional event sources and may be "implemented or extended" using security events and "improve[d]" using network logon events and NTLM events. *Id.*, [0051].

53. A "ranking module" ranks paths in ascending order of path-rate score, and Stokes explains that the highest ranked paths (lowest scores) are the rarest and "more likely to correspond to malicious lateral movement." *Id*., [0047]. During *forensic analysis*, highly ranked paths around known compromised nodes may be investigated to identify additional compromised nodes (e.g., as in the example shown in Figure 3). *Id*., [0047], FIG. 3. *General detection* focuses on nodes that have both highly ranked rare paths and a remote file execution detection. *Id*.

B. Crabtree

54. Crabtree (EX1005), published on August 2, 2018, from an application filed November 27, 2017 (EX1005, Cover), is prior art under at least 35 U.S.C. § 102(a)(1) because the earliest effective filing date of the Challenged Claims is no earlier than January 30, 2020, which is more than a year later than Crabtree's August 2018 publication date.

55. Crabtree lists as inventors two of the ’627 patent’s three inventors, and Crabtree issued from U.S. Patent Application No. 15/823,363 (EX1005, Cover), which is an ancestor to the ’627 patent through several continuation-in-part applications (EX1001, 1:9-26).

56. Crabtree shares many of the same disclosures as the ’627 patent. *See* Section VI, *above*; EX1019 (redline comparison showing changes between Crabtree specification and ’627 patent specification). Like the ’627 patent, Crabtree discloses “a cyber-physical system graph (CPG)” that provides “a visualization of hierarchies and relationships between devices and resources in a security infrastructure, contextualizing security information with physical device relationships that are easily understandable for security personnel and users.” EX1005, [0076]. Crabtree also describes such a “graph” as a “representation of information and relationships, where each primary unit of information makes up a ‘node’ or ‘vertex’ of the graph and the relationship between two nodes makes up an edge of the graph.” *Id.*, [0044].

57. Also like the ’627 patent, Crabtree discloses receiving time-series information. Specifically, time-series data “may be received continuously, passively collecting events and monitoring activity over time while feeding [] collected information … for use in producing time-series graphs [] of states and changes over time”, thereby “providing live monitoring” of the state of the network. *Id.*, [0075]. Crabtree further discloses using that information to produce “a graph visualization

of users, servers, devices, and other resources correlating physical relationships … with logical relationships (such as access privileges or database connections), to produce a meaningful and contextualized visualization of a security infrastructure that reflects the current state of the internal relationships present in the infrastructure." *Id.*, [0076]. Crabtree further describes "updating CPG information as changes occur and events are logged." *Id.*; *see also id.*, [0081] (explaining that when "a new connection is detected [] a CPG may be updated [] with the new connection information").

58. Crabtree also discloses that the CPG includes information used to evaluate and identify security risks in the network. Specifically, the CPG may contain "behavior analytics information" concerning behavior patterns in the network that may be analyzed to recognize deviations from established behavior patterns. *Id.*, [0072], [0076]. The CPG may also contain "impact assessment score[s]" reflecting access capabilities of user accounts in a network, "representing the potential risk should that account be compromised." *Id.*, [0074], [0076].

C. Combining Stokes And Crabtree

59. POSITAs would have been motivated with a reasonable expectation of success to implement a Stokes-like system—with its graph-based attack-path identification and analysis—with Crabtree's teachings on updating a graph to include additional nodes and edges corresponding to entities and relationships after

receiving additional event data, and outputting results of graph analysis. A skilled artisan would have combined these respective teachings into an improved design (hereinafter "Stokes-Crabtree") that included benefits from each reference.

60. A POSITA would have been motivated to combine Stokes and Crabtree for several reasons. First, Stokes and Crabtree address closely related problems using similar graph-based cybersecurity analysis and provide complementary teachings. Stokes describes processing event-derived network-connection data using a graph in which nodes correspond to entities such as user accounts and computers and edges correspond to relationships among those entities, and then traversing, scoring, and ranking paths in that graph to identify suspicious lateral movement or attack paths. *See* Section VIII.A, *above*; EX1004, [0037]-[0040], [0047]-[0048], [0064]-[0071]. Crabtree likewise describes maintaining and updating a cyber-physical graph of users, servers, devices, and other resources and their relationships, based on time-series and event information, and presenting graph analysis results through reports, dashboards, and visualizations. *See* Section VIII.B, *above*; EX1005, [0063]-[0064], [0072], [0076], [0080]-[0081]. Thus, the teachings of the two references complement one another: Stokes provides graph-based attack-path identification techniques, while Crabtree provides techniques for updating and enriching graphs and reporting results of graph-based analysis.

61. Second, a POSITA implementing Stokes would have understood the benefit—if not imperative—of analyzing security issues for the most recent state of the network. Thus, a POSITA would have been motivated to implement Stokes's streaming environment teachings with the server continually updating the graph to add new nodes and edges as they were detected based on incoming event data. A POSITA would have understood that Stokes's attack-path identification approach benefits from the completeness and currency of the underlying graph due to the criticality in cybersecurity applications of identifying and responding to potential anomalies, intrusions, and attacks as quickly as possible. EX1021, [0003] (2019 patent publication explaining that in computer "security-related applications…, *the ability to identify* [] *anomalous data… fairly quickly may be critical*") (emphasis added). For example, Stokes describes how network attacks exploit an initial vulnerability and then expand "to additional computers" (EX1004, [0002]), which illustrates the importance of quickly detecting attacks. *See also id.*, [0019]. Crabtree expressly teaches updating CPG information "as changes occur and events are logged" and using received time-series and event information to produce graph representations of users, servers, devices, and other resources and their relationships. EX1005, [0072], [0076]. Crabtree also confirms the benefit of continuously updating a network graph, explaining that updating "[w]hen a new connection is detected," allows for identification of "potential risk" and can lead to "quick handling" of any

risky connections. EX1005, [0081]. Crabtree further describes the benefits of these updates, as they "continuously reflect the current state of the organization, reflecting any recent changes, newly-discovered or announced vulnerabilities, software or hardware updates, newly-added or removed devices or services" EX1005, [0062]. Crabtree also notes how rapid detection can beneficially "alert an enterprise of ongoing cyberattack early in the attack cycle, possibly thwarting it but at least mitigating the damage." EX1005, [0063]. Accordingly, a POSITA would have recognized that a Stokes-like system benefited from Crabtree's teachings on graph-updating and reporting because they provided a graph that promptly reflects newly observed entities and relationships, thus allowing more prompt identification of suspicious paths.

62. Third, combining Crabtree with Stokes would have predictably improved the results of Stokes's attack-path identification system. Stokes seeks to identify suspicious or malicious paths by analyzing relationships among entities in a network-connection graph. EX1004, [0037]-[0040], [0047]-[0048], [0064]-[0071]. Crabtree teaches updating graphs with additional entity and relationship information and further teaches presenting graph analysis results in reports, dashboards, alerts, and visualizations. EX1005, [0063]-[0064], [0076], [0080]-[0081]. Applying Crabtree's graph updating and reporting techniques to Stokes would enable more informative and useful attack-path analysis, because suspicious paths could be

identified from a graph that more fully captures current entities and relationships and the resulting analysis could be communicated more effectively to security personnel.

63. Fourth, the teachings of Crabtree follow the same core principles as Stokes. Both references model a computer network environment as a graph of nodes (representing entities) and edges (representing relationships); both use event-derived information to update or analyze that graph; and both are directed to identifying and investigating suspicious or attack-related activity. EX1004, [0024], [0025], [0037]-[0038]; EX1005, [0076]. Therefore, a POSITA seeking to improve Stokes's attack-path identification would naturally have looked to Crabtree's teachings regarding graph updating and reporting of graph analysis results, and would have found those teachings directly applicable to Stokes's attack path identification techniques.

64. Fifth, a POSITA would have been motivated to apply Crabtree's teachings regarding performing network reconnaissance using "both passive, semi-passive and active scanning and reconnaissance" (EX1005, [0057]) in Stokes's system because Stokes teaches that "[o]ther types of detectors may be used, alone or in conjunction with the remote file execution detector," to provide corroborating evidence of anomalous behavior involving particular entities. EX1004, [0024]. A POSITA would have understood Crabtree's active reconnaissance and passive reconnaissance techniques as different ways of obtaining information about network entities, the results of which could be used in a Stokes-like system alongside results

from other detectors, and that incorporating those results would have predictably improved identification of relevant entities and relationships for graph analysis. Further, a POSITA would have understood the respective benefits and drawbacks of active and passive reconnaissance techniques for detection of network entities, and that both passive and active reconnaissance techniques could be performed together to maximize coverage and obtain the benefits of both techniques. EX1022, 19 (2006 IEEE publication explaining that "[a]ctive scanning has the benefit of proactively contacting all hosts on a network to determine their status,"), 26 (explaining that "passive scanning is an important part of network situational awareness because it provides completely new information about services not seen by the active scans"); 27 (explaining that "relying on passive and active mapping together is better than relying on only active mapping alone"); EX1023, 57 (2007 Internet Measurement Conference publication explaining that "passive monitoring is well suited for quickly finding popular services" whereas "[a]ctive scanning is better suited to rapidly finding all servers"), 60 (explaining that "[i]n cases where completeness is key, a combination of both [active and passive] methods is advantageous"), 70 ("passive and active service discovery are complimentary methods for discovering services on a network"). Accordingly, depending on the circumstances, the Stokes-Crabtree system would monitor network entities using passive reconnaissance, active reconnaissance, or both passive and active reconnaissance together.

65. A POSITA would have had a reasonable expectation of success in combining Stokes and Crabtree. The two references operate in the same technical field of graph-based cybersecurity analysis and rely on compatible inputs, structures, and objectives. Stokes already uses event-derived graph relationships to identify suspicious paths (*e.g.,* EX1004, [0003], [0026], [0038]-[0039], [0063]-[0067]), and Crabtree likewise uses event and time-series information to update and analyze a graph of users, devices, servers, and other resources (*e.g.* EX1005, [0044], [0075]-[0076]). Because Crabtree's graph updating and reporting techniques would have been applied to the same type of graph-based cybersecurity system already disclosed in Stokes, a POSITA would have understood that incorporating Crabtree's teachings into Stokes would involve the predictable use of known techniques in a closely related system, with no technological incompatibility or undue experimentation required. A POSITA therefore would have reasonably expected that combining Crabtree's graph updating and reporting concepts with Stokes's attack-path identification framework would successfully yield a more current and informative graph-based cybersecurity system.

66. So implemented, the combined Stokes-Crabtree system includes the following features:

- An environment, including a server with memory storing instructions for carrying out attack path identification, as taught by

Stokes. EX1004, [0022], [0082]-[0083], FIG. 7. Wherein the server performs the following functions upon executing instructions stored in tangible media, as taught by Stokes (*id.*, [0030], [0081], [0084]-[0085], [0087]):

- Create a directed graph including nodes representing computer and user account entities and edges representing relationships between entities, and store that graph in memory on the server for analysis, as taught by Stokes. *Id.*, [0024]-[0026], [0030], [0037]-[0039], [0047], [0057]-[0067], [0072], [0081], FIGS. 1-3.
- Receive streaming data comprising time-stamped data about events relating to entities represented in the graph, as taught by Stokes (*id.,* [0023], [0038]-[0039]).
- Detect new entities and relationships in incoming event data and modify the graph by adding nodes and edges corresponding to those detected entities and relationships, as taught by Crabtree. EX1005, [0050], [0076], [0078]-[0079], [0081].
- Identify an attack path based on the modified graph, as taught by Stokes. EX1004, [0003], [0029], [0038]-[0039].

- Output results that identify entities involved in identified attack paths, as taught by Stokes (*id.*, [0007], [0027], [0047]), such as in the form of a report as taught by Crabtree (EX1005, [0076], [0081]-[0083], FIG. 11).

67. Further motivations to combine Stokes and Crabtree are discussed for individual claim elements below.

D. Claim 1

1. Element [1.1] [1]

[1.1] A computer system comprising:

68. Stokes-Crabtree satisfies the preamble. Stokes discloses a "server" (*computer system*) within its "exemplary environment … for tracking malicious lateral movement across a computer network." E.g., EX1004, [0021]. Stokes teaches implementing its server using a "computing device" such as that illustrated in FIG. 7. *Id*., [0022]; *see also id*., FIG. 7, [0082]; *see generally* [0079]-[0086] (describing "computing device environment").

[1] Reference numbers have been added throughout, in the format [claim#.element#].

2. Element [1.2]

[1.2] a hardware memory, wherein the computer system is configured to execute software instructions stored on nontransitory machine-readable storage media comprising software instructions that:

69. Stokes-Crabtree satisfies this element. Stokes teaches implementing its server using a computing device that includes "memory" (EX1004, [0082]) and "additional storage (removable and/or non-removable) including, but not limited to, magnetic or optical disks or tape" (*id.*, [0083]). *See also id.*, [0022].

70. Figure 7 shows the "memory 704," removable storage 708, and non-removable storage 710 of an exemplary computing device 700:

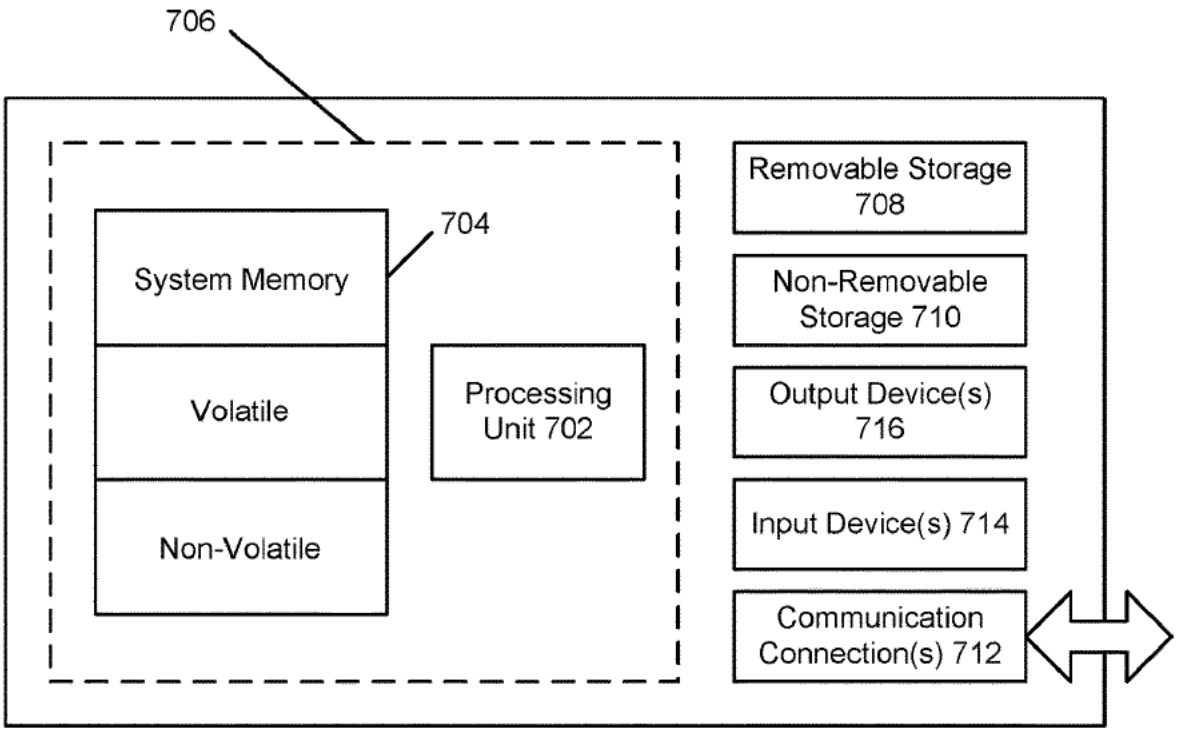


FIG. 7

Id., FIG. 7.

71. Stokes also discloses its server 130 comprising several different modules that carry out its functionality. *E.g.*, *id.*, [0030]. Stokes further discloses that its computing devices (such as its server) execute "computer-executable instructions, such as program modules" to "perform particular tasks or implement

particular abstract data types." *Id.*, [0081]. These "program modules and other data may be located in both local and remote computer storage media including memory storage devices." *Id.*; *see also id.*, [0084]-[0085]. Stokes teaches that all or part of its "methods and apparatus" "may take the form of program code (i.e., instructions) embodied in tangible media" such as any "machine-readable storage medium." *Id.*, [0087]. "[W]hen the program code is loaded into and executed by a machine, such as a computer, the machine becomes an apparatus for practicing the presently disclosed subject matter." *Id.* Stokes thus teaches configuring its server to execute software instructions stored on nontransitory machine-readable storage media comprising software instructions, to carry out its disclosed operations. As explained in the following paragraphs, this carries out the steps recited in the claim.

3. Element [1.3]

[1.3] store in the hardware memory a representation of a first graph, wherein the representation of the first graph comprises representations of a first plurality of nodes corresponding to a first plurality of entities and further comprises representations of a first plurality of edges, wherein the first graph is a directed graph,

72. Store in the hardware memory a representation of a first graph Stokes-Crabtree satisfies this element. Stokes discloses its server having "a network graph construction module 165." EX1004, [0030]; *see also*, *e.g.*, *id.*, [0025]-[0027]. The network graph construction module 165 "uses system and security events to construct a network connection graph 173." *Id.*, [0037]; *see also id.,* FIG. 1.

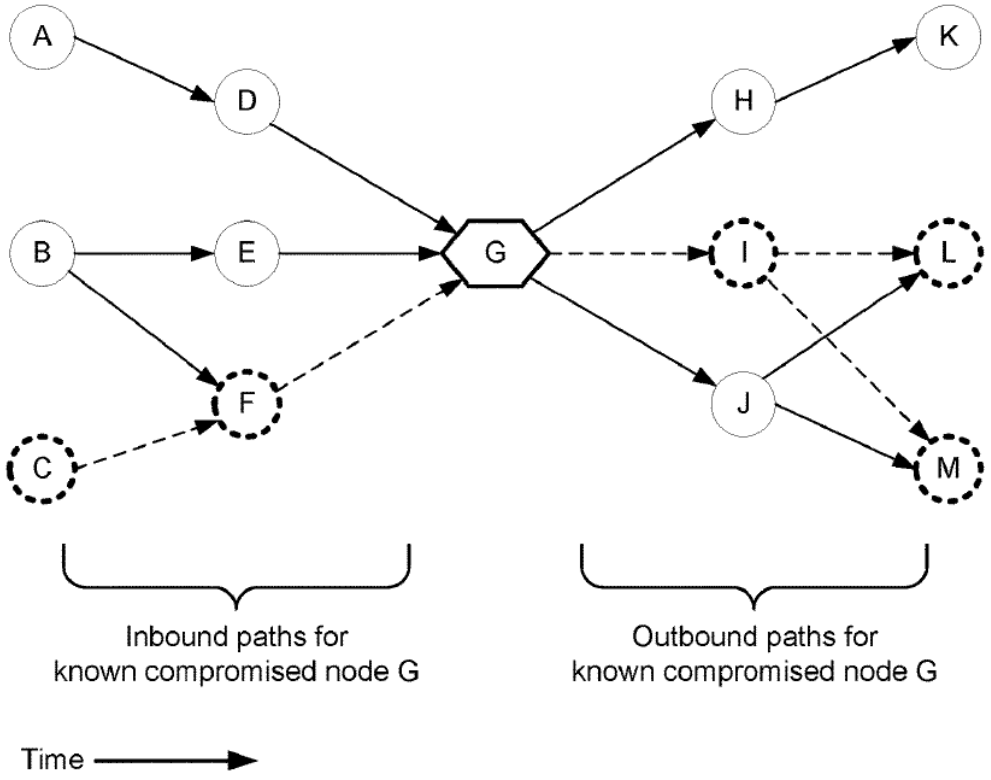
73. Stokes explains that such graphs can be represented mathematically as "G=(V,E)," "where V denotes the set of objects (v1, v2 , . . .), called nodes, and E is the set of connection relationships (e1, e2 , . . .) between pairs of nodes, called edges." *Id.*, [0026]. A POSITA would understand that in a server-based system like Stokes-Crabtree, data defining the graph, including data identifying the sets V and E, would be stored and maintained in hardware memory for use by the network graph construction module 165 and other server components.

74. Stokes further discloses that its server includes several other modules that interact with the graph, including "remote file execution detector 160, … a general detection module 170, a path-rate score module 175, an automatic account disabling module 177, a ranking module 180, and a forensic analysis module 185"). *Id.*, [0030].

75. Stokes discloses using the network connection graph 173 for subsequent server processing and operations. *See, e.g.*, EX1004, [0057]-[0062], FIG. 2, [0047], [0068]-[0071], FIG. 4, [0075]-[0078], FIG. 6. A POSITA would understand that the data representing the graph is stored in memory both during its creation and during subsequent operations involving the graph. *E.g.*, *id.*, [0049] (describing implementation having "datasets" with millions of nodes and edges). Stokes further discloses that its "memory" and other storage media stores information such as "data structures … or other data." EX1004, [0085]; *see also id.*,

[0081]. Because the network connection graph 173 is used in subsequent server processing a POSITA would have understood that it is maintained by the server 130 as stored data for use in those subsequent operations.

76. Wherein the representation of the first graph comprises representations of a first plurality of nodes corresponding to a first plurality of entities and further comprises representations of a first plurality of edges: Stokes discloses that "in [the] network connection graph 173, computers and user accounts are modeled as nodes, and computer-to-computer connections or user logon events are modeled as edges." *Id.*, [0025]; *see also id.*, [0024], [0026], and [0063]-[0067] (further describing graphs). Stokes's Figure 3 (reproduced below) shows an example network connection graph comprising a plurality of nodes (A-M) representing computer and user account entities, as well as a plurality of edges shown as arrows between the nodes:



300

FIG. 3

Id., FIG. 3.

77. Stokes shows the exemplary Figure 3 graph is a *directed graph*, with arrows indicating the direction of each edge. More generally, Stokes discloses using a *directed graph* with "order between the connection of a pair of nodes" and "directed edges." *Id.*, [0038]; *see also id.*, [0039] ("*directed* acyclic graphs (DAGs)") (emphasis added); *id.*, [0027] ("If orientations of edges are considered, i.e., order exists between the connection of each pair of nodes, such edges are directed edges.").

Id., FIG. 3; *see also id.*, [0038], [0063], FIG. 5, [0072]; *see also* EX1001, 7:67-8:4 (explaining that a "directed" graph is one where edges represent a relationship that flows from one node to another in one direction").

4. Element [1.4]

[1.4] wherein the first plurality of entities comprises a plurality of accounts and a plurality of resources, and

78. Stokes-Crabtree satisfies this element. In the Stokes network connection graph "computers and user accounts are modeled as nodes." *E.g.*, EX1004, [0025]; *see also id.*, [0006]-[0007], claim 1, claim 12. Stokes's "user accounts [] modeled as nodes" are the claimed plurality of accounts of the first plurality of entities. A POSITA would understand that the "computers" are resources on the Stokes network because they carry out computing functions.

79. Further regarding "resources," the '627 patent characterizes computing devices as resources, as it describes its graph visualizing "servers, devices, and other resources" and correlating other "resources" such as a "personal computer or smartphone." EX1001, 21:13-14, 21:33-35, 24:2-14. It also "represents the network devices comprising an organization's network infrastructure as nodes (also called vertices) in the graph." *Id.*, 28:15-23; *see also id.*, 28:34-40 ('627 patent example graph with nodes representing "seven computers"); *id.* FIG. 22; *id.*, 21:30-35.

80. Thus, the Stokes "computers … modeled as nodes" are the claimed plurality of resources of the first plurality of entities. And the "computers and user accounts" make up a first plurality of entities. EX1004, [0025].

5. Element [1.5]

[1.5] wherein each edge of the first plurality of edges corresponds to a respective relationship between a respective pair of entities;

81. Stokes-Crabtree satisfies this element. Stokes discloses that its graph "edges" model "computer-to-computer connections or user logon events." EX1004, [0025]. Stokes more generally teaches, as was known in the art, that in mathematical representations, the set of graph edges was denoted by "E" and "E is the set of connection relationships … between pairs of nodes, called edges." EX1004, [0026]; Section VIII.A *above*. Stokes therefore teaches that each edge of the plurality of edges in its graph corresponds to a relationship between a respective pair of nodes.

6. Element [1.6]

[1.6] receive streaming data comprising time-stamped data about events relating to one or more entities of the first plurality of entities,

82. Stokes-Crabtree satisfies this element, as Stokes and Crabtree each disclose real-time processing of streaming data for building or revising graphs.

83. Stokes teaches processing events "in real-time" (EX1004, [0023]) and implementing its graph-based framework "in a streaming environment" (*id*., [0038]). Stokes's disclosures of real-time event processing in a streaming environment teach

receiving streaming data because they describe event data being received and processed as an ongoing flow. EX1018, 499 (computer dictionary defining stream as a "data transmission… that occurs in a continuous flow").

84. Stokes further discloses that this streaming event data comprises time-stamped data about events relating to entities represented in a graph. *E.g.*, EX1004, [0037]-[0039]. For example, Stokes describes its server's network graph construction module collecting source and destination computers and user accounts from Kerberos event logs where "[e]ach Kerberos record represents an edge in the graph, with a source node, a destination node, and a timestamp indicating when this connection occurred." EX1004, [0038]. These records thus comprise time-stamped data about *events* (e.g., connection events) *relating to one or more entities of the first plurality of entities* (computers and user accounts). *Id.*; *see also id.*, [0037], [0039], [0045] (describing use of edge timestamps). Because Stokes teaches implementing its system in a streaming environment (*id.*, [0038]) and processing events "in real-time" (*id.*, [0023]), it follows that the same types of event information used by Stokes to build and analyze its graph would be received and processed in real time, instead of being processed from event logs. *Id.*, [0023], [0038].

85. Crabtree likewise discloses receiving a continual data stream, further confirming that a POSITA would have implemented Stokes-Crabtree with a Stokes-like server that received and processed streaming time-stamped event data. Crabtree

describes "retriev[ing] one or more streams of data from a plurality of sources" and "employing standardized protocols for converting streams of information into graph representations of that data." EX1005, [0050]. Because Crabtree converts streams of event information into graph representations, and further teaches producing a cyber-physical graph that includes nodes representing infrastructure elements (e.g., users, servers, devices, and other resources) and their relationships, Crabtree's streaming data comprises data about events relating to one or more entities represented by graph nodes. *Id.*, [0050], [0076].

86. Crabtree further discloses "live monitoring" of a network by collecting and receiving data continuously. EX1005, [0075]. Crabtree explains that time-series data "may be received continuously," with the system "passively collecting events and monitoring activity over time" while feeding the collected information into time-series graphs and related graph-analysis components. *Id*. Crabtree further explains that the graph information is updated "as changes occur and events are logged," reflecting an ongoing intake of event data rather than a one-time static import. *Id.*, [0076].

87. Crabtree's disclosure of streaming data matches the disclosures that the Applicant of the '627 patent relied on during prosecution. In amending the pending independent claims to recite "receiv[ing] streaming data…," the Applicant cited support at ¶¶ [93] and [94]. EX1002, 231/290. With regard to receiving data, those

cited paragraphs describe receiving "time-series data (as described above, referring to Fig. 10)." *Id*., 108/290. That description of time-series data in the context of Figure 10 is found in ¶ [90], which includes the exact same disclosures concerning live, continuous data collection as paragraph [0075] of Crabtree, relied on above. See EX1002, 106-107; EX1005, [0075]; EX1019, 33-35.

88. As for the data about the events being *time-stamped data*, Crabtree further teaches associating timestamps with changes in a time-series graph. EX1005, [0061]. For example, Crabtree explains that "timestamps may be associated with ongoing changes to reveal these updates over time." *Id*. Therefore, the Stokes-Crabtree combination includes receiving streaming time-stamped data about events relating to one or more entities of the first plurality of entities.

7. Element [1.7]

[1.7] based on a first portion of the streaming data, identify a first entity that does not correspond to any of the first plurality of nodes, wherein the first entity is not of the first plurality of entities,

89. In the streaming system taught by Stokes, the server receives and processes event data that arrives in portions, one after another, and uses this event data "to generate a graph." EX1004, [0037]-[0038]; *supra* Section VIII.D.6. For example, Stokes describes exemplary Kerberos records and how each "record" (portion of event data) "represents an edge in the graph, with a source node, a destination node, and a timestamp." EX1004, [0038]. These events are "processed

in real-time" (*id.*, [0023]) by the server's network graph construction module, which uses them "in the generation of the network connection graph" (*id.*, [0037]). Stokes describes ongoing generation of events corresponding to Kerberos records, over a period of hours or days, in a streaming environment. *E.g.*, EX1004, [0038], [0051], [0023].[2] Thus, Stokes teaches its server receiving streaming data made up of time-stamped portions (e.g., Kerberos records). EX1004, [0037]-[0038].

90. Stokes further teaches using this event data "to generate a graph 173 that shows the connections between the various nodes (e.g., the accounts and computers and their connections and relationships)." *Id.*, [0037]. Given these teachings, a POSITA would expect the obvious case of event data for certain later events arriving after initial graph generation, with this later event data indicating a new account or computer recently added to the network. Stokes teaches—and thus renders obvious (if not inherent)—a server configured to process this later portion of streaming event data, identify any newly added accounts or computers, and add corresponding nodes to the graph as part of its ongoing graph generation. *Id.*, [0037-

[2] Although the "hourly, daily" collection of "event logs" refers to Stokes's offline implementation (*id.*, [0038]), it shows ongoing receipt of events over time and Stokes teaches that these same "events may be processed in real-time instead of being processed from event logs." *Id.*, [0023].

38]; *see also id.*, [0051] (describing scenario when "[t]he graph can be implemented or extended"). As these nodes are "new," they are not of the first plurality of nodes previously in the graph and, similarly, the corresponding computer or account is *a first entity that does not correspond to any of the first plurality of nodes* and *is not of the first plurality of entities*.

91. While Stokes teaches a server configured to satisfy this claim element, it does not expressly describe the obvious scenario of identifying entities not already represented in the graph. To the extent Stokes alone does not render this element obvious, Crabtree expressly teaches using incoming event data to update its graph to include nodes and edges corresponding to entities and relationships not yet represented in the existing graph. EX1005, [0075] ("As new events are recorded…, these events are automatically incorporated"); [0076] (describing "updating CPG [graph] information as changes occur and events are logged"). Crabtree also describes identifying changes such as "new users" based on analysis of time-series data. *Id.*, [0078]; *see also id.*, [0079-81].

92. The graph modifications described by Crabtree would involve identifying entities not already represented by an existing node in the graph, not least because a POSITA would have understood that extending or updating a graph based on incoming event data necessarily involves determining whether an entity is already represented by an existing node in the graph. This is because a graph structure

represents entities as unique nodes, and the addition of duplicate nodes for the same entity would undermine the integrity and usefulness of the graph by fragmenting relationships and preventing accurate traversal and analysis; in some instances it could even be an outright error. Accordingly, when new event data is received, a system implementing graph updates must first evaluate whether the entity referenced in the event corresponds to an existing node. If the entity is not already represented, a new node must be created to incorporate that entity into the graph; if it is already represented, the existing node is used. This check-and-add process is inherent in any practical graph-based implementation that incrementally incorporates new event data; this fundamental series of steps is commonly taught to undergraduate Computer Science students usually in the first or second semester of their curriculum when learning basic data structures.

93. Notably, the alleged support identified by Applicant for the amendment adding this claim element during prosecution consisted solely of disclosure that appears verbatim in Crabtree. In particular, Applicant cited ¶¶ [93]-[94] and [96] of the as-filed specification as support for amending claim 1 to incorporate element [1.7], which are identical to ¶¶ [0078]-[0079] and [0081] of Crabtree. EX1002, 108-109/290, 231/290; EX1005, [0078]-[0079] and [0081]. Thus, the very disclosure Applicant relied upon as written description support for element [1.7] was already disclosed in Crabtree before the alleged invention date.

94. Thus, based on Stokes alone, or further in view of Crabtree, a POSITA would have implemented a Stokes-like system such that the server used a first portion of streaming data to identify an entity that is not of the first plurality of entities, such as a new user account or other entity not previously represented in the graph. *See supra* Section VIII.C (explaining the combination).

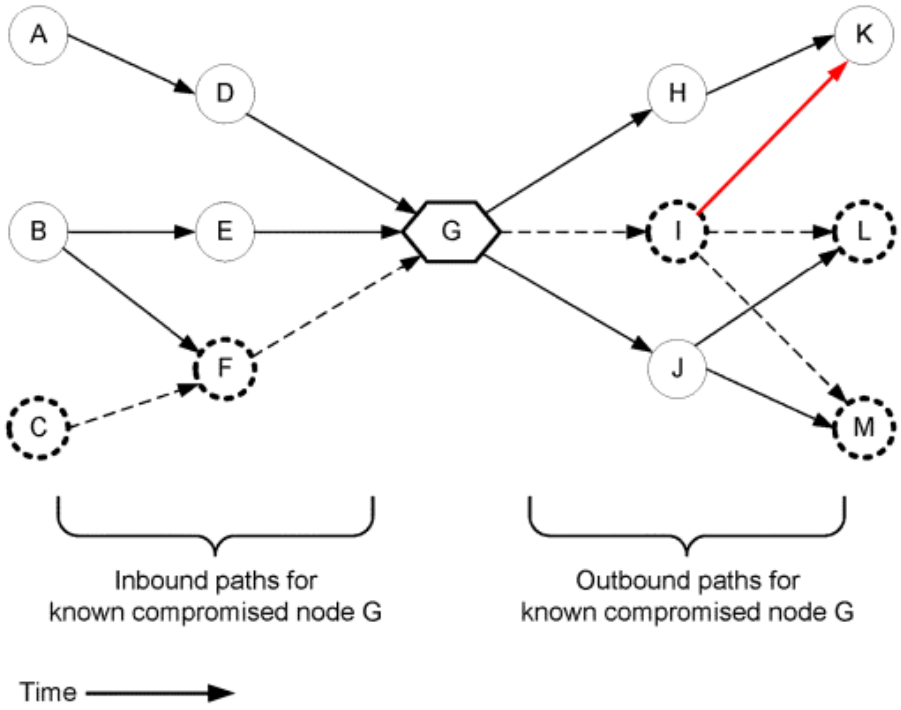
8. Element [1.8]

[1.8] based on a second portion of the streaming data, wherein the second portion is not identical to the first portion, identify a first relationship between a pair of entities of the first plurality of entities that does not correspond to any of the first plurality of edges,

95. As explained above, Stokes teaches its server receiving streaming data made up of time-stamped portions (e.g., corresponding to Kerberos records). Because those portions are received at different times for different events, later portions (e.g., a second portion) are not identical to earlier portions (e.g., a first portion). For example, a first portion of streaming event data could include data pertaining to a first set of events (e.g., one or more time-stamped Kerberos records), whereas a second, later-received portion of the streaming event data could include data pertaining to a second, different set of events (e.g., one or more Kerberos records having time stamps later than those of the first set of events). EX1004, [0037]-[0038].

96. Stokes-Crabtree includes a server configured to update its graph based on streaming time-series data for events reflecting new entities or relationships.

Supra Section VIII.D.7. Accordingly, the Stokes-Crabtree system would identify new relationships between pairs of entities that correspond to nodes already present in the graph, such as a first relationship between a pair of entities of the first plurality of entities that does not correspond to any of the first plurality of edges. This includes updating the graph with "new connection information" "[w]hen a new connection is detected," as taught by Crabtree. EX1005, [0081]; FIG. 16. Thus, the Stokes-Crabtree server identifies any new connections (i.e., new relationships) in incoming portions of event data, including new relationships between pairs of entities already represented by nodes in the graph. Crabtree describes its "new connection" generically, without limiting it to new devices (EX1005, [0081]; FIG. 16) and a POSITA would recognize the obvious use case of a new relationship between entities already represented by nodes in the graph (i.e., a pair of the first plurality of entities), because a POSITA understood that pre-existing network entities would often have new connections/relationships with each other. The Stokes-Crabtree server is configured to identify such relationships for inclusion in its graph as new edges. Figure 3 of Stokes shows that each node in the network may connect with multiple other nodes, thus allowing for new edges to reflect new relationships, including between previously graphed nodes. The following annotated Figure 3 shows a new edge (red arrow) added based on a new relationship identified between entities corresponding to nodes I and K.



EX1004, FIG. 3 (annotated), [0013]. Any newly-identified relationship does not correspond to any of the initially graphed (i.e., first) plurality of edges, as confirmed by the fact that it results in a new edge.

97. Further, as explained above and for element [1.7], Section VIII.D.7 *above*, Crabtree teaches identifying and incorporating into a graph both new entities and new connections as new data is received. EX1005, [0076], [0078], [0081]. Crabtree likewise explains that graph information is updated "as changes occur and events are logged." *Id.*, [0076]. Accordingly, a POSITA would have recognized the obvious use case in the Stokes-Crabtree system in which a first portion of streaming data identifies a previously unseen entity not yet represented in the graph, and a

second, later-received portion of streaming data identifies a new relationship between entities already represented in the graph, where that relationship is not yet represented by an edge. For example, the system could first identify a previously unseen entity based on a first portion of streaming data, such as a newly added user (*see id*., [0078]), and later, based on a second portion of streaming data, identify a new connection between entities already represented in the graph, prompting the graph to be "updated… with the new connection information" (*id*., [0081]). A POSITA would have understood that updating a graph based on incoming event data necessarily involves determining whether a relationship indicated by that data is already represented by an existing edge in the graph. In a graph representation, edges encode relationships between entities, and maintaining accurate relationships requires avoiding duplicate or inconsistent edges for the same interaction. Accordingly, when new event data indicates a relationship between entities corresponding to existing nodes, the system must determine whether an edge representing that relationship is already present. If the relationship is not already represented, a new edge must be added to accurately incorporate the newly observed interaction; if the edge is already represented, the existing edge is used or updated as appropriate. This check-and-add process is inherent in any graph-based implementation that incrementally incorporates event data, because without such a determination the graph would either omit newly observed relationships or

redundantly represent the same relationship multiple times, undermining subsequent graph traversal and analysis. This fundamental series of steps is commonly taught to undergraduate Computer Science students usually in the first or second semester of their curriculum when learning basic data structures.

98. Finally, given that Stokes-Crabtree receives ongoing streaming event data, a POSITA would contemplate the server identifying a new entity based on one portion of event data (first portion, *supra* Section VIII.D.7) and a new relationship based on a different portion of event data (second portion). *E.g.*, EX1005, [0081], FIG. 16.

99. Furthermore, as with element [1.7], the alleged support identified by Applicant for the amendment adding element [1.8] during prosecution consisted solely of disclosure that appears verbatim in Crabtree. *See* Section VIII.B *above.* In particular, Applicant cited ¶¶ [93]-[94] and [96] of the as-filed specification as support for amending claim 1 to incorporate element [1.8], which are identical to ¶¶ [0078]-[0079] and [0081] of Crabtree. EX1002, 108-109/290, 231/290; EX1005, [0078]-[0079] and [0081]. Thus, here again, the disclosure Applicant relied upon as written description support for element [1.8] was already disclosed in Crabtree before the alleged invention date.

9. Element [1.9]

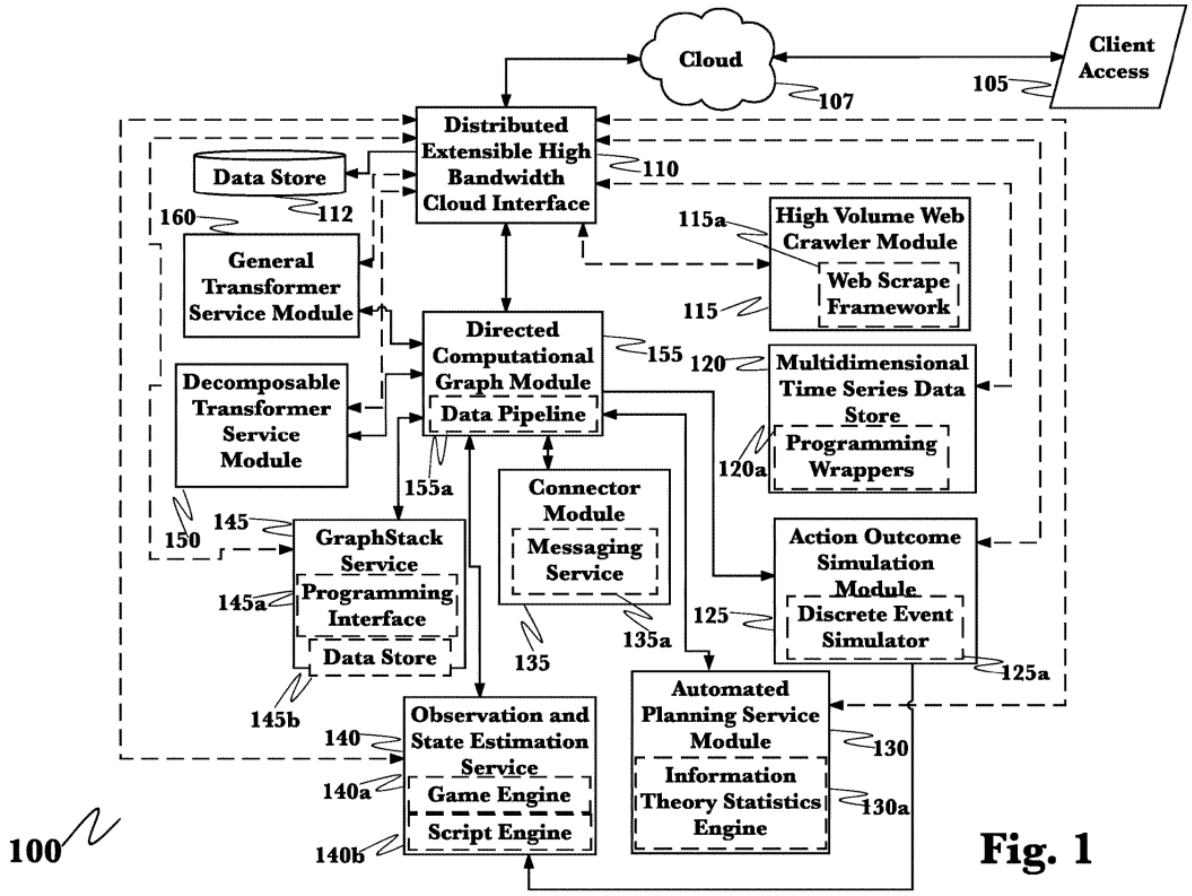
[1.9] modify, in the hardware memory, the representation of the first graph to generate a modified representation of the first graph, wherein the modified

representation of the first graph comprises a representation of a first node corresponding to the first entity and a representation of a first edge corresponding to the first relationship, wherein the first node is not of the first plurality of nodes and the first edge is not of the first plurality of edges,

100. Stokes-Crabtree satisfies this element.

101. Modify, in the hardware memory, the representation of the first graph to generate a modified representation of the first graph: As discussed above for element [1.3], Section VIII.D.3 *above*, Stokes's graph would be stored in hardware memory of the Stokes system (e.g., its computer storage media). Thus, in the Stokes-Crabtree system, any graph modifications would be reflected in hardware memory.

102. Crabtree reinforces that graph representations stored in hardware memory are modified when new connection information is detected. EX1005, [0081]. For example, Crabtree discloses that when "a new connection is detected [] a CPG may be updated [] with the new connection information." *Id*. Crabtree further discloses that a "graph stack service module 145 represents data in graphical form" and "stores it in a graph-based data store 145*b*." *Id.*, [0050]. The graph stack service module 145 and the graph-based data store 145b are schematically shown in Figure 1 of Crabtree:



Id., FIG. 1. Thus, Crabtree's graph-based data store would be implemented using the underlying physical memory and storage hardware of the computer system, and updating the graph would entail modifying the stored graph representation in the system's hardware memory.

103. Wherein the modified representation of the first graph comprises a representation of a first node corresponding to the first entity and a representation of a first edge corresponding to the first relationship, wherein the first node is not of the first plurality of nodes and the first edge is not of the first plurality of edges: As discussed above for elements [1.7] and [1.8], Sections VIII.D.7 and VIII.D.8 *above*,

the combined Stokes-Crabtree system would apply Crabtree's teachings to update the graph of Stokes to reflect newly received data, including not previously graphed entities and newly detected relationships. Because Stokes' graph reflects entities as nodes and relationships as edges (Section VIII.D.3 *above*) the Stokes-Crabtree graph would be modified to 1) include a new node (i.e. a first node) corresponding to the first entity as described for element [1.7] (Section VIII.D.7 *supra*), and 2) include a new edge (i.e., a first edge) corresponding to the first relationship as described for element [1.8] (Section VIII.D.8 *supra*). Because the first entity is a newly detected entity, its new node (the first node) would not be of the first plurality of nodes; and because the first relationship is a newly detected relationship, its edge (the first edge) is not of the first plurality of edges

104. Crabtree reinforces this understanding by teaching that when "a new connection is detected [] a CPG may be updated [] with the new connection information." EX1005, [0081]. Accordingly, when the new connection information involves an entity not yet represented among the graph's nodes, updating the graph would include adding a node corresponding to that entity, and when the new connection information involves a relationship not yet represented among the graph's edges, updating the graph would include adding an edge corresponding to that relationship.

105. Further, as with elements [1.7] and [1.8], the alleged support identified by Applicant for the amendment adding element [1.9] during prosecution consisted solely of disclosure that appears verbatim in Crabtree. *See* Section VIII.B *above.* In particular, Applicant cited ¶¶ [93]-[94] and [96] of the as-filed specification as support for amending claim 1 to incorporate element [1.9], which are identical to ¶¶ [0078]-[0079] and [0081] of Crabtree. EX1002, 108-109/290, 231/290; EX1005, [0078]-[0079] and [0081]. Thus, yet again, the disclosure Applicant relied upon as written description support for element [1.9] was already disclosed in Crabtree before the alleged invention date.

10. Element [1.10]

[1.10] identify, based on the modified representation of the first graph, an attack path that could be involved in an attack involving the first entity,

106. Stokes-Crabtree satisfies this element. Stokes describes using its graph to identify lateral movement paths associated with malicious nodes. For example, Stokes explains that, "[w]ith forensic analysis, given a malicious node, possible lateral movement leading into or out of the node is identified," and that, "[s]tarting with an infected computer or user account, forensic analysis identifies other compromised computers." EX1004, [0028]; *see also id.*, [0038] (explaining that "a series of lateral movements can be represented as a path through the network connection graph 173"). Thus, Stokes identifies lateral movements in a graph by first identifying a malicious node and then analyzing the graph's inbound and outbound

paths from that node based on their path-rate scores. *Id*., [0068]-[0071], FIG. 4. Stokes further explains that an infected computer "may be the source of the initial infection, an intermediate node in the lateral movement path, or the final destination in the attack." *Id*., [0062].

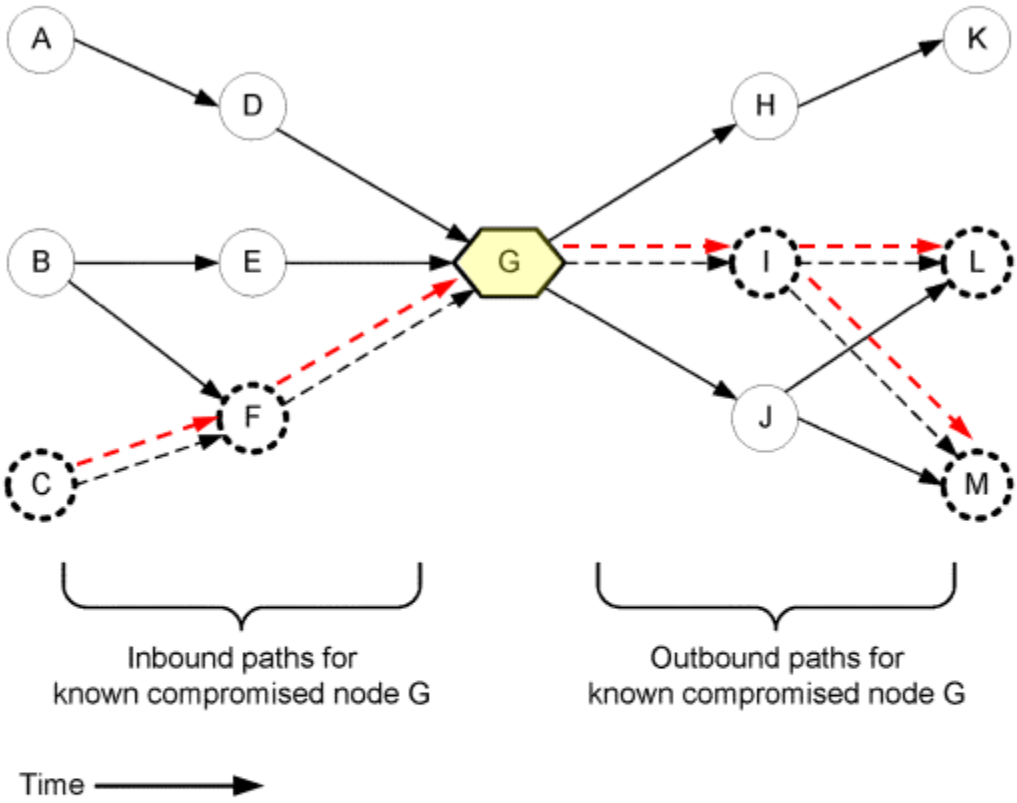
107. Stokes expressly frames such lateral movement paths as attacker activity, explaining that lateral movement "can be viewed as a set of malicious paths corresponding to the attacker's activity." EX1004, [0003]; *see also id*., [0002] ("Next, the attacker begins to use lateral movement techniques to explore the network, i.e., the process an attacker uses to move from one computer to the next in a compromised network."). Accordingly, Stokes's identified malicious paths disclose the claimed "attack path that could be involved in an attack."

108. As discussed above for elements [1.7] and [1.9], Sections VIII.D.7 and VIII.D.9 *supra*, the Stokes-Crabtree combination would include a modified graph representation that incorporates a newly-identified first entity as a first node in the graph. A POSITA would understand that after the first node is added to the modified graph, Stokes's forensic analysis techniques would determine whether that node lies along a malicious path. For example, when a node representing a computer or account not represented in the initial graph is added to the modified graph, Stokes's forensic analysis would identify a malicious path in the modified graph that includes that computer or account. EX1004, [0038]-[0039], [0047], [0066].

109. Stokes's teachings regarding forensic analysis further reinforce this conclusion. Stokes explains that, "[w]ith forensic analysis, given a malicious node, possible lateral movement leading into or out of the node is identified," and that, "[s]tarting with an infected computer or user account, forensic analysis identifies other compromised computers." EX1004, [0028]. Stokes thereby identifies lateral movements in a graph by first identifying a malicious node "on a network connection graph" and analyzing the graph's inbound and outbound paths from that node based on their path-rate score. *Id*., [0068]-[0071], FIG. 4. Stokes further explains that an infected computer "may be the source of the initial infection, an intermediate node in the lateral movement path, or the final destination in the attack." *Id*., [0062].

110. Stokes provides an example of identifying attack paths in its Figure 3. Figure 3 depicts an example network connection graph including nodes A-M; nodes A-M include a known compromised (i.e., infected) node G as well as nodes that are one and two hops away from node G along inbound and outbound paths. EX1004, FIG. 3, [0063]. Stokes describes performing inbound and outbound path analysis for compromised node G based on path rate scores using a forensic analysis module. *Id*., [0067]-[0071]. In the example of Figure 3, this analysis identifies "the unknown compromised computers and user accounts (dashed nodes C, F, I, L, and M) *along the malicious access path (dashed lines).*" *Id*., [0066] (emphasis added). Figure 3 is reproduced and annotated below. In annotated Figure 3, compromised node G is

highlighted and malicious access paths (shown in Fig. 3 as dashed black dashed arrows) are annotated using red dashed arrows to highlight those paths.



300

FIG. 3

Id., FIG. 3 (annotated). As explained above for elements [1.7] and [1.9], Sections VIII.D.7 and VIII.D.9 *above*, Stokes-Crabtree teaches identifying a new first entity and modifying the graph to include a first node corresponding to that first entity. EX1004, [0048], [0066]-[0071]. A POSITA would have recognized an obvious use

case in which the Stokes-Crabtree combination identifies a malicious path (i.e., attack path) based on the modified graph, and where the identified attack path involves the first entity. For example, a POSITA would have recognized an obvious use case in which compromised node G (highlighted above) corresponds to an entity not represented in the initial graph, and the Stokes-Crabtree system identifies the red dashed paths of annotated Figure 3, above, as attack paths involved in an attack involving the entity that corresponds to compromised node G (i.e., the first entity). Such use cases include a known malicious entity or a computer or account included in Stokes's "compromised computer and account list" newly accessing the network or the known compromised node otherwise having been newly added to the graph (for example for not having been previously accessed within the relevant time period for the graph). EX1004, [0045], [0048]. A POSITA would understand that in other obvious use cases, identified attack paths would be involved in attacks involving other entities, such as an entity which corresponds to another node of Figure 3 and was not represented in the initial graph.

111. Accordingly, Stokes-Crabtree teaches identifying attack paths involving a particular entity represented as a node in the graph.

11. Element [1.11]

[1.11] wherein identifying the attack path comprises:
identifying a second entity that can be reached using the first entity, wherein the second entity corresponds to a second node, and the second node is related by

one or more edges to the first node corresponding to the first entity in the modified representation of the first graph; and,

identifying a third entity that can be reached using the second entity, wherein the third entity corresponds to a third node, and the third node is related by one or more edges to the second node in the modified representation of the first graph; and

112. Stokes-Crabtree satisfies this element.

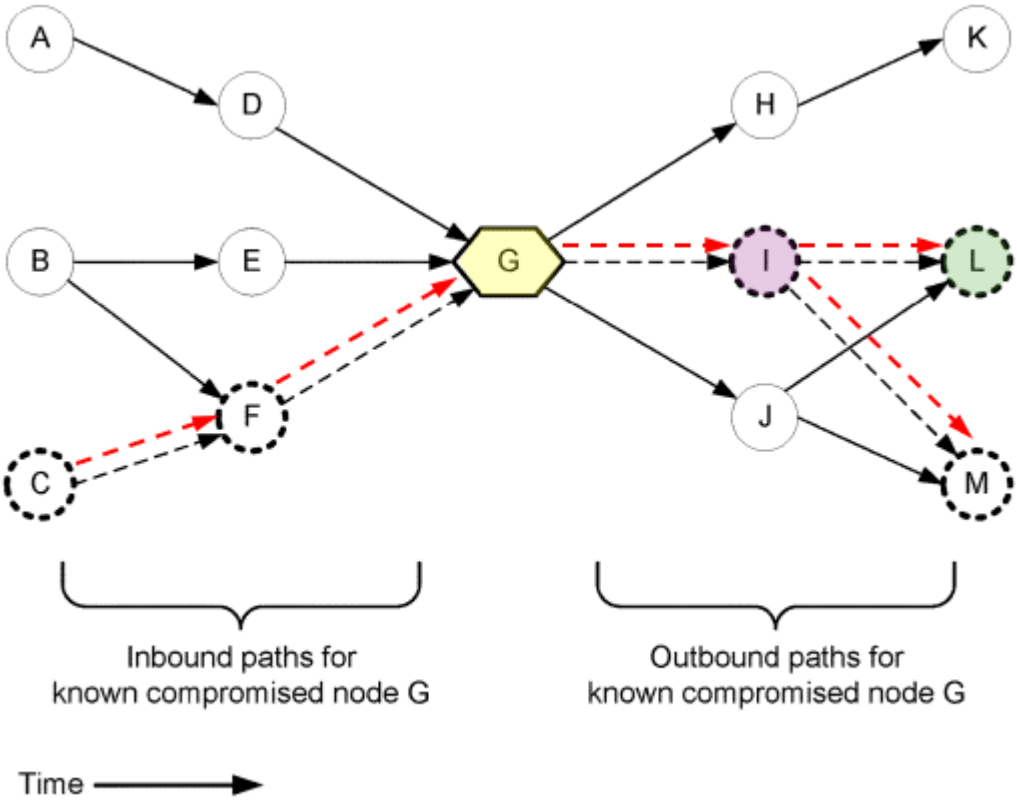
113. Wherein identifying the attack path comprises: identifying a second entity that can be reached using the first entity, wherein the second entity corresponds to a second node, and the second node is related by one or more edges to the first node corresponding to the first entity in the modified representation of the first graph: As described above for element [1.10], Section VIII.D.10 *above*, Stokes explains that "a series of lateral movements can be represented as a path through the network connection graph." EX1004, [0038]. Further, Stokes explains that "one input parameter to the system is K, the number of hops in suspicious paths which can also be considered the desired subpath length," where "K may be set equal to 2" or "longer subpaths with length K≥3 may be used." *Id.*, [0041]. Thus, Stokes teaches identifying paths through the graph comprising sequences of nodes connected by edges, including paths in which a second node is one hop away from a first node (i.e., related to the first node by at least one edge).

114. Stokes's forensic analysis discussion likewise describes "possible lateral movement leading into or out of the node," including direct connections that are one hop away from a first node (e.g., such that a second node is related by an

edge to the first node, and such that an entity corresponding to the second node can be reached by an entity corresponding to the first node). EX1004, [0005], [0028], [0064]-[0065].

115. Wherein identifying the attack path comprises … identifying a third entity that can be reached using the second entity, wherein the third entity corresponds to a third node, and the third node is related by one or more edges to the second node in the modified representation of the first graph: Stokes's forensic analysis discussion further describes indirect connections that are two or more hops away from a first node (e.g., such that a third node is related by an edge to the second node, and such that an entity corresponding to the third node can be reached by the entity corresponding to the second node). EX1004, [0064]-[0065]. Consistent with Stokes's disclosure of K-hop paths, these indirect connections correspond to paths that traverse from the first node to a second node and then to a third node, thereby identifying a third entity that can be reached using the second entity, as set forth in claim 1.

116. Stokes's teaching of element [1.11] is further reflected in Figure 3, reproduced and annotated below.



Id., FIG. 3 (annotated). As explained for element [1.10], Section VIII.D.10 *above*, Figure 3 teaches a first node (compromised node G, highlighted in yellow) and identifying the malicious access paths (i.e., attack paths, annotated as red dashed arrows) flowing through that node and the unknown compromised entities (computers and accounts) connected by those paths. As shown in Figure 3, above, this includes identifying a second node (node I, highlighted in purple) that corresponds to a second entity and is connected (i.e., related) by an edge to the first node (compromised node G), as well as identifying a third node (node L, highlighted in green) that corresponds to a third entity and is connected by an edge to the second node.

12. Element [1.12]

[1.12] generate a report comprising an identification of the first entity and at least one of the second entity and the third entity.

117. Stokes-Crabtree satisfies this element.

118. Stokes discloses a ranking module that is "configured to rank the plurality of paths… and output results of the ranking," EX1004, [0007] *see also id.,* [0047] ("The results 190 are output"), Fig. 1 (depicting results 190 output from the server). Stokes further explains that "[t]he ranking module [] may rank the outputs from the general detection module [] and/or the forensic analysis module [] to aid in the determination of previously unknown compromised nodes," where "[t]he highest ranked paths, with the lowest score, are the most rare and are more likely to correspond to malicious lateral movement." Stokes goes on to describe that "[i]n the forensic analysis case, a user (e.g., an analyst) can investigate other highly ranked paths to discover additional compromised nodes," and that "[f]or general detection, a user (e.g., an analyst) can focus on nodes with a remote file execution detection and highly ranked path-rate scores for manual investigation or additional monitoring." *Id*., [0047]. Such output allows an "analyst" or "system administrator" to then investigate identified suspicious paths manually. *Id*., [0048]. A POSITA would understand such output of results to a user to constitute a "report." *See* EX1018, 450 (defining a "report" as "[t]he presentation of information about a given topic"). Because the ranked paths are sequences of edges connecting sequences of

nodes, and because Stokes teaches that a user can investigate certain paths and focus on certain nodes, the output results would identify the entities included in each path (e.g., the computers and/or user accounts corresponding to the nodes in the path, and thus, for paths with at least two hops, the first entity and at least one of the second entity or the third entity included in the path). *Id.*, [0027].

119. Crabtree reinforces this understanding by teaching that cybersecurity analysis results may be presented to users in the form of reports providing visualizations identifying relevant entities and resources. For example, Crabtree discloses the ability to "format and deliver customized reports and dashboards." EX1005, [0063]-[0064]. Crabtree further discloses producing "a graph visualization of users, servers, devices, and other resources" and a "recommendation report." *Id.*, [0076], [0080]-[0081]. Such reports, alerts, dashboards, and visualizations identify the relevant entities implicated by the underlying cybersecurity analysis.

120. While Stokes already teaches one form of reports, a POSITA would have been motivated with a reasonable expectation of success to enhance the Stokes reporting with the formatted and customized reports taught by Crabtree. *Supra* Section C. This would have involved minor changes to include the reporting modules taught by Crabtree. Accordingly, Stokes teaches generating output results that identify entities involved in suspicious paths, and Crabtree confirms that it was well known to present such graph analysis results in a report identifying the relevant

entities for investigation. A POSITA therefore would have been motivated to generate a report including the data Stokes teaches outputting, in order to present the results of the graph-based analysis in a format that is more useful to an analyst conducting an investigation, including the formatted and customized reports taught by Crabtree. Further, in view of at least the disclosures of Stokes and Crabtree, generating such a report would have been routine and well within the ordinary skill of a POSITA.

121. Further, as with elements [1.7]-[1.9], the alleged support identified by Applicant for the amendment adding element [1.12] during prosecution consisted solely of disclosure that appears verbatim in Crabtree. *See* Section VIII.B *above.* In particular, Applicant cited ¶¶ [91] and [96]-[98] and Figure 11 of the as-filed specification as support for amending claim 1 to incorporate element [1.12], which are substantively identical to ¶¶ [0076] and [0081]-[0083] and Figure 11 of Crabtree[3]. EX1002, 107/290, 109-110/290, 231/290; EX1005, [0076], [0081]-[0083], FIG. 11. Accordingly, the disclosure Applicant relied upon as written

[3] ¶ [0076] of Crabtree has a typographical error ("method 50966" should be "method 1100"), but is otherwise identical to ¶ [91] of the as-filed specification. ¶¶ [0081]-[0083] of Crabtree are identical to ¶¶ [96]-[98] of the as-filed specification.

description support for element [1.12] was already disclosed in Crabtree before the alleged invention date.

E. Claim 2

[2] The computer system of claim 1, wherein identifying the attack path further comprises: identifying a first plurality of event flows that include a first anomalous event associated with the first entity.

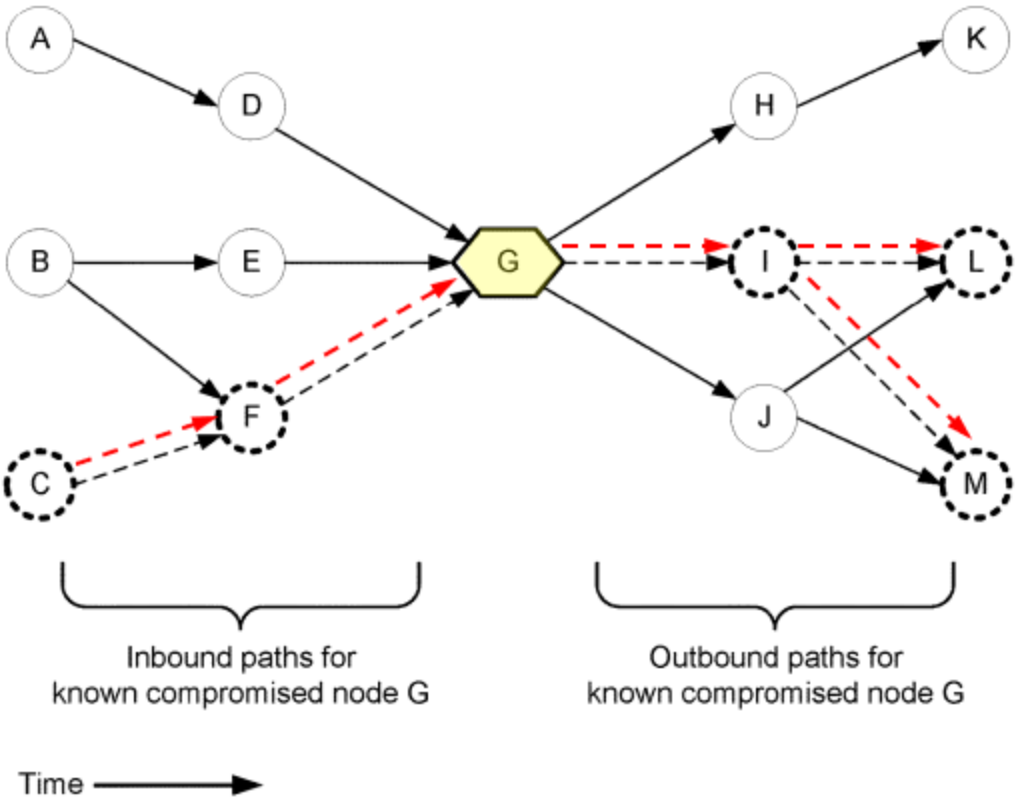
122. Stokes-Crabtree satisfies this element. Stokes discloses "[g]raph-based detection systems and techniques… to identify potential malicious lateral movement paths," where "[a] series of lateral movements across computers can be seen as paths constructed by a sequence of edges in the same direction." EX1004, [0020], [0025]. As edges in Stokes represent events (e.g., "computer connections and user logon events"), the disclosed "paths constructed by a sequence of edges in the same direction" satisfy the claimed "event flows." *Id*., [0006], [0025]. This is consistent with the '627 patent's description of "possible event flows" and "interaction pathways" through a behavior graph, which likewise treats event flows as directed pathways of events and interactions between affected nodes. EX1001, 21:54-22:42.

123. As explained for element [1.10], Section VIII.D.10 *supra*, Stokes identifies malicious lateral movements involved in a potential attack in a graph, including inbound and outbound paths from a known compromised node, based on path-rate scores. *See* EX1004, [0067], [0069]. Stokes's path-rate score module

"generates all possible paths and calculates the corresponding path-rate score, p, for each path" (*Id*., [0047]). This path rate score reflects "how rare (i.e., **anomalous**) a path is in a computer network" (*id*., [0044] (emphasis added)), such that the "highest ranked paths… are the most rare and are more likely to correspond to malicious lateral movement" (*id*., [0047]). Further, Stokes discloses the claimed "first anomalous event associated with the first entity" at least by describing various detectors used to detect "anomalous and security-significant behavior" involving accounts and/or hosts that correspond to nodes in the graph, such as a remote file execution (RFE) detector, PowerShell behavioral detectors, process execution anomaly detectors, account enumeration activity detectors, and process execution and command line behavioral detectors. EX1004, [0024], [0029]. For example, Stokes discloses "detect[ing] remote file executions using a remote file execution detector [] for use in tracking malicious lateral movement across a computer network, such as a compromised computer network … and/or hosts." *Id.*, [0024]. Such detected anomalous or security-significant behavior constitutes anomalous events associated with corresponding entities, such as the accounts or computers represented by nodes in the graph.

124. Stokes's Figure 3 provides a concrete example. With reference to the graph depicted in Figure 3, reproduced below, Stokes describes that dashed lines indicate a malicious access path associated with known compromised node G and

explains that, "given a malicious node, possible lateral movement leading into or out of the node is identified." EX1004, [0028], [0066]; *see also* [0044] (explaining that Stokes's forensic analysis of malicious lateral movements uses a "path-rate score" to "measure how rare (i.e., *anomalous*) a path is in a computer network" (emphasis added)). Stokes further states that the task is to identify "unknown compromised computers and user accounts (dashed nodes C, F, I, L, and M) along the malicious access path (dashed lines)." *Id.*, [0063], [0066].



300

FIG. 3

Id., FIG. 3. As shown above, Figure 3 depicts a plurality of event flows involving node G, including a first event flow through nodes C-F-G-I-L and a second event flow through nodes C-F-G-I-M. In view of Stokes's disclosures discussed above regarding anomalous or security-significant behavior involving corresponding accounts and/or hosts, these event flows include an anomalous event associated

with the entity represented by node G. Accordingly, Stokes teaches identifying a plurality of event flows including an anomalous event associated with an entity.

125. Further, as discussed above with respect to claim 1, in a system like Stokes that processes incoming system and security events over time and correspondingly modifies a graph, when a detected anomalous event involves an entity not already represented in the graph, that entity would correspond to a node added during an update to the graph (i.e., a first node corresponding to a first entity). Thus, the detected anomalous event would be associated with an entity that is not among a plurality of entities represented in the graph prior to the modification.

F. Claim 3

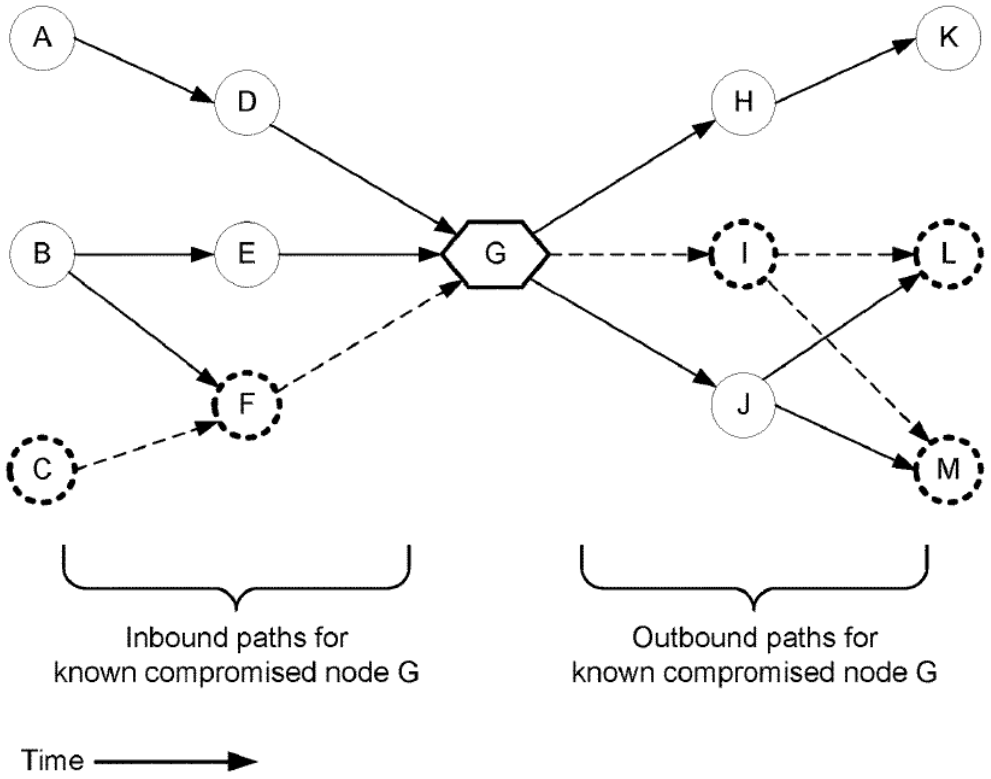
[3] The system of claim 2, wherein identifying the attack path further comprises: identifying a point of origin for the first anomalous event based on the first plurality of event flows.

126. Stokes-Crabtree satisfies this element.

127. As explained for elements [1.10] and [2], Sections VIII.D.10 and VIII.E *above*, Stokes utilizes a graph of nodes (including a known compromised node) and edges and performs forensic analysis using the graph to "identify the unknown compromised computers and user accounts… along the malicious access path." EX1004, [0066]-[0071]. And, Stokes explains that "[s]earching for potentially compromised nodes … considers inbound paths," meaning that Stokes identifies the

upstream series of edges and nodes along a malicious access path leading to the compromised node. *Id*., [0066].

128. Figure 3 of Stokes, for example, depicts the identification through forensic analysis of inbound paths and associated nodes (C and F) within "two hops" upstream of known compromised node G.



300

FIG. 3

Id., FIG. 3. A POSITA would understand the obvious use case of the Stokes forensic analysis identifying the point of origin of an anomalous event, such as the malicious

lateral movement depicted in Figure 3. Further, a POSITA would recognize the obvious use case of the source entity (and associated node) for a malicious intrusion (i.e., an anomalous event) accessing the network within one or two hops of the known compromised node and performing a lateral movement to the known compromised node. That is, the source node of the anomalous event would access the compromised node directly via an edge (1-hop) or would access the compromised node through a neighboring node (2-hops). In the example of Figure 3, Stokes's forensic analysis would identify such a point-of-origin node because it identifies the unknown compromised nodes (i.e., nodes C and F) within two hops of the known compromised node G. EX1004, [0039]-[0041], FIG. 3. Further, Stokes teaches that the number of hops in the graph may be larger than 2. *Id*., [0041] ("[O]ne input parameter to the system is K, the number of hops in suspicious paths…. [L]onger subpaths with length K≥3 may be used"). Thus, implementing Stokes's graph with a larger number of hops would identify any point-of-origin node within the given number of hops of a known compromised node.

G. Claim 5

[5] The computer system of claim 1, wherein the modified representation of the first graph comprises a representation of a node corresponding to the first entity, wherein the first entity is identified based on active reconnaissance results.

129. Stokes-Crabtree satisfies this element. As discussed above for element [1.9], Section VIII.D.9 *above*, the combination of Stokes and Crabtree teaches a modified graph representation that comprises a representation of a node corresponding to a first entity (i.e., the first node recited in claim 1). And, as discussed above for element 1.7, Section VIII.D.7 *supra*, Stokes-Crabtree would identify the first entity based on a first portion of streaming event data.

130. Stokes describes detectors for identifying anomalous activity, including a "remote file execution detector," that provide corroborating evidence of anomalous and security-significant behavior involving particular accounts and/or hosts; Stokes uses the detection results to filter network paths in the context of detecting malicious lateral movement. EX1004, [0024], [0029]. Accordingly, Stokes associates detector outputs with particular entities, such as accounts or hosts, and uses information associated with those entities in its path analysis. Although Stokes refers to "reconnaissance" in describing attacker behavior, it does not expressly specify whether its reconnaissance or detection techniques are active reconnaissance. *Id.*, [0019].

131. Crabtree describes active reconnaissance in the context of graph-based cybersecurity analysis. EX1005, [0059], [0060]. Crabtree describes performing "active scanning and reconnaissance," among other reconnaissance techniques, and analyzing results thereof to "[form] a graph of Internet-accessible organization resources and their evolving state information over time." EX1005, [0057]. Crabtree further teaches "performing external reconnaissance via a network" for "gathering information about hosts and services connected to a network" (*id*., [0052]), and teaches that received scan responses may be processed to produce a "hybrid graph/time-series database" forming a graph of Internet-accessible resources, with scanning information linked to CPG graphs (*id.*, [0057]). Crabtree further explains that these techniques may involve "both passive, semi-passive and active scanning and reconnaissance." *Id*. Thus, Crabtree teaches using active reconnaissance results to identify entities and related information for use in graph-based cybersecurity analysis (e.g., for inclusion as nodes in the graph.). *Id*. Crabtree's examples of DNS leak reconnaissance and web application reconnaissance reinforce this understanding. EX1005, [0059]-[0060].

132. The '627 patent's discussion of active reconnaissance substantially mirrors Crabtree. For example, the discussion of active reconnaissance at 13:43-14:22 of the '627 patent is identical to [0059]-[0060] of Crabtree. *See also* FIGS. 2B and 2C of Crabtree, which are identical to FIGS. 2B and 2C of the '627 patent.

133. As described in Section VIII.C *above*, a POSITA would have been motivated to apply Crabtree's active reconnaissance teachings in Stokes's graph-based detection framework to predictably improve identification of relevant entities and relationships used in analyzing suspicious paths. Further, a POSITA would have found it obvious to try identifying entities in the Stokes system using active reconnaissance because Crabtree describes "active scanning and reconnaissance" as one of a finite number of recognized techniques for obtaining such information. EX1005, [0057]. Accordingly, in such a system, network entities, including the first entity, would be identified through active reconnaissance and it would therefore have been obvious to the POSITA for the first entity to be identified using such active reconnaissance. And a POSITA would have had a reasonable expectation of success in doing so because Stokes already teaches the use of detector outputs to identify system entities and suggests using additional detector types in its system to improve detection. Section VIII.C *above*; EX1004, [0024]. *See also* Section VIII.P *below*, further describing incorporating Crabtree reconnaissance techniques into Stokes.

H. Claim 6

[6] The computer system of claim 1, wherein the modified representation of the first graph comprises a representation of a node corresponding to the first entity, wherein the first entity is identified based on passive reconnaissance results

134. Stokes-Crabtree satisfies this element. As discussed above for element [1.9], Section VIII.D.9 *above*, the Stokes-Crabtree combination teaches a modified graph representation that comprises a representation of a node corresponding to a first entity (i.e., the first node recited in claim 1). And, as discussed above for element 1.7, Section VIII.D.7 *above*, Stokes-Crabtree would identify the first entity based on a first portion of streaming event data.

135. As discussed above for element [5], Section VIII.G *above*, Stokes associates detector outputs with particular entities, such as accounts or hosts, and uses information associated with those entities in its path analysis. EX1004, [0024], [0029].

136. Crabtree explains that its techniques may involve "passive, semi-passive and active scanning and reconnaissance," and further describes "general steps… for performing passive network reconnaissance." EX1005, [0057]-[0058]. These passive reconnaissance steps include identifying network address ranges and domains/sub-domains associated with targets, external sites, people/groups, technologies used, and content of interest, as well as using publicly available

information to identify vulnerabilities. *Id*. A POSITA would have understood these disclosures to teach identifying an entity based on passive reconnaissance results.

137. Relatedly, Crabtree describes the use of passive data collection in the context of cybersecurity analytics and continuous workflow monitoring. For example, Crabtree describes using "passive information feeds" to analyze behavior patterns and identify anomalous behavior, and also describes "passively collecting events and monitoring activity over time" while feeding the collected information into a graphing service to produce graphs reflecting states and changes over time. *Id*., [0072], [0075].

138. The '627 patent's discussion of passive reconnaissance substantially mirrors Crabtree. For example, the discussion of passive reconnaissance at 12:62-13:42 of the '627 patent is identical to [0057]-[0058] of Crabtree; the discussion of passive reconnaissance at 22:43-67 of the '627 patent is identical to [0072] of Crabtree; and the discussion of "passively collecting events and monitoring activity over time while feeding [] collected information into a graphing service [] for producing time-series graphs [] of states and changes over time" at 23:34-54 of the '627 patent is identical to [0075] of Crabtree. See also FIGS. 2A, 8, and 10 of Crabtree, which are identical to FIGS. 2A, 8, and 10 of the '627 patent.

139. As described in Section VIII.C *above*, a POSITA would have been motivated to apply Crabtree's passive reconnaissance teachings in Stokes's graph-

based detection framework to predictably improve identification of relevant entities and relationships used in analyzing suspicious paths. Further, a POSITA would have found it obvious to try using passive reconnaissance because Crabtree describes "passive [] scanning and reconnaissance" as one of a finite number of recognized techniques for obtaining such information. EX1005, [0057]. Accordingly, in such a system, network entities, including the first entity, would be identified through passive reconnaissance and it would therefore have been obvious to the POSITA for the first entity to be identified using such passive reconnaissance. And a POSITA would have had a reasonable expectation of success in doing so for the same reasons explained for element [5], Section VIII.G *above*. *See also* Section VIII.P *below*, further describing incorporating Crabtree reconnaissance techniques into Stokes.

I. Claim 11

[11] The computer system of claim 1, wherein the computer system is further configured to execute software instructions stored on nontransitory machine-readable storage media comprising software instructions that determine whether an event, of the events relating to one or more entities of the first plurality of entities, is anomalous, wherein determining whether the event is anomalous comprises: determining that the event relates to the first entity, determining at least one behavior pattern associated with the first entity, and comparing the event to the at least one behavior pattern.

140. Stokes-Crabtree satisfies this element. Stokes describes representing network connection events as edges in a graph, where the edges connect nodes corresponding to entities such as computers or user accounts. EX1004, [0037]-

[0038]. Thus, Stokes teaches determining whether a given event relates to particular entities represented in the graph. *Id.*

141. Stokes further describes assigning weights to edges based on an "X-day history of data" (e.g., $w(v,v')=C(v,v')/X$, reflecting a historical rate of connection between nodes). EX1004, [0044]. This history-based connection information represents at least one behavior pattern associated with an entity. *Id*. Stokes then uses the assigned weights to compute a path-rate score that provides a measure of how rare, i.e., anomalous, a path is. EX1004, [0044]. Thus, Stokes teaches determining whether activity associated with an event is anomalous based on comparison to a history-based behavior pattern associated with the relevant entity or entities. *Id*.

142. Crabtree describes using behavior analytics in the context of graph-based cybersecurity analysis. EX1005, [0072], [0076], [0082]; *see also* FIGS. 8, 11, 17. In particular, Crabtree describes processing activity information using a graph to analyze behavior patterns and recognizing anomalous behavior based on a threshold of variance from an established pattern or trend. *Id*. Crabtree further describes incorporating behavior analytics information into a cyber-physical graph and using that graph to track changes and detect behavioral anomalies. *Id*.

143. A POSITA would have been motivated to apply Crabtree's behavior analytics teachings in Stokes's graph-based detection framework to improve

determination of whether network events associated with particular entities are anomalous. Doing so would have predictably improved identification of suspicious activity in the context of graph-based analysis of malicious lateral movement. *See* Section VIII.C *above*.

J. Claim 12

[12] The computer system of claim 11, wherein comparing the event to the at least one behavior pattern comprises using a threshold.

144. Stokes-Crabtree satisfies this element. As discussed above for element [11], Section VIII.I *above*, Stokes describes using history-based connection information associated with an entity as a behavior pattern and determining anomalousness based on comparison to that history-based information. EX1004, [0043]-[0044]. Stokes further discloses using explicit thresholds in that analysis. For example, Stokes describes identifying paths having a "path-rate score [] less than the threshold D," where the path-rate score provides a measure of how rare, i.e., anomalous, a path is. *Id*.

145. Crabtree likewise describes threshold-based anomaly recognition in the context of behavior analytics. As discussed above for element [11], Section VIII.I *above*, Crabtree describes recognizing anomalous behavior based on "a threshold of variance from an established pattern or trend." EX1005, [0072].

146. A POSITA would have been motivated to apply Crabtree's threshold-based behavior-analytics teachings in Stokes's graph-based detection framework to improve determination of whether events associated with particular entities are anomalous. In the combined system, an event would be compared to at least one behavior pattern using a threshold (in view of Stokes's use of explicit thresholds in evaluating anomalousness and Crabtree's teaching of recognizing anomalous behavior based on a threshold of variance from an established pattern or trend). The proposed modification would have predictably improved Stokes's threshold-based determination of anomalous activity associated with particular entities in the context of graph-based analysis of malicious lateral movement. *See* Section VIII.C *above*.

K. Claim 13

[13] The system of claim 11, wherein comparing the event to the at least one behavior pattern further comprises: determining that the first entity is associated with a resource, wherein the resource is a sensitive resource.

147. Stokes-Crabtree satisfies this element. As discussed above for element [11], Section VIII.I *above*, the Stokes-Crabtree system would determine whether an event associated with a particular entity is anomalous by comparing the event to a behavior pattern associated with that entity.

148. Crabtree teaches that the significance of a compromised account depends on the resources that account can access. In particular, Crabtree teaches generating an "impact assessment score" for an account based on the account's

"access capabilities," such as what files, directories, devices, or domains the account may access, where the score reflects "the potential risk should that account be compromised." EX1005, [0074]. Crabtree further explains that such analysis may identify "exactly what resources are at risk" and "high risk accounts or resources." *Id*. Crabtree also teaches that an unknown connection to "an internal data server with sensitive information" may present much greater risk than an unknown connection to an externally facing web server. *Id*., [0081]. A POSITA would therefore understand that a user account with access to an internal data server containing sensitive information is an entity associated with a sensitive resource.[4]

149. Crabtree further describes recognizing anomalous behavior, including behavior associated with "high-risk users," based on analysis of behavior patterns in activity information, which a POSITA would understand to include information regarding events. EX1005, [0072]. A POSITA would further understand that, in

[4] The written description of the '627 patent does not use the term "sensitive resource." Instead, it discloses "particularly sensitive data stores" in the context of using behavior patterns to identify anomalies, and further discloses, like Crabtree, examining the context of a connection for risk potential, "for example, an unknown connection to an internal data server with sensitive information." EX1001, 21:5-47, 25:22-39; EX1019, 81.

Crabtree, whether a user is high-risk depends at least in part on the risk level or sensitivity of the resources accessible to that user. EX1005, [0074], [0081]. Thus, a POSITA would understand that when the Stokes-Crabtree system compares an event associated with a user account to a behavior pattern associated with that account, the system would also determine whether the account is associated with a sensitive resource.

150. A POSITA would have been motivated to incorporate Crabtree's impact assessment teachings into Stokes's system to improve Stokes's response to suspicious activity after it is identified. Stokes's forensic analysis identifies suspicious users, accounts, and assets involved in anomalous paths, and Crabtree teaches using impact assessment information to determine what resources are at risk and whether those resources are high-risk or sensitive. Sections VIII.A and VIII.B *above*; EX1005, [0074], [0081]. A POSITA would have understood that using Crabtree's determination of resource risk level/sensitivity in a Stokes-like system would improve alert prioritization and response selection by helping an analyst, or an automated response mechanism, determine which user account or asset should have access restricted or cut off first based not only on anomalous behavior, but also on the sensitivity of the resources associated with that entity.

L. Claim 14

[14] The computer system of claim 1, wherein the computer system comprises a plurality of physical computing machines

151. Stokes-Crabtree satisfies this element. Stokes's "environment 100" expressly includes "one or more client devices 110*a*, 110*b* and a server 130. EX1004, [0021]. Stokes's Figure 1 depicts the client devices 110a, 110b and server 130 in the context of environment 100:

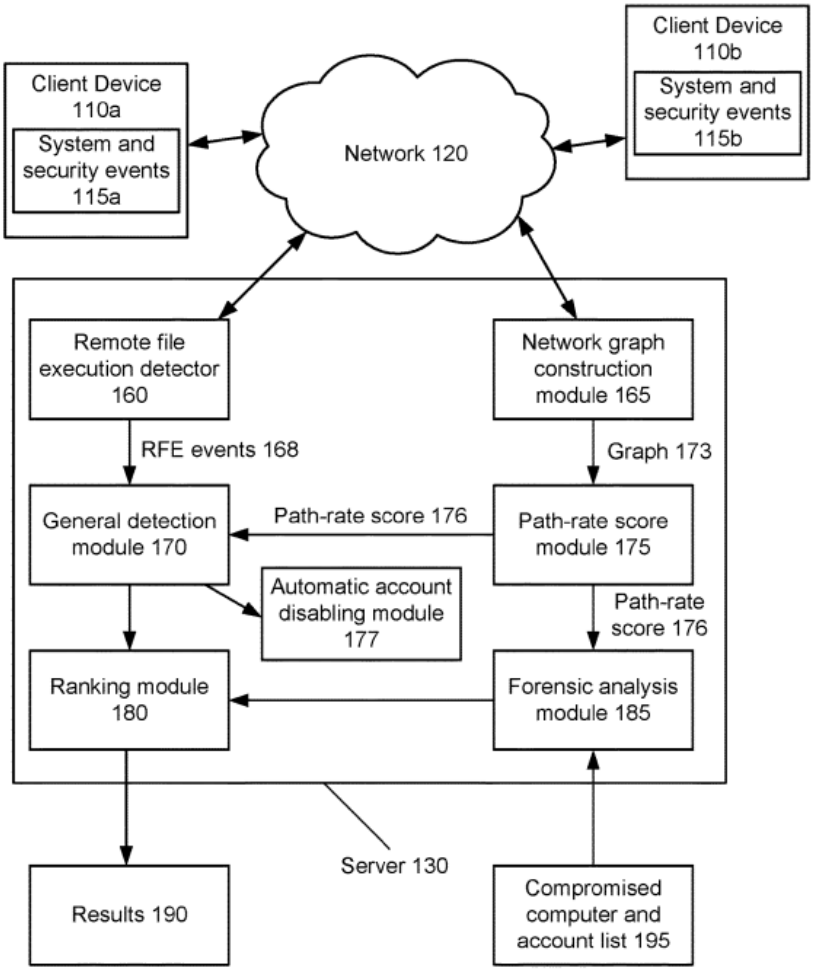


FIG. 1

Id., FIG. 1. Thus, Stokes discloses a computer system comprising a plurality of physical computing machines.

M. Claim 16

[16] The computer system of claim 1, wherein at least one entity of the first plurality of entities is at least one of a user, a place, a device, a resource, a group, or a service.

152. Stokes-Crabtree satisfies this element. Stokes describes nodes that represent computers (i.e., devices) and user accounts. EX1004, [0006], [0023]. Section VIII.D *above*.

N. Claim 17

[17] The computer system of claim 1, wherein the representation of the first graph comprises a representation of at least one node that does not correspond to an entity.

153. Stokes-Crabtree satisfies this element. Stokes describes a graph-based cybersecurity system in which nodes correspond to entities such as user accounts and computers. EX1004, [0037]-[0038]. Stokes does not expressly describe a graph comprising a node that does not correspond to an entity.

154. Crabtree, however, describes a graph in which at least some nodes do not correspond to entities. In particular, Crabtree explains that its directed computational graph module "represents all data as directed graphs where the transformations are nodes and the result messages between transformations edges of the graph." EX1005, [0050]. Such transformation nodes represent processing or analytic steps performed on data, rather than entities such as users, accounts, devices,

or resources. Crabtree therefore teaches a graph representation comprising at least one node that does not correspond to an entity. EX1005, [0050].

155. A POSITA would have been motivated to incorporate non-entity nodes into a Stokes-like system, such as those taught by Crabtree, because doing so would have predictably improved the graph's ability to represent processing or analytic steps relevant to suspicious path analysis, thereby improving the graph's usefulness for tracing and analyzing suspicious activity. This is consistent with the state of the art. *See*, *e.g.*, EX1006, [0069] (describing a graph representation in which nodes represent "subjects (processes)" and "objects (files, sockets)," and explaining that the resulting "scenario graph" serves as the basis for "attack detection as well as causality analysis"). Thus, a POSITA would have understood that graph-based cybersecurity analysis is not limited to graphs containing only nodes representing conventional entities such as accounts and computers, but instead can benefit from richer graph representations that also include non-entity nodes.

O. Claim 18

1. Elements [18.1]-[18.5], [18.7], and [18.9]

156. Elements [18.1]-[18.5], [18.7], and [18.9] are substantively identical to elements [1.1]-[1.5], [1.6] and [1.8], respectively. EX1016, 1-2. Stokes-Crabtree satisfies those elements for the same reasons explained above for elements [1.1]-

[1.5] and, [1.6], and [1.8] in Sections VIII.D.1-VIII.D.5, VIII.D.6 and VIII.D.8, *above*.

2. Element [18.6]

[18.6] identify, based on the representation of the first graph, a first plurality of attack paths comprising a first entity of the first plurality of entities, wherein each attack path of the first plurality of attack paths targets a second entity of the first plurality of entities,

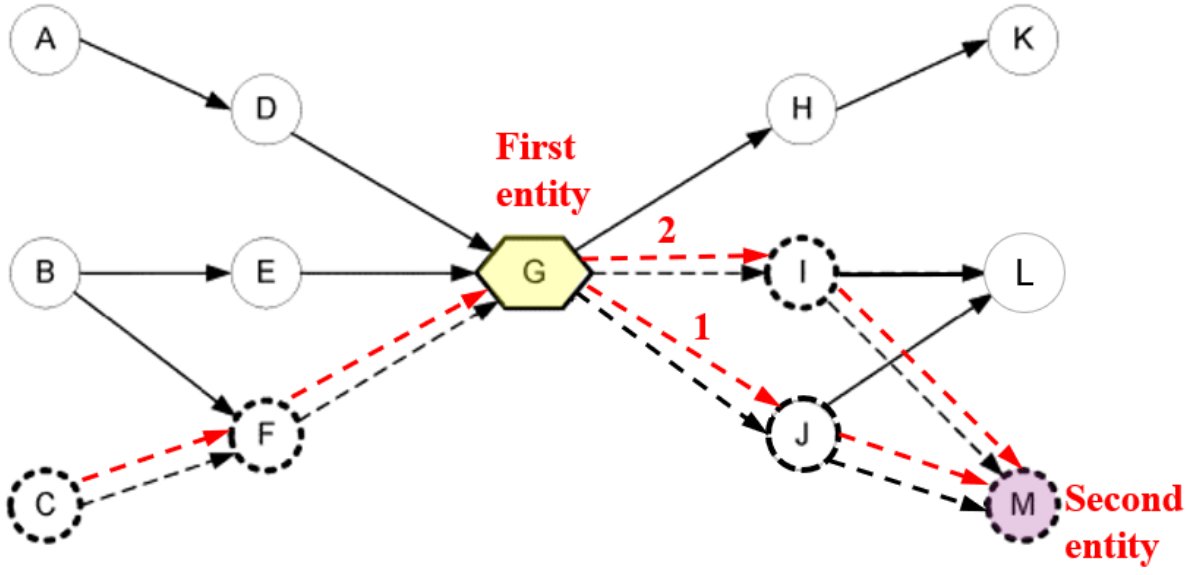
157. Stokes-Crabtree satisfies this element. As discussed above for element [1.10], Section VIII.D.10 *above*, Stokes teaches identifying plural paths through the graph that correspond to malicious lateral movement and thus correspond to the claimed "attack paths." EX1004, [0003] (explaining that lateral movement "can be viewed as a set of malicious paths corresponding to the attacker's activity"), [0029] ("to discover a new attack, general detection identifies unknown lateral movement across nodes which are not known to be compromised"), [0038]-[0039] ("a series of lateral movements can be represented as a path through the network connection graph… [m]alicious lateral movement subgraphs may be identified in the network connection graph," and the system may "search for malicious subpaths (i.e., rare K-hop paths), and then construct the overall malicious graph by joining these subpaths together").

158. Stokes provides an example of identifying such malicious paths in its Figure 3. Stokes explains that "in FIG. 3, a known compromised node (denoted as

[node] G) is discovered and the task is to identify the unknown compromised computers and user accounts (dashed nodes C, F, I, L, and M) along the malicious access path (dashed lines).” EX1004, [0066]. As an illustrative example, a first modified version of Stokes’s Figure 3 is provided below with annotations[5] (red dashed lines) depicting an obvious use case of Stokes graph in a network setting. The modified Figure 3 shows a plurality of malicious paths identified using Stokes’s techniques, including a first path (including nodes C, F, G, J, and M) and a second path (including nodes C, F, G, I, and M). In modified Figure 3, node G (representing a first entity included in the first and second paths) is highlighted in yellow, and node M (representing a second entity at the end of the first and second paths) is highlighted in purple. As shown, the first and second paths each include the edge between nodes

[5] In the first modified version of Figure 3, for ease of explanation, the edge leading from node I to node L has been converted from a dashed line to a solid line, node L has been converted from a dashed node to a solid node, node J has been converted from a solid node to a dashed node, and the edges leading from node G to node J and from node J to node M have been converted from solid lines to dashed lines. Further, the bracketed inbound and outbound path labels and time arrow have been omitted.

C and F and the edge between nodes F and G; the paths then diverge downstream of node G, as indicated by the red numeric labels.



As shown above, the first and second paths each comprise a first entity, represented by node G, and terminate at a second entity, represented by node M. In addition to being the endpoint at which the first and second paths terminate, node M is one of the "unknown compromised computers and user accounts" included in these paths. EX1004, [0066]. Accordingly, node M corresponds to the claimed second entity targeted by the first plurality of attack paths. This is consistent with the '627 patent, in which the exemplary attack paths shown in Figure 22 terminate at a target device 2206. EX1001, FIG. 22, 28:4-65.

159. Thus, Stokes teaches identifying, based on a representation of a first graph, a first plurality of attack paths that each comprise a first entity and target a

second entity, where the first and second entities are among a plurality of entities represented by nodes in the graph.

3. Element [18.8]

160. Element [18.8] differs from element [1.7] only in that the former recites a "third entity" where the latter recites a "first entity." EX1016. As explained for element [1.7], supra Section VIII.D.7, Crabtree-Stokes identifies a newly detected entity, not corresponding to any of the first plurality of nodes, in a first portion of streaming data. While such disclosure was mapped to a "first entity" for purposes of element [1.7], such disclosure equally satisfies the "third entity" for purposes of element [18.8] for the same reasons explained for element [1.7].

4. Element [18.10]

161. Element [18.10] differs from element [1.9] in that the former recites a "*third entity*," a "*node* corresponding to the *third entity*," and "*an edge* corresponding to the first relationship" where the latter recites a "*first entity*," a "*first node* corresponding to the *first entity,*" and "a *first edge* corresponding to the first relationship." EX1016. As explained for element [1.9], *supra* Section VIII.D.9, Crabtree-Stokes modifies a graph to include a new node corresponding to a newly detected entity and not of the first plurality of nodes, as well as updating the graph to include a new edge corresponding to a newly detected relationship and not of the first plurality of edges. While such disclosures were mapped to the "first entity,"

"first node corresponding to the first entity," and "a first edge corresponding to the first relationship" for purposes of element [1.9], such disclosure equally satisfies, respectively, the "third entity," "node corresponding to the third entity," and "an edge corresponding to the first relationship" for purposes of element [18.10] for the same reasons explained for element [1.9].

5. Element [18.11]

[18.11] identify, based on the modified representation of the first graph, a second plurality of attack paths comprising the first entity, wherein each attack path of the second plurality of attack paths targets the second entity, and wherein an attack path of the second plurality of attack paths comprises the third entity.

162. Stokes-Crabtree satisfies this element. As discussed above for element [18.6], Section VIII.O.2 *above*, Stokes describes representing a series of lateral movements as a path through a network connection graph; the example graph shown in Figure 3 includes a plurality of such paths. EX1004, FIG. 3, [0003], [0029], [0038]-[0039], [0066].

163. Still further, as discussed above for elements [1.9] and [18.10], Section VIII.D.9 and VIII.O.4 *above*, the Stokes-Crabtree combination teaches a modified graph representation that includes nodes corresponding to entities that were not represented by nodes in the graph representation before the modification, including the claimed third entity. EX1004, [0051]; EX1005, FIG. 1, [0050], [0081]. In an example where a newly added node corresponds to an unknown compromised

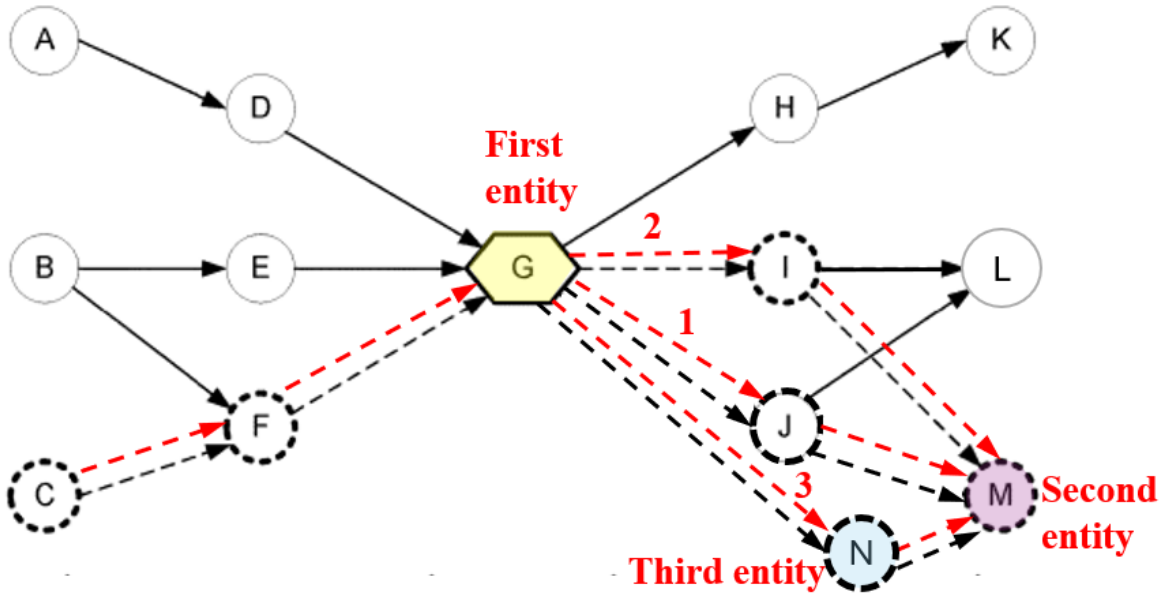
computer or user account, the Stokes-Crabtree system would identify, based on the modified graph, a malicious path that includes the entity corresponding to that node, (i.e., the claimed third entity). A POSITA would have understood that malicious lateral movements in a network would involve a malicious actor exploring and accessing multiple resources throughout the network, including to acquire additional credentials enabling access to additional network resources, and that the Stokes-Crabtree graph would be updated to reflect such exploration and access.. EX1004, [0002] ("Next, the attacker begins to use lateral movement techniques to explore the network, i.e., the process an attacker uses to move from one computer to the next in a compromised network. This is usually done by logging onto additional computers using the user's compromised credentials, exploring or creating network shares to these computers in the network."), [0031] ("Once an attacker has obtained the credentials to a privileged account, such as a domain administrator, a remote file execution (RFE) can be used to infect other computers on the network at will."); EX1017, 4:10-21 ("Through the attachment the attacker may gain access to a host device, and *subsequently use that initial access to gain further access to a system resource such as stored data or a process. Such further access may then be leveraged to acquire credentials enabling additional access to system resources.* In this way, an attacker may explore a computing environment…. Through this exploration, the

attacker's access may spread until it is able to gain access to a potentially valuable target.") (emphasis added).

164. Such a scenario is depicted in a second modified version of Stokes's Figure 3 provided below with annotations[6]. The second modified version of Figure 3 includes annotations (red dashed lines) showing a second plurality of attack paths including a first path (including nodes C, F, G, J, and M) and a second path (including nodes C, F, G, I, and M). Node G (corresponding to the first entity) is highlighted in yellow, and node M (corresponding to the second entity) is highlighted in purple. Further, a new node N added to the modified representation of the first graph (corresponding to a third, newly-identified entity) is highlighted in

[6] As in the first modified version of Stokes's Figure 3 provided for element [18.6], for ease of explanation, the edge leading from node I to node L has been converted from a dashed line to a solid line, node L has been converted from a dashed node to a solid node, node J has been converted from a solid node to a dashed node, and the edges leading from node G to node J and from node J to node M have been converted from solid lines to dashed lines in the second modified version. Further, new black dashed lines have been added to represent the edge between nodes G and N and the edge between nodes N and M, and the bracketed inbound and outbound path labels and time arrow have been omitted.

blue. Node N is included in a third path of the second plurality of attack paths; the third path also includes nodes C, F, G, and M. As shown, the first, second, and third paths each include the edge between nodes C and F and the edge between nodes F and G; the paths then diverge downstream of node G, as indicated by the red numeric labels.



When new entities are identified in incoming event data in the Stokes-Crabtree system, the graph would be modified to include nodes corresponding to the newly identified entities; node N represents such a node. In the above example, the first, second, and third paths each comprise the first entity (i.e., the same first entity included in the first plurality of attack paths discussed above for element [18.6], Section VIII.O.2 *above*, represented by node G) and terminate at the second entity (i.e., the same second entity at which the first plurality of attack paths terminate,

represented by node M). Further, as discussed above for element [18.6], Section VIII.O.2 *above*, node M corresponds to an entity that is targeted by the first and second plurality of attack paths (i.e., the claimed second entity); thus, the third path targets the second entity (node M), at which it terminates. Furthermore, as shown, node N of the third path corresponds to a newly-identified entity (i.e., the claimed third entity); thus, the third path shown in modified Figure 3 is an attack path of the second plurality of attack paths that comprises the third entity.

165. A POSITA would recognize the obvious use case (as reflected in the second modified version of Figure 3) in which an attacker, after unsuccessfully attempting to access the second entity (node M) through the entities corresponding to nodes I and J, tries to access a different entity in the network (for example one having different access credentials), specifically the third entity (node N), in an attempt to reach the second entity via the third entity.

166. A POSITA would also recognize a malicious entity targeting the second entity (node M) through additional entities not represented by nodes in the second modified version of Figure 3. In such an example, the graph would be modified to include additional nodes between nodes G and M that correspond to the additional entities (i.e., in addition to node N), for use in identifying additional malicious paths including those entities.

167. Accordingly, Stokes-Crabtree teaches identifying, based on the modified representation of the first graph, a second plurality of attack paths comprising the first entity, wherein each attack path of the second plurality of attack paths targets the second entity, and wherein an attack path of the second plurality of attack paths comprises the third entity.

P. Claim 22

> [22] The computer system of claim 18, wherein a portion of the representation of the first graph is derived from reconnaissance data received by the computer system from a plurality of computer systems, wherein a first computer system of the plurality of computer systems performs passive reconnaissance, and wherein a second computer system of the plurality of computer systems performs active reconnaissance.

168. Stokes-Crabtree satisfies this element. Stokes does not expressly describe its detection system performing reconnaissance.[7] Crabtree, however, provides complementary teachings on using both passive and active reconnaissance to improve cybersecurity analysis. *E.g.*, EX1005, [0057]-[0060], [0072], [0075], [0081]. Crabtree teaches using reconnaissance-derived streaming data to identify entities and relationships and represent them in graph-based models. Active and passive external reconnaissance "feeds information into a distributed computational

[7] Stokes uses "reconnaissance" to describe malicious activity by "attackers," which is not pertinent to the challenged claims. *E.g.*, EX1004, [0019].

graph data pipeline to produce hybrid graph/time-series data" (EX1005, [0005]), time-series data is analyzed by a directed computational graph module (*id.*, [0007]-[0009]), and the directed computational graph module retrieves "one or more streams of data," while the graph stack service converts "streams of information into graph representations of that data" (*id.*, [0049]-[0050]). Crabtree explains that reconnaissance scan responses are processed through data pipelines to produce a hybrid graph/time-series database, "forming a graph of Internet-accessible organization resources and their evolving state information over time." *Id.*, [0057]. Crabtree specifies that for its external reconnaissance, "techniques used may involve *both passive*, semi-passive and *active scanning and reconnaissance*." *Id.* (emphasis added).

169. Crabtree gives concrete examples of entities and relationships identified through passive and active reconnaissance, including hosts and services connected to a network (*id.*, [0052]), IP addresses, domains, external sites, people, technologies, content of interest, vulnerabilities (*id.*, [0058]), DNS trust maps (*id.*, [0059]), and host or application details derived from response messages (*id.*, [0060]). It further teaches that such reconnaissance reveals "additional insights, relationships, and vulnerabilities" (*id.*, [0061]), and that repeated scanning results may be collected into a time-series hybrid data structure that evolves over time to reflect newly discovered or changed devices and services (*id.*, [0062]).

170. A POSITA would have been motivated to incorporate Crabtree's complementary reconnaissance teachings into Stokes's streaming-environment graph analysis. As explained above, the two references describe similar graph-based techniques for analyzing network security, and provide complementary teachings towards this common end. *Supra* Section VIII.C. Crabtree teaches converting reconnaissance responses into time-series and graph representations of entities and relationships (*e.g.*, EX1005, [0005], [0007]-[0009], [0057]), while Stokes teaches real-time construction and analysis of a network graph using a stream of successive internal event records (*e.g.*, EX1004, [0005]-[0008], [0023]-[0024], [0037]-[0038], [0051]).

171. A POSITA would have understood Crabtree's reconnaissance teachings as providing an additional stream of relevant entity/relationship information that could be incorporated into, or used alongside, Stokes's event-derived streaming graph analysis. A POSITA would have reasonably expected success in combining these teachings because both references perform graph-based security analysis using streaming and/or time-series inputs, and Crabtree expressly teaches transforming reconnaissance responses into graph/time-series representations of entities and relationships compatible with the graph data Stokes analyzes. Using Crabtree's reconnaissance-derived graph information in Stokes's streaming graph framework would have been a straightforward application of

compatible graph- and time-series-processing techniques to enrich the graph with additional entities and relationships.

172. A POSITA understood the respective benefits of active and passive reconnaissance techniques for identifying network entities, and that performing both techniques together would maximize coverage and obtain the benefits of both techniques. EX1022, 19 (2006 IEEE publication explaining that "[a]ctive scanning has the benefit of proactively contacting all hosts on a network to determine their status,"), 26 (explaining that "passive scanning is an important part of network situational awareness because it provides completely new information about services not seen by the active scans"); 27 (explaining that "relying on passive and active mapping together is better than relying on only active mapping alone"); EX1023, 57 (2007 Internet Measurement Conference publication explaining that "passive monitoring is well suited for quickly finding popular services" whereas "[a]ctive scanning is better suited to rapidly finding all servers"), 60 (explaining that "[i]n cases where completeness is key, a combination of both [active and passive] methods is advantageous"), 70 ("passive and active service discovery are complimentary methods for discovering services on a network").

173. The resulting Stokes-Crabtree combination would derive a portion of the first graph representation from active and passive reconnaissance data received from multiple computer systems as taught by Crabtree. *E.g.*, EX1005, [0005],

[0007]-[0009], [0049]-[0050], [0057-[0061]. In the combination, the reconnaissance data received from the multiple computer systems—including data gathered passively by one of the computer systems and actively by another one of computer systems—would supply entity and relationship information to the Stokes server. That server, in turn, would use the reconnaissance data to generate a portion of the first graph representation alongside Stokes's event-derived graph data. This would have predictably provided a more comprehensive graph for subsequent attack path identification, as taught by Stokes. EX1004, [0024].

Q. Claim 24

[24] The computer system of claim 18, further configured to execute software instructions stored on nontransitory machine-readable storage media comprising software instructions that analyze an attack path of the plurality of attack paths using a graph analysis algorithm.

174. Stokes-Crabtree satisfies this element. Stokes analyzes lateral movement paths (i.e., attack paths) by processing the network connection graph to identify K-hop paths and subpaths. Stokes then scores, filters, and ranks those paths using the path-rate score, including through inbound and outbound path analysis in forensic mode. EX1004, [0039]-[0040], [0047], [0064]-[0071]. Accordingly, Stokes's graph-based attack path analysis, including K-hop path identification, traversal, scoring, filtering, and ranking, is a type of graph analysis algorithm used to analyze an attack path. As discussed above for elements [18.6] and [18.11],

Sections VIII.O.2 and VIII.O.3 *above*, Stokes-Crabtree teaches a first plurality of attack paths and a second plurality of attack paths, collectively constituting a plurality of attack paths. A POSITA would have recognized that the Stokes-Crabtree system would use Stokes's graph-based attack path analysis (i.e., graph analysis algorithm) to analyze an attack path of the plurality of attack paths.

R. Claim 27

[27] The computer system of claim 18, wherein the representation of the first graph comprises an identification of the third entity as a sensitive resource, and wherein the computer system is further configured to execute software instructions stored on nontransitory machine-readable storage media comprising software instructions that generate a report that identifies the third entity as a sensitive resource.

175. Wherein the representation of the first graph comprises an identification of the third entity as a sensitive resource: As discussed in Section VIII.A, *above*, Stokes teaches a graph-based cybersecurity system in which computers and user accounts are represented as nodes in a network connection graph and entities implicated in suspicious or compromised paths are identified through graph-based analysis. EX1004, [0007]-[0008], [0037]-[0039], [0047]-[0048], [0063], [0072]-[0073]. As discussed above for claim 13, Section VIII.K *above*, Crabtree teaches that graph-based cybersecurity analysis may distinguish resources having heightened security significance, such as sensitive resources. Further, Crabtree teaches incorporating impact assessment information into a graph to add

risk-assessment context to graph-based security analysis. EX1005, [0074], [0076], [0079]-[0081]. For example, Crabtree teaches that "impact assessment scores … may be received and incorporated in the CPG [graph] information, adding risk assessment context to the behavior information." *Id.*, [0076]. Crabtree also teaches that, when an event occurs, the event is "compared against the CPG [graph] to determine the impact," and that this analysis is "enhanced with the inclusion of impact assessment information [] for any affected resources." *Id.*, [0079]. And, Crabtree teaches that some resources present materially greater risk than others, explaining that "an unknown connection to an internal data server with sensitive information may be considered a much higher risk than an unknown connection to an externally-facing web server." *Id*., [0081].

176. A POSITA would have been motivated to incorporate Crabtree's teachings regarding incorporating the impact assessment information into the graph of a Stokes-like system because doing so would have enabled the graph-based analysis to account for whether a given resource was high-risk or sensitive, thereby improving prioritization of suspicious paths for investigation and response. Section VIII.K *above*; EX1004, [0007]-[0008], [0047]-[0048], [0063], [0072]-[0073]; EX1005, [0074], [0076], [0079]-[0081]. And because Crabtree teaches that incorporated impact assessment information provides risk context for affected resources and distinguishes higher-risk resources, such as an internal data server

with sensitive information, a POSITA would have understood that, where that information indicated that a third entity was such a resource, the graph would identify the third entity as a sensitive resource. EX1005, [0076], [0079], [0081].

177. Wherein the computer system is further configured to execute software instructions stored on nontransitory machine-readable storage media comprising software instructions that generate a report that identifies the third entity as a sensitive resource: As discussed above for element [1.12], Section VIII.D.12 *above*, Stokes teaches outputting results that identify entities involved in suspicious paths, and Crabtree teaches presenting results of graph-based cybersecurity analysis in reports, dashboards, alerts, visualizations, or similar user-facing output identifying relevant entities for investigation. EX1004, [0007], [0047]; EX1005, [0063]-[0064], [0076], [0080]-[0081]. Further, as discussed above and for claim, Section VIII.K *above*, the graph-based analysis performed by the Stokes-Crabtree system would identify when the third entity implicated in a suspicious path is a sensitive resource. Stokes further teaches that suspicious paths are reviewed and investigated by an analyst and that network access to certain entities may be disabled when suspicious lateral movement is detected with sufficient confidence. EX1004, [0048], [0062], [0078]. A POSITA therefore would have been motivated to include, in a report output by the Stokes-Crabtree system, an identification that the third entity is a sensitive resource, because doing so would provide an analyst, or an automated

response mechanism, better context for evaluating suspicious paths and deciding whether to disable or restrict access to certain entities.

S. Claim 28

[28] The computer system of claim 18, wherein the representation of the first graph comprises an identification of a security vulnerability associated with the third entity, and wherein an attack path of the second plurality of attack paths is based on the security vulnerability.

178. Stokes-Crabtree satisfies this element. As discussed above for elements [18.6] and [18.11], Sections VIII.O.2 and VIII.O.3 *above*, Stokes teaches a graph-based attack-path analysis framework in which entities identified through the analysis are used to identify suspicious paths. EX1004, [0047]-[0048]; *see also* FIG. 4.

179. Crabtree describes incorporating vulnerability information into a graph and analyzing that information together with graph data. EX1005, [0077]. In particular, Crabtree describes collecting information on publicly disclosed vulnerabilities into a CPG and analyzing the combined data to identify relationships between known vulnerabilities and risks exposed by components of the infrastructure, thereby producing a combined CPG that incorporates both the internal risk level of network resources, user accounts, and devices and the actual risk level based on analysis of known vulnerabilities and security risks. *Id.* Accordingly, Crabtree teaches a graph representation comprising an identification of a security

vulnerability associated with an entity, and graph analysis in which risk associated with graph relationships is evaluated based on identified security vulnerabilities. *Id*.

180. A POSITA would have been motivated to modify Stokes's graph-based attack-path analysis framework to incorporate Crabtree's teachings regarding identification of security vulnerabilities associated with entities, including a third entity represented in the modified graph, because doing so would have predictably improved the system's ability to identify and evaluate attack paths based on whether entities along those paths presented known vulnerabilities that could be exploited by an attacker. A POSITA would have understood that vulnerability information associated with a network entity would be useful in determining whether a lateral-movement path through that entity was plausible, exposed greater risk, or warranted higher priority in the graph-based analysis. Thus, a POSITA would have recognized the obvious use case in which, in the Stokes-Crabtree system, an attack path of the second plurality of attack paths is based on a security vulnerability associated with the third entity.

181. Further, as with elements [1.7]-[1.9] and [1.12] (as well as elements [18.8]-[18.10] which are substantively identical to elements [1.7]-[1.9]), the alleged support identified by Applicant for the amendment adding claim 28 during prosecution consisted solely of disclosure that appears verbatim in Crabtree. In particular, Applicant cited ¶ [98] as support for new claim 37, which corresponds to

claim 28 of the issued patent and is substantively identical to ¶ [0083] of Crabtree. EX1002, 110/290, 234/290; EX1005, [0083]. Accordingly, the disclosure Applicant relied upon as written description support for claim 28 was already disclosed in Crabtree before the alleged invention date.

IX. GROUND 2: CLAIM 3 IS OBVIOUS OVER STOKES IN VIEW OF CRABTREE AND SEKAR

182. As explained below, claim 3 is further obvious over Stokes in view of Crabtree and Sekar.

A. Sekar (EX1006)

183. Sekar is a U.S. patent application filed August 19, 2019, and claiming priority to a provisional application filed August 17, 2018, and thus qualifies as prior art under at least 35 U.S.C. § 102(a)(2). EX1006, Cover. Sekar is analogous art to the '627 patent because they are both in the same field of endeavor of graph-based detection of attack-related activity in computer networks. EX1001, Abstract, 2:52-4:13; EX1006, [0027]-[0034].

184. Sekar discloses graph-based backward analysis for identifying the point of origin or entry point of an attack based on event flows or dependencies in a graph. In particular, Sekar describes processing suspect vertices and tracing backward to reachable source vertices to identify each entry point of an attack, and further describes determining the attacker's initial point of entry through backward analysis before performing forward analysis to assess impact. EX1006, [0239], [0269].

B. Combining Stokes, Crabtree, And Sekar

185. A POSITA would have found it obvious to implement a combination of Stokes and Crabtree, as discussed above at Section VIII.C. A POSITA further would have been motivated, with a reasonable expectation of success, to combine the teachings of Stokes-Crabtree with Sekar's teachings regarding identifying a point of origin of an attack (hereinafter "Stokes-Crabtree-Sekar").

186. First, Sekar, like Stokes and Crabtree, discloses graph-based cybersecurity analysis of attack-related activity. Stokes-Crabtree uses event-derived information to update and analyze graphs of entities and relationships to identify suspicious paths and other attack-related information. EX1004, [0037]-[0040], [0047]-[0048], [0064]-[0071]; EX1005, [0063]-[0064], [0072], [0076], [0080]-[0081]. Sekar likewise discloses graph-based attack analysis, including backward analysis to identify each entry point of an attack and forward analysis to assess impact. EX1006, [0239], [0269]. Sekar thus complements Stokes-Crabtree by providing explicit techniques for identifying a point of origin of an attack.

187. Second, a POSITA implementing Stokes-Crabtree would have recognized the need not only to identify suspicious paths and anomalous activity, but also to determine where that activity originated. Stokes already performs backward analysis of inbound paths in a graph to determine event flows. EX1004, [0048], [0064], [0070]. Sekar expressly teaches determining the initial point of entry

of an attacker through backward analysis of event flows, including source vertices from which suspect vertices are reachable. EX1006, [0239], [0269]. A POSITA thus would have been motivated to incorporate Sekar's backward origin-identification techniques into Stokes-Crabtree to improve identification of the point of origin for anomalous events and attack paths. A POSITA would recognize the benefit of identifying the point of origin of an anomalous event because it would allow for remedial action against such source. *See* EX1004, [0061] (disclosing that "identified malicious computers and/or user accounts can be shut down (i.e., disabled)").

188. Third, combining Sekar with Stokes-Crabtree would have predictably improved the results of the Stokes-Crabtree system. Stokes identifies suspicious attack paths, Crabtree updates the graph used in that analysis and reports results, and Sekar adds explicit backward analysis of event flows to identify an attack entry point. EX1004, [0037]-[0040], [0047]-[0048], [0064]-[0071]; EX1005, [0063]-[0064], [0072], [0076], [0080]-[0081]; EX1006, [0239], [0269]. Applying Sekar's teachings would allow suspicious paths to be traced back to their point of origin, thereby improving investigation of anomalous activity.

189. Finally, a POSITA would have had a reasonable expectation of success in combining Stokes, Crabtree, and Sekar. The references operate in the same technical field of graph-based cybersecurity analysis and rely on compatible inputs, structures, and objectives. EX1004, [0037]-[0040], [0047]-[0048], [0064]-[0071];

EX1005, [0063]-[0064], [0072], [0076], [0080]-[0081]; EX1006, [0239], [0269]. Because Sekar's point of origin identification techniques would have been applied to the same type of graph-based cybersecurity system disclosed in Stokes-Crabtree, incorporating Sekar's teachings would involve the predictable use of known techniques in a closely related system, with no technological incompatibility or undue experimentation required. A POSITA therefore would have reasonably expected that combining Sekar's techniques for identifying an attack point of origin with the Stokes-Crabtree framework would successfully yield a graph-based cybersecurity system capable of more effective root-cause analysis of anomalous activity.

C. Claim 3

[3] The system of claim 2, wherein identifying the attack path further comprises: identifying a point of origin for the first anomalous event based on the first plurality of event flows.

Stokes-Crabtree-Sekar satisfies claims 1 and 2, from which claim 3 depends, for the same reasons as Stokes-Crabtree. *See* Sections VIII.D and VIII.E *supra*.

190. As explained in Section VIII.E, *supra*, Stokes determines event flows associated with malicious lateral movements in a graph, including analysis of inbound paths. EX1004, [0048], [0064], [0070].

191. Sekar discloses identifying a point of origin for an attack based on event flows. EX1006, [0239]. In particular, Sekar describes backward analysis in which

the system aims to identify each entry point of an attack, where an entry point is a source vertex, and processes nodes to find source vertices from which a suspect vertex is reachable. *Id*. Sekar further describes determining the initial point of entry of the attacker using backward analysis and then moving forward to assess the impact of the campaign. *Id.*, [0269].

192. A POSITA would have been motivated to apply Sekar's backward, flow-based origin-identification teachings in the Stokes-Crabtree system to improve identification of the point of origin for anomalous events. Stokes already performs backward analysis of inbound paths in a graph to determine event flows. EX1004, [0048], [0064], [0070]. Incorporating Sekar's backward origin-identification techniques would have predictably improved root-cause analysis of anomalous activity in the Stokes-Crabtree system.

I, John R. Black, Jr., do hereby declare and state, that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Dated: April 7, 2026



John R. Black, Jr.

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 *7/08–present*

Research

Cryptography, Network Security, Computer Security.

Past Employment

SecureSet LLC *9/15–7/18*
Vice President of Education

University of Colorado at Boulder *7/02–7/08*
Assistant Professor of Computer Science.

University of Nevada, Reno *7/00–6/02*
Assistant Professor of Computer Science.

University of California, Davis *7/97–7/00*
Research Assistant

University of California, Davis *8/95–6/97*
Teaching Assistant

Ingres Corporation *6/88–4/94*
Senior Member of Technical Staff

Education

University of California, Davis *9/95–9/00*
Ph.D. in Computer Science.
Thesis: Message Authentication Codes.
Advisor: Phillip Rogaway.

California State University at Hayward *9/84–6/88*
B.S. in Computer Science and Mathematics, 1988.
Honors: Summa Cum Laude

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

PAN Exhibit 1024

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.

PAN Exhibit 1024

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.

PAN Exhibit 1024

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.

PAN Exhibit 1024

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. NCIIA 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.

PAN Exhibit 1024

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.
Joesph 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.

PAN Exhibit 1024

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.

PAN Exhibit 1024

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, SECRYPT 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, EUROCRYPT 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.

PAN Exhibit 1024

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.

PAN Exhibit 1024