

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

MICROSOFT CORPORATION,
Petitioner,

v.

QOMPLX LLC,
Patent Owner.

IPR2026-00326
U.S. Patent No. 12,301,628
Issued: May 13, 2025
Application No. 18/892,295
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.

TABLE OF CONTENTS

	Page
I. INTRODUCTION AND ENGAGEMENT	4
II. BACKGROUND AND QUALIFICATIONS.....	4
III. MATERIALS CONSIDERED AND INFORMATION RELIED UPON REGARDING '628 PATENT	9
IV. UNDERSTANDING OF PATENT LAW	11
V. LEVEL OF SKILL IN THE ART, AND STATE OF THE ART	15
A. Person Of Ordinary Skill In The Art.....	15
B. State Of The Art	16
VI. THE '628 PATENT	17
VII. CLAIM CONSTRUCTION	20
VIII. GROUND 1: CLAIMS 1, 4-6, 9-11, 13-15, 17, 20, AND 22 ARE OBVIOUS OVER BREZINSKI (EX1004) AND CRABTREE (EX1005).....	21
A. Brezinski.....	21
B. Crabtree	27
C. Combining Brezinski And Crabtree.....	28
D. Claim 1	35
1. Element [1.1].....	35
2. Element [1.2].....	35
3. Element [1.3].....	37
4. Element [1.4].....	41
5. Element [1.5].....	45
6. Element [1.6].....	48
7. Element [1.7].....	54

8.	Element [1.8].....	56
9.	Element [1.9].....	61
10.	Element [1.10].....	65
11.	Element [1.11].....	77
12.	Element [1.12].....	82
13.	Element [1.13].....	86
E.	Claim 4	87
F.	Claim 5	88
G.	Claim 6	92
H.	Claim 9	94
I.	Claim 10	94
J.	Claim 11	96
1.	Elements [11.1]-[11.5], [11.8], [11.11] And [11.15]-[11.17]..	96
2.	Elements [11.6]	97
3.	Elements [11.7]	100
4.	Elements [11.9]	101
5.	Elements [11.10]	101
6.	Elements [11.12]	102
7.	Elements [11.13]	103
8.	Elements [11.14]	103
K.	Claim 13	104
L.	Claim 14	105
M.	Claims 15 & 17.....	105
N.	Claim 20	106
O.	Claim 22	106

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. 12,301,628 (hereinafter “the ’628 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, 4-6, 9-11, 13-15, 17, 20, and 22 (each a “Challenged Claim” and collectively the “Challenged Claims”) of the ’628 patent would have been obvious to a person having ordinary skill in the art (“POSITA”) as of April 22, 2021. 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 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 '628 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. 12,301,628 ("' 628 patent ")
1002	File History of U.S. Patent No. 12,301,628
1004	U.S. Patent No. 9,225,730 (" Brezinski ")

No.	Description
1005	U.S. Patent App. Pub. No. 2018/0219919A1 (“Crabtree”)
1006	U.S. Patent App. Pub. No. 2022/0014560A1
1007	U.S. Patent App. Pub. No. 2020/0296137A1
1008	William Eberle & Lawrence Holder, <i>Incremental Anomaly Detection in Graphs</i> , IEEE 13 th Int’l Conference on Data Mining Workshops (IEEE 2013)
1009	Leman Akoglu <i>et al.</i> , <i>Graph Based Anomaly Detection Description: A Survey</i> , Data Mining and Knowledge Discovery, Vol. 29, Issue 3 (Springer 2014)
1010	Listing and Comparison of ’628 Patent Challenged Claims
1011	Plaintiff Qomplx’s Preliminary Infringement Contentions, <i>Qomplx LLC v. Microsoft Corp.</i> , Case No. 25-CV-01383-ADA (W.D. Tex. Nov. 24, 2025)
1012	Defendant’s Preliminary Claim Constructions, <i>Qomplx LLC v. Microsoft Corp.</i> , Case No. 25-CV-01383-ADA (W.D. Tex. Mar. 2, 2026)
1013	Plaintiff Qomplx LLC’s Preliminary Disclosure of Proposed Claim Constructions, <i>Qomplx LLC v. Microsoft Corp.</i> , Case No. 25-CV-01383-ADA (W.D. Tex. Mar. 2, 2026)
1014	Microsoft Computer Dictionary, 5 th Ed. (2002)
1015	U.S. Patent App. Pub. No. 2019/0081876A1
1016	U.S. Patent App. Pub. No. 2018/0069885A1
1017	U.S. Patent No. 10,909,241
1018	U.S. Patent App. Pub. No. 2018/0367548A1
1019	Thomas H. Cormen, <i>et al.</i> , <i>Introduction to Algorithms</i> , 3d Ed. (MIT Press 2009) (excerpt)
1020	Redline Comparison of Crabtree to ’628 Patent Specification

No.	Description
1021	Complaint For Patent Infringement, <i>Qomplx LLC v. Microsoft Corp.</i> , Case No. 25-CV-01383-ADA (W.D. Tex. Aug. 28, 2025)
1022	U.S. Patent App. Pub. No. 2020/0177617A1
1023	<i>Breadth-First Search</i> , Carnegie Mellon Univ. School of Computer Science (Apr. 29, 2015), available at: https://www.cs.cmu.edu/afs/cs/academic/class/15210-s15/www/lectures/bfs-notes.pdf

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 '628 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 '628 patent or any assignee of the '628 patent that any secondary considerations would rebut the obviousness of any Challenged Claim of the '628 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, AND STATE OF 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 '628 patent of April 22, 2021. 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 '628 patent in April 2021 ("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. The state of the art, immediately below, identifies certain technologies within the knowledge and skill of a POSITA.

B. State Of The Art

37. Graph structures have long been used to represent and analyze data, including entities and their behaviors. *See* EX1008, 522 (“Much 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 omitted). Such techniques have been applied to computer networks more than a decade before the ’628 patent’s filing date. *Id.*, 521 (2013 IEEE publication 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”). 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. *See id.* (“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”).

VI. THE '628 PATENT

38. The '628 patent, titled “Correlating Network Event Anomalies Using Active And Passive External Reconnaissance To Identify Attack Information” issued on May 13, 2025. The '628 patent issued from U.S. Patent Application No. 18/892,295 (the “'295 application”), filed on September 20, 2024, which is a continuation of Application No. 17/237,346 (the “'346 application”), filed on April 22, 2021. The '628 patent alleges priority through a series of continuation-in-part applications to an application filed October 28, 2015.

39. The '628 patent describes a “system and method for correlating network event anomalies to identify attack information, that identifies anomalous events within the network, identifies correlations between anomalies and other network events and resources, [and] generates a behavior graph describing an attack pathway derived from the correlations.” EX1001, Abstract.

40. The '628 patent discloses “mapping 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., 23:58-65. This “cyber-physical graph of the organization... compris[es] nodes representing entities associated with the organization and edges representing relationships between entities associated with the organization.” Id., 3:16-20. Figure 22 of the ’628 patent depicts an example cyber-physical graph representing “the network devices comprising an organization's network infrastructure as nodes (also called vertices) in the graph and the physical or logical connections between them as edges between the nodes.” Id., 28:17-21, Fig. 22.

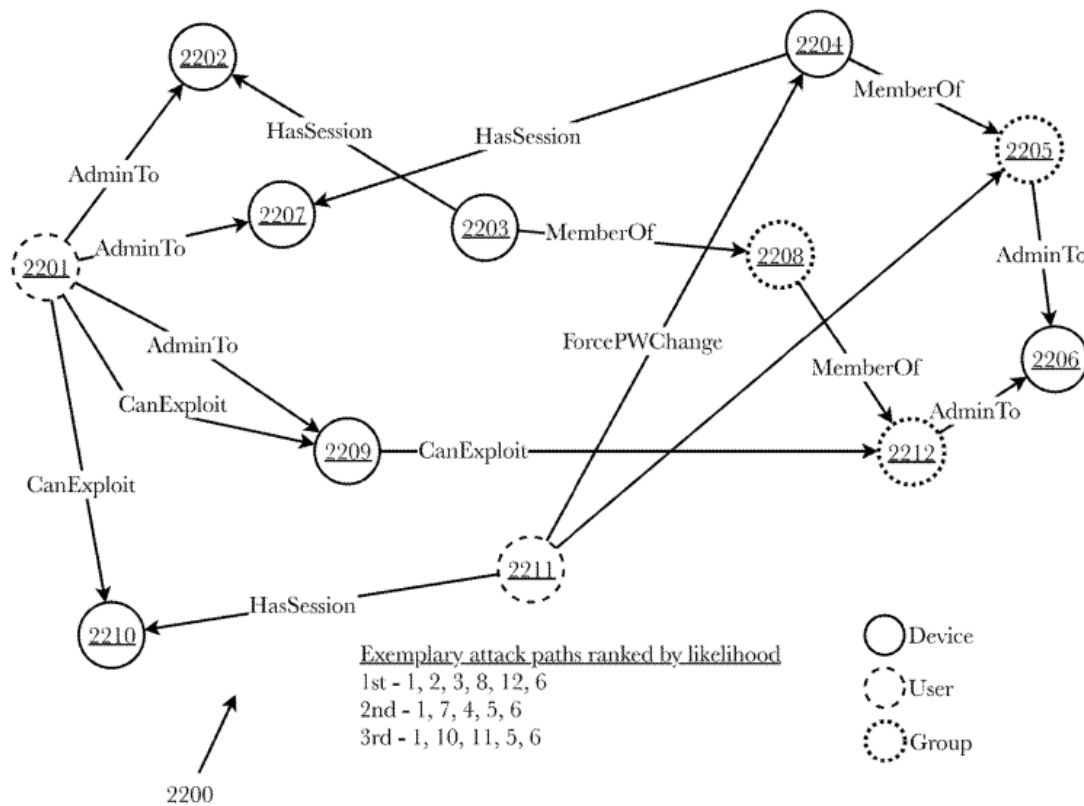


Fig. 22

Id., Fig. 22.

41. The '628 patent discloses receiving time-series information and “updating CPG information as changes occur and events are logged.” Id., 24:5-8. Subsequently, “[t]his information may then be used 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., 24:8-16 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., 23:43-55.

42. The '628 patent also teaches its CPG including information for evaluating and identifying security risks in the network. Specifically, the CPG may contain “behavior analytics information” concerning behavior patterns in the network and analyzed to recognize deviations from established behavior patterns. Id., 22:46-23:16, 23:65-24:1. The CPG may also contain “impact assessment scores” reflecting access capabilities of user accounts in a network, “representing the potential risk should that account be compromised.” Id., 23:17-30, 24:1-5.

43. The '628 patent further describes a “method for identifying correlations between an anomaly and other network events and resources.” Id., 21:9-11. The '628 patent describes the following:

According to this method, when an anomaly is identified 3010..., it may be correlated with other events, network resources, behavior patterns, or any other available information that may be contextually linked to the anomaly. Using a CPG for the network, correlations may be identified by first comparing resources involved in the anomaly 3020, for example any machines that participated in the anomalous event or any databases or other resources that were accessed. This information may then be used to further correlate with other events involving those resources 3030, such as events taking place at the same (or nearly the same) time, or events with a recognizable causative relationship such as non-anomalous events that may have been triggered by the anomaly.

Id., 21:11-25

VII. CLAIM CONSTRUCTION

44. Unless otherwise expressly discussed below, I apply the plain and ordinary meaning of all claim terms. I understand that neither petitioner nor patent owner have proposed a construction for any claim term, other than the parties agreeing that “account” be given its “plain and ordinary meaning.” EX1012, 2; EX1013, 4.

**VIII. GROUND 1: CLAIMS 1, 4-6, 9-11, 13-15,
17, 20, AND 22 ARE OBVIOUS OVER
BREZINSKI (EX1004) AND CRABTREE (EX1005)**

45. As explained below, claims 1, 4-6, 9-11, 13-15, 17, 20, and 22 are obvious over Brezinski and Crabtree.

A. Brezinski

46. Brezinski (EX1004) issued on December 29, 2015, from an application filed March 19, 2014, and is prior art under at least 35 U.S.C. § 102(a)(1). EX1004, 1. Brezinski was not cited during prosecution of and is not listed on the face of the '628 patent.

47. Brezinski discloses a computing environment comprising a plurality of host computing devices. *Id.*, Fig. 1, 4:39-48. The networked host devices connect to an analysis device that collects event data concerning events in the network, such as connections between host devices, processes, services, and other entities (*id.*, Fig. 1, 1:18-22; Abstract), and uses the collected data “to generate a graph that interrelates or correlates a plurality of events” (*id.*, 2:31-32). Figure 10 of Brezinski (reproduced below), for example, “depicts a flow diagram of a process for generating a graph that correlates multiple events present in the event data.” *Id.*, 1:54-56.

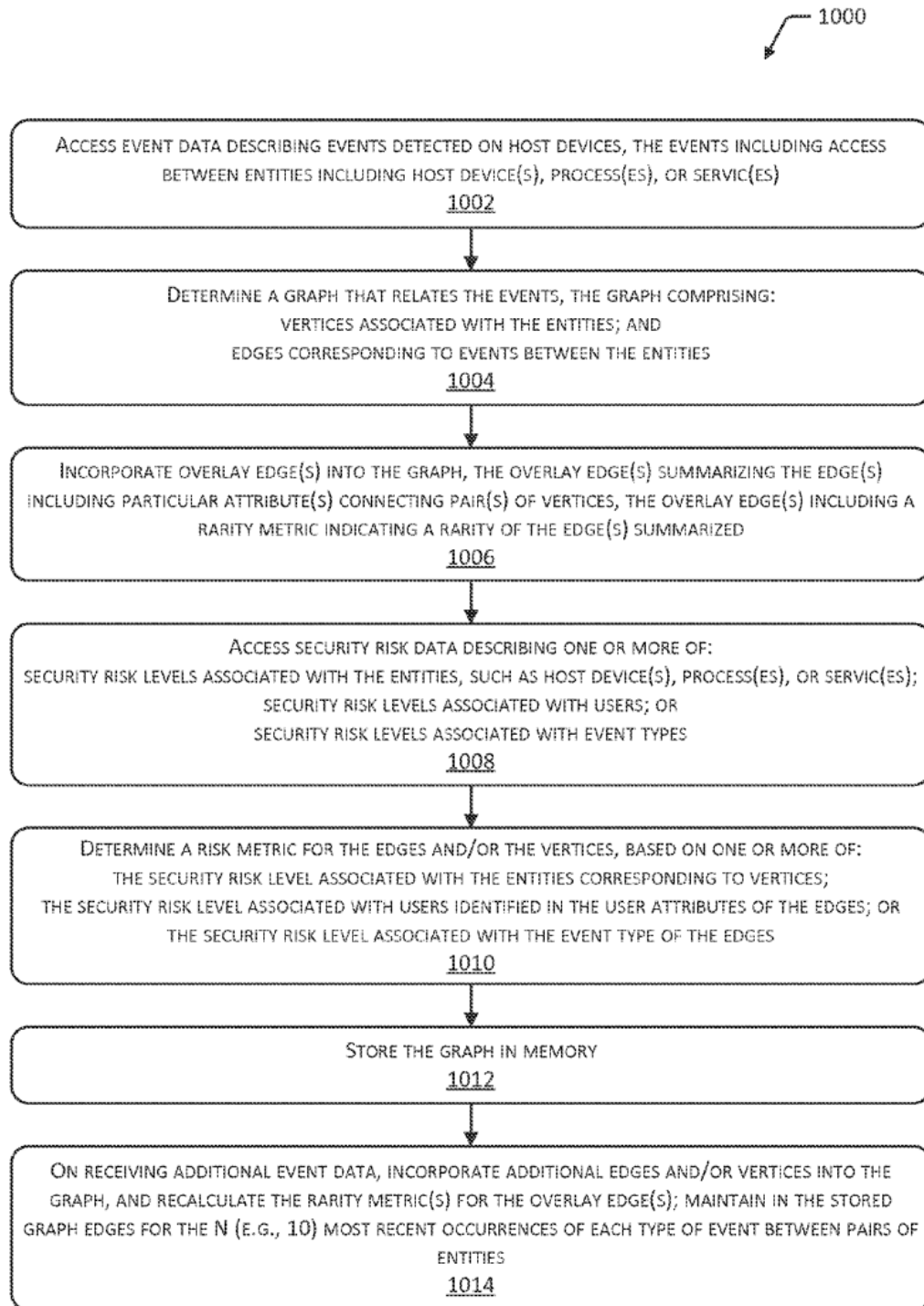


FIG. 10

Id., FIG. 10.

48. Brezinski explains that “the collected event data may be employed to generate a graph that includes a plurality of vertices [i.e., nodes], each vertex corresponding to an entity (e.g., a host device, process, or service) in a computing environment” as well as “any number of edges, each edge connecting two vertices to indicate an event involving the two entities corresponding to the two connected vertices.” *Id.*, 2:45-53. An example of such a graph of network entities and relationships is shown in Brezinski’s Figure 4, reproduced below, showing an example of a graph 118 having multiple nodes or vertices 402(1)-402(10) connected by respective directed edges 404(1) representing events, such as connections or interactions, between the vertices. *Id.*, Fig. 4, 1:27-31, 8:29-62.

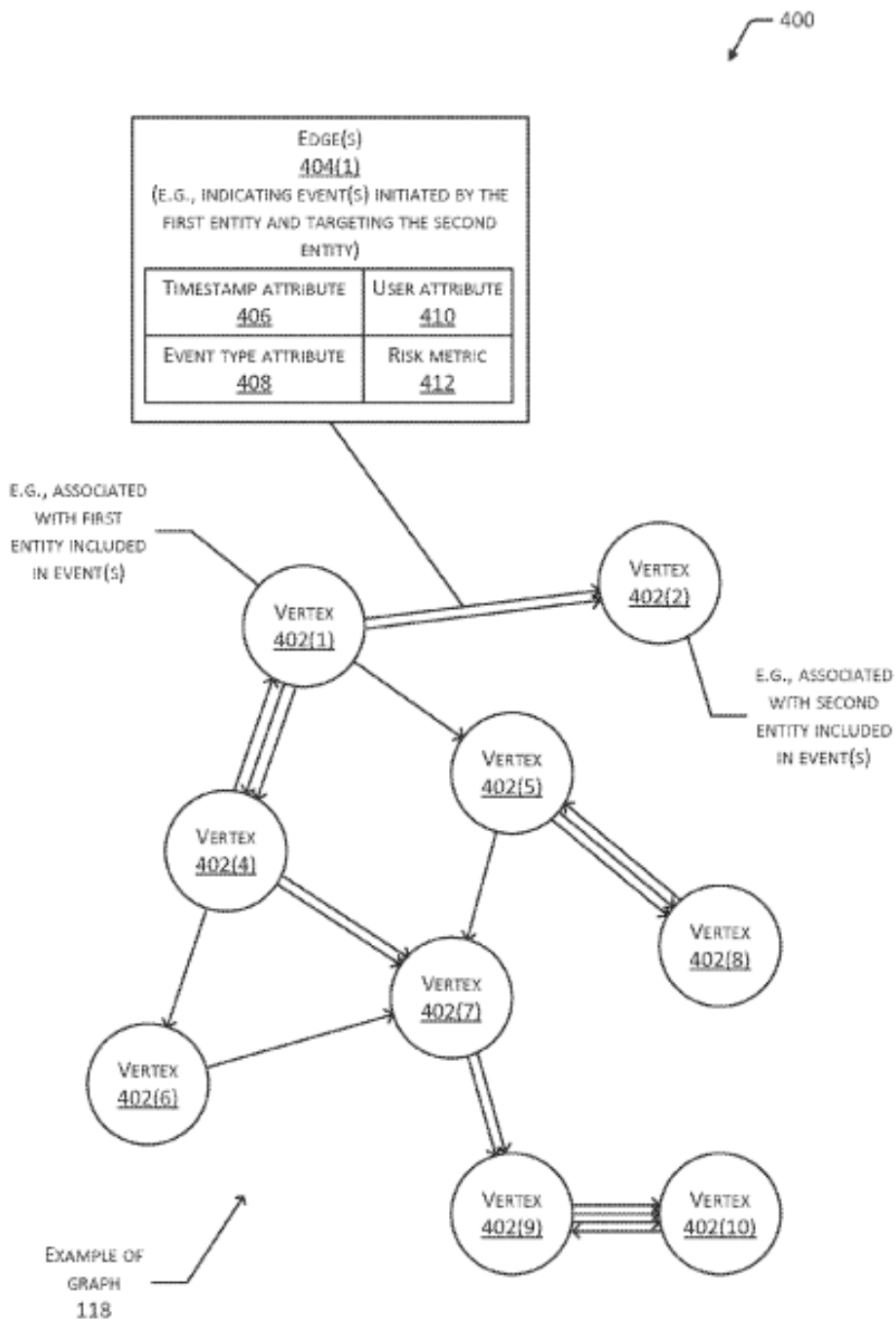


FIG. 4

Id., Fig. 4.

49. Brezinski discloses using its graph structure to identify anomalous behavior in the network, as “the graph may be traversed to identify anomalous

activity data included in the graph.” *Id.*, 3:52-58. It describes starting at one vertex and then traversing “at least a portion of the edges in the graph to identify those edges that include” rarity and risk metrics that satisfy predetermined thresholds. *Id.* Such identified edges and their connected vertices are added to a subset graph representing anomalous activity data. *Id.*, 3:64-67, 12:12-14.

50. Brezinski further teaches that “one or more additional criteria may be employed to determine whether the edges 404 or vertices 402 are to be included in the subset of the graph 118 designated as the anomalous activity data 122.” *Id.*, 12:35-38. For example, “an edge 404 may be included in the anomalous activity data 122 if it connects a vertex 402 that is associated with a particular host device 102 or other entity” such as a designated honey pot. *Id.* 12:42-47. Figure 1 of Brezinski, reproduced below, depicts Brezinski’s computing environment, showing an analysis device 114 collecting event data from a plurality of host devices, using graph generation module 116 to generate the graph as explained above, and using a graph analysis module 120 to analyze the graph to identify edges and nodes constituting anomalous activity data and representing the anomalous activity data in a subset graph. *Id.* Fig. 1 (reproduced below), 4:39-40, 5:20-23, 8:5-6, 8:23-27, 11:12-18, 11:26-28.

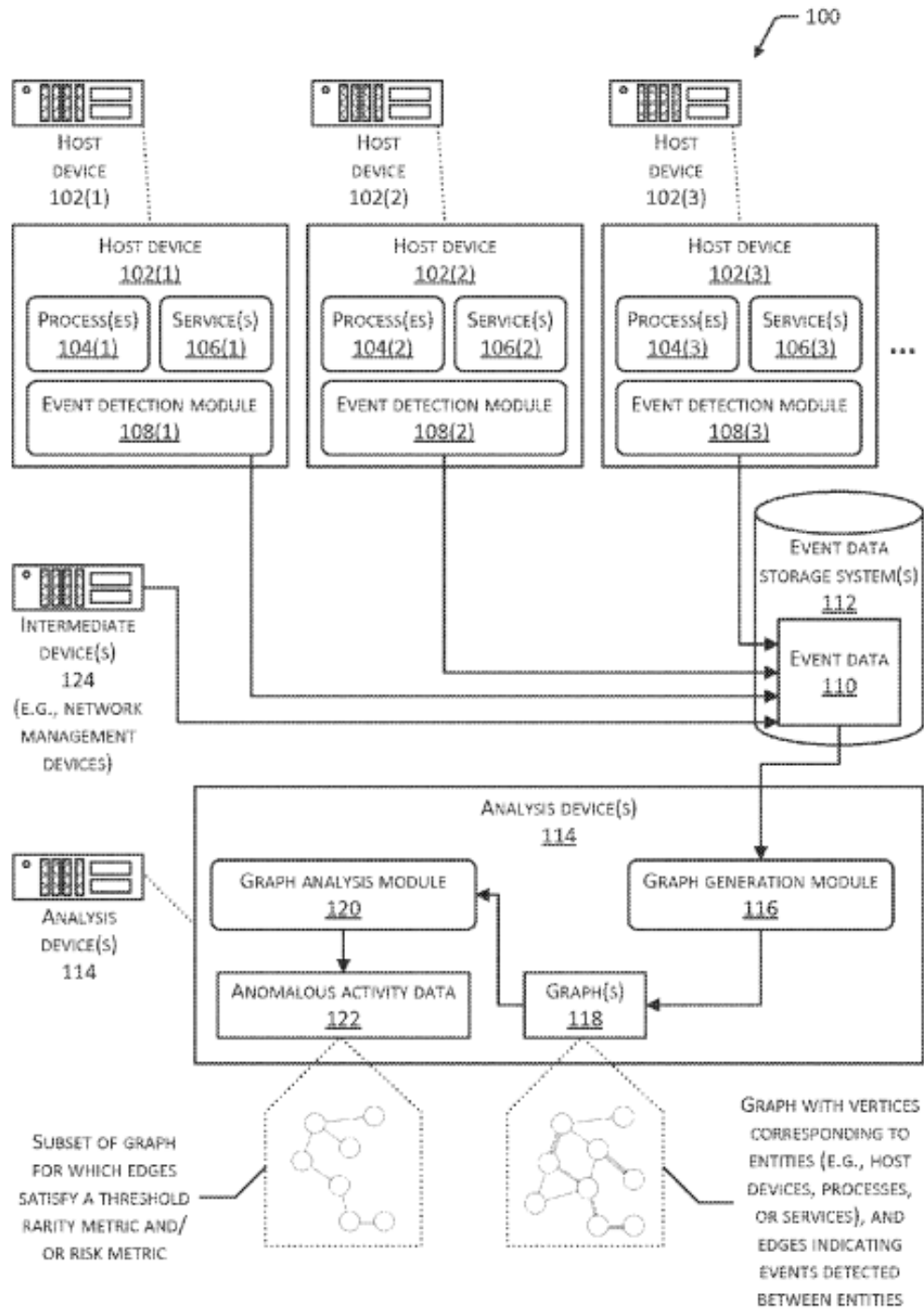


FIG. 1

B. Crabtree

51. Crabtree (EX1005) published on August 2, 2018, from an application filed November 27, 2017 (EX1005, Cover) is prior art under at least Section 102(a)(1) because the earliest effective filing date of the Challenged Claims is not supported under 35 U.S.C. § 112 by any priority application filed within a year of Crabtree's August 2018 publication date.

52. Crabtree lists as inventors two of the '628 patent's three inventors, and Crabtree issued from U.S. Patent Application No. 15/823,363 (EX1005, Cover), which is an ancestor to the '628 patent through several continuation-in-part applications (EX1001, 1:9-24).

53. Crabtree shares many of the same disclosures as the '628 patent. *See* Section VI, *above*; EX1020 (redline comparison showing changes between Crabtree specification and '628 patent specification). Like the '628 patent, Crabtree discloses “mapping 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.” EX1005, [0076]. Crabtree discloses receiving time-series information and “updating CPG information as changes occur and events are logged.” *Id.* Subsequently, “[t]his information may then be used 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.* 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].

54. Also like the '628 patent, Crabtree teaches its CPG including information for evaluating and identifying security risks in the network. Specifically, the CPG may contain “behavior analytics information” concerning behavior patterns in the network and analyzed to recognize deviations from established behavior patterns. *Id.*, [0072]-[0073], [0076]. The CPG may also contain “impact assessment scores” reflecting access capabilities of user accounts in a network, “representing the potential risk should that account be compromised.” *Id.*, [0074], [0076].

C. Combining Brezinski And Crabtree

55. As explained below, POSITAs would have been motivated with a reasonable expectation of success to implement a Brezinski-like system—with its network graphs and graph-based anomaly detection—with Crabtree’s teachings of receiving and analyzing streaming data. A POSITA would have been motivated to

combine Brezinski with certain additional Crabtree teachings, namely Crabtree's teachings of 1) using of graph nodes to represent user accounts; 2) identifying an anomalous event by comparing one or more events to patterns of normal event behavior; and 3) including in a graph information about security vulnerabilities associated with graphed entities.

56. First, Brezinski and Crabtree address closely related problems using similar graph-based cybersecurity analysis, and provide complementary teachings. Brezinski describes collecting event data from a networked computing environment, generating a graph in which vertices correspond to entities and edges correspond to events or interactions, and traversing that graph to identify anomalous activity based on risk and rarity metrics. *See* Section VIII.A, *above*; EX1004, Abstract, 2:21-25, 2:45-48, 2:60-64, 3:32-33, 3:52-58, 8:29-37. Figs. 4-7. Crabtree likewise describes generating and updating a graph of entities and relationships in a networked security environment, including graph visualizations of users, servers, devices, and other resources, based on time-series and event information, for the purpose of identifying cybersecurity risks, anomalous behavior, and attack-related information. *See* Section VIII.B, *above*, EX1005, [0072]-[0076].

57. Second, a POSITA implementing Brezinski would have recognized the benefit of combining Brezinski's graph-based anomaly detection based on time-stamped event data with Crabtree's teachings of receiving such data in a streaming

manner. EX1004, 17:23-28. Crabtree, in turn, notes known benefits of continuous network assessment using streaming data, such as allowing “live assessment of attacks as they occur” and “proactive and high-speed reactive defense capabilities against a variety of cyberattack threats.” EX1005, [0073], [0079]. A POSITA understood the benefits, if not criticality, of promptly detecting anomalous behavior indicating potential cybersecurity attacks. *See* EX1015, [0003] (explaining that in “security-related applications..., **the ability to identify data anomalous data** (i.e., unusual data records or data patterns) **fairly quickly may be critical**”); And a POSITA further knew that real-time analysis of streaming data would allow a system to more quickly identify anomalies. EX1015, [0003] (explaining that it is “beneficial” to capture and analyze “streaming data [] in real time” to meet the critical need of identifying anomalous data quickly); EX1016, [0004]-[0005] (explaining that “more effective and timely detection of cyber threats are achieved through application of the teachings described herein,” including receiving and extracting “parameters associated with a computer network event... wherein the parameters are extracted in real-time”); EX1017, 2:39-41 (“[P]erforming analytics with low latency and rapid results with streaming data is needed when finding relevant security events and being operationally aware in real-time.”); Given this, a POSITA would have been motivated to implement a Brezinski-like system to receive and analyze network event data in a live, continuous (i.e., streaming) manner

as taught by Crabtree to timely and effectively identify and respond to anomalous activities including potential security threats. A POSITA would have further recognized that implementing Brezinski's graph-based anomaly detection using streaming data, as taught by Crabtree, would allow the system to incrementally update its graph representation as events are observed, rather than relying on delayed or batch processing of accumulated event logs. This enables the system to incorporate newly observed entities and relationships into the graph more quickly, ensuring that subsequent graph-based analyses are performed on a more current and complete representation of network activity. A POSITA would have understood that such timely incorporation of event data improves both the accuracy and responsiveness of anomaly detection.

58. Third, the teachings of Crabtree follow the same core principles as Brezinski. Both references model cybersecurity-relevant environments as graphs of nodes/entities and edges/relationships (*e.g.*, EX1004, Abstract, 2:21-25; EX1005, [0076]; both analyze those graphs to identify anomalous or attack-related behavior (*e.g.*, EX1004, Abstract, 3:52-53; EX1005, [0072]; and both are directed to helping security personnel understand and respond to suspicious network activity (EX1004, 12:24-27; EX1005, [0074], [0076]. Therefore, a POSITA seeking to improve Brezinski's graph-based anomaly detection would naturally have looked to

Crabtree's teachings would have found those teachings directly applicable to Brezinski's graph-generation and graph-analysis processes.

59. Fourth, combining Crabtree with Brezinski would have predictably improved the results of Brezinski's anomaly-detection system. Brezinski seeks to detect anomalous activity by identifying suspicious subgraphs within a network event graph. EX1004, 11:16-24, 11:54-57, 12:12-24. Crabtree teaches graphing particular logical relationships, such as access privileges for network users, and further teaches incorporating behavior analytics and impact assessment information into the graph. EX1005, [0072]-[0073] (behavioral analytics information), [0074] (impact assessment scores), [0076], [0080]. A POSITA would have understood that enriching Brezinski's graph with Crabtree's additional contextual information including behavioral analytics and impact assessment scores would enable more informative and accurate anomaly analysis, enabling additional ways to identify anomalous activity, and providing additional context concerning the risk profile of entities in the network. *Id.* This would have improved Brezinski's ability by both "enabling rapid response to threats to mitigate any potential impact" as well as "identify[ing] points of weakness" within the network that would benefit from additional security measures. EX1005, [0073], [0080].

60. Finally, a POSITA would have had a reasonable expectation of success in combining Brezinski and Crabtree. The two references operate in the same

technical field of graph-based cybersecurity analysis and rely on highly compatible inputs, structures, and objectives. *See* EX1004, Abstract; EX1005, [0072]-[0074], [0076]. Brezinski collects event data, generates a directed graph of entities and events, and analyzes that graph to identify anomalous activity. *See* EX1004, Abstract, 2:21-25, 2:45-48, 2:60-64, 3:32-33, 3:52-58, 8:29-37. Crabtree likewise uses continuously collected time-series and event information to generate and analyze a graph of users, devices, servers, and other resources for cybersecurity monitoring and risk assessment. EX1005, [0075]-[0076]. Because Crabtree's live-monitoring and contextual graphing techniques would have been applied to the same type of graph-based security system already disclosed in Brezinski, a POSITA would have understood that incorporating Crabtree's teachings into Brezinski involved the predictable use of known techniques in a closely related system, with no technological incompatibility or undue experimentation required.

61. A POSITA would further reasonably expect success in combining Brezinski with Crabtree's data streaming because streaming data processing was commonly implemented in similar cybersecurity systems. EX1015, [0003], [0015]; EX1016, [0004]-[0005]; EX1017, 3:20-38; EX1018, [0023], [0038] (explaining that in a system for the "tracking of [a] malicious lateral movement across [a] computer network," "events may be processed in real-time" and that such a system modeling network behavior as a graph "may be implemented in a streaming environment.")

Brezinski already receives and processes newly received event data to update its event graph and analyze recently detected events using its analysis device (EX1004, 8:23-29, 17:23-29), and Crabtree expressly teaches continuous real-time monitoring, receiving streaming data, and incorporation of events into visualizations (e.g., graphs) as they occur (EX1005, [0050], [0075]). Moreover, Crabtree does not require any specific hardware, structure, or algorithm for receiving streaming data, explaining that such functionality is performed simply by black-box “modules” and using “standardized protocols for converting streams of information into graph representations of that data” (EX1005, [0050]), and it would therefore be well within a POSITA’s capability to implement the same streaming functionality of Crabtree in the analysis device of Brezinski.

62. So implemented, the combined Brezinski-Crabtree system includes the computing environment of Brezinski receiving event data comprising event descriptions of reflecting interactions between network entities with such event data/descriptions being received in a continuous, streaming manner as taught by Crabtree.

63. Further motivations to combine Brezinski and Crabtree are discussed for individual claim elements below.

D. Claim 1**1. Element [1.1]¹**

[1.1] A computer system comprising:

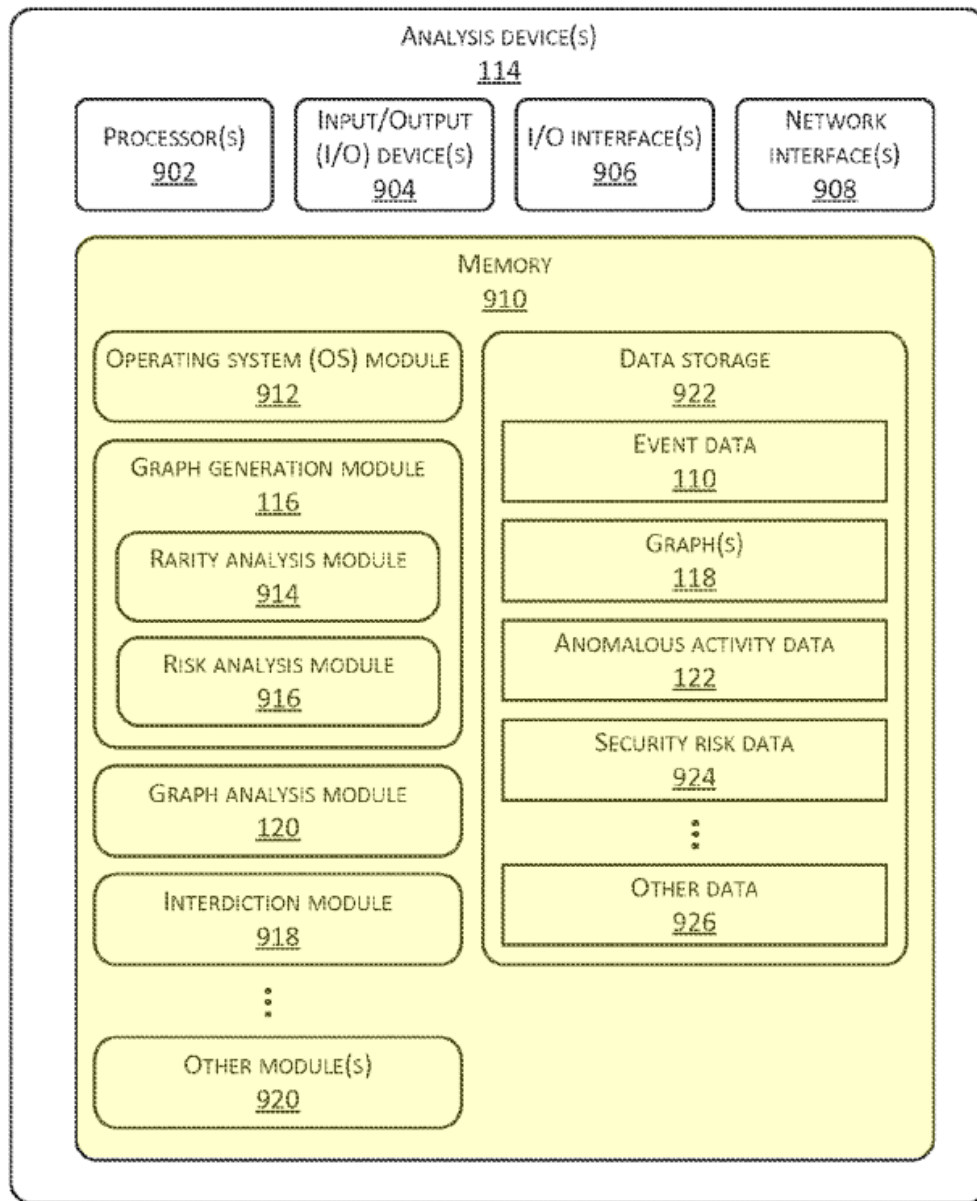
64. Brezinski-Crabtree satisfies the preamble. Brezinski discloses one or more “analysis devices” (*computer system*) that “may comprise any type of computing device.” *E.g.*, EX1004, 1:18-22, 4:40-41, 8:6-7, Figs. 1, 9; *compare with* EX1001, 34:10-42 (describing “computer system”).

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:

65. Brezinski discloses that each analysis device includes a memory 910 and a processor 902 “configured to execute one or more stored instructions.” EX1004, 15:7-20; *id.*, 19:50-20:20 (describing “non-transitory computer-readable storage media having stored thereon instructions”). Brezinski Fig. 9 shows details of the analysis device(s), including memory 910 and processor 902:

¹ Reference numbers in the format of [claim#.limitation#] are added throughout for ease of reference.



Id., Fig. 9 (highlighting added); *see also id.*, 14:23-34, 15:19-21, 23:13-24-50 (CRM claims). It further discloses modules “executing on the analysis device(s) 114, such as the graph generation module 116 and the graph analysis module 120,” among other executable modules. *Id.*, Fig. 9, 15:28-57. Consistently, Brezinski describes performing its disclosed methods using “instructions” stored on “non-transitory

computer-readable storage media.” *Id.*, 19:51-60. The subsequent sections describe functions performed on Brezinski’s analysis device and a POSITA would therefore have understood such functions to be performed by analysis device’s software instructions for performing its functions.

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,

66. Brezinski discloses this limitation. Its analysis devices store graph(s) 118 in “memory 910,” as shown in Figure 9. *Id.*, 15:58-59, 15:65-67, Fig. 9; *above* Section VIII.D.2. Exemplary FIG. 10 describes “a process for generating a graph 118” performed by “modules executing on the analysis device(s)” (EX1004, 16:15-21) that includes “stor[ing] the graph in memory” (*id.*, 17:15-22; FIG. 10 (Step 1012)). *See generally, id.*, 16:15-17:38 (process for generating, storing, and modifying graph).

67. The stored graph includes plural “vertices associated with the entities” (exemplary *representations of a first plurality of nodes corresponding to a first plurality of entities*) and “edges corresponding to events between the entities” (exemplary *representations of a first plurality of edges*). *Id.*, FIG. 10 (Steps 1004, 1012), 16:22-36. Brezinski discloses that each graph “includes a plurality of

vertices” (*nodes*) with “each vertex corresponding to an entity” (*plurality of nodes corresponding to a plurality of entities*). *Id.*, 2:45-48; 2:31-37 (explaining how “graph is a representation” that represents each object as a vertex/node and each relationship as an edge). Brezinski specifies a “node” as an exemplary “vertex” (*e.g.*, EX1004, 2:35-36) and thus a POSITA would understand any disclosure of vertex/vertices as a disclosure of node/nodes, consistent with the interchangeability of those terms in the art (*e.g.*, EX1001, 7:47-49, 28:17-21; EX1004, 2:35-36). The terms “vertex” and “node” are used interchangeably herein, as was commonly done in the relevant art. *E.g.*, EX1001, 7:47-49 (defining a “graph” as “a representation of information and relationships, where each primary unit of information makes up a ‘node’ or ‘vertex’ of the graph”); 28:17-21 (“A cyber-physical graph, in its most basic form, represents the network devices comprising an organization's network infrastructure as nodes (also called vertices) in the graph”). EX1004, 2:35-36 (“A graph may represent each of the plurality of objects as a vertex (*e.g.*, a node).”

68. Brezinski teaches “a graph that includes a *plurality of vertices, each vertex corresponding to an entity* (*e.g.*, a host device, process, or service) in [the] computing environment.” *Id.*, 2:45-48 (emphasis added); *see also id.*, Abstract. “A pair of vertices in the graph may be connected by any number of edges corresponding to any number of events” between the connected vertices, such as a login, network connection, or resource or process access from one entity to another.

Id., 2:21-25, 2:60-64, 3:32-33. Brezinski Fig. 4, reproduced below, depicts an exemplary graph with a first plurality of nodes (vertices 402) connected by a first plurality of edges (edges 404). *Id.*, Fig. 4, 8:29-37.

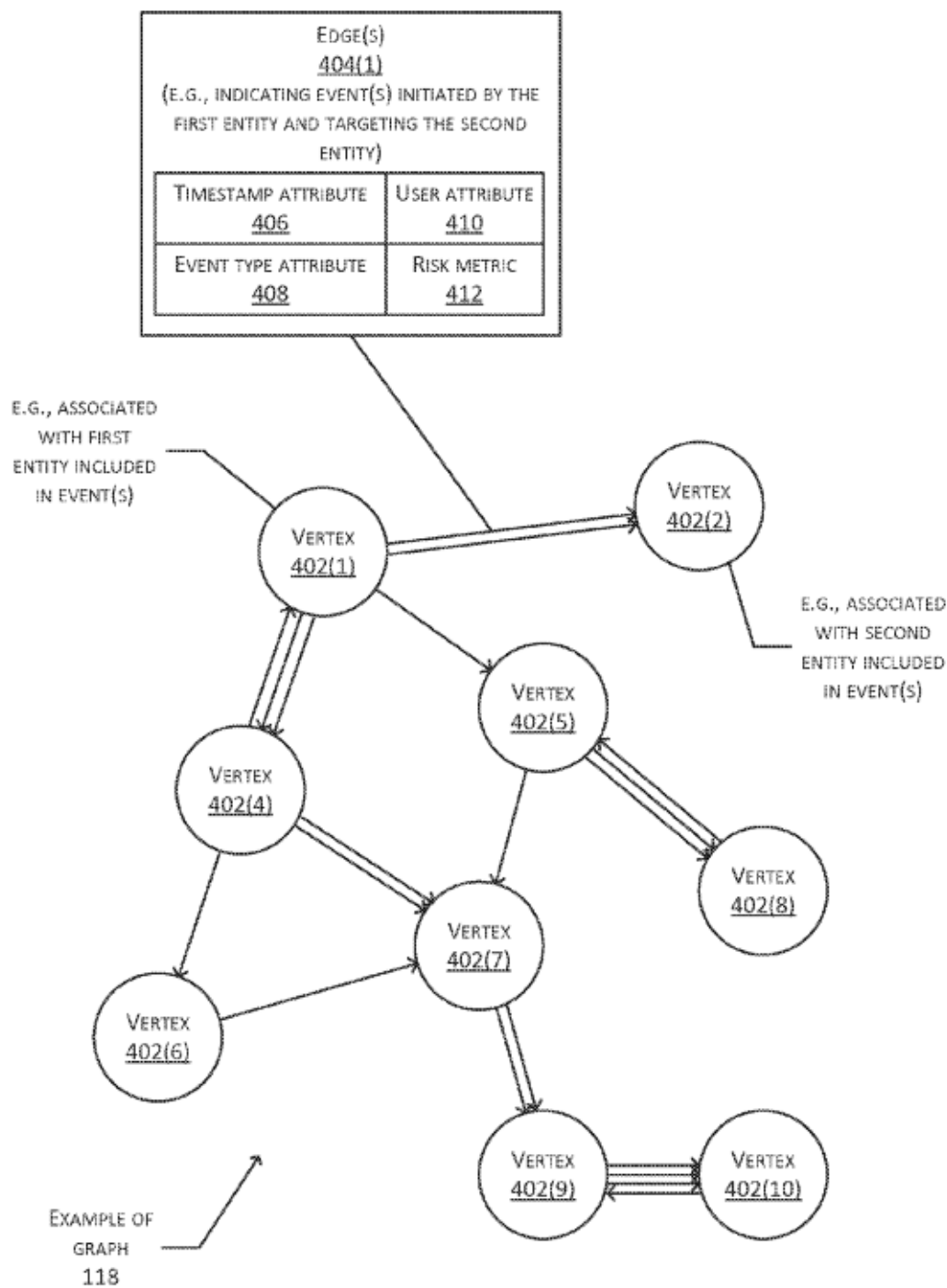


FIG. 4

69. Finally, Brezinski teaches that “the graph 118 may be a directed graph” and FIG. 4 depicts a directed graph, with edges including a directionality. *See id.*, Fig. 4, 8:53-56 (“Each edge 404(1) may include a directionality that indicates which entity initiates the event and which entity is targeted by the event. In this way, the graph 118 may be a directed graph.”); *see also id.*, 2:42-44.

4. Element [1.4]

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

70. Brezinski teaches this element. Brezinski discloses that its graph nodes represent entities such as host devices, processes, services, and files. EX1004, 2:45-48 (“each vertex corresponding to an entity (e.g., a **host device, process or service**)”) (emphasis added). Each such entity is a resource consistent with how the ’628 patent’s description thereof. EX1001, 24:9-12 (describing “servers, devices, and other resources” including “a user’s personal computer or a smartphone”). Brezinski therefore teaches graphs wherein *the first plurality of entities comprises a plurality of resources*.

71. Brezinski also teaches nodes “corresponding to other types of entities, such as **users**.” EX1004, 2:49-51 (emphasis added). A POSITA understood Brezinski’s description of a “user” to be a user account because a POSITA recognized that an “account” is the way in which users are tracked in a network.

EX1021 (dictionary defining an “account” as “[t]he record-keeping mechanism used by networks... for keeping track of authorized users” and “to validate users.”). In other words, a computer network defines a given user through that user’s account, and a POSITA would understand Brezinski’s recitation of a node corresponding to a user to correspond to that user’s account. This is consistent with Brezinski’s description of a “user ID that identifies one or more particular users” or “user credentials.” EX1004, 7:41-45, 16:59-60 (“users or user groups identified by the user ID(s)”), 9:39-41, 10:31. At the very least, user accounts would have been the most obvious design choice for tracking user entities in the Brezinski system. This understanding is consistent with the ’628 patent specification’s repeated descriptions of an “account” as a user account. *See* EX1001, Fig. 9 (“Use DCG to analyze **user accounts** and determine access capabilities” and “[g]enerate impact assessment score for each account”), 23:16-21, 24:29-33 (“This produces a combined CPG 1204 that incorporates both the internal risk level of network resources, **user accounts**, and devices... based on the analysis of known vulnerabilities and security risks.”), 24:36-38 (“[T]ime-series data... may be collected 1301 for **user accounts**.”) (emphasis added) and describing “users” as entities represented by the graph (*e.g.*, EX1001, 27:27-29 (“entities ... for example, devices, users, ...”), 28:8-13 (same), 19:48-50, 24:8-10). Indeed, the only exemplary graph presented in the ’628 patent depicts nodes corresponding to “users.” *Id.*, Fig. 22, 28:37-40 (“[T]he cyber-physical

graph 2200 contains 12 nodes (vertices) comprising: ... two users designated by dashed-line circles.”).

72. Alternatively, if patent owner argues that Brezinski’s teachings on users/credentials are insufficient, Crabtree teaches a graph that includes “the internal risk level of network resources, **user accounts**, and devices.” EX1005, [0077] (emphasis added). Crabtree teaches using time series data “**for user accounts**” to produce “a graph visualization of users, servers, devices, and other resources,” thereby including users as graph entities alongside other network resources. EX1005, [0075]-[0076], [0078] (emphasis added). Accordingly, Crabtree teaches a graph whose nodes represent entities comprising a plurality of accounts and resources.

73. A POSITA implementing Brezinski’s graph would have been motivated to include nodes representing user accounts along with nodes representing resources because Brezinski already discloses including nodes representing “users” (EX1004, 2:49-50) and Crabtree demonstrates that a “user account[]” is an obvious way of representing a user in a graph-based cybersecurity analysis EX1005, [0077]. And a POSITA would include in Brezinski’s graph both accounts and resources in order to provide a holistic view of the entities acting within a monitored computer network so as to maximize coverage of the anomaly detection method of Brezinski. Including user accounts as a class of entity would allow the Brezinski graph to

identify anomalous activity associated with interactions performed by specific user accounts that might otherwise be missed, thereby improving Brezinski's anomalous activity identification. *Id.* Crabtree, further, recognizes that graphed user accounts may include "high risk accounts" having a high "potential risk should that account be compromised" (EX1005, [0074]) and a POSITA would therefore understand the importance of tracking such entities in the graph of Brezinski in order to identify anomalous events involving such high risk accounts and be able to take timely action in response.

74. Moreover, a POSITA was familiar with presenting heterogeneous entity types (e.g., both users and resources) in a single graph as a POSITA recognized that this provided a holistic view of all the actors and assets within a network and their interactions. EX1022, Fig. 3, [0056] (describing Fig. 3 as a network topology graph including nodes and edges, with nodes corresponding to different entities, including computers, users, files, and keys).

75. Further, a POSITA would understand it to be an obvious use case for the software instructions of Brezinski-Crabtree to store a graph having nodes corresponding to both a *plurality* of accounts and a *plurality* of resources. Brezinski states that its graph "may include any number of vertices." EX1004, 2:37-39. And a POSITA recognized that computer networks included many entities accessing and operating on the network. EX1004, 1:6-8 ("Providers of online services or other

computing services may deploy a large number of computing devices to execute processes and services within a computing environment.”); EX1009, 673 (recognizing that a network graph of a large network includes a “large number of compute nodes” and corresponding “large number of edges” between such nodes); EX1018, [0004] (recognizing that a network’s “computer connection history can be large in scale, with hundreds of thousands computers in datasets each generating an average of 200-300 connection events in a typical day”), EX1022, [0056], Fig. 3 (depicting network graph including pluralities of different entity types). Accordingly, a POSITA would recognize an obvious use case of a stored graph reflecting a network with many different nodes which would therefore likely contain at least two (a plurality of) nodes representing resources and at least two (a plurality of) nodes representing accounts.

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 of the first plurality of entities;

76. Brezinski teaches this element. Brezinski explains that its graph includes “edge(s) 404(1) that each correspond to a particular event” between the two nodes/vertices connected by the edge. EX1004, 8:37-45 (further describing “edge(s) ... connecting the pair of vertices” for events occurring “between each pair of vertices”). Graphed events reflect relationships of varying types between a first

entity and a second entity, including, among others, “a network connection,” “a communication over a network,” “a login,” or an “access to a resource” or “access . . . to a process.” *Id.*, 6:45-46, 6:48-49, 6:65-66, 7:4-5, 7:21-22. Each such type of relationship is indicated in “event type 210” data included in an “event description 202,” which in turn is part of “event data 110.” *Id.*, 5:20-23, 5:66-6:1, 6:40-44. Fig. 4, reproduced below, depicts an example of Brezinski’s graph structure with a plurality of edges (404(1)) (represented as directed arrows) connecting respective pairs of entities (represented as vertices 402(1)-402(10)).

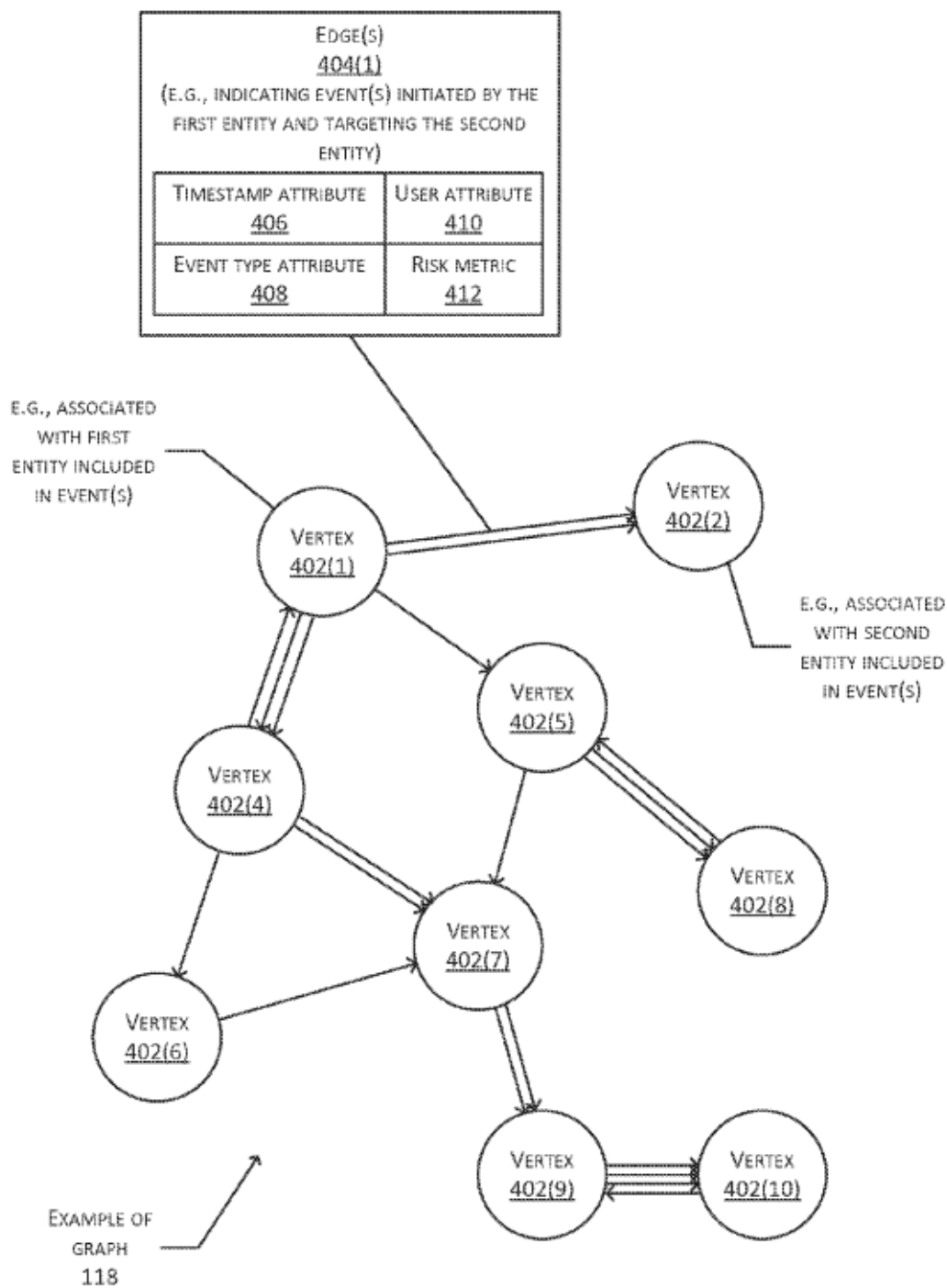


FIG. 4

Id., Fig. 4, 8:43-48.

77. Brezinski's claims confirm the same mapping, stating that the graph includes "a plurality of edges," where "individual ones of the plurality of edges indicat[e] an event by connecting two of the plurality of vertices corresponding to the ... entities included in the event." *Id.*, 20:49-52 (claim 1). Thus, for each edge in Brezinski's graph, the edge represents (i) a specific event/interaction—i.e., a relationship—and (ii) that relationship is between the particular pair of entities whose corresponding vertices the edge connects.

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;

78. Brezinski-Crabtree satisfies limitation 1.6. Brezinski discloses its analysis device(s) 114 receiving a flow of event data from host devices 102, as shown in the following annotated Figure 1:

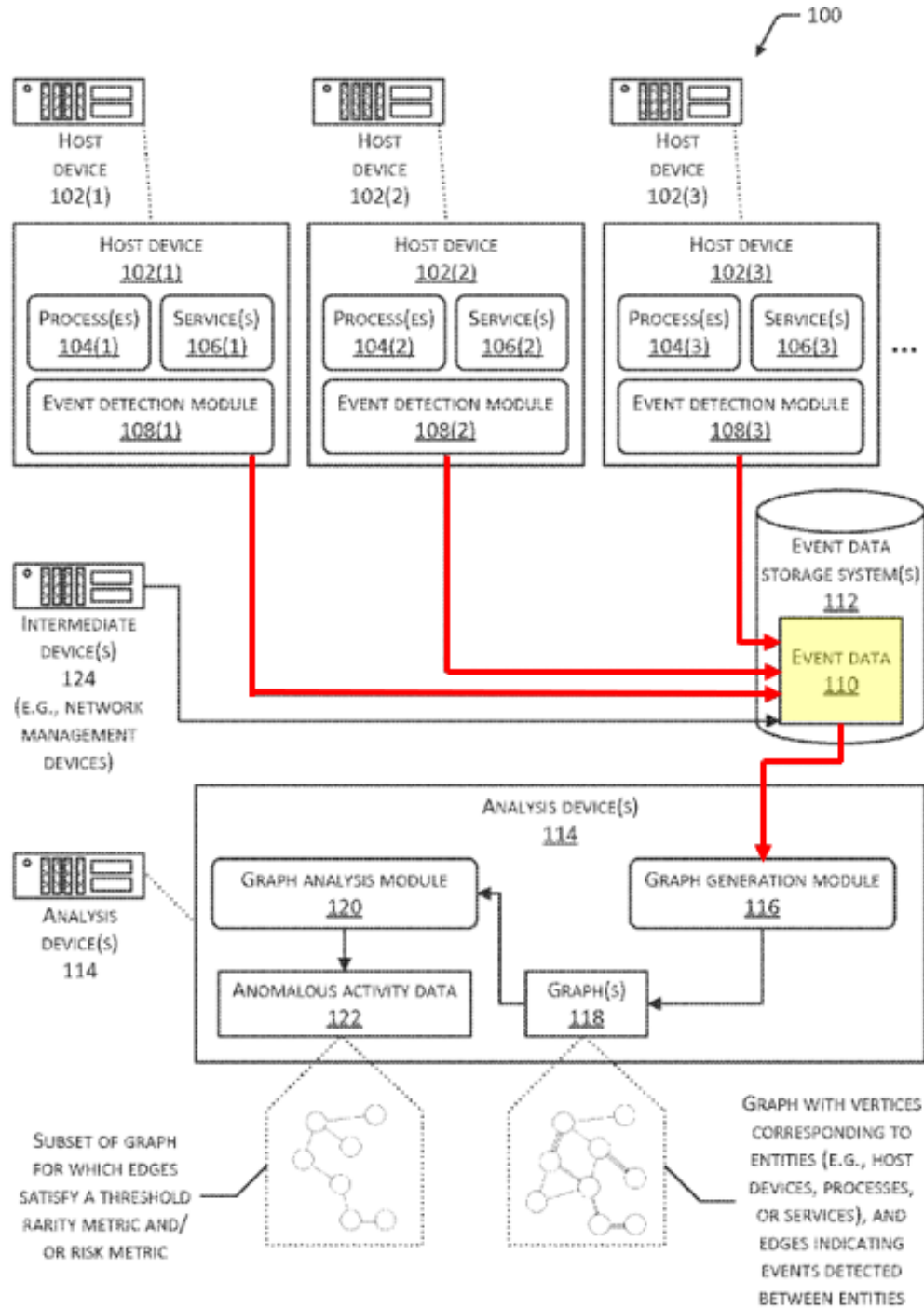
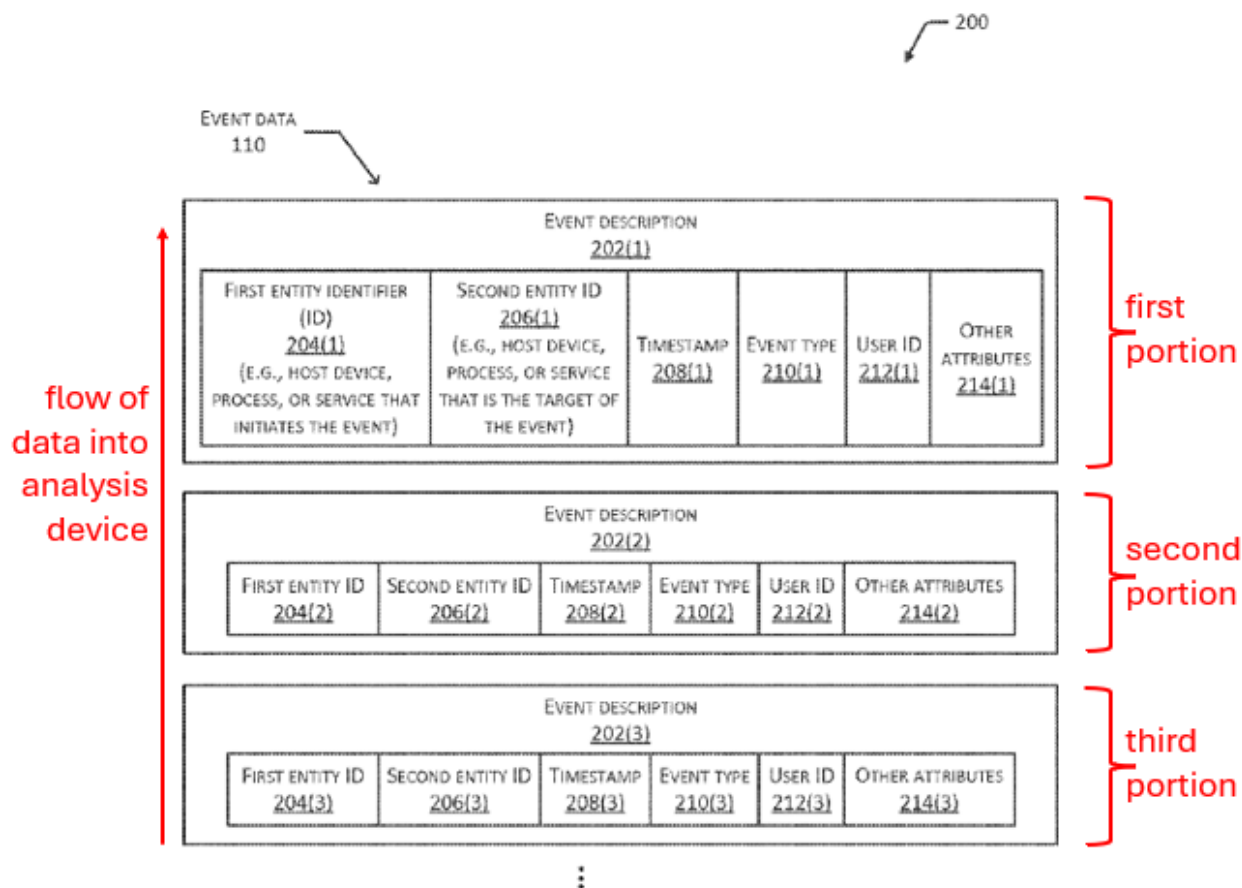


FIG. 1

Id., Fig. 1 (annotated). FIG. 1 (annotated); EX1004, 5:45-48, 8:23-27, 17:23-26. Brezinski also teaches implementing event data storage 112 within analysis device 114. *Id.*, 12:63-66.

79. Brezinski discloses exemplary event data as an ongoing series of event descriptions 202 that each includes a “first entity ID 204,” “second entity ID 206,” “timestamp 208, and “event type 210” (*time-stamped data about events relating to one or more entities*). *Id.*, 5:65-6:10, 6:37-44. Brezinski FIG. 2 shows example event descriptions 202(1-3), with an ellipsis indicating that the flow of event descriptions continues beyond those depicted:



EX1004, FIG. 2 (annotated); *see also id.*, 5:20-23 (describing “event data 110 describing one or more events that occur on the host device(s) 102”).

80. Brezinski also contemplates the obvious use case of this flow including event descriptions that relate to entities already represented as nodes in the graph (*events relating to one or more entities of the first plurality of entities*). For example, Brezinski describes “additional event data” that leads to incorporating “additional edges” into the graph (EX1004, 17:23-26)—a POSITA would understand that these additional edges are between pre-existing nodes that correspond to known entities, as the description makes no mention of new nodes/entities. Brezinski reinforces this understanding by going on to describe “cases where the recently detected events involve entities not already described by vertices 402 in the graph.” EX1004, 6:26-29. Since some “cases” of additional events involve new entities, a POSITA would understand that other cases would involve entities already represented in the graph. By describing that in some cases additional event data will lead to incorporation of an additional node/vertex to represent a new entity, this conveys to the reader that in at least some other cases additional event data will not involve new entities, i.e., will involve entities already in the graph and thus of the “first plurality of entities”

Crabtree Teachings on “Receiving Streaming Data”

81. Brezinski recognizes the need to keep its graph current to conditions of a network, teaching to modify its graph to incorporate additional edges and vertices

in response to “recently detected events.” *Id.*, 17:23-28. Crabtree expressly discloses “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]. Crabtree further describes “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]. A POSITA would have understood Crabtree’s disclosures of live, continual collection and intake of network activity data to be streaming data because it constitutes a “data transmission... that occurs in a continuous flow.” EX1014, 499 (computer dictionary defining a data “stream”)

82. Because Crabtree’s streamed data is “received continuously,” a POSITA would modify Brezinski’s system such that event data is provide to Brezinski’s analysis device in a continuous manner such that events are captured and analyzed as soon as an event occurs. Accordingly, in a streaming implementation of Brezinski, as soon as an event in the network is captured, the analysis device receives event data comprising a single event description describing the details of

the event. *Id.* With reference to Brezinski's Figure 2, such a system would receive event data 110 continuously, with individual event descriptions (202(1), 202(2), etc.) received for corresponding events as those events occur in the network. EX1004, Fig. 2, 5: 64-6:1. This would allow the combined Brezinski-Crabtree system to provide "live monitoring" (EX1005, [0075]) of the events of its system and to incorporate such events into the graph accordingly for rapid analysis and reaction.

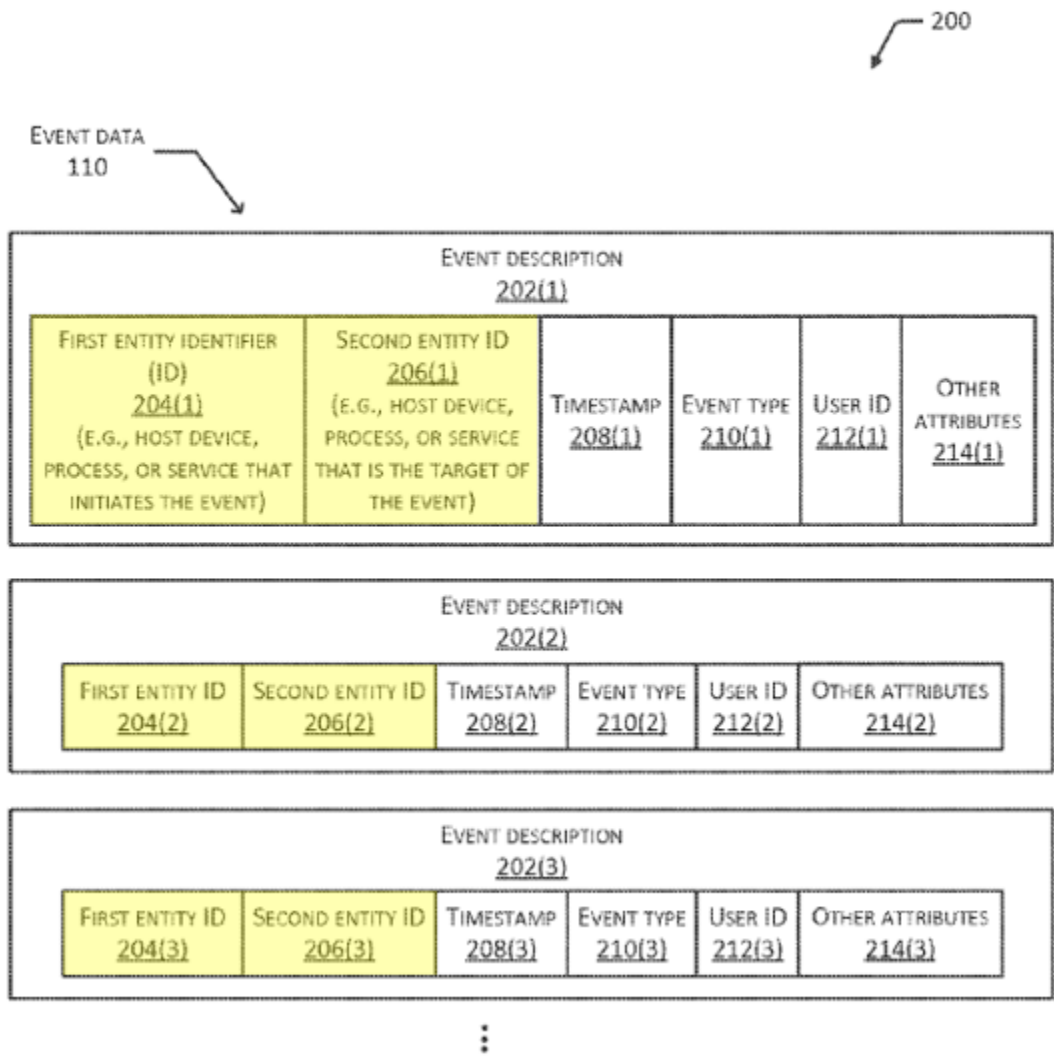
83. Crabtree's disclosure of streaming data matches the disclosures that the '628 patent Applicant relied on during prosecution. In amending the pending independent claims to recite "receiv[ing] streaming data...", the Applicant identified such amendment to be supported by "¶¶[93] and [94] of the as-filed application." EX1002, 214/279. With regard to receiving data, those cited paragraphs of the as-filed application describe receiving "time-series data (as described above, referring to Fig. 10)." *Id.*, 91/279. That as-filed applications description of time-series data and Fig. 10 is found in ¶[90] of the as filed application, which includes the exact same disclosures concerning live, continuous data collection as paragraph [0075] of Crabtree, relied on above. *See* EX1002, 89-90/279; EX1005, [0075]; EX1020.

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;

84. Brezinski-Crabtree satisfies this limitation. As explained above, for element [1.6], Section VIII.D.6 *above*, the Brezinski-Crabtree analysis device receives streaming event data comprising an ongoing series of event descriptions (each a *portion of the streaming data*).

85. Brezinski explains that its event description 202 in received event data “may include a first entity identifier (ID) ... identifying the entity ... that initiates the event” and a “second entity ID ... identifying the entity ... that receives the event.” EX1004, 6:1-7. The respective first ID and second ID within event data used to identify entities is depicted in Figure 2, reproduced below.



Id., Fig. 2 (annotated).

86. After storing the initial graph in memory (*above* Section VIII.D.3), the Brezinski analysis device uses “additional event data” to identify “entities not already described by vertices 402 in the graph 118” (*a first entity that does not correspond to any of the first plurality of nodes*). EX1004, 17:23-29; *see also id.*, 6:1-8 (describing ID “identifying the entity”), FIG. 2 (showing entity identifiers/IDs in the event description data). Brezinski describes modifying the graph to

incorporate a new vertex/node newly detected entities (EX1004, 17:26-29) and a POSITA would thus understand that this involves identifying any new entity that appears in an incoming event description. As Brezinski teaches event data comprising event descriptions (*above* Section VIII.D.6), a POSITA would understand that this new entity identification is based on an event description (*first portion of the streaming data*). And since the new entity was “not already described” in the graph (EX1004, 17:23-29), it is *not of the first plurality of entities*, which comprises, e.g., those entities represented in the graph before arrival of the additional event data. Accordingly, in the Brezinski-Crabtree system, after initially constructing a network graph, the analysis device will receive and process streaming data including an event description for a new event (i.e., a first portion of streaming data) and, in doing so, will identify a new entity not associated with an existing graph node. *Id.*; Further, as an initially graphed set of nodes represents the first plurality of entities, any later-identified new entities are *not of the first plurality of entities*.

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;

87. Brezinski-Crabtree satisfies this limitation as explained below. The Brezinski-Crabtree system receives streaming event data comprising an ongoing

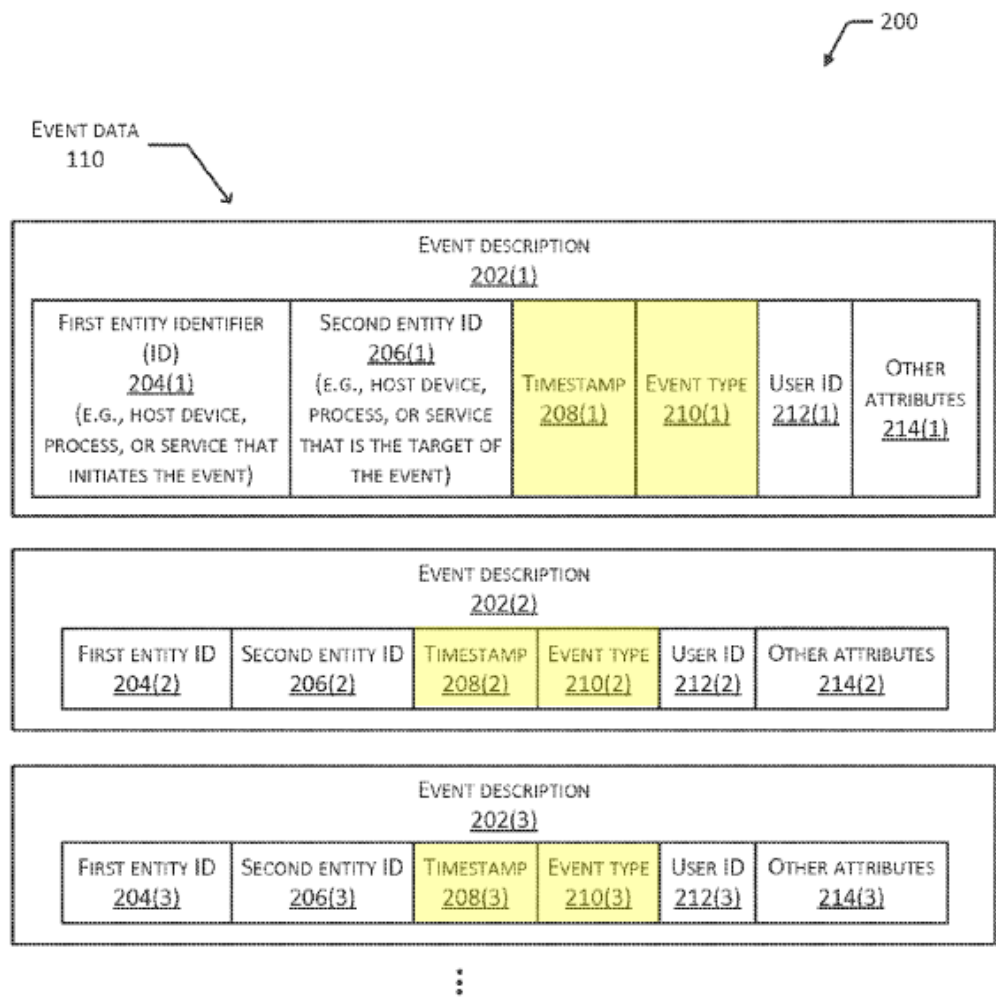
series of event descriptions (each a *portion of the streaming data*). Above Section VIII.D.6.

“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”

88. Brezinski discloses identifying a new edge corresponding to a new relationship between previously identified entities (i.e., the first plurality of entities). Brezinski explains that “on receiving additional event data ... the graph 118 may be modified to incorporate one or more additional edges 404 corresponding to recently detected events.” EX1004, 17:23-26. Such “additional edges” are by their name not pre-existing edges and a POSITA would understand that their creation happens when the analysis device identifies a new event in the additional event data (*identify a first relationship that does not correspond to any of the first plurality of edges*). And because Brezinski describes incorporating additional edges without mention of adding new nodes (EX1004, 17:23-26), a POSITA would understand that these additional edges reflect a relationship between existing nodes-entities (*a first relationship between a pair of entities of the first plurality of entities*). In any event, such a scenario would be an obvious use case given Brezinski’s teachings on processing event data and additional event data. That is, POSITA would recognize the obvious use case of such new edge connecting entities corresponding to nodes in the initial graph (i.e., the first plurality of entities) as explained for element [1.3],

above Section VIII.D.3. For example, Brezinski discloses multiple edges connecting the same pair of entity-nodes, with each edge corresponding to an event/relationship (e.g., network connections) between those two entities, suggesting a newly identified edge being added between existing nodes. *E.g.*, EX1004, FIG. 4, 8:43-52 (describing “multiple edges [] connecting the pair of vertices”).

89. As explained above, for element [1.6], Section VIII.D.6 *above*, Brezinski-Crabtree system receives streaming event data comprising event descriptions for respective events as they occur. Brezinski’s event description includes data used to define the edges of its cybersecurity graph, including an event type and a timestamp. EX1004, 2:53-62 (“An edge may include a timestamp attribute indicating a time (e.g., and date) when the event occurred.... An edge may also include a type attribute indicating a type of the event corresponding to an edge.”), 6:37-44 (“The event description 202 may include a timestamp 208 indicating a date, a time, or both a date and time when the event occurred. The timestamp 208 may be described in any format. The event description 202 may also include an event type 210 that indicates a type, category, or class of event.”); 10:10-20 (describing Example 1). Brezinski’s Figure 2, reproduced below, shows that the event description data corresponding to entity type and a timestamp used for defining an edge.



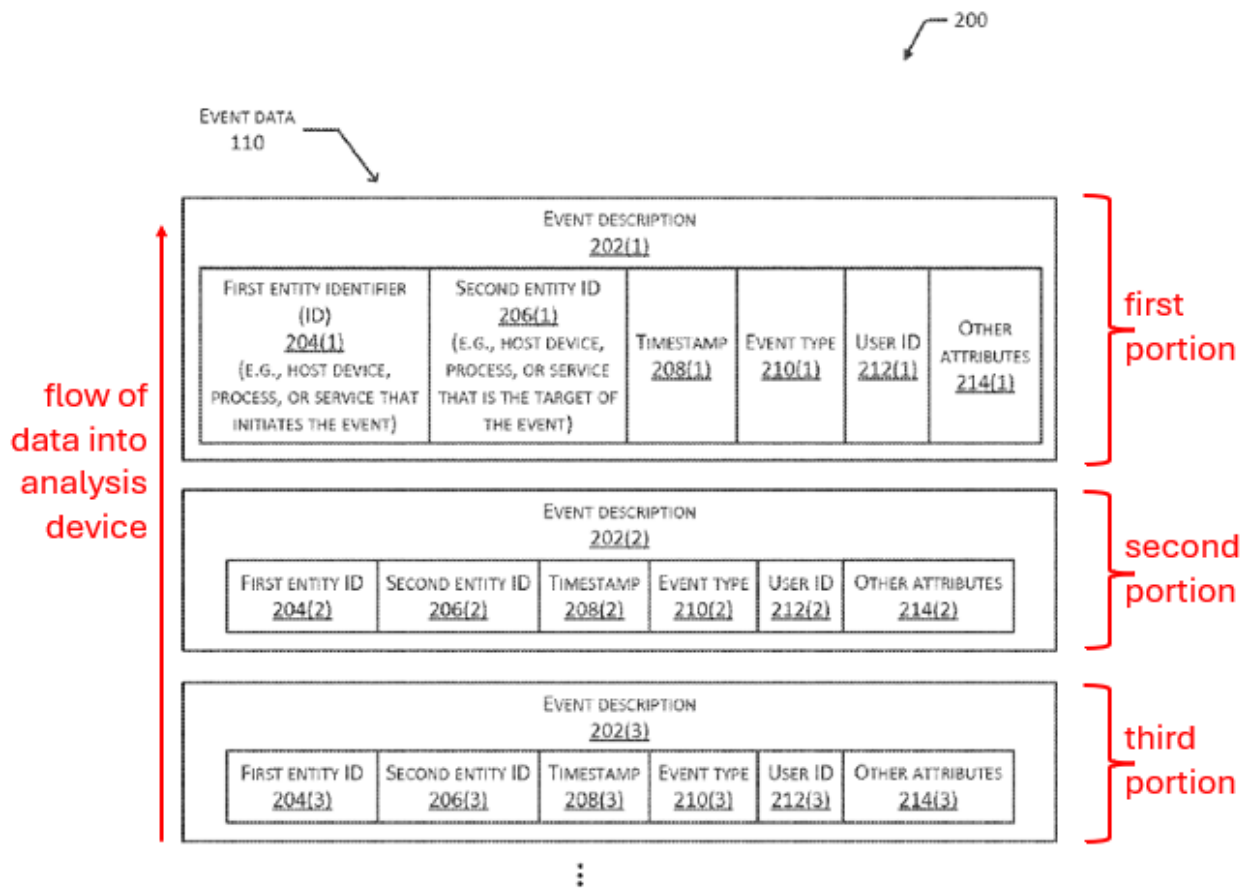
EX1004, FIG. 2. A POSITA would therefore understand that Brezinski-Crabtree would use received event descriptions in a data stream to identify relationships between a pair of entities.

“Based on a second portion of the streaming data, wherein the second portion is not identical to the first portion”

90. As explained for limitation [1.6], *above* Section VIII.D.6, Brezinski-Crabtree includes an analysis device that receives streaming event data, with one event description after another, as the event occurs. And as explained in this section

and the preceding section, Brezinski gives two exemplary types of “additional event data” that lead to different graph modifications: one type resulting in additional edges (EX1004, 17:23-26) and the other in additional nodes/vertices (*id.*, 17:26-29). Brezinski thus teaches receiving non-identical event descriptions within its flow of data. As such, in Brezinski-Crabtree the portion of streaming data that leads to new edges is different from the portion that leads to new nodes (a second portion of the streaming data, wherein the second portion is not identical to the first portion).

91. The POSITA would therefore recognize the obvious use case of a later event description for a new event between original graph node-entities (*second portion of streaming data*) arriving after an earlier event description for a new event with a new nodes (*first portion of streaming data*). For example, with reference to Fig. 2,, below, in the streaming environment of Brezinski-Crabtree, a POSITA would recognize the use case of a new entity/node being identified based on event description 202(1) received at a first time (i.e., a first portion of streaming data), and then a new relationship/edge being identified based on later received event description 202(2) (i.e., a different portion of received streaming data).



EX1004, Fig. 2 (annotated). Brezinski’s “additional event data” and “recently detected events” terminology reflects this temporal sequencing: the system may first identify a previously unseen entity from earlier-received event data, and later (based on subsequently received event data) identify a new relationship/event between entities that is not yet represented by an edge, prompting addition of an “additional edge[.]” *Id.*, 17:23-26.

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;

92. Brezinski teaches this limitation. Brezinski discloses incrementally updating its stored graph with newly identified nodes and newly identified edges when new event data is received. Brezinski explains that “on receiving additional event data ... the graph 118 may be modified” and that the modification includes adding new edges for newly detected relationships/events and adding new vertices for newly detected entities not previously represented in the graph. EX1004, 17:23-29; *see also id.*, Fig. 10 (Step 1014: “On receiving additional event data, incorporate additional edges and/or vertices into the graph). In particular, Brezinski states that the graph “may be modified to incorporate one or more additional edges 404 corresponding to recently detected events,” and also “may ... be modified to incorporate one or more additional vertices 402, in cases where the recently detected events involve entities not already described by vertices 402 in the graph 118.” *Id.*

93. As explained for limitation [1.7], *above* Section VIII.D.7, Brezinski teaches identifying a new first entity, and as explained for limitation [1.8], *above* Section VIII.D.8, Brezinski teaches identifying a new first relationship. Accordingly, upon receiving the event data reflecting the newly identified first entity and first relationship, Brezinski would modify its stored graph to reflect this information by adding corresponding first node for the new first entity and a

corresponding first edge for the first relationship. As explained for limitation [1.7], the newly identified first entity does not correspond to any of a pre-existing first plurality of nodes and its corresponding newly added first node would therefore also not be a part of the first plurality of nodes. Likewise, as explained for limitation [1.8], the newly identified first relationship is between a pair of entities of the first plurality of entities and does not correspond to any of a pre-existing first plurality of edges and its corresponding newly added first edge would therefore not belong to the first plurality of edges.

94. As an example, a POSITA would recognize an obvious use case of how the software instructions of the Brezniski-Crabtree combination would modify a graph to incorporate a node and a first edge with reference to a modified version of Brezniski's Figure 4, below.

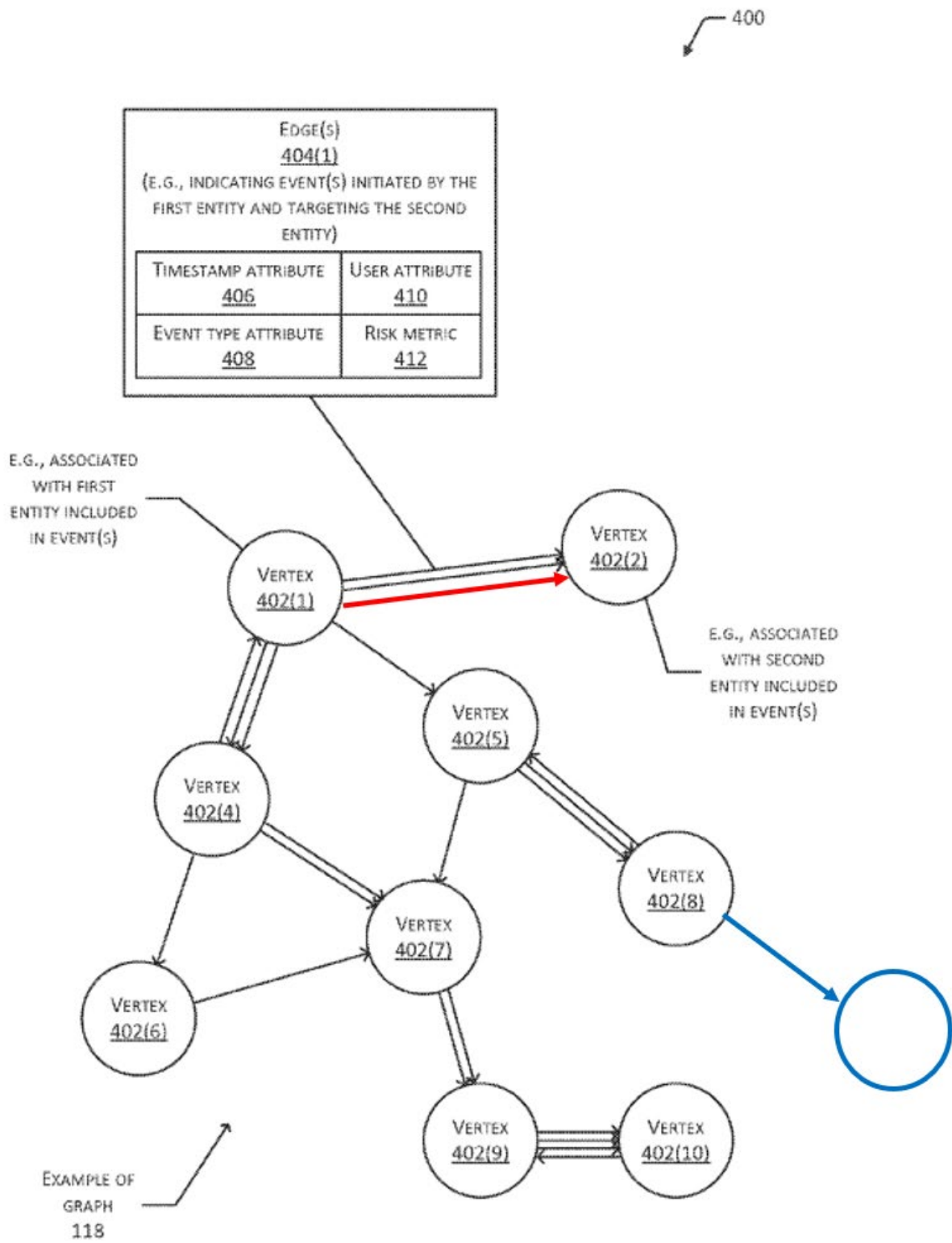


FIG. 4

EX1004, Fig. 4 (modified and annotated). The modified figure depicts the graph of Figure 4 being modified to comprise a new first node (in blue) not part of a first plurality of nodes and a new first edge (in red) not part of a first plurality of edges.

10. Element [1.10]

[1.10] for an anomalous event associated with a node in the modified representation of the first graph, perform a first correlation using the modified representation of the first graph to identify a first plurality of correlated nodes,

wherein each of the first plurality of correlated nodes corresponds to a respective event or resource, wherein each respective event or resource is associated with the anomalous event, and

wherein each of the first plurality of correlated nodes is connected by a respective edge of a second plurality of edges to the node associated with the anomalous event in the modified representation of the first graph;

95. Brezinski teaches this limitation. As explained more fully below, Brezinski teaches performing a first correlation using a modified representation of a first graph (i.e. the graph modified to include newly identified nodes and edges as described for limitation [1.9]) in two alternative ways. First, Brezinski discloses traversing a graph along edges from a starting node and identifying as anomalous activity data those edges having a rarity and risk metric over a certain metric (indicating that the event associated with such edge is anomalous) along with the nodes connected by such edges. Second, Brezinski teaches designating an entity (such as a honey pot) as being known as associated with anomalous behavior and

correlating such the node corresponding to that entity with any nodes connected to it.

“For an anomalous event associated with a node in the modified representation of the first graph, perform a first correlation using the modified representation of the first graph to identify a first plurality of correlated nodes,”

96. Brezinski teaches to “traverse a graph” along edges and identify “the edges 404 and the vertices 402” associated with a rarity and/or risk metric that satisfies a certain threshold (*perform a correlation ... to identify a first plurality of correlated nodes*), for inclusion “in a subset of the graph 118 designated as anomalous activity data 122.” *E.g.*, EX1004, 11:12-43, FIGS. 6-7. “The graph may be traversed according to the risk and rarity metrics, to identify patterns of anomalous activity in the event data.” *Id.* Abstract; *see also id.*, 3:52-53. The traversal may also be “breadth-first.” *Id.*, 3:60-63. This breadth-first graph traversal performs the claimed correlation because it uses the graph to correlate a plurality of nodes with anomalous network activity. *E.g.*, 3:52-67, 11:12-29, 12:12-14, 17:39-49, 18:22-44; 19:4-22. Brezinski teaches identifying these anomalous nodes by “analyzing a graph to identify anomalous activity that includes chains of related events” (*e.g.*, 17:39-43) and thus the correlation is *for an anomalous event associated with a node in the graph*, as recited in element 1.10(a). For example, the traversal process identifies nodes connected to edges that satisfy a “risk threshold”

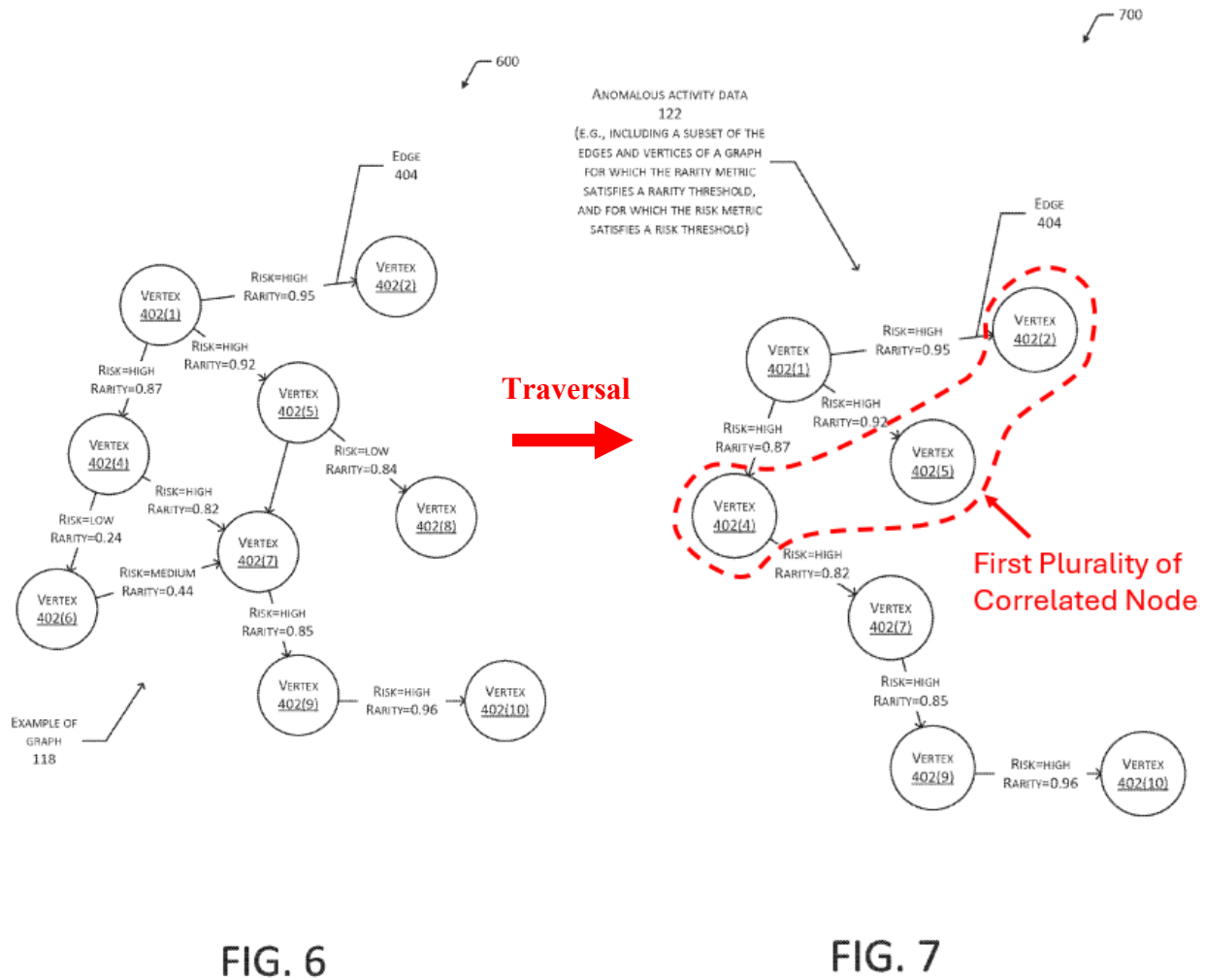
(e.g., *id.*, 18:23-25) and—because edges represent events—these edges indicate an anomalous event associated with a node in the graph.

97. Brezinski further teaches performing this correlation *using the modified representation of the first graph*. Brezinski teaches its analysis device performing graph traversal on “graph(s) 118 ... stored in memory” (EX1004, 11:12-18), which includes modified graphs because “the graph 118 may be modified to incorporate one or more additional edges 404 [or] vertices 402” (*id.*, 17:23-29; FIG. 10 (Step 1014)). See also *id.*, 8:23-27 (“graph generation module”), 16:15-21 (“process for generating a graph 118”). Confirming this, the first step 1102 of Brezinski’s graph traversal process is broadly “a graph 118 may be accessed.” *Id.*, 17:39-49; FIG. 10 (Step 1102 (“access a graph relating events detected on host devices”). Thus, Brezinski teaches identifying anomalous activity by traversing a modified graph such as that described for Element 1.9 (*supra* Section VIII.D.9). *Id.*, 17:39-49. A POSITA would recognize the obvious use case of performing such a traversal on a modified representation of a first graph described, e.g., to identify anomalous activity caused by newly identified entities and relationships/events.

98. Brezinski’s traversal procedure starts from a “start vertex” or “current vertex” within the graph and traverses “along at least a portion of the edges in the graph to identify those edges that include a rarity metric that satisfies a predetermined rarity threshold and that include a risk metric that satisfies a

predetermined risk threshold.” EX1004, 3:52-58. Brezinski explains that an “edge 404 is selected that originates from the current vertex 402,” and, where edges originating from the current vertex satisfy risk and rarity metric thresholds, that “the current edge 404, and the vertices 402 connected by the current edge 404, may be included in the subset of the graph 118 to be stored as anomalous activity data 122.” *Id.*, 18:9-31 Therefore, once an anomalous edge/event is identified from a starting vertex, the vertices connected by that edge are included in the anomalous subset along with the starting vertex, rendering that starting vertex an node associated with an anomalous event.

99. Figures 6 and 7 of Brezinski show an example of such the traversal procedure. Figure 6 depicts “an example graph 118, including example values of the risk metric 412 and the rarity metric 502 shown for the edges 404 in the graph 118.” *Id.*, Fig. 6, 11:44-47. Figure 7 depicts a subset graph based on the traversal of example graph of Figure 6 to identify “anomalous activity data 122” by determining edges “for which the rarity metric 502 satisfies and is at least a rarity threshold of 0.8, and for which the risk metric 412 satisfies a risk threshold of ‘high’” and vertexes connected by those edges. *Id.*, 12:1-4. Figures 6 and 7 are reproduced below and annotated to show a first plurality of correlated nodes identified by Brezinski’s graph traversal procedure.



Id., Figs. 6 & 7 (annotations in red). Brezinski specifically discloses that its traversal of the graph (such as that shown in Fig. 6) “may be breadth-first such that neighboring nodes are explored prior to exploring deeper along a possible branch.” *Id.*, 3:60-63. A POSITA understood “breadth-first” to be an iterative search methodology that first identifies all nodes directly neighboring a start vertex (i.e., first-level nodes) then iterates outward to identify the nodes directly neighboring the first-level nodes, and so forth. *See* EX1019, 594 (explaining that in a breadth first search, the “algorithm discovers all vertices at distance k from s before discovering

any vertices at distance $k + 1$,” where the distance is the number of edges), EX1023. In the example of Figs. 6 & 7, a POSITA would understand that performing a breadth first traversal of the graph in Fig. 6 from the vertex labeled 402(1) would identify the vertices labeled as 402(2), 402(4), and 402(5) (and the edges connecting them to vertex 402(1)) as anomalous activity data in a first level of the breadth first search. Brezinski’s graph traversal therefore includes a first “correlation” because it uses the structure of the graph—relationships captured by edges between vertices—to identify other graph elements related to the anomalous activity. And the vertices identified in the first level of such traversal as shown in annotated Fig. 7, above, constitute a first plurality of correlated nodes.

100. Alternatively, Brezinski teaches that anomalous activity can be detected and correlated by starting from a node/vertex known to be compromised or involved in suspicious activity—such as a honeypot or other known entity—and then including connected graph elements as anomalous activity data. Brezinski explains:

In some cases, an edge 404 may be included in the anomalous activity data 122 if it connects a vertex 402 that is associated with a particular host device 102 or other entity. For example, a host device 102, process 104, or service 106 may be configured as a honey pot in the environment 100. The honey pot may be accessible to other (e.g., malicious) entities in the environment 100, but authorized entities may have no reason to interact with the

honey pot. Accordingly, any interaction with the honey pot entity may be flagged as potentially malicious, and included in the anomalous activity data 122 for further investigation.

Id., 12:43-53.

101. Accordingly, Brezinski discloses performing a first correlation by identifying a known compromised entity and identifying as anomalous activity data all nodes (i.e., a first plurality of correlated nodes) connected to the node associated with the known compromised entity. As an example, referring back to the graphs depicted in Figs. 6 & 7 of Brezinski, a POSITA would recognize an obvious use case vertex 402(1) being associated with a known compromised entity and according to Brezinski's teachings including all edges connecting that node and the nodes connected by those edges as anomalous activity data. In the context of the exemplary graphs of Figures 6 & 7, these would correspond to vertices 402(2), 402(4), and 402(5) being identified as anomalous activity data, resulting in a correlation of those vertices as a "first plurality of correlated nodes."

102. This disclosure therefore teaches that, for anomalous activity reflected by an event/edge associated with a node/vertex, Brezinski performs a graph-based correlation that identifies multiple connected vertices—i.e., a "first plurality of correlated nodes"—using the graph representation.

“wherein each of the first plurality of correlated nodes corresponds to a respective event or resource,”

103. Brezinski teaches this limitation. As explained above for limitation [1.4], *above* Section VIII.D.4, Brezinski discloses its graph includes vertices/nodes corresponding to a plurality of resources. Accordingly, a POSITA would recognize the obvious use case of Brezinski’s traversal procedure identifying as anomalous a plurality of resource nodes. Such identified nodes would constitute a first plurality of correlated nodes corresponding to a respective resource. In addition, as explained for limitation [1.5], *above* Section VIII.D.5, Brezinski discloses receiving event data reflecting relationships between vertices/nodes and used to define edges connecting vertices/nodes corresponding to the event. Therefore, each node represented in Brezinski’s graph, including a first plurality of correlated nodes determined by Brezinski’s traversal procedure, corresponds to a respective event.

“wherein each respective event or resource is associated with the anomalous event”

104. Brezinski teaches this limitation in two ways. First, Brezinski teaches this limitation because its anomalous-activity correlation identifies and outputs a subset of the event graph comprising vertices and edges selected because they are associated with the anomalous activity (i.e., the anomalous event) as reflected by risk and rarity metrics and the graph’s connectivity. Brezinski explains that

anomalous activity is determined by traversing the event graph using anomaly-related criteria, including risk and rarity thresholds. Specifically, Brezinski states that “the graph 118 may be traversed along the edges 404 for which the rarity metric 502 satisfies [a criterion] and the risk metric 412 satisfies [a criterion]” and adds edges and vertices identified during such traversal as “anomalous activity data” EX1004, 11:54-57, 12:12-14, 1:41-46 (explaining that in a subset graph generated by a graph traversal, “the subset represents anomalous activity and a potential security risk”). A POSITA would understand that selecting edges (events) for traversal based on such thresholds identifies the events that are anomalous because Brezinski adds and identifies the neighboring vertices connected by those events as resources/entities associated with that anomalous activity. *See id.*, Abstract (“The graph may be traversed according to the risk and rarity metrics, to identify patterns of anomalous activity in the event data.”). Because the traversal includes only edges meeting the risk/rarity criteria and then includes those edges and their connected vertices in anomalous activity data, the resulting vertices (and their respective resources) and edges (and respective events) are associated with the determined anomalous activity (i.e., anomalous event).

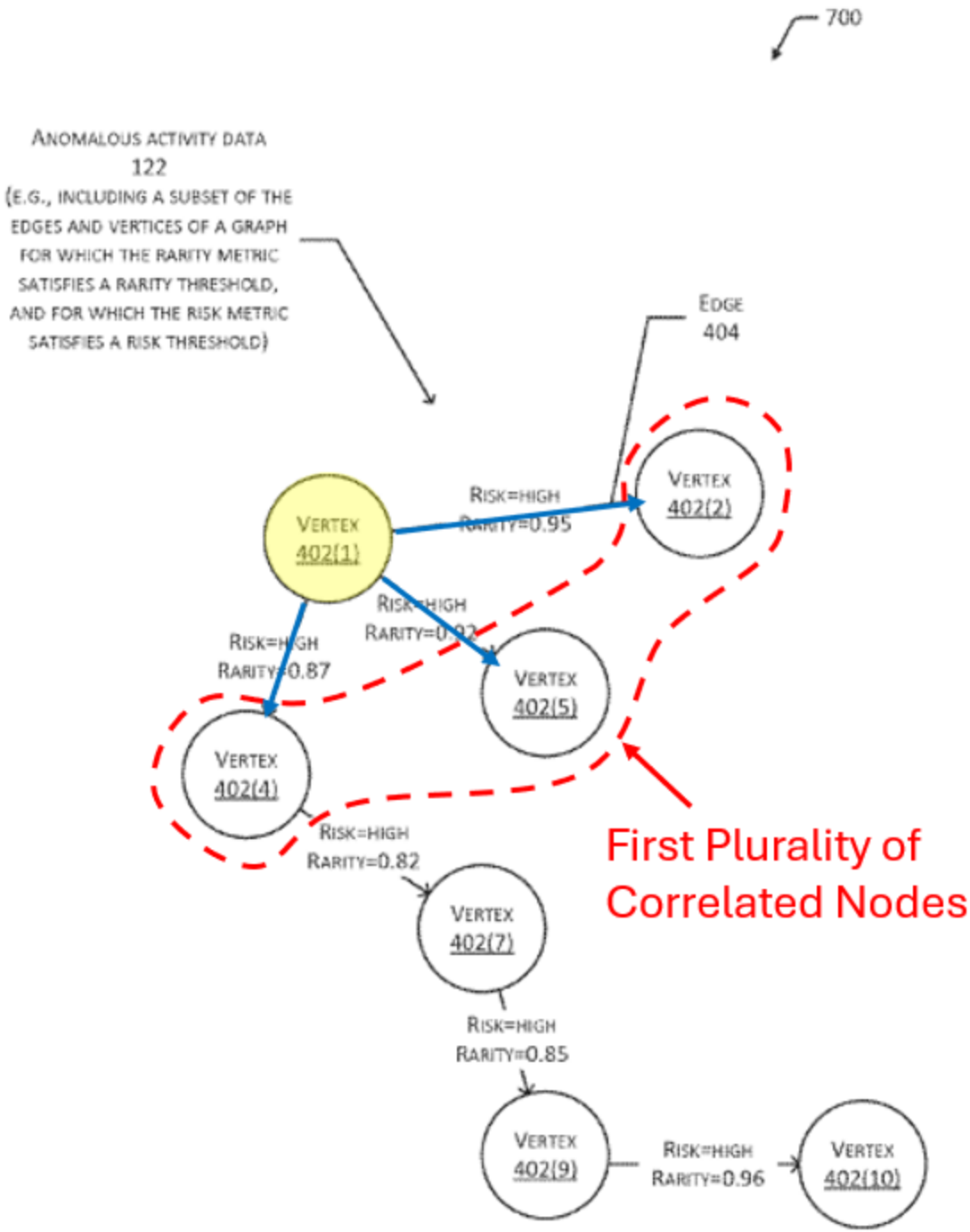
105. Second, as explained above, Brezinski discloses an alternative first correlation of identifying all nodes interacting with a known compromised entity as anomalous activity data. Therefore, such nodes (i.e., respective resources) and their

connected edges (i.e., respective events) are identified as being associated with an anomalous event (i.e., interactions by a compromised entity). This disclosure of Brezinski is consistent with the '628 patent's description of performing a first correlation. *See* EX1001, 21:11-20 (“[W]hen an anomaly is identified ..., it may be correlated with other events, network resources, behavior patterns, or any other available information that may be contextually linked to the anomaly. Using a CPG for the network, correlations may be identified by first comparing resources involved in the anomaly [], for example any machines that participated in the anomalous event or any databases or other resources that were accessed.”).

and wherein each of the first plurality of correlated nodes is connected by a respective edge of a second plurality of edges to the node associated with the anomalous event in the modified representation of the first graph;

106. For each of the two alternative first correlations taught by Brezinski, the first plurality of correlated nodes is connected by a respective edge of a plurality of edges (i.e., a second plurality of edges) to the node associated with the anomalous event. When performing a breadth-first traversal of the modified graph as taught by Brezinski such a traversal identifies all edges immediately connecting the starting vertex having a risk and rarity rating above a threshold, and including the starting vertex, those identified edges, and vertices connected by those edges to the starting vertex to the anomalous activity data. When identifying anomalous activity data

based on a known compromised entity, Brezinski teaches identifying the vertices having directly interacted with the vertex corresponding to the compromised entity as anomalous activity data. In either alternative, all of the first plurality of correlated nodes are connected by respective edges directly to the originating node (i.e., the node associated with the anomalous event). For example, in a graph such as depicted in Brezinski's Figure 7, a POSITA would recognize the use case of vertex 402(1) being the originating vertex (either by being designated as the starting vertex for a breadth-first traversal or as having been designated as a known compromised node). Either alternative would result in first plurality of correlated nodes (in red) connected by a second plurality of edges (in blue) to a node associated with an anomalous event (in yellow) as shown in annotated Figure 7, below.

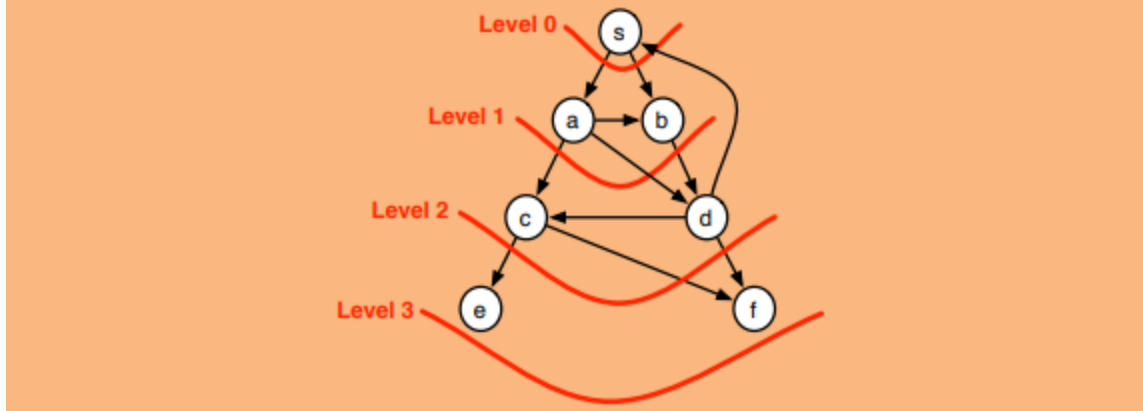


11. Element [1.11]

[1.11] for one or more of the first plurality of correlated nodes, perform a further correlation using the modified representation of the first graph to identify a second plurality of correlated nodes, wherein each of the second plurality of correlated nodes is connected through a respective edge of a third plurality of edges to the respective node of the first plurality of correlated nodes in the modified representation of the first graph;

107. Brezinski teaches this limitation. As explained above for limitation [1.10], *above* Section VIII.D.10, Brezinski teaches performing a breadth first traversal of a modified graph to identify a first plurality of correlated nodes. Further, A POSITA would understand that a breadth-first search (BFS) performs correlation in levels: (i) first identifying a set of nodes directly connected to a starting/anomalous node (a first “plurality of correlated nodes”), and then (ii) continuing the traversal from those first-level nodes to identify additional nodes connected to the first-level nodes by additional edges (a second “plurality of correlated nodes”). BFS is (along with its cousin depth-first search (DFS)) taught to undergraduates in a Computer Science curriculum in most universities: it begins from a certain node (the “root node”) and first examines all nodes reachable via an edge from the root node (these are the “layer 1 nodes”), then examines each node connected to a layer 1 node, reaching all layer 2 nodes; this continues until all nodes in the graph have been examined. An example of a breadth-first search (BFS) iteratively identifying nodes in a graph level-by-level is shown in the exemplary figure below:

Example 14.1. A graph and its levels illustrated. BFS visits the vertices in level 0, 1, 2, and 3 in that order.



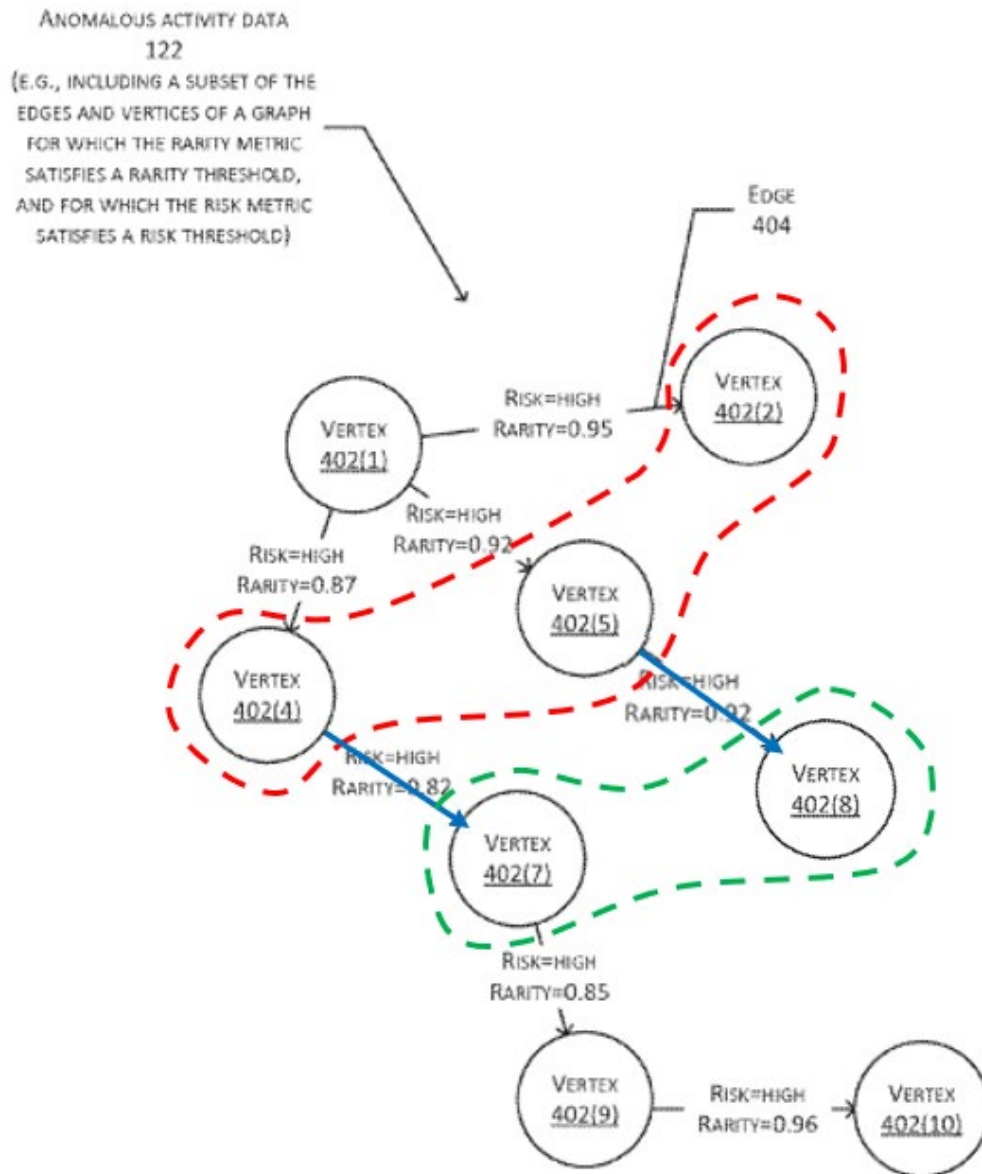
EX1023. A breadth first traversal procedure such as Brezinski's therefore is iterative and proceeds by exploring nodes level-by-level. Therefore, Brezinski's breadth first search would perform a further traversal/correlation from each vertex identified in the first plurality of correlated nodes to identify edges (and associated nodes) connected to the first plurality of correlated nodes that satisfy risk and rarity thresholds and add such edges and nodes to as anomalous activity data.

108. Alternatively, as explained above for limitation [1.10], Brezinski teaches performing a first correlation to determine a first plurality of correlated nodes based on identifying nodes that interacted with a known-compromised node/honey pot. It would have been obvious to a POSITA to subsequently implement Brezinski's breadth first traversal procedure beginning with the vertices that are directly connected (via interaction edges) to the honeypot/known-compromised node, because those directly connected vertices represent the

immediately implicated accounts/resources in the suspicious interaction. From there, breadth-first search expands correlation outward (and/or inward) from those directly connected vertices to identify additional nodes connected through additional edges—precisely to determine the potential spread of anomalous activity through the network from the known-compromised node/honey pot.

109. Thus, whether viewed as (i) BFS second-level expansion following an first-level traversal, or (ii) BFS anchored to a known compromised/honeypot node by first selecting the nodes that interacted with that node and then expanding outward, Brezinski teaches performing a further correlation from one or more first-level correlated nodes to identify second-level correlated nodes connected through respective edges, satisfying limitation [1.11]. An example is provided in the below figure modifying Brezinski's Figure 7. As shown in the below figure, such a breadth-first search identifies a second plurality of correlated nodes (in green) each connected by one of a third plurality of edges (in blue) to a respective node of the first plurality of correlated nodes (in red). As illustrated in the modified version of Figure 7, Brezinski's anomaly-detection process identifies a subset of vertices and edges that satisfy risk and rarity thresholds, thereby forming a subgraph of interest. A POSITA would understand that the particular vertices included in such a subgraph depends on the specific threshold criteria and the starting node or nodes used for traversal. In some instances, vertices directly connected to an anomalous or high-

risk node (shown in red) would form the first plurality of correlated nodes, and vertices connected to those nodes (shown in green) would form the second plurality of correlated nodes.



EX1004, FIG. 7 (modified and annotated to show alternative use cases). The figure modifies Figure 7 to show an additional node connected by an edge to Vertex 402(5)

While Figure 6 shows such a node (Vertex 402(8)), it is not included in the anomalous activity data reflected in Brezinski's Figure 7 due to the "low" risk metric of its connecting edge. Figure 7 is exemplary, however, and a POSITA would have understood that in other obvious use cases such an additional node would satisfy the risk/rarity threshold and therefore be included in the graph of anomalous activity, as shown in this modified figure. Brezinski discloses that its "risk metric" reflects such attributes as access privileges, sensitive information, password protections, and encryption. EX1004, 2:65-3:16. In light of that disclosure, and because Figure 6 only shows a graph of exemplary events in a network, a POSITA would have found it obvious that in a different network scenario, Vertices 402(5) and 402(8) and their interactions would have different attributes (such as access, information sensitivity, password protections, encryption) such that the risk metric for their interaction would be "high." This result in Vertex 402(8) being included in the subset graph. While the above figure shows one example of a traverse identifying a plurality of second-level nodes (i.e., a second plurality of correlated nodes) a POSITA would recognize additional obvious use cases. A POSITA understood that networks included many (potentially thousands or more) assets and interactions between them. EX1009, 673; EX1018, [0004]. The POSITA would therefore have found it obvious for a graph of a network, similar to the one depicted in Figure 6 of Brezinski, to depict additional edges between nodes, including edges between first level Vertices

(402(2), 402(4), 402(5)) and other nodes not shown in Figure 6, having associated risk and rarity ratings such that they would be included as second-level nodes in the subset graph.

12. Element [1.12]

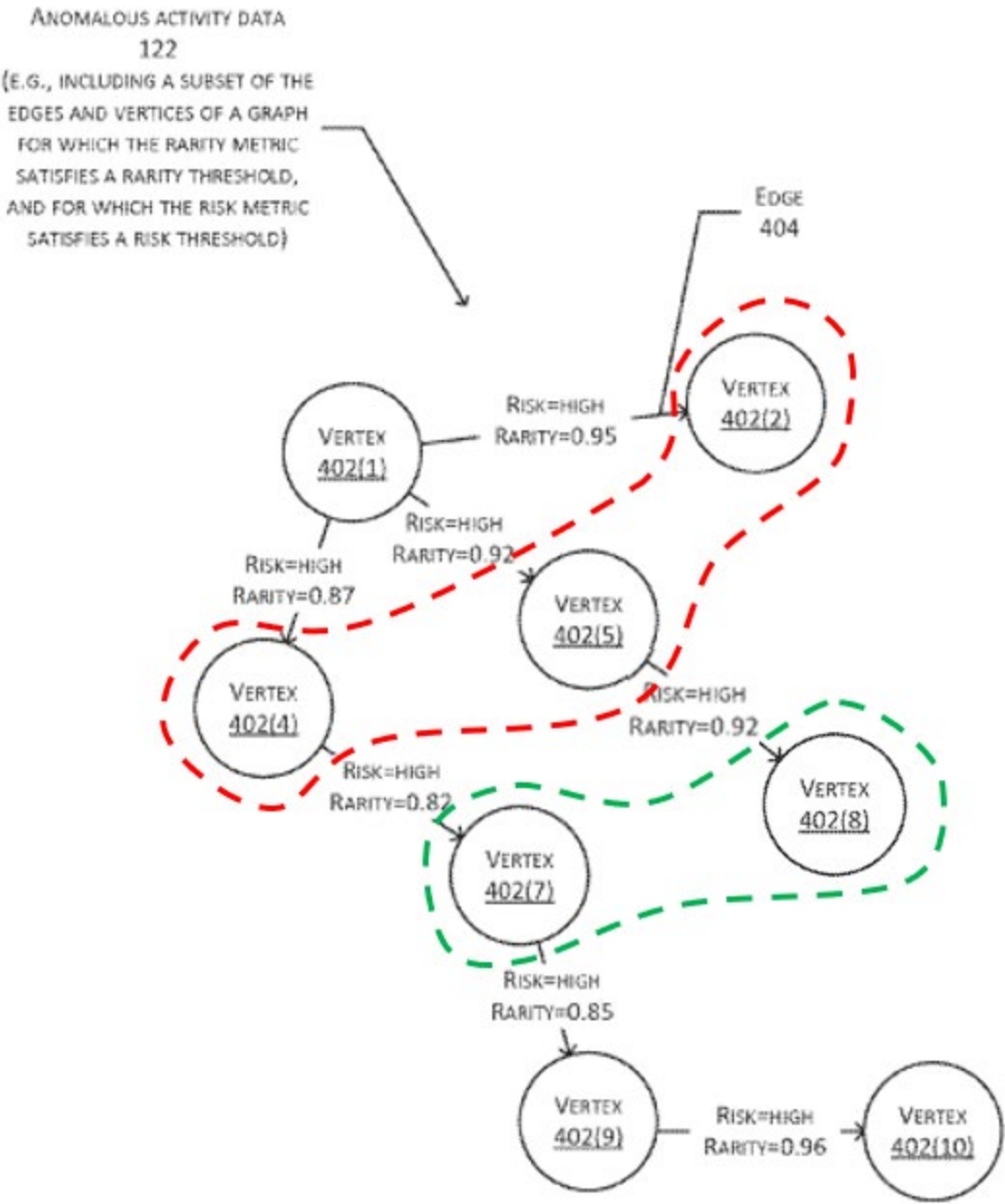
[1.12] generate a representation of a second graph comprising representations of one or more of the first plurality of correlated nodes, representations of one or more of the second plurality of correlated nodes, representations of one or more of the second plurality of edges, and representations of one or more of the third plurality of edges, wherein one or more of the second plurality of edges together with one or more of the third plurality of edges represent one or more event flows that could be involved in a cybersecurity attack; and

110. Brezinski teaches this limitation. Brezinski teaches generating and outputting a second graph representation in the form of a subset of the event graph describing anomalous activity. Brezinski explains that the system traverses the stored event graph to determine a subset indicative of anomalous activity, and that “[t]he subset of the graph 118 determined by traversing the graph 118 may be described in anomalous activity data 122,” which may be “presented ... as the subset of the graph 118.” EX1004, 12:12-24. Thus, Brezinski expressly discloses generating a graph-based representation (a “subset of the graph”) separate from the full underlying graph—i.e., a “second graph” representation comprising the correlated vertices and edges. As explained above for limitations [1.11] and [1.12], Sections VIII.D.11-12, such anomalous activity data and resulting subset graph

includes the first and second pluralities of correlated nodes and the second and third pluralities of edges.

111. Brezinski teaches that this output subset graph captures event-flow structures in the form of linked vertex-edge-vertex tuples arranged as a chain/tree. Brezinski states that anomalous activity data “may include a description of one or more events as tuples of vertex-edge-vertex,” including edge attributes such as timestamp and event type. EX1004, 12:27-29. Brezinski further describes the anomalous activity output as an “activity chain” and a “tree of linked [V-e-V] tuples,” reflecting multi-step sequences of events across multiple vertices connected by multiple edges. *Id.*, 19:17-22. A POSITA would understand that such linked tuples (and the presented subset graph including such linked tuples) depict event flows through the environment—i.e., sequences of events connecting entities/resources—of the type investigated in cybersecurity attacks (e.g., movement from one compromised entity to another through successive interactions). *see also* EX1004, 4:5-8 (“[A]n attacker may seek to leverage the initial point of compromise to gain access to other host devices, processes, services, or stored data that enables further access along a chain of multiple accesses.”), 4:34-38 (“By correlating event data in a graph, and traversing the graph according to risk and rarity of events, implementations may enable the detection of an attack that involves a complex chain of accesses in a computing environment.”).

112. The below figure depicts such a second graph including event flows. The figure modifies Brezinski Figure 7 (as previously described and presented for element [1.11], Section VIII.D.11 *above*). As with Figure 7 of Brezinski, the modified graph below depicts a subset of a graph following traversal and identification of edges and associated nodes exceeding a risk and rarity metric threshold. EX1004, 1:41-46;



EX1004, FIG. 7 (modified and annotated to show alternative use cases). As shown, the figure depicts two separate anomalous event flows (annotated in red), each of which include one of the third plurality of edges (shown in orange) together with one of the second plurality of edges (shown in blue) (i.e., a first event flow from

Vertex 402(1) to 402(4) to 402(7) and a second event flow from 402(1) to 402(5) to 402(8)).

13. Element [1.13]

[1.13] generate a report comprising information associated with the one or more event flows.

113. Brezinski teaches this limitation in that it discloses graphically presenting its anomalous activity data, including the subset graph and the event flows depicted in such graph. EX1004, 12:16-24 ([T]he anomalous activity data 122 may be accessed and examined by system administrators, security analysts, system developers, or other personnel to determine whether the events included in the anomalous activity data 122 indicate an attack, exploit, or other form of security breach. In such cases, the anomalous activity data 122 may be presented in a graphical form (e.g., as the subset of the graph 118) to facilitate analysis.”). By providing a graphical presentation of the subset graph including event flows within the graph for user analysis, Brezinski generates a report comprising information associated with the event flows (such information includes edges comprising event flows and nodes involved in event flows). *see also* EX1014, 450 (defining a “report” as “[t]he presentation of information about a given topic”).

114. Crabtree reinforces this understanding by teaching that cybersecurity analysis results may be presented to users in the form of reports, dashboards, alerts,

and visualizations identifying relevant entities and resources. For example, Crabtree discloses the ability to “format and deliver customized reports and dashboards,” as well as to issue “specialized alerts, updates and reports.” EX1005, [0063]-[0064]. Crabtree further discloses producing “a graph visualization of users, servers, devices, and other resources” and a “recommendation report,” and sending an alert to an administrator with contextual information for a connection when the connection is a risk. *Id.*, [0076], [0080]. Accordingly, Brezinski teaches graphically presenting (i.e., reporting) its subset graph including anomalous event flows, and Crabtree reinforces that it was well known to present such graph analysis results in a report.

E. Claim 4

[4] 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 identify the anomalous event.

115. Brezinski teaches identifying an anomalous event as part of its graph-based anomaly detection. Brezinski explains that anomalous activity (i.e., an anomalous event) is identified by traversing the event graph and selecting edges/events meeting anomaly criteria, stating that “the graph 118 may be traversed along the edges 404 for which the rarity metric 502 satisfies [a criterion] and the risk metric 412 satisfies [a criterion].” EX1004, 11:54-57; *see also id.*, Abstract (“The

graph may be traversed according to the risk and rarity metrics, **to identify patterns of anomalous activity in the event data.**” (emphasis added)) As explained for element [1.10], *above* Section VIII.D.10, Brezinski also teaches identifying anomalous activity by identifying edges and nodes interacting with a known compromised entity. Brezinski further explains that the resulting “subset of the graph 118 determined by traversing the graph 118” is “described in anomalous activity data 122” and may be presented “as the subset of the graph 118,” i.e., the identified anomalous event. EX1004, 12:12-24, 12:35-38. Brezinski’s graph traversal procedure is performed by Brezinski’s graph analysis module, which, as explained for element [1.2] comprises computer instructions stored in nontransitory memory of Brezinski’s analysis device.

F. Claim 5

[5] The computer system of claim 4, wherein the anomalous event is identified by comparing one or more of the events relating to one or more entities of the first plurality of entities to a pattern of normal event behavior.

116. Brezinski meets this limitation because its **rarity metric–based graph traversal** identifies anomalous activity by comparing observed events against what is **normal/typical** for the environment and treating sufficiently *rare* (i.e., abnormal) events as anomalous.

117. As explained for element [1.10] and claim 4, Brezinski teaches that anomalous activity is identified by traversing the event graph along edges (events)

selected using a rarity metric. As Brezinski explains, its rarity metric indicates “a rarity or a frequency of events of that type that have occurred between the entities.” EX1004, 3:43-46. A POSITA would have understood that concept of a “rarity” metric requires a baseline of what is normal—i.e., the frequency/typicality of event behavior across the environment—so that an edge/event can be assessed as rare relative to that baseline. In other words, a POSITA would understand Brezinski’s rarity metric as representing a comparison of a current event against a “pattern of normal event behavior” (events that commonly occur, in typical frequencies and combinations) to determine whether the current event is. This comparison is expressly reflected in Brezinski’s Formula 1, reproduced below, used to calculate a rarity metric.

$$\text{Rarity} = \frac{1}{1 + \log_2 |V1, \{E | \text{exclude timestamp}\}, V2|} \quad \text{Formula 1}$$

Id., 10:37-39.

118. As Brezinski explains, this formula is based on the count of the number of edges between two vertices, and for a given event type determines how rare or common such event type is within total count of edges (i.e., events) connecting the two vertices. EX1004, 10:41-58. Brezinski thereby determines how rare an event between a pair of nodes is based on the normal pattern of behavior between that node

pair. And Brezinski subsequently uses that rarity metric to identify an anomalous event, i.e., by traversing a graph and including edges and connected nodes as anomalous activity data based on their exceeding a rarity threshold.

119. Alternatively, this element is taught by Crabtree. Crabtree explicitly discloses “collect[ing] activity information” on a network and processing such information to “analyze behavior patterns.” EX1005, [0072]. Crabtree further explains that such “behavior analytics information... may be received at a graphing service 145 for inclusion in a [cyber-physical graph].” *Id.*, [0076]. Moreover, Crabtree’s determined behavior patterns are used to recognize anomalous behavior. *Id.*, [0072] (explaining that “anomalous behavior may be recognized... based on a threshold of variance from an established pattern or trend”). Crabtree therefore compares events to relating to entities in a network graph to a pattern of normal behavior by comparing observed events to established behavioral patterns to determine whether the observed behavior exceeds a threshold of variance. The ’628 patent describes identifying anomalous behavior based on a pattern of normal behavior in the same manner. *See* EX1001, 20:66-21:5 (“When an event occurs that is recognized to be outside the normal patterns, for example if the event exceeds a configured threshold for aberrance... it may then be identified as an anomaly.”).

120. A POSITA would have been motivated to combine the teachings of Brezinski with Crabtree’s teaching that “anomalous behavior may be recognized ...

based on a threshold of variance from an established pattern or trend” (EX1005, [0072]) because that teaching would have predictably improved Brezinski’s graph-based anomaly detection by providing an additional, complementary way to distinguish suspicious activity from ordinary network behavior. Brezinski already seeks to identify anomalous activity by traversing a graph of entities and events using rarity and risk criteria (*see* EX1004, 11:12-29); Crabtree’s variance-from-normal-pattern approach addresses the same objective—detecting cybersecurity-relevant anomalies—but does so by evaluating deviations from established behavioral baselines over time (EX1005, [0072]). A POSITA would have recognized that incorporating Crabtree’s behavior-pattern threshold based detection into Brezinski would make Brezinski’s anomaly determinations more robust, allowing the system not only to identify anomalies based on meeting risk and rarity criteria, but also to recognize when activity departs from normal trends for users, devices, or other entities represented in the graph. And Crabtree provides an explicit motivation for using its behavior-based anomaly detection, explaining that its method “for behavioral analytics enables proactive and high-speed reactive defense capabilities against a variety of cyberattack threats, including anomalous human behaviors as well as nonhuman ‘bad actors’ . . . , enabling rapid response to threats to mitigate any potential impact” and a POSITA would implement such method into the system of Brezinski for the same reasons. EX1005, [0073]. Because both

references operate in the same field, use graph-based security information, and aim to detect anomalous or attack-related behavior, a POSITA would have had a reasonable expectation of success in implementing Crabtree's behavioral analysis into Brezinski's existing anomaly-analysis framework.

G. Claim 6

[6] 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 identify a portion of the modified representation of the first graph that matches a known attack pattern, wherein the portion of the modified representation of the first graph comprises the node associated with the anomalous event.

121. Brezinski teaches this limitation. As explained for element [1.10], Section VIII.D.10 *above*, Brezinski discloses identifying anomalous nodes based on interactions with a known compromised node or honey pot. And Brezinski discloses “identify[ing] a portion of the modified representation of the first graph” matching a known attack pattern by identifying as anomalous activity any edge connecting a node with such a known compromised/honey pot node. EX1004, 12:50-53. (“[A]ny interaction with the honey pot entity may be flagged as potentially malicious, and included in the anomalous activity data 122.”). By automatically classifying any interaction with a known compromised node/honey pot as anomalous activity, Brezinski treats such interaction as matching a known attack pattern because Brezinski recognized that other entities would have no other reason to interact with

the known malicious entity/honey pot unless such interaction occurred as part of a potential cybersecurity attack; that is Brezinski recognized that interactions with a known compromised node or honey pot constituted known attack patterns. *See id.*, 12:47-50 (“The honey pot may be accessible to other (e.g., malicious) entities in the environment 100, but authorized entities may have no reason to interact with the honey pot.”). The identification of nodes and edges constituting anomalous activity data is performed by Brezinski’s graph analysis module (*see id.*, Fig. 1), which, as explained for element [1.2] comprises computer instructions stored in nontransitory memory of Brezinski’s analysis device for analyzing Brezinski’s graph structure.

122. The above teaching of Brezinski is consistent with the ’628 patent’s disclosures for this limitation because ’628 patent’s specification describes identifying anomalous events based on a known attack pattern in the very same way. See EX1001, 20:61-21:6² (“When an event occurs that is recognized to be outside the normal patterns, for example if the event ... involves a known-compromised

² In prosecution, the Applicant identified this disclosure in the ’628 patent’s as-filed application as providing support for claim 6. *See* EX1002, 215/279 (identifying ¶ [83] of the as filed specification as providing specification support for claim 5), 86/279.

node or external address or resource (such as a deliberate ‘honeypot’ node for intrusion detection), it may then be identified as an anomaly.”).

H. Claim 9

[9] 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, an activity, an event, a group, or a service.

123. Brezinski teaches this limitation because the claim disjunctively claims (using the term “or”) eight different entity types and Brezinski’s teaches a first plurality of entities containing at least one of those types. As explained for limitation [1.4], *above* Section VIII.D.4, the first plurality of entities comprises a plurality of resources. Brezinsky further describes “each vertex corresponding to an entity (e.g., a host **device, process or service**).” EX1004, 2:45-58 (emphasis added).

I. Claim 10

[10] The computer system of claim 1, wherein at least some of the edges of the second plurality of edges or of the third plurality of edges in the representation of the second graph are assigned a numerical weight that indicates a likelihood of a successful cybersecurity attack gaining access from one node to another.

124. Brezinski meets this limitation because its **risk metric–based graph traversal** identifies anomalous activity based on the likelihood of gaining access from one node to another. Brezinski recognizes that a cyberattack often involves a chain of access through multiple entities in a network. Brezinski explains that in a “malicious intrusion or attack..., an attacker may seek to leverage the initial point

of compromise to gain access to other host devices, processes, services, or stored data that enables further access along a chain of multiple accesses.... Through this exploration, the attacker's access may spread until it is able to gain access to a potentially valuable target.” EX1004, 4:1-22. Accordingly, Brezinski discloses traversing its network graph using risk metric assigned to respective edges, and that such risk metrics “may be based on a potential impact or access capability of the associated event. For example, Brezinski discloses assigning a higher risk score for an edge associated with a user having “wide access privileges” (*id.*, 9:19) or if the associated event for the edge “accesses a host device 102, a process 104, or a service 106 that enables access to sensitive, confidential, or private information, or if the associated event accesses information that enables such access.” *Id.*, 9:26-30. Brezinski therefore teaches assigning a higher risk score to edges associated with greater access to other entities (e.g., resources) in the network, with such greater access indicating a higher likelihood of a successful cybersecurity attack gaining access to such entities (i.e., from one node to another). Similarly, Brezinski teaches that its risk metric for an edge may be based on whether a connection is secured (e.g., encrypted) or unsecured (e.g., unencrypted). *Id.*, 3:5-12. A POSITA would recognize that whether a connection is secured or not would determine the likelihood of success of a cyberattack gaining access to a node through such connection, and that this would be reflected in the risk metric associated with the connection.

Brezinski also discloses that its risk score may be reflected as a numerical weight. *Id.*, 9:42-44 (“The risk metric 412 may range from 0 to 10, with a higher number corresponding to a higher security risk for an entity or an attribute of an event.”). Brezinski therefore teaches assigning such numerical risk scores indicating a likelihood of a successful cybersecurity attack gaining access to a node to edges within its graph, including to the respective second and third pluralities of edges described for elements [1.10] and [1.11] (*above* Sections VIII.D.10-11).

J. Claim 11

1. Elements [11.1]-[11.5], [11.8], [11.11] And [11.15]-[11.17]

125. Elements [11.1]-[11.5], [11.8], [11.11], [11.15]-[11.17], are substantively identical to elements [1.1]-[1.5], [1.6], [1.8] and [1.11]-[1.13], respectively.³ *See* EX1010 (claims listing showing claims side-by-side). Brezinski-Crabtree satisfies those elements for the same reasons explained above for elements [1.1]-[1.5], [1.6], [1.8] and [1.11]-[1.13]. *Above* Sections VIII.D.1-6, 8, 11-13.

³ Elements [1.12] and [11.15] differ only in that the former uses the term “represent” while the latter uses “represents” but this does not create a substantive difference in the elements. The remaining elements are verbatim identical. EX1010.

2. Elements [11.6]

[11.6] wherein a representation of a first node of the first plurality of nodes includes information about a first security vulnerability associated with a first entity, wherein the first node corresponds to the first entity, and

126. Brezinski teaches this limitation. Specifically, Brezinski discloses that edges in its graphical representation are represented with a “risk metric indicating a level of security risk associated with the event.” EX1004, 2:65-67; *see also id.*, Fig. 6 (depicting edges assigned with risk metric ratings). Brezinski also discloses that “the risk metric [] for an edge [] may be determined by accessing security risk data which lists a risk metric [] for one or more particular host devices [], processes [], [or] services” (i.e., entities). *Id.*, 9:37-41. A POSITA would understand that such a risk metric associated with an entity constitutes a “security vulnerability” because Brezinski describes such metric reflecting whether such entity “enables access to sensitive, confidential, or private information,” that is, it describes the extent to which the entity makes the system vulnerable to a cybersecurity attack. *Id.*, 9:26-30; *see also id.*, Fig. 10 (Step 1008: “Access security risk data describing one or more security risk levels associated with the entities”), 16:51-54. And Brezinski expressly teaches placing risk/security information directly on the vertex (node) representation of an entity. EX1004, 17:11-14 (disclosing assigning, in a graph “a risk metric . . . to one or both of the vertices [] connected by [an] edge [], in addition to or instead of the risk metric [] included in the edge [].”). Accordingly, a POSITA implementing

Brezinski would have understood it to be desirable to store, as part of the node/vertex representation for an entity (e.g., a host), entity-specific security-risk information derived from “security risk data” (such as creating network exposure vulnerabilities) to provide a more complete and accurate representation of the security risks present in the network. Accordingly, a POSITA would include such a representation of a security vulnerability (i.e., a first security vulnerability) for a first node corresponding to a first entity.

127. Alternatively, Crabtree teaches including information about security vulnerabilities about entities in a network security graph. Crabtree discloses a “method [] for continuous network resilience scoring” to “measure an overall level of risk for a network infrastructure.” EX1005, [0077]. This method involves “first collecting [] information on publicly-disclosed vulnerabilities, such as (for example) using the Internet or common vulnerabilities and exploits (CVE) process.” *Id.* “This information may then [] be incorporated into a CPG..., and the combined data of the CPG and the known vulnerabilities may then be analyzed [] to identify the relationships between known vulnerabilities and risks exposed by components of the infrastructure.” *Id.* Crabtree therefore teaches including information about known security vulnerabilities for the entities in its network graph.

128. A POSITA would have been motivated to combine Brezinski with Crabtree’s teaching of including security-vulnerability information for entities in the

graph because that addition would have predictably improved Brezinski's graph-based anomaly analysis by making the graph more informative about which nodes and event paths pose the greatest real-world attack risk. Crabtree expressly describes the benefit of its approach, stating that "the combined data of the CPG and the known vulnerabilities may then be analyzed 1203 to identify the relationships between known vulnerabilities and risks exposed by components of the infrastructure." EX1005, [0077]. A POSITA would have therefore recognized that incorporating Crabtree's vulnerability information into Brezinski's graph would make Brezinski's graph more useful for analysis purposes, because suspicious events involving vulnerable hosts, services, or other resources could be prioritized and interpreted in light of the entities' known exposure to attack. A POSITA would have had a reasonable expectation of success in doing so because Brezinski already teaches a graph of entities and events, and it further teaches using stored security risk data for entities such as host devices, processes, and services to determine graph risk metrics. EX1004, 2:65-67, 17:11-14. Crabtree complements that approach by teaching that a cyber-physical graph can incorporate network vulnerability information and known and possible exploits, and by expressly teaching to continuously collect information on publicly-known vulnerabilities and incorporate vulnerability info into the CPG. EX1005, [0077]. A POSITA would therefore find it obvious to include in

Brezinski's graph such security vulnerability information (i.e., information about a first security vulnerability) for the first entity corresponding to a first node.

3. Elements [11.7]

[11.7] wherein the first security vulnerability is identified based on a database of known security vulnerabilities;

129. Brezinski teaches identifying an entity's risk metric (i.e., security vulnerability) from a database maintained by the analysis device. Brezinski explains that the analysis device's memory includes "data storage 922" and that "[t]he data storage 922 may comprise a database ... and may be a relational or a non-relational datastore." EX1004, 15:58-62. Brezinski further explains that this data storage "may store security risk data 924 that may provide a risk metric [] associated with" entities such as "host devices [], processes [], [and] services," and that the security risk data is accessed and used by the risk analysis module. EX1004, 16:1-5. Accordingly, Brezinski teaches a database of known security vulnerabilities for entities within the network used to provide risk metrics for those entities, including for the first security vulnerability described for limitation [11.6] (Section VIII.J.2 *above*).

130. Alternatively, as explained for element [11.6], *above* Section VIII.J.2, Crabtree teaches known security vulnerabilities including a first security vulnerability. Crabtree expressly discloses "collecting [] information on publicly-disclosed vulnerabilities." EX1005, [0077]. A POSITA implementing Crabtree's

teaching of including security-vulnerability information for entities in the graph-based system of Brezinski would have naturally, after collecting such information, stored it in a database, such as Brezinski's disclosed data storage database for its analysis device, in order to make such information available to the analysis device in building and analyzing its network graph structure. *See* EX1004, 15:58-60, Such vulnerability information for an entity would therefore be identified based on its presence in the database.

4. **Elements [11.9]**

131. Element [11.9] differs from element [1.7] only in that the former recites “identify a second entity that does not correspond to any of the first plurality of nodes” while the latter recites “identify a first entity that does not correspond to any of the first plurality of nodes.” EX1010. As explained for element [1.7], Section VIII.D.7 *above*, Brezinski discloses identifying a newly detected entity in newly received event data and adding it to a graph as a new node. While such disclosure was mapped to a “first entity” for purposes of element [1.7], such disclosure equally satisfies the “second entity” for purposes of element [11.9].

5. **Elements [11.10]**

[11.10] identify a second security vulnerability associated with the second entity,

132. As explained for element [11.6], Section VIII.J.2 *above*, Brezinski teaches presenting risk metrics (i.e., security vulnerabilities) for entities in a

graphical representation of nodes corresponding to respective entities. Accordingly, for a second entity Brezinski teaches identifying a risk metric (i.e, security vulnerability) associated with such entity in order to present the entity's risk metric in the network graph. Alternatively, as explained for element [11.6], Crabtree teaches including in a graph security vulnerability information associated with the graph's entities, and therefore teaches identifying a second security vulnerability associated with a second entity for the same reasons as for a first security vulnerability associated with a first entity.

6. Elements [11.12]

133. Element [11.12] differs from element [1.9] only in that the former recites a "second entity" and "second node" where the latter respectively recites a "first entity" and "first node." EX1010. As explained above (*above* Section VIII.J.4) Brezinski's disclosures teaching a "first entity" for purposes of claim 1 also equally apply to the "second entity" for purposes of claim 11. And, as explained for element [1.9], Section VIII.D.9 *above*, Brezinski teaches modifying a graph to add a new node corresponding to a newly identified entity. While such disclosure was mapped to a "first node" for purposes of element [1.9], such disclosure equally satisfies the "second node" for purposes of element [11.12].

7. Elements [11.13]

[11.13] and wherein the modified representation of the first graph includes information about the second security vulnerability associated with the second entity,

134. As explained for element [11.10], Section VIII.J.5 *above*, Brezinski teaches identifying a second security vulnerability associated with a second entity. And as explained for element [11.6], Section VIII.J.2 *above*, Brezinski teaches storing security risk data for entities in a database and presenting risk metrics (i.e., security vulnerabilities) for such entities in a graphical representation of nodes corresponding to respective entities. Accordingly, Brezinski teaches including in its modified graph information about the second vulnerability for the same reasons as explained for the first vulnerability of element [11.6].

8. Elements [11.14]

135. Element [11.14] differs from element [1.10] only in that the former recites “a first event” while the latter recites “an anomalous event.” EX1010. Brezinski teaches an anomalous event for the reasons explained for element [1.10],

above Section VIII.D.10, and this same anomalous event is a “first event” for purposes of element [11.14].

K. Claim 13

[13]. The computer system of claim 11, wherein the further correlation using the modified representation of the first graph further identifies a plurality of respective correlated edges that each involve the respective event or resource.

136. Brezinski teaches this limitation. As explained for limitations [1.10] and [11.14], *above* Sections VIII.D.10 & J.8, Brezinski teaches a first correlation identifying nodes corresponding to respective resources (e.g., a device). And as explained for limitations [1.11] and [11.15], *above* Sections VIII.D.11 & J.1, Brezinski teaches a further iteration identifying edges and nodes connecting the first plurality of correlated nodes. A POSITA would understand the obvious use case of one of the first plurality of correlated nodes having multiple outgoing connections/edges to multiple neighboring nodes. *See e.g.*, EX1004, Fig. 4 (showing, as an example, Vertex 402(4) connected by multiple edges to neighboring Vertices 402(6) and 402(7)). In the situation where corresponding risk and rarity metrics exceeded thresholds, then the Brezinski analysis device would include the multiple outgoing edges (i.e., a plurality of respective correlated edges that each involve the respective resource) as anomalous data in a further correlation. Alternatively, the node included in the first plurality of correlated nodes may have multiple connections and corresponding edges to the same neighboring node with

each edge included as anomalous activity data. *See e.g., id.*, Fig. 4 (showing multiple edges connecting Vertex 402(5) and Vertex 402(8)). Accordingly, those multiple outgoing edges constitute a plurality of respective correlated edges that involve the respective resource because they each connect to the node corresponding to the respective resource.

L. Claim 14

[14] The computer system of claim 11, wherein the first event is an anomalous event.

137. As explained for element [11.14], Section VIII.J.8 *above*, that element differs from element [1.10] in that the former recites “a first event” while the latter recites “an anomalous event.” Brezinski teaches an anomalous event for the reasons explained for element [1.10], *above* Section VIII.D.10, and this same anomalous event is a “first event.”

M. Claims 15 & 17

138. Aside from their claim dependencies, claims 15 and 17 are substantively the same as claims 5 and 6, respectively, with the only difference being that claims 15 and 17 recite a “first event” and claims 5 and 6 recite an anomalous event. EX1010. As explained for element 11.14 and claim 14, *above* Sections VIII.J.8 & L, Brezinski teaches a first event that is an anomalous event. This first

event meets the limitations of claim 15 and 17 for the same reasons explained for claims 5 and 6, *above* Sections VIII.F & G, respectively.

N. Claim 20

139. Aside from their claim dependencies, claim 20 is substantively the same as claim 9 and Brezinski teaches claim 20 for the same reasons as explained for claim 9, *above* Section VIII.H.

O. Claim 22

[22] The computer system of claim 11, wherein the node associated with the first event corresponds to a third security vulnerability.

140. As explained for limitation [11.6], *above* Section VIII.J.2, Brezinski teaches presenting risk metrics (i.e., security vulnerabilities) for nodes corresponding to entities in its graphical representation, which would also apply to the node corresponding to the first event. Accordingly, Brezinski teaches associating a risk metric (i.e, third security vulnerability) to the node associated with the first event. Alternatively, as explained for element [11.6], Crabtree teaches including in a graph security vulnerability information associated with the graph's entities, and therefore teaches identifying a third security vulnerability corresponding to an entity (and its corresponding node) associated with the first event for the same reasons as for a first security vulnerability associated with a first entity.

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 3rd, 2026

A handwritten signature in black ink, appearing to read 'J. R. Black, Jr.', written over a horizontal line.

Signature

Appendix A

John R. Black, Jr.

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

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

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

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

**Journal
Publications**

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

**Book
Chapters**

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

**Conference
Publications
(Refereed)**

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

**Conference
Publications
(cont.)**

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

**Workshop
Publications
(Non-
Refereed)**

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

Selected Talks

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

Patents

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

Funding

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

Teaching History

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

CMPSC 290G — Intro to Cryptoanalysis (UCSB).

CS 365 — Discrete Mathematics (UNR).

CS 425 — Software Engineering (UNR).

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Graduate Students

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

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

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

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

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

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.

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

**External
Service**

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

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

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

Graduate Student Mixer organizer, CRYPTO 2005.

**Internal
Service**

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

Chair, Space Committee, 2012–2015.

Liaison, Casey Feldman Foundation, 2010–2015.

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

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

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

Chair, Departmental Search Committee, 2008–2009.

Member, Graduate Committee, 2005–2006.

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