

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent of: Fabrice Delhoste, et al.
U.S. Patent No.: 11,387,982 Attorney Docket No.: 56407-0019IP1
Issue Date: July 12, 2022
Appl. Serial No.: 16/958,252
Filing Date: November 12, 2018
Title: METHOD FOR AUTHENTICATING A HASH OF A DATA
PACKET FROM COLLECTED CONTEXT DATA BASED ON A
HASH DISTANCE GENERATION STEP, FIRST
AUTHENTICATION DEVICE, SECOND AUTHENTICATION
DEVICE, AND AUTHENTICATION SYSTEM

DECLARATION OF DR. PATRICK TRAYNOR

I declare that all statements made herein on 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 under Section 1001 of Title 18 of the United States Code.

Date: 27 Feb

By: 
Patrick Traynor, Ph.D.

TABLE OF CONTENTS

I.	QUALIFICATIONS AND BACKGROUND INFORMATION.....	5
II.	LEGAL PRINCIPLES.....	14
A.	Anticipation	14
B.	Obviousness.....	15
III.	OVERVIEW OF CONCLUSIONS FORMED	16
IV.	BACKGROUND KNOWLEDGE ONE OF SKILL IN THE ART WOULD HAVE HAD PRIOR TO THE PRIORITY DATE OF THE '982 PATENT	17
V.	INTERPRETATIONS OF THE '982 PATENT CLAIMS AT ISSUE.....	17
VI.	TECHNOLOGY OVERVIEW.....	18
A.	Hashing Systems	18
1.	Hashing to Detect Identical Data Sets	19
2.	Hashing to Detect Similar Data Sets.....	20
B.	Policy-based Management	22
1.	Policy Examples.....	22
2.	Policy Specification	26
VII.	THE '982 PATENT	29
A.	Overview of the '982 Patent.....	29
B.	Prosecution History of the '982 Patent	34
VIII.	OVERVIEW OF PRIOR ART REFERENCES.....	36
A.	Overview of Starner (EX1004)	36
B.	Overview of Leskovec (EX1007)	43
C.	Overview of Shaashua (EX1005).....	47

D.	Overview of Bowman (EX1009)	51
IX.	GROUND 1A: CLAIMS 1-3, 5-6, 8-10 ARE OBVIOUS IN VIEW OF THE STARNER-LESKOVEC COMBINATION	54
A.	Combining Starner and Leskovec	54
1.	Overview of the Combination (SLC).....	54
2.	Motivations to Combine	60
3.	Reasonable Expectation of Success	63
B.	Analysis with Respect to Claims 1-3, 5-6, 8-10	63
1.	Claim 1	63
2.	Claim 2	79
3.	Claim 3	83
4.	Claim 5	87
5.	Claim 6	88
6.	Claim 8	91
7.	Claim 9	94
8.	Claim 10	97
X.	GROUND 1B: CLAIMS 1-6, 8-10 ARE OBVIOUS IN VIEW OF THE STARNER-LESKOVEC-SHAASHUA COMBINATION	99
A.	Combining Starner-Leskovec with Shaashua	99
1.	Overview of the Combination (SLSC)	99
2.	Motivations to Combine	104
3.	Reasonable Expectation of Success	105
B.	Analysis with Respect to Claims 1-6, 8-10	105
1.	Claim 1	106

2.	Claim 4.....	108
3.	Claim 5.....	110
4.	Claim 6.....	111
5.	Claim 8.....	112
6.	Claim 10.....	112
XI.	GROUND 1C: CLAIMS 6-7 ARE OBVIOUS IN VIEW OF THE STARNER-LESKOVEC-BOWMAN COMBINATION	112
A.	Combining Starner-Leskovec with Bowman	112
1.	Overview of the Combination (SLBC).....	112
2.	Motivations to Combine	115
3.	Reasonable Expectation of Success	117
B.	Analysis with Respect to Claims 6-7	117
1.	Claim 6.....	117
2.	Claim 7.....	119
XII.	GROUND 1D: CLAIMS 1, 6-7 ARE OBVIOUS IN VIEW OF THE STARNER-LESKOVEC-SHAASHUA-BOWMAN COMBINATION ...	121
XIII.	CONCLUSION.....	122

DECLARATION OF DR. PATRICK TRAYNOR

I, Patrick Gerard Traynor, of Gainesville, Florida, declare that:

I. QUALIFICATIONS AND BACKGROUND INFORMATION

1. My name is Patrick Gerard Traynor and I have been retained as an expert witness by Okta Inc. in the petition for IPR of U.S. Patent No. 11,387,982.

My qualifications for forming these conclusions are summarized below.

2. I earned a B.S. in Computer Science from the University of Richmond in 2002 and an M.S. and Ph.D. in Computer Science and Engineering from the Pennsylvania State University in 2004 and 2008, respectively. My dissertation, entitled “Characterizing the Impact of Rigidity on the Security of Cellular Telecommunications Networks,” focused on security problems that arise in cellular infrastructure when gateways to the broader Internet were created.

3. I am currently a Professor in the Department of Computer and Information Science and Engineering (CISE) at the University of Florida. I was hired under the “Rise to Preeminence” Hiring Campaign and serve as the Interim Chair in my Department. I also hold the endowed position of the John and Mary Lou Dasburg Preeminent Chair in Engineering.

4. Prior to joining the University of Florida, I was an Associate Professor from March to August 2014 and an Assistant Professor of Computer Science from

2008 to March 2014 at the Georgia Institute of Technology. I have supervised many Ph.D., M.S., and undergraduate students during the course of my career.

5. My area of expertise is security, especially as it applies to mobile systems and networks, including cellular networks. As such, I regularly teach students taking my courses and participating in my research group to program and evaluate software and architectures for mobile and cellular systems. I have taught courses on the topics of network and systems security, cellular networks, and mobile systems at both Georgia Tech and the University of Florida. I also advised and instructed the Information Assurance Officer Training Program for the United States Army Signal Corps in the Spring of 2010.

6. I have received numerous awards for research and teaching, including being named a Kavli Fellow (2017), a Fellow of the Center for Financial Inclusion (2016), and a Research Fellow of the Alfred P. Sloan Foundation (2014). I also won the Lockheed Inspirational Young Faculty Award (2012), was awarded a National Science Foundation (NSF) CAREER Award (2010), and received the Center for Enhancement of Teaching and Learning at Georgia Tech's "Thanks for Being a Great Teacher" Award (2009, 2012, 2013).

7. I have published over 100 articles in top conferences and journals in the areas of information security, mobile systems, and networking. Many of my results are highly cited, and I have received multiple "Best Paper" Awards. I have also

written a book entitled “Security for Telecommunications Networks”, which is used in wireless and cellular security courses at a number of top universities.

8. I am a Senior Member of the Association for Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers (IEEE). I am also a member of the USENIX Advanced Computing Systems Association.

9. I serve as an Associate Editor for IEEE Security and Privacy Magazine, have been the Program Chair for eight conferences and workshops, and have served as a member of the Program Committee for over 50 different conferences and workshops. I also served as the Security Subcommittee Chair for the ACM US Technology Policy Committee (USACM).

10. I was a co-Founder and Research Fellow for a private start-up, Pindrop Security, from 2012 to 2014. Pindrop provides anti-fraud and authentication solutions for Caller-ID spoofing attacks in enterprise call centers by creating and matching acoustic fingerprints. Pindrop Security currently employs over 200 people, and their technology is based off of my research (US Patent 9,037,113 B2).

11. I was a co-Founder and Chief Executive of a private start-up, CryptoDrop. CryptoDrop developed a ransomware detection and recovery tool to provide state of the art protection to home, small business, and enterprise users. This technology was also based off of my research (US Patent 10,685,114 B2).

12. I was also a co-Founder and Chief Executive of a private start-up, Skim Reaper. Skim Reaper developed tools to detect credit card skimming devices, and worked with a range of banks, international law enforcement, regulators, and retailers. This technology was also based off of my research (US Patent 10,496,914 B2).

13. I am a named inventor on more than ten US patents. These patents detail methods for determining the origin and path taken by phone calls as they traverse various networks, cryptographically authenticating phone calls, providing a secure means of indoor localization using mobile/wireless devices, detecting credit card skimmers, identifying cloned credit cards, and blocking ransomware from encrypting data.

14. My curriculum vitae, included with this declaration as Appendix A, includes a list of publications on which I am a named author. It contains further details regarding my experience, education, publications, and other qualifications to render an expert opinion in connection with this proceeding.

15. In writing this Declaration, I have considered the following: my own knowledge and experience, including my work experience in mobile systems and networks; my experience in teaching those subjects; and my experience in working with others involved in those fields. In addition, I have analyzed the following publications and materials, in addition to other materials I cite in my declaration:

- EX1001 U.S. Patent No. 11,387,982 to Delhoste, et al. (“the ’982 Patent”)
- EX1002 Excerpts from the Prosecution History of the ’982 Patent (“the Prosecution History”)
- EX1004 U.S. Pat. App. Pub. No. 2014/0337634 (“Starner”)
- EX1005 U.S. Pat. No. 9,355,234 (“Shaashua”)
- EX1007 (Excerpts) Leskovec, et al., “Mining Massive Datasets,” Second Edition, Cambridge University Press, 2014 (“Leskovec”)
- EX1009 U.S. Pat. App. Pub. No. 2018/0069699 (“Bowman”)
- EX1014 Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.3, draft-ietf-tls-tls13-11” December 28, 2015 (“Rescorla”)
- EX1015 Rescorla, IETF RFC 2818, “HTTP over TLS,” May 2000 (“Rescorla-2”)
- EX1016 Krawczyk, IETF RFC 5869, “HMAC-based Extract-and-Expand Key Derivation Function (HKDF),” May 2010 (“Krawczyk”)

- EX1022 Dursun, et al., A generic policy-conflict handling model.
In International Symposium on Computer and Information Sciences (pp. 193-204). Berlin, Heidelberg: Springer Berlin Heidelberg. (“Dursun”)
- EX1023 Merriam-Webster’s Advanced LEARNER’S English Dictionary, Merriam-Webster, Incorporated, 2017.
- EX1028 U.S. Pat. App. Pub. No. 2014/0108800 (“Lawrence”)
- EX1029 U.S. Pat. App. Pub. No. 2017/0109514 (“Cheng”)
- EX1030 U.S. Pat. App. Pub. No. 2006/0020796 (“Aura”)
- EX1031 U.S. Pat. App. Pub. No. 2008/0181403 (“Sakamoto”)
- EX1032 U.S. Pat. No. 9,514,753 (“Sharifi”)
- EX1033 U.S. Pat. App. Pub. No. 2004/0221153 (“Chul-min Kim”)
- EX1034 EP 2924603 A2 (“Bisikalo”)
- EX1035 Westerinen, et al., IETF RFC 3198, Terminology for Policy-Based Management, November 2001 (“Westerinen”)

- EX1036 Dang, “Recommendation for Applications Using Approved Hash Algorithms,” NIST SP 800-107 Rev. 1, August 2012 (“Dang”)
- EX1037 Lee, et al., “Lee, D.C., Ke, Q. and Isard, M., 2010, September. Partition min-hash for partial duplicate image discovery. In European Conference on computer vision (pp. 648-662). Berlin, Heidelberg: Springer Berlin Heidelberg (“Lee”)
- EX1038 U.S. Pat. App. Pub. No. 2017/0286544 (“Hunt”)
- EX1039 Chum, et al., “Scalable Near Identical Image and Shot Detection,” In Proceedings of the 6th ACM international conference on Image and video retrieval (pp. 549-556), July 2007 (“Chum”)
- EX1040 Chum, et al., “Near Duplicate Image Detection: min-Hash and tf-idf Weighting,” Bmvc (Vol. 810, pp. 812-815), September 2008 (“Chum-2”)
- EX1041 “Secure Hash Standard (SHS),” Federal Information Processing Standards Publication, FIPS PUB 180-4, August 2015
- EX1042 U.S. Pat. App. Pub. No. 2013/0290521 (“Labovitz”)

- EX1043 U.S. Pat. App. Pub. No. 2013/0254833A1 (“Nicodemus”)
- EX1044 Damianou, et. al., “The ponder policy specification language,” In International Workshop on Policies for Distributed Systems and Networks (pp. 18-38). Berlin, Heidelberg: Springer Berlin Heidelberg, January 2001 (“Damianou”)
- EX1045 U.S. Pat. App. Pub. No. 2014/0331225 (“Helander”)
- EX1046 Sutcu, *et al.*, “A secure biometric authentication scheme based on robust hashing,” in Proceedings of the 7th Workshop on Multimedia and Security, August 2005 (pp. 111-116) (“Sutcu”)
- EX1047 U.S. Pat. App. Pub. No. 2008/0189758 (“Akeel”)
- EX1048 U.S. Pat. App. Pub. No. 2007/0220061 (“Tirosh”)
- EX1049 Lorch, et al., "First experiences using XACML for access control in distributed systems." In Proceedings of the 2003 ACM workshop on XML security, pp. 25-37. 2003 (“Lorch”)
- EX1050 Verma, et. al., (2002). Policy-based management of content distribution networks. IEEE network, 16(2), 34-39. (“Verma”)

- EX1051 Lobo, et. al., (1999). A policy description language. AAI/IAAI, 1999, 291-298. (“Lobo”)
- EX1052 Stone, et al., 2001. Network policy languages: a survey and a new approach. IEEE network, 15(1), pp.10-21. (“Stone”)
- EX1053 Gonçalves, et al., 2009, October. A graphical user interface for policy composition in CIM-SPL. In 2009 International Conference on Ultra Modern Telecommunications & Workshops (pp. 1-7). IEEE. (“Gonçalves”)
- EX1054 Shay, et al., "Encountering stronger password requirements: user attitudes and behaviors." In Proceedings of the sixth symposium on usable privacy and security, pp. 1-20. 2010. (“Shay”)
- EX1055 Moses, et al., “eXtensible Access Control Markup Language (XACML) Version 2.0,” Committee Draft 04, December 6, 2004 (“Moses”)
- EX1056 U.S. Pat. No. 6,460,050 (“Pace”)
- EX1059 Krawczyk, et al., “HMAC-based Extract-and-Expand Key Derivation Function (HKDF),” IETF RFC 5869 (“Krawczyk”)

- EX1060 “Programming languages — C++,” International Standard ISO/IEC 14882, Second Edition, October 15, 2003. (Excerpts)
- EX1061 “Let’s Encrypt Stats,”
- EX1062 “HTTPS encryption on the web – Google Transparency Report”
- EX1063 Swami, et al., “QoS web service Security Access Control Case Study Using HTTP Secured Socket Layer Approach,” Proceedings of the International Conference on Engineering & MIS, pp. 1-9 (2015)
- EX1064 Chen, et al., “Client-Aware Cloud Storage,” IEEE 30th Symposium on Mass Storage Systems and Technologies (MSST), pp. 1-12 (2014)

II. LEGAL PRINCIPLES

A. Anticipation

16. I have been informed that a patent claim is invalid as anticipated under 35 U.S.C. § 102 if each and every element of a claim, as properly construed, is found either explicitly or inherently in a single prior art reference. Under the principles of inherency, if the prior art necessarily functions in accordance with, or includes the claimed limitations, it anticipates.

17. I have been informed that a claim is invalid under 35 U.S.C. § 102(a) if the claimed invention was known or used by others in the U.S., or was patented or published anywhere, before the applicant's invention. I further have been informed that a claim is invalid under 35 U.S.C. § 102(b) if the invention was patented or published anywhere, or was in public use, on sale, or offered for sale in this country, more than one year prior to the filing date of the patent application. And a claim is invalid, as I have been informed, under 35 U.S.C. § 102(e), if an invention described by that claim was described in a U.S. patent granted on an application for a patent by another that was filed in the U.S. before the date of invention for such a claim.

B. Obviousness

18. I have been informed that a patent claim is invalid as "obvious" under 35 U.S.C. § 103 in light of one or more prior art references if it would have been obvious to a POSITA, taking into account (1) the scope and content of the prior art, (2) the differences between the prior art and the claims, (3) the level of ordinary skill in the art, and (4) any so called "secondary considerations" of non-obviousness, which include: (i) "long felt need" for the claimed invention, (ii) commercial success attributable to the claimed invention, (iii) unexpected results of the claimed invention, and (iv) "copying" of the claimed invention by others. For purposes of my analysis, and at the direction of counsel, I have applied the December 29, 2017 filing date of EP Application No. 17306977 listed on the face of the '982 Patent as

the date of invention in my obviousness analyses, although in many cases the same analysis would hold true even at an earlier time than December 29, 2017. I refer to December 29, 2017 as the “Critical Date.”

19. I have been informed that a claim can be obvious in light of a single prior art reference or multiple prior art references. To be obvious in light of a single prior art reference or multiple prior art references, there must be a reason to modify the single prior art reference, or combine two or more references, in order to achieve the claimed invention. This reason may come from a teaching, suggestion, or motivation to combine, or may come from the reference or references themselves, the knowledge or “common sense” of one skilled in the art, or from the nature of the problem to be solved, and may be explicit or implicit from the prior art as a whole. I have been informed that the combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results. I also understand it is improper to rely on hindsight in making the obviousness determination.

III. OVERVIEW OF CONCLUSIONS FORMED

20. This expert Declaration explains the conclusions that I have formed based on my analysis. To summarize those conclusions, based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claims 1-10 of the '982 Patent are obvious for the reasons explained below.

IV. BACKGROUND KNOWLEDGE ONE OF SKILL IN THE ART WOULD HAVE HAD PRIOR TO THE PRIORITY DATE OF THE '982 PATENT

21. Based on the foregoing and upon my experience in this area, a person of ordinary skill in the art ("POSITA") relating to the subject matter of the '982 Patent would have had at least a bachelor's degree in computer science, computer engineering, or a related field, and at least two years of experience in researching, designing, or developing software and authentication systems, or the equivalent, as of the Critical Date. Additional education could substitute for professional experience, and professional experience for education.

22. Based on my experiences, I have a good understanding of the capabilities of a POSITA as I was such an individual at the time of the Critical Date. Moreover, I have taught, participated in organizations, and worked closely with many such persons over the course of my career.

V. INTERPRETATIONS OF THE '982 PATENT CLAIMS AT ISSUE

23. I have been informed by Counsel and understand that the best indicator of claim meaning is its usage in the context of the patent specification as understood by one of ordinary skill. I further understand that the words of the claims should be given their plain meaning unless that meaning is inconsistent with the patent specification or the patent's history of examination before the Patent Office. Counsel has also informed me, and I understand that, the words of the claims should be

interpreted as they would have been interpreted by one of ordinary skill at the time of the invention was made (not today). I have been informed by counsel for the Petitioner that I should use December 29, 2017 as the point in time for claim interpretation purposes.

VI. TECHNOLOGY OVERVIEW

A. Hashing Systems

24. Dang defines a hash algorithm as a “function that maps a bit string of arbitrary length to a fixed-length bit string.” EX1036, 7. As I discuss further below, hash function can be used for various purposes including efficiently detecting whether two data sets are identical (often called “cryptographic hashes”) and efficiently whether two data sets are similar, e.g., using “locality-sensitive” hash functions.

25. As one example, in a 2005 paper, Sutcu “propose[s] a secure biometric based authentication scheme which fundamentally relies on the use of a robust hash function.” EX1046, 2. Sutcu explains that “[d]uring authentication, a matching algorithm tries to match the biometric data acquired by a sensor with the templates stored in the template database,” and “[a]ccording to the result of the matching algorithm, authentication succeeds or fails.” *Id*

26. Note that hashing algorithms used to determine whether data sets are identical are not necessarily well-suited for determining whether data sets are nearly

identical, so different hash algorithms can be selected depending on the intended use of the hash output – determining exact matches, determining similarity, etc.

1. Hashing to Detect Identical Data Sets

27. A cryptographic hash function – that is, a hash function used to detect whether two data sets are identical -- is expected to have “collision resistance,” meaning that it “is computationally infeasible to find two different inputs to the hash function that have the same hash value.” EX1036, 9. Put more simply, different inputs to the hash function should, with very high likelihood, produce different output values. Even two strings with all but a single bit being the same should produce dramatically different outputs when cryptographic hashes are used. Therefore, with very high statistical likelihood, two data sets are identical if and only if they hash to the same value.

28. Note that a hash function does not guarantee that two input values never hash to the same output value since the output values have fewer bits than the input values, such a guarantee is not possible. However, hash algorithms are designed to resist such collisions.

29. One common use for a hash function is to ensure message integrity. For example, if a message is transmitted with both its contents and a hash value computed from the message, a receiver can compute the hash value for the received message and compare the computed hash value with the transmitted hash value. If

the two hash values are the same, then with high likelihood, the message was transmitted correctly; if the two values differ, the message was corrupted (or altered intentionally) during transmission. Since hash values are typically shorter than the message that is hashed, comparing hash values is more computationally efficient than comparing entire messages.

30. For example, SHS, a publication from the National Institute of Standards and Technology (NIST), is a standard that “specifies secure hash algorithms, SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224 and SHA-512/256,” which are “algorithms [that] are iterative, one-way hash functions that can process a message to produce a condensed representation called a message digest.” EX1041, 8. “These algorithms enable the determination of a message’s integrity: any change to the message will, with a very high probability, result in a different message digest.” *Id.*

2. Hashing to Detect Similar Data Sets

31. As noted above, hashing can also be used to determine not only whether two data sets are identical, but also whether two data sets are nearly duplicates. For example, Lee explains that the “min-hash” algorithm “is a standard hashing scheme for discovering near-duplicate text documents or web pages.” EX1037, 1.

32. Min-hash is an example of a “Locality Sensitive Hashing scheme [] that approximates similarity between sets.” EX1037, 3. Locality Sensitive Hashing

(LSH) “relies on the principle that if documents are similar they should hash to approximately the same value.” EX1038, [0106]. Therefore, “given some similarity threshold and N hash functions, a MinHash¹ function may be sampled in such a way that two documents are candidate pairs for similarity if and only if at least one of their LSH buckets are identical and share the same offset within the signature.” *Id.*

33. While min-hash was originally used to compare documents, since at least 2007, min-hash has also been used to compare images. For example, Lee explains that “min-hash and its variants have been successfully applied to discovering near duplicate images [], image clustering, image retrieval and object discovery.” EX1037, 1. Chum also “adapt[ed] a method originally developed for text near-duplicate detection [] to near-duplicate detection of images.” EX1040, 4-5; *see also*, EX1039.

34. The algorithms used to min-hash documents and to min-hash images are essentially the same, even though the underlying data are different. As I explain further in reference to Leskovec (*see* §VIII.B), when min-hashing documents, a set of “shingles” is created from the document, min-hash signatures are computed from

¹ Starner writes “MinHash;” Leskovec writes “minhash;” still other references use “min-hash.” A POSITA would have understood that these terms (and other syntactic variants that might exist) all refer to the same hashing method.

the shingles, and a similarity is computed. EX1007, 91. When min-hashing images, “visual words” are created from the images, and the “hash function is applied to all visual words in an image.” EX1037, 1. In another example, Chum-2 describes his adaptation of min-hash to images by first explaining min-hash “using textual words, and then explain[ing] the adaptation to visual words.” EX1040, 4; *see also*, EX1039. Chum notes that “a bag of visual words with *tf-idf* (term frequency – inverse document frequency) weighting [] has proven to be a very successful approach for image and particular object retrieval, even on large corpora.” EX1039, 2.

B. Policy-based Management

35. As Westerinen explains, a policy can be defined as a “definite goal, course or method of action to guide and determine present and future decisions.” EX1035, 7. Policies can be expressed as a set of rules to administer, manage, and control access to resources. *Id.* Westerinen further notes that “individual rules may be defined in support of business goals.” EX1035, 8.

36. By 2017, policy-based management was widely used to configure and govern the operation of computer systems. In this section, I provide a broad overview of policy-based management but note that a comprehensive overview of policy is beyond the scope of my declaration.

1. Policy Examples

37. In this section, I provide examples of how policy was before the Critical Date, but given the extensive use of policy in computing systems, these examples are neither comprehensive nor limiting. Additionally, the categories listed below are non-exclusive. As one simple example, a system configuration policy can relate to access control.

(a) System Configuration

38. Policy is frequently used by systems administrators to specify what can and cannot be done on and to a computer system, when certain actions must be taken, and constraints of configurations. One common example is password length. For example, Shay describes a “Password Policy Change” in which “passwords would be required to contain at least eight characters, and include at least one uppercase letter, one lowercase letter, one digit, and one symbol.” EX1054, 2. Nicodemus provides password-related examples, including the “[p]assword reset age or date” and “[p]assword expiration age or date.” EX1043, [0440]-[0441].

39. Nicodemus further lists many examples of such configuration-related policies, including “[p]ermitted applications per named user, [p]ermitted application versions per named user, [p]ermitted applications for a specified endpoint hardware configuration, [p]ermitted transactions per named application per named user, [and] [p]rerequisites for a named application to be permitted to run,” among many others. EX1043, [0445]-[0450], *see also, e.g.*, [0451]-[0467]. Nicodemus further enumerates

data backup policies, including the “[m]aximum number of days, or hours between data backups for incremental backups” and the “[m]aximum number of days, or hours between data backups for full backups.” EX1043, [0484]-[0501].

(b) Access Control

40. Damianou explains that “[a]ccess control is concerned with limiting the activity of legitimate users who have been successfully authenticated.” EX1044, 2.

41. Damianou describes categories of access control policies, including:

- Authorization² policies: “define what activities a member of the subject domain can perform on the set of objects in the target domain,” which “protect resources and services from unauthorized access.” EX1044, 2.
- Information Filtering Policies: “transform the information input or output parameters in an action, e.g., “location service might only permit access to detailed location information, such as a person is in a specific room, to users within the department.” EX1044, 3.
- Delegation Policies: constrain “the ability of a user to delegate access rights to another” user. EX1044, 3-4. “A delegation policy permits subjects to grant privileges, which they possess (due to an existing authorisation policy, to grantees to perform an action on their behalf *e.g.* passing read rights to a printer spooler in order to print a file.” EX1044, 4.

² Damianou uses the spelling “authorisation.” EX1044, 2-3. I adopt the American spelling here.

- Refrain policies: “define the actions that subjects must refrain from performing (must not perform) on target objects even though they may actually be permitted to perform the action.” EX1044, 4.

42. Nicodemus also lists examples of authentication and data access policies. EX1043, [0439]-[0444], [0468]-[0483].

(c) Differentiates Resource Use

43. Another challenge faced by computer programmers is finding ways to address the needs of mobile users using the resource-limited devices and networks. One approach was to use policies, often expressed as a set of rules (as explained below), to manage competing demand for resources. Solutions to these problems were developed by adapting known techniques in computer systems and networks (*e.g.*, policy-based priority queuing), as well as developing new, faster devices and networks.

44. Another use of policy was to differentiate resource usage in various environments. For example, a policy might specify that data from interactive applications, such as multi-player games, are assigned the highest priority to ensure the best user experience. These policies were implemented to manage and allocate computing resources efficiently in diverse computing environments. For instance, differentiated services emerged as a crucial policy in this context, especially as mobile devices such as phones began to traverse heterogeneous environments,

transitioning between Wi-Fi and cellular networks seamlessly. By employing policies like differentiated services, computing systems could optimize resource allocation, ensuring a smooth and reliable user experience while adapting to changing network conditions and device capabilities in these dynamic environments. Such techniques were particularly important to address the challenges faced by early mobile devices.

2. Policy Specification

45. Dursun explains that “[t]here are two main approaches [to] policy specification and implementation issue; implementing policies as a piece of code and interpreting them as objects.” EX1022, 11. Dursun further explains that “*Policy as code*” is more flexible using a standard programming language” (emphasis in the original). *Id.*

46. A policy can also be expressed as a “collection of rules.” EX1045, [0058]; EX1042, [0006]; EX1047, [0031]; EX1048, [0007]. Westerinen similarly explains that a policy can be defined as “a set of rules to administer, manage, and control access to network resources.” EX1035, 6.

47. In one specific example, the eXtensible Access Control Markup Language (XACML) is “a standard access control language” that is “defined using XML.” EX1049, 2. In XACML, policies “consist of an arbitrary tree of sub-policies,” where each “tree represents a target, while the leaves of the tree contain a

collection of rules.” EX1049, 2. In the XML specification, the “<policy> element contains a set of <rule> elements.” EX1055, 13.

48. There are many alternatives for specifying such rules, but one common approach is to specify a rule as a condition and corresponding action. For example, Verma describes the “form that typical policy rules” take, *e.g.*, as an expression of the form “if *condition* then *action*” (emphasis added) where “the conditional part describes a specific situation that can be encountered by a network device, and the action component specifies the action that is to be performed by the device in that situation.” EX1050, 2. Lobo extends condition-action to “event-condition-action” such that “[a]n event-condition-action rule is triggered when the event occurs, and if the condition is true the action is executed.” EX1051, 1. For simplicity, I describe the condition-action example, but a POSITA would have understood that event-condition-action and other forms were known and could have been used.

49. In this construction, the two aspects of a policy rule, a condition aspect and an action aspect, influence how the system under management operates. The condition is typically a Boolean expression that compares some measurable state of the computing system to a value or to another measurable state. Rao suggests typical examples such as “if the application name is ‘App1’,” “if the application type is VoIP,” and “if the application is running in the Background,” although a POSITA would have recognized that virtually any condition could be used. For example,

provisioning systems might specify conditions related to CPU utilization, caching systems might specify conditions relates to cache hit rate, a content distribution network (CDN) might measure popularity of a piece of content, an access control system might define access levels based on a user's ID and/or role, and so on.

50. Conditions can be simple Boolean expressions (*e.g.*, “if X equals Y”) or more complex expressions that include multiple clauses (*e.g.*, “if ((X equals Y) OR (A is less than B))”). Stone gives an example, in which:

```
{traffic_class = {video, voice}, used_bw <= allotted_bw}  
  
{allotted_bw = 50M, loss_rate (data) > 40% : allotted_bw := 40M}
```

EX1052, 10.

51. In this example, “[v]oice and video traffic are provided with an allotted bandwidth of 50 Mb/s, but when the network loss rate is greater than 40 percent, a compromise will be made to lower the allotted bandwidth to 40 Mb/s.” *Id.* (Lowering the allotted bandwidth is an action taken when the condition is true.)

52. Gonçalves similarly describe conditions that are “ORed or ANDed.” EX1053, 4. Further, CIM-SPL defined conditions as Boolean expressions of arbitrary complexity. EX1053, 5. In summary, Boolean conditions including multiple clauses were well-known as of the Critical Date.

53. The action aspect determines how the system reacts when the condition evaluates to TRUE. Verma teaches that an “action may be construed as a procedure

that is defined and stored in the repository (*e.g.* a piece of Java code, or a program in a language like CacheL or PONDER)” or a “set of simple operations that need to be performed.” EX1050, 2. Such reactions depend both on the type of a system and its goals. For example, when a provisioning system determines that CPU utilization of a system exceeds a threshold, it might provision a new server, providing additional computing capacity; a CDN might replicate a piece of content to more distribution nodes.

54. The policy rules can be specified using many different syntaxes. For example, Stone provides an overview of policy languages. *See* EX1052, *generally*. Some hardware and software vendors often defined their own proprietary policy-specification languages.

VII. THE '982 PATENT

A. Overview of the '982 Patent

55. The '982 Patent generally describes “an authentication method and an authentication device.” EX1001, 1:11-12. The disclosed authentication method involves a client device capturing and hashing context data for use in authenticating a current context (or environment) of the client device under survey. EX1001, Abstract, 5:14-18. The client sends the hash to a server that acts “as a second (authentication) device” and “manages a set of context accounts and respective reference (context) hashes that have been previously respectively registered while

using the same invention authentication process.” EX1001, 2:4-5, 5:30-34. The specification explains that sending a hash “reduces the number of bits (or elements of information) to be transferred with respect to the original context data element(s),” and therefore obviates the “need [for] a large bandwidth to transmit data[] ... relating to the concerned environment or context, since the [hashed] data relating to the concerned context includes only a few data (up to a few kbytes) to be transmitted.” EX1001, 2:65-3:3.

56. The server then determines a “hash distance” between the hash received from the client and the stored hashes. EX1001, 2:5-8. The “hash distance may be generated through a hash type comparison algorithm.” EX1001, 2:44-45. The hash distance allows the system to “determine whether the generated hash is more or less close to a predefined reference hash,” which “allows inferring whether the corresponding captured context data element(s) is(are) or is(are) not successfully authenticated, *i.e.* sufficiently close to respective predetermined reference captured context data element(s).” EX1001, 2:36-43. Using a hash distance further allows the system “to be resilient to more or less small changes with respect to the predefined reference hash while still authenticating successfully the concerned (local) environment or context, *i.e.* the value of at least some captured (or collected) context data.” EX1001, 2:50-54.

57. The '982 Patent explains that “[e]ach context data element is preferably a string of bits (or units of information) with an individual fixed length.” EX1001, 6:51-54. “To capture the context data element(s), the invention does not impose any constraint,” for example, “[t]he client device may use a camera(s), a microphone(s) and/or one or several sensors that is(are) incorporated within or connected to the client device.” EX1001, 2:13-17. The '982 Patent provides one example in which context data includes “image data relating to the user face and image data relating to the TV 13, other data, such as a phone ID, a Service Set Identifier (or SSID), an OS version ID, geolocation data (like *e.g.*, Global Positioning System (or GPS) data), a user 11 heart rate (not represented) and/or other data.” EX1001, 6:64-7:2.

58. The '982 Patent further explains that the “phone 14 is adapted to constitute, based on the collected context data element(s), a (resulting) data packet by *e.g.* concatenating the collected context data element(s).” EX1001, 8:18-20. In some examples, “[t]he data packet is a string of bits (or units of information) with a resulting length that may vary depending on the collected context data element length(s) between separate context authentication sessions,” and serves as “a kind of passphrase comprising, as word(s), the collected context data element(s).” EX1001, 8:21-26.

59. The '982 Patent additionally explains that “[t]he authentication policy comprises a set of identified context data elements to be collected, so as to define

context data element(s) to be collected.” EX1001, 6:60-62. Further, it can define “the number and/or the nature of the context data element(s) to be captured”. EX1001, 2:9-11. Additionally, an “authentication policy may comprise data for identifying a user authentication to be used for authenticating the user 11, when applicable, *i.e.* when the user authentication data is to be included in the context data element(s).” EX1001, 7:51-54. “For example, the user 11 has to authenticate by providing a PIN, an OTP, a password and/or a biometric user feature(s), such as a one or several user fingerprints, a user iris, a user face and/or a user palm(s).” EX1001, 7:54-59.

60. In one example, a “phone 14 is arranged to collect, based on a predetermined authentication policy, an image (or picture) (and/or a video signal) including image data relating to the user 11 face and image data relating to the TV 13, as data elements included in the context (relating to the phone 14).” EX1001, 6:43-47. FIG. 2 depicts “an example of a message flow 20 that involves the phone 14 and the server 16, so that the server 16 authenticates the context 12 based on a hash distance between a (received) generated hash and a (registered) predetermined reference hash” (EX1001, 11:19-23):

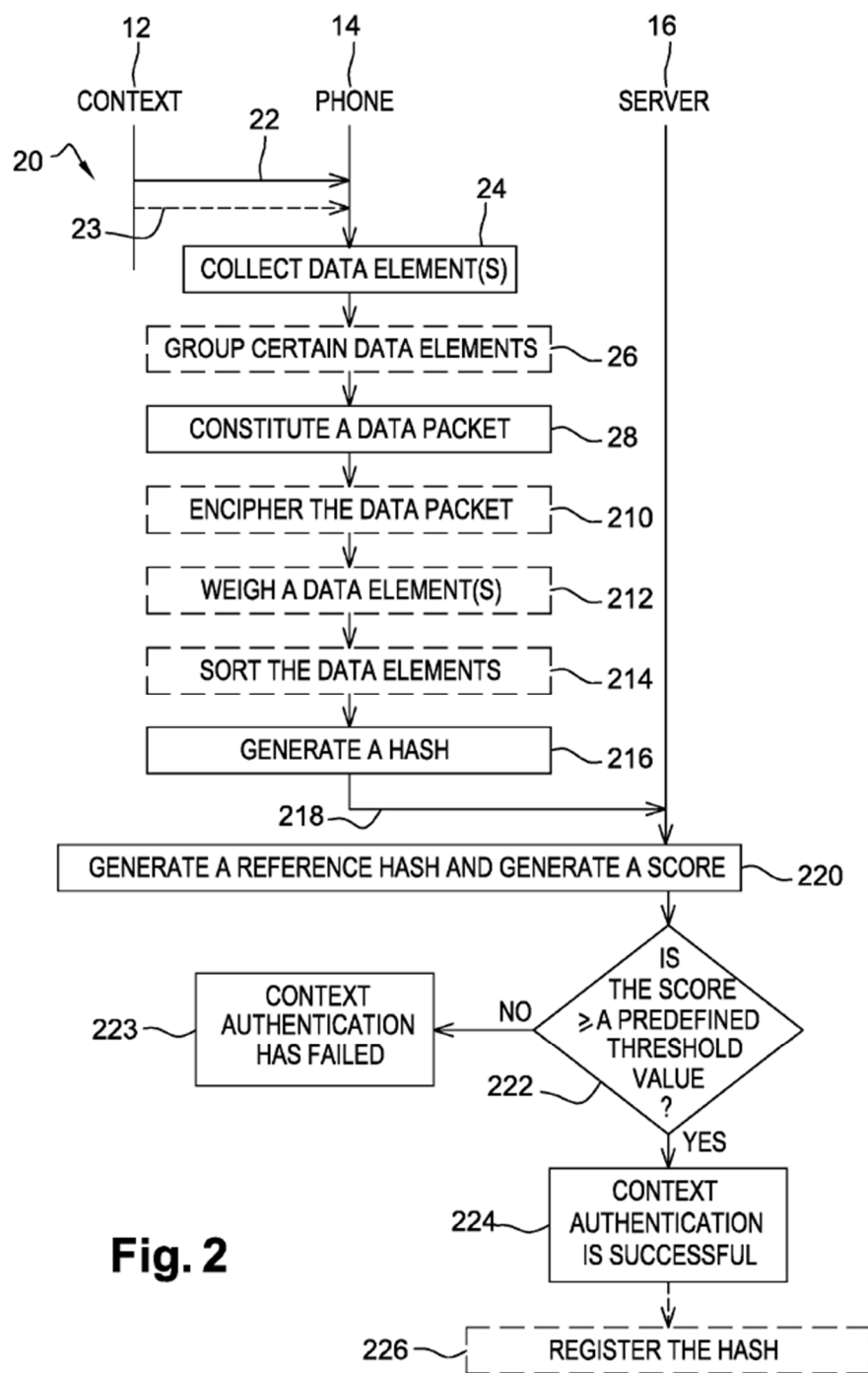


Fig. 2

'982 Patent, FIG. 2

B. Prosecution History of the '982 Patent

61. The Examiner issued just a single Office Action during the course of examination of the '982 Patent, which rejected the pending claims under § 103 over combinations of prior art including Lawrence (EX1028), Cheng, (EX1029), Aura (EX1030), Sakamoto (EX1031), Sharifi (EX1032), Chul-min (EX1033), and Bisikalo (EX1034). *See* EX1002, 201-218.

62. The Applicant submitted no claim amendments in response to the Office Action. *See* EX1002, 271. The Applicant instead argued that the examined claims already distinguished the cited prior art because the “idea underlying the invention is to avoid the bandwidth-related problems found in the prior art that, once the user captures the context data as defined by the authentication policy, he/she needs to send that entire and private information to the authentication server for its subsequent authentication.” EX1002, 271. The Applicant explained that bandwidth and matching issues arise in “authentication policies [that] require[e] environment or context input” because, “unlike usernames and passwords,” environmental or context-based authentication data is “not fixed and may slightly differ in an acceptable manner,” and therefore, “if the protocol requires simply hashing the data packet before being sent, to save bandwidth while avoiding disclosing private data, the data originally captured should be identical to the template stored in the server, which is statistically impossible.” EX1002, 271. Thus, “the present invention

dispense[s] with uploading the entire captured signal of the environment that needs several Mb while, second, still allows the server to authenticate the received data even if [it] slightly differs in an acceptable manner with the stored template.” EX1002, 271. In this context, the Applicant asserted that “none of the cited documents either recites or suggests inter alia the steps of: i. generating, as a hash distance generation step, a hash distance between the generated hash and a predetermined reference hash, and ii. authenticating successfully or not based on the generated hash distance, as an authentication step.” EX1002, 278.

63. Following these remarks, the Examiner and the Applicant agreed to an Examiner’s amendment in which limitations [1.6.2] (“wherein said at least one context data element is a captured image represented as a string of bits with a fixed length for constituting said data packet”) and [1.6.3] (“wherein said hash is a byte stream generated from said string of bits representing said captured image for said data packet”) were added to independent claims 16, 23 and 25³. EX1002, 306-309. The Examiner identified limitation [1.6.2] “in combination with other limitations” as the basis for allowance of the ’982 Patent, even though independent claim 24 was

³ Independent claims 16, 23, and 25 of the application would issue as claims 1, 8, 10 of the ’982 Patent. *See* EX1001, Claims 1, 8, 10.

never amended to include this limitation.⁴ *Id.* More, as demonstrated below by Grounds 1A-1D of this Petition, the limitations added by Examiner’s amendment were well known in the prior art before the Critical Date, and indeed the Challenged Claims as a whole were obvious. The Challenged Claims never should have been allowed and the ’982 Patent never should have issued.

VIII. OVERVIEW OF PRIOR ART REFERENCES

A. Overview of Starner⁵ (EX1004)

64. Like the ’982 Patent, Starner describes “biometric authentication” techniques. EX1004, Abstract. In Starner, “biometric data may be used to authenticate a user in lieu of a password, *e.g.*, so that a user’s HMD [head-mounted display] can log the user on to a webpage on their laptop.” EX1004, [0022]. In one example, “a wearer of a head-mountable device (HMD) wants to access secured information on a second computing device, such as via a password-protected webpage displayed on the wearer’s laptop.” EX1004, [0021]. In this example, “the

⁴ Independent claim 24 of the application would issue as claim 9 of the ’982 Patent. *See* EX1001, Claim 9.

⁵ The general descriptions of this and other references and combinations are incorporated into each subsection and mapping of the claims that includes citations to these references. All emphasis is added unless otherwise indicated.

HMD may detect a quick response (QR) code included on the login webpage that is displayed by the laptop,” and “[i]n response, the HMD may obtain biometric data identifying the wearer.” EX1004, [0021]. “The biometric data may then be sent to a web server that is associated with the QR code,” and the “web server can then use the biometric data to identify the particular wearer and authorize access to secure content on the laptop, *e.g.*, by logging the wearer into the webpage on the laptop.” EX1004, [0021]. FIG. 5B illustrates a user wearing an HMD configured to capture a QR code 502 and obtain biometric data from the user that can be used to authenticate the user and login to the website:

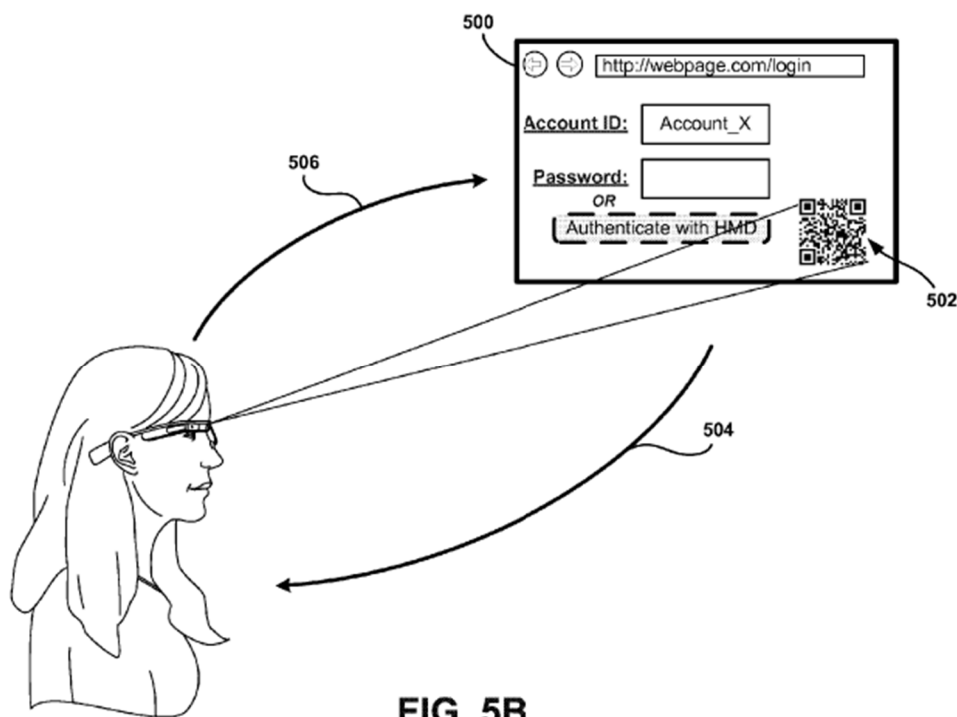


FIG. 5B

Starner, FIG. 5B

See also EX1004, [0112]-[0125], Figs. 5A-5F.

65. In Starner, “[v]arious types of biometric data may be utilized to verify a user's identity and authorize access to secure content.” EX1025, [0025]. To perform the biometric authentication, the “HMD acquires biometric data via one or more biometric sensors,” which “may include eye-scan data received from one or more sensors coupled to the HMD.” EX1004, [0076]. Additionally, “[o]ther forms of biometric data, such as fingerprint data, hand/finger geometry, vein geometry, and voiceprints, among other possibilities, may be utilized in addition or in the alternative to eye data.” EX1004, [0025].

66. Starner further explains that “the HMD can continuously verify the wearer during use.” EX1004, [0096]. During “normal use of the HMD,” the HMD can scan the wearer’s eye and “compare [the scan] with the authenticated wearer.” *Id.* “If at some point during use the HMD [] cannot find a good match for the sections of the wearer's eye, the HMD can begin operating in a locked mode.” *Id.*

67. “The HMD may use the data captured by the one or more sensors to create, select and/or digitize an image of the eye.” EX1004, [0077]. Starner further explains that “[m]ultiple scans may provide a more detailed digital template to the HMD since the sensors will have more time to capture individual images of a user's eye.” EX1004, [0062]. The “HMD may analyze the images to identify patterns in an iris, and convert the patterns into a digital template” and “may identify patterns of light and dark image data that correspond the dark pupil and the sclera (*e.g.*, the

white of the eye) to create the digital template.” EX1004, [0077]. For example, “the digital template may include a 1024-byte record (32×32×1 byte grayscale or color).” EX1004, [0077].

68. Starner describes two approaches for securely authenticating a user using biometric data: using hashed biometric data and using encrypted biometric data. EX1004, [0023], [0078], [0080]-[0088], [0090]-[0092].

69. When a secure website requires hashed biometric data, Starner explains that “a digital signature of the eye-scan data (and/or other biometric data) may be generated by the HMD (or other computing device) from the eye-scan data and used for authorization rather than the actual images of the wearer’s eye(s).” EX1004, [0078]. The digital signature can be generated using a hash algorithm, *e.g.*, “a locality-sensitive hash (LSH) function, such as MinHash, may be used by the HMD to generate the digital signature from the eye-scan data.” EX1004, [0078]. “The digital signature may be unique to the wearer of the HMD, such that unauthorized parties cannot obtain (*e.g.*, reconstruct) the images of the wearer’s eye(s) from the digital signature.” EX1004, [0078]. Further, “the actual biometric data may not ever need to be stored locally at the HMD or remotely at the authentication server (or other server).” EX1004, [0078]. Thus, Starner’s solution “provid[es] additional security for the wearer’s biometric data.” EX1004, [0078]. Additionally, “[b]y using an LSH function to generate an initial digital signature for eye-scan data, the same

digital signature can then be generated from any image of the wearer's eye(s) that substantially matches the initial image,” and “even though there may be variations of the captured images of the wearer's eye(s) (*e.g.*, lighting and eye position), the LSH function allows for the same digital signature to be generated from each of the various images.” EX1004, [0092].

70. Alternatively, when a secure website requires hashed biometric data, “[t]he HMD can encrypt the biometric data (and/or other information needed for authentication) with its private key, send the encrypted data to a server, and the server can decrypt the data using the public key of the HMD,” and Starner notes that “[o]ther cryptographic methods and additional security measures are possible as well.” EX1004, [0023]

71. In Starner, a single eye scan can be used to authenticate with multiple secure websites. For example, Starner explains that the “presence of other security measures may render repeated eye-scanning unnecessary, and only an initial eye-scan may be needed from the owner of the HMD.” EX1004, [0094]. Alternatively, “the HMD may re-scan a wearer's eye and/or re-determine biometric data of the wearer each time a challenge request or other such request for authorization is determined based on the HMD detecting the machine-readable code.” EX1004, [0094].

72. To enable biometric matching at a server, in Starner, “an enrollment process may be implemented to establish the predetermined stored biometric data, such as stored eye-scan data.” EX1004, [0089]. “The enrollment process may involve a similar, but more detailed scan of the eye as discussed above,” *e.g.*, “[o]ne or more sensors may be configured to capture several individual digital still images of a wearer's eye during the enrollment process. The HMD (and/or other computing device) may then analyze the patterns in the eye and convert them into a stored digital template.” EX1004, [0089]. “The stored digital template may be used as stored eye-scan data, which can be matched to a later eye scan to identify the particular wearer of the HMD.” EX1004, [0089]. In an example embodiment, “stored eye-scan data could take the form of a 1024-byte data record.” EX1004, [0089].

73. “Once the biometric data (*e.g.*, eye-scan data) is stored at the authentication server, the authentication server (or other computing device, such as the HMD) may compare the received data and corresponding digital template with the stored data to determine if there is a match.” EX1004, [0090]; *see also id.*, [0078], [0092].

74. In some embodiments, “the biometric data is eye-scan data, the actual eye-scan image (*e.g.*, iris print image) [and] may not be encrypted, but rather at least one copy of the eye-scan image of the same resolution, lower resolution, and/or

higher resolution.” EX1004, [0081]. Alternatively, the “authentication information may comprise encrypted biometric data, such as encrypted eye-scan data, generated by the HMD based on the acquired biometric data.” EX1004, [0081]. In such cases, the “authentication server may decrypt the encrypted biometric data and other authentication information using the public key of the HMD” and “compare the stored biometric data to the decrypted biometric data.” EX1004, [0088].

75. In summary, Starner explains that a computing device such as an HMD “detects a machine-readable code,” *e.g.*, a QR code. EX1004, [0005], [0021]. The “machine-readable code identifies protected information that is viewable via the second computing device”, *e.g.*, a web server hosting a protected web page. EX1004, [0005], [0021]. The “computing device acquires biometric data via one or more biometric sensors,” either generates “a digital signature of the eye-scan data (and/or other biometric data),” using, *e.g.*, “a locality-sensitive hash (LSH) function, such as MinHash,” or encrypted biometric data. Starner’s HMD then “generates an authentication message based at least in part on the biometric data,” and “sends the authentication message to an authentication server for verification of the authentication information.” EX1004, [0005], [0023], [0078]. The “authentication server (or other computing device, such as the HMD) may compare the received data and corresponding digital template with the stored data to determine if there is a match.” EX1004, [0090].

B. Overview of Leskovec (EX1007)

76. Leskovec is an academic textbook that describes additional detail about algorithms like those referenced in Starner for data mining and matching digital signatures, including locality-sensitive hash (“LSH”) functions such as MinHash. *See generally* EX1007, ix-x, 73-128.

77. For example, Leskovec explains that “minhashing” “compresses large sets in such a way that we can still deduce the similarity of the underlying sets from their compressed versions.” EX1007, 68. Since “there may be far too many pairs of items to test each pair for their degree of similarity,” Leskovec explains that “a technique called ‘locality-sensitive hashing’” focuses the search “on pairs that are most likely to be similar.” EX1007, 68. As I explained in §VIII.B, Min-hash is an example of a “Locality Sensitive Hashing scheme [] that approximates similarity between sets.” EX1037, 3.

78. Leskovec explains that, when comparing large amounts of information⁶, it is often useful “to replace large sets by much smaller representations called

⁶ While Leskovec describes examples that involve comparing document signatures using min-hash, long before the Critical Date, min-hash was also used for a variety of purposes including biometric authentication and to compare image

‘signatures,’” such that signatures can be compared and “similarity of the underlying sets [estimated] from the signatures alone.” EX1007, 75. “It is not possible that the signatures give the exact similarity of the sets they represent, but the estimates they provide are close.” EX1007, 75. “For example, if we replace [] 200,000-byte hashed-shingle sets that derive from 50,000-byte documents by signatures of 1000 bytes, we can usually get within a few percent.” EX1007, 75. Accordingly, “we can use minhashing to compress large documents into small signatures and preserve the expected similarity of any pair of documents.” EX1007, 82. Since minhash preserves similarity, minhash (and other locality-sensitive hashes (LSH)) are used to efficiently compare minhash signatures. EX1002, 82-86.

79. Leskovec explains that a “signature matrix” can be formed from a matrix M representing the data such that the “ i th column of M is replaced by the minhash signature for (the set of) the i th column.” EX1007, 78. Leskovec further explains that the “minhash signature” for a set is the “vector $[h_1(S), h_2(S), \dots, h_n(S)]$ ”, where $h_1, h_2, \dots, h_n(S)$ are permutations that determine the minhash function. EX1007, 78. Further this list of hash values is normally represented as a column. EX1007, 78; *see also, e.g.*, EX1007, 79-80; 83-84.

signatures, as I explained in §VI.A. *Supra*, §VI.A; EX1007, 107-108; EX1037, 1; EX1040, 4; EX1039, *generally*.

80. Leskovec further explains that “[e]ven though we can use minhashing to compress large documents into small signatures and preserve the expected similarity of any pair of documents, it still may be impossible to find the pairs with greatest similarity efficiently.” EX1007, 82. “[E]ven if there are not too many documents,” “the number of pairs of documents may be too large.” EX1007, 82. “However, often we want only the most similar pairs or all pairs that are above some lower bound in similarity,” so “we need to focus our attention only on pairs that are likely to be similar, without investigating every pair.” EX1007, 82. “[*L*]ocality-sensitive hashing (LSH)” provides such focus. EX1007, 82.

81. One “approach to LSH is to “hash” items several times, in such a way that similar items are more likely to be hashed to the same bucket than dissimilar items are,” “then consider any pair that hashed to the same bucket for any of the hashings to be a candidate pair,” “check[ing] only the candidate pairs for similarity.” EX1007, 82. When using “minhash signatures for the items, an effective way to choose the hashings is to divide [a] signature matrix into b bands consisting of r rows each.” EX1007, 82.

82. Leskovec provides “an approach to finding the set of candidate pairs for similar documents and then discovering the truly similar documents among them.” EX1007, 85-86.

1. Pick a value of k and construct from each document the set of k -shingles.

Optionally, hash the k -shingles to shorter bucket numbers.

2. Sort the document-shingle pairs to order them by shingle.

3. Pick a length n for the minhash signatures. Feed the sorted list to the algorithm of Section 3.3.5⁷ to compute the minhash signatures for all the documents.

4. Choose a threshold t that defines how similar documents have to be in order for them to be regarded as a desired “similar pair.” Pick a number of bands b and a number of rows r such that $br = n$, and the threshold t is approximately $(1/b)^{1/r}$. If avoidance of false negatives is important, you may wish to select b and r to produce a threshold lower

⁷ Leskovec’s §3.3.5 describes “Computing Minhash Signatures.” EX1007,

than t ; if speed is important and you wish to limit false positives, select b and r to produce a higher threshold.

5. Construct candidate pairs by applying the LSH technique of Section 3.4.1.⁸

6. Examine each candidate pair's signatures and determine whether the fraction of components in which they agree is at least t .

7. Optionally, if the signatures are sufficiently similar, go to the documents themselves and check that they are truly similar, rather than documents that, by luck, had similar signatures.

C. Overview of Shaashua (EX1005)

83. Like Starner and the '982 Patent, Shaashua describes biometric authentication techniques. Shaashua explains that “[b]iometric authentication refers to identification of human beings by their individual characteristics or traits,” including “facial recognition, retina recognition, fingerprint recognition, handwriting recognition, and voice recognition.” EX1005, 10:7-11. Such authentication can be required “prior to allowing the users 30 to access protected resources such as sensitive files and other content, email, VPN access, transactions, games, etc.” EX1005, 5:31-34.

⁸ Leskovec's §3.4.1 describes “LSH for Minhash Signatures.” EX1007, 82-

84. “During biometric authentication, current biometric measurements (*e.g.*, current facial dimensions derived from a current facial image) are compared with expected biometric measurements (*e.g.*, expected facial dimensions derived from one or more previously captured facial images).” EX1005, 1:12-16. “If the current biometric measurements match the expected biometric measurements, authentication is considered successful,” but “if the current biometric measurements do not match the expected biometric measurements, authentication is considered unsuccessful.” EX1005, 1:16-21.

85. Shaashua further explains that when a user attempts biometric authentication, “surrounding conditions such as noise level, crowdedness, light levels, movement, etc. can produce false negative authentication results,” so “authentication for the legitimate person may be burdensome (*i.e.*, the legitimate person needs to retry authentication) as well as frustrating (*i.e.*, the legitimate person may be prevented from accessing an important but protected resource).” EX1005, 1:58-64. For example, if “a human tries to speak into a microphone for voice recognition but the current environment is very noisy,” “although the human may be authentic (*i.e.*, the actual legitimate person so that authentication should succeed), authentication may fail due to the noisy environment.” EX1005, 1:40-45.

86. To address these limitations, in Shaashua, authentication “involves selecting among different biometric methods dynamically.” EX1005, 2:3-4. “For

example, although an arsenal of biometric methods may be available for use, poorly suited biometric methods of the arsenal can effectively be ruled out (*i.e.*, deemed unavailable for use),” and “the best suited biometric method can be selected based on a variety of factors including device factors (*i.e.*, what circuitry is available to capture biometric data), environmental factors (*i.e.*, the states of ambient conditions), and even biometric method strength factors (*i.e.*, weights defining particular security strengths of the available biometric methods).” EX1005, 2:3-13. Shaashua’s “techniques improve the user's experience by lowering the false negative rate (*i.e.*, users do not fail authentication easily due to unsuitable environment or device factors).” EX1005, 2:13-16.

87. In Shaashua, the “[d]etermination of which biometric methods from the set of suitable biometric methods are to be used (*i.e.*, which of the above-described arrangements should be used) can be dictated by a predefined policy.” EX1005, 9:2-6, 11:1-7. In some “arrangements, the authentication server 24 applies the biometric method of the set of suitable biometric methods having the highest suitability score,” which “may be appropriate in situations in which the users 30 should not be overburdened during biometric collection (*e.g.*, it could be too burdensome to ask a user 30 to concurrently provide a facial image, a voice sample, a fingerprint scan, and a handwriting sample all in the same authentication attempt).” EX1005, 8:52-60.

88. In Shaashua, “upon receipt of an authentication request,” an authentication server running “the authentication application” “obtains a set of suitability factors from a smart device 22 of a user 30.” EX1005, 6:66-7:2. “The set of suitability factors include, among other things, device factors, environmental factors, and biometric method strength factors,” and “may be acquired by the authentication server 24 during a handshaking exchange with the smart device 22 (directly or perhaps through another device 26)” or “may have been included with the authentication request.” EX1005, 7:2-10. The device factors further “identify the capabilities of the smart device” “and thus which of the various available biometric methods provisioned in the authentication server 24 can be applied to authenticate the user.” 7:10-14.

89. The authentication server next “selects, from the different available biometric methods” “a set of suitable biometric methods to apply during user authentication” by “(i) generat[ing] a suitability score for each available biometric method: facial recognition, voice recognition, handwriting analysis, etc., and (ii) compar[ing] the suitability score to a predefined threshold score.” EX1005, 8:10-19. The authentication server then selects biometric methods only for which “the suitability score for that biometric method is at least as high as the predefined threshold score.” EX1005, 8:19-30. Then, the authentication server “obtains

biometric authentication factors from the user 30 via the smart device 22 and derives current user data from the biometric authentication factors.” EX1005, 8:38-41.

90. In some arrangements, “the authentication server 24 applies the biometric method of the set of suitable biometric methods having the highest suitability score,” which “may be appropriate in situations in which the users 30 should not be overburdened during biometric collection (*e.g.*, it could be too burdensome to ask a user 30 to concurrently provide a facial image, a voice sample, a fingerprint scan, and a handwriting sample all in the same authentication attempt).” EX1005, 8:53-60. “If the current user data matches expected user data [] within a predefined set of tolerances (*e.g.*, 3% accuracy, 5% accuracy, etc. depending on the particular biometric method(s) being applied), authentication is successful,” but “if the current user data does not match the expected user data within the predefined set of tolerances, authentication is unsuccessful.” EX1005, 8:41-47.

D. Overview of Bowman (EX1009)

91. Bowman describes methods “to encrypt and decrypt data” on an “electronic device.” EX1009, Abstract. An electronic device can be a “wearable electronic device,” “personal data assistant (PDA), smart phone,” etc. EX1009, [0029].

92. In Bowman, “an electronic device can be considered to be in a locked state when the device is powered on but most functionalities are unavailable for a

user to use” and “in an unlocked state when the device is powered on and the device functionalities are available for the user to use.” EX1009, [0011]. When locked, “user authentication may be required to access these functionalities when the device is in the locked state,” and “[i]f the user authentication succeeds, the device can transition to an unlocked state.” *Id.* The “electronic device can encrypt [] data received or generated during the locked state and store the encrypted data in a secured storage on the electronic device.” EX1009, [0012]. The “encrypted data cannot be decrypted by entities other than the holder of a master private key that is associated with a public key used to encrypt the data.” EX1009, [0012].

93. Bowman’s method begins when “data to be encrypted is received.” EX1009, [0031]. The “data can be any type of data, e.g., a message received over a network or a file stored on the electronic device” and the data format can be “plaintext, image, audio, video, or any other formats.” EX1009, [0031].

94. To encrypt the data, “an ephemeral key pair is generated,” and the ephemeral key pair can include an ephemeral public key and an ephemeral private key.” EX1009, [0033]. Then, “a shared secret is generated ... based on a master public key and the ephemeral private key” and “a current secret key is generated based on the shared secret.” EX1009, [0034], [0037]. A “hash, a key derivation function (KDF), or any other cryptographic one-way function can be applied to the shared secret to derive the current secret key.” EX1009, [0037]. “For example, a

one-way function such as the [NIST] KDF Alternative 1 using SHA-512 can be used to generate the current secret key.” EX1009, [0037]; *see also id.*, [0014], [0038], [0040].

95. The “current secret key can be used to encrypt the current portion of the data.” EX1009, [0038]. Further, “a current initialization vector (IV), or other current encryption information can also be generated based on the shared secret using a hash, a KDF, or any other cryptographic one-way function.” EX1009, [0038]. “In some cases, a current authentication key, e.g., a current Message Authentication Code (MAC) key, can also be generated based on the shared secret.” EX1009, [0038].

96. Once the current secret key is derived, “the shared secret is deleted,” which can prevent the encrypted data from being decrypted by entities other than the private key holder.” EX1009, [0039]. Then, “the current portion of data is encrypted with the current secret key and the current IV to generate a current ciphertext portion,” e.g., “using an AES-256 block cipher.” EX1009, [0040].

97. Additionally, “a data integrity algorithm,” such as “a keyed-hash message authentication code (HMAC) algorithm, e.g., the HMAC-SHA-256 algorithm,” can be “applied to the current ciphertext portion to generate a HMAC

tag for the current portion.”⁹ EX1009, [0041]. “The data integrity algorithm can use the current authentication key, the current IV, or a combination thereof to generate the HMAC tag for the current portion,” and the “HMAC tag for each portion can be sent to the entity requesting the encryption,” where the HMAC “tags can be used to authenticate the encrypted data during the decryption process.” EX1009, [0041].

IX. GROUND 1A: CLAIMS 1-3, 5-6, 8-10 ARE OBVIOUS IN VIEW OF THE STARNER-LESKOVEC COMBINATION

A. Combining Starner and Leskovec

1. Overview of the Combination (SLC)

98. Starner explains that “a digital signature of the eye-scan data (and/or other biometric data) may be generated by the HMD (or other computing device) from the eye-scan data and used for authorization rather than the actual images of the wearer’s eye(s), thus providing additional security for the wearer’s biometric data.” EX1004, [0078]; *supra*, §VIII.A. Starner provides an example in which “a locality-sensitive hash (LSH) function, such as MinHash, may be used by the HMD to generate the digital signature from the eye-scan data.” EX1004, [0078]. Starner explains that “[b]y using an LSH function to generate an initial digital signature for eye-scan data, the same digital signature can then be generated from any image of

⁹ HMAC stands for “Hashed Message Authentication Code.” EX1059, 1.

the wearer's eye(s) that substantially matches the initial image,” and “even though there may be variations of the captured images of the wearer's eye(s) (e.g., lighting and eye position), the LSH function allows for the same digital signature to be generated from each of the various images.” EX1004, [0092].

99. Starner assumes that a POSITA would know how to apply, or obtain information about how to apply, a “locality-sensitive hash (LSH) function, such as MinHash,” as Starner provides little detail about the MinHash algorithm itself (or LSH generally). *See* EX1004, [0078]. A POSITA therefore would have turned to references such as Leskovec that describe the details. EX1004, [0078], [0092], *generally*; §VIII.B. It would have been obvious to apply Leskovec's LSH algorithms, including Leskovec's MinHashing algorithm, to Starner as a straightforward way of achieving Starner's goal of utilizing such algorithms in its biometric authentication process.

100. In more detail, per Starner, the Starner-Leskovec Combination's (SLC's) “enrollment process” “establish[es] the predetermined stored biometric data” by “analyz[ing] the patterns in the eye and convert[ing] them into a stored digital template,” which can “take the form of a 1024-byte data record.” EX1004, [0089]. The digital template “may be used as stored eye-scan data, which can be matched to a later eye scan to identify the particular wearer of the HMD.” EX1004, [0089].

101. The stored biometric data are used to authenticate (or not) the user, as explained further below. EX1004, [0077], [0087], [0089]-[0090]. As Starner explains, “the biometric data may be used to authenticate a user in lieu of a password, *e.g.*, so that a user’s HMD can log the user on to a webpage on their laptop,” and “[a]s such, the biometric data can serve as a replacement for passwords entirely.” EX1004, [0022]. Per Starner, the biometric data used for authentication can either be encrypted or hashed, and in SLC, encrypted biometric data are used for continuous authentication. EX1004, [0023], [0078], [0080]-[0088], [0090]-[0092].¹⁰ Further, per Starner, “the HMD can continuously verify the wearer during use.” EX1004, [0096]; §VIII.A.

102. When SLC stores the enrolled biometric data as a digital signature or “fingerprint”¹¹ generated by MinHashing using the techniques of Leskovec, the digital signature can improve security since “the actual images of the wearer’s eye(s)

¹⁰ When using encrypted eye-scan data instead of MinHashed data, comparison uses Hamming distance, *e.g.*, as explained in Starner. EX1004, [0090]-[0091]; EX1007, 91.

¹¹ The term “fingerprint” in this context refers to a unique data signature, not to a human’s fingerprint.

may never need to be stored at the authentication server.” EX1004, [0078], [0092]; EX1007, 85-86.

103. Further, per Starner, SLC’s computing device, “detects a machine-readable code,” *e.g.*, a QR code. EX1004, [0005], [0021]. The “machine-readable code identifies protected information that is viewable via the second computing device,” *e.g.*, a web server hosting a protected web page. EX1004, [0005], [0021]. The “computing device acquires biometric data via one or more biometric sensors.” EX1004, [0005].

104. Based on Starner’s teachings, in SLC, “[v]arious types of biometric data may be utilized to verify a user’s identity and authorize access to secure content.” EX1025, [0025]. To perform the biometric authentication, the “HMD acquires biometric data via one or more biometric sensors,” which “may include eye-scan data received from one or more sensors coupled to the HMD.” EX1004, [0076]. Further, “[m]ultiple scans may provide a more detailed digital template to the HMD since the sensors will have more time to capture individual images of a user’s eye.” EX1004, [0062]. Additionally, “[o]ther forms of biometric data, such as fingerprint data, hand/finger geometry, vein geometry, and voiceprints, among other possibilities, may be utilized in addition or in the alternative to eye data.” EX1004, [0025].

105. Further, per Starner, “a digital signature of the eye-scan data (and/or other biometric data) [is] generated by the HMD (or other computing device) from the eye-scan data and can be used for authorization rather than the actual images of the wearer’s eye(s), thus providing additional security for the wearer’s biometric data.” EX1004, [0078]. Per Starner, the HMD “analyze[s] the patterns in the eye and convert them into a stored digital template.” EX1004, [0089].

106. In SLC, a locality-sensitive hash (LSH) function, MinHash, is used by the HMD to generate the digital signature from the eye-scan data. EX1004, [0078]. Starner explicitly states that “a locality-sensitive hash (LSH) function, such as MinHash, may be used by the HMD to generate the digital signature from the eye-scan data.” EX1004, [0078]. Since Starner’s HMD captures digital images of the eye and applies MinHash to the capture, a POSITA would have understood that Starner is describing the use of MinHash on images. EX1004, [0077]-[0078].

107. The MinHash is determined using the techniques of Leskovec and applied to images. EX1007, 82-86. As I explained in §VIII.B, while Leskovec explains the well-known MinHash algorithm in the context of documents, as of the Critical Date, “min-hash and its variants have been successfully applied to discovering near duplicate images.” EX1037, 1. Chum provides more detail, describing “how we adapt a method originally developed for text near-duplicate detection [2] to near-duplicate detection of images. We describe it using textual

words, and then explain the adaptation to visual words in the following sub-section.”

EX1040, 4-5; *see also* EX1039.

108. Therefore, in the context of images, SLC:

1. Pick[s] a value of k and construct[s] from each [image] the set of k - [visual words].

2. Sort[s] the [visual words] to order them by shingle.

3. Pick[s] a length n for the minhash signatures. Feed[s] the sorted list to the algorithm of Section 3.3.5 [describing “Computing Minhash Signatures”] to compute the minhash signatures for all the [visual words].

EX1007, 75-80; 86-86.

109. In summary, in SLC, a POSITA would have understood that “visual words” replace document shingles, as taught, for example, by Lee. EX1037, 1 (When min-hashing images, “visual words” are created from the images, and the “hash function is applied to all visual words in an image.”).

110. As described in Starner, the SLC device then “generates an authentication message based at least in part on the biometric data,” and “sends the authentication message to an authentication server for verification of the authentication information.” EX1004, [0005].

111. Upon receiving the authentication message, SLC’s authentication server compares “the received data and corresponding digital template with the

stored data to determine if there is a match.” EX1004, [0090]. Per Starner, the enrolled and stored biometric data are encoded by MinHash using the techniques described in Leskovec. EX1004, [0078]; EX1007, 82-86.

112. SLC determines whether the digital template matches the enrolled templates using the techniques of Leskovec. EX1007, 82-86; §VIII.B. SLC “[c]hoose[s] a threshold t that defines how similar documents have to be in order for them to be regarded as a desired ‘similar pair;’ then “[e]xamine[s] each candidate pair’s signatures and determine whether the fraction of components in which they agree is at least t .” EX1007, 85-86.

113. In SLC, per Starner, [b]y using an LSH function to generate an initial digital signature for eye-scan data, the same digital signature can then be generated from any image of the wearer’s eye(s) that substantially matches the initial image,” and “even though there may be variations of the captured images of the wearer’s eye(s) (*e.g.*, lighting and eye position), the LSH function allows for the same digital signature to be generated from each of the various images.” EX1004, [0092]. Further, in SLC, by “[u]sing such a process, the actual images of the wearer’s eye(s) may never need to be stored at the authentication server, unlike processes that utilize calculation of a Hamming Distance.” EX1004, [0092].

2. Motivations to Combine

114. Multiple reasons would have prompted a POSITA to implement SLC based on the combined teachings of Starner and Leskovec as described in the preceding section. *Supra*, §IX.A.1.

115. First, the teachings of Starner and Leskovec are complementary. Starner explains that “a locality-sensitive hash (LSH) function, such as MinHash, may be used by the HMD to generate the digital signature from the eye-scan data,” but Starner’s disclosure omits the implementation details for these functions. EX1004, [0078]. A POSITA therefore would have turned to a reference, such as Leskovec, that describes the operation of LSH and MinHash to achieve Starner’s express goal of allowing similarity comparisons of digital signatures generated by HMD.

116. Second, as Leskovec explains, “minhashing,” “compresses large sets in such a way that we can still deduce the similarity of the underlying sets from their compressed versions.” EX1007, 68. Therefore, a POSITA would have been motivated to combine Starner with Leskovec to achieve the operational efficiency that arises from such compression.

117. Third, a POSITA would have been motivated to use Leskovec’s techniques for creating a digital signature in the form of a unique fingerprint by hashing because hashing was known to compress data and therefore reduce the amount of data required to be transmitted from the device to the authentication server.

As I explain in reference to element [1.4], long before the Critical Date, the benefits of using hashing to compress data, e.g., to reduce network burden, were well-known. §IX.B.1[1.4]; EX1056, 3:66-67, 4:35-36.

118. Fourth, a POSITA would have appreciated that the use of Leskovec's techniques in Starner's device would achieve predictable security improvements. For example, per Starner, SLC computes, transmits and stores "a digital signature for the initial biometric data," "thus providing additional security for the wearer's biometric data." EX1004, [0078]. The "digital signature may be unique to the wearer of the HMD, such that unauthorized parties cannot obtain (*e.g.*, reconstruct) the images of the wearer's eye(s) from the digital signature." EX1004, [0078]. Leskovec describes techniques for computing such digital signatures. EX1007, 86-86. Thus, SLC "[u]sing such a process, the actual images of the wearer's eye(s) may never need to be stored at the authentication server, unlike processes that utilize calculation of a Hamming Distance." EX1004, [0092].

119. Fifth, a POSITA would have pursued the Starner-Leskovec Combination because it merely involves the application of a known technique (*e.g.*, Leskovec's LSH and MinHash techniques) to a known system (*e.g.*, Starner's biometric authentication system) to achieve predictable results. EX1004, [0078]; EX1007, 75-86. Further, a POSITA would have recognized that there was a known problem (*e.g.*, effective biometric authentication), that Leskovec helped address that

issue, and that combining the teachings of Starner and Leskovec was not beyond the skill of a POSITA. Here, Starner raises the problem of how to use “a locality-sensitive hash (LSH) function, such as MinHash,” and Leskovec addresses that problem by providing further information about how to implement well-known LSH functions including MinHash of the type contemplated in Starner.

3. Reasonable Expectation of Success

120. A POSITA would have expected success implementing SLC, which would have required only routine programming knowledge well within the skill of a POSITA. Indeed, implementation of SLC merely involves integration in Starner of common hashing and search/matching functions (*e.g.*, LSH and MinHash) as described in Leskovec where the feasibility of those functions was well established before the earliest possible effective filing date of the '982 Patent. *Supra*, §VIII.A; §VIII.B; §IX.A.1; EX1007, 68. Combining these aspects of Starner and Leskovec would have been well within the skill of a POSITA.

B. Analysis with Respect to Claims 1-3, 5-6, 8-10

1. Claim 1

Element [1pre] An authentication method comprising:

121. To the extent the preamble is limiting, SLC renders [1pre] obvious.

122. For example, Starner describes “[m]ethods, apparatus, and computer-readable media are described herein related to biometric authentication.” EX1004,

Abstract; *see also*, e.g., EX1004, [0005], [0018], [0021]-[0026], [0067]-[0125] (“Example *methods* and scenarios”), claims 1-14.

Element [1.1] collecting, based on a predetermined authentication policy, at least one context data element;

123. SLC renders [1.1] obvious.

124. For example, per Starner, in SLC, “biometric data may be used to authenticate a user in lieu of a password, e.g., so that a user’s HMD [head-mounted display] can log the user on to a webpage on their laptop.” EX1004, [0022]. The “HMD acquires biometric data via one or more biometric sensors,” which “may include eye-scan data received from one or more sensors coupled to the HMD.” EX1004, [0076]; *see also id.*, [0025] (additionally, “[o]ther forms of biometric data ... may be utilized”). The acquired biometric data can include one or more images (*at least one context data element*), e.g., one or more images of the user’s eye(s). EX1004, [0062]-[0066]. In some cases, the images can include images from recorded video. EX1004, [0035], [0118].

125. “The HMD may use the data captured by the one or more sensors to create, select, and/or digitize an image of the eye.” EX1004, [0077]. “Individual images of the eye may be captured by the one or more sensors using a frame grabber during the eye-scan process.” EX1004, [0077]. “Multiple scans may provide a more detailed digital template to the HMD since the sensors will have more time to capture individual images of a user’s eye.” EX1004, [0062]. “[T]he sensors capture images

of the eye(s) over a given period of time.” EX1004, [0064]; *see also id.*, [0118] (“The image(s) can be provided as one or more still images or a video of the wearer’s iris.”).

126. Further, according to Starner, biometric image(s) of the user are a form of “biometric data,” *e.g.*, “eye-scan data” for image(s) of the user’s eyes. EX1004, [0076] (“eye-scan data received from one or more sensors coupled to the HMD”), [0081] (“eye-scan data” includes “eye-scan image”). Starner explains that “stored eye-scan data could take the form of a 1024-byte data record.” EX1004, [0089]. The 1024-byte data record of the biometric/eye-scan data (*e.g.*, an image of the user’s eye) can further be “include[d]” in a “digital template.” EX1004, [0077]. For example, “the digital template may include a 1024-byte record (32x32x1 byte grayscale or color).” *Id.* Thus, like context data elements described in the ’982 Patent, SLC’s biometric data (*e.g.*, eye-scan image) can be represented as a string of bits with a fixed length (*e.g.*, a “1024-byte data record”). *Cf.*, EX1001, 6:51-54, 2:13-17; §VII.A.

127. Further, in SLC, per Starner, the biometric data including eye-scan image(s) (*at least one context data element*) are collected based on a “*predetermined authentication policy*.” The “*predetermined authentication policy*” may be encoded in “machine-readable code” such as a “quick response (QR) code” where the QR code is displayed on a second computing device such that it can be

detected by a camera on the HMD (first computing device) when the second device is in the field of view of the camera. EX1004, [0067]-[0075], FIGS. 3A-3B.

128. To this point, Starner explains that “accessing [] protected information may involve multiple authentication factors, such as multiple types of biometric data, password information, pass-token output, and/or other information,” and “the HMD may detect in the machine-readable code requests to provide such authentication factors.” EX1004, [0074] (emphasis added); *see also id.*, [0072] (“A request for the HMD (or other device) to provide authentication information may also be encoded in the bar code.”), [0073], [0075]. Starner’s “machine-readable code” thus defines an “***authentication policy***” that requests collection of certain authentication factor(s).

129. A POSITA also would have been familiar with many well-known techniques for encoding policies in machine-readable code. For example, as I explained in §X.A, a POSITA would have understood that “[t]here are two main approaches about policy specification and implementation issue; implementing policies as a piece of code and interpreting them as objects,” where “[p]olicy as code is more flexible using a standard programming language.” EX1022, 11. It would have been obvious to implement these in SLC to achieve Starner’s objective of authenticating a user according to information encoded in the machine-readable code (*e.g.*, QR-code).

130. Regarding the term “predetermined,” the Merriam-Webster Advanced Learner’s English dictionary defines the verb “predetermine” as “to decide (something) before it happens or in advance.” EX1023, 3. Therefore, a POSITA would have understood “predetermined” to indicate that something was decided in advance.

131. At least for that reason, in SLC, the authentication policy is “*predetermined*”; indeed, per Starner, the machine-readable code is displayed even before being captured and evaluated by the HMD to determine which authentication factors are requested. EX1004, [0074]; EX1023, 3.

132. Additionally, SLC’s authentication policy is substantially similar to the example policies described in the ’982 Patent itself. *Cf.*, EX1001, 2:9-11, 6:60-62. For example, the ’982 Patent states that “[t]he authentication policy defines notably the number and/or the nature of the context data element(s) to be captured” and “[t]he authentication policy comprises a set of identified context data elements to be collected, so as to define context data element(s) to be collected.” EX1001, 2:9-11, 6:60-62.

Element [1.2] constituting a data packet, based on the at least one collected context data element;

133. SLC renders [1.2] obvious.

134. For context, the ’982 Patent broadly describes a “data packet” as including “a string of bits (or units of information) with a resulting length that may

vary depending on the collected context data element length(s) between separate context authentication sessions.” EX1001, 8:23-28. In some examples, “[t]he data packet length corresponds to an addition of all of the individual length(s) relating to the collected context data element(s).” *Id.* A data packet can be created simply “by *e.g.* concatenating the collected context data element(s).” *Id.*

135. Per Starner, and like the '982 Patent, SLC constitutes a data packet based on the at least one collected context data element at least by inserting the context data element in a data packet. For example, Starner explains that biometric data in the form of “[i]ndividual images of the eye” (*collected context data elements*) “may be captured by [] one or more sensors using a frame grabber during the eye-scan process.” EX1004, [0077]; *see also id.*, [0062]-[0066], [0076], [0089]. Individual images, which may optionally undergo further processing (*e.g.*, to identify “patterns” such as “patterns in an iris”), are stored as data records and inserted in a digital template (*data packet*). EX1004, [0089] (“stored eye-scan data could take the form of a 1024-byte data record”), [0077] (“the digital template may include a 1024-byte record (32x32x1 byte grayscale or color)”), [0062] (“Multiple scans may provide a more detailed digital template to the HMD ...”). The digital template or a hash thereof is provided as “authentication information” in an “authentication message” that the HMD generates and sends to an authentication server. EX1004, [0080]-[0081], [0085]. By constituting a digital template from one

or more biometric images (e.g., eye-scan images), per Starner, SLC “*constitut[es] a data packet, based on the at least one collected context data element*” as claimed.

Element [1.3] generating a hash, by using a predetermined hash type algorithm and the data packet, as input to the predetermined hash type algorithm

136. SLC renders [1.3] obvious.

137. Per Starner, in SLC, a digital template (***data packet***) is provided as input to a predetermined hash-type algorithm that is used to generate the hash. For example, as Starner describes, biometric data (e.g., one or more the eye-scan images) (***at least one context data element***) is used to create a digital template (***data packet***), as explained in §IX.B.1[1.2]. EX1004, [0077]. [0089]. Then, a “locality-sensitive hash (LSH) function, such as MinHash” (***predetermined hash type algorithm***) is “used by the HMD to generate [a] digital signature from the eye-scan data.” EX1004, [0078]. Since, per Starner, a predetermined hash function (e.g., LSH such as MinHash) is applied to generate a “digital signature” from biometric data such as “eye-scan data,” and the biometric data “take[s] the form of a 1024-byte data record” included in a “digital template,” SLC generates a ***hash*** (e.g., Starner’s digital signature), by using a ***predetermined hash type algorithm*** (e.g., LSH/MinHash) and the ***data packet*** (e.g., Starner’s digital template), as input to the predetermined hash type algorithm (e.g., LSH/MinHash). EX1004, [0078] (“a digital signature of the eye-scan data”), [0089] (“digital template may be used as [] eye-scan data”), [0077],

[0062]; §IX.B.1[1.2]. Indeed, Starner discloses that the “digital signature” can be generated by hashing a broad range of “eye-scan data []and/or other biometric data,” which includes the digital template. EX1004, [0078]; *see also id.*, [0092] (“digital signatures that have been generated ... from the biometric data using an LSH function, rather than the actual biometric data”), [0062] (“biometric data” includes “a more detailed digital template”), [0089]. SLC’s hash-type algorithm is predetermined; indeed, the HMD must use the same hash-type algorithm on the new digital signature as previously determined and applied to the enrolled digital signature. *Infra*, §IX.B.1[1.5].

138. In SLC, the hash is generated using a MinHash function like that described in Leskovec. EX1007, 78-80, 85-86; §IX.A. Leskovec describes details on “Minhashing” (EX1007, §3.3.2 (pp. 76-77)), computing “minhash signatures” (EX1007, §§3.3.4-3.3.5 (pp. 78-82)), and “LSH for Minhash Signatures” (§§3.4-3.4.3 (pp. 82-86)). *Supra*, §VIII.B; EX1007, 119-121. It would have been obvious to apply the details of Leskovec’s MinHash function to Starner for the reasons addressed above in the overview of SLC. *Supra*, §IX.A.

Element [1.4] sending the generated hash;

139. SLC renders [1.4] obvious.

140. For example, Starner explains that “the HMD *sends* the authentication message to an authentication server for verification of the authentication information,

where verification of the authentication information allows access to the protected information via the second computing device.” EX1004, [0085], FIG. 3A. “The authentication information may comprise encrypted biometric data.” EX1004, [0081]; *see also*, EX1004, claims 1, 15. As described above, in Starner, the authentication information sent in an authentication message includes a digital signature in the form of a “***generated hash***” of a digital template of the eye-scan data. EX1004, [0092] (“the authentication server ... may receive encrypted digital signatures that have been generated at the HMD from the biometric data using an LSH function”), [0078], [0089]; *see also id.*, [0080]-[0084] (generating authentication message), [0085]-[0086] (sending authentication message including generated hash); *supra*, §IX.B.1[1.3]. Thus, per Starner, SLC sends the digital signature (***generated hash***) for verification by an authentication server. EX1004, [0078].

141. For context, it was well-known before the Critical Date that, rather than sending an entire data item, sending a generated hash computed from the data item reduces network bandwidth consumption. For example, Pace describes a system that “generates digital IDs in accordance with the MD5 hash,” explaining that the “digital ID usage in this context reduces bandwidth which is required to be transported across the network.” EX1056, 3:66-67, 4:35-36.

Element [1.5] generating, as a hash distance generation step, a hash distance between the generated hash and a predetermined reference hash; and

142. SLC renders [1.5] obvious.

143. Starner describes creating a *predetermined reference hash*. For example, “an enrollment process may be implemented to establish [] predetermined stored biometric data such as stored eye-scan data.” EX1004, [0089]. “The enrollment process may involve a similar, but more detailed scan of the eye as discussed above,” *e.g.*, “[o]ne or more sensors may be configured to capture several individual digital still images of a wearer’s eye during the enrollment process. The HMD (and/or other computing device) may then analyze the patterns in the eye and convert them into a stored digital template.” EX1004, [0089]. “The stored digital template may be used as stored eye-scan data, which can be matched to a later eye scan to identify the particular wearer of the HMD.” EX1004, [0089]. In an example embodiment, “stored eye-scan data could take the form of a 1024-byte data record.” EX1004, [0089].

144. In some examples, per Starner, reference data created through the enrollment process is predetermined by hashing a digital template of the eye-scan image(s). *See supra*, §IX.B.1[1.3]-[1.4]. For example, Starner explains that “the authentication server (and/or other computing devices) may receive encrypted digital signatures that have been generated at the HMD from the biometric data using

an LSH function, rather than the actual biometric data.” EX1004, [0092]. “By using an LSH function to generate an initial digital signature for eye-scan data, the same digital signature can then be generated from any image of the wearer's eye(s) that substantially matches the initial image.” EX1004, [0092]. Further, “the authentication server may decrypt the encrypted digital signature and compare the encrypted digital signature to other digital signatures generated from previously-obtained eye-scan data of the wearer to determine if the encrypted digital signature matches the other digital signatures.” EX1004, [0092]; *see also id.*, [0077]-[0078], [0089]. The initial digital signature created by the LSH function and used by the authentication server as a reference for comparison to a new digital signature received from the HMD thus provides a “*predetermined reference hash*.”

145. The '982 Patent explains that “[t]he LSH type algorithm, as a hash type algorithm, is advantageously used to generate a score, as a hash distance used for comparing the generated hash to the predefined reference hash,” and the “generated score is then checked with respect to a predefined threshold value, so as to authenticate (or not) the concerned context.” EX1001, 3:32-38. Further, in the '982 Patent, “the score, as the hash distance, [may be] more or less permissive with respect to the predefined reference hash that has to be approached by the generated hash from the collected context data element(s).” EX1001, 3:39-42. Thus, the '982 Patent explains that the score is a type of hash distance. EX1001, 3:32-42.

146. Like the '982 Patent, Starner indicates that the use of digital signatures generated by an LSH function allows a match to be determined using a new digital signature and the initial/reference signature generated during enrollment even if the biometric data are similar but not identical. *See* EX1004, [0092] (“By using an LSH function ..., the same digital signature can then be generated from any image of the wearer's eye(s) that substantially matches the initial image.”). Further, even though there may be variations of the captured images of the wearer’s eye(s) (e.g., lighting and eye position), the LSH function allows for the same digital signature to be generated from each of the various images.”); *cf.* EX1001, 3:32-42.

147. In SLC, Leskovec likewise discloses an approach to generating a hash distance. *See, e.g.,* EX1007, 85-86; *see also id.*, §3.3 (pp. 75-76), §3.3.3 (pp. 77-78), §§3.4-3.4.1 (pp. 82-84), §3.5 (p. 87), §3.5.3 (pp. 88-89). The approach can include “comput[ing] the minhash signatures,” for all items (e.g., documents or digital templates) then “[e]xamin[ing] each candidate pair’s signatures and determin[ing] whether the fraction of components in which they agree is at least t .” EX1007, 85-86. The “fraction of components” in which pairs of hashes agree represents a “**hash distance**.” EX1007, 85-86. Leskovec relatedly explains that “1 minus the Jaccard similarity is a distance measure ... called the Jaccard distance,” and the Jaccard distance, which is another expression of the claimed “**hash distance**,” can be computed between the minhash signatures of two items. EX1007, §3.5 (p. 87),

§3.5.3 (pp. 88-89), §3.3.3 (p. 77) (“The probability that the minhash function for a random permutation of rows produces the same value for two sets equals the Jaccard similarity of those sets.”), §3.3 (p. 75) (“compare the signatures of two sets and estimate the Jaccard similarity of the underlying sets from the signatures alone”), §3.1.1 (p. 69), §3 (p. 68) (“‘minhashing[]’ ... compresses large sets in such a way that we can still deduce the similarity of the underlying sets from the compressed versions”).

148. In the combination, per Leskovec, SLC’s method thus includes the authentication server comparing a minhash signature of a new biometric digital template of the user to a minhash signature of a predetermined (reference/initial) digital template created during enrollment by computing a hash distance such as the Jaccard distance or “fraction of components” that either agree or disagree¹² between the signatures (*generating a hash distance between the generated hash and the predetermined reference hash*). EX1004, [0092]; EX1007, 85-86. For example,

¹² A POSITA would have understood that the fraction of components that agree represents a similarity measure, while the fraction of components that do not agree represents a distance measure. EX1007, 87. There is a straightforward mathematical relationship between these measures and either would have been obvious to try in SLC. *Id.*

Chum explains that “in near duplicate image retrieval, we are interested in image pairs with a similarity greater than 35%.” EX1040, 5.

Element [1.6.1] authenticating successfully or not based on the generated hash distance, as an authentication;

149. SLC renders [1.6.1] obvious.

150. For example, per Starner, in SLC, the authentication server “compare[s] the encrypted digital signature to other digital signatures generated from previously-obtained eye-scan data of the wearer to determine if the encrypted digital signature matches the other digital signatures.” EX1004, [0092]; *supra*, §IX.B.1[1.5].

151. Further, per Leskovec, in SLC the generated ***hash distance*** (*see supra*, §IX.B.1[1.5]) is compared to a “threshold” to determine whether authentication is successful. EX1007, 85-86; *supra*, §VIII.B; §IX.A. Leskovec describes use of a threshold t that “defines how similar documents have to be in order for them to be regarded as a desired ‘similar pair’.” EX1007, 85-86. Further, per Leskovec, “discovering the truly similar documents,” includes “[e]xamin[ing] each candidate pair’s signatures and determine whether the fraction of components in which they agree is at least t .” EX1007, 85-86. Leskovec likewise explains that the Jaccard distance can be calculated as “1 minus the ratio of the sizes of the intersection and union of sets x and y ” and “[t]hus, the Jaccard distance $d(x,y)$ is the probability that a random minhash function *does not* send x and y to the same value.” EX1007, 88-89.

152. Based on these teachings, SLC's method includes the authentication server comparing a minhash signature of a new biometric digital template of the user to a minhash signature of a predetermined (reference/initial) digital template created during enrollment by computing a hash distance such as the Jaccard distance or "fraction of components" that either agree or disagree between the signatures (*generating a hash distance*), and comparing the determined fraction of components of the signatures that either agree or disagree to a threshold t to determine whether their similarity is sufficiently high (or their distance sufficiently small) to authenticate the user (*authenticating successfully or not based on the generated hash distance, as an authentication*). EX1007, 85-86.

Element [1.6.2] wherein said at least one context data element is a captured image represented as a string of bits with a fixed length for constituting said data packet, and

153. SLC renders [1.6.2] obvious for the reasons addressed above with respect to [1.1]. *Supra*, §IX.B.1[1.1].

Element [1.6.3] wherein said hash is a byte stream generated from said string of bits representing said captured image for said data packet.

154. SLC renders [1.6.3] obvious.

155. In SLC, per Starner and Leskovec, the digital template (*data packet*) includes biometric data such as eye-scan image(s) (*at least one context data element*) represented as a 1024-byte image/data record (*a string of bits with a fixed length*),

and the digital template including the 1024-byte image/data record is hashed by an LSH function such as MinHash (*generate a hash*). EX1004, [0077]-[0078], [0089], FIG. 3A; *supra*, §IX.B.1[1.1], [1.3], [1.6.2]. The digital template includes biometric image(s) for the data packet. EX1004, [0077] (“digital template may include a 1024-byte record (32×32×1 byte grayscale or color)”), [0062], [0089]; *supra*, §V.A.4.(a)[1.2]; *supra*, §IX.B.1[1.2].

156. Moreover, in SLC, the hash of the digital signature provides a *byte stream* as claimed. For example, per Starner, the result of hashing biometric data (such as the digital template and its constituent image(s)), is to produce a “digital signature.” EX1004, [0078], [0092]. A digital signature comprises digital data, and a POSITA would have appreciated that computers conventionally arranged the digital data of a file as a byte stream for storage, processing, and transmission by the Critical Date of the ’982 Patent, and indeed for decades before. For example, Swami explains that “Transport Layer (Layer4): This layer ... delivers the stream of bytes,” and Chen explains that “[w]hen data is written to the storage system, the storage system only sees a stream of bytes.” EX1063, 6; EX1064, 1. It also would have been obvious to generate the hash in SLC as a byte stream, precisely because the approach was exceedingly common and doing so would ensure that conventional computer memory and processors could efficiently store, process, and transmit the hash. *Id.* Notably, the ’982 Patent itself identifies nothing new or unique about

generating a byte stream; on the contrary, the '982 Patent simply acknowledges that the natural result of computing a “hash or digest” is to produce “a byte stream relating to original captured context data element(s).” *Cf.* EX1001, 2:56-60.

157. Leskovec likewise explains that, when computing a minhash signature, a “signature matrix” can be formed from a matrix X such that the “*i*th column of M is replaced by the minhash signature for (the set of) the *i*th column.” EX1007, 78. Leskovec further explains that the “minhash signature” for a set is the “vector [$h_1(S)$, $h_2(S)$, ... , $h_n(S)$]”, where h_1 , h_2 , ... $h_n(S)$ are permutations that determine the minhash function. EX1007, 78. Further, this list of hash values is represented as a column. EX1007, 78. A POSITA would have understood or found obvious that a vector (*e.g.*, [$h_1(S)$, $h_2(S)$, ... , $h_n(S)$]) of values would be represented as a byte stream. EX1007, 78. For example, the ISO C++ standard states that “[t]he elements of a vector are stored contiguously.” EX1060, 4

158. Therefore, SLC renders obvious the hash (*e.g.*, result of the MinHash) being *a byte stream generated from said string of bits representing said captured image for said data packet.*

2. Claim 2

Element [2.1] The method according to claim 1, wherein, the predetermined hash type algorithm comprises a locality sensitive hash type algorithm, and wherein the method further comprises, as the hash distance generation step and the authentication step, the following steps:

159. SLC renders [2.1] obvious.

160. For example, as I explained with reference to element [1.3], Starner states that “***a locality-sensitive hash (LSH) function***, such as MinHash, may be used by the HMD to generate the digital signature from the eye-scan data.” EX1004, [0078]; §IX.B.1[1.3]. As I further explained with reference to element [1.3], MinHash is a predetermined hash type algorithm. Therefore, consistent with Starner’s disclosure, SLC further implements Leskovec’s minhash LSH function in Starner’s biometric authentication process. EX1007, 68, 75-80; *supra*, §VIII.B; §IX.A. I discuss the hash distance generation step and the authentication step with reference to elements [2.2]-[2.4] below.

Element [2.2] generating a score, by using the locality sensitive hash type algorithm, as a level of similarity between the generated hash and the predetermined reference hash;

161. SLC renders [2.2] obvious.

162. As explained above in §IX.B.1[1.5], in SLC, per Leskovec, the method calculates a “fraction of components” (***score***) of the user’s current digital signature (***generated hash***) that “agree” with components of the initial digital signature created during enrollment (***predetermined reference hash***) to determine ***a level of***

similarity between the generated hash and the predetermined reference hash.

EX1007, 85-86; *supra*, §IX.B.1[1.5]. The score is generated using a *locality sensitive hash type algorithm* such as MinHash as taught in both Starner and Leskovec. *Supra*, §IX.B.1[1.3].

163. More specifically, Starner explains that, to authenticate a user based on data sent by the HMD (*supra*, §IX.B.1[1.4]), “the authentication server (and/or other computing devices) may receive encrypted digital signatures that have been generated at the HMD from the biometric data using an LSH function, rather than the actual biometric data.” EX1004, [0092]. Further, “the authentication server may decrypt the encrypted digital signature and compare the encrypted digital signature to other digital signatures generated from previously-obtained eye-scan data of the wearer to determine if the encrypted digital signature matches the other digital signatures.” EX1004, [0092].

164. In SLC, when comparing digital signatures, LSH determines a *score*, which is a level of similarity between the *generated hash* and the *predetermined reference hash*. EX1007, 85-86; *supra*, §IX.B.1[1.5]. For example, Leskovec describes an “approach to finding the set of candidate pairs for similar documents and then discovering the truly similar documents among them.” EX1007, 85-86. The approach includes “comput[ing] the minhash signatures for all documents,” then “[e]xamin[ing] each candidate pair’s signatures and determine whether the fraction

of components in which they agree is at least t .” EX1007, 85-86. Thus, per Leskovec, SLC computes a score, *e.g.*, “the fraction of components of the minhash signatures for the new and enrolled digital template that agree. EX1007, 85-86; §VIII.B; §IX.A; Per Leskovec, SLC can also compute a ***score*** as the Jaccard similarity between minhash signatures. EX1007, 87, 88, 95; EX1007, 87, 88, 95; *supra*, §IX.B.1[1.5].

Element [2.3] comparing the score to a predetermined threshold value;

165. SLC renders [2.3] obvious.

166. As discussed above (*supra*, §IX.B.1[1.6.1]), per Leskovec, SLC’s process for authenticating a user by comparing minhash signatures involves “[c]hoos[ing] a ***threshold*** t that defines how similar” the signatures “have to be in order for them to be regarded as a desired ‘similar pair,’” then “[e]xamining each candidate pair’s signatures and determine[ing] whether the fraction of components in which they agree is at least t .” EX1007, 85-86. Thus, SLC discloses comparing the ***score*** (*e.g.*, the fraction of components in which candidate pair signatures agree, or the Jaccard similarity) to a ***threshold value***, t .

Element [2.4] authenticating successfully only if the score is equal to or greater than the predetermined threshold value.

167. SLC renders [2.4] obvious.

168. Per Leskovec, in SLC, authentication is successful only when “the fraction of components in which they agree is at least t ” (***only if the score is equal***

to or greater than the predetermined threshold value). EX1007, 85-86; EX1004, [0092] (user authenticated if “the encrypted digital signature matches the other digital signatures”), [0118]-[0119], FIGS. 3B (310), 5A-5D. A POSITA would have understood that the statement “is at least t ” indicates that the score must be equal to or greater than “ t ”.

169. Leskovec further explains that “if the signatures are sufficiently similar, go to the documents themselves and check that they are truly similar, rather than documents that, by luck, had similar signatures.” EX1007, 85-86. Since this optional step only occurs if “the fraction of components in which they agree is at least t ,” authentication is successful only when the score (the fraction of components in which they agree) is equal to or greater than the predetermined threshold value (t). EX1007, 85-86.

170. Note that, with this optional step, in some cases (“by luck, had similar signatures”) authentication can fail even when the score is equal to or greater than the predetermined threshold. EX1007, 85-86. However, the claim states that “authenticating successfully only if the score is equal to or greater than the predetermined threshold value,” not “authenticating successfully **if and** only if the score is equal to or greater than the predetermined threshold value.”

3. Claim 3

Element [3] The method according to claim 1, wherein, prior to generating the hash, the method further comprises sorting, according to a numerical order, the at least one collected context data element within the data packet.

171. SLC renders [3] obvious in several ways.

172. First, per Starner, multiple “forms of biometric data, such as fingerprint data, hand/finger geometry, vein geometry, and voiceprints, among other possibilities, may be utilized in addition or in the alternative to eye data.” EX1004, [0025]. Biometric data in Starner is represented in data records (***context data elements***), and a POSITA would have understood or found obvious that the data records for each of the multiple forms of biometric data used are arranged in a digital template (***data packet***) (or multiple digital templates would be arranged within a larger template) to allow the authentication server to authenticate the user based on each of the various forms of authentication data. EX1004, [0077]-[0078], [0089], [0092]. Because the minhash function is sensitive to the value of the input message and generally produces different outputs (digests) for different inputs, and because the value of the digital template provided as input to the minhash function would vary depending on the order that data records for the different forms of biometric data were arranged, it would have been obvious to ***sort*** the data records in a user’s new digital template according to a numerical order that matches the order of data records for the different forms of biometric data in the digital template that was hashed to produce the enrolled digital signature at the authentication server. If the

data records were not sorted, the hash of a new digital template would not match the enrolled digital signature, thus resulting in an inability to properly authenticate authorized users. *Id.* Thus, the data records must be sorted “*prior to generating the hash.*” *Id.*

173. In another example, Starner discloses that “the owner of the HMD may have previously stored one or more images resulting from a scan of the wearer’s iris, and that previously stored image(s) may be compared with [an] iris scan image(s) of the wearer in the authentication information so as to verify the identity of the wearer,” and “[t]he image(s) can be provided as one or more still images or a video of the wearer’s iris.” EX1004, [0118]. Starner thus makes clear that either a single image or multiple images (*context data element(s)*) of the wearer’s eye as data records in a digital template can be used for authentication, and in the case of multiple images, the images can be a “video” of the wearer’s eye. EX1004, [0077]; *supra*, §IX.B.1[1.2]. In the case where just a single image is represented in the digital template (*e.g.*, a single 1024-byte record), it would have been obvious that the HMD sorts the data record for the single image appropriately within the digital template according to the one and only possible numerical order for the single image.¹³ In the

¹³ Claim 3 refers to sorting “at least one context data element within the data packet,” and therefore encompasses sorting even a single context data element.

case where a sequence of images in a video are represented in the digital template, it would have been obvious that the HMD sorts the images within the data packet according to the numerical order of frames in the video to maintain the video sequence, consistent with ordinary video encoding techniques. Starner teaches that the digital template (*data packet*) is formed before it is hashed to generate a digital signature; therefore, the sorting of image(s) within the digital template occurs “*prior to generating the hash.*” EX1004, [0077]-[0078].

174. Additionally, per Leskovec, SLC’s technique for determining whether pairs (e.g., hashed, enrolled eye-scan data and hash eye-scan data used for authentication) are similar includes:

1. Pick[ing] a value of k and construct from each document the set of k -shingles. Optionally, hash the k -shingles to shorter bucket numbers.
2. *Sort[ing]* the document-shingle pairs to order them by shingle.
3. Pick a length n for the minhash signatures. Feed the sorted list to the algorithm of Section 3.3.5 to *compute the minhash signatures* for all the documents.

EX1007, 85-86.

175. Thus, per Leskovec, the sorting step (2) occurs before the step of computing the minhash signatures (3). EX1007, 85-86.

176. Further, a POSITA would have understood or found obvious that since the MinHash produces numeric values (“hash function that maps integers 0, 1, . . . , $k-1$ to **bucket numbers** 0 through $k-1$ ”), Leskovec’s sorting operation is performed according to a numerical order. EX1007, 78, *see also* EX1007, 79-80.

4. Claim 5

Element [5] The method according to claim 1, wherein, prior to constituting the data packet, the method further comprises grouping together, according to the authentication policy, at least two context data elements comprised within the at least one context data element, to constitute at least one corresponding context data group.

177. SLC renders [5] obvious.

178. In SLC, per Starner, the HMD can capture multiple images since “[m]ultiple scans may provide a more detailed digital template to the HMD since the sensors will have more time to capture individual images of a user's eye.” EX1004, [0062]. Starner explains that “[i]ndividual images of the eye” (***at least two context data elements***) “may be captured by the one or more sensors using a frame grabber during the eye-scan process.” EX1004, [0077]. For example, an “eye-scan prompt may involve displaying an indication to perform one or more hand gestures,” and the “wearer may then track their own hand movements with their eye(s) while keeping their head stationary as the sensors capture images of the eye(s) over a given period of time.” EX1004, [0064]. “[T]he HMD may analyze the images to identify patterns in an iris, and convert the patterns into a digital template.” EX1004, [0077].

In another example, Starner groups together multiple eye-scan images used for authentication in a “video” sequence. EX1004, [0118]. By analyzing and extracting features from groups of images (*at least two context data elements*) before forming the digital template (*data packet*) as taught in Starner, SLC discloses and renders obvious that, *prior to constituting the data packet (e.g., digital template), the method groups together at least two context data elements (e.g., at least two eye-scan images) to constitute at least one corresponding context data group*. EX1004, [0077].

179. It also would have been obvious that the grouping operations are performed in accordance with the *authentication policy* since the HMD must obtain the eye-scan images and process them to form the digital template in a predetermined manner, and thus according to a policy, that would allow it to be compared and matched against the reference data created during enrollment. *Supra*, §IX.B.1[1.1], [1.6.1]; EX1004, FIG. 3B. As discussed above, Starner’s indication of the authentication policy may be encoded in the machine-readable code (*e.g.*, QR-code) displayed on the second computing device and detected by the HMD. EX1004, [0072], [0074]; *supra*, §IX.B.1[1.1].

5. Claim 6

Element [6] The method according to claim 1, wherein, prior to generating the hash, the method further comprises ciphering the data packet, according to the authentication policy.

180. SLC renders [6] obvious.

181. Per Starner, the HMD computes a MinHash. EX1004, [0078]; §IX.B.1[1.3]; §IX.B.2[2.1]. Per Leskovec, before generating a MinHash signature for input data, the input data can be segmented (*e.g.*, to create shingles) and the segmented input data can be hashed to reduce the size of the data segments from which the MinHash signature is subsequently generated. For example, Leskovec explains that “[i]nstead of using substrings directly as shingles, we can pick a hash function that maps strings of length *k* to some number of buckets and treat the resulting bucket number as [a] shingle,” *e.g.*, “thereby reducing the necessary storage to “four bytes instead of nine.” EX1007, 74; *see also id.*, 85-86 (“1. ... Optionally, hash the *k*-shingles to shorter bucket numbers. ... 3. ... compute the minhash signatures”). In the context of the ’982 Patent, hashing is a form of ciphering. *See* EX1001, 9:14-18 (“[s]uch a generated hash, as a cipher ...”). Accordingly, Leskovec teaches operations for ciphering data before generating a hash by first hashing the input data segments (*e.g.*, shingles) and then separately hashing the hashed data segments to generate the MinHash signature. EX1007, 74, 85-86. In the combination with Starner, SLC would thus cipher the digital template by hashing segments of the digital template (*cipher the data packet*) prior to

generating the MinHash signature (*generating the hash*). SLC would further cipher before generating the hash according to the *authentication policy* indicated by the machine-readable code as described above (*supra*, §IX.B.1[1.1]); indeed, the HMD would cipher the biometric digital template before generating the hash to match similarly generated enrolled data.

182. As another example, in some cases, SLC requires multiple instances of biometric authentication, e.g., to access multiple information sources. For instance, per Starner, a user can access “an email account, bank account, or user profile account.” EX1004, [0069], [0119]. A POSITA would have understood or found obvious that such accounts can reside on different servers (e.g., bank server, email provider), each requiring biometric-based authentication. EX1004, [0060]-[0061] (each remote server 230 “may include a computing cluster associated with a particular web service”), [0085] (“authentication server may be one or more computing devices separate from the HMD and the second computing device” or “the second computing device may function as the authentication server”), [0126]-[0133]. Further, Starner explains that “the HMD can continuously verify the wearer during use,” also verifying the user’s identity using eye-scan data. EX1004, [0096].

183. Per Starner, the HMD creates stored “eye-scan data” and forms digital templates from the eye-scan data. EX1004, [0089]. Also per Starner, in some instances, when authenticating, SLC transmits an encrypted digital template

(representing the eye-scan data), and in other instances transmits a hashed digital template. EX1004, [0023], [0078], [0080]-[0084], [0086], [0089], [0115]-[0118], claim 4; §§IX.B.1[1.3]-[1.4]. Further, SLC can reuse eye-scan data. EX1004, [0094] (“only an initial eye-scan may be needed from the owner of the HMD”). Therefore, when SLC first uses eye-scan data to access a first resource requiring encrypted biometric data for authentication, then accesses a second resource requiring hashed biometric data, SLC first *ciphers the data packet* for authentication to the first resource and then *hashes* the data packet for authentication to the second resource.

184. Additionally, in SLC, ciphering is performed according to an *authentication policy*. For example, per Starner, in SLC, when a wearer of an HMD “wants to access secured information on a second computing device, such as via a password-protected webpage,” the HMD may detect a quick response (QR) code included on the login webpage,” and a request “to provide authentication information may also be encoded in the bar code.” EX1004, [0021], [0072]; *supra*, §IX.B.1[1.1]. “In response, the HMD may obtain biometric data identifying the wearer.” EX1004, [0021]. Starner’s QR code, which is a “machine-readable code” thus defines an “*authentication policy*” that requests collection of certain authentication factor(s). EX1004, [0021], [0072].

6. Claim 8

Element [8pre] A first authentication device, wherein the first authentication device is configured to:

185. To the extent the preamble is limiting, SLC renders [8pre] obvious.

186. For example, per Starner, SLC includes a “first computing device” (***first authentication device***) that “acquires biometric data via one or more biometric sensors associated with the first computing device, generates an authentication message based at least in part on the biometric data,” and “sends the authentication message to an authentication server for verification of the authentication information.” EX1004, [0005]; *supra*, §VIII.A. Starner provides an example of “a wearable computing system” that “takes the form of a head-mountable device (HMD) 102 (which may also be referred to as a head-mounted display),” but “example systems and devices may take the form of or be implemented within or in association with other types of devices, without departing from the scope of the invention.” EX1004, [0030]; *see also id.*, [0067]-[0068], FIGS. 1A-1G, 2, 5B.

Element [8.1] collect at least one context data element, based on a predetermined authentication policy;

187. Starner renders [8.1] obvious.

188. SLC collects at least one context data element based on a predetermined authentication policy. *Supra*, §IX.B.1[1.1].

189. In SLC, per Starner, the context data element is collected by the first authentication device. For example, Starner states that the HMD (***first biometric***

device) can “obtain biometric data of a wearer of the HMD utilizing one or more sensors (e.g., infrared proximity sensors, infrared photodiodes, and/or cameras) or other devices coupled to the HMD” and “the HMD acquires biometric data via one or more biometric sensors.” EX1004, [0062], [0076].

Element [8.2] constitute a data packet, based on the at least one collected context data element;

190. Starner renders [8.2] obvious.

191. SLC constitutes a data packet, based on the at least one collected context data element. *Supra*, §IX.B.1[1.2].

192. In SLC, the data packet is constituted by the first authentication device (e.g., HMD). EX1004, [0077].

Element [8.3] generate a hash, by using a predetermined hash type algorithm and the data packet, as input to the predetermined hash type algorithm; and

193. Starner renders [8.3] obvious.

194. In SLC, the first authentication device (e.g., HMD) generates a hash, by using a predetermined hash type algorithm and the data packet, as input to the predetermined hash type algorithm. *Supra*, §IX.B.1[1.3].

Element [8.4.1] send the generated hash to a second authentication device;

195. Starner renders [8.4.1] obvious.

196. In SLC, the first authentication device (e.g., HMD) sends the generated hash to a second authentication device (e.g., an authentication server). EX1004, [0085]-[0087], FIGS. 3A-3B; *supra*, §IX.B.1[1.4].

Element [8.4.2] wherein said at least one context data element is a captured image represented as a string of bits with a fixed length for constituting said data packet, and

197. Starner renders [8.4.2] obvious. *Supra*, §IX.B.1[1.6.2].

Element [8.4.3] wherein said hash is a byte stream generated from said string of bits representing said captured image for said data packet.

198. Starner renders [8.4.3] obvious. *Supra*, §IX.B.1[1.6.3].

7. Claim 9

Element [9pre] A second authentication device, wherein the second authentication device is configured to:

199. To the extent the preamble is limiting, SLC renders [9pre] obvious.

200. SLC includes a second authentication device. For example, per Starner, in SLC, the “first computing device can [] send [an] authentication message to an authentication server” (*second authentication device*) “for verification of the authentication information.” EX1004, Abstract. “The authentication server may be one or more computing devices separate from the HMD.” EX1004, [0085]. The “authentication server [] process[es] and decrypt[s] the encrypted authentication information to identify the wearer of the HMD.” EX1004, [0086], [0111] (“other computing device”), FIGS. 3A-3B.

201. Further, the authentication server is a device. For example, Starner writes that examples of the “other computing device” include “the authentication server.” EX1004, [0111]. Additionally, Starner writes that “the encrypted authentication information may be provided to the authentication server or other computing device.” EX1004, [0086]. Since Starner describes an “other computing device” in relation to the authentication server, a POSITA would have understood that the authentication server is also a device.

Element [9.1] receive a generated hash;

202. SLC renders [9.1] obvious.

203. In SLC, the authentication server (***second authentication device***) receives the generated hash. As explained above in §IX.B.1[1.4], per Starner, in SLC, a first authentication device (*e.g.*, HMD) sends a generated hash (*e.g.*, digital signature) to the authentication server (***second authentication device***). *Supra*, §IX.B.1[1.4]. For example, Starner states that the “first computing device can then send the authentication message to an authentication server for verification of the authentication information.” EX1004, Abstract. Starner further explains that the authentication server “may ***receive*** encrypted digital signatures that have been generated at the HMD from the biometric data using an LSH function, rather than the actual biometric data.” EX1004, [0092]; *see also id.*, [0077]-[0078], [0089], FIGS. 3A-3B; *supra*, §IX.B.1[1.3]-[1.4].

Element [9.2] generate a hash distance between the generated hash and a predetermined reference hash; and

204. SLC renders [9.2] obvious.

205. SLC generates a hash distance between the generated hash and a predetermined reference hash. *Supra*, §IX.B.1[1.5].

206. In SLC, the authentication server (*second authentication device*) generates the hash distance. For example, per Starner, the authentication server “decrypt[s] the encrypted digital signature and compare[s] the encrypted digital signature to other digital signatures generated from previously-obtained eye-scan data of the wearer to determine if the encrypted digital signature matches the other digital signatures.” EX1004, [0092]; *see also id.*, [0077]-[0078], [0089], FIGS. 3A-3B; *supra*, §IX.B.1[1.3]-[1.4].

207. Per Leskovec, in SLC, the authentication server (*second authentication device*) compares the signatures by generating a hash distance, as explained in §IX.B.1[1.5].

Element [9.3] authenticate successfully or not based on the generated hash distance.

208. SLC renders [9.3] obvious.

209. SLC authenticates successfully or not based on the generated hash distance. *Supra*, §IX.B.1[1.5]-[1.6.1];

210. Per Starner, the authentication server (*second authentication device*) authenticates successfully or not. For example, per Starner, in SLC, the “authentication server may then access the stored biometric data and compare the stored biometric data to the decrypted biometric data.” EX1004, [0088]. Additionally, per Starner, in SLC, the authentication server (*second authentication device*) “compare[s] the encrypted digital signature to other digital signatures generated from previously-obtained eye-scan data of the wearer to determine if the encrypted digital signature matches the other digital signatures.” EX1004, [0092]. The authentication can be based on the generated hash distance calculated according to Leskovec’s techniques. *Supra*, §VIII.B, V.A.4.(a)[1.6.1], V.A.4.(b).

8. Claim 10

Element [10pre] An authentication system, comprising

211. To the extent the preamble is limiting, SLC renders [10pre] obvious. *See, e.g.*, §IX.B.1[1pre].

212. For example, Starner explains that the “above detailed description describes various features and functions of the disclosed *systems*, devices, and methods with reference to the accompanying figures.” EX1004, [0127]; *see also id.*, [0018], [0027], FIGs. 1A-1G, 2, 5B. Starner further describes “example methods and systems” and “[s]ystems and devices in which example embodiments may be

implemented will now be described in greater detail.” EX1004, [0018], [0027], FIGS. 1A-1G, 2, 5B (depicting disclosed authentication systems).

Element [10.1] a first authentication device and

213. SLC renders [10.1] obvious. *Supra*, §IX.B.5[8pre].

Element [10.2] at least one second authentication device,

214. SLC renders [10.2] obvious. *Supra*, §IX.B.7[9pre].

Element [10.3] wherein the first authentication device is configured to: collect at least one context data element, based on a predetermined authentication policy;

215. SLC renders [10.3] obvious. *Supra*, §IX.B.1[1.1].

Element [10.4] constitute a data packet, based on the at least one collected context data element;

216. SLC renders [10.4] obvious. *Supra*, §IX.B.1[1.2].

Element [10.5] generate a hash, by using a predetermined hash type algorithm and the data packet, as input to the predetermined hash type algorithm; and

217. SLC renders [10.5] obvious. *Supra*, §IX.B.1[1.3].

Element [10.6] send to the second authentication device, the generated hash;

218. SLC renders [10.6] obvious. *Supra*, §IX.B.1[1.4].

Element [10.7] and wherein the second authentication device is configured to: receive the generated hash from the first device;

219. SLC renders [10.7] obvious. *Supra*, §IX.B.7[9.1].

Element [10.8] generate a hash distance between the generated hash and a predetermined reference hash; and

220. SLC renders [10.8] obvious. *Supra*, §IX.B.1[1.5].

Element [10.9] authenticate successfully or not based on the generated hash distance;

221. SLC renders [10.9] obvious. *Supra*, §IX.B.1[1.6.1].

Element [10.10] wherein said at least one context data element is a captured image represented as a string of bits with a fixed length for constituting said data packet, and

222. SLC renders [10.10] obvious. *Supra*, §IX.B.1[1.6.2].

Element [10.11] wherein said hash is a byte stream generated from said string of bits representing said captured image for said data packet.

223. SLC renders [10.11] obvious. *Supra*, §IX.B.1[1.6.3].

X. GROUND 1B: CLAIMS 1-6, 8-10 ARE OBVIOUS IN VIEW OF THE STARNER-LESKOVEC-SHAASHUA COMBINATION

A. Combining Starner-Leskovec with Shaashua

1. Overview of the Combination (SLSC)

224. Per Starner, in SLC, “[v]arious types of biometric data may be utilized to verify a user's identity and authorize access to secure content,” *e.g.*, “a mobile device such as an HMD or mobile phone may be configured to capture an image of a user's eye.” EX1004, [0025]; *supra*, §IX.A-B. Starner further explains that “[o]ther forms of biometric data, such as fingerprint data, hand/finger geometry, vein geometry, and voiceprints, among other possibilities, may be utilized in addition or

in the alternative to eye data.” EX1004, [0025]. Per Starner, SLC’s HMD “can obtain biometric data of a wearer of the HMD utilizing one or more sensors (*e.g.*, infrared proximity sensors, infrared photodiodes, and/or cameras) or other devices coupled to the HMD.” EX1004, [0062]. Starner explains that “the HMD may detect in the [a] machine-readable code” such as a QR-code “requests to provide [] authentication factors,” and “the QR code ... may be representative of a challenge request for authentication,” where “the challenge request may indicate that biometric data must be provided in order to access the protected information.” EX1004, [0074]-[0075]. Per Starner, the QR code encodes information that, when detected, causes the HMD to authenticate according to an authentication policy (*supra*, §IX.B.1[1.1]; EX1004, [0025], [0093]), although Starner does not describe in detail how the authentication policy is defined or the conditions for selecting different authentication factors.

225. To improve the selection of biometric factors in Starner, a POSITA would have looked to references such as Shaashua that describe an authentication method that “involves selecting among different biometric methods.” EX1005, 2:3-4; *supra*, §VIII.C. Shaashua explains that, “although an arsenal of biometric methods may be available for use, poorly suited biometric methods of the arsenal can effectively be ruled out (*i.e.*, deemed unavailable for use),” and “the best suited biometric method can be selected based on a variety of factors including device factors (*i.e.*, what circuitry is available to capture biometric data), environmental

factors (*i.e.*, the states of ambient conditions), and even biometric method strength factors (*i.e.*, weights defining particular security strengths of the available biometric methods).” EX1005, 2:3-13. Shaashua’s “techniques improve the user's experience by lowering the false negative rate (*i.e.*, users do not fail authentication easily due to unsuitable environment or device factors).” EX1005, 2:13-16.

226. It would have been obvious to apply Shaashua’s techniques for selecting among different biometric methods according to an authentication policy to the Starner-Leskovec system of Ground 1A.

227. The resulting Starner-Leskovec-Shaashua Combination (SLSC) includes a computing device, *e.g.*, an HMD (per Starner) and an authentication server (per Starner and Shaashua). EX1004, Abstract; EX1005, 4:62-67; §VIII.A; §X.A.

228. Per Starner, SLSC’s “enrollment process” “establish[es] the predetermined stored biometric data, such as stored eye-scan data.” EX1004, [0089]. The stored biometric data are used to authenticate (or not) the user, as explained further below. EX1004, [0077], [0087], [0089]-[0090]. Such authentication can be required “prior to allowing the users 30 to access protected resources such as sensitive files and other content, email, VPN access, transactions, games, etc.” EX1005, 5:31-34. Starner similarly explains that “the biometric data may be used to authenticate a user in lieu of a password, *e.g.*, so that a user's HMD can log the user

on to a webpage on their laptop,” and “[a]s such, the biometric data can serve as a replacement for passwords entirely.” EX1004, [0022].

229. Further, per Starner, SLSC’s computing device, “detects a machine-readable code,” *e.g.*, a QR code. EX1004, [0005], [0021]. The “machine-readable code identifies protected information that is viewable via the second computing device,” *e.g.*, a web server hosting a protected web page. EX1004, [0005], [0021]. The “computing device acquires biometric data via one or more biometric sensors.” EX1004, [0005].

230. In the resulting Starner-Leskovec-Shaashua Combination (SLSC), per Shaashua, the “[d]etermination of which biometric methods from the set of suitable biometric methods are to be used (*i.e.*, which of the above-described arrangements should be used) can be dictated by a predefined policy,” and “one or more of the suitable biometric methods (*e.g.*, based on policy) is applied during authentication to determine whether the user 30 of the smart device 22 is legitimate.” EX1005, 9:2-4, 11:4-7. For example, in some implementations, SLSC “applies the biometric method of the set of suitable biometric methods having the highest suitability score,” which can be “appropriate in situations in which the users 30 should not be overburdened during biometric collection (*e.g.*, it could be too burdensome to ask a user 30 to concurrently provide a facial image, a voice sample, a fingerprint scan, and a handwriting sample all in the same authentication attempt). EX1005, 8:53-60.

The suitability score is “based on (i) the set of suitability factors and (ii) a biometric method strength factor which defines a particular security strength for that available biometric method relative to other security strengths for other available biometric methods.” EX1005, 3:28-33. The “biometric method strength factors” are determined by “weights defining particular security strengths of the available biometric methods.” EX1005, 2:3-13.

231. Per Starner, “a digital signature of the eye-scan data (and/or other biometric data) [is] generated by the HMD (or other computing device) from the eye-scan data and used for authorization rather than the actual images of the wearer's eye(s), thus providing additional security for the wearer's biometric data.” EX1004, [0078]. In SLSC, a locality-sensitive hash (LSH) function, MinHash, is used by the HMD to generate the digital signature from the eye-scan data. EX1004, [0078]. The MinHash is determined using the techniques of Leskovec. EX1007, 82-86.

232. Per Starner, the SLSC device then “generates an authentication message based at least in part on the biometric data,” and “sends the authentication message to an authentication server for verification of the authentication information.” EX1004, [0005].

233. Upon receiving the authentication message, SLSC’s authentication server compares “the received data and corresponding digital template with the stored data to determine if there is a match.” EX1004, [0090]. SLSC uses the

techniques of Leskovec to determine whether the hashed biometric factor matches an enrolled entry. §§VIII.A-C, §IX.B.1[1.5]-[1.6.1].

2. Motivations to Combine

234. The following reasons, taken separately or collectively, support the above-described Starner-Leskovec-Shaashua Combination (SLSC).

235. First, a POSITA would have applied Shaashua's techniques in SLSC to improve the user's experience by ensuring the user can more often authenticate according to a biometric method suitable or optimized to the user's current situation. EX1005, 2:13-16. An authorized user would thus experience fewer instances of failed authentication attempts due to problems with the selected biometric method(s). *Id.* For example, Shaashua explains that the "techniques improve the user's experience by lowering the false negative rate (*i.e.*, users do not fail authentication easily due to unsuitable environment or device factors)." EX1005, 2:13-16 8:53-60.

236. Second, a POSITA would have applied Shaashua's techniques in SLSC to reduce the user's burden in successfully completing authentication. Shaashua explains that its techniques are "appropriate in situations in which the users 30 should not be overburdened during biometric collection (*e.g.*, it could be too burdensome to ask a user 30 to concurrently provide a facial image, a voice sample,

a fingerprint scan, and a handwriting sample all in the same authentication attempt).”
EX1005, 8:53-60.

237. Third, a POSITA would have been prompted to implement SLSC because it merely involves the application of known techniques as described in Shaashua to a known system as described in Starner and Leskovec to achieve predictable results (*e.g.*, a biometric authentication system with improved accuracy and lower user burden). Each element in SLSC merely performs the same function as it does separately. For example, the selection of non-optimal biometric methods was a known problem and Shaashua helps address that issue. Further, combining the teachings of Starner, Leskovec and Shaashua was not beyond the skill of an ordinary artisan. The evidence shows that each of these factors that are sufficient to demonstrate that a motivation to combine are present here.

3. Reasonable Expectation of Success

238. A POSITA would have reasonably expected success implementing SLSC. For example, SLSC includes a known biometric authentication system and uses known policy-based selection of authentication factors to improve the user experience (reduces false negative results while requiring only a single biometric factor). §VIII.A-3; §§X.A-3. Integrating these aspects of Starner, Leskovec and Shaashua would have been well within the skill of a POSITA.

B. Analysis with Respect to Claims 1-6, 8-10

239. Except for the claim elements specifically addressed below, SLSC provides each element of claims 1-6 and 8-10 for the same reasons that SLC provides those claim elements as discussed above in Ground 1A. *Supra*, §IX.B.

1. Claim 1

Element [1.1] collecting, based on a predetermined authentication policy, at least one context data element;

240. SLSC renders [1.1] obvious.

241. Per Starner, and as described above in Ground 1A, SLSC collects biometric data (*at least one context data element*) including image(s) of a user's eyes. *Supra*, §IX.B.1[1.1]. For example, Starner explains that a “first computing device acquires biometric data via one or more biometric sensors associated with the first computing device.” EX1004, [0005], [0076]. Per Starner, the SLSC device is not limited to authenticating based on eye-scan data, but instead contemplates various forms of biometric data including iris scans, “fingerprint data, hand/finger geometry, vein geometry, and voiceprints.” EX1004, [0025]. Shaashua similarly explains that the authentication server “obtains biometric authentication factors from [a] user 30 via [a] smart device 22 and derives current user data from the biometric authentication factors.” EX1005, 8:38-41; *supra*, §VIII.C.

242. Per Shaashua, in SLSC, a predetermined authentication policy determines the context data elements to be collected. For example, Shaashua explains that the “[d]etermination of which biometric methods from the set of

suitable biometric methods are to be used (*i.e.*, which of the above-described arrangements should be used) can be dictated by a ***predefined policy***.” EX1005, 9:2-6, 11:1-28 (“one or more of the suitable biometric methods (e.g., based on policy) is applied during authentication”). Shaashua’s policy is predetermined; indeed, the policy informs the selection of biometric method(s) and therefore must be defined even before the biometric method(s) are selected or applied. Note that I explained a POSITA’s understanding of “predefined” in the context of element [1.1] in Ground 1A. EX1023, 3; *supra*, §IX.B.1[1.1].

243. Shaashua further explains that, in some “arrangements, the authentication server 24 applies the biometric method of the set of suitable biometric methods having the highest suitability score,” which “may be appropriate in situations in which the users 30 should not be overburdened during biometric collection (*e.g.*, it could be too burdensome to ask a user 30 to concurrently provide a facial image, a voice sample, a fingerprint scan, and a handwriting sample all in the same authentication attempt).” EX1005, 8:52-60. Shaashua explains that the predefined policy determines which biometric methods are used to avoid “overburden[ing] [users] during biometric collection.” *Id.* Accordingly, a POSITA would have understood or found obvious that the policy determines which biometric method(s) are used and which corresponding biometric data samples (e.g., context data elements) are collected.

244. Per Shaashua, in some examples, SLSC’s *predetermined authentication policy* directs the collection of one or more biometric factors having the highest suitability score (*collecting at least one context data element*), such as “eye-scan data received from one or more sensors coupled to the HMD.” EX1004, [0074]; EX1005, 8:55-60, 9:1-6, 11:1-13.

2. Claim 4

Element [4] The method according to claim 1, wherein, prior to generating the hash, the method further comprises weighting, according to the authentication policy, the at least one context data element.

245. SLSC renders [4] obvious.

246. For example, Shaashua explains that “the best suited biometric method can be selected based on a variety of factors including device factors (*i.e.*, what circuitry is available to capture biometric data), environmental factors (*i.e.*, the states of ambient conditions), and even biometric method strength factors (*i.e.*, ***weights*** defining particular security strengths of the available biometric methods).” EX1005, 11:1-27; *supra*, §VIII.C. The “[d]etermination of which biometric methods from the set of suitable biometric methods are to be used ... can be dictated by a predefined policy,” that defines a suitability score “based on (i) [a] set of suitability factors and (ii) a biometric method strength factor which defines a particular security strength for that available biometric method relative to other security strengths for other available biometric methods.” EX1005, 3:28-33; *see also id.*, 1:36-45, 2:3-13, 3:18-

25, 6:62-6:10. Thus, per Shaashua, SLSC weights the biometric data/factors (*at least one context data element*), which can include eye-scan data and other biometric data as taught in Starner and Shaashua, before collecting the data. Since the weighting occurs to select which biometric methods to apply and which biometric data to collect, SLSC weights the biometric data even before a data packet is constituted from the biometric data and hashed.

247. In SLSC, the weights are defined by rules in the SLSC authentication policy. (As I described in §VI.B.2, a policy can be a collection of rules. *Supra*, §VI.B.2; EX1045, [0058]; EX1042, [0006]; EX1047, [0031]; EX1048, [0007]; EX1035, 6.) For example, per Shaashua, in SLSC, the “[d]etermination of which biometric methods from the set of suitable biometric methods are to be used (*i.e.*, which of the above-described arrangements should be used) can be dictated by a predefined policy,” that defines a suitability score “based on (i) the set of suitability factors and (ii) a biometric method strength factor which defines a particular security strength for that available biometric method relative to other security strengths for other available biometric methods.” EX1005, 3:28-33. The “biometric method strength factors” are determined by “*weights* defining particular security strengths of the available biometric methods).” EX1005, 2:3-13.

248. In addition, Shaashua explains that “the authentication server 24 is able to choose among different biometric methods which are available for use. Such

selection is based on a variety of factors including (i) device factors (*i.e.*, what capabilities are available on the smart devices 22 of the users 30), (ii) environmental factors (*i.e.*, what aspects of the current environments 32 affect suitability the available biometric methods), and (iii) biometric method strength (*i.e.*, certain biometric methods are more accurate), perhaps among other factors.” EX1005, 6:62-7:10. Additionally, “the authentication server 24 rules out (*i.e.*, eliminates) unsuitable biometric methods that would not work, be ineffective or provide poor results based on these factors.” *Id.* For example, as Shaashua explains, “suppose that a human tries to speak into a microphone for voice recognition but the current environment is very noisy,” then “although the human may be authentic (*i.e.*, the actual legitimate person so that authentication should succeed), authentication may fail due to the noisy environment.” EX1005, 1:36-45.

249. In such cases, SLSC weights a first biometric factor, *e.g.*, an eye scan (at least one context data element) at 100%, and rules out another factor (*e.g.*, voice recognition), SLSC weighting it at 0%.

3. Claim 5

Element [5] The method according to claim 1, wherein, prior to constituting the data packet, the method further comprises grouping together, according to the authentication policy, at least two context data elements comprised within the at least one context data element, to constitute at least one corresponding context data group.

250. SLSC renders [5] obvious for the reasons discussed above with respect to [5] in Ground 1A. *Supra*, §IX.B.4.

251. Additionally, per Shaashua, SLSC applies a predetermined authentication policy as discussed above with respect to [1.1] in Ground 1B (*supra*, §IX.B.1[1.1]), and the policy groups different biometric methods and corresponding context data elements. For example, Shaashua discloses “applying only the top N highest ranked biometric methods ... during the authentication operation,” thus grouping together biometric data/factors (context data elements) for the top N highest ranked biometric methods to the exclusion of biometric data for other methods. EX1005, 8:61-9:6. Shaashua also groups biometric data for different biometric methods based on their relation to device capabilities, environmental conditions, and the like: “By way of example, facial recognition, handwriting analysis, and voice recognition have relatively high suitability scores perhaps due to the availability of certain capabilities on the smart device 22 of the user 30, optimal environmental conditions, and strength (based on weights) of these particular biometric methods.” EX1005, 9:22-33.

4. Claim 6

Element [6] The method according to claim 1, wherein, prior to generating the hash, the method further comprises ciphering the data packet, according to the authentication policy.

252. SLSC renders [6] obvious the reasons discussed above with respect to [6] in Ground 1A. *Supra*, §IX.B.5. Additionally, per Shaashua, SLSC applies a predetermined authentication policy as discussed above with respect to [1.1] in Ground 1B. *Supra*, §IX.B.1[1.1].

5. Claim 8

Element [8.1] collect at least one context data element, based on a predetermined authentication policy;

253. SLSC renders [8.1] obvious for the reasons discussed above with respect to [1.1] in Ground 1B. *Supra*, §IX.B.6[8.1].

6. Claim 10

Element [10.3] wherein the first authentication device is configured to: collect at least one context data element, based on a predetermined authentication policy;

254. SLSC renders [10.3] obvious for the reasons discussed above with respect to [1.1] in Ground 1B. *Supra*, §X.B.1[1.1].

XI. GROUND 1C: CLAIMS 6-7 ARE OBVIOUS IN VIEW OF THE STARNER-LESKOVEC-BOWMAN COMBINATION

A. Combining Starner-Leskovec with Bowman

1. Overview of the Combination (SLBC)

255. Per Starner, SLC includes a computing device, *e.g.*, an HMD, and an authentication server. EX1004, Abstract; §VIII.A. Per Starner, when SLC “cannot

find a good match for the sections of the wearer's eye, the HMD can begin operating in a locked mode (e.g., a guest-operation mode) and request the user to re-authenticate.” EX1004, [0096]. Starner further explains that “[l]imiting guest-operation mode functionality may help to prevent unauthorized users from gaining access to personal information saved on the computing device.” EX1004, [0099].

256. A POSITA would have understood or found obvious that when a user begins to wear the HMD (e.g., as the day begins), since the user has not yet been authenticated, the device can begin in guest mode. EX1004, [0096]. For example, Starner explains that the login procedure can begin in response to “a voice command such as ‘Ok, HMD, log me in.’” EX1004, [0070]. Per Starner, the device then gathers biometric data to enable wearer verification. EX1004, [0096].

257. However, to the extent Starner does not provide details on how the HMD prevents access to personal information, e.g., the biometric data collected to authenticate the user when the device becomes active, a POSITA would have looked to references such as Bowman, which explains that an “electronic device can encrypt the data received or generated during the locked state and store the encrypted data in a secured storage on the electronic device.” EX1009, [0012]. The “encrypted data cannot be decrypted by entities other than the holder of a master private key that is associated with a public key used to encrypt the data.” EX1009, [0012].

258. In the resulting Starner-Leskovec-Bowman Combination (SLBC), when a user begins wearing a locked device (e.g., as the day begins), SLBC collects eye-scan data to authenticate the user. EX1004, [0076]-[0078], [0096]. Since the HMD is locked, the HMD encrypts the eye-scan data and representations thereof (e.g., the digital template) “generated during the locked state and store[s] the encrypted data in a secured storage on the electronic device.” EX1009, [0012]. The “encrypted data cannot be decrypted by entities other than the holder of a master private key that is associated with a public key used to encrypt the data.” EX1009, [0012].

259. Additionally, per Bowman, SLBC uses a current secret key to encrypt the biometric data (e.g., the digital template of the user’s eye-scan data), and the current secret key is generated by applying to a shared secret “a hash, a key derivation function (KDF), or any other cryptographic one-way function,” such as “the National Institute of Standards and Technology (NIST) KDF Alternative 1 using SHA-512.” EX1009, [0037]-[0038], [0014]. To ensure the integrity of the encrypted data, “a keyed-hash message authentication code (HMAC) algorithm, e.g., the HMAC-SHA-256 algorithm,” may be “applied to the current ciphertext portion to generate a HMAC tag for the current portion.” EX1009, [0041]. SLBC uses the

HMAC¹⁴ tags “to authenticate the encrypted data during the decryption process.” EX1009, [0041]. Per Starner, the decrypted data are used to authenticate the user. EX1004, [0078], [0085]-[0087], [0092], [0096].

260. Once the user has been authenticated, SLBC operates analogously to SLC. §§XIIIA-B; §IX.A. For example, per Starner, “only an initial eye-scan may be needed from the owner of the HMD,” rendering “repeated eye-scanning unnecessary.” EX1004, [0094]; §VIII.A. When accessing secured information, SLBC hashes a digital template of the eye-scan data, transmits the hashed digital template (*e.g.*, a digital signature) for authentication, etc. §§VIII.A, §IX.A, §IX.B.1.

2. Motivations to Combine

261. The following reasons, taken separately or collectively, support the above-described Starner-Leskovec-Bowman Combination (SLBC).

262. First, the resulting combination provides improved privacy since (per Bowman), eye-scan data are stored securely when the device is in a locked state. §VIII.D.

¹⁴ Bowman has an obvious typographical error, writing “HAMC” instead of “HMAC.” EX1009, [0041].

263. Second, while Starner explains that functionality of a locked HMD is limited, Bowman provides implementation details enabling such limitation, e.g., by using HMAC-SHA-256. EX1004, [0076]-[0078], [0096]; EX1009, [0041].

264. Third, by preventing the encrypted data from being decryptable until the private key holder returns (per Bowman), the HMD can be made to comply with common security standards and policies. EX1009, [0003] (“National Information Assurance Partnership (NIAP) requires that data be encrypted with public-key cryptography and that, once the data is encrypted, the encrypted data not be decryptable until the private key holder returns.”).

265. Fourth, SLBC would beneficially attain multiple technical and security advantages as described in Bowman. For example, Bowman explains that “[e]ncrypting the current secret key to generate the current encrypted key” advantageously allows “the current secret key [to be] recovered by decrypting the current encrypted secret key,” thereby “accelerat[ing]” the decrypting process. EX1009, [0077]. “The time and processing power saved may be significant as the number of chained keys increases.” *Id.* “Furthermore, during a locked state, the context key is deleted” so that “other entities cannot recover the current secret key from the current encrypted secret key without the master private key.” *Id.*; *see also id.*, [0015]-[0016].

266. Fifth, a POSITA would have been led to SLBC because it merely involves the application of a known technique (*e.g.*, Bowman's technique for securely encrypting and storing data) to a known system (*e.g.*, Starner's biometric authentication system in light of Leskovec's techniques) to achieve predictable results (*e.g.*, secure storage of data received in a locked state). §§VIII.A-B; §VIII.D; §IX.A. Each element in combination merely performs the same or substantially similar function as it does in the references individually. *Id.* There was a known problem, how to secure data received while the device is in a locked state, Bowman's techniques helped address that issue, and combining the teachings of Starner, Leskovec and Bowman was not beyond the skill of an ordinary artisan.

3. Reasonable Expectation of Success

267. A POSITA would have reasonably expected success implementing SLBC, which would have required routine programming knowledge well within the skill of a POSITA. For example, SLBC includes a known secure storage system, known cryptographic algorithms, and a known biometric authentication system. §XI.A. Integrating these aspects of Starner, Leskovec and Bowman would have been well within the skill of a POSITA by the Critical Date.

B. Analysis with Respect to Claims 6-7

1. Claim 6

Element [6] The method according to claim 1, wherein, prior to generating the hash, the method further comprises ciphering the data packet, according to the authentication policy.

268. SLBC renders [6] obvious.

269. SLBC ciphers the data packet that contains the eye-scan data collected to authenticate the user, e.g., when the user first wears the HMD, or otherwise before the user is authenticated and the HMD is in a locked state. §XI.A. For example, Bowman explains that the “electronic device can encrypt the data received or generated during the locked state and store the encrypted data in a secured storage on the electronic device.” EX1009, [0012]. Further, Starner explains that the login procedure (before which, the device is locked) can begin in response to “a voice command such as ‘Ok, HMD, log me in.’” EX1004, [0070]. In SLBC, a digital template of eye-scan data that is to be cached or stored temporarily before the user is authenticated and the device fully unlocked is ciphered (e.g., encrypted) according to Bowman’s techniques. *Id.* It would have been obvious for SLBC to cipher the digital template before it is hashed and sent to an authentication server to improve security according to Bowman’s teachings. *Supra*, §XI.A.1. For example, per Bowman, the HMD may “encrypt the data received or generated during the locked state and store[s] the encrypted data in a secured storage on the electronic device.” EX1009, [0012] *see also id.*, Abstract, [0037]-[0038], [0014]. Additionally, the HMD may encrypt and store the digital template of the biometric data temporarily

while the device (*e.g.*, HMD) waits to establish a network connection by which the signature can be sent to the server, or to allow re-use of the digital template (*e.g.*, to allow the same digital template to be sent to multiple servers to authenticate to different services). *Id.* Thus, in SLBC, prior to generating the hash, the method further comprises ciphering the data packet. *Id.*; *supra*, §IX.B.1[1.3]. The ciphering occurs according to the authentication policy, which per Bowman, requires such encryption and storage to protect “data received or generated during the locked state” to prevent unauthorized access to the user’s biometric data. EX1009, [0012].

2. Claim 7

Element [7] The method according to claim 6, wherein a key used for the data packet ciphering includes a key generated by using a hash-based message authentication code key type derivation function.

270. SLBC renders [7] obvious.

271. Per Bowman, SLBC uses a secret key to encrypt biometric data (*e.g.*, a digital template of the user’s eye-scan data), where the secret key is generated by applying to a shared secret “a hash, a key derivation function (KDF), or any other cryptographic one-way function.” EX1009, [0037]-[0038], [0014]; §XI.A. For example, “(NIST) KDF Alternative 1 using SHA-512 can be used to generate the current secret key.” EX1009, [0037]. A POSITA also would have understood that hash-based message authentication code key type derivation functions were well known and commonly implemented by the Critical Date of the ’982 Patent. For

example, Krawczyk explains that a “a simple Hashed Message Authentication Code (HMAC)-based key derivation function (HKDF)[] ... can be used as a building block in various protocols and applications. The key derivation function (KDF) is intended to support a wide range of applications and requirements.” EX1016, 1. Krawczyk further explains that “HKDF is intended for use in a wide variety of KDF applications.” EX1016, §4. The selection of HKDF would have been an obvious design choice in light of Bowman’s direction to use any suitable “hash, a key derivation function (KDF), or any other cryptographic one-way function.” EX1009, [0037]-[0038]. A POSITA also would have been motivated to apply an HKDF for generating the current secret key in SLBC due to its “simplicity and multi-purpose nature ..., as well as [] its analytical foundation,” as Krawczyk explains. EX1016, §4.

272. Per Bowman, SLBC’s ciphering operations would also include “generat[ing] [an] HMAC tag for the current portion” of the data packet ciphertext (e.g. the encrypted digital template of eye-scan data) for purposes of verifying “data integrity.” EX1009, [0041]. For example, Bowman explains that “a keyed-hash message authentication code (HMAC) algorithm, e.g., the HMAC-SHA-256 algorithm,” is “applied to the current ciphertext portion to generate a HMAC tag for the current portion. EX1009, [0041]. “An example of the data integrity algorithm is a keyed-hash message authentication code (HMAC) algorithm, e.g., the HMAC-

SHA-256 algorithm.” *Id.* According to Bowman, “the current HMAC key can [] be used as input for the KDF to generate a subsequent HMAC key.” *Id.*, [0042]; *see also id.*, [0043]-[0044]. Accordingly, a current HMAC key used for ciphering the data packet by generating an HMAC tag includes a key generated by using an HKDF. (Krawczyk explains HMAC KDFs in IETF RFC 5869. EX1059, 1-14.)

273. Additionally, since Starner explains that the HMD is accessing webpages, and by the Critical Date, a majority of web traffic was transmitted over HTTPS, a POSITA would have understood and found obvious to cipher such data using TLS, which uses keys established using HKDF. See, for example, Bowman, Rescorla, Rescorla-2 describing HTTPS and TLS (EX1009, [0096]; EX1014, 6, 40, 74-78; EX1015, 1¹⁵) and “Let’s Encrypt Stats” and “HTTPS encryption on the web” describing the prevalence of HTTPS as of the Critical Date. EX1061, 1-2; EX1062, 1-4.

XII. GROUND 1D: CLAIMS 1, 6-7 ARE OBVIOUS IN VIEW OF THE STARNER-LESKOVEC-SHAASHUA-BOWMAN COMBINATION

¹⁵ A POSITA would have understood that HTTPS is HTTP over TLS.

“HTTP/TLS is differentiated from HTTP URIs by using the 'https' protocol identifier in place of the 'http' protocol identifier. An example URI specifying HTTP/TLS is: <https://www.example.com/~smith/home.html>.” EX1015, 4

274. For the reasons addressed above in Ground 1B (*supra*, §X), the combined teachings of Starner in view of Leskovec and Shaashua render claim 1 obvious. It also would have been obvious to further modify the Starner-Leskovec-Shaashua combination of Ground 1B in view of Bowman's teachings for the same reasons that it would have been obvious to modify the Starner-Leskovec combination in view of Bowman as provided in Ground 1C. *Supra*, §X.A. The resulting Starner-Leskovec-Shaashua-Bowman combination would have rendered claims 6 and 7 obvious for the reasons discussed above in Ground 1C. *Supra*, §X.B. Additionally, per Shaashua, SLSC applies a predetermined authentication policy as discussed above with respect to [1.1] in Ground 1B, and thus ciphers the data packet according to this authentication policy. *Supra*, §X.B.1[1.1].

XIII. CONCLUSION

275. For all the reasons I have noted in the foregoing paragraphs, claims 1-10 of the '982 Patent are obvious in view of the references discussed above.

276. I currently hold the opinions set expressed in this declaration. But my analysis may continue, and I may acquire additional information and/or attain supplemental insights that may result in added observations.