

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

MICROSOFT CORPORATION,
Petitioner,

v.

QOMPLX LLC,
Patent Owner.

IPR2026-00182
Patent: USP 12,218,934
Issued: February 4, 2025
Application No. 18/885,510
Filed: September 13, 2024

Title: CONTEXTUAL AND RISK-BASED
MULTI-FACTOR AUTHENTICATION

DECLARATION OF DR. JOHN R. BLACK, JR.

MICROSOFT CORP.
EXHIBIT 1003

TABLE OF CONTENTS

	Page
I. INTRODUCTION AND ENGAGEMENT	4
II. BACKGROUND AND QUALIFICATIONS.....	4
III. MATERIALS CONSIDERED AND INFORMATION RELIED UPON REGARDING '934 PATENT.....	9
IV. UNDERSTANDING OF PATENT LAW	14
V. LEVEL OF SKILL IN THE ART, AND STATE OF THE ART.....	18
A. Person Of Ordinary Skill In The Art.....	18
B. State Of The Art	19
1. Authentication With Username And Password Was Long-Known And Regularly Used	20
2. Additional (i.e., Multi-Factor) Authentication Was Known And Regularly Used For Added Security	24
3. It Was Well Known To Implement Systems To Detect Anomalous Log-In Attempts, E.g., Brute Force Attacks	39
4. Virtual Machines.....	46
VI. THE '934 PATENT	47
A. The Claims' Earliest Effective Filing Date.....	47
B. The '934 Patent's Specification	47
VII. CLAIM CONSTRUCTION	50
VIII. GROUNDS 1-2: CLAIMS 1-30 ARE OBVIOUS OVER KIRTI (EX1004) ALONE OR COMBINED WITH COFFIN (EX1005).....	50
A. Kirti (EX1004)	51
B. Coffin (EX1005).....	53
C. Combining Kirti And Coffin.....	54

1.	Claim 1	58
	a) Element [1.1]	58
	b) Elements [1.2], [1.4]	59
	c) Element [1.3]	65
	d) Element [1.5]	69
	e) Elements [1.6]-[1.7]	73
	f) Element [1.8]-[1.11]	84
2.	Claim 2	88
3.	Claim 3	89
4.	Claim 4	92
5.	Claim 5	93
6.	Claim 6	98
7.	Claim 7	100
8.	Claim 8	101
9.	Claim 9	103
10.	Claim 10	104
11.	Claims 11-17	106
12.	Claims 18-25 And 26-30	106

IX.	GROUND 3-4: CLAIMS 10 AND 25 ARE OBVIOUS OVER KIRTI AND VEMULAPALLI (GROUND 3) OR KIRTI, COFFIN, AND VEMULAPALLI (GROUND 4)	107
-----	---	-----

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. 12,218,934 (hereinafter “the ’934 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 1-30 (each a “Challenged Claim” and collectively the “Challenged Claims”) of the ’934 patent would have been obvious to a person having ordinary skill in the art (“POSITA”) as of October 19, 2017. It is my opinion that each Challenged Claim would have been obvious to a POSITA after reviewing the prior art discussed herein. My analysis would not be materially different even as of October 28, 2015.

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 call “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 '934 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. 12,218,934 (“’934 patent”)
1002	File History of U.S. Patent No. 12,218,934
1004	U.S. Patent No. 10,063,654 (“Kirti”)
1005	David Coffin, <i>Expert Oracle and Java Security: Programming Secure Oracle Database Applications with Java</i> , Apress (2011) (“Coffin”)
1006	U.S. Patent No. 8,572,391 (“Golan”)
1007	U.S. Patent No. 7,979,899 (“Guo”)
1008	International Publication No. WO 2015/109947 (“Zhang”)
1009	U.S. Patent No. 8,312,540 (“Kahn”)
1010	U.S. Patent No. 10,021,108 (“Mankovskii”)
1011	U.S. Patent No. 7,908,645 (“Varghese”)
1012	U.S. Patent Application Pub. No. 2007/0079135 (“Saito”)
1014	U.S. Patent No. 9,237,143 (“Dotan”)
1015	U.S. Patent No. 9,148,424 (“Yang”)
1016	U.S. Patent No. 9,319,419 (“Sprague”)
1017	U.S. Patent Application Pub. No. 2008/0249947 (“Potter”)
1018	International Pub. No. WO 2015/158874 (“Enqvist”)
1019	U.S. Patent No. 5,774,525 (“Kanevsky”)

No.	Description
1020	Olli Jarva, <i>Intelligent Two-Factor Authentication: Deciding Authentication Requirements Using Historical Context Data</i> , Aalto Univ. (May 13, 2014) (“Jarva NPL”)
1021	U.S. Patent No. 9,298,890 (“Bajenov”)
1022	Affidavit of Mina Ching dated October 14, 2025 (“Azure Wayback”)
1023	U.S. Patent No. 9,955,349 (“McClintock”)
1024	U.S. Patent No. 11,184,766 (“Lord”)
1025	U.S. Patent No. 11,184,392 (“Thomas”)
1026	U.S. Patent No. 9,160,726 (“Kaufman”)
1027	U.S. Patent No. 9,122,866 (“Kolman”)
1028	U.S. Patent Application Pub. No. 2003/0115142 (“Brickell”)
1029	U.S. Patent Application Pub. No. 2014/0208419 (“Chang”)
1030	U.S. Patent Application Pub. 2018/0189470 (“Kim”)
1031	U.S. Patent Application Pub. No. 2014/0289833 (“Briceno”)
1032	U.S. Patent Application Pub. No. 2016/0191512 (“Tatourian”)
1033	U.S. Patent Application Pub. No. 2015/0150090 (“Carroll”)
1034	U.S. Patent Application Pub. No. 2017/0063896 (“Muddu”)
1035	U.S. Patent No. 10,432,605 (“Lester”)
1036	U.S. Patent Application Pub. No. 2010/0211996 (“McGeehan”)
1037	U.S. Patent No. 9,779,236 (“Abrams”)
1038	U.S. Patent No. 9,639,678 (“Moore”)

No.	Description
1039	U.S. Patent No. 9,648,034 (“Bailey”)
1040	U.S. Patent Application Pub. No. 2018/0027006 (“Zimmermann”)
1041	Shammi Ishara Hewamadduma, <i>Detection and Prevention of Possible Unauthorized Login Attempts Through Stolen Credentials from a Phishing Attack in an Online Banking System</i> , IEEE Int’l Conf. on Rsch. and Innovation in Info. Sys., at 13-18 (ICRIIS 2017) (“Hewamadduma”)
1042	U.S. Patent No. 8,266,682 (“Lee”)
1043	U.S. Patent No. 9,397,996 (“Roskind”)
1044	Affidavit of Mina Ching dated November 14, 2025 (“Phone Factor + Oracle Wayback”)
1045	Rajkumar Buyya, et al., <i>Cloud Computing Principles and Paradigms</i> , Wiley (2011) (“Buyya NPL”)
1046	U.S. Patent No. 9,448,852 (“Vemulapalli”)
1047	Ilia Pietri and Rizos Sakellariou, <i>Mapping Virtual Machines onto Physical Machines in Cloud Computing: A Survey</i> , ACM Computing Survs., Vol. 49, No. 3, Article 49 (Oct. 2016) (“Pietri”)
1049	Affidavit of Mina Ching dated November 14, 2025 (“Coffin Wayback”)
1050	Lucas Jellema, <i>Just Launched: The Oracle Identity Cloud Service – for Authentication and Authorization Across the Cloud and on Premises</i> , AMIS Tech. Blog (Nov. 2, 2016), available at: https://technology.amis.nl/platform/just-launched-the-oracle-identity-cloud-service-for-authentication-and-authorization-across-the-cloud-and-on-premises/

No.	Description
1051	John Ribeiro, <i>Oracle Will Acquire Cloud Security Vendor Palerra</i> , FoundryCo, Inc. (Sep. 18, 2016), available at: https://www.networkworld.com/article/955458/oracle-will-acquire-cloud-security-vendor-palerra-2.html
1052	Sarah Kuranda, <i>Oracle Acquires Cloud Access Security Brokerage Startup Palerra</i> , The Channel Co. (Sep. 19, 2016), available at: https://www.crn.com/news/security/300082128/oracle-acquires-cloud-access-security-brokerage-startup-palerra
1053	<i>Hiding in Plain Sight, How a Cloud Access Security Broker With Built-In User Behavior Analytics Unmasks Insider Threats in the Cloud</i> , Oracle (Jan. 2017)
1054	Ganesh Kirti, <i>Dealing With Dropbox: Unmasking Hackers With User Behavior Analytics</i> , Cloud Sec. Alliance (Sep. 7, 2016), available at: https://cloudsecurityalliance.org/blog/2016/09/07/dealing-dropbox-unmasking-hackers-user-behavior-analytics
1055	Mitch Wagner, <i>Oracle Buys Palerra to Boost Cloud Security</i> , Informa TechTarget (Sep. 18, 2016), available at: https://www.lightreading.com/security/oracle-buys-palerra-to-boost-cloud-security
1058	Affidavit of Mina Ching dated December 12, 2025 (“2016 White Paper”)
1059	U.S. Patent No. 10,742,647 (“Crabtree 647”)
1060	Guohui Wang and T.S. Eugene Ng, <i>The Impact of Virtualization on Network Performance of Amazon EC2 Data Center</i> , IEEE INFOCOM 2010, at 1163-1171 (Mar. 2010) (“Wang NPL”)

No.	Description
1061	U.S. Patent Application Pub. No. 2011/0314558 (“Song”)
1062	U.S. Patent No. 11,218,474 (“Crabtree 474”)
1063	U.S. Patent Application Pub. No. 2015/0319185 (“Kirti App.”)
1065	Patent Comparison Chart

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 '934 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 '934 patent or any assignee of the '934 patent that any secondary considerations would rebut the obviousness of any Challenged Claim of the '934 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. The person of ordinary skill in the art for the purposes of the '934 patent in October 2017 ("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 or Ph.D., would compensate for less work experience and vice versa. For purposes of my analysis herein, there would be no material difference in the definition of a POSITA in October 2015.

36. A POSITA would have been familiar with common, then-existing technologies for securing networks, applications, databases, and other digital systems, including the following state of the art:

- Authentication techniques based on an identifier (e.g., username) and password (*infra* Section V.B.1);
- Additional or “multi-factor” authentication techniques used for added security (e.g., a one-time passcode or “OTP” sent to a mobile phone) (*infra* Section V.B.2);
- Tracking login attempts to determine standard behaviors and detect anomalous login attempts, in order to remediate security threats, such as a hacker cracking a user’s password (*infra* Section V.B.3); and
- Using virtual machines in cloud computing systems (*infra* Section V.B.4)

See also infra Section V.B.

B. State Of The Art

37. The knowledge and skill of a POSITA included the following claim-recited features, as reflected in the ’934 patent itself, background references, and the Kirti and Coffin prior art. These features were well-known, widely deployed (including in commercial systems), discussed in numerous patents, and often combined with one another.

1. Authentication With Username And Password Was Long-Known And Regularly Used

38. It was well known to require a user to enter a username (or other unique identifier) and a password to gain access to their account. The '934 patent acknowledges that a “login and password” was a “traditional” way to authenticate a user. EX1001, 2:22. Indeed, this authentication method was commonly employed by at least 2004.

- EX1006, 1:47-48 (describing in the Background section that a “login and password” was “the most prevalent method of authentication”), 6:1-33, 9:44-51, 13:64-66, 14:31-43;
- EX1007, 1:14-17 (describing in the Background Section that “[a] typical user authentication mechanism verifies a user’s credentials, such as by validating a username and password, as the user attempts to access a protected resource (e.g., a server accessed via the Internet)”), 1:43-50, 3:4-20, 8:4-13;
- EX1008, [0078] (“During login user account[], the user is requested to input login credentials [], such as the user account name and the password for user account”), FIG. 4B;

- EX1009, 1:17-19 (“Commonly the authenticating information includes a username and user-entered password”), 2:7-16, 6:34-50, 7:51-8:4, 11:48-12:3;
- EX1010, 1:18-37 (“Authentication may refer to the process of verifying that credentials provided by a user or entity are valid ... (e.g., confirming that a correct password has been entered for a given username)”), 10:50-58, 11:32-60, FIGS. 2-3C;
- EX1011, 1:45-53, 28:40-48, FIGS. 1, 4B, 13C;
- EX1012, [0005] (“As a representative method for conducting authentication ... the method has been known in which a user enters a user name and a password”);
- EX1014, 5:67-6:4 (“During an authentication, [the authentication system] first checks that a user ID presented in the authentication request is found among the recorded user IDs, then checks that a password also presented in the authentication request matches the password recorded with the valid user ID.”);
- EX1015, 1:14-16 (“Username and secret password pairs are a standard form of authentication in significant numbers of systems.”);
- EX1044, 9/34 (“[T]he dominant authentication system in production today is based on usernames and passwords.”).

39. Zhang, Varghese, and Mankovskii, for example, illustrate exemplary user interfaces requiring a user to enter their username and password to access their account.

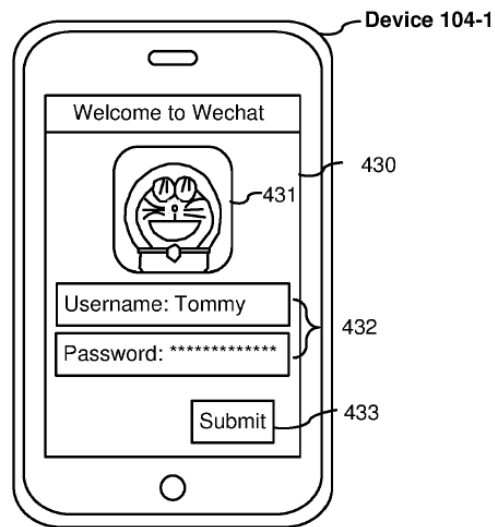
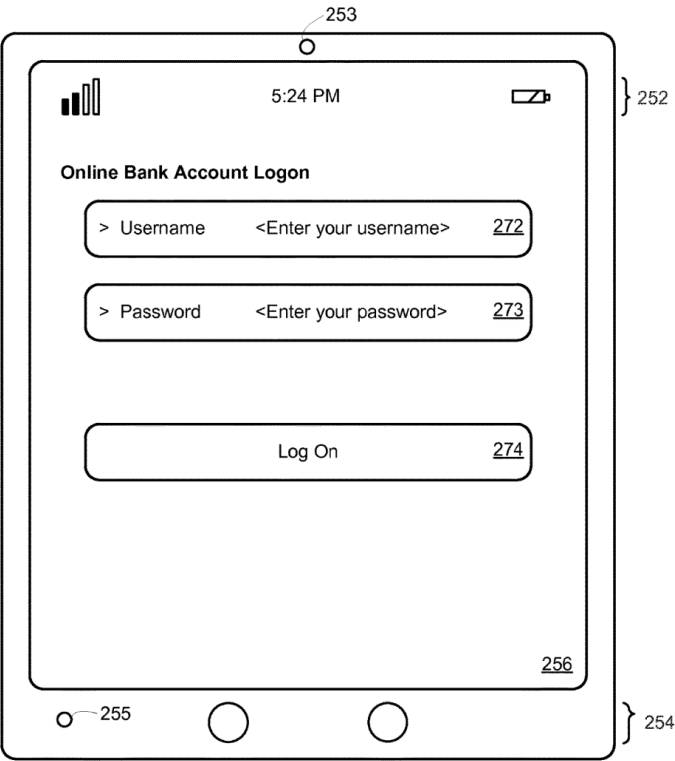


Figure 4B

EX1008, FIG. 4B.



Mobile Device 141

FIG. 2

EX1010, FIG. 2

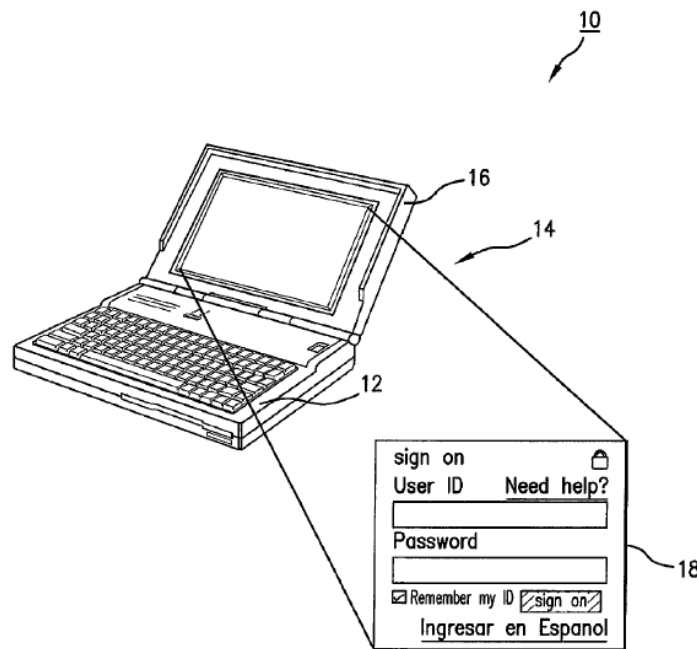


FIG. 1
PRIOR ART

EX1011, FIG. 1.

**2. Additional (i.e., Multi-Factor) Authentication
Was Known And Regularly Used For Added Security**

40. It was also well known to require additional forms of authentication in combination with a username and password. This technique, often called “multi-factor authentication,” increased security and reduced access risks posed by stolen or compromised usernames and passwords.

- EX1005, 177 (describing various additional verification methods that “assure [] that the person sitting at the keyboard is who they claim to be”);

- EX1006, 1:20-49 (describing how traditional username and password protection is no longer sufficient due to the sophistication of fraudsters), 4:41-65 (describing requiring “an additional authentication step (e.g., a recent transaction number, an answer to a secret question)”), 12:41-57 (describing other additional authentication steps, such as a dynamic out-of-band PIN), 19:11-17 (claim 20);
- EX1007, 1:14-21 (“[I]f the username and password are stolen by a malicious entity, the entity may access the user’s account from anywhere in the world”), 1:36-56, 7:25-35 (describing requiring two factor authentication such as authentication via a trusted device, response to secret question, fingerprint scan, etc.);
- EX1008, [0040], [0103]-[0104] (describing scanning a QR code as an additional form of verification), FIGS. 4C, 4F;
- EX1010, 5:61-6:20, 13:12-55 (“[A]dditional authentication information may comprise additional multi-factor authentication information, such as an answer to a challenge question”), 15:37-51, FIG. 3;
- EX1011, 9:64-10:2, 11:16-23, 11:38-46, 12:31-52 (describing secondary authentication including answering a personal question and/or providing biometric identification information);

- EX1012, Abstract, [0013] (describing a one-time password);
- EX1014, 4:17-19 (describing requiring a “one-time password (OTP) process in addition to a conventional password-based process”);
- EX1016, 1:18-26 (“Multi-factor authentication is an approach that has recently become more widespread for use in online authentication systems Multi-factor online authentication typically requires a user to enter a username and password, as well as pass an additional identification test specified by the online service provider.”), 1:38-47, 2:7-14, 10:34-45 (describing security questions, one-time passwords, and biometric scans);
- EX1017, [0003] (describing how “security laws, regulations, and internal rules mandate that certain institutions ... utilize a multi-factor authentication technique”);
- EX1018, 3:3-15 (describing several forms of authentication and describing that “[m]ulti-factor authentication, or two-factor authentication, is an approach to authentication which requires the presentation of two or more of the above mentioned three authentication factors”), 4:13-14 (“Authentication methods that depend on more than one factor are more difficult to compromise than single-factor methods”);

- EX1019, Abstract, 2:5-23, 5:46-65, 10:17-11:3 (describing requiring an answer to a secret question, such as a birth date), 11:10-36;
- EX1020, 7 (describing how one-time passwords were well known);
- EX1021, 7:20-39 (describing requiring “an additional security challenge,” such as “security related questions,” “a security code ... sent to the user through email or an SMS message” a “social CAPTCHA,” or other types of security challenges that were “well known to those skilled in the art,” if a user login is deemed suspicious);
- EX1022, 11, 15-16 (describing how Microsoft “Azure Multi-factor authentication is a method of verifying who you are that requires the use of more than just a username and password. It provides a second layer of security to user sign-ins and transactions. ... The following are a list of methods that can be used for this second verification. ... Phone Call[:] A call is placed to a user’s smart phone asking them to verify that they are signing in by pressing the # sign ... Text Message[:] A text message will be sent to a user’s smart phone with a 6 digit code ... Mobile App Notification[:] A Verification request will be sent to a user’s smart phone asking them to complete the verification by selecting Verify from the mobile app ... Verification code with Mobile

App[:] A Verification code will be sent to the mobile app that is running on a user's smart phone.”);

- EX1023, 1:36-40 (describing the demand for and ease in implementing multifactor authentication systems), 2:7-49 (describing requiring multifactor authentication when an anomaly is detected), 7:61-8:22, 9:23-49 (describing various types of information that may be used as part of additional verification challenge), 19:61-20:3;
- EX1024, 1:17-19 (“Conventional approaches that implement single factor authentication using usernames and passwords are clearly insufficient to safeguard access to protected services.”);
- EX1025, 43:30-52 (“[W]hen the pattern of notifications indicates a presence of a malicious attempt at lateral movement, the process 900 may include any suitable remediation. ... [S]teps may be taken to address a user, such as by ... requiring a step-up authentication for the endpoint (e.g., by requiring manual re-entry of credentials, by presenting a security question, or some other user verification procedure.”);
- EX1044, 9-12/34 (describing requiring a user to answer a phone call and provide a PIN after entering a username and password);

- EX1058, 15/24 (“Most business entities today are looking at some form of multifactor authentication to reduce risk. ... IDC defines dynamic (or adaptive) authentication as the evaluation of available information (i.e., IP address, location, time of day, and biometrics) to prove an identity after a user session has been initiated. The key to dynamic authentication is to use additional methods for a user to verify his/her identity in circumstances that are potentially higher risk — and fewer methods in lower-risk environments. The goal is to be more secure while providing a productive environment for users. Dynamic authentication has been used in high-risk industries such as financial services for some time and is now being extended into other verticals.”), 20/24.

41. Indeed, the patent itself acknowledges that “[m]ulti-factor authentication (MFA) is widely used today as an additional verification step often used in conjunction with a traditional login and password as a way to further secure a user’s online accounts.” EX1001, 2:20-23.

42. Some of these known verification methods included “a second password or PIN code ... a CAPTCHA ... personal questions ... a biometric scan[], like fingerprint, retina, or facial recognition ... [and] pass codes sent to [] e-mail, pager, or cell phone.” EX1005, 177. Users could also “respon[d] to an email sent to

a registered email address” (EX1011, 12:49-50), “answer[] the phone and ... enter[] an optional PIN” (EX1044, 9/34), and/or scan an encoded image such as a QR code (e.g., EX1016, 17:2-10; EX1008, [0040], [0103]-[0104], FIGS. 4C, 4F). *See also*, EX1001, 2:23-27 (describing “one-time use codes,” “uniquely generated link,” “authenticator devices and apps”).

43. Indeed, challenge questions were known by at least 1997 (*see* EX1019), one-time passwords were commonly employed by at least 2007 (*see* EX1017), and biometric scans were commonly employed by at least 2006 (*see* EX1011).

- EX1005, 177-208 (describing how to send one-time passcodes to a separate device);
- EX1006, 12:41-57 (describing several additional authentication steps, including a dynamic out-of-band PIN);
- EX1007, 1:36-56 (describing how “fingerprint scan, retina scan, HIP solutions, secret questions, etc.” were known), 7:25-35 (describing requiring two factor authentication such as authentication via a trusted device, response to secret question, fingerprint scan, etc.);
- EX1010, 11:18-20 (“An identity of the end user may be determined by applying facial recognition techniques to a captured image of the end user.”);

- EX1011, 11:34-45 (describing a secondary authentication that includes “biometric identification information (e.g., fingerprints, retinal scans, or the like)”), 12:31-52, 14:40-44, FIG. 14 (describing biometrics to include “voiceprint”);
- EX1017, [0003] (“[U]sers are sometimes required to enter a username and password (knowledge) along with a one time password (provided information . . .)”);
- EX1019, Abstract, 2:5-23, 5:46-65, 10:17-11:3 (“[I]f the user enters the correct password ... system control [] may activate the dialog module [] to begin questioning of the user. ... For example, the dialog module [] may ask the user to enter his/her birth date. ...”), 11:10-36 (claims 1-3);
- EX1020, 7 (describing how, by 2011, one-time passwords were “deployed as an optional feature by all major Internet companies, for example, Facebook [], Twitter [], LinkedIn [], Google [] and Amazon []”);
- EX1044, 9/34 (describing requiring a user to answer a phone call and enter a PIN after entering their username and password).

44. Further, it was known for a single system to offer several multi-factor verification options, each of which prompts a user to complete the additional

verification method (e.g., enter the one-time password) and checks to ensure the user completed the additional verification method correctly (e.g., the user-entered passcode matches the passcode sent to the user).

- EX1005, 177 (describing a plurality of additional verification options), 183 (“Once the user receives the two-factor pass code, they will submit that along with their request for data.”), 190, 193, 194 (“[W]e test the two-factor code to see if it passes muster by calling the `is_currently_cached` function. If the two-factor code is good, we set the secure application role; however, if not we let them know by raising a `NO_DATA_FOUND` exception: they entered the wrong code, or perhaps just an old (older than 10 minutes) code.”), 201 (“[W]e want to cache [the code] for comparison to any code the user enters.”), 206 (“One definite potential error is the possibility that the user has entered a wrong or old code. . .”), 207 (“Users will have to enter those 2-factor codes in order to get access to our application data.”);
- EX1006, 6:26-33 (“[I]f there is a need for extra authentication, the risk based authentication server [] may ask the user [] to perform the authentication in the method selected according to the risk level. In operation 207, the data may be verified. In operation 208, if the data matches, the risk based authentication server [] may communicate with

the risk based authentication local module [] (e.g., via browser redirection) and the user [] may be allowed to proceed.”), 11:51-12:19 (“... Following assessment and prioritisation of available authentication methods it may be necessary that the Transaction provider check what constraints it has with respect to deploying each of the methods, and selecting the optimal method that meets the constraints.”), 14:1-4, 17:11-16 (“[A]n authentication and collection server may for example display a page requesting the customer or user to perform another level of authentication; for example enter more information. The authentication and collection server may verify the data entered.”), 19:11-22;

- EX1007, Abstract, 1:36-56, 7:25-25;
- EX1008, [0030], [0075]-[0077] (“Server system [] promotes [] the user of client device 104-1 to scan the encoded image displayed on client device 104-2. ... [S]erver system [] receives the scanned encoded image from client device 104-1 [and] verifies []the scanned encoded image. For example, server system [] determines whether the scanned encoded image matches the encoded image sent to client device 104-2.”), [0079], [0082], [0103]-[0106], FIGS. 4A, 4C, 4F;

- EX1010, 5:61-6:20, 13:12-39 (“[A]n additional authentication challenge request may be transmitted to the computing device making the access request If a correct answer is provided by the computing device, then access may be granted to the protected resource.”), 15:37-45, 21:59-61;
- EX1016, 3:56-5:25, 10:34-45, 12:47-14:16, 16:10-17:15, 21:63-22:32 (claims 13-20) (describing a multitude of additional verification options and checking whether user-entered responses such as a password, speech, fingerprint, facial scan, etc., is successful / matches what the system has on file);
- EX1017, Abstract, [0006]-[0007], [0030]-[0033] (describing that “the user is prompted for a one time password,” “a one time password is received from the user,” “the system determines whether the received one time password is valid ... if it matches [the] one time password which was previously provided to ... the user,” and “[i]f the system determines that the received one time password is valid, the system authenticates the user” and “the user can access the service”); EX1021, 7:20-49 (describing several types of “additional security challenge[s] [that] may be provided to the user” and determining the “success or

failure of the [additional] authentication procedure ..., at which access to the user account is either granted ... or denied”);

- EX1018, 3:3-4:18 (describing a plurality of additional verification methods, such as “one-time passwords,” “biometric information,” and “mobile phone,” and describing how the system must validate the user-provided information “for authentication to occur”);
- EX1019, Abstract, 2:9-22, 4:38-64, 5:46-65 (“Dialog interpretation includes the function of determining if a response to a question is correct Questions generated by the Q&A module [] are ... routed to the user input/output device [] for communication to the user. ...”), 7:65-9:7, 10:50-58;
- EX1020, 7-8;
- EX1022, 15-16 (“Phone Call[:] A call is placed to a user’s smart phone asking them to verify that they are signing in by pressing the # sign ... Text Message[:] A text message will be sent to a user’s smart phone with a 6 digit code ... Mobile App Notification[:] A Verification request will be sent to a user’s smart phone asking them to complete the verification by selecting Verify from the mobile app ... Verification code with Mobile App[:] A Verification code will be sent to the mobile app that is running on a user’s smart phone.”);

- EX1023, 5:20-26, 7:61-8:22 (“... a set of authentication challenges ... are presented to the user at the client Responses [] to one or more of the challenges [] are obtained from the client [] and then evaluated by the response validation service []. The response validation service [] is configured to determine whether the response [] to a challenge [] is correct. ...”), 18:28-40, 19:64-20:3.

45. Further, it was known to select a particular authentication method, including an additional authentication method from a plurality of authentication methods, such as based on the risk level of the access request, which authentication methods are available, cost, user preferences, etc.

- EX1006, 4:50-53, 6:25-29, 11:51-12:19, 19:11-22 (claims 20-21);
- EX1011, 6:21-39, 8:18-32, 9:59-10:6, 11:30-46, 12:31-52, 13:15-41, 14:30-44, 27:37-52, 27:53-62, 28:31-39 (“Additional authentication, if needed, is performed 514 in accordance with the rules specified by the service provider”), 29:6-17;
- EX1016, 3:24-32, 3:66-4:2, 4:16-17, 5:31-51, 9:9-26, 10:34-45, 12:32-46, 16:10-18, 16:55-17:15;
- EX1026, Abstract, 1:38-41, 2:5-14, 4:65-5:1, 5:21-25 (“As shown in FIG. 4, the exemplary authentication method selector 400 selects a suitable authentication method based on attributes 410 of the available

authentication methods and a context 420 of the activity demanding the authentication”), 5:26-34, 6:64-7:2, 7:29-34, FIG. 4;

- EX1027, 6:45-53, 7:41-47, 8:18-26 (“At step 320, the method comprises selecting a user authentication challenge to issue to the user in response to receiving the authentication request”), 8:30-40, 8:41-45, 8:57-65, 9:6-20;
- EX1028, Abstract, [0026], [0044], [0045], [0047], [0050], [0051], [0053], [0072], FIG. 10;
- EX1029, [0049], [0050], [0052];
- EX1030, [0135]-[0136], [0137] (“Furthermore, according to some embodiments, the authentication device may determine an additional authentication method based on an authentication distance to an authentication target, a pre-set user preference, a frequency of using an external device, type of authentication data that may be obtained from the external device, obtained authentication score, and reliability information, as described above”);
- EX1031, Abstract, [0090], [0128], [0129], [0230], [0245], [0252] (“In any case, at 2405 a set of one or more authentication techniques are selected based on the requirements specified via the rule(s) and the authentication capabilities of the client.”), [0513];

- EX1032, Abstract, [0011], [0027], [0074], [0092], [0098], [0108], [0126] (“There is further disclosed an example, wherein the predictive security engine is further operable for selecting an additional authentication mechanism based at least in part on the predictive authentication score.”), [0138];
- EX1033, [0050], [0055] (“The authentication mechanisms control 322 automatically recommends an authentication mechanism to the sender based on the trust score of the selected user (e.g., trust score 312 of the signer).”), [0060];
- EX1061, [0040], [0042] (“In some embodiments, the second authentication mechanism may be chosen from among a set of possible authentication mechanisms based on the severity of the change in context data.”).

46. Indeed, more than 13 years before the '934 patent's alleged priority date, Golan disclosed “selecting the optimal [additional authentication] method” (EX1006, 12:18-19) based on, e.g., a risk assessment, available authentication methods, and deployment constraints. EX1006, 4:50-53, 6:25-29, 11:51-12:19, 19:11-22 (“[D]etermining which additional authentication options are available, wherein the one or more additional authentication details are selected from among the determined available authentication options”).

3. It Was Well Known To Implement Systems To Detect Anomalous Log-In Attempts, E.g., Brute Force Attacks

47. It was well known for systems to detect anomalous activities (including anomalous login requests) to, for example, prevent, report, and/or remediate security breaches (e.g., by requiring multi-factor authentication).

- EX1004, Abstract, [0004], [0017], [0035], [0040], [0042] (“[W]hen a threat is detected ... apply[] remedial measures, such as adding additional steps to authentication.”), [0043], [0044] (“As will be discussed further below, embodiments of the invention may include remediation functions that provide manual and/or automated processes in response to threats.”), [0066]-[0068], [0071] (“[A] recommendation engine tracks user activity for anomalous behavior to detect attacks and unknown threats.”), [0073] (“[T]echniques such as outlier detection establish a baseline that is useful for detecting anomalous activities. Such anomalous activities along with contextual threat intelligence can provide more accurate prediction of threats with low prediction errors.”), [0079]-[0086], [0099]-[0116] (describing various “behavior analytics algorithms” for detecting “anomalous activity” in the user’s activity data to “determine the likelihood of various threats”), [0119], [0120] (“The threat model(s) can be used to determine [] the likelihood

of anomalous activity such as, but not limited to, using behavior analytics algorithms as discussed further above.”), [0124], claim 1, FIG. 5B;

- EX1009, 11:22-37, 16:19-18:6 (describing requiring additional authentication if the login is deemed suspicious, such as if the login is from an unusual time of day or unusual network subnet);
- EX1010, Abstract, 6:21-52 (“The improved security may derive from enforcing additional multi-factor authentication upon detection of an anomaly ...”), 12:44-13:39, 15:3-51;
- EX1011, 4:36-41 (“Another advantage of the present invention is that it enables a service provider to identify possible in-process fraudulent authentication transactions, based on both user and device historical data analysis. Transactions can be approved, declined, or put on hold for verification based [] a set of predetermined rules.”), 5:6-22, 9:66-10:2, 11:38-46, 12:31-52 (“[W]hen the system finds an elevated risk score ... it performs an “online or out of band secondary authentication, preferably based on a challenge response model ...”), 16:11-19:28;
- EX1018, 7:11-15, 11:6-8 (“A user might for instance usually buy his electrical devices in black. If he then suddenly buys a pink device, the

behavior will create an anomaly in the user profile and might launch additional security checks.”);

- EX1021, 5:65-8:64 (describing detecting suspicious behavior, such as deviation from a usual geographic location);
- EX1023, Abstract (“Multifactor authentication occurs during a communication session in response to detecting a trigger event, such as an anomalous condition. Historical metrics, such as performance metrics (e.g., rendering speeds), behavioral metrics (e.g., click-stream behavior), environmental metrics (e.g., noise), etc., can be used as a baseline to compare against metrics for a current communication session. An anomalous condition, such as a current session metric exceeding a threshold, can result in an authentication service transmitting a multifactor authentication request.”), 2:7-49, 10:40-65;
- EX1024, 3:22-30, 4:4-12 (“According to one embodiment, the method further comprises acts of: detecting an anomaly in one or more of: the patterns of movement, the patterns of activity, or the authentication history”);
- EX1025, Abstract (“Attempts at lateral movement are detected by monitoring failed login attempts across a number of endpoints in a network.”), 42:28-43:52;

- EX1034, Abstract (“By visually presenting analytical results scored with risk ratings and supporting evidence, the security platform enables network security administrators to respond to a detected anomaly or threat, and to take action promptly.”), [0137], [0140], [0147], [0149], [0151], [0152] (“In the batch processing path, historical data and third-party data are processed, optionally with the incoming real-time event data, to uncover, for example, more subtle anomalies and threats than the real-time processing path can uncover because of the real-time processing path’s responsive time constraints.”), [0255], FIG. 13B;
- EX1035, 2:15-25 (“[D]etermine whether the user behavior deviates from the historical user behavior above a threshold level, and generate an alert when the user behavior deviates above the threshold level.”), 3:4-16, 3:46-49 (“The information may be used to assess risk, and the method may further include creating an identity trust score for comparison to an activity risk score to determine if additional authentication is needed.”);
- EX1036, [0021]-[0023] (describing taking “appropriate remediation action” depending on the degree of suspicion associated with the user session);

- EX1037, 4:56-5:35, 5:60-10:16 (describing evaluating a user's risk level based on historical data);
- EX1038, 5:30-41, 7:8-37, 8:29-67;
- EX1039, 4:1-60, 24:3-25:3;
- EX1040, [0110] ("anomaly detection around login events"), [0132], [0151]-[0152], [0180], [0264], [0286], [0355], [0384], [0557], [0562], [0565]-[0566], [0570], [0574];
- EX1041, 1, 3-6 (describing methods for detecting anomalous login attempts that utilize stolen credentials from a phishing attack).

48. The '934 patent lists several types of cyberattack profiles as examples of anomalous activities, including a brute force attack in which a hacker tries different username/password combinations (EX1001, 8:35-46), and admits these cyberattack profiles were "just ... a very small sample of the cyberattack profiles known to those skilled in the field." EX1001, 8:45-46. Indeed, brute force attacks were well known.

- EX1004, 15:49-51 ("A brute force attack scenario may refer to an attacker's attempts to try many passwords in order to discover a correct password and compromise a user account.");
- EX1015, 2:35-36 ("[M]any different passwords are attempted with a single username.");

- EX1042, 2:9-23 (describing how a brute force attack “may obtain a user password by trying different username/password combinations until the malicious code successfully identifies a username and corresponding password ... that allow access to the network system”);
- EX1043, 2:7-11 (“Hackers can use computer programs capable of systematically trying various alphanumeric combinations to launch brute force attacks on systems they are trying to access until a combination is found that grants them access to the targeted system.”).

49. Another type of well-known anomaly is impossible travel (e.g., a user logs in from Portland, Oregon at 10:00 AM and minutes later logs in from Frankfurt, Germany), which can indicate credential theft and/or VPN abuse or proxy masking techniques typically employed by hackers.

- EX1004 19:14-15 (describing a “maximum distance between any two IP addresses used by a user within a 24 hour time period”), FIG. 5C (describing one “Cloud Threat Feature[]” as an “IP-GEO distance”); EX1011, 17:17-21 (“[A] user who submits requests that appear to originate from multiple machines, or multiple cities, or multiple locations over an impossibly short duration of time has been found likely to be [a] hacker.”);

- EX1022, 6 (describing one type of “Risk Event” as “Impossible travel to atypical locations”);
- EX1036, [0049] (“[I]f the new location is at a significant distance from any vetted location compared to the distance threshold, the new location is not considered safe.”);
- EX1037, 8:8-15 (“[A] likelihood of malicious activity may be increased when a user context property of a user account event indicates that a login location to a service is different than a profile location of the user (e.g., a user shopping account may be logged into at 9:00am from China, and then logged into at 9:30 am from London, which may indicate that malicious malware, distributed across multiple nations, has taken over the user account)”);
- EX1040, [0132] (“Behavioral indicators of potential account compromise may include ... simultaneous login from distant locations.”), [0264], [0562] (“One example of anomaly detection is referred to herein as velocity detection. For example, if a user connects to a Salesforce™ platform in California and then to a Google™ platform in Boston five minutes later, that is an indication of suspicious activity, because the user would have needed to travel at a velocity in excess of available transportation options.”).

4. Virtual Machines

50. It was well known that cloud computing systems regularly utilized virtual machines to “provide on-demand scalability, increased reliability, and decreased capital costs for users.” EX1046, 1:25-26.

- EX1044, 14/34 (“[T]he capacity in a single server is so large that it is almost impossible for most workloads to effectively use it. The best way to improve resource utilization, and at the same time simplify data center management, is through virtualization.”), 16 (describing that the benefits of virtual computers include resource optimization, consolidation, reduced cost, agility, protection from server failure, and ease of migration);
- EX1045, 126 (describing how “[v]irtual machine[s] [] make[] it very flexible and easy to manage resources in cloud computing environments, because they improve the utilization of such resources by multiplexing many virtual machines on one physical host (server consolidation)”); *see also id.*, 123-187;
- EX1046, 1:20-40 (describing how cloud computing systems include virtual machines, and how, by 2009, these systems had become “increasingly popular”);

- EX1047, 1-3 (describing how VMs provide cost-efficient resource utilization because they allow applications of different users to share the resources of a physical machine).

VI. THE '934 PATENT

A. The Claims' Earliest Effective Filing Date

51. For this declaration, I have been asked to assume that the claims' earliest possible effective filing date is October 19, 2017. That said, my opinions herein would not be materially different if the effective filing date was October 28, 2015.

B. The '934 Patent's Specification

52. The '934 patent purports to relate to “multi-factor [] authentication” (*e.g.*, EX1001, 2:15-16) and its claims are directed to this concept, *e.g.*, requiring additional verification after verifying the user's identifier and password (*e.g.*, *id.*, 18:34-41).

53. Yet the '934 patent also admits that “[m]ulti-factor authentication (MFA) [was] widely used [] as an additional verification step ... in conjunction with a traditional login and password,” (*id.*, 2:20-22) yet faults existing systems for their “over-reliance on a single method of deliver[ing]” the additional verification. *Id.*, 2:28-29.

54. The '934 patent alleges that “[w]hat is needed is a system that uses a combination of verification methods.” *Id.*, 2:35-36. It purports to solve this problem by “requiring a user to use a plurality of verification methods to ... gain access to the network resource.” *Id.*, 3:38-40; *see also id.*, 2:53-55, 3:7-9, 11:40-51. For example, after the user performs “some initial form of authentication, such as a login and password,” “the server may request that the user use a plurality of verification methods to reach the verification score needed before access is granted.” *Id.*, 11:41-47. These additional verification methods include biometric scans (e.g., fingerprint), badge scans (e.g., RFID, NFC), one-time-use codes, and passive information such as IP address and device ID. *Id.*, 10:16-11:27, 11:44-48, FIG. 4.

55. Consistent with this emphasis, two earlier patents in the same family have claims that require a *plurality* of additional verification methods. EX1062, 19:11-20:14 (claim 1) (“select a plurality of verification methods”); EX1059, 17:49-18:25 (claim 1) (“select a plurality of verification methods”).

56. The claims of the '934 patent, however, require only “an additional verification method” selected from the plurality of verification methods, as shown in Element [1.9] of claim 1, which recites:

Claim Element	Claim Language
[1.1]	A computer system configured to execute software instructions stored on nontransitory machine-readable storage media, wherein the software instructions comprise instructions that:

Claim Element	Claim Language
[1.2]	receive a request to authenticate a client, wherein the request comprises an identifier and a password,
[1.3]	store, in a multidimensional time-series database, information about the request,
[1.4]	determine whether the password corresponds to a user account identified by the identifier,
[1.5]	determine whether an additional verification is required to grant access, wherein determining whether the additional verification is required to grant access comprises:
[1.6]	retrieving, from the multidimensional time-series database, historical information about previous access requests associated with the user account, and
[1.7]	determining, based at least on the historical information, whether the user account is associated with a previous access request to a network resource, wherein the previous access request to the network resource is anomalous relative to a baseline profile of access requests; and,
[1.8]	based on the additional verification being required to grant access:
[1.9]	select an additional verification method from a plurality of verification methods,
[1.10]	cause the client to be prompted to complete the additional verification method, and
[1.11]	determine whether the additional verification method has been completed correctly.

57. Independent claim 11 is substantively identical to claim 1 but is written as a method instead of a computer system.

58. The other two independent claims (18 and 26) have many elements identical to claims 1 and 11 but recite language relating to a brute force attack instead of an anomalous access request (Element [1.7]).

VII. CLAIM CONSTRUCTION

59. For this declaration, I apply the plain and ordinary meaning of all claim terms, and do not rely on any specific constructions. I reserve the right to offer my opinions on any constructions or construction issues presented by Patent Owner QOMPLX.

VIII. GROUNDS 1-2: CLAIMS 1-30 ARE OBVIOUS OVER KIRTI (EX1004) ALONE OR COMBINED WITH COFFIN (EX1005)

60. As explained below, claims 1-30 are obvious over Kirti alone (Ground 1) or Kirti and Coffin (Ground 2), as Coffin teaches implementation details for multi-factor authentication and other additional verification processes.

61. For the independent claims, Kirti teaches all claim elements but lacks express implementation details for certain well-known features. For example, while it teaches performing additional steps for authenticating a user, Kirti does not expressly state that this involves prompting the user to complete the additional authentication or confirming correct completion of the additional authentication. These details were at least obvious based on the knowledge and skill of a POSITA (*see e.g., supra* Section V.B.2). Given the knowledge and skills of a POSITA, Kirti alone renders all challenged claims obvious under Ground 1.

62. To the extent Patent Owner argues otherwise, Coffin expressly teaches the basic implementation details for the techniques disclosed by Kirti. As explained

in greater detail below (*infra* Section VIII.C), a POSITA implementing Kirti would have been motivated with a reasonable expectation of success to incorporate Coffin’s more detailed teachings on multi-factor authentication, rendering all claims further obvious under Ground 2.

A. Kirti (EX1004)

63. Kirti relates generally to a cloud-based system in which users log in to various cloud-based applications and other services. The Kirti system analyzes user login activity over time, to detect and remediate security threats such as brute force hacking attempts. *E.g.*, EX1004, 15:49-59, Abstract, claim 1. The “remedial measures” in Kirti include “additional steps to authentication.” *Id.*, 5:27-35; *see also*, *e.g.*, *id.*, 6:6-27, 28:21-36.

64. Kirti explains how a given user “may have user accounts with various cloud applications” (*e.g.*, EX1004, 5:9-14) and the “[c]loud applications connect a user’s device to remote services” (*id.*, 1:31-32). Much like the ’934 patent would later do, Kirti depicts a system in which various clients connect with other aspects of the system via a network (“Internet”), as shown in Kirti FIG. 1:

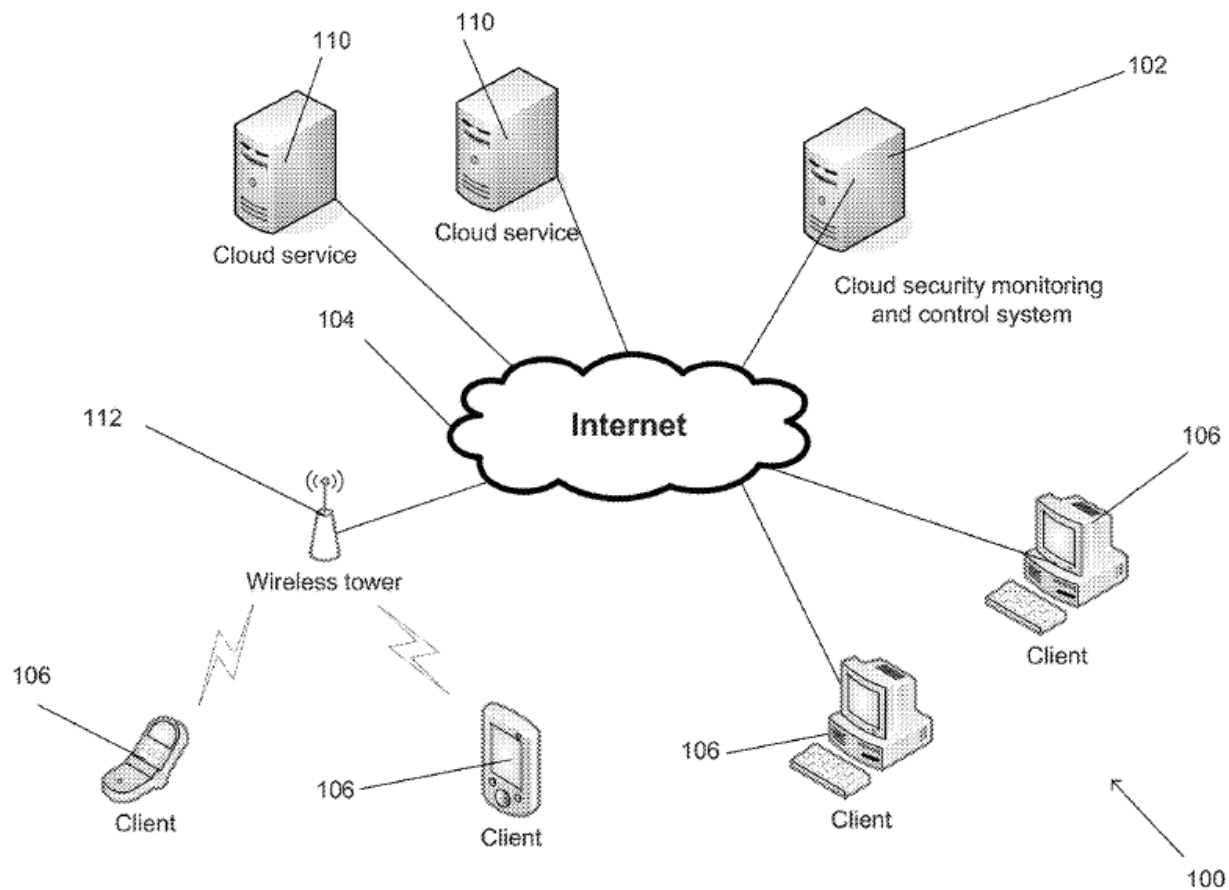


FIG. 1

EX1004, FIG. 1.

65. Based on a user's login activity over time, Kirti creates a "baseline user profile" and then identifies "anomalous activity using the baseline user profile." *Id.*, 1:47-59, 2:8-16, 29:37-46; 28:66-67; *see also id.*, 5:25-37, 10:9-59, 14:5-7, 18:14-42, 24:9-32. Login activity data includes the time of the login attempt (*e.g.*, *id.*, 12:28-57, 14:46-67, 16:53-17:8, 18:9-26); whether the login was successful, *e.g.*, correct username and password (*see e.g.*, *id.*, 10:33-36, 18:38-41); the device and/or IP address from which the attempt was made (*e.g.*, *id.*, 10:33-38, 12:5-13:2, 13:52-

59, 16:28-32, 16:53-17:19, 17:44-46, 18:14-25, 18:38-41); and the resources accessed (*e.g.*, *id.*, 10:33-44, 12:5-18, 16:53-61, 18:14-27).

66. In response to detecting anomalous activity or threats, the Kirti system may require “a higher level of security controls” (EX1004, 28:32) such as requiring “elevated authentication or OTP validation” (*id.*, 28:33). *See also id.*, 5:34 (“adding additional steps to authentication”).

B. Coffin (EX1005)

67. Coffin is a book that explains how to increase security for Oracle database applications. EX1005, 1. Coffin teaches how to implement several types of additional verification methods, beyond username/password, in an Oracle database system. These additional verification methods include: a “second password or PIN,” “a CAPTCHA,” an “answer [to] personal questions,” a “biometric scan[], like fingerprint, retina, or facial recognition,” and “pass codes sent to [] e-mail, pager, or cell phone.” EX1005, 177. Coffin also explains that this additional verification beneficially provides an added layer of security by “assur[ing] [] that the person sitting at the keyboard is who they claim to be.” *Id.*, 177.

68. Coffin provides detailed step-by-step instructions explaining how to implement several types of one-time passcodes in Oracle database applications, such as an SMS to a phone, a URL to a pager, and an email to supporting devices. *Id.*, 178-208. Coffin expressly teaches details for selecting one or more of the plurality

of verification methods (*e.g.*, *id.*, 183, 185, 200, 201), prompting the user to complete the additional verification method (*e.g.*, *id.*, 183, 190, 193, 194, 201, 207), and determining whether the user correctly completed the additional verification method (*e.g.*, *id.*, 194, 201, 206). For example, “[b]y preference, [the system] will send the two-factor code to the user’s pager and cell phone. If neither of those is available, [the system] will send the code to the user’s e-mail.”). *Id.*, 200. “Once the user receives the two-factor pass code” (*id.*, 183), “[they] will have to enter those 2-factor codes in order to get access” (*id.*, 207). The computer system then “test[s] the two-factor code to see if it passes muster.” *Id.*, 194.

C. Combining Kirti And Coffin

69. The Kirti+Coffin combination combines Kirti’s threat detection and remediation system with Coffin’s implementation details on how to select and perform an additional verification process, such as 2-factor authentication. As noted above, Kirti teaches provides less implementation detail, however, on selecting and performing its additional steps for authentication, and does not expressly disclose some of the more basic steps recited in Elements [1.8]-[1.11] of the challenged claims. Yet POSITAs knew that implementing an additional authentication requires these common steps: “select[ing] an additional verification method from a plurality of verification methods, caus[ing] the client to be prompted to complete the additional verification method, and determin[ing] whether the additional verification

method has been completed correctly” (EX1001, 18:36-41 (claim 1)). *Supra* Section V.B.2.

70. To the extent a POSITA implementing Kirti (an Oracle patent) did not already know these details, she would have naturally looked to Coffin (a book on improving security in Oracle systems) for its detailed teachings on well-known types of additional verification methods and step-by-step instructions for how to implement these additional verification methods.

71. A POSITA would have been motivated with a reasonable expectation of success to incorporate Coffin’s additional verification teachings in Kirti’s system for several reasons:

1. Kirti teaches requiring and performing well-known techniques for additional verification but provides few implementation details;
2. Coffin provides details on how to implement these additional verification techniques, with step-by-step instructions;
3. Kirti and Coffin both offer the same motivation for including this additional verification (added security), as was well known in the art (*supra* Section V.B.2); *see also* EX1005, 177 (Two-factor authentication “assure[s] [] that the person sitting at the keyboard is who they claim to be.”);

4. Coffin’s additional verification teachings would have been easily implemented in Kirti’s system—requiring only minimal software modifications that were well within the skill of a POSITA; and

5. Both Kirti and Coffin relate to Oracle cloud applications, making them a natural pairing (Kirti is an Oracle patent and Coffin a textbook on how to program secure Oracle database applications).

72. Kirti already teaches requiring additional verification. *E.g.*, EX1004, 28:33 (“elevated authentication or OTP validation”) and 5:34 (“adding additional steps to authentication”). Coffin provides step-by-step instructions explaining how to implement at least some of these additional verification methods. *E.g.*, EX1005, 177-78 & 183; *id.*, 207 (“[W]e developed a process to send 2-factor authentication codes to our application users on their cell phones, pagers and e-mail accounts.”); *see generally id.*, 177-208 (Chapter 9). Moreover, implementing these additional verification teachings in Kirti’s system would have required simple, predictable, and straightforward software modifications. *E.g.*, EX1005, xxv (explaining Coffin purpose to “provide you tools and knowledge that you can put to immediate use. ... [and] get you thinking about how you can write bulletproof applications of your own”), xxvi (explaining how book provides “a template that we can provide to other application programmers so that they can implement the same security structures”). Chapter 9 of Coffin gives detailed instructions on how to implement two-factor

authentication (“2FA”) via a variety of techniques: the program can send an OTP to a pager, a cell phone (via SMS), or via email. EX1005, 195. Coffin explains that Oracle’s products at that time had support for sending an OTP via any of these methods so that a programmer could easily add 2FA to their code with minimal time and effort: Oracle’s products supported SMTP for example, which enables the sending of email as well as SMS if the cell carrier allows SMTP-to-SMS translation. By 2017, there were a host of other software packages and libraries that made adding 2FA to a software package straightforward: for example, Passport.js for Javascript, PyOTP for Python, and Otp.NET for Microsoft’s .NET framework were all available before 2017.

73. Further, Kirti is a patent assigned to Oracle and Coffin provides teachings on how to improve security in Oracle systems. Indeed, Kirti includes what appears to be a screen shot from Palerra’s LORIC product (EX1004, FIG. 5D), which Oracle integrated into its CASB Cloud Service after acquiring Palerra in September 2016 (EX1053, 3 n.1; EX1050; EX1051; EX1052; EX1054; EX1055). This further establishes both motivation and reasonable expectation of success in applying Coffin’s Oracle-centered teachings to the threat detection systems of Kirti. Moreover, Coffin provides “a foundation for secure programming in Java and Oracle, and their common ground, Java Stored Procedures (JSP).” EX1005, xxv; *see also id.*, 1 (“[M]uch of what we will do here can be acquired through commercial

products from Oracle corporation and elsewhere.”), 2 (explaining that Coffin provides a foundation for achieving “application security using the basic Oracle Database and Java services”), 27 (noting that Oracle provides the Java Development Kit), 193 (describing Oracle “Java stored procedure” used “to send two-factor codes to a user”).

74. The narrative claim mappings below provide additional explanation, as pertinent, on why and how a POSITA would have incorporated Coffin’s teachings into a system like that taught by Kirti.

1. Claim 1

a) Element [1.1]¹

[1.1] A computer system configured to execute software instructions stored on nontransitory machine-readable storage media, wherein the software instructions comprise instructions that:

75. If the preamble is limiting, Kirti satisfies it. Kirti teaches a system, such as “system 100 including a cloud security monitoring and control system 102, client devices 106 ... and cloud services 110.” *E.g.*, EX1004, 4:14-19, FIG. 1. The system includes “software modules” and a processor, and Kirti teaches storing the software

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

modules “in volatile or non-volatile memory and,” that these modules “configure the processor 201 to perform certain functions or processes.” *Id.*, 5:38-44. The software modules include Kirti’s threat detection and prediction analytics and other applications. *Id.*, 5:44-6:35. *See also id.*, 2:37-51 (discussing memory containing applications), 29:66-39: (claim 15), which recites *inter alia* “memory including one or more instructions that, when executed by the processor, cause the processor to” carry out various steps such as determine anomalous activity using a user profile and send instructions to change security controls), 4:5-13 (describing system components including “software applications and/or modules that configure a server or other computing device to perform processes”), 4:29-45. These software modules include instructions that cause the system to carry out the steps recited in the body of the claim, as explained in the following sections.

b) Elements [1.2], [1.4]

[1.2] receive a request to authenticate a client, wherein the request comprises an identifier and a password,

[1.4] determine whether the password corresponds to a user account identified by the identifier,

76. Kirti teaches these claim elements at least because it teaches tracking whether requests to authenticate (“login attempts”) are successful and that “login attempts” include an identifier (e.g., “identifier,” “username”) and “password.” *E.g.*, EX1004, 10:9-46, 22:3-10. A POSITA would have understood or at least found it

obvious that, as part of tracking whether the requests are successful, the Kirti system receives each such request and determine whether the password is correct (i.e., corresponds to the user account identified by the identifier).

77. Kirti discloses users logging in to cloud applications from various “client devices,” as shown in FIG. 1:

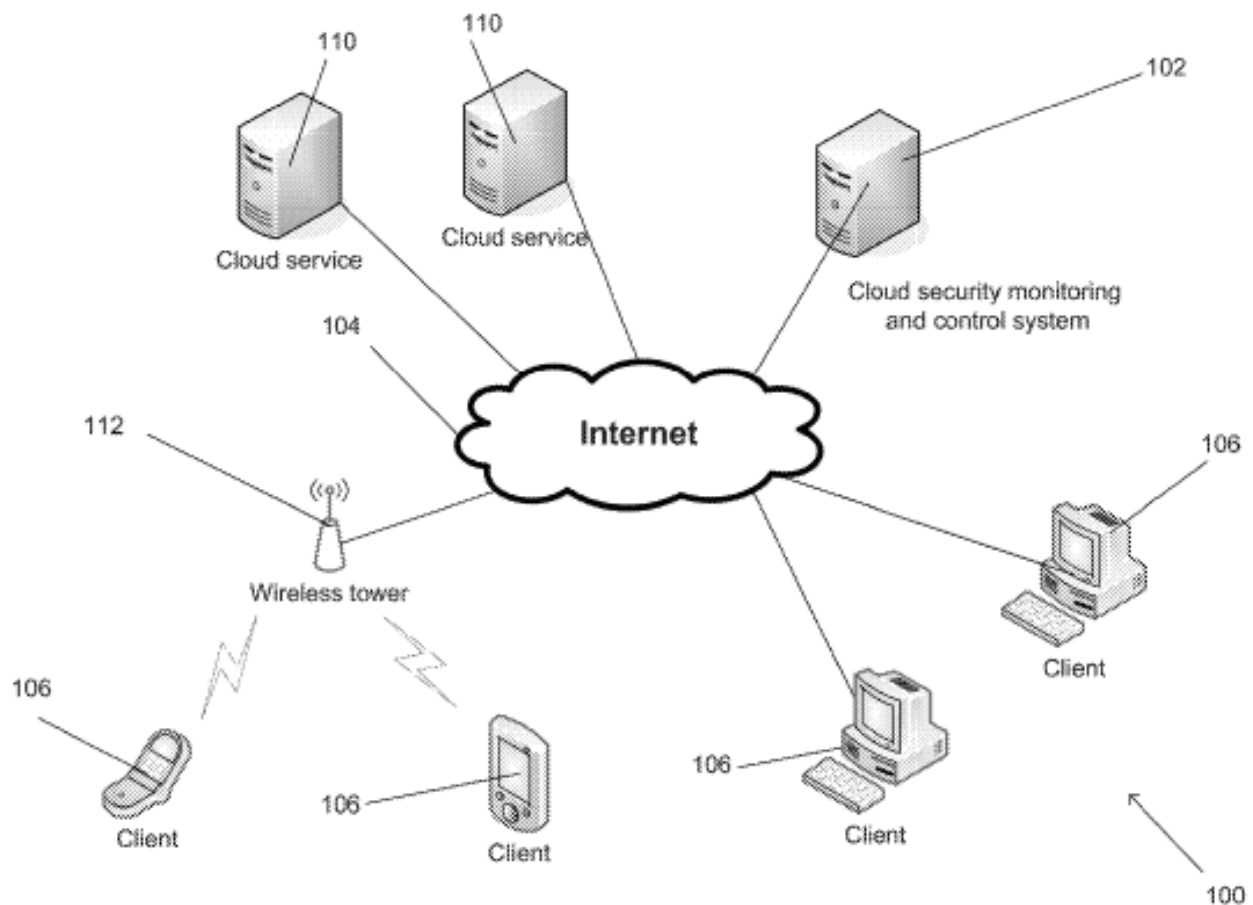


FIG. 1

EX1004, FIG. 1; *id.*, 4:15-20 (“client devices 106 that can be used to access the cloud security system 102”), 1:31-34 (“Cloud applications connect a user’s device to remote services that provide an additional functionality.”), 9:38 (“user login[]

attempts”), 15:36-37, 16:31-32 (“how a user does or is expected to log in or access a cloud application”), 17:4-5, 18:40-41, 28:4 (“user logins”).

78. Kirti FIG. 5A shows an example of “cloud application login” data, including time and location of login, that is used to detect anomalous login attempts:

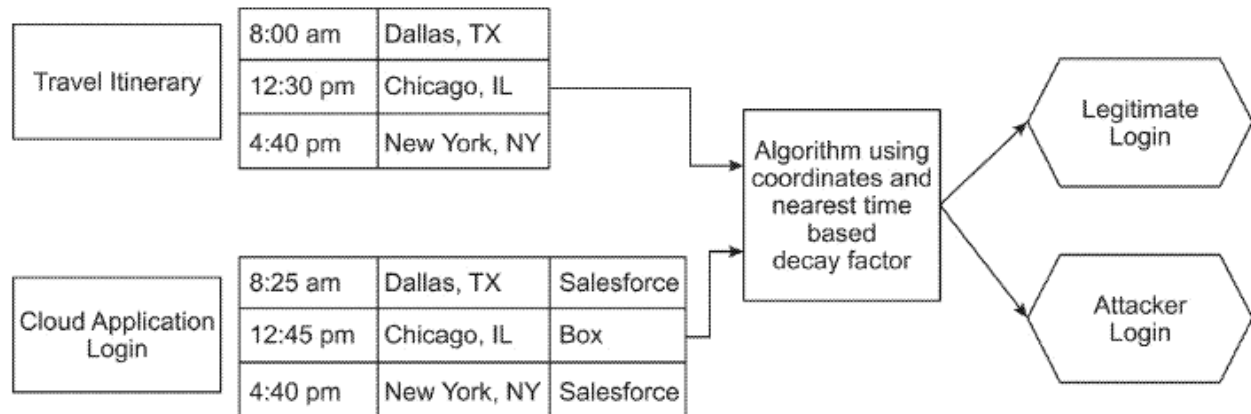


FIG. 5A

EX1004, FIG. 5A.

79. Kirti also describes “processes for threat detection” based on “login events from multiple different geolocations during a short time window”). *Id.*, 17:56-18:7.

80. Kirti discloses login attempts that include a username or other identifier. *E.g.*, EX1004, 15:55-59 (describing “login attempts with invalid or terminal/suspended usernames”), 13:47-51 (“A single user can be identified across multiple clouds using one or more attributes or identification factors, such as a primary user identifier (ID).”), 22:3-12 (tracking distance between “login attempts,

successful logins, and/or failed logins,” accounting for a user’s “different usernames” for “different cloud applications”), 18:17-23 (“events such as ... logging in to the cloud application . . . [include] event details such as user name”); *see also id.*, 10:30-46 (“types of activity data can include ... login and logout statistics [and] can include the user account or other user identifier for the user associated with the events or statistics.”), 7:30-67 (Tables 1-2, “Credentials and Identifiers”), 13:62-67 (describing “the user identifier associated with” a user’s different cloud application accounts and providing example identifiers “jdoe, john.doe, etc.”).

81. Kirti also teaches that its login attempts include a password and discloses that the system determines whether a given login attempts succeeds or fails. *Id.*, 10:35-36 (tracking “login and logout statistics (including attempts and successes)”; 2:23-25 (tracking failures); *see also id.*, 12:9, 12:19-20, 12:65-13:17, 13:52-59, 15:4-7, 15:14-16, 15:47-59, 18:38-41, 19:25-59 (Table 4), 21:19-52, 22:8, 29:53-55, FIGS. 8C (“Failed logins”), 8D (“failed logins”).

82. For example, Kirti discloses detecting a threat if a user has several failed login attempts because such activity “indicate[s] a concerted effort to crack the user’s password.” *Id.*, 13:14-16; *see also id.*, 13:56-57 (identifying “users who have several failed login attempts and then change their password”), 15:49-59 (“A brute force attack scenario may refer to an attacker's attempts to try many passwords in order to discover a correct password and compromise a user account.”). Kirti also

discloses changing and/or resetting the user's password (*id.*, 5:35, 12:13, 13:57, 24:45, 25:34-40, 25:48) and password policies (*id.*, 7:30-67 (Tables 1-2), 14:40-45, 19:25-59 (Table 4), 26:33-38, 27:23-26, 27:31-35, 27:39-42, FIG. 8E (illustrating different password options for a given user account)). Kirti (EX1004) FIG. 8E shows a user interface with "Password Controls" that include example "password requirements for a user account." EX1004, FIG. 8E, 26:29-38; *id.*, 6:32-38.

Manage app instances Register an app instance Modify app instances Remove app instances	Register an app instance																						
	1. Select an app type ✓	3. Select security controls ⓘ																					
	2. Select monitoring type ✓	Standard Stringent Custom																					
	3. Select security controls →	▼ Password Controls <table border="1"> <thead> <tr> <th>Security Control</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Minimum required characters</td> <td>10</td> </tr> <tr> <td>Require numbers(s)</td> <td>2</td> </tr> <tr> <td>Require special character(s)</td> <td>1</td> </tr> <tr> <td>Require at least one uppercase letter</td> <td>☐</td> </tr> <tr> <td>Prevent common words / email address as a password</td> <td>☐</td> </tr> <tr> <td>Password resets: Require users to reset password every:</td> <td>30 days</td> </tr> <tr> <td>Prevent reusing passwords from</td> <td>Last 10 times</td> </tr> <tr> <td>Notify admins when users request a forget password email</td> <td>☐</td> </tr> <tr> <td>Notify admins when users change passwords in Settings</td> <td>☐</td> </tr> <tr> <td>Require strong passwords for external collaborators</td> <td>☐</td> </tr> </tbody> </table> ▶ Authentication Policies ▶ Session Policies ▶ Settings	Security Control	Value	Minimum required characters	10	Require numbers(s)	2	Require special character(s)	1	Require at least one uppercase letter	☐	Prevent common words / email address as a password	☐	Password resets: Require users to reset password every:	30 days	Prevent reusing passwords from	Last 10 times	Notify admins when users request a forget password email	☐	Notify admins when users change passwords in Settings	☐	Require strong passwords for external collaborators
Security Control	Value																						
Minimum required characters	10																						
Require numbers(s)	2																						
Require special character(s)	1																						
Require at least one uppercase letter	☐																						
Prevent common words / email address as a password	☐																						
Password resets: Require users to reset password every:	30 days																						
Prevent reusing passwords from	Last 10 times																						
Notify admins when users request a forget password email	☐																						
Notify admins when users change passwords in Settings	☐																						
Require strong passwords for external collaborators	☐																						
4. Enter credentials	box test																						
Approval ☐ I understand and explicitly approve the seedings of controls in to "Box - test" with controls setting as "Stringent" < Previous Enter credentials > Cancel																							

FIG. 8E

EX1004, FIG. 8E.

83. POSITAs understood that the success or failure of a given login attempt in Kirti resulted from the system determining whether the received password

corresponded to a user account identified by the received username or other identifier, consistent with known login conventions. *Supra* Section V.B.1. The '934 patent confirms that this was conventional by noting “traditional login and password” in its “State of the Art” section without providing any detail on how to implement this “traditional” functionality. EX1001, 2:18-23.

84. A POSITA also would have understood that a user enters a username and password when connecting to the cloud application(s) at least because this is what an administrator does when connecting to the cloud. EX1004, 5:20-24, 6:60-65, 10:9-15, 26:49-54.

85. Thus, Kirti teaches that a client (“user”) submits a request to authenticate (“login attempt[]”), which includes an identifier (e.g., “identifier,” “username”) and “password,” and that the system receives this request and determines whether the password corresponds to a user account identified by the identifier (i.e., determines, based on the received username and password, whether the login succeeds or fails). Further supporting that it was obvious to implement Kirti’s system to receive and check the user’s username and password, this was a well-known, standard, and conventional technique for authenticating a user. *Supra* Section V.B.1. Given conventional login processes, the most natural way to implement Kirti was such that the system received the user login request and

determined whether the provided username or other identifier corresponded to the password.

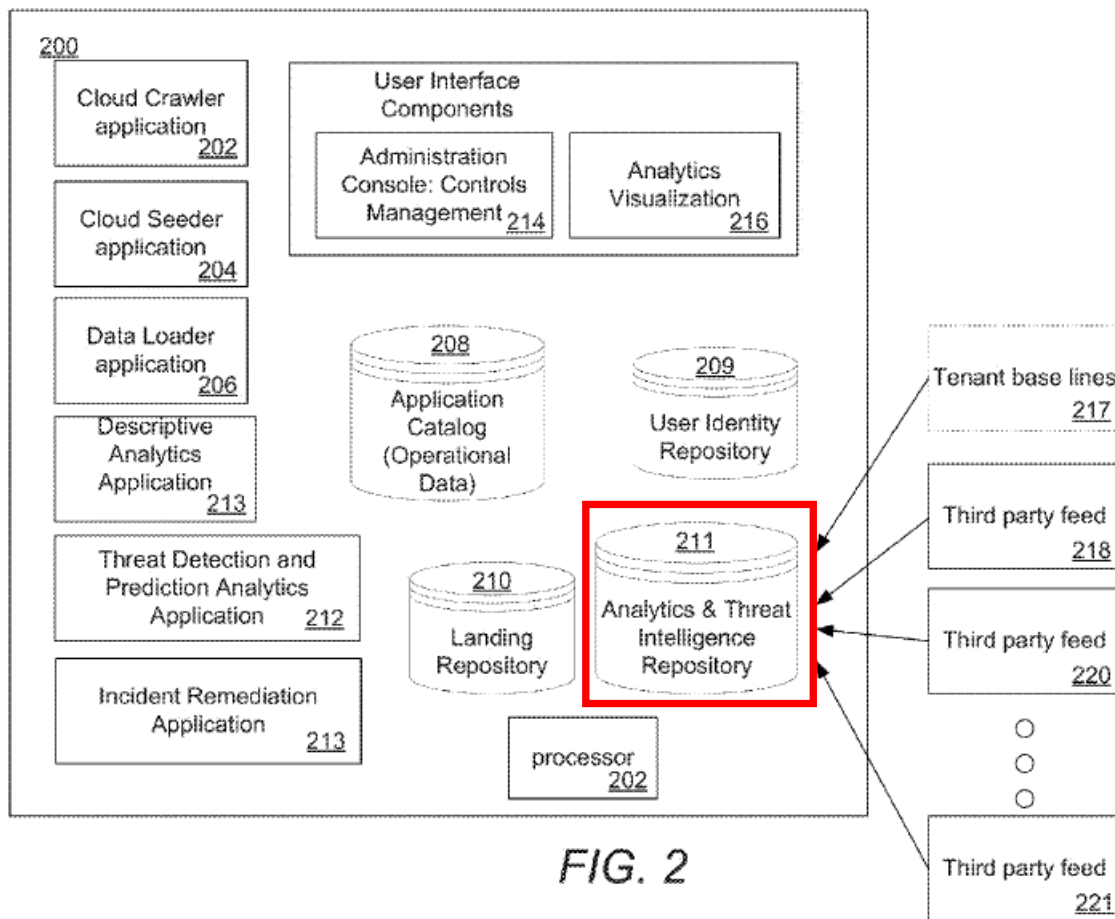
c) **Element [1.3]**

[1.3] store, in a multidimensional time-series database, information about the request,

86. Kirti teaches this claim element at least because it discloses that its system stores, in a multidimensional time-series database (e.g., “analytics and threat intelligence repository database”), information about requests for authentication (e.g., “activity data,” for “login and logout statistics (including attempts and successes)”). *E.g.*, EX1004, 10:33-36, 10:60-61. The information includes timestamp (*e.g.*, EX1004, 12:35-54, 14:48-54, 16:49-60, 18:9-23), whether the request was successful (*e.g.*, *id.*, 10:34-44, 18:38-41), the device and/or IP address of the request (*e.g.*, *id.*, 10:34-44, 12:1-18, 12:35-13:29, 13:44-59, 16:28-40, 16:55-67, 17:9-55, 18:17-23, 18:38-41), and what resources were accessed (*e.g.*, *id.*, 10:34-44, 12:7-18, 16:49-60, 18:17-23). *Id.*, 10:35-39 (“activity data can include ... login and logout statistics (including attempts and successes), IP addresses used to access the application, devices used to access the application, and cloud resources that were accessed.”).

87. Kirti’s analytics and threat intelligence repository database is a multidimensional time-series database. It is multidimensional because each time

point includes activity data for multiple parameters / attributes (e.g., IP address, login attempt result (success or failure), identifier (e.g., username), resources accessed, etc.). EX1004, 4:46-5:3, 10:25-50, 10:60-11:4, 12:1-18, 12:35-13:29, 13:44-59, 14:48-54, 16:28-40, 16:55-67, 17:9-55, 18:9-23, 18:38-41). And it is a time-series database at least because it includes the same activity data (i.e., values for the same parameters / attributes) for multiple time points. *E.g.*, EX1004, 13:27-30 (“[D]ata collected over time is used to build models of normal behavior (e.g., patterns of events and activity) and flag behavior that deviates from normal as abnormal behavior.”).



EX1004, FIG. 2 (red annotations added).

88. Consistent with its time-series nature, Kirti's database stores weeks' worth of activity data that "can be utilized to generate reports that may be presented visually." EX1004, 4:54-59; *see also id.*, 6:35-44, 11:15-58, 12:5-35, 12:50-57, 14:46-15:16, FIGS. 8B-8D. For example, FIG. 8C "displays a count of events by date in each of the color coded categories such as activities at an unusual time, after-hours downloads, failed logins, etc.," for prior four- and twelve-week periods, thus

confirming that Kirti's multidimensional time-series database stores timestamped data for multiple attributes/parameters over a period of time. *Id.*, 15:5-7.

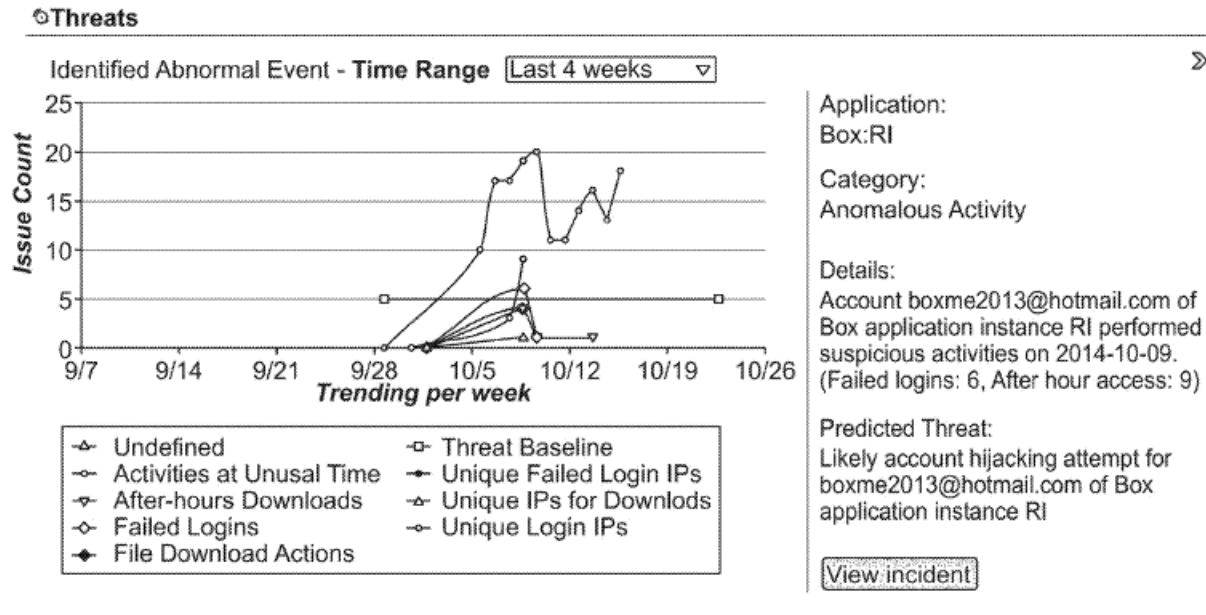


FIG. 8C

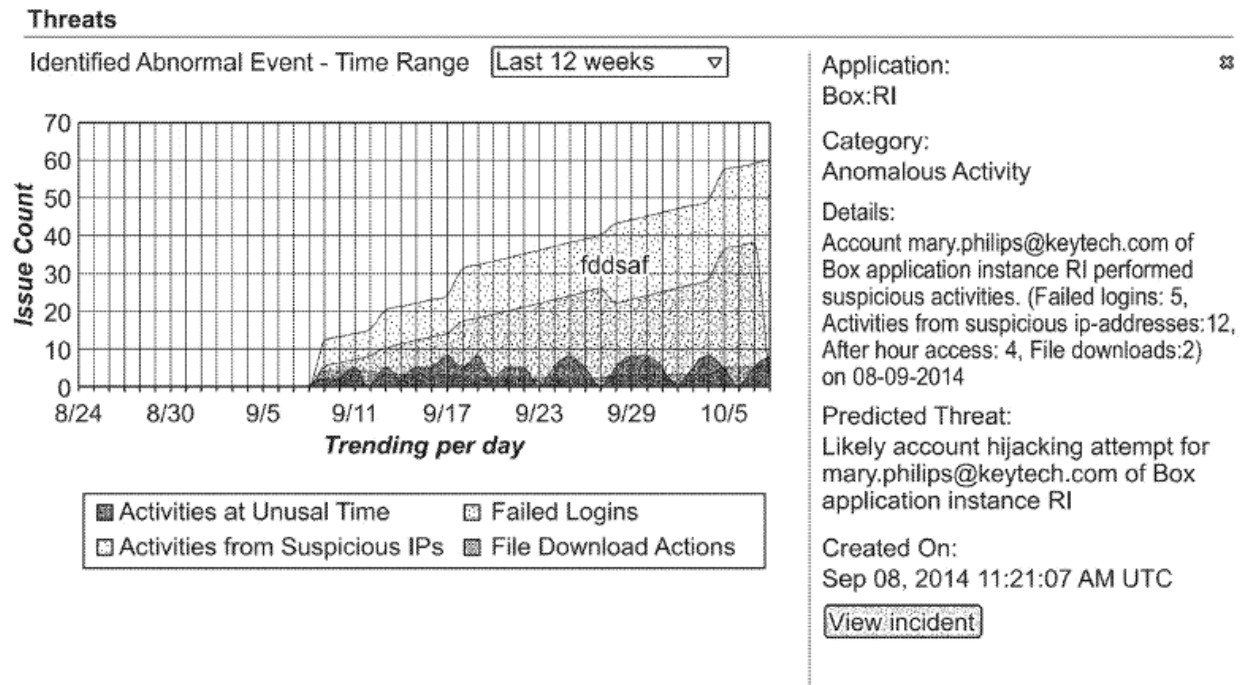


FIG. 8C (Cont.)

Id., FIG. 8C.

d) **Element [1.5]**

[1.5] determine whether an additional verification is required to grant access,

89. Kirti teaches this claim element. The Kirti system analyzes a user's historic login activity to identify potential threats and determine whether to apply "remedial measures, such as **adding additional steps to authentication.**" *E.g.*, EX1004, 5:25-37 (emphasis added); *see also id.*, 3:53-58 (system performs "threat detection" and recommends "appropriate responses to different categories of threat"), 3:58-4:3 (system determines "models of normal and/or abnormal behavior in user activity," detects "patterns of suspicious activity," and then recommends

“remedial measures”). Kirti teaches carrying out this remediation “automatically.” 25:49-52.

90. Kirti explains that “[a]ny of a variety of security measures may be taken to address an identified threat such as, but not limited to ... setting stronger security controls.” *Id.*, 25:46-49 (within section titled “Remediation”); *see also id.*, 25:16-20 (describing alert with recommended “remediation action(s), such as implementing stronger security controls”). Example stronger security controls include requiring “elevated authentication or OTP validation.” *Id.*, 28:31-33. POSITAs knew that “OTP validation” in Kirti referred to “one-time passwords,” a commonly-used secondary authentication technique. *E.g.*, EX1020, 2 (2014 thesis, stating “the most commonly used second factor is one-time passwords”); *see also* EX1001, 2:23-25 (“MFA methods commonly used today includes one-time use codes sent to a user’s mobile device or email.”); EX1014, 4:17-18 (“some users may be required to engage in a token-based one-time password (OTP) process”); EX1016, 1:58-59 (“Another authentication approach is for a service provider to equip the user with a One Time Password (OTP) device.”); EX1018, 3:3-4 (“One solution to avoid attacks with replaying captured reusable passwords is to use one-time passwords (OTP).”).

91. POSITAs would have understood that Kirti’s “additional steps to authentication” and other “stronger security controls” are additional verification beyond the traditional username/identifier and password authentication used by Kirti

client devices. Kirti describes logins as involving username and password (*supra* Section VIII.C.1.b) and POSITAs would have understood that the “stronger security controls” are in addition to that baseline username/password requirement.

92. Thus, Kirti’s recommended “additional steps to authentication”—such as elevated authentication or OTP validation—satisfy the claimed “additional verification [] required to grant access.” The ’934 patent itself characterizes types of additional authentication as additional verification: “Multi-factor authentication (MFA) is widely used today as an additional verification step often used in conjunction with a traditional login and password” EX1001, 2:20-23.

93. Thus, Kirti determines that additional verification is required to grant a client access to a cloud application or other network resource. That Kirti makes this determination is further shown by the fact that Kirti discloses performing the additional steps to authentication (additional verification). *Infra* Section VIII.C.1.f) (mapping Element [1.8]-[1.11]).

Coffin’s teachings

94. To the extent Kirti’s remedial measures alone do not satisfy the claimed “additional verification,” a POSITA would have found it obvious to implement Kirti’s system with the specific types of measures taught by Coffin, thus satisfying these claim elements.

95. Coffin expressly discloses several types of additional verification for an Oracle system, including: a “second password or PIN,” “a CAPTCHA,” an “answer [to] personal questions,” a “biometric scan[], like fingerprint, retina, or facial recognition,” and “pass codes sent to [] e-mail, pager, or cell phone.” EX1005, 177; *see also id.*, 178-208 (explaining how to implement some of these additional verification methods). Coffin also explains the benefits of this additional verification: it “assure[s] us that the person sitting at the keyboard is who they claim to be.” *Id.*, 177.

96. Thus, to the extent Kirti does not render obvious implementing its system to require this additional authentication, a POSITA would have been motivated to do so in view of Coffin, which teaches use of this authentication to provide an added layer of security. *Supra* Section VIII.C. A POSITA would have reasonably expected to succeed in implementing Coffin’s additional verification teachings in Kirti’s disclosed system for at least three reasons: (1) Kirti already discloses performing remedial actions and expressly mentions additional verification steps including one-time passwords (OTPs) (*e.g.*, EX1004, 3:53-4:1, 5:27-37, 5:55-6:34, 14:48-54, 14:61-67, 24:43-47, 24:66-67, 25:16-20, 25:34-52, 25:61-26:5, 28:22-36, FIGS. 5B, 6), (2) Coffin expressly teaches a POSITA how to implement at least some of the additional verification methods (EX1005, 178-208), and (3) implementing this additional verification functionality in Kirti’s disclosed

system requires only minor software changes that were well within the skill of a POSITA. *See supra* Section V.B.2 (explaining that additional authentication was well known in the art).

e) **Elements [1.6]-[1.7]**

[1.6] wherein determining whether the additional verification is required to grant access comprises:

retrieving, from the multidimensional time-series database, historical information about previous access requests associated with the user account, and

[1.7] determining, based at least on the historical information, whether the user account is associated with a previous access request to a network resource, wherein the previous access request to the network resource is anomalous relative to a baseline profile of access requests; and,

97. Kirti teaches these claim elements. For example, Kirti teaches determining whether to require additional authentication steps using “analytics and security intelligence processes.” *E.g.*, EX1004, 24:67-25:5 (“process for remediating threats” “includes identifying (602) a threat”); *see also id.*, FIG. 6. These processes involve retrieving historic user login data from Kirti’s analytics and threat repository database and using that information to detect “anomalous behavior” relative to “baseline user profiles over various periods of time.” *Id.*, 24:33-50; *see also id.*, FIG. 5B (exemplary process for “detecting threats”), 3:53-4:1 (“detecting patterns of suspicious activity” and “recommending remedial measures”), 14:5-14 (system

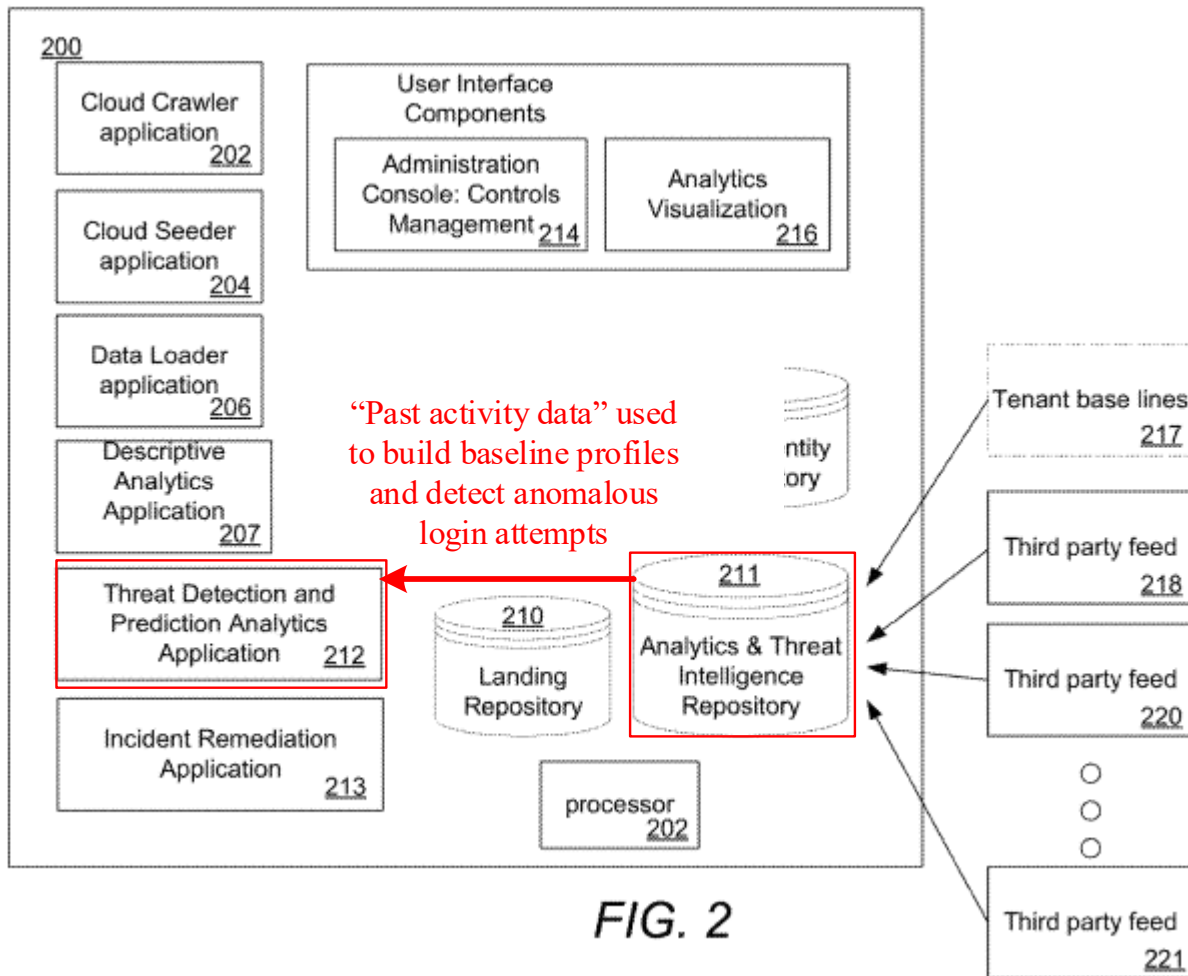
“tracks user activity for anomalous behavior to detect attacks and unknown threats” and provides “recommendations for remediation”); *id.*, 28:22-36.

98. As explained above for element [1.3] (*supra* Section VIII.C.1.c), the activity data stored in Kirti’s analytics and threat intelligence repository database (a multidimensional, time-series database) is historical at least because the database stores data collected over a relatively long time period (e.g., several weeks) and receives new data in batches (e.g., from the most recent 24-hour period). *E.g.*, EX1004, 4:54-59, 6:35-44, 11:15-58, 12:5-35, 12:50-57, 13:27-30, 14:46-15:16, FIGS. 8B-8D. Thus, even the most recent data in the database is historical.

99. And, as also explained above for element [1.3] (*supra* Section VIII.C.1.c), the historical activity data stored in Kirti’s analytics and threat intelligence repository database includes information about previous access requests, such as “login and logout statistics (including attempts and successes)” (EX1004, 10:35-40), a timestamp (*e.g.*, EX1004, 12:35-54, 14:48-54, 16:49-60, 18:9-23), whether the request was successful (*e.g.*, *id.*, 10:34-44, 18:38-41), the device and/or IP address of the request (*e.g.*, *id.*, 10:34-44, 12:1-18, 12:35-13:29, 13:44-59, 16:28-40, 16:55-67, 17:9-55, 18:17-23, 18:38-41), and what resources were accessed (*e.g.*, *id.*, 10:34-44, 12:7-18, 16:49-60, 18:17-23). *E.g.*, EX1004, FIG. 4 (step 406).

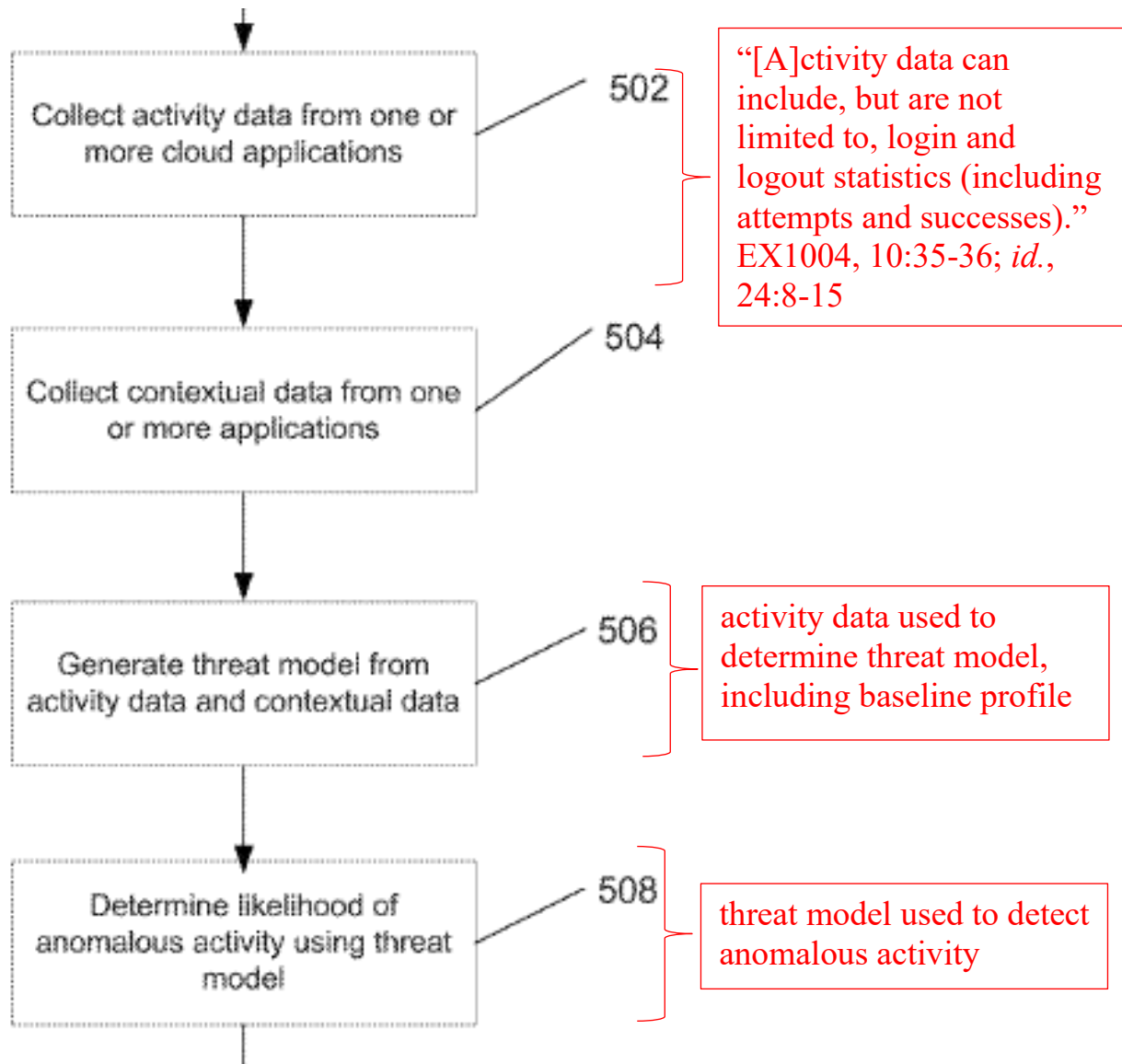
100. Kirti teaches its “Threat Detection and Prediction Analytics Application 212,” retrieves this past activity data from Kirti’s “Analytics & Threat Intelligence Repository 211” to detect threats. *E.g.*, EX1004, 18:10-14. Indeed, Kirti describes how its analytics tool (e.g., threat detection and prediction analytics application 212) builds a baseline user profile based on data collected over time and stored in the analytics and threat intelligence repository, and uses this profile to detect anomalous login attempts (e.g., in the most recent batch of data). For example, Kirti describes how “data collected over time is used to build models of normal behavior (e.g., patterns of events and activity) and flag behavior that deviates from normal as abnormal behavior” (EX1004, 13:27-30), including behaviors related to “failed logins” (*id.*, 13:10-17). *See also id.*, 1:47-59 (“determining the likelihood of anomalous activity using the baseline user profile”), 1:66-2:18 (“the baseline profile is derived from activity data collected over a time period”), 2:37-51, 4:46-5:3 (“The aggregation of activity information in the analytics repository 211 concerning access patterns and other event statistics enables the system to establish baselines of user behavior.”), 5:46-54 (“The threat detection and prediction analytics application 212 can generate analytics Analytics may be performed using data stored in the analytics and threat intelligence repository 211.”), 14:5-14 (“a recommendation engine tracks user activity for anomalous behavior to detect attacks and unknown threats” and provides “recommendations for remediation”); 14:27-31

(“[T]echniques such as outlier detection establish a baseline that is useful for detecting anomalous activities. Such anomalous activities along with contextual threat intelligence can provide more accurate prediction of threats with low prediction errors.”), 15:24-16:18 (describing detecting specific types of threats, including brute force attacks and describing how “[d]etection processes may track a user’s normal behavior and generate alerts when events or activities associated with the user’s account(s) deviate from the norm”), 19:61-24:47 (describing various “behavior analytics algorithms” for detecting “anomalous activity” in the user’s activity data to “determine the likelihood of various threats”), 29:23-27.



EX1004, FIG. 2 (red annotations added).

101. Kirti FIG. 5B shows an exemplary process in which activity data is retrieved and used to generate threat models (step 506) “that include baseline user profiles over various periods of time.” EX1004, 24:33-38.



Kirti, FIG. 5B (cropped and annotated in red); *see also id.*, FIG. 4 (showing process with steps 406, “[s]tore activity data in analytics database,” and 410, “[g]enerate system reports and/or threat intelligence”); *id.*, 10:60-61 (“retrieved activity data is stored (406) in an analytics and threat intelligence repository database 211.”); *id.*, 10:35-36 (“activity data can include, but are not limited to, login and logout statistics (including attempts and successes).”).

102. These threat models are then “used to determine (508) the likelihood of anomalous activity that may warrant additional verification.” *Id.*, 24:39-42.

103. As shown in FIG. 5, Kirti teaches that its analyzer (e.g., threat detection and prediction analytics application 212) retrieves historic activity data from the Analytics and Threat Intelligence Repository 211, to “[d]iscover threats and risks.”

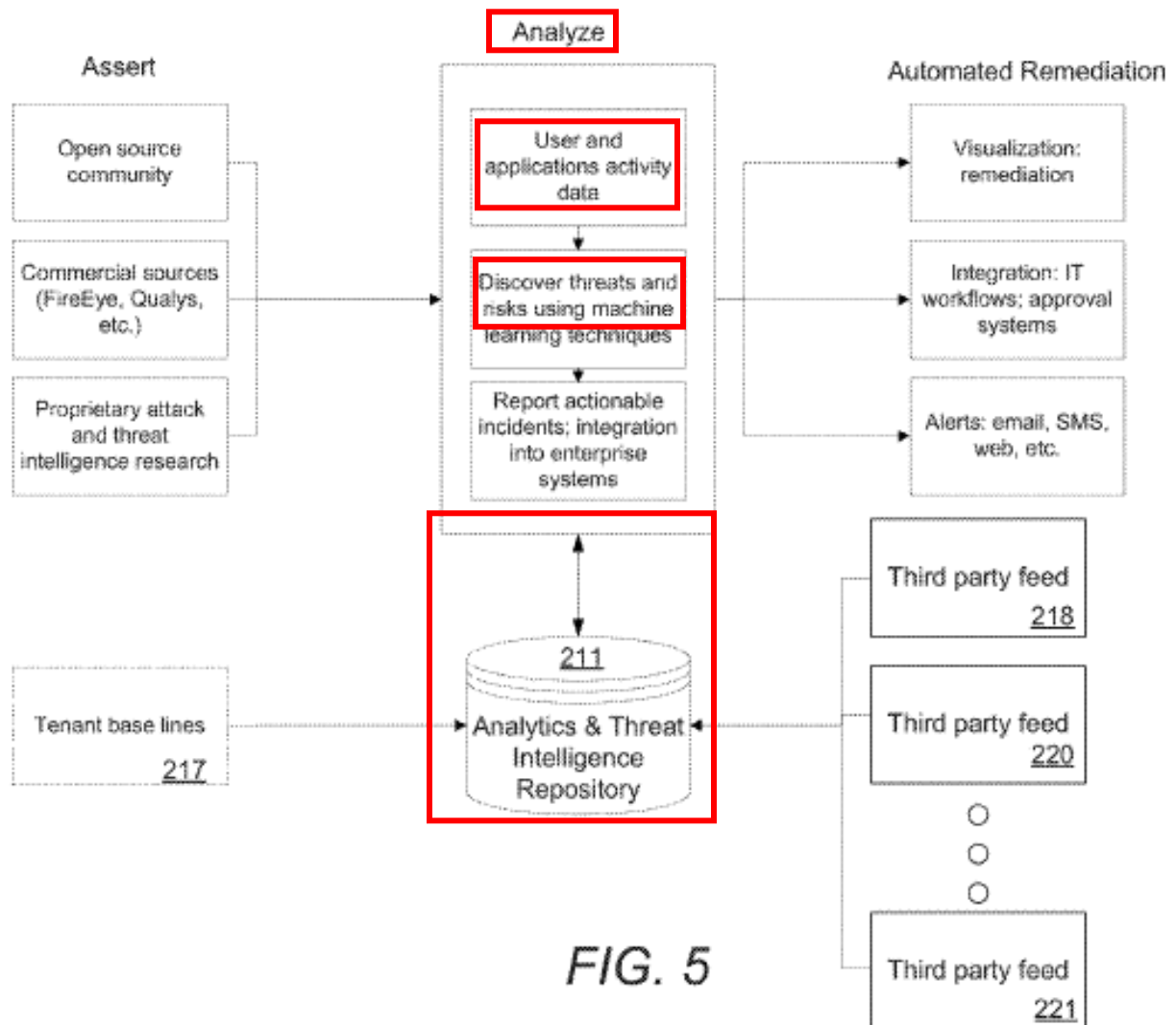


FIG. 5

Kirti, FIG. 5 (red annotations added); *see also id.*, Abstract, 1:44-59, 2:37-51, 3:53-4:3, 4:60-66 (“The aggregation of activity information in the analytics repository

211 concerning access patterns and other event statistics enables the system to establish baselines of user behavior.

104. Machine learning techniques can then be applied to detect threats and provide recommendations concerning how to respond to threats.”), 5:25-37, 5:53-54, 5:55-6:5, 12:1-57.

105. The Kirti system thus satisfies the claimed **“retrieving ... historical information about previous access requests”** at least by its analytics and threat detection processes that retrieve a user’s historic login activity over time and then analyze that historical information to determine a baseline profile. *E.g.*, EX1004, 1:47-59, 2:37-51, 4:46-5:3, 5:46-54 (“The threat detection and prediction analytics application **212** can generate analytics Analytics may be performed using data stored in the analytics and threat intelligence repository **211**.”), 29:23-27. And the Kirti system satisfies the claimed **“determining, based at least on the historical information whether the user account is associated with a previous [anomalous] access request ...”** at least by using the retrieved historic login data to generate baseline user-behavior models and then determining anomalous activity, such as anomalous login attempts in the most recent batch of past activity data. These anomalous login attempts (access requests) are “previous” because they are from the past activity data that is collected in batches (*e.g.*, each batch includes activity data from the most recent collection period, such as the most recent 24-hour period). *E.g.*,

EX1004, 10:25-42. Even if Kirti's system identified a current login attempt as anomalous, that anomalous login attempt would still be a "previous" login attempt by the time of the next login attempt where the system would require additional verification.

106. Kirti teaches different types of anomalous access requests, including requests that originate from a new IP address not associated with the user's baseline profile and/or from an IP address that has been flagged as suspicious. EX1004, 1:66-2:7 ("the baseline profile includes a list of IP addresses ... [and] a threat is recognized when activity occurs from a geolocation that is not on the list of IP addresses"), 5:57-6:5, 12:28-57, 15:31-48, 16:6-11, 16:60-17:8, 28:26-31, 29:31-36, FIG. 8C.

107. Another anomalous access request is a brute force attack, where a hacker uses different passwords to try and brute force crack the user's password. EX1004, 15:49-59.

108. Further, Kirti, discloses an "advanced persistent threat (APT) scenario," in which, for example, a user does "not access[] a particular cloud application in several months and then show[s] high activity in the next month." *Id.*, 13:21-27.

109. Moreover, Kirti at least suggests other types of anomalous access requests, such as impossible travel (*id.*, 1:63-65, 13:52-56, 15:31-43, 17:13-19, 19:12-15, FIG. 5C), that were well known in the art. *Supra* Section V.B.3.

110. After detecting previous anomalous access requests, Kirti teaches displaying them (e.g., to an administrator) in a historical report, further confirming that the anomalous access requests are *previous* access requests. *E.g.*, EX1004, 4:46-59, 6:35-44, 9:34-66, 11:15-58, 12:5-35, 12:50-57, 14:46-15:16, FIGS. 8B-8D. For example, FIG. 8C illustrates a report from October 26 that shows previous anomalous access requests (six failed logins and nine after hours accesses) on October 9.

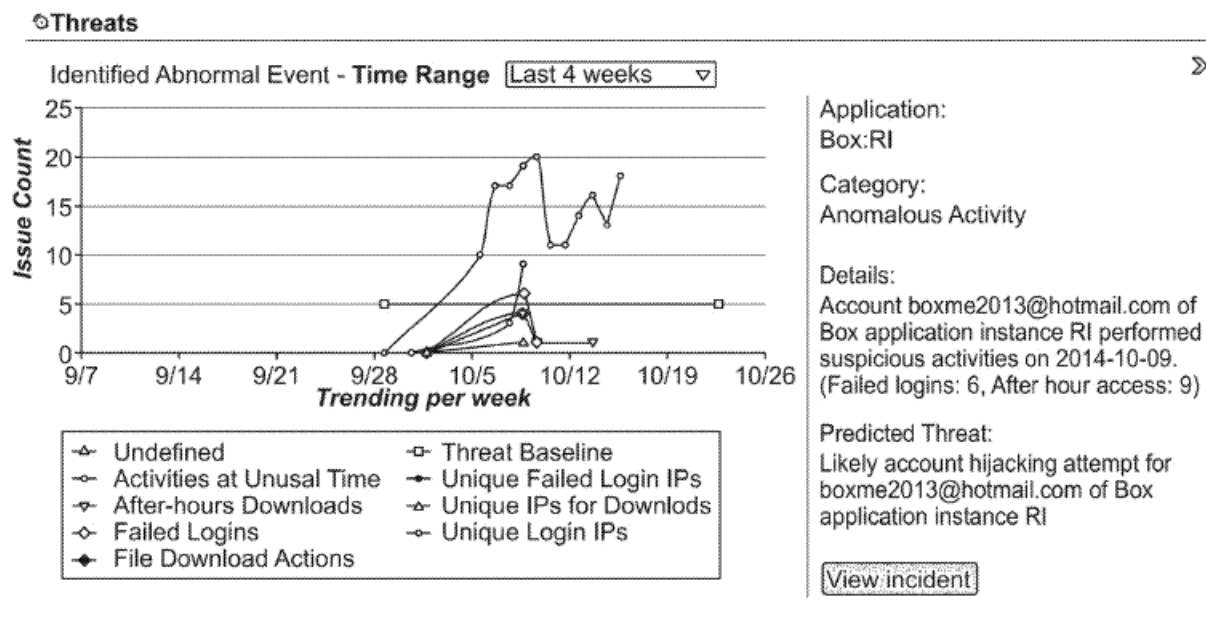


FIG. 8C

Id., FIG. 8C.

111. FIG. 8C (Cont.) also illustrates a report from October that indicates previous anomalous access requests (five failed logins, 12 suspicious IP addresses, four after hour accesses) on September 8.

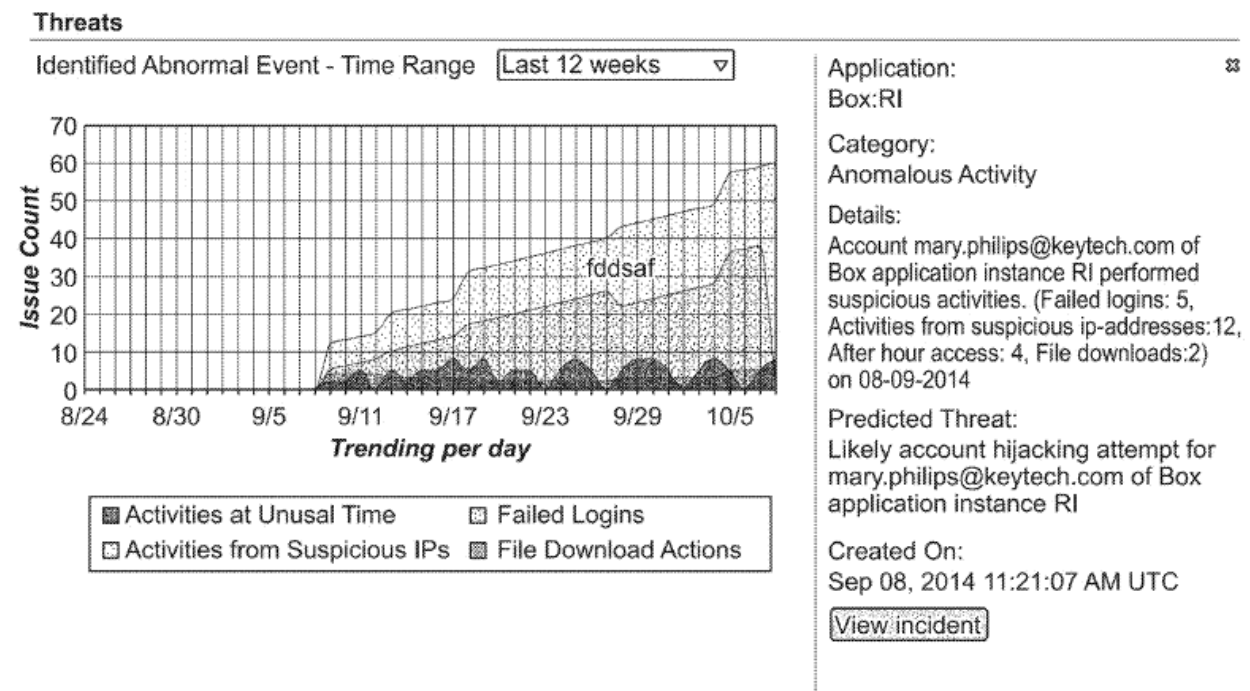


FIG. 8C (Cont.)

Id.

f) **Element [1.8]-[1.11]**

[1.8] based on the additional verification being required to grant access:

[1.9] select an additional verification method from a plurality of verification methods,

[1.10] cause the client to be prompted to complete the additional verification method, and

[1.11] determine whether the additional verification method has been completed correctly.

112. Kirti alone or in combination with Coffin teaches these claim elements. As explained above for elements [1.5]-[1.7] (*supra* Sections VIII.C.1.d-VIII.C.1.e), Kirti teaches recommending and performing a remedial action when a threat is detected and mentions that one such remedial action is “adding additional steps to authentication,” i.e., requiring completion of an additional verification step (e.g., enter a one-time password). *E.g.*, EX1004, 5:27-37, 28:24-36.

113. Kirti discloses manually or automatically selecting which one of a plurality of remedial actions to use. *E.g.*, EX1004, 5:27-37, 5:55-57 (“embodiments of the invention may include remediation functions that provide manual and/or automated processes in response to threats.”), 6:10-13, 14:64-67, 25:15-20, 25:45-52, 27:10-12.

114. And Kirti further specifies that the selected remedial action is “performed.” *E.g.*, *id.*, 25:40-44 (“Remediation action(s) to address a threat may be

performed automatically, if a response to such threats is predetermined, or may be instructed [] by a user selecting a remediation option from the alert that was presented.”). In the cases where the remedial action involved “additional steps to authentication,” such as a one-time password, performing that authentication would necessarily involve prompting the client to complete the additional authentication and then determining whether it was corrected completely. At the very least, this would have been obvious, as the most obvious way to perform such additional authentication would be to present the additional authentication steps for completion (i.e., “prompt”) and then confirm correct completion before allowing the login to complete. It was well known that these were necessary steps in any user-performed additional verification method (e.g., biometric (face or fingerprint) scan, one-time passcode entry, answer to secret question, etc.). *Supra* Section V.B.2.

Coffin’s teachings

115. While Kirti teaches selecting and performing additional verification (e.g., “additional steps for authentication,” such as “elevated authentication or OTP validation”), it provides little implementation detail for these steps. To the extent a POSITA did not find these details obvious based on Kirti alone, they were obvious further in view of Coffin, which expressly teaches details for selecting one of a plurality of verification methods, prompting the user to complete the additional verification method, and determining whether the user correctly completed the

additional verification method. EX1005, 177-208. Coffin provides these teachings in the context of an Oracle system, whereas Kirti is a patent assigned to Oracle, and reflects technology ultimately deployed as part of Oracle systems. EX1005, i, 1, 178; EX1004, Cover.

116. Coffin includes an entire chapter on implementing “Two-Factor Authentication.” EX1005, 177-208. Therein, Coffin teaches additional verification/authentication methods such as: a “second password or PIN,” “a CAPTCHA,” an “answer [to] personal questions,” a “biometric scan[], like fingerprint, retina, or facial recognition,” and “pass codes sent to [] e-mail, pager, or cell phone”). EX1005, 177. Coffin also explains how to implement additional verification methods that include a passcode, such as an SMS to a phone, a URL to a pager, and an email to supporting devices. *Id.*, 178-208.

117. Coffin also teaches a heuristic for selecting an additional verification method from the plurality of passcode verification methods: select the user’s pager and/or cell phone if these devices are available, otherwise select email. *E.g., id.*, 183, 185, 200 (“By preference, [the system] will send the two-factor code to the user’s pager and cell phone. If neither of those is available, [the system] will send the code to the user’s e-mail.”), 201. It also was well known in the art to select an additional verification method from a plurality of verification method, for example, to ensure

successful delivery of the additional verification based on the availability of the verification methods, cost, user preferences, etc. *Supra* Section V.B.2.

118. Once the additional verification method is selected, Coffin teaches sending the passcode to the device(s), which then prompts the user to enter the provided passcode to complete the additional verification method. *Id.*, 183 (“Once the user receives the two-factor pass code, they will submit that along with their request for data.”), 190, 193-194, 201, 207 (“Users will have to enter those 2-factor codes in order to get access to our application data.”).

119. The computer system then checks to ensure the user entered the correct code. *Id.*, 194 (“[W]e test the two-factor code to see if it passes muster by calling the `f is cur cached cd` function. If the two-factor code is good, we set the secure application role; however, if not we let them know by raising a `NO DATA FOUND` exception: they entered the wrong code, or perhaps just an old (older than 10 minutes) code.”), 201 (“[W]e want to cache [the code] for comparison to any code the user enters.”), 206 (“One definite potential error is the possibility that the user has entered a wrong or old code.”).

120. As explained above in Section VIII.C, a POSITA implementing Kirti, which teaches additional authentication methods but without much detail, would have been motivated to incorporate Coffin’s detailed teachings on how to implement additional authentication methods. Given Kirti’s lack of detail on these aspects, a

POSITA would have been motivated to look for implementation details elsewhere, naturally leading them to Coffin and its teachings on how to implement specific types of remedial measures in an Oracle database system. Coffin also expressly provides motivation for implementing the additional authentication: it ensures the user is who they say they are. EX1005, 177.

121. A POSITA would have reasonably expected success in implementing Coffin's additional verification teachings in Kirti's disclosed system at least because it would require only minor software changes that were well within the skill of a POSITA. *See supra* Section V.B.2 (explaining that additional authentication was well known in the art). The reasonable expectation of success is further supported by the fact that both Kirti and Coffin relate to the same type of Oracle systems.

2. Claim 2²

[2.1] The computer system of claim 1, wherein the historical information about previous access requests associated with the user account comprises historical information about previous requests to authenticate, wherein the previous requests to authenticate comprise the identifier.

122. Claim 2 is obvious at least because Kirti teaches the additional features recited by this dependent claim. As explained above for elements [1.6] and [1.7],

² For each dependent claim mapped herein, I incorporate my analysis for any claim from which that dependent claim depends.

Kirti teaches determining whether the user's historical information (e.g., "past activity data") contains an anomalous access request. *Supra* Section VIII.C.1.e. This activity data comprises historical information about previous requests to authenticate that include the identifier (e.g., "user name"). EX1004, 10:25-56, 12:50-13:42, 14:5-31, 18:10-27. For example, "[a]ctivity data can include ... login and logout statistics (including attempts and successes) ... [and] "the user account or other user identifier for the user associated with the events or statistics." *Id.*, 10:25-46; *see also id.*, 18:17-21 ("User activities may include events such as, but not limited to, logging in to the cloud application Such event activities can be logged with event details such as, but not limited to, a user name"). Thus, claim 2 is obvious at least because Kirti teaches that the historical information about previous access requests associated with the user account comprises historical information about previous requests to authenticate, wherein the previous requests to authenticate comprise the identifier.

3. Claim 3

[3.1] The computer system of claim 1, wherein determining whether the additional verification is required to grant access further comprises processing endpoint data from entities connected to the network.

123. Claim 3 is obvious at least because Kirti teaches the additional features recited by this dependent claim. As explained above for claims elements [1.5]-[1.7], Kirti teaches determining whether additional verification is required based on whether a threat (e.g., an anomalous access request) is detected in the past activity

data. *Supra* Sections VIII.C.1.d-VIII.C.1.e. This involves processing endpoint data (e.g., “IP addresses,” “infected node points” and login information) from entities connected to the network (endpoints within the network). *E.g., supra* Section VIII.C.1.b; EX1004, FIGS. 4, 5B; *id.*, FIG. 1 (showing different endpoints).

124. For example, Kirti discloses that activity from suspicious IP addresses is a type of threat, and discloses identifying and including these suspicious IP addresses in a threat report to a system administrator. EX1004, 5:55-6:5, 10:34-46, 12:18-27, 12:35-57, 12:65-13:4, 15:38-59, 28:22-36, FIG. 8C. Kirti processes the “IP addresses ... [and] devices used to access the application” (*id.*, 10:36-38), and identifies suspicious nodes/IP addresses based on external intelligence feeds and/or login activity, such as a number of failed login attempts, login velocity, etc. *Id.*, 5:55-6:5, 12:18-27, 12:35-57 (“external information ... relating to potential security threats such as identification of infected node points”), 12:65-13:4, 15:38-59 (“Metrics used for detection can include ... a count of the number of unique IP addresses used by a user per day and/or a velocity ... [and] unusually high number of login failures for existing valid accounts”), 28:22-36. Further, Kirti discloses that “[a]ctivities from suspicious ip-addresses” is a type of threat that is reported to a system administrator. *Id.*, FIG. 8C.

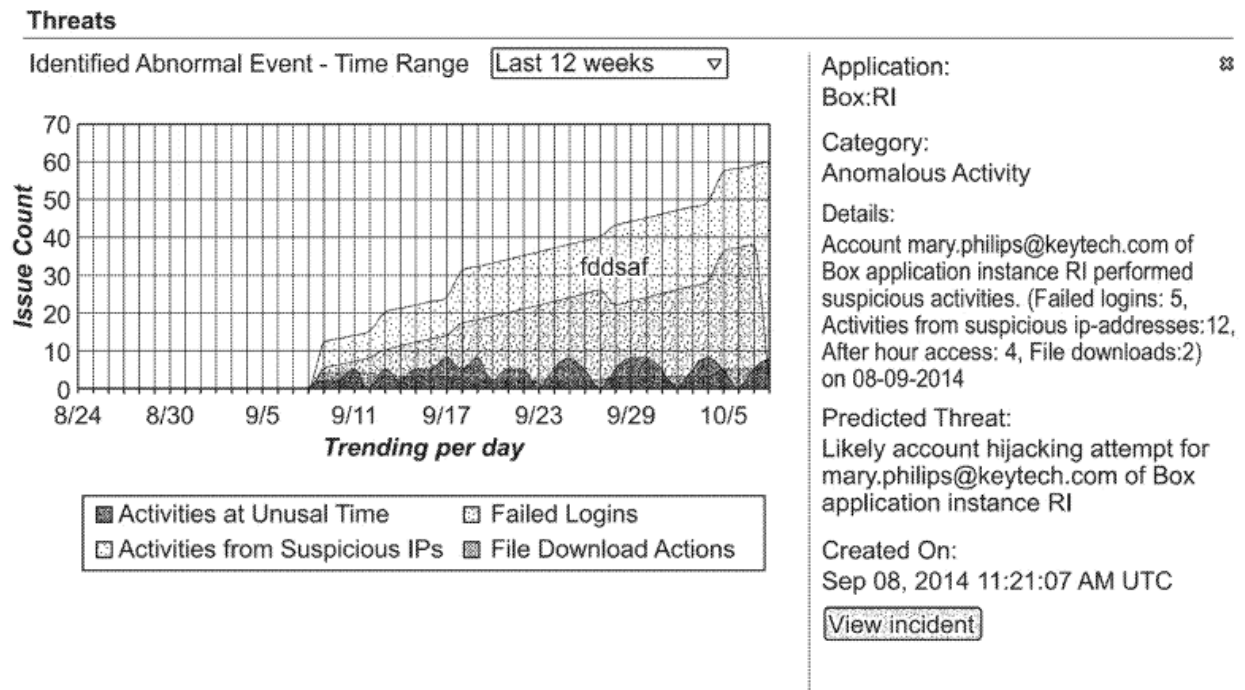


FIG. 8C (Cont.)

Id., FIG. 8C.

125. Thus, a POSITA would have understood or at least found it obvious that Kirti's system processes endpoint data from entities connected to the network at least because Kirti identifies endpoints ("nodes" / "IP addresses") that are suspicious. Further a POSITA would have understood or at least found it obvious that Kirti's system performs this processing of endpoint data when determining whether to require additional verification to grant access at least because Kirti teaches treating activity from suspicious endpoints as a threat, which, as explained above for elements [1.5]-[1.7] (*supra* Sections VIII.C.1.d-VIII.C.1.e), trigger the additional verification requirement.

4. Claim 4

[4.1] The computer system of claim 1, wherein determining whether the additional verification is required to grant access further comprises processing an external threat intelligence feed.

126. Claim 4 is obvious at least because Kirti teaches the additional features recited by this dependent claim. As explained above for claims elements [1.5]-[1.7], Kirti teaches determining whether additional verification is required based on whether a threat (e.g., an anomalous access request) is detected in the past activity data. *Supra* Sections VIII.C.1.d-VIII.C.1.e. Kirti further teaches that this threat detection involves processing an external threat intelligence feed: “Threats can also be identified by comparing activity data with external threat intelligence information, such as information provided by third-party providers” EX1004, 4:67-5:3; *see also id.*, 5:55-6:5 (“[A]nalytics can utilize information received from external third party feeds ... to augment the threat intelligence by providing external information of security threats such as, but not limited to, identification of infected node points, malicious activity from a particular source IP address, malware infected email messages, vulnerable web browser versions, and known attacks on clouds.”), 11:51-58, 12:35-57, 12:65-13:4, 17:8-13. Thus, Kirti teaches processing information from an external threat intelligence feed (e.g., information indicating that the request came from an infected node), as part of determining whether to require additional verification.

5. Claim 5

[5.1] The computer system of claim 1, wherein the software instructions further comprise instructions that:

[5.2] based on the additional verification being required to grant access:

[5.3] determine that a probable cyberattack is detected, and

[5.4] provide an alert,

[5.5] wherein the alert includes the identifier and an indicator that a probable cyberattack is detected, and

[5.6] wherein the alert is designated to be provided to an administrator of the network.

127. Claim 5 is obvious at least because Kirti teaches the additional features recited by this dependent claim. EX1004, 3:53-4:1, 4:54-59, 5:27-36, 13:12-17, 14:5-16:5, 25:3-44, 25:55-26:5, 28:22-37, FIGS. 6, 8B-8D.

128. Kirti teaches that based on the additional verification being required to grant access (e.g., due to a series of failed login attempts, a login attempt from an unknown/suspicious IP address, etc.), determining that a probable cyberattack (e.g., a brute force attack) is detected. *E.g.*, EX1004, 13:12-17, 14:5-7 (“detect[ing] attacks and unknown threats”), 15:31-48, 15:49-59 (“Detection [of a brute force attack] may involve evaluating the velocity of failed login attempts”).

129. Because Kirti discloses to “update statistics” “on various user behavioral activities” (EX1004, 18:28-31), a POSITA would have understood or at least found it obvious that if Kirti’s system determined that a user needs to supply additional verification information due to an anomalous login attempt in the most recent batch of activity data, such as an attempt from an unknown/suspicious IP

address (*see supra* Sections VIII.C.1.d-VIII.C.1.e; EX1004, 1:66-2:7, 5:57-6:5, 12:28-57, 15:31-48, 16:6-11, 16:60-17:8, 28:26-31, 29:31-36, FIG. 8C), the system would also determine that a probable cyberattack is detected if that same user account is associated with other anomalies, such as repeated failed login attempts, especially since Kirti teaches to identify certain types of cyberattacks, including brute force attacks (*e.g.*, EX1004, 15:23-16:24). Thus, a POSITA would have understood or at least found it obvious based on Kirti's teachings to determine, for example, that a probable brute force cyberattack is detected from a suspicious IP address based on that IP address being required to provide additional verification (because it is unknown/suspicious) and having failed multiple login attempts.

130. Kirti further discloses providing an alert to an administrator that includes the identifier (*e.g.*, username) and an indicator that a probable cyberattack is detected, such as a category or description of the probable cyberattack. EX1004, 3:53-4:1, 4:54-59, 5:27-36, 13:12-17, 14:38-15:16, 15:63-65, 25:3-44, 25:55-26:5, 28:22-37, FIGS. 6, 8B-8D. For example, Kirti discloses to “preemptively alert a system administrator with respect to threats” (*id.*, 5:31-32; *see also id.*, 14:39-40, 25:25-30, 25:55-56) and describes and illustrates a system administer user interface that displays alerts that include the user identifier (*e.g.*, email) and an indicator that a probable cyberattack is detected. *Id.*, 6:35-44, 9:52-65, 14:46-15:16, FIGS. 2, 8B-8D.

131. For example, FIG. 8B illustrates an example of a list of alerts for several different user accounts (e.g., boxme2013@hotmail.com, ralomari, ellen.schwab@keytech.com) that include the alert category and the alert details.

Incidents Create New Incident

Search Filter

ID

Cloud App

Category

Assigned to

Created on

Priority

Status

All

All

All

All

Open

Search

Reset

ID	Application	App Instance	Category	Priority	Assigned To	Created On	Details	Remediation Type	Status	Action
1020225	Box	RI	Security Control	High	i4qhckkju3fa	2014-10-23	Security level lower than recommended	Manual	Open	  
1020200	Box	RI	Security Control	High	i4qhckkju3fa	2014-10-23	Security level lower than recommended	Manual	Open	  
1020250	Box	RI	Anomalous Activity	Medium		2014-10-23	User Behavior Risk Related to Login Activity: Observed for boxme2013@hotmail.com of Box Application and Instance RI on 2014-10-24	Manual	Open	  
1020275	Box	RI	Anomalous Activity	High	i4qhckkju3fa	2014-10-20	Account boxme2013@hotmail.com of Box application instance RI performed suspicious activities	Manual	Open	  
1020300	Box	RI	Anomalous Activity	High	i4qhckkju3fa	2014-10-20	Account boxme2014@hotmail.com of Box application instance RI performed suspicious activities	Manual	Open	  
2840001	AWS	aws_inst	Anomalous Activity	High	i4qhckkju3fa	2014-10-20	Account ralomari of AWS application instance aws_inst performed suspicious activities	Manual	Open	  
1020105	SFDC	RI	Anomalous Activity	High	Support	2014-10-09	Account ellen.schwab@keytech.com of SFDC application instance RI performed suspicious activities	Manual	Open	  
12830108	SFDC	RI	Anomalous Activity	High	Admin	2014-10-09	User Behavior Risk Related to Mass Delete, Transfer, or Export: Observed for edward.murali@keytech.com of SFDC Application and Instance RI	Manual	Open	  
User Behavior Risk Related to Manage Users Changes:										

FIG. 8B

Id., FIG. 8B.

132. FIG. 8C illustrates “[l]ikely account hijacking attempt[s]” for two users: boxme2013@hotmail.com and mary.philips@keytech.com.

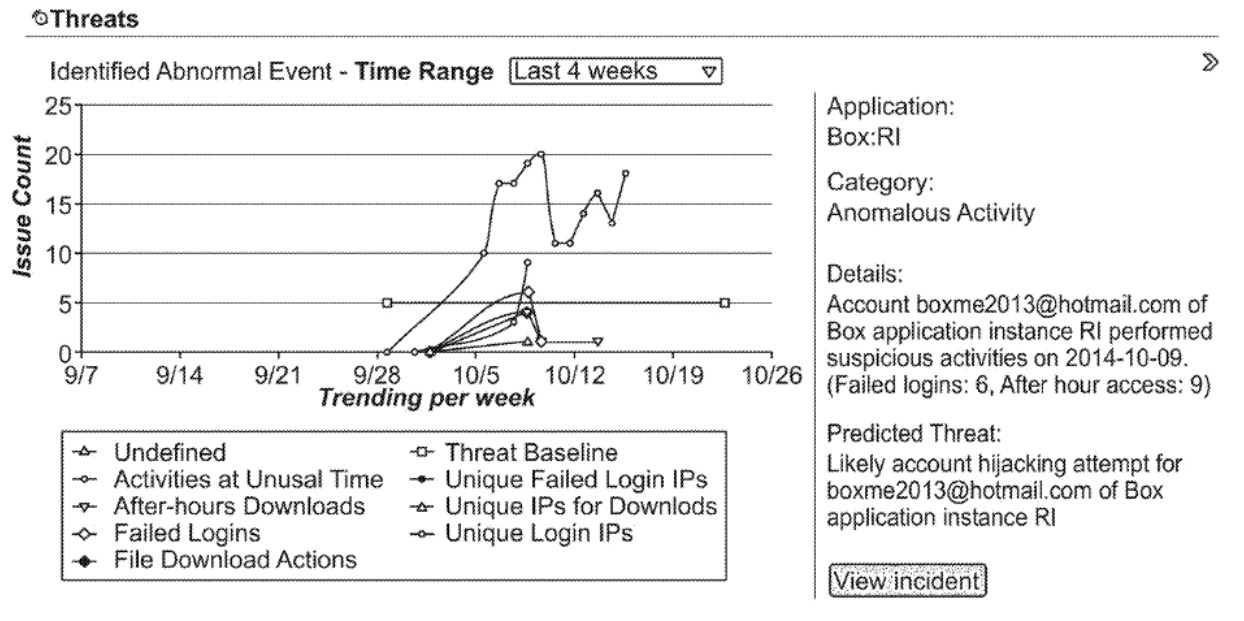


FIG. 8C

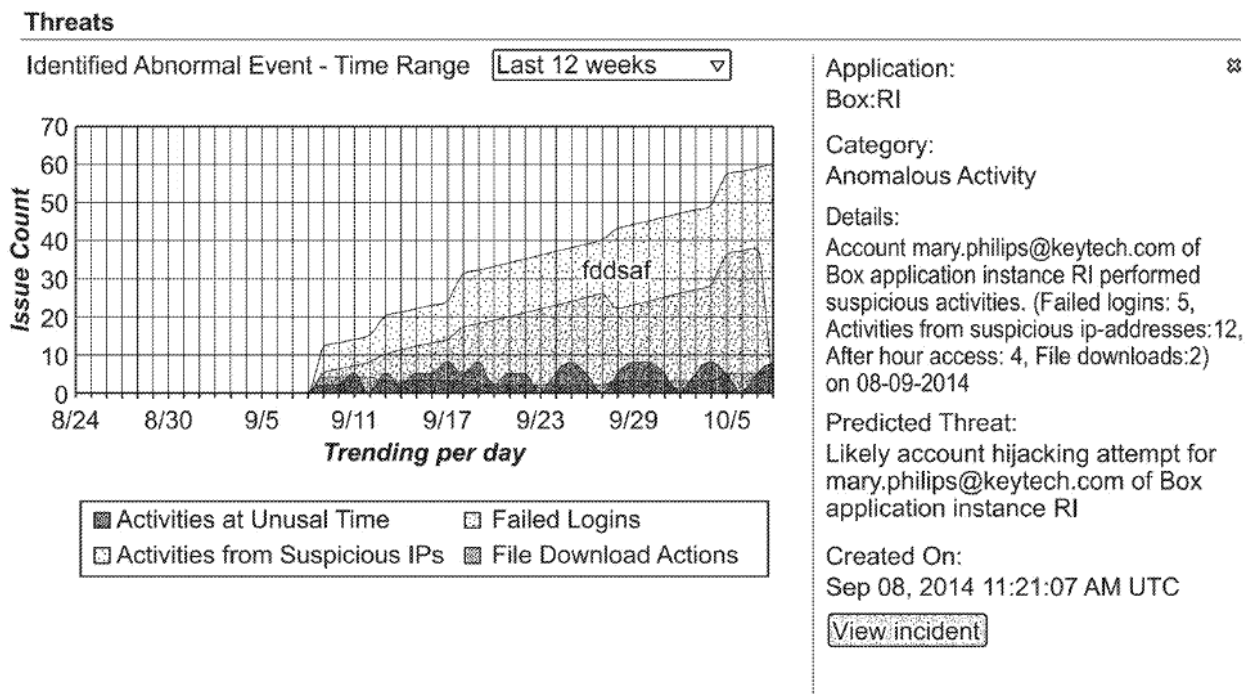


FIG. 8C (Cont.)

Id., FIG. 8C.

133. Similarly, FIG. 8D illustrates a list of alerts that each include the associated username and a description of the alert. For example, for user boxme2014@hotmail.com the alert is for suspicious activities—six failed login attempts.

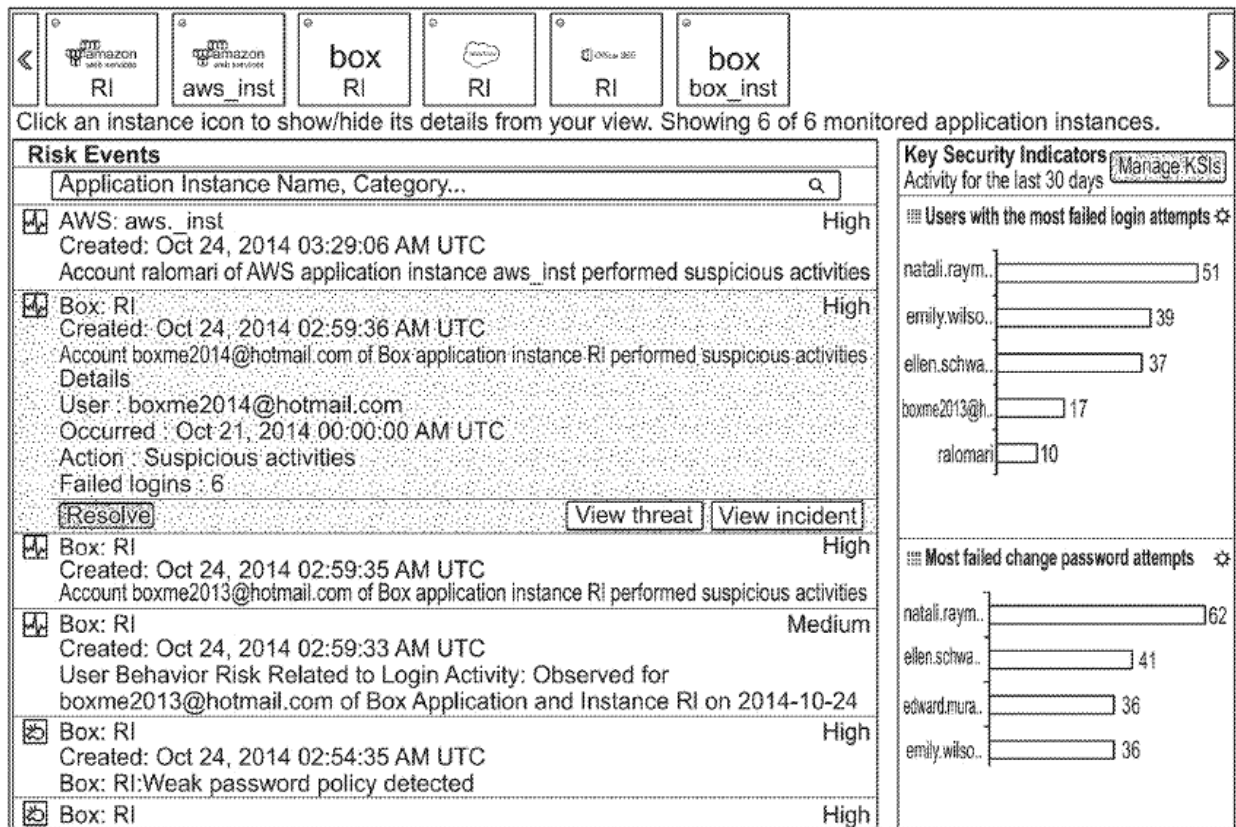


FIG. 8D

Id., FIG. 8D.

134. Kirti's detailed description further explains that the alerts "can include information about the detected event such as, but not limited to an event identifier, date, time, risk level, event category, user account and/or security controls associated with the event, cloud application associated with the event, description of the event,

remediation type (e.g., manual or automatic), and/or event status (e.g., open, closed).” *Id.*, 14:48-54.

6. Claim 6

[6.1] The computer system of claim 5, wherein the alert further includes an indication of where the probable cyberattack may have originated.

135. Claim 6 is obvious at least because Kirti teaches the additional features recited by this dependent claim. Specifically, Kirti teaches that the alert further includes an indication of where the probable cyberattack may have originated at least because Kirti discloses recording the IP address from which each access request was received, checking the IP addresses’ reputation (e.g., whether it is on a watch list), and recommending a remedial action if an anomaly is detected from that IP address (e.g., the IP address matches a blocked IP address or an IP address on a watch list, the IP address has suspicious activity such as a high volume of failed logins, etc.). EX1004, 3:58-4:1, 7:3-67, 9:60-65, 10:34-44, 12:7-18, 12:35-57, 12:64-13:2, 13:40-59, 15:31-43, 16:6-11, 16:55-17:55, 18:19-23, 18:38-42, 28:22-36, FIG. 8C (“Activities from suspicious ip-addresses”).

136. The system tracks the IP addresses associated with the requests at least because the activity data includes this information. “Activity data can include ... [the] IP addresses used to access the application.” *Id.*, 10:33-37; *see also id.*, 18:38-42. Indeed, Kirti teaches “display[ing] high-level information such as a map of IP

addresses of user accounts associated with the tenant's account that have accessed cloud applications.” *Id.*, 9:61-63; *see also id.*, FIG. 8A (suggesting a list of “Regular, Suspicious & Unknown Verified IPs”).

137. At least some of the threats (e.g., cyberattacks) that Kirti teaches detecting include an indication of where the detected threat originated (e.g., the IP address, geolocation). For example, “a threat can be identified based on an account accessing one or more files or failing a series of login attempts from an IP address that is flagged (by a third party feed or otherwise) as malicious.” *Id.*, 12:66-13:2; *see also, e.g., id.*, 6:3-4 (“identification of infected node points, malicious activity from a particular source IP address”), 12:41, 28:26-31. In these examples, POSITAs would have understood the account or IP address as the origination of the cyberattack. As another example, an application misuse threat is detected if the user logs in and performs suspicious activity from an unusual IP address or geolocation. *Id.*, 16:6-11 (“Application misuse is a scenario that may include ... a malware-infected device performing an unusual number of file downloads/uploads using valid credentials but an unusual geolocation or IP address.”).

138. Further, Kirti expressly discloses that the external intelligence feeds inform the system of the “source of attacks” (*id.*, 12:49), and the reports displayed to administrators include “login statistics (e.g., users with the most failed logins, IP address based login history including consideration of IP reputation, geolocation,

and other factors).” *Id.*, 12:7-11. Given this, POSITAs would have naturally implemented Kirti to provide this helpful and readily available information in the alert to beneficially allow the individuals or systems receiving the alert to better carry out any needed responsive action.

139. Thus, a POSITA would have understood or at least found it obvious that the alert includes an indication of where the probable cyberattack may have originated at least because Kirti tracks the IP addresses, detects IP address-based threats, and provides an administrator with information about specific IP addresses.

7. Claim 7

[7.1] The computer system of claim 5, wherein the alert further includes an indication of what enterprise information may be at risk in the probable cyberattack.

140. Claim 7 is obvious at least because Kirti teaches the additional features recited by this dependent claim. Specifically, Kirti teaches that the alert further includes an indication of what enterprise information may be at risk in the probable cyberattack at least because Kirti teaches tracking the resources the user accesses and generating an alert when the user accesses or downloads certain information, such as credit card numbers. EX1004, 10:34-44, 14:5-14, 15:60-16:5.

141. For example, the activity data includes the “cloud resources that were accessed (including, but not limited to, files and folders in a file management cloud application ... employees and contractors in a human resource cloud application ...

and contacts and accounts in a customer relationship management cloud application ...).” *Id.*, 10:38-44.

142. Further an alert is generated for certain high-risk operations, such as “downloading a file containing credit card numbers, copying encryption keys, elevating privileges of a normal user.” *Id.*, 14:11-13. Other activities that Kirti teaches tracking include “an unusually high use of corporate resources such as a high number of downloads and/or an employee with a low rating downloading or sharing an unusually high number of files/folders, deleting code from a source code control system, or downloading, deleting, or modifying customer information/contracts.” *Id.*, 15:66-6:5.

143. Thus, a POSITA would have understood or at least found it obvious that the alert includes an indication of what enterprise information may be at risk in the probably cyberattack at least because Kirti teaches tracking the enterprise information accessed by the user and flagging suspicious activity, such as downloads.

8. Claim 8

[8.1] The computer system of claim 1, wherein the software instructions further comprise instructions that:
deny the request based on determining that the user identification and the password are not associated with any user account, and
deny the request based on determining that the client failed to complete the additional verification method correctly.

144. Claim 8 is obvious at least because Kirti alone or in combination with Coffin teaches the additional features recited by this dependent claim, and because these features were well within the knowledge of a POSITA.

145. Kirti teaches denying the request if the user identification and the password are not associated with any user account at least because Kirti discloses “failed login attempts” (*e.g.*, EX1004, 13:12, 13:14, 13:58, 13:59, 15:7), which a POSITA would have understood entails denying the request based on determining that the user identification and the password are not associated with any user account since this was well known in the art, as acknowledged by the patent itself. EX1001, 2:22; *supra* Section V.B.1.

146. Similarly, Kirti teaches requiring “additional steps to authentication” and/or “elevated authentication or OTP validation” (EX1004, 5:34-35, 28:33), which a POSITA would have understood entails denying the request based on determining that the user failed to complete the additional verification method correctly since this was well known in the art, as acknowledge by the patent itself. EX1001, 2:20-34; *supra* Section V.B.2. Further confirming that it was well known to deny access requests if the additional verification is not completed correctly, Coffin expressly discloses that the system checks the additional verification information (*e.g.*, “two-factor code”) submitted by the user, and if it is incorrect, the system denies the access request (the user is not allowed to access the application data). EX1005, 183, 190,

193, 194 (“[W]e test the two-factor code to see if it passes muster by calling the f is cur cached cd function. If the two-factor code is good, we set the secure application role; however, if not we let them know by raising a NO DATA FOUND exception: they entered the wrong code, or perhaps just an old (older than 10 minutes) code.”), 201, 206, 207 (“Users will have to enter those 2-factor codes in order to get access to our application data.”).

147. Moreover, a POSITA would have understood or at least found it obvious to deny the request if the user entered the wrong username or password or failed the additional verification at least because the purpose of these verification processes is to ensure that only the user of the account can gain access to their account.

9. Claim 9

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

148. Claim 9 is obvious at least because Kirti teaches the additional features recited by this dependent claim. For example, Kirti teaches that the computer system comprises “multiple hardware platforms that are in communication with each other.” EX1004, 4:8-9; *see also id.*, FIG. 1; *supra* Section VIII.C.1.a).

10. Claim 10

[10.1] The computer system of claim 1, wherein the computer system comprises a plurality of virtual computing machines.

149. Claim 10 is obvious at least because Kirti expressly describes “instances of virtual machines” (EX1004, 12:57), “relates generally to cloud computing” (*id.*, 1:18-19), and teaches systems that include cloud resources (*e.g.*, *id.*, FIGS. 1, 3, 4). Indeed, a focus of Kirti is its “cloud security monitoring and control system.” *E.g.*, *id.*, 1:18-20, 1:24-40, 1:44-2:51, 3:41-58, 4:5-28. POSITAs knew that cloud computing implicated virtual machines, and thus would have understood the cloud aspects of Kirti’s system to include virtual machines—at the very least, it would have been obvious to implement those cloud aspects using virtual machines. For example, by as early as 2009, others in the field noted the “increasing popular[ity]” of “cloud computing,” and described it as “a single computer system” that “hosts a large number of ‘virtual machines.’” EX1046, 1:20-29 (further noting Amazon Web Services as one such cloud environment). Consistent with this common understanding in the art, the ’934 patent itself acknowledges “network computing clouds” as one example of “virtualized computing environments” (EX1001, 13:25-26) and provides no meaningful explanation of virtual machines. Moreover, Kirti specifically describes its users having “EC2 machines” (EX1004, 7:45 (Table 1), 12:11-12, 19:38 (Table 4)), which were known to be Amazon virtual

machines. EX1060, 1 (“Amazon EC2 ... use[s] Xen virtualization [] to support multiple virtual machine instances on a single physical server.”).

150. It was commonplace to utilize virtual computing machines because of their well-known benefits: improved resource utilization, scalability, increased reliability, and decreased cost. *Supra* Section V.B.4. For example, a 2014 Oracle tutorial explained the reasons “driving virtualization for industry standard computing today.” EX1044, 14/34. Because “the capacity in a single server is so large that it is almost impossible for most workloads to effectively use it,” “[t]he best way to improve resource utilization, and at the same time simplify data center management, is through virtualization.” *Id.*

151. Given Kirti’s teachings and disclosures on virtual machines and cloud computing, and POSITAs knowledge of these technologies, a POSITA would have found it at least obvious (if not inherent) when implementing a cloud computing system like Kirti’s to include a plurality of virtual computing machines due to their ubiquity and well-known benefits.

152. To the extent Patent Owner argues otherwise, Grounds 3-4 (*infra*, Section IX) explain how claim 10 is further obvious in view of Vemulapalli, which expressly teaches virtual machines and their benefits.

11. Claims 11-17

153. Claims 11-17 are method claims that recite elements identical or near-identical to those of system claims 1-7, respectively, and are obvious for the same reasons as claims 1-7. *See* EX1065 (claims listing, showing claims 1-7 and 11-17 in columns 1 and 2, respectively); *supra* Sections VIII.C.1–VIII.C.7.

12. Claims 18-25 And 26-30

154. Independent claims 18 and 26 are substantively identical and differ only in that the former is written in system form and the latter in method form. The only material difference between these two claims and independent claim 1 is that claims 18 and 26 relate to determining “a brute force attack” (Elements [18.7] and [26.7]) and claim 1 relates to determining a request is “anomalous relative to a baseline profile of access requests” (Element [1.7]). *See* EX1065 (claims listing showing elements in corresponding rows of columns 1, 3, and 4).

155. As explained above, however, Kirti expressly teaches that a brute force attack is one type of anomaly that it detects and, upon detecting, triggers additional verification. EX1004, 15:24-30, 15:49-59; *supra* Sections VIII.C.1.d-VIII.C.1.e); *see generally supra* Section VIII.C.1. Accordingly, Kirti teaches the determining whether an additional verification is required to grant access based on whether the user account is associated with a plurality of requests to authenticate comprising a brute force attack, as recited in Element [18.7] and [26.7].

156. Given this and the reasons explained for claim 1, Kirti also renders claims 18 and 26 obvious. *Supra* Section VIII.C.1.

157. Claims 19-25 depend, directly or indirectly, from claim 18. Claims 19-23 are obvious for the same reasons as claims 3-7, respectively, as these two sets of dependent claims recite elements that are materially indistinct from one another. EX1065; *Supra*, Sections VIII.C.3-VIII.C.7. Claims 24-25 are obvious for the same reasons as claims 9-10, respectively, as these two sets of dependent claims recite elements that are materially indistinct from one another. EX1065; *Supra* Sections VIII.C.9-VIII.C.10

158. Claims 27-30 depend, directly or indirectly, from claim 26. Claims 27-30 are obvious for the same reasons as claims 3-6, respectively, at least because these two groups of claims recite substantively identical claim elements. *Supra* Sections VIII.C.3-VIII.C.6.

**IX. GROUND 3-4: CLAIMS 10 AND 25 ARE OBVIOUS OVER
KIRTI AND VEMULAPALLI (GROUND 3) OR
KIRTI, COFFIN, AND VEMULAPALLI (GROUND 4)**

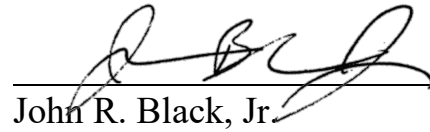
159. As explained above, Kirti alone or in combination with Coffin renders claims 10 and 25 obvious at least because POSITAs in October 2017 (and October 2015) knew that virtual machines were extremely beneficial, if not a given, in any cloud computing system like Kirti's. *Supra* Section VIII.C.10. These claims are further obvious in view of Vemulapalli (EX1046), which expressly teaches using

virtual machines in cloud computing systems, and explains the benefits of doing so: “The use of ‘cloud computing,’ in which a single computer system ... hosts a large number of ‘virtual machines’ has become increasingly popular recently. Such environments provide on-demand scalability, increased reliability, and decreased capital costs for users, who no longer are forced to purchase dedicated hardware on which to run their applications. ...” EX1046, 1:20-40; *see also id.*, 11:35-41, 20:5-12, 20:54-57, FIGS. 3, 5.

160. A POSITA would have been motivated to incorporate virtual machines in the Kirti and Kirti-Coffin systems for the advantages expressly stated in Vemulapalli, which were also well known in the art. *Supra* Section V.B.4. A POSITA would have reasonably expected success in doing so at least because (1) Vemulapalli (also assigned to Oracle) describes implementing VMs in the same type of cloud computing system disclosed by Kirti (*e.g.*, EX1004, Abstract, 1:18-20, 1:44-2:51; EX1046, Abstract, 2:18-20, 6:5-39, 21:32-22:39 (claim 1)), and (2) VMs were well known and routinely—if not necessarily—implemented in a cloud computing system like Kirti’s (*supra* Sections V.B.4, VIII.C.10).

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: Dec 29, 2025



John R. Black, Jr.

Appendix A

John R. Black, Jr.

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

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

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

Position	Assoc. Prof. Computer Science, University of Colorado at Boulder	<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, SECURE 2006.
Program Committee, ACSAC 2006.
Program Committee, CRYPTO 2005.
Program Committee, SAC 2005.
Program Committee, ICISC 2005.
Program Committee, CANS 2005.
Program Committee, IEEE SISW 2005.
Program Committee, CRYPTO 2004.
Program Committee, EUROCRYPT 2004.
Program Committee, RSA-CT 2003.
NSF CISE Panelist, 2001, 2003, 2005, 2006, 2007, 2009.
Referee for Journal of Cryptography, 1999–2006.
Referee for Software: Practice and Experience, 2005.
Referee for IEEE Communications Magazine, 2005.
Referee for IEEE Transactions on Circuits and Systems I, 2005.
Referee for IEEE Computer, 2005.
Referee for Journal of Computer Security, 2004.
Referee for IEEE Transactions on Information Theory, 2003.
Referee for IEEE Transactions on Computers, 2002.
Reviewer for CRYPTO 1999–2002, SODA 1998, SPAA 2002, Asiacrypt 2004, EURO-CRYPT 2006.

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

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

Graduate Student Mixer organizer, CRYPTO 2005.

**Internal
Service**

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

Chair, Space Committee, 2012–2015.

Liaison, Casey Feldman Foundation, 2010–2015.

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

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

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

Chair, Departmental Search Committee, 2008–2009.

Member, Graduate Committee, 2005–2006.

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