

Filed on behalf of Google LLC by:

Paper No. ____

Elisabeth H. Hunt, Reg. No. 67,336
Richard F. Giunta, Reg. No. 36,149
Marc S. Johannes, Reg. No. 64,978
WOLF, GREENFIELD & SACKS, P.C.
600 Atlantic Avenue
Boston, MA 02210
(617) 646-8000 Phone
(617) 646-8646 Fax

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,
Petitioner,

v.

SECURITY FIRST INNOVATIONS, LLC
Patent Owner.

Case No. IPR2024-00212
Patent No. 11,178,116

**PETITION FOR INTER PARTES REVIEW
UNDER 35 U.S.C. §§ 311-319 AND 37 C.F.R. § 42.1 et seq**

TABLE OF CONTENTS

MANDATORY NOTICES.....	xi
A. Real Party-In-Interest	xi
B. Related Matters	xi
1. United States Patent & Trademark Office	xi
2. U.S. District Court for the Eastern District of Virginia.....	xii
C. Counsel and Service Information - § 42.8(b)(3) and (4)	xii
I. STANDING	1
II. UNPATENTABILITY GROUNDS.....	1
III. The '116 Patent.....	2
A. Specification.....	2
B. Person of Ordinary Skill in the Art (“POSA”).....	4
C. Prosecution History	4
D. Challenged Claims	5
IV. CLAIM INTERPRETATION	7
V. <u>GROUND 1</u> : CLAIMS 1 AND 5-8 ARE ANTICIPATED OR ALTERNATIVELY OBVIOUS OVER TAJIMA.....	8
A. Tajima (EX1005).....	8
B. Claim 1	13
1. 1[PRE] A method for securing a data set	14
2. [1A] distributing the data set into a plurality of data chunks, wherein none of the data chunks are, by themselves, sufficient to reconstruct the data set	15
3. [1B] encrypting each of the data chunks with a respective one of a plurality of encryption keys.....	18
4. [1C] obfuscating each of the plurality of different encryption keys	19
5. [1D] separately storing each data chunk of the plurality of data chunks together with one of the plurality of obfuscated different encryption keys on a plurality of different storage devices	23

a.	Tajima meets [1D] the same way the '116 patent's embodiments do	23
b.	[1D] only requires storing each data chunk/key pair once	25
c.	If [1D] were construed to require storing each data chunk/key pair on multiple different storage devices, it is met by an obvious implementation of Tajima	28
C.	Claim 5: wherein storing...comprises storing each of the plurality of data chunks together with a respective one of the plurality of obfuscated different encryption keys	29
D.	Claim 6: wherein storing...comprises including, with a data chunk of the plurality of data chunks, data indicative of an encryption key that was used to encrypt a different data chunk of the plurality of data chunks.....	30
E.	Claim 7: wherein distributing the data...comprises using a random technique or pseudorandom technique.....	31
F.	Claim 8: further comprising generating the data set by encrypting primary data.....	32
VI.	<u>GROUND 2: CLAIMS 1-2, 9, 11-14 WOULD HAVE BEEN OBVIOUS OVER TAJIMA AND FOSTER</u>	33
A.	Security Risk of Plaintext Keys	33
B.	Reasons to Encrypt Tajima's Keys	34
C.	Foster's Key-Encryption Technique	36
D.	Tajima-Foster Combination	37
E.	Reasonable Expectation of Success	39
F.	Claim 1	40
1.	1[PRE]-[1D].....	40
2.	[1C] obfuscating... ..	41
G.	Claim 2: wherein obfuscating...comprises performing a transform on data indicative of at least one of the plurality of different encryption keys	42
H.	Claim 9	43
1.	9[PRE]-[9B].....	43

2.	[9C] transforming each of the plurality of different encryption keys	43
3.	[9D] separately storing each data chunk of the plurality of data chunks together with one of the plurality of transformed different encryption keys on a plurality of different storage devices	44
I.	Claim 11: storing each of the plurality of data chunks with a respective one of the plurality of transformed different encryption keys.....	44
J.	Claim 12: including, with a data chunk of the plurality of data chunks, data indicative of an encryption key of the plurality of encryption keys that was used to encrypt a different data chunk of the plurality of data chunks”	45
K.	Claim 13: distributing the data set...using a random technique or pseudorandom technique”	45
L.	Claim 14: further comprising generating the data by encrypting primary data.....	46
VII.	<u>GROUND 3: CLAIMS 9-14 WOULD HAVE BEEN OBVIOUS OVER TAJIMA AND RIVEST</u>	46
A.	Rivest.....	46
B.	Tajima-Rivest Combination	46
C.	Claim 9	47
D.	Claim 10	48
E.	Claims 11-14	48
VIII.	<u>GROUND 4: CLAIMS 1-3, 6 WOULD HAVE BEEN OBVIOUS OVER TAJIMA, RIVEST AND SHYU</u>	48
A.	Tajima-Rivest-Shyu Combination	48
B.	Claim 1	49
C.	Claim 2	50
D.	Claim 3:.....	50
E.	Claim 6	50
IX.	<u>GROUND 5: CLAIMS 1-2, 5, 9 AND 11 WOULD HAVE BEEN OBVIOUS OVER FOSTER AND HAYHURST</u>	51

A.	Foster (EX1006).....	51
B.	Hayhurst (EX1007)	54
1.	Hayhurst Disclosure.....	55
2.	Hayhurst was effectively filed as of the Hayhurst-Provisional	58
a.	Hayhurst-Provisional Supports Hayhurst’s Claim 1	59
b.	Hayhurst-Provisional Supports Hayhurst’s Relied-Upon Subject-Matter	61
C.	Foster-Hayhurst.....	61
1.	Reasons to Combine	61
2.	The Foster-Hayhurst Combination	64
3.	Reasonable Expectation of Success.....	67
D.	Claim 1	67
1.	1[PRE]	67
2.	[1A] distributing... ..	68
3.	[1B] encrypting... ..	70
4.	[1C] obfuscating... ..	71
5.	[1D] separately storing... ..	72
a.	Foster-Hayhurst meets [1D] the same way the ’116 patent’s embodiments do	72
b.	Foster-Hayhurst meets [1D] if construed to require redundantly storing	74
E.	Claim 2	74
F.	Claim 5:	75
G.	Claim 9	75
1.	9[PRE]-[9B].....	75
2.	[9C] transforming... ..	75
3.	[9D] separately storing... ..	76
H.	Claim 11	77
X.	No Basis Exists for Discretionary Denial.....	77
A.	Section 314(a)	77

1. Stay Potential	78
2. Trial Timing	78
3. Litigation Investment	78
4. Issue Overlap	79
5. Litigation Defendants	79
6. Other Circumstances	79
B. Section 325(d)	79
XI. CONCLUSION	80
XII. APPENDIX: CLAIM LISTING	82

TABLE OF AUTHORITIES

Cases

<i>10X Genomics, Inc. v. Bio-Rad Labs.</i> , IPR2020-00086, Paper 8 (Apr. 27, 2020).....	7
<i>Apple Inc. v. Fintiv, Inc.</i> , IPR2020-00019, Paper 11 (Mar. 20, 2020)	77, 78
<i>Cisco Sys., Inc. v. Ramot at Tel Aviv Univ. Ltd.</i> , IPR2022-01283, Paper 12 (Feb. 22, 2023).....	80
<i>Dish Network LLC v. Broadband iTV, Inc.</i> , IPR2020-01332, Paper 14 (Jan. 27, 2021).....	78
<i>DJI Europe BV v. Textron Innovations, Inc.</i> , IPR2022-00453, Paper 10 (Aug. 3, 2022)	80
<i>Dynamic Drinkware v. Nat’l Graphics</i> , 800 F.3d 1375 (Fed. Cir. 2015)	58
<i>Huawei v. Optis Cellular</i> , IPR2018-00807, Paper 10 (PTAB Nov. 5, 2018).....	58
<i>Intel Corp. v. PACT XPP Schweiz AG</i> , 61 F.4th 1373 (Fed. Cir. 2023)	64
<i>KSR Int’l Co. v. Teleflex Inc.</i> , 550 U.S. 398 (2007).....	passim
<i>Markforged Inc. v. Continuous Composites</i> , IPR2022-00679, Paper 7 (Oct. 25, 2022)	79
<i>Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.</i> , 868 F.3d 1013 (Fed. Cir. 2017)	7
<i>PEAG LLC d/b/a JLab Audio v. VARTA Microbattery GmbH</i> , IPR2020-01214, Paper 8 (Jan. 6, 2021).....	78
<i>Quest USA v. PopSockets</i> , IPR2018-00497, Paper 8 (PTAB Aug. 13, 2018).....	54

<i>Sand Revolution II, LLC, v. Cont’l Intermodal Group–Trucking,</i> IPR2019-01393, Paper 24 (June 16, 2020).....	78, 79
<i>ShenZhen Apaltek Co., Ltd. v. Asetek Danmark A/S,</i> IPR2022-01317, Paper 7 (Feb. 6, 2023)	79
<i>SynQor, Inc. v. Artesyn Tech., Inc.,</i> 709 F.3d 1365 (Fed. Cir. 2013)	26

Regulations

37 C.F.R. §42.100(b)	7
37 C.F.R. §42.104(a).....	1

Statutes

35 U.S.C. § 102	1, 54
35 U.S.C. § 103	1
35 U.S.C. § 119	58
35 U.S.C. § 314(a)	77
35 U.S.C. § 325(d)	79

APPENDIX LISTING OF EXHIBITS

Exhibit	Description
1001	U.S. Patent No. 11,178,116 (“’116 patent”)
1002	Prosecution History of U.S. Patent No. 11,178,116
1003	Declaration of Samrat Bhattacharjee, Ph.D.
1004	Curriculum Vitae of Samrat Bhattacharjee, Ph.D.
1005	U.S. Patent Application Publication 2003/0028493 (“Tajima”)
1006	U.S. Patent Application Publication 2003/0200176A1 (“Foster”)
1007	International Application Publication WO 2004/008289A2 (“Hayhurst”)
1008	U.S. Provisional Application 60/396357 (“Hayhurst-Provisional”)
1009	Coversheet of U.S. Provisional Application 60/396357
1010	Application Data for Provisional Application 60/396357
1011	Security First Innovation’s 2023-08-14 Infringement Contentions, Chart A in <i>Security First Innovations, LLC v. Google LLC</i> , 2-23-cv-00097 (E.D. Va.)
1012	Security First Innovation’s 2023-08-14 Infringement Contentions, Chart C in <i>Security First Innovations, LLC v. Google LLC</i> , 2-23-cv-00097 (E.D. Va.) (“Contentions”)
1013	U.S. Patent Application Publication 2004/0049687 (“Orsini”)
1014	International Application Publication WO 2001/22322 (“Dickinson”)
1015	U.S. Patent No. 10,452,854 (“’854 patent”)
1016	U.S. Patent No. 6,021,391 (“Shyu”)
1017	U.S. Patent Application Publication 2004/0049700 (“Yoshida”)
1018	U.S. Patent Application Publication 2004/0003272A1 (“Bantz”)
1019	International Application Publication WO 2001/015162A2 (“Benaloh”)
1020	U.S. Patent Application Publication 2002/0194209 (“Bolosky”)
1021	U.S. Patent No. 6,819,766 (“Weidong”)
1022	U.S. Patent No. 4,238,853 (“Ehrsam”)
1023	U.S. Patent No. 7,277,941 (“Ignatius”)

Exhibit	Description
1024	U.S. Patent Application Publication 2005/0216754 (“Ehud”)
1025	International Application Publication WO 2004/057461A2 (“Ogihara”)
1026	U.S. Patent No. 7,178,033 (“Garcia”)
1027	U.S. Patent No. 6,490,353 (“Tan”)
1028	U.S. Patent Application Publication No. 2003/0167314 (“Marking”)
1029	U.S. Patent No. 7,529,834 (“Birrell”)
1030	U.S. Patent Application Publication 2001/0037474A1 (“Steegmans”)
1031	U.S. Patent Application Publication 2004/0172476A1 (“Chapweske”)
1032	U.S. Patent No. 7,099,994 (“Thayer”)
1033	U.S. Patent No. 6,151,659 (“Solomon”)
1034	R. L. Rivest, “ <i>All-or-nothing encryption and the package transform</i> ,” Proc. Of the 4 th Intl. Workshop on Fast Software Encryption 1997, (E. Biham, ed.), Lecture Notes in Computer Science, 1267 (1997) pp. 210–218. (“Rivest”)
1035	U.S. Patent No. 6,308,277 (“Vaeth”)
1036	U.S. Patent Application Publication 2001/0053221A1 (“Takeda”)
1037	Security First Innovation’s 2023-08-14 Infringement Contentions, Chart D in Security First Innovations, LLC v. Google LLC, 2-23-cv-00097 (E.D. Va.)
1038	Rule 16(b) Scheduling Order in <i>Security First Innovations, LLC v. Google, LLC</i> , 2:23-cv-00097-JWK-RJK, Dkt. 54 (July 20, 2023)
1039	U.S. Patent Application Publication 2018/0367509A1 (“116 Publication”)
1040	ABS: The Apportioned Backup System (“Cooley”) https://pdos.csail.mit.edu/archive/6.824-2004/reports/cooley.pdf
1041	U.S. Patent Application Publication 2003/0081790A1 (“Kallahalla”)
1042	Google LLC’s Proposed Claim Terms for Construction in <i>Security First Innovations, LLC v. Google, LLC</i> , 2:23-cv-00097-JWK-RJK (Nov. 11, 2023)

Exhibit	Description
1043	Letter from A. Mayron to D. Perlson re Preliminary Proposed Constructions in <i>Security First Innovations, LLC v. Google, LLC</i> , 2:23-cv-00097-JWK-RJK (Nov. 11, 2023)
1044	U.S. Patent No. 9,338,140 (“140 patent”)

MANDATORY NOTICES

A. Real Party-In-Interest

Petitioner Google LLC is the Real Party-in-Interest.¹

B. Related Matters

1. United States Patent & Trademark Office

The application from which U.S. Patent No. 11,178,116 issued is a Continuation of U.S. Patent Application No. 13/915,081 (U.S. Patent No. 9,992,170), filed June 11, 2023, which is a Division of 13/468,383 (U.S. Patent No. 9,338,140), filed May 10, 2012, which is a Continuation of U.S. Patent Application No. 11/258,839 (U.S. Patent No. 8,266,438), filed October 25, 2005, which claims priority to U.S. Provisional Application No. 60/718,185, filed September 16, 2005, and U.S. Provisional Application No. 60/622,146, filed October 25, 2004.

The following U.S. patent application claims the benefit of priority to U.S. Patent No. 11,178,116:

(i) U.S. Patent Application No. 17/111,262, filed December 3, 2020.

¹ Google LLC is a subsidiary of XXVI Holdings Inc., which is a subsidiary of Alphabet Inc. XXVI Holdings Inc. and Alphabet Inc. are not real parties-in-interest to this proceeding.

2. U.S. District Court for the Eastern District of Virginia

(i) *Security First Innovations, LLC v. Google LLC f/k/a Google Inc.*, Case No. 1:23-cv-00329; (ii) *Security First Innovations, LLC v. Google LLC*, Case No. 2:23-cv-00097.

C. Counsel and Service Information - § 42.8(b)(3) and (4)

Lead Counsel	Elisabeth H. Hunt, Reg. No. 67,336
Backup Counsel	Richard F. Giunta, Reg. No. 36,149 Marc S. Johannes, Reg. No. 64,978
Service Information	<u>E-mail</u> : EHunt-PTAB@wolfgreenfield.com RGiunta-PTAB@wolfgreenfield.com MJohannes-PTAB@wolfgreenfield.com <u>Post and hand delivery</u> : Wolf, Greenfield & Sacks, P.C. 600 Atlantic Avenue Boston, MA 02210-2206 <u>Telephone</u> : 617-646-8000 <u>Facsimile</u> : 617-646-8646

A power of attorney is submitted with the Petition. Counsel for Petitioner consents to service of all documents via electronic mail.

Petitioner requests *inter partes* review of claims 1-3 and 5-14 (“the challenged claims”) of U.S. Patent No. 11,178,116 (“the ’116 patent”) (EX1001).

I. STANDING

Petitioner certifies that the ’116 patent is available for IPR and that Petitioner is not barred or estopped from requesting IPR as to the challenged claims. 37 C.F.R. §42.104(a).

II. UNPATENTABILITY GROUNDS

The table below identifies the references, applicable claims, and basis for each ground of unpatentability.

Ground Number and Reference(s)		Claims	Basis
1	Tajima (EX1005)	1, 5-8	§§102/103
2	Tajima, Foster (EX1006)	1-2, 9, 11-14	§103
3	Tajima, Rivest (EX1034)	9-14	§103
4	Tajima, Rivest, Shyu (EX1016)	1-3, 6	§103
5	Foster, Hayhurst (EX1007)	1-2, 5, 9, 11	§103

Tajima, Foster, Rivest, and Shyu are each prior art under pre-AIA 35 U.S.C. §102(b) because each published more than one year before the ’116 patent’s earliest-alleged priority date of October 25, 2004. Hayhurst is prior art under 35 U.S.C. §§102(a),(e) as discussed in §IX.B *infra*. Tajima, Foster, Hayhurst and Shyu were not cited during prosecution. Rivest was listed in an IDS but was not used to reject any claim.

III. THE '116 PATENT

A. Specification

Of the lengthy specification, the challenged claims most relate to Figs. 21-24 and the discussion at 57:10-61:13 and 77:13-43. For example, Fig. 21 (annotated below) illustrates a process of securing a data set by: (1) dividing the data set into “shares” (the “Parse” step boxed in **green**); (2) encrypting each share using a different encryption key (the “Encrypt Pieces” step boxed in **blue**); and (3) assigning each encryption key to a share it did not encrypt (the “Obfuscate” step boxed in **red**). EX1001, 57:41-63. Bhattacharjee, ¶¶33-34

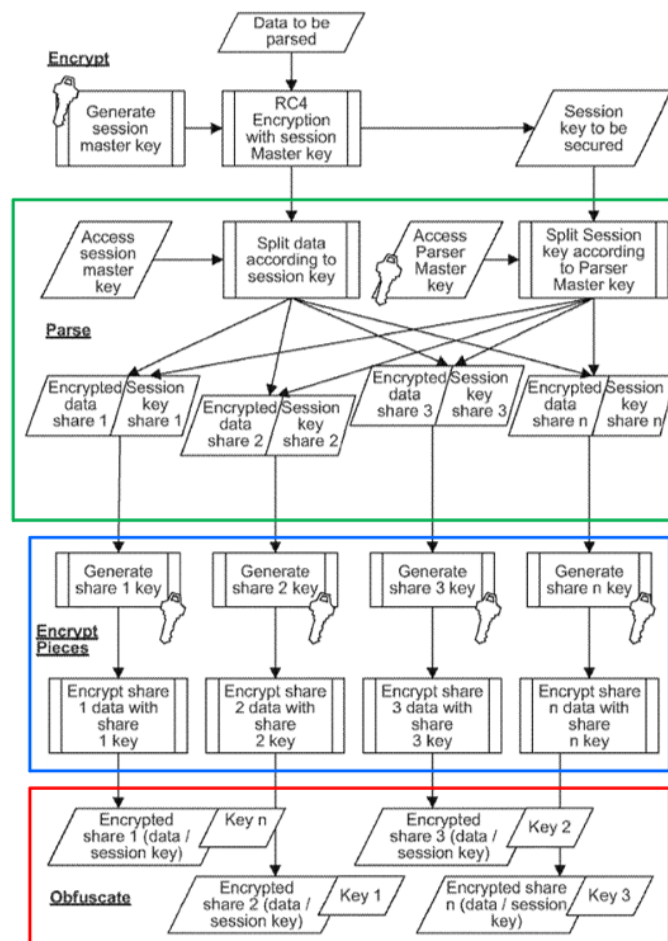


FIG. 21

Steps (1) and (2) relate to the “distributing” and “encrypting” steps in both claims 1 and 9 and step (3) relates to the “obfuscating” step in claim 1 (see §III.D *infra*). Bhattacharjee, ¶¶35-36.

The ’116 patent further discusses storing each encrypted share from step (3) in different locations “so the key required to decrypt the share is stored in a different data depository.” EX1001, 60:49-52. One technique for achieving this is shown in the specification at 60:53-56 (quoted in **purple** below) where SD1-SDn are *n* shares that each include a respective encrypted data portion of a data set S, and where K1-Kn are the *n* respective encryption keys used to encrypt the same-numbered data portions, so that each depository stores a share with a key that was used to encrypt a different share. This discussion relates to the “separately storing” limitations in both claims 1 and 9. Bhattacharjee, ¶¶37-38.

**“Depository 1: SD1, Kn
Depository 2: SD2, K1
Depository 3: SD3, K2
Depository n: SDn, K3.”**

The ’116 patent further explains that prior to parsing, the data set can first be encrypted (the “**Encrypt**” in step in annotated Fig. 21 above). EX1001, 57:45-46. Bhattacharjee, ¶39.

Claim 9 recites “transforming” encryption keys (see §III.D *infra*). The ’116 patent discloses that “an All or Nothing Transform (AoNT)” may be used to

transform “splitting keys, encryption keys, [or] any other suitable data elements” (77:13-17) and that “[i]t will be understood that the AoNT may be applied to encryption keys used to encrypt the data portions following parsing and splitting” (77:37-39). Bhattacharjee, ¶40.

B. Person of Ordinary Skill in the Art (“POSA”)

A POSA at the time of the ’116 patent’s October 25, 2004 earliest claimed priority date would have had at least a bachelor’s degree in computer science, computer engineering, or a related field, with three years of experience in the area of securing data from unauthorized access or use. A higher level of education may substitute for less experience. Bhattacharjee, ¶¶41-45.

C. Prosecution History

The ’116 patent’s application was filed with one claim that was then canceled. A preliminary amendment added claims 2-25 (EX1002, 246-252), which were later amended in response to a rejection over “Dickinson” (EX1014) and “Bolosky” (EX1020). EX1002, 133-155 (Office Action), 117-125 (Amendment). Bhattacharjee, ¶46.

Specifically, claims 2 and 10 (issued claims 1 and 9) were amended as follows (EX1002, 118-119):

“separately storing each data chunk of the plurality of data chunks together with one of the plurality of [obfuscated/transformed] different encryption keys on a plurality of different storage devices.”

Claims 16 and 22 (issued claims 15 and 21) were similarly amended (EX1002, 120-122). Amended claims 2-25 were allowed without further amendment and issued as claims 1-24. EX1002, 89-99. Bhattacharjee, ¶¶47-50.

D. Challenged Claims

Claims 1-3 and 5-14 are reproduced in the Claim Listing (*infra* §XII). The challenged independent claims recite similar subject matter but differ in whether encryption keys are “obfuscated” (claim 1) or “transformed” (claim 9). The below reproduced claim reflects claims 1 and 9 where “**obfuscating**” is in claim 1 but not claim 9, “**transforming**” is in claim 9 but not claim 1, and all other words are in both claims:

[PRE] A method for securing a data set, the method comprising:

[A] distributing the data set into a plurality of data chunks, wherein none of the data chunks are, by themselves, sufficient to reconstruct the data set;

[B] encrypting each of the data chunks with a respective one of a plurality of encryption keys;

[C] **obfuscating/transforming** each of the plurality of different encryption keys; and

[D] separately storing each data chunk of the plurality of data chunks together with one of the plurality of **obfuscated/transformed** different encryption keys on a plurality of different storage devices.

Claim 1 and 9's shared limitations ([A]-[B] and [D]) recite (1) dividing data into "data chunks," (2) encrypting the data chunks with respective encryption keys, and (3) storing the data chunks with encryption keys on different storage devices, all of which was routine and conventional. Bhattacharjee, ¶¶31-32 (*citing* EX1005, [0111], [0113], [0046]-[0048]; EX1014, 20:20-22:7; EX1013, [0293]-[0368]; [0026]-[0031]; EX1018, [0029]-[0038]; EX1040, 3-4), 53.

"Obfuscating" encryption keys (claim 1) was also a known technique as demonstrated by Tajima (EX1005), which in addition to the above-enumerated steps, also discloses storing an encryption key with an encrypted data fragment that the key did not encrypt—i.e., the same way the '116 patent obfuscates. Bhattacharjee, ¶54; §III *supra*. As Ground 1 demonstrates, Tajima anticipates claim 1.

"Transforming" (claim 9) was also a known technique used to secure data, for example, the AoNT that the '116 patent acknowledges was known and obvious to apply to keys. EX1001, 77:13-39. As Ground 3 demonstrates, claim 9 would have been obvious in view of Tajima and Rivest (EX1034) (disclosing AoNT). Bhattacharjee, ¶55.

Additionally, Patent Owner Security First Innovations ("SFI") has taken the position in the district court litigation that both "obfuscating" and "transforming" encryption keys are met by encrypting the encryption keys (see, e.g., EX1012, 5,

12). Encrypting keys was yet another routine and conventional technique for protecting encryption keys, for example, as taught by Foster (EX1006).

Bhattacharjee, ¶56. As Grounds 2 and 5 demonstrate, claims 1 and 9 would have been obvious in view of Tajima and Foster (Ground 2) and Foster and Hayhurst (Ground 5) at least under SFI’s claim interpretations. *Id.*

The challenged dependent claims recite conventional features disclosed in the Grounds’ references. Bhattacharjee, ¶57.

IV. CLAIM INTERPRETATION

Claim terms are construed herein according to their ordinary and customary meaning in light of the specification and prosecution history. 37 C.F.R. §42.100(b). The Board “need only construe terms that are in controversy, and only to the extent necessary to resolve the controversy” regarding unpatentability. *Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017) (internal quotes omitted). In the concurrent district-court proceeding, the parties have served competing proposed constructions of various claim terms. EX1042-EX1043. As discussed in detail where relevant within the Grounds below, this Petition presents alternative mappings of the prior-art disclosures to both parties’ proposed constructions of disputed terms, thus demonstrating unpatentability regardless of which proposed construction is correct. *See, e.g., 10X Genomics, Inc. v. Bio-Rad Labs.*, IPR2020-00086, Paper 8 at 18-22 (Apr. 27,

2020) (“Petitioner proposes alternative claim constructions and presents at least one ground of unpatentability for each construction.”).

V. GROUND 1: CLAIMS 1 AND 5-8 ARE ANTICIPATED OR ALTERNATIVELY OBVIOUS OVER TAJIMA

A. Tajima (EX1005)

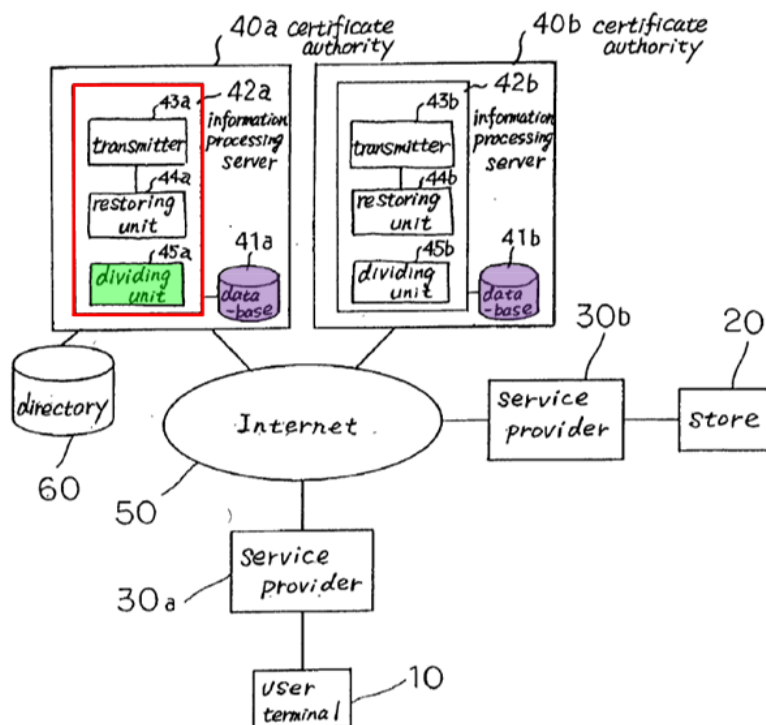
Tajima discloses techniques for securely storing users’ personal information used in online transactions. Tajima, [0067]-[0074], [0100]-[0114]. Tajima noted that previous techniques that stored a user’s personal information at just one location provided insufficient security because all the user’s personal information could be obtained through “unauthorized access from the outside” or by “someone inside the system.” Tajima, [0042]; Bhattacharjee, ¶61.

To address this vulnerability, Tajima’s system (1) divides personal information into data fragments “such that each data fragment is completely unintelligible when taken independently” (Tajima, [0072], [0122]) and (2) “registers” (i.e., stores) each data fragment in a database “under separate control” (Tajima, [0048]) of a respective different service site so that breaching a single service site could procure only one “unintelligible” data fragment. Tajima, [0042], [0046]-[0048], [0067], [0118], [0152]; Bhattacharjee, ¶62.

Tajima’s Fig. 5 (color-annotated below) illustrates a system in which personal information submitted by a user via terminal 10 (e.g., to conduct an online transaction with online store 20) is divided into data fragments that are stored

separately at different service sites (here, certificate authorities) on those sites' respective databases. Tajima, [0048], [0054], [0067]-[0068], [0073]-[0074], [0079]. Bhattacharjee, ¶63.

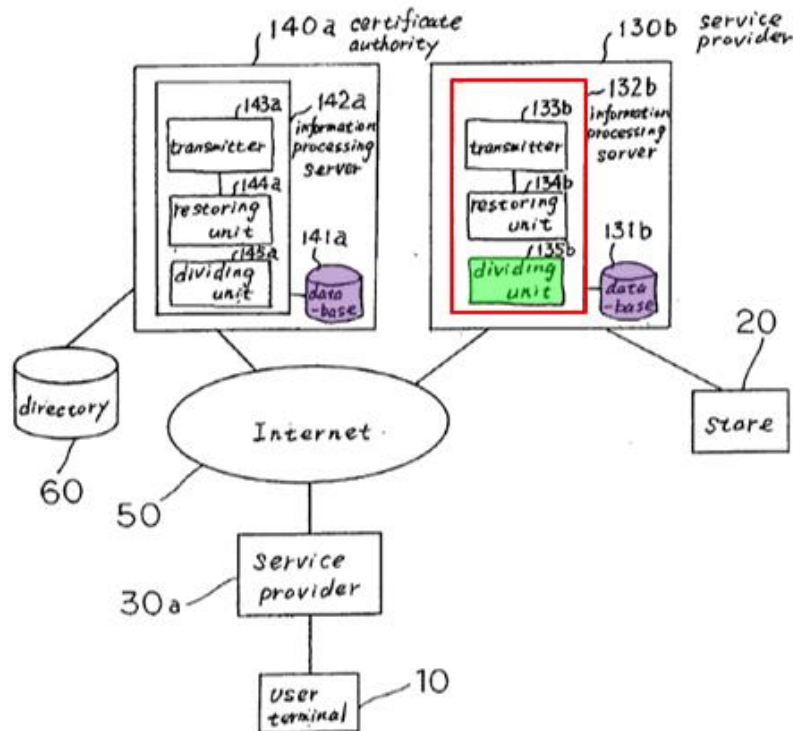
Fig. 5



Specifically, certificate authority 40a comprises server 42a (boxed **red**) that implements dividing unit 45a (highlighted **green**) configured to: (1) divide personal information into data fragments X and Y; (2) store data fragment X on database 41a (highlighted **purple**); and (3) send data fragment Y via Internet 50 to certificate authority 40b for storage on database 41b (also highlighted **purple**).

Tajima, [0067], [0073]-[0074]; Bhattacharjee, ¶64. Tajima's Fig. 14 (color-annotated below) illustrates a similar system but where securing is performed by service provider 130b. Tajima, [0118]; Bhattacharjee, ¶65.

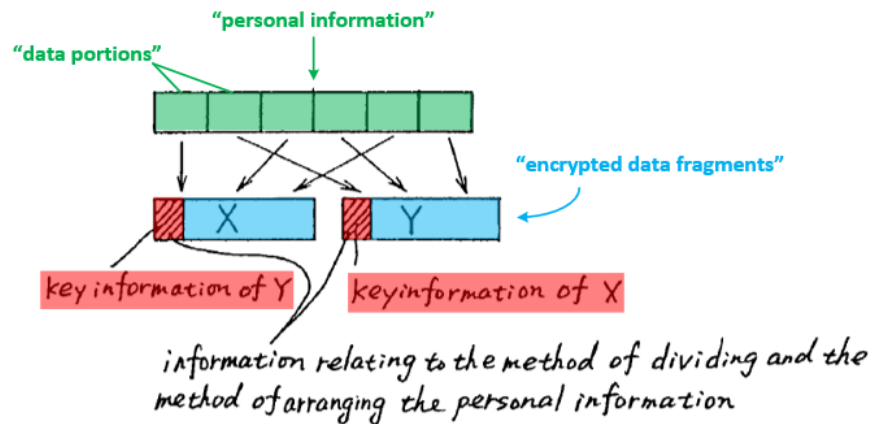
Fig. 14



Each of Tajima's service sites (whether a certificate authority or service provider) has its own information-processing server comprising a dividing unit configured to divide personal information into data fragments that are then distributed to one or more other service sites for storage. Tajima, [0048], [0067]-[0068], [0073]-[0074], [0079], [0093], [0118], [0122]-[0123], [0151]-[0153], [0156]. Bhattacharjee, ¶66. The service site that divided the personal information stores "link information" that identifies where each data fragment is stored so that the service site can retrieve them to recover the personal information. Tajima, [0046]-[0047], [0073], [0093], [0111]-[0116], [0123], [0144], [0153]; Bhattacharjee, ¶67.

Tajima describes different “method[s] of dividing personal information” (Tajima, [0100]) illustrated in Figs. 8-13 that each service site can be configured to perform. Tajima, [0100]-[0116], [0118], [0122]. In Tajima’s Fig. 12 embodiment (annotated below), a user’s personal information is “first divided into a plurality of data portions” (annotated **green**) that are then “arranged into two data fragments, data fragment X and data fragment Y, according to a set method of arrangement.” Tajima, [0111]. Bhattacharjee, ¶¶68-69.

Fig. 12



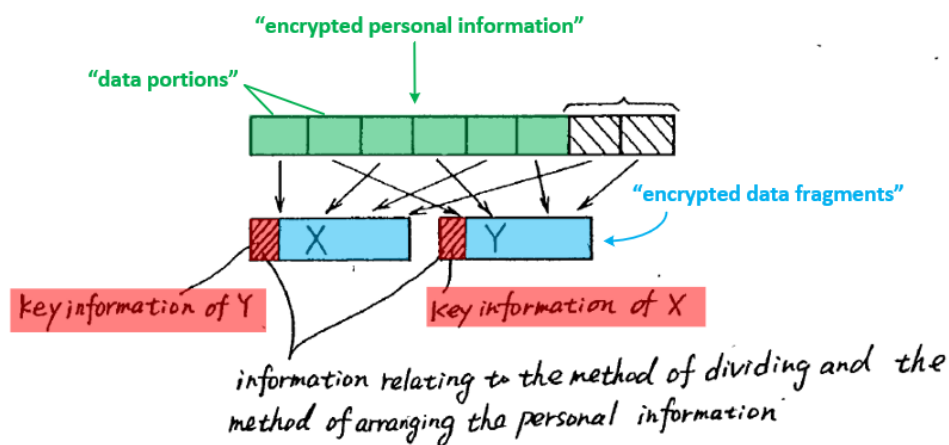
“Data fragments X and Y are then each encrypted” (highlighted **blue**) with a respective key (highlighted **red**) that is then attached to the other fragment—i.e., key information of fragment Y is attached to encrypted data fragment X, and key information of fragment X is attached to encrypted data fragment Y. Bhattacharjee, ¶70.

Finally, encrypted data fragments X and Y (each with key information for the other data fragment) are separately “registered” (i.e., stored) in databases 41a

and 41b “under separate control” of the different certificate authorities 40a and 40b, respectively (highlighted in **purple** in Fig. 5 above). Tajima, [0048], [0111]; Bhattacharjee, ¶71. The above-colored steps in Tajima correspond to the same-colored steps of the ’116 patent’s Fig. 21 embodiment (§III *supra*) and map to the same color-coded limitations in claim 1 in the table in §V.B below. Bhattacharjee, ¶72.

Tajima’s Fig. 13 embodiment (annotated below) is identical to the Fig. 12 embodiment but adds an additional step that mirrors the first “Encrypt” step in Figs. 21-24 of the ’116 patent where the data set can be optionally encrypted before being divided into portions/shares. EX1001, 57:45-46, 58:58-59:10, 59:34-57; Bhattacharjee, ¶73.

Fig. 13



In Tajima’s Fig. 13 embodiment, personal information is “first encrypted, and the encrypted personal information and information on the key that was used

in encrypting the personal information are then divided into a plurality of data portions.” Tajima, [0113]. Thereafter, the same sequence of steps performed on Fig. 12’s “data portions” are performed on Fig. 13’s “data portions.” Tajima, [0111], [0113]; Bhattacharjee, ¶¶74-75.

B. Claim 1

As the table below illustrates by color-coding the correspondence between claim 1 and Tajima’s Fig. 12 embodiment described in [0111], Tajima anticipates claim 1 (as further demonstrated in the subsections below). Bhattacharjee, ¶76.

Claim 1	Tajima
A method for securing a data set, the method comprising: distributing the data set into a plurality of data chunks, wherein none of the data chunks are, by themselves, sufficient to reconstruct the data set; encrypting each of the data chunks with a respective one of a plurality of encryption keys; obfuscating each of the plurality of different encryption keys; and separately storing each data chunk of the plurality of data chunks together with one of the plurality of obfuscated different encryption keys on a plurality of different storage devices.	“[0111] As yet another method of dividing personal information, as shown in FIG. 12, personal information that has been submitted to certificate authority 40a by the user of user terminal 10 is first divided into a plurality of data portions...and this plurality of data portions is then arranged into two data fragments, data fragment X and data fragment Y, according to a set method of arrangement. Data fragments X and Y are then each encrypted, and encrypted data fragment X is then registered with database 41a of certificate authority 40a and encrypted data fragment Y is registered with database 41b of certificate authority 40b. Information regarding the key that was used to encrypt data fragment Y is attached to encrypted data fragment X, and information regarding the key that was used in the encryption of data fragment X is attached to encrypted data fragment Y.”

The Fig. 13 embodiment described in [0113] performs all the same steps on the encrypted personal information (and repeats [0111] nearly verbatim), and therefore meets claim 1 in the same way as the Fig. 12 embodiment, as discussed in §§V.B.1-V.B.5 *infra*. By first encrypting the personal information, the Fig. 13 embodiment also meets claim 8 (see §V.F *infra*). Bhattacharjee, ¶¶77-78.

1. 1[PRE] A method for securing a data set

Tajima discloses that “an object of the present invention” is to provide a “personal information management *method*” that can “*improve the security of personal information*” that is registered with areas that are connectible to the Internet.”² Tajima, [0044]. Such methods for securing personal information that meet 1[PRE] are described in Tajima’s Fig. 12 and 13 embodiments. Tajima, [0111], [0113]; Bhattacharjee, ¶¶79-80.

In Tajima’s Fig. 12 embodiment, personal information being secured meets the claimed “*data set*” under both parties’ proposed constructions because the personal information is a collection of separate elements of information that are related by their connection to the user (e.g., the user’s name, address, telephone number, e-mail address, etc.) that are treated as a unit in handling the data in preparing it for storage at the service sites (see *supra* §V.A). Tajima, [0006], [0067], [0111]-[0114]; EX1042, 9 (Google construing “data set” as “collection of

² All emphases herein are added unless otherwise indicated.

related information made up of separate elements that can be treated as a unit in data handling”); EX1043, 6 (SFI construing as “collection of information for storage”); Bhattacharjee, ¶81.

Tajima’s Fig. 13 embodiment encrypts the personal information that meets the claimed “*data set*” before dividing it. The encrypted personal information (before dividing) meets the claimed “*data set*” because an undivided data set is still a data set after being encrypted. This understanding is evidenced by the ’116 patent’s dependent claim 8 which recites that “*encrypt[ed] primary data*” is still a “*data set.*” Bhattacharjee, ¶82.

Each of Tajima’s Fig. 12 and Fig. 13 securing techniques thus meets 1[PRE]. *Id.*

2. [1A] distributing the data set into a plurality of data chunks, wherein none of the data chunks are, by themselves, sufficient to reconstruct the data set

Tajima’s method of dividing the user’s personal information into multiple “data fragments” where “each data fragment is completely unintelligible when taken independently” (Tajima, [0072]) meets [1A]. Bhattacharjee, ¶83.

Tajima forms multiple data fragments ([1A]’s “*plurality of data chunks*”) by dividing personal information (the claimed “*data set*” in the Fig. 12 embodiment) or by dividing encrypted personal information (the claimed “*data set*” in the Fig. 13 embodiment) into “a plurality of data portions” of “fixed” or “arbitrary length”

that are “then arranged into two data fragments, data fragment X and data fragment Y, according to a set method of arrangement.” Tajima, [0111], [0113]. This meets “*distributing the data set into a plurality of data chunks*” under both Google’s construction (“splitting the data set into a plurality of data chunks”—EX1042, 2) and SFI’s construction (“dividing and placing the data set into a plurality of data chunks”—EX1043, 2). Bhattacharjee, ¶¶84-85. As shown in annotated Figs. 12 and 13 below, Tajima splits the data set into data portions that are placed into a plurality of data chunks.

Fig. 12

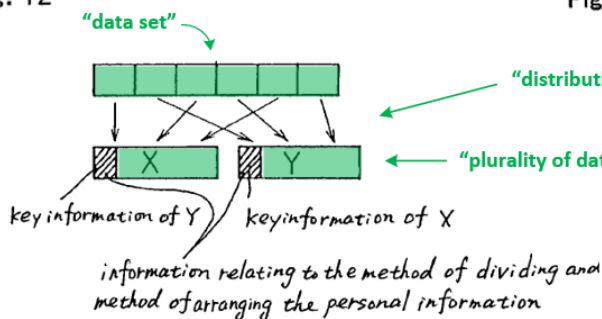
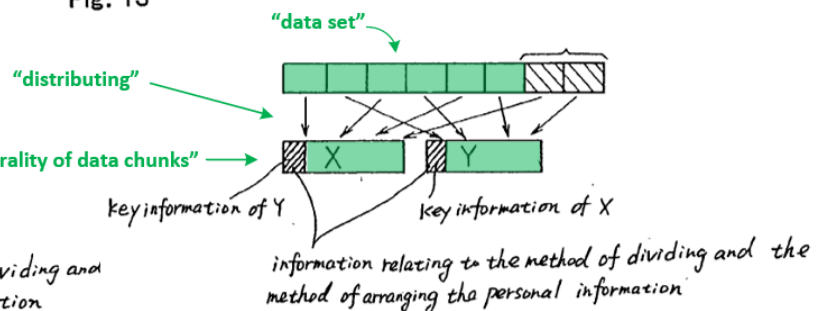


Fig. 13



Tajima’s data fragments X and Y each include a non-contiguous subset of the divided data portions of the data set (personal information in Fig. 12 and encrypted personal information in Fig. 13) “such that each data fragment is completely unintelligible when taken independently.” Tajima, [0072]); *see also* Tajima, [0111], [0113], [0122]; Bhattacharjee, ¶¶85-86. Thus, none of the data fragments X and Y (the claimed “*data chunks*”) in Tajima “*are, by themselves, sufficient to reconstruct the data set*” as recited in [1A] because: (1) each fragment

is only a subset of the data set and/or because (2) each subset is itself unintelligible. Bhattacharjee, ¶86.

Nothing in the intrinsic record supports an interpretation of a “data chunk” as requiring anything more than what limitation [1A] recites—i.e., that “*none of the data chunks are, by themselves, sufficient to reconstruct the data set.*” The ’116 patent uses “portions” or “shares” when describing subsets into which a data set is split, which “portions” or “shares” may include a subset of the bits of the data set that are contiguous or non-contiguous, encrypted or plaintext. EX1001, 21:55-22:4, 58:58-62; 77:51-82:2; Bhattacharjee, ¶87.

The specification uses the term “chunk” in a single paragraph (81:61-82:2) to refer to an intermediate collection of bits (one bit from each of a plurality of “segments” into which a data set was divided) that are then distributed into a plurality of shares, demonstrating that a “chunk” can be a subset of non-contiguous bits of the data being secured. EX1001, 81:38-82:2. There is no evidence that the claimed “data chunk” cannot encompass a portion including some but not all of the data set. Bhattacharjee, ¶88. Thus, Tajima’s dividing encrypted/unencrypted personal information into data fragments meets [1A] for the same reasons the ’116 patent’s embodiments do. *Id.*

3. [1B] encrypting each of the data chunks with a respective one of a plurality of encryption keys

Tajima meets [1B] because in both the Fig. 12 and Fig. 13 embodiments “data fragment X and data fragment Y are each **encrypted**” using a different key, and each key is then stored with a data fragment that key did not encrypt, as illustrated in Figs. 12 and 13 (shown below with encrypted data fragment X and the key information used to encrypt it highlighted **red** and encrypted data fragment Y and the key information used to encrypt it highlighted **blue**). Tajima, [0111], [0113]. Bhattacharjee, ¶89.

Fig. 12

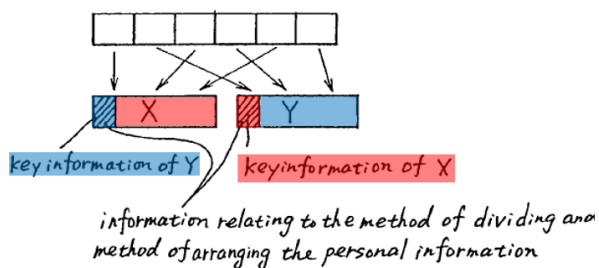
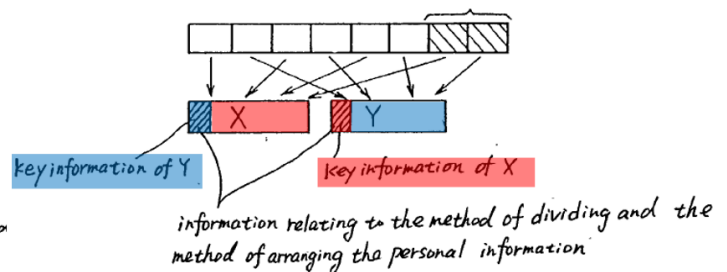


Fig. 13



POSAs understood that Tajima’s information regarding “the key that was used in the encryption of data fragment X” and “the key that was used in the encryption of data fragment Y” (Tajima, [0111], [0113]) means keys that are different from each other—otherwise there would be no reason for Tajima to store each fragment’s key with the other fragment or for the key information “to be used to decrypt” a different data fragment from the fragment with which it was stored. Tajima, [0112], [0114], Figs. 12 and 13 (differentiating “key information of X”

and “key information of Y”); Bhattacharjee, ¶90. Thus, Tajima encrypts each data fragment “with a respective one of a plurality of encryption keys” even if [1B] requires distinct keys.

Even if Tajima had not disclosed different keys for data fragments X and Y, that would have been an obvious and conventional way to a POSA to implement what Tajima describes, motivated by the well-known use of different keys for different data portions towards achieving Tajima’s goal of increased security (Tajima, [0044]) with reasonable expectation of success based on the technique’s conventionality. Bhattacharjee, ¶¶91-92 (*citing* EX1006, [0004], [0064]-[0071]; EX1016, 1:24-42, 3:56-60; EX1036, [0040]-[0048]; EX1025, 7:2-10:15).

4. [1C] obfuscating each of the plurality of different encryption keys

Tajima meets [1C] because “key information of Y” is attached to encrypted data fragment X and “key information of X” is attached to encrypted data fragment Y, in the Fig. 12 and 13 embodiments annotated below. Tajima, [0111], [0113]; Bhattacharjee, ¶93.

Fig. 12

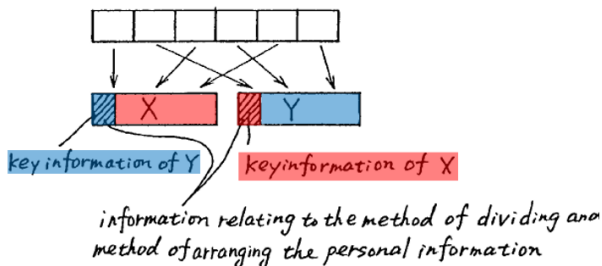
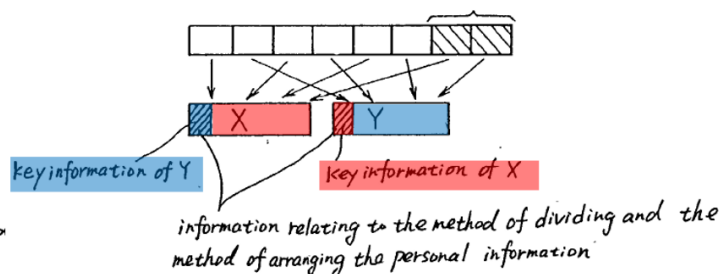


Fig. 13



As discussed *supra* §III, the '116 patent obfuscates in this same way: in the “**Obfuscate**” step (boxed **red** in the portion of '116 Fig. 21 below), “Share 1 gets Key 4, Share 2 gets Key 1, Share 3 gets Key 2, Share 4 gets Key 3.” EX1001, 57:59-62; Bhattacharjee, ¶94.

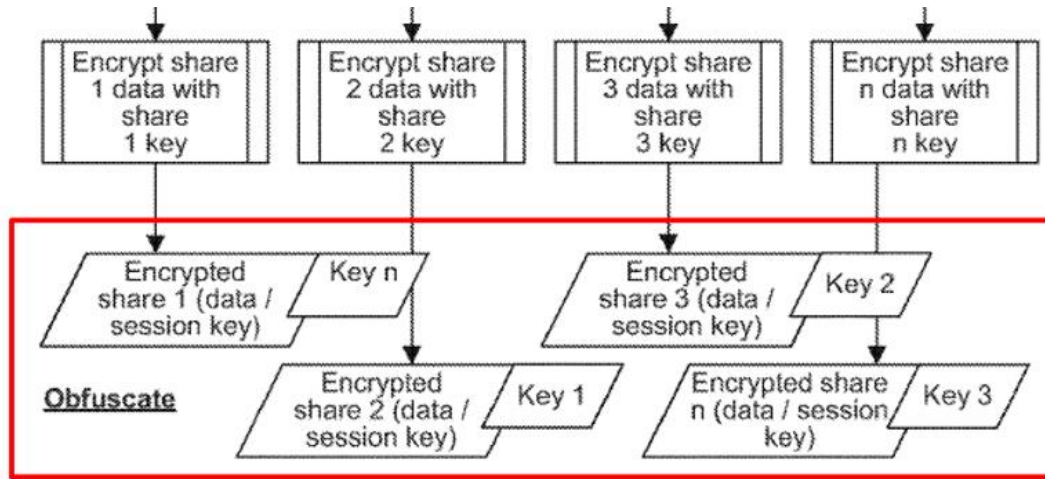


FIG. 21

By attaching key information to a data fragment that it did not encrypt, Tajima obfuscates the same way the '116 patent obfuscates, and thus meets [1C]. Bhattacharjee, ¶95; EX1042, 7; EX1043, 5. POSAs understood that Tajima’s “key information”—i.e., “information regarding the key that was used for the encryption of [the respective] data fragment” (Tajima, [0111], [0113])—includes *at least* the corresponding key for several reasons. Bhattacharjee, ¶96.

First, Tajima discloses that to restore secured personal information, “key information that has been attached to the encrypted data fragment X is used to decrypt data fragment Y, and key information that has been attached to encrypted

data fragment Y is used to decrypt data fragment X.” Tajima, [0112], [0114]. This teaches that the key information includes the key, because Tajima never says that anything else beyond the key information is needed for decrypting the corresponding data fragment and does not disclose any additional steps nor any separate key management infrastructure. Bhattacharjee, ¶¶97-98, 103. Tajima also teaches storing with the data fragments how personal information was split and arranged, reinforcing that Tajima’s data fragments include sufficient information for the authorized service site to recover the personal information. Tajima, [0104]-[0105], [0107]-[0108], [0112], [0114]; Bhattacharjee, ¶98.

Second, Tajima’s claims reinforce that “key information” includes the corresponding encryption key. Specifically, Tajima’s claim 8 recites a system that “encrypts said personal information and divides the encrypted personal information and the **encryption key** used in the encryption into a plurality of data portions” that are arranged into “at least two data fragments” (see claim 4 from which claim 8 depends), which is described in the Fig. 11 and 13 embodiments (Tajima, [0109]-[0110], [0113]-[00114]) where “information regarding the key” (referred to interchangeably as “key information”) used to encrypt personal information is divided and arranged in the data fragments. Tajima, [0109]-[0114], Figs. 11-13; Bhattacharjee, ¶¶99-101. Thus, what Tajima calls “key information” is claimed to include “the encryption key” itself, reinforcing that the “key information” used to

encrypt data fragments (Tajima, [0111]-[0114]) also includes the corresponding encryption key. Bhattacharjee, ¶102.

Even if Tajima had not disclosed that its key information includes the corresponding key, that would have been an obvious and conventional way to implement what Tajima discloses—i.e., a system that needs only the key information to decrypt data fragments and not additional key management infrastructure. Bhattacharjee, ¶104.

Moreover, even if Tajima’s key information did not include the key, attaching the key information to a different data fragment which that key information cannot decrypt would still meet [1C], because pairing a key’s information with a non-corresponding data fragment obfuscates the key by making it unclear or obscure to an attacker which data fragment the key can decrypt. Bhattacharjee, ¶¶105-106; *see* EX1043, 5 (SFI interpreting an “obfuscated” key as one “rendered unclear or obscure”).

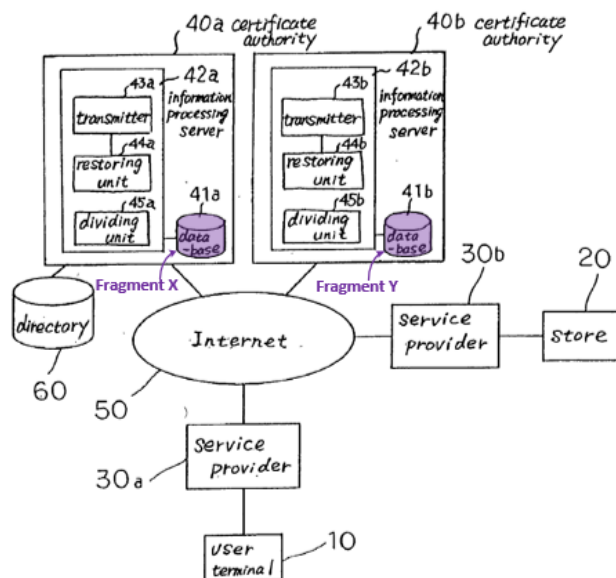
5. [1D] separately storing each data chunk of the plurality of data chunks together with one of the plurality of obfuscated different encryption keys on a plurality of different storage devices

a. Tajima meets [1D] the same way the '116 patent's embodiments do

Tajima's attaching key information to encrypted data that it did not encrypt and registering the resulting data fragments on respective different databases meets [1D]. Tajima, [0111], [0113], [0122], [0156]; Bhattacharjee, ¶107.

POSAs understood that (1) registering a data fragment with a data base includes storing it and (2) Tajima's separate databases that each "belong to" the corresponding service site (Tajima, [0129], [0152]) at which respective data fragments are stored (e.g., fragments X and Y stored on databases 41a and 41b of certificate authorities 40a and 40b, respectively, as in Fig. 5 below) are part of "different storage devices" as recited in [1D]. Bhattacharjee, ¶¶71, 108.

Fig. 5



Indeed, Tajima emphasizes storing different data fragments in “areas that are connectible to the Internet” and “each under separate control” (Tajima, [0048]) and that are “closed to the outside by means of the access control function of the server or firewall” (Tajima, [0079]) so that full personal information cannot be recovered by a security breach of a single service site. Tajima, [0042], [0048], [0073], [0079], [0111]-[0114]; Bhattacharjee, ¶109. POSAs understood this to disclose that respective service sites’ databases are implemented on respective different storage devices, at least because providing these databases on the same storage device would defeat a primary purpose of Tajima’s improved security system (preserving security in the event of a breach at one of the service sites). Bhattacharjee, ¶109.

Thus, Tajima separately stores each encrypted data fragment together with an “*obfuscated different encryption key*” (i.e., the key information that encrypted the other data fragment) on “*a plurality of different storage devices*” (i.e., a plurality of different databases, each under separate control of a respective different service site) in the same way the ’116 patent discloses storing encrypted data portions, each with the encryption key of a different data portion, on different depositories, as in 60:53-56 of the ’116 patent quoted below. Bhattacharjee, ¶110.

“Depository 1: SD1, Kn
Depository 2: SD2, K1
Depository 3: SD3, K2
Depository n: SDn, K3.”

Specifically, each encrypted share (SD1-SDn) is paired with one of the encryption keys (K1-Kn) used to encrypt a different share, and each SD/K pair is stored on a different depository (EX1001, 60:30-56), which is the same technique Tajima discloses in the Fig. 12 and 13 embodiments. Tajima, [0111]-[0114], [0116], [0153], [0156]; Bhattacharjee, ¶111. Applying the '116 patent's notation to Tajima's Fig. 5 system, Tajima discloses:

Database 41a: encrypted data fragment X, key information of Y

Database 41b: encrypted data fragment Y, key information of X

Thus, Tajima meets [1D] at least because Tajima separately stores data fragment/key pairs on different storage devices in the same way the '116 patent does, which also meets Google's proposed construction (EX1042, 7).

Bhattacharjee, ¶112. SFI proposes that [1D] requires "separately storing each data chunk...together with one of the...different encryption keys that has been rendered unclear or obscure." EX1043, 5. Tajima meets that construction because separately storing each data fragment with a key that did not encrypt that fragment renders the key unclear or obscure as to which fragment the key can decrypt. Bhattacharjee, ¶112.

b. [1D] only requires storing each data chunk/key pair once

Construing [1D] to require that each data chunk/encryption key pair be stored on multiple different storage devices would exclude the above-described

embodiments the '116 patent describes for storing data portion/key pairs on different storage devices. *SynQor, Inc. v. Artesyn Tech., Inc.*, 709 F.3d 1365, 1378–79 (Fed. Cir. 2013) (“A claim construction that excludes the preferred embodiment is rarely, if ever, correct and would require highly persuasive evidentiary support.”) (internal quotations omitted). *Bhattacharjee*, ¶113.

No intrinsic evidence supports interpreting [1D] to require storing each chunk/key pair on multiple different storage devices. As originally presented, [1D] recited “*storing the plurality of data chunks with the plurality of obfuscated different encryption keys on a plurality of different storage devices*” (EX1002, 247), which broadly covers at least the '116 patent's above-discussed storing technique. *Bhattacharjee*, ¶114.

SFI then amended claim 1 to read: “separately storing each data chunk of the plurality of data chunks together with one of the plurality of obfuscated different encryption keys on a plurality of different storage devices,” citing paragraphs [0431]-[0438] of the published application (EX1039) for support (corresponding to EX1001, 63:11-67). EX1002, 123. None of the cited paragraphs disclose redundantly storing each share/encryption key pair on multiple different depositories, and EX1039, [0431] (EX1001, 63:11-16) states “Share Encryption Keys may be stored in different shares than the share that was encrypted,” reiterating the '116 patent's above-described storing technique (EX1001, 60:53-56)

and reinforcing that the amendment did not exclude those embodiments.

Bhattacharjee, ¶¶115-116.

SFI's prosecution arguments further reinforce that the claims were not amended to newly require redundantly storing share/key pairs but were "amended to clarify" that **each** data chunk is paired with **one** of the obfuscated encryption keys that is stored separately from the other data chunk/obfuscated encryption key pairs, rather than storing them all together as SFI argued. EX1002, 123;

Bhattacharjee, ¶¶116-117. Specifically, in distinguishing Bolosky, SFI alleged:

"there is no disclosure in Bolosky that recites separately storing each data chunk with one of the plurality of obfuscated different encryption keys...Consequently, Bolosky merely describes a single data stream that stores all of the data and keys for decrypting all of the data, which is not the same as separate storing as require [sic] by amended claim 2." EX1002, 124 (emphasis original).

SFI's emphasis reinforces that the amended claims require that **each** data chunk is stored with **one** of the obfuscated encryption keys and separately from other data chunk/encryption key pairs (i.e., as the '116 patent discloses).

Bhattacharjee, ¶¶116-117 (*citing* EX1002, 120-125). SFI never alleged that the claims require that **each** data chunk/encryption key pair be stored on **multiple** different storage devices. Bhattacharjee, ¶¶118-119 (further describing the

different scope of parallel system claims 15 and 21 if [1D] required redundant storage).

- c. If [1D] were construed to require storing each data chunk/key pair on multiple different storage devices, it is met by an obvious implementation of Tajima**

A POSA would have been motivated to redundantly store Tajima's fragments (which each include a data fragment/key pair) in Tajima's system for fault tolerance reasons as was routinely done as a standard practice in secure data storage. Bhattacharjee, ¶120. For example, it was known to employ RAID technology to redundantly store data (EX1035, 8:13-23, 10:36-43; EX1032, 1:5-13; EX1033, 1:9-12; EX1013, [0338]), including to provide redundant storage of a certificate authority database in an online merchant context like Tajima's (EX1035, 8:13-23, 10:36-43), providing motivation and reasonable expectation of success in implementing Tajima's system to use RAID technology—i.e., each service site would replicate its stored data segment/key pairs on multiple RAID disks. Bhattacharjee, ¶¶120-121.

It was also known to make copies of segmented data that are redundantly stored on different storage devices at different locations to provide fault tolerance (EX1007, [0007]-[0009]; EX1017, [0025]-[0031], [0058]-[0059]; EX1030, [0005]-[0010]), providing motivation and reasonable expectation of success in configuring Tajima's service sites to copy encrypted fragments resulting from Tajima's

dividing techniques (Tajima, [0111]-[0115], [0156]) to redundantly store each data fragment/key pair at multiple different service sites. Bhattacharjee, ¶¶120, 122.

The obvious implementation of either conventional redundant-storage technique in Tajima's system would merely have been the use of known techniques (redundant storage) for its intended purpose to yield the predicted and intended result of providing fault tolerance in Tajima's system. *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 415-421 (2007); Bhattacharjee, ¶¶121-122.

C. Claim 5: wherein storing...comprises storing each of the plurality of data chunks together with a respective one of the plurality of obfuscated different encryption keys

Tajima's disclosure of attaching to a data fragment the key information (i.e., the "*obfuscated different encryption keys*") used to encrypt a different data fragment meets claim 5.

Claim 5's recitation that each data chunk is stored "*together with a respective one of the plurality of obfuscated different encryption keys*" does not require a data chunk to be stored with the obfuscated encryption key used to encrypt that data chunk. Bhattacharjee, ¶124. Claim 5 recites "a respective one" rather than "*the* respective one," and thus the stored encryption key in claim 5 need not be the same encryption key recited in [1B] ("*encrypting each of the data chunks with a respective one of a plurality of encryption keys*") for that data chunk.

Instead, claim 5 introduces a new “**a respective one of the plurality of obfuscated different encryption keys**” with which the data chunk is stored. *Id.*

Thus, the obfuscated key with which a data chunk is stored in claim 5 may be different from the obfuscated key used in [1B] to encrypt that data chunk, which (1) narrows claim 1’s limitation [1D] that does not require each chunk to be stored with a “*respective*” encryption key, only that it be stored with “*one*” of them, and is (2) consistent with and would not exclude the ’116 patent’s embodiments describing storing data chunks together with encryption keys (see, e.g., EX1001, 57:25-61:9). Bhattacharjee, ¶125; §§III, V.B.5 *supra*.

D. Claim 6: wherein storing...comprises including, with a data chunk of the plurality of data chunks, data indicative of an encryption key that was used to encrypt a different data chunk of the plurality of data chunks

The ’116 patent does not use the “*data indicative of*” language outside the claims. SFI has taken the position in the district court litigation that “*data indicative of*” a key is met by the key itself. EX1012, 8 (alleging that Google performs “a transform on data indicative of at least one of the plurality of different encryption keys” recited in claim 2 because “Google ***performs a transform on each of the different encryption keys when it encrypts the DEKs using the***

KEKs.”³; EX1043, 3-4 (SFI’s preliminary construction that information that “makes known” or “shows” a key is “data indicative of” the key); Bhattacharjee, ¶¶126-128.

Thus, Tajima’s storing key information with a data fragment that the key information did not encrypt (Tajima, [0111]-[0114]) meets claim 6 under SFI’s interpretation because Tajima’s key information (1) includes the key itself (EX1012, 8) and/or (2) “makes known” or “shows” the key (EX1043, 3-4)—i.e., for the key information to be “used to decrypt” the corresponding data fragment (Tajima, [0111]-[0114]), the key information would need to show or make the key known. Bhattacharjee, ¶¶126-128. The obvious use of metadata about the key (e.g., key length and/or encryption type) in Tajima’s key information meets claim 6 under Google’s construction. EX1042, 5; Bhattacharjee, ¶129-133.

E. Claim 7: wherein distributing the data...comprises using a random technique or pseudorandom technique

Tajima discloses that the personal information (Fig. 12 embodiment) or the encrypted personal information (Fig. 13 embodiment) may be divided into data portions of “arbitrary length” (Tajima, [0111], [0113]) determined “according to a

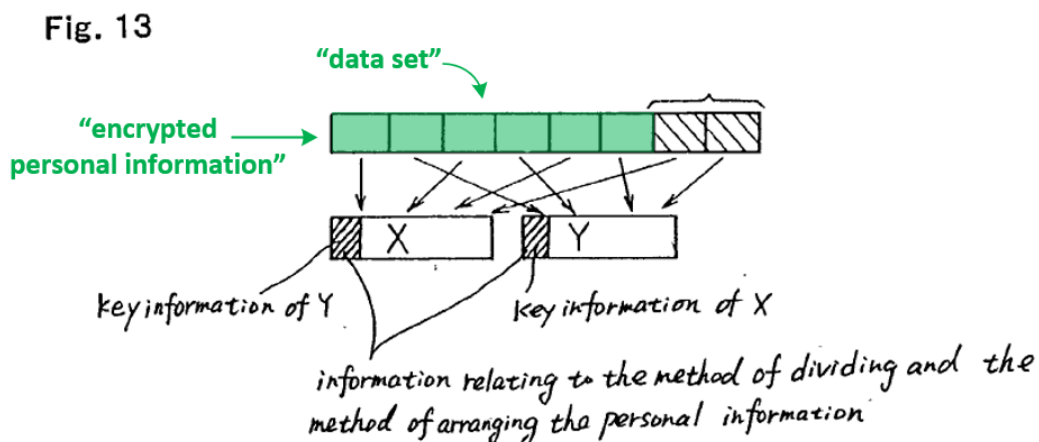
³ See also EX1011, 9 and EX1037, 34 (alleging for co-asserted patents EX1015 and EX1044 that storing data indicative of a key is met by storing the key itself).

function of, for example, **random numbers**” that are then arranged into the data fragments. (Tajima, [0103], [0106], [0111], [0113]).

Tajima’s technique of using “random numbers” to determine the length of each data portion into which the data set is divided and distributed into the data fragments meets claim 7. Bhattacharjee, ¶¶134-136.

F. Claim 8: further comprising generating the data set by encrypting primary data

In Tajima’s Fig. 13 embodiment, the user’s personal information “is first encrypted, and the encrypted personal information and information on the key that was used in encrypting the personal information are then divided into a plurality of data portions.” Tajima, [0113]. As discussed *supra* §V.B.1 in connection with 1[PRE], the “encrypted personal information” meets the claimed “*data set*” as illustrated in annotated Fig. 13 below.



Thus, Tajima’s Fig. 13 embodiment meets claim 8 because the “personal information” meets the claimed “*primary data*” which is then encrypted to provide

“encrypted personal information” that meets the claimed “*data set.*” *Id.*;

Bhattacharjee, ¶¶137-138; §§V.B.1-V.B.5 (demonstrating Fig. 13’s technique meets claim 1), V.A.

VI. GROUND 2: CLAIMS 1-2, 9, 11-14 WOULD HAVE BEEN OBVIOUS OVER TAJIMA AND FOSTER

A. Security Risk of Plaintext Keys

It was known that storing plaintext (i.e., unencrypted) keys with encrypted data results in a vulnerability. Bhattacharjee, ¶140, *citing* EX1021, 1:45-65 (explaining that simple-to-use methods “where the data encryption keys are stored or transmitted together with the encrypted data” are likewise potentially simple to defeat), EX1023, 17:44-18:8 (storing plaintext keys with the information that they protect “is somewhat akin to locking a house and then putting the keys under the doormat”); EX1022, 10:42-48 (“cryptographic keys should never occur in clear form outside the cryptographic device”).

While Tajima achieves some protection by storing the key used to encrypt a data fragment with a different data fragment, POSAs understood that by storing the key in plaintext, Tajima’s approach subjects the data fragments to risk because if an attacker is able to recover all of Tajima’s data fragments (e.g., both data fragments X and Y), the attacker will possess all the plaintext keys used to encrypt the entire data set. Bhattacharjee, ¶141. Decrypting the individual data fragments would become merely a matter of trying different key/data-fragment combinations,

which was within the capabilities of hackers in the relevant timeframe.

Bhattacharjee, ¶¶141-43 (describing known techniques to address the vulnerability of plaintext keys).

B. Reasons to Encrypt Tajima's Keys

It was well-known that in systems that store keys with encrypted data, encrypting the keys increases security. Bhattacharjee, ¶¶144-145; EX1021, 1:66-2:4 (“Security for such a system, where the encryption keys are included with the encrypted message, is not to store or transmit the encryption keys in an easily readable format but rather to encrypt the encryption keys themselves”); EX1023, 18:28-31 (“[when] data encryption keys are stored on the media, [] we additionally encrypt them”); EX1006, [0004], [0007]-[0008], [0064]-[0071] (encrypting keys for attachment to encrypted data).

A POSA would have encrypted Tajima's keys for numerous reasons.

First, POSAs would have been motivated to encrypt Tajima's keys so that acquiring all data fragments would not provide an attacker with all necessary keys in plaintext form, and POSAs had good reason to add this further security in Tajima's system. Bhattacharjee, ¶¶146-147. Tajima's stated security improvement was to divide personal information into data fragments that are “completely unintelligible when taken independently” (Tajima, [0072], [0122]) and distributively store those fragments so that personal information could not be

recovered by breach of any one system. Tajima, [0042]-[0048], [0079]; Bhattacharjee, ¶147. Tajima's Fig. 8-10 embodiments rely solely on this technique. Tajima, [0101]-[0108].

But Tajima then discloses further levels of security (e.g., first encrypting the personal information before dividing it (Figs. 11 and 13) and/or encrypting the data fragments after division (Figs. 12 and 13)). Tajima, [0109]-[0113]. This demonstrates to POSAs that Tajima recognized the possibility of failure of only storing its data fragments on different databases, such that additional techniques would be desirable to increase security. Bhattacharjee, ¶148. POSAs would have been motivated to apply known key-encryption techniques to Tajima so that an attacker that has acquired all the data fragments would not have all the information needed to decrypt the individual data fragments. This multi-level protection of Tajima's keys is consistent with Tajima's teaching to employ multiple layers of security in protecting users' personal information and such "belt-and-suspenders" approaches to security were well-known. Bhattacharjee, ¶¶148-149 (citing EX1016, 3:12-60, 4:21-27; EX1036, [0003]-[0009], [0053]-[0054]; EX1026, 10:7-28 for corroboration).

Second, encrypting Tajima's keys would have been nothing more than applying a known technique (key encryption) for its intended purpose (to increase security) to yield the predictable result of increasing security of a user's personal

information. *KSR*, 550 U.S.at 415-421; Bhattacharjee, ¶150. POSAs would have reasonably expected success as encrypting keys in systems like Tajima’s was routine and conventional. *Id*; EX1006, [0064]-[0068]; EX1016, 4:21-24; EX1021, 1:66-2:4; EX1023, 18:28-31; EX1040, 4; EX1018, [0032].

C. Foster’s Key-Encryption Technique

A simple, known way to address the above-described vulnerability of storing plaintext keys would have been to encrypt Tajima’s key information using a key-encrypting key (“KEK”) as described in Foster (EX1006). Bhattacharjee, ¶151; Foster, [0064]-[0068]. POSAs would have understood the desirability of Foster’s KEK technique in providing a straight-forward way to encrypt Tajima’s keys while requiring little modification to Tajima’s system.

As shown in steps shaded **blue** in Foster’s Fig. 8 below, Foster divides digital content 100 into content units 102, each having a data payload 106 that is encrypted with a different encryption key 108 (Foster, [0064]-[0068], [0071]), like Tajima divides personal information into data fragments and encrypts the fragments (Tajima, [0111], [0113]). Bhattacharjee, ¶152.

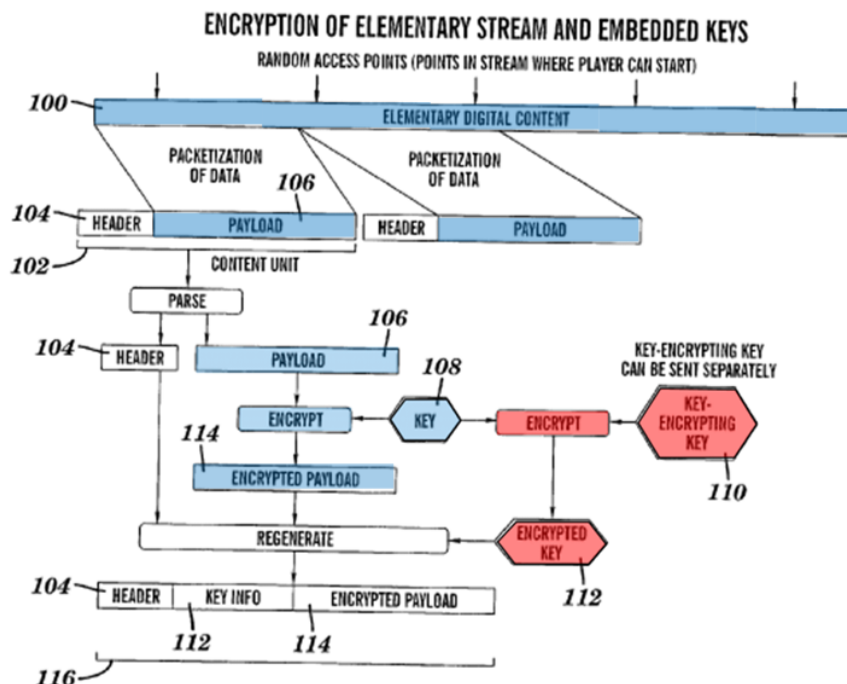


FIG. 8

Foster teaches encrypting each key 108 with a key-encrypting key (KEK) 110 so that encrypted key 112 (shaded **red** above) can be securely stored/transmitted with the encrypted payload. Foster, [0064]-[0071]. POSAs would have recognized the desirability of using Foster's key-encryption technique to encrypt Tajima's keys before attaching them to different encrypted fragments to benefit from the increased level of security discussed in §VI.B *supra*. POSAs would have appreciated that this benefit would be achieved with little modification to Tajima's system (see §VI.E *infra*). Bhattacharjee, ¶¶153-154.

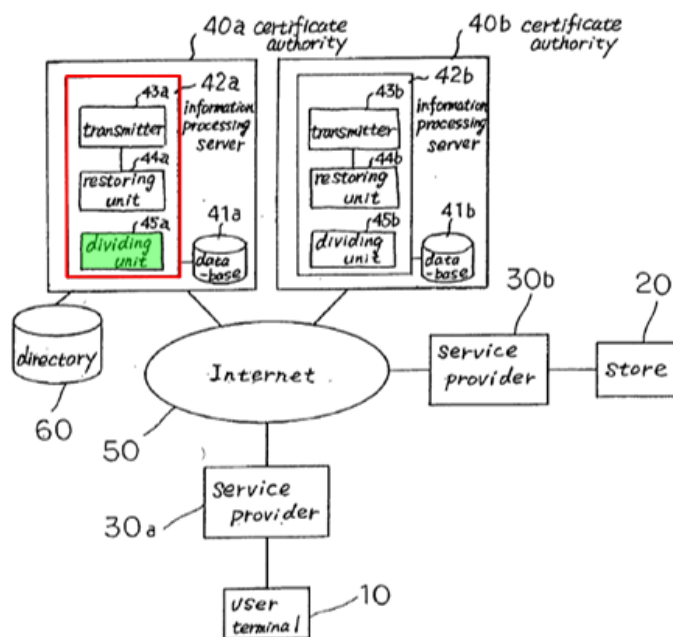
D. Tajima-Foster Combination

Tajima-Foster uses (1) Tajima's techniques of dividing personal information and securely storing data fragments on databases under control of different

respective service sites; and (2) Foster’s key-encryption technique (e.g., steps highlighted in **red** in Foster’s Fig. 8 above). Bhattacharjee, ¶155.

As discussed in §V.A above, each service site’s dividing unit implemented by its information-processing server—e.g., certificate authority 40a’s dividing unit 45a in Fig. 5 below (or service provider 130b’s dividing unit 135b in Fig. 14)—performs Tajima’s Fig. 12 and Fig. 13 dividing techniques. Tajima, [0067], [0100], [0111]-[0113], [0118], [0122]-[0123]; Bhattacharjee, ¶156.

Fig. 5



A POSA seeking to implement Foster’s key-encryption technique in Tajima-Foster to prevent an attacker having acquired all data fragments from possessing all the keys in plaintext needed to decrypt the data fragments (see §§VI.A-VI.C *supra*) would have programmed Tajima’s dividing units to also encrypt each data

fragment's key information with a KEK as described in Foster. Bhattacharjee, ¶157. Tajima-Foster would have improved security both (1) where the service site that divides the personal information stores one of the resulting data fragments (Tajima, [0073]-[0074]), and (2) where that service site only stores link information and all resulting data fragments are stored at other service sites (Tajima, [0116], [0153], [0156]). Bhattacharjee, ¶¶67, 157-158.

POSAs would have recognized that Tajima-Foster would have beneficially provided even further security for storage technique (2) above because encrypted data fragments/key pairs **and** the KEK needed to decrypt the keys are each stored on different databases under separate control of respective service sites—i.e., the KEK at the service site that divided the personal information, and the resulting encrypted data fragments and encrypted keys at other sites—requiring an attacker to not only acquire all data fragments and encryption keys stored on multiple different databases, but to also gain unauthorized access to another system to acquire the KEK. Bhattacharjee, ¶159.

E. Reasonable Expectation of Success

Tajima's dividing units are already configured to encrypt data with encryption keys (Tajima, [0067]-[0068], [0109]-[0114], [0118], [0122]) such that configuring them to perform an additional encryption step (i.e., Foster's key-encryption technique) would have been a straightforward modification requiring

only conventional and routine skills that a POSA would have had a reasonable expectation of successfully implementing. Bhattacharjee, ¶160.

Moreover, Tajima’s service sites already maintain “link information” identifying the registration destinations where data fragments are stored so that the authorized service site can retrieve the the data fragments to restore a user’s personal information (Tajima, [0046], [0073], [0093], [0116], [0123], [0144], [0156])—so also maintaining a KEK would impose little additional burden. Bhattacharjee, ¶¶161-162 Furthermore, Tajima disclosure that this “link information may also be encrypted and held” (Tajima, [0093], [0144]) demonstrates Tajima’s service sites already securely maintain encryption information. *Id.*

Thus, POSAs would have recognized that the increased security imparted by encrypting Tajima’s keys would not have imposed any significant key management burden on Tajima’s system and would have required nothing more than conventional and routine skill to implement. Bhattacharjee, ¶¶162-163.

Tajima-Foster renders obvious claims 1-2, 5-9 and 11-14, as detailed below.

F. Claim 1

1. 1[PRE]-[1D]

Tajima-Foster (*supra* §VI.D) uses Tajima’s dividing, encrypting, obfuscating and storing techniques that meet 1[PRE]-[1D] as demonstrated in

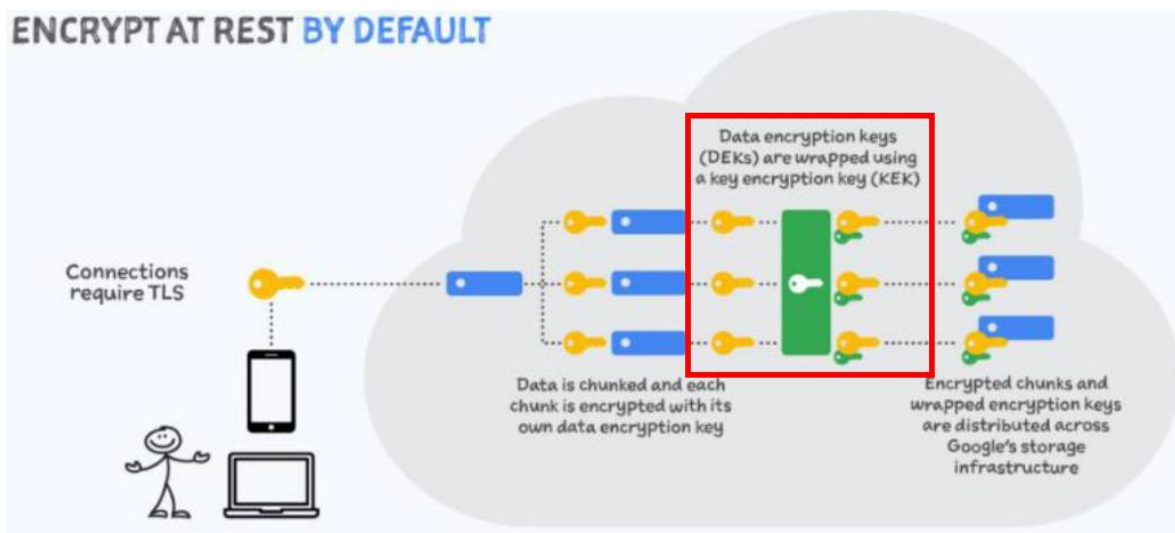
§§V.B.1-V.B.5 above, and additionally encrypts Tajima’s key information using Foster’s key-encryption technique. Encrypting the key information does not change how 1[PRE]-[1D] are met. Therefore, Tajima-Foster meets 1[PRE]-[1D] for the same reasons Tajima alone does (including the obvious implementations of Tajima discussed *supra* §V), and alternatively meets [1C] for the additional reason discussed below. Bhattacharjee, ¶164.

2. [1C] obfuscating...

Tajima-Foster (§VI.D) obfuscates keys by attaching key information to a different data fragment as discussed *supra* §V.B.4 and additionally obfuscates Tajima’s keys according to SFI’s interpretation that “*obfuscating*” a key is met by encrypting it (EX1012, 4-5) and that an “obfuscated key” is one rendered “unclear or obscure” (EX1043, 4-5). Bhattacharjee, ¶165.

Specifically, SFI alleged in its infringement contentions that “Google ***obfuscates*** each of the different encryption keys ***when it encrypts*** the [data encryption keys] DEKs using the [key encryption keys] KEKs” and that the image below “show[s] Google’s obfuscation of the DEKs by the KEKs.”. EX1012, 4-5.

ENCRYPT AT REST BY DEFAULT



Therefore, Tajima-Foster (§VI.D) *additionally* meets [1C] under SFI's interpretation because (1) Tajima-Foster obfuscates by encrypting Tajima's keys with a KEK (EX1012, 4-5) and (2) encrypting Tajima's keys renders them "unclear or obscure" (EX1043, 5). Bhattacharjee, ¶¶166-167.

G. Claim 2: wherein obfuscating...comprises performing a transform on data indicative of at least one of the plurality of different encryption keys

SFI alleges that claim 2's "*performing a transform on data indicative of...the different encryption keys*" is met by encrypting the keys themselves. EX1012, 8. Specifically, SFI asserted in the litigation that Google performs "a transform on *data indicative of* at least one of the plurality of different encryption keys" because "Google performs a transform ***on each of the different encryption keys when it encrypts the DEKs using the KEKs.***" EX1012, 8. Tajima-Foster (§VI.D) therefore meets claim 2 under SFI's infringement interpretation by

encrypting Tajima’s key information (which includes the corresponding key) with a KEK. Bhattacharjee, ¶168.

Tajima-Foster also meets claim 2 under SFI preliminary constructions of (1) “*a transform*” because encrypting Tajima’s key information “chang[es] the form” (EX1043, 5-6) of the keys from plaintext (unencrypted) to ciphertext (encrypted), and (2) “*data indicative of*” a key because Tajima’s key information “makes known” or “shows” the key (EX1043, 3-4) as discussed *supra* §V.D. Bhattacharjee, ¶¶168-169; §VI.F.2 *supra*.

H. Claim 9

1. 9[PRE]-[9B]

9[PRE]-[9B] recite the identical limitations recited in 1[PRE]-[1B], and therefore are met by Tajima-Foster for the same reasons Tajima alone (see §§V.B.1-V.B.3 *supra*) and Tajima-Foster (see §VI.F.1 *supra*) meet 1[PRE]-[1B]. Bhattacharjee, ¶¶170-171.

2. [9C] transforming each of the plurality of different encryption keys

Tajima-Foster (§VI.D *supra*) encrypts Tajima’s key information that includes the corresponding key used to encrypt individual data fragments with a KEK (Foster’s key-encrypting technique). Bhattacharjee, ¶172. In alleging infringement of claim 9, SFI alleged that [9C] is met because “Google transforms each of the different encryption keys when it *encrypts the DEKs using the KEKs.*”

EX1012, 12; EX1043, 5-6 (construing transforming as “changing the form”);
Bhattacharjee, ¶172; §VI.G *supra* (claim 2’s transform met by encrypting).

Tajima-Foster therefore meets [9C] under SFI’s interpretation by encrypting Tajima’s keys with a KEK (EX1012, 12), thereby “changing the form” (EX1043, 5-6) of Tajima’s keys from unencrypted to encrypted form. Bhattacharjee, ¶173.

3. [9D] separately storing each data chunk of the plurality of data chunks together with one of the plurality of transformed different encryption keys on a plurality of different storage devices

Limitation [9D] differs from [1D] in that the stored encryption keys are “transformed” encryption keys. In Tajima-Foster (§VI.D), each encrypted key—i.e., a “transformed” encryption key under SFI’s interpretation that encrypting keys transforms them (see [9C]’s “transforming” step in §VI.H.2 *supra*)—is attached to a data fragment it did not encrypt and stored on a respective different database. Therefore Tajima-Foster meets [9D]’s “separately storing” of data chunks with “transformed” keys for the same reasons Tajima alone (including obvious implementations) (discussed §V.B.5) and Tajima-Foster (§VI.F.1) meet [1D]’s “separately storing” of “obfuscated” keys. Bhattacharjee, ¶¶174-175.

I. Claim 11: storing each of the plurality of data chunks with a respective one of the plurality of transformed different encryption keys

Tajima-Foster (§VI.D) meets claim 11 because each of Tajima’s data fragments is stored with a different (i.e., “respective”) one of Tajima’s encrypted

(i.e., “*transformed*” under SFI’s interpretation) encryption keys. Claim 11’s storing “*with a respective one of*” limitation mirrors claim 5’s storing “*with a respective one of*” and requires nothing more for the same reasons discussed in §V.C *supra* in connection with claim 5. Bhattacharjee, ¶176.

J. Claim 12: including, with a data chunk of the plurality of data chunks, data indicative of an encryption key of the plurality of encryption keys that was used to encrypt a different data chunk of the plurality of data chunks”

Tajima-Foster (§VI.D) stores each data fragment with encrypted key information that includes the key that was used to encrypt a different data fragment (Tajima, [0111], [0113]). Tajima-Foster therefore meets claim 12 at least under SFI’s interpretation of “*data indicative of an encryption key*” as discussed in §V.D *supra* regarding claim 6’s same limitation. EX1012, 8; EX1043, 3-4; Bhattacharjee, ¶177.

K. Claim 13: distributing the data set...using a random technique or pseudorandom technique”

Tajima-Foster (§VI.D) uses Tajima’s technique for distributing divided portions of personal information into data fragments—i.e., using “random numbers” to divide personal information (Tajima, [0106], [0111], [0113])—and meets claim 13 for the same reason Tajima meets claim 7 (§V.E *supra*). Bhattacharjee, ¶178.

L. Claim 14: further comprising generating the data by encrypting primary data

Tajima-Foster (§VI.D) uses Tajima's technique of encrypting personal information before dividing it (Tajima, [0113]) and thus meets claim 14 for the same reasons Tajima meets claim 8 (§V.F *supra*). Bhattacharjee, ¶179.

VII. GROUND 3: CLAIMS 9-14 WOULD HAVE BEEN OBVIOUS OVER TAJIMA AND RIVEST

A. Rivest

Rivest (EX1034) published in renowned Lectures Notes in Computer Science, Vol. 1267 of which published the 1997 Proceedings of the 4th Intl. Workshop on Fast Software Encryption, and its cover information bears (1) well-known publisher Springer-Verlag's 1997 copyright notification, (2) serial/book numbers, (3) and library stamps demonstrating public accessibility at least by Sept 1997. Rivest, 1-4; Bhattacharjee, ¶¶181-183.

B. Tajima-Rivest Combination

A POSA would have recognized the benefit of applying Rivest's "all-or-nothing" transforms (AoNT) to Tajima's key information to protect Tajima's plaintext keys (see §§VI.A-VI.B *supra*) without needing to maintain secret keys. Rivest, 212 ("all-or-nothing transform...makes no use of any secret key information")-214; Bhattacharjee, ¶¶184-185. Additionally, transforming Tajima's key information using AoNT would have been nothing more applying a known technique according to its intended purpose (transforming plaintext to a different

form) to yield the predictable result of providing transformed keys that can be more securely kept with encrypted data to achieve Tajima's objectives in securing users' personal information. Rivest, 212-214; KSR, 550 U.S. at 416-417; Bhattacharjee, ¶185. The '116 patent itself acknowledges that AoNT was "well-known" and that "***it will be understood*** that the AoNT may be applied to encryption keys used to encrypt the data portions following parsing and splitting." EX1001, 77:13-43; Bhattacharjee, ¶184.

Tajima-Rivest applies Rivest's AoNT using the user's public key provided to Tajima's service sites (Tajima, [0070]-[0071], Fig. 6 (S2-S3) and thus beneficially avoids using secret keys. Rivest, 213-214; Bhattacharjee, ¶¶186-192.

C. Claim 9

Tajima-Rivest (§VII.B) transforms Tajima's key information using AoNT and thereby changes the bits (and likewise changes the form) of Tajima's keys where "no secret keys are used" (Rivest, 213), meeting [9C]'s "*transforming*" limitation under both Google's construction ("Changing without the use of a secret key the bits of a key"—EX1042, 8) and SFI's construction ("changing the form"—EX1043, 5-6). Bhattacharjee, ¶193; §VI.G *supra*. Claim 9's other limitations are met for the same reasons discussed in §§VI.H.1 and VI.H.3 above because the manner in which Tajima's keys are transformed does not change the way 9[PRE]-[9B] and [9D] are met by Tajima-Rivest. Bhattacharjee, ¶194.

D. Claim 10

Tajima-Rivest (§VII.B) transforms Tajima’s keys using AoNT, thus “performing an all-or-nothing transform on data indicative of at least one of the plurality of different encryption keys” under SFI’s interpretation that “data indicative of” a key encompasses the key itself. Bhattacharjee, ¶195.

E. Claims 11-14

Tajima-Rivest (§VII.B) meets claims 11 and 12 in the same ways that Tajima-Foster does (*supra* §§VI.IVI.J)—i.e., Tajima-Rivest’s transformed keys are stored with the data fragments in the same manner—and meets claims 13 and 14 for the same reasons Tajima-Foster does (*supra* §§VI.K and VI.L, respectively). Bhattacharjee, ¶¶196-199.

VIII. GROUND 4: CLAIMS 1-3, 6 WOULD HAVE BEEN OBVIOUS OVER TAJIMA, RIVEST AND SHYU

A. Tajima-Rivest-Shyu Combination

POSAs would have recognized the benefit of applying different encryption algorithms to different data fragments as taught by Shyu (EX1016) to avoid known security risks of using a fixed encryption algorithm. Shyu, 1:24-42; 3:12-38, 3:56-60; Bhattacharjee, ¶¶201-204 (*citing also* EX1036, [0003]-[0009]). Moreover, it would have obvious to vary the encryption algorithm in Tajima’s system as doing so is merely the use of a known technique according to its intended purpose (increasing security) to yield the predictable result of making it more difficult for

an attacker to decrypt different data fragments to achieve Tajima's objective of securing a user's personal information. *KSR*, 550 U.S. at 416-417; Bhattacharjee, ¶205.

In addition to including in Tajima's key information metadata identifying what encryption algorithm the corresponding key is used with as taught by Shyu (Shyu, 3:12-26, 3:56-60), POSAs would have known to include additional metadata, such as the length of the key. Bhattacharjee, ¶¶129-133, 206-207 (citing Shyu, 3:12-26; EX1036, [0053]-[0054], [0077]). Tajima-Rivest-Shyu thus (1) encrypts Tajima's data fragments using Shyu's variable-encryption technique with Tajima's different keys and stores the encryption type (and optionally additional metadata) with the corresponding key in Tajima's key information (Shyu, 3:12-60), and (2) transforms the resulting key information using AoNT as in Tajima-Rivest (§VII.B *supra*), beneficially providing multiple protections for Tajima's keys. Bhattacharjee, ¶¶206-207; §VI.A *supra*.

B. Claim 1

Tajima-Rivest-Shyu (§VIII.A) renders obvious claim 1 for the same reasons Tajima meets claim 1 (see §V.B *supra*) because using (1) Shyu's variable-encryption technique; and (2) Rivest's AoNT to transform Tajima's key information does not change the way claim 1's limitations are met by Tajima alone. Bhattacharjee, ¶208.

C. Claim 2

Tajima-Rivest-Shyu (§VIII.A) “perform[s] a transform” (i.e., AoNT) on “data indicative of at least one of the plurality of different encryption keys” (i.e., Tajima’s key information including metadata about the key) and therefore meets claim 2 under both Google’s and SFI’s constructions because (1) AoNT is a “transform” under both constructions as discussed in §VII.C *supra*, and (2) Tajima’s key information includes the corresponding key and metadata about the key (e.g., the encryption type used with the key, key length, etc.) and therefore is “data indicative of” the corresponding key under SFI’s interpretation (EX1012, 8; EX1043, 3-4; §V.D *supra*) and Google’s requiring “metadata about the key” (EX1042, 5-6). Bhattacharjee, ¶209.

D. Claim 3:

Tajima-Rivest-Shyu (§VIII.A) applies AoNT to Tajima-Rivest-Shyu’s key information, therefore “performing an all-or-nothing transform on the data indicative of at least one of the plurality of different encryption keys,” under both parties’ constructions. Bhattacharjee, ¶210; EX1042, 5-6, 8; EX1043, 3-6; EX1012, 8; §§V.D, VII.C-VII.C, VIII.C *supra*.

E. Claim 6

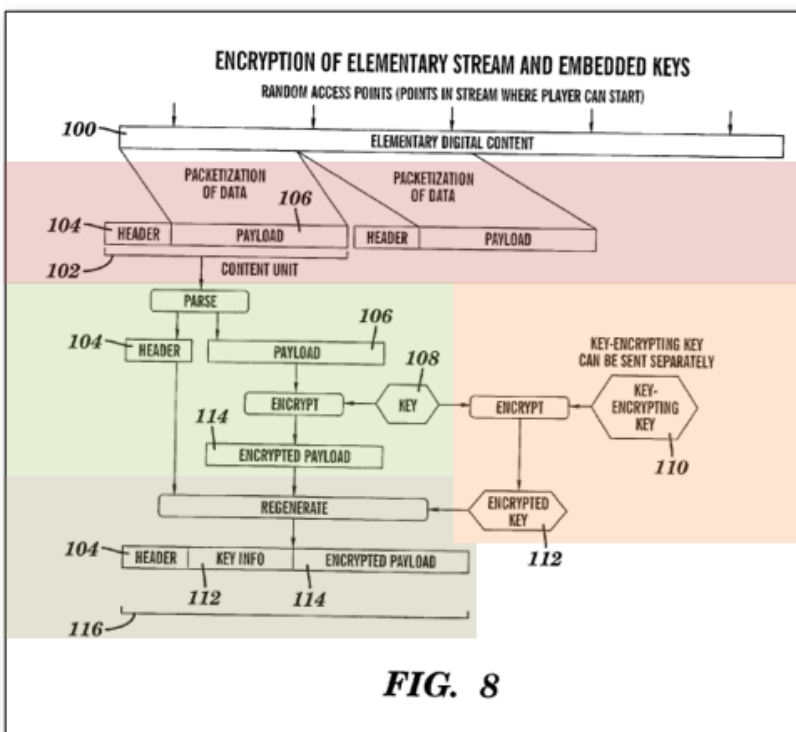
Tajima-Rivest-Shyu (§VIII.A) stores each data fragment with Tajima-Rivest-Shyu’s key information that was used to encrypt a different data fragment,

and therefore meets claim 6 under both parties' constructions. *Bhattacharjee*, ¶211; EX1042, 5-6, 8; EX1043, 3-4; EX1012, 8; §§V.D, VIII.C *supra*.

IX. **GROUND 5: CLAIMS 1-2, 5, 9 AND 11 WOULD HAVE BEEN OBVIOUS OVER FOSTER AND HAYHURST**

A. **Foster (EX1006)**

Foster discloses an encryption technique for encrypting digital content (e.g., “video and audio data,” [0004]) that parcels the content into multiple units, encrypts the data portion (“payload”) of the parceled content units with different encryption keys, encrypts the encryption keys with a key-encrypting key (KEK), and attaches each encrypted key to the corresponding data portion that it encrypted, as described in connection with Fig. 8 annotated below. Foster, [0064]-[0071]; FIG. 8; *Bhattacharjee*, ¶¶213-215; §VI.C *supra*.



(1) Parcel content into pieces

(2) Encrypt pieces with keys

(3) Encrypt keys with KEK

(4) Generate processed content units

(1) “Digital content 100” (e.g., video data) is first “parceled into content units 102.” Foster, [0064]. Each content unit comprises a data portion or “payload 106” (also called a “content packet 106,” *id.*) and a “header 104” that stores information about the corresponding content packet. Foster, [0064], FIG. 8.

(2) Payload 106 is “encrypted with title key 108” to “yield encrypted content packet 114.” Foster, [0064]. Foster teaches that “each content packet could be encrypted with a **different** title key” (Foster, [0065]), which “provides enhanced security by preventing an entire content stream from being accessed with a single key.” Foster, [0004], [0071].

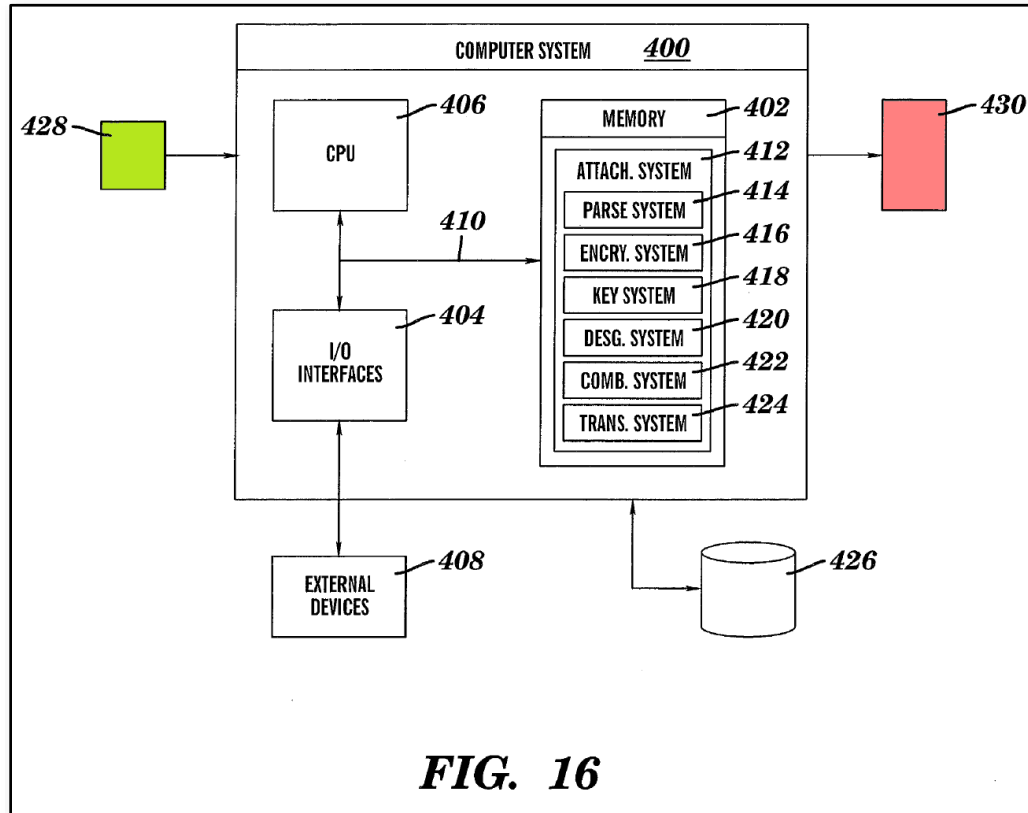
(3) Each title key is encrypted with “key encrypting key 110” to generate encrypted key 112. Foster, [0048], [0064], [0066], FIGS. 8-10.

(4) Finally, encrypted key 112, encrypted content packet 114, and header 104 are combined as a “processed content unit 116,” which can be “stored by a recipient” in a receiver (e.g., stored in a consumer’s set-top box). Foster, [0064], [0040]-[0041]. Bhattacharjee, ¶¶217-220.

Foster’s use of different encryption keys benefits from the known “enhanced security” of “preventing an entire content stream from being accessed with a single key” (Foster, [0004]) and Foster’s KEK technique allows encryption keys to be securely stored/transmitted with the encrypted data, addressing problems with conventional content delivery systems that stored/transmitted keys separately from

the encrypted data. Foster, [0004]-[0007], [0064]-[0068], [0071]; Bhattacharjee, ¶¶214-215, 221.

Foster's Fig. 16 (annotated below) depicts an exemplary system in which its encryption technique can be employed. Foster, [0030], [0075]. "Source 428" (**green**) can be a "content owner...such as a movie studio." Foster, [0038], [0042], [0078]. "Recipient 430" (**red**) can be a "receiver" such as a "set-top box" or other "consumer device" that receives content from a Content Service Provider, which is "an entity, such as a cable service provider, that provides the 'pipeline' through which content is delivered from a content owner to a consumer." Foster, [0038]-[0041], [0078]. Bhattacharjee, ¶222.



“Computer system 400” disposed between source 428 and recipient 430 is “any system capable of processing content units of an elementary content stream received from source 428.” Foster, [0078]. POSAs understood from the above disclosures that computer system 400 can be part of the “Content Service Provider” that acts as a “pipeline” to deliver data from content owner 428 to recipients 430 and that implements Foster’s encryption system. Foster, [0039]-[0041], [0064]-[0071], [0074]-[0078]; Bhattacharjee, ¶223-224.

B. Hayhurst (EX1007)

Hayhurst is prior art under (1) pre-AIA §102(a) because it published on January 22, 2004 and (2) pre-AIA §102(e) because Hayhurst is the English-language publication (WO2004/008289) of the international application (PCT/US2003/022214) filed on July 16, 2003 under the Patent Cooperation Treaty (PCT) designating the US. Both Hayhurst’s publication and filing dates are before the ’116 patent’s earliest claimed priority date of October 24, 2005.

Hayhurst additionally claims priority to US Provisional Application No. 60/396357 (EX1008, “Hayhurst-Provisional”) filed July 17, 2002, and is entitled to that earlier §102(e) date as discussed in §IX.B.2 *infra*. Should SFI attempt to antedate Hayhurst’s actual filing of July 16, 2003 or its priority date of July 17, 2002, such questions concerning the challenged claims’ “invention date would be more properly considered following trial on a full evidentiary record.” *Quest USA*

v. *PopSockets*, IPR2018-00497, Paper 8 at 11 (PTAB Aug. 13, 2018) (instituting review despite evidence of purported earlier invention).

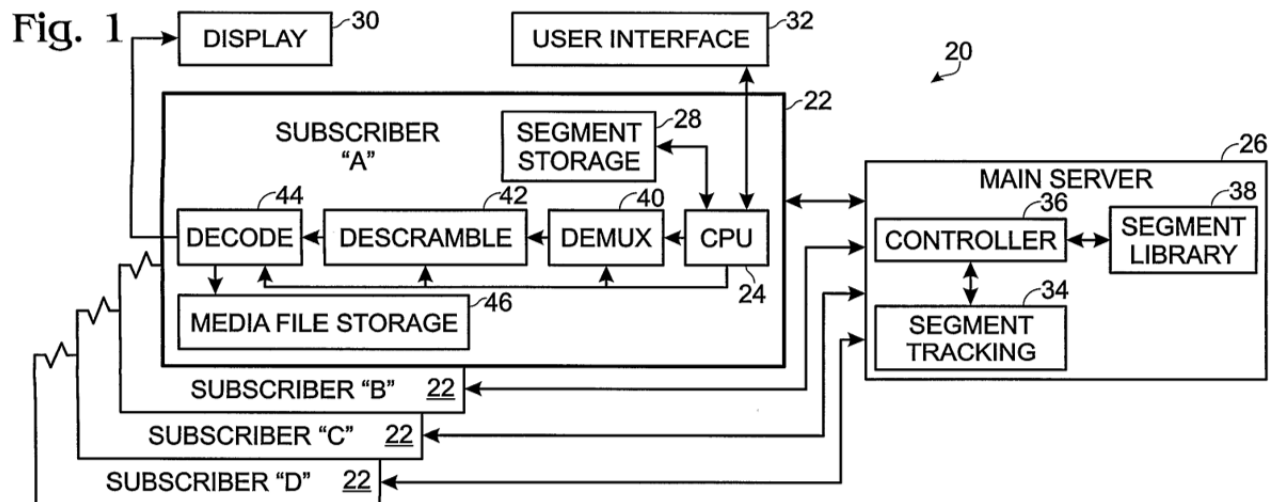
1. Hayhurst Disclosure⁴

Like Foster, Hayhurst concerns “electronic media delivery, such as video-on-demand” and the “secure handling and storage” of media files (e.g., movies). Hayhurst, [0002] (Hayhurst-Provisional, 6). Hayhurst discloses that providing “delivery-on-demand” to subscribers for a large movie library “requires an extensive amount of bandwidth” between the content provider and subscribers. Hayhurst, [0004] (Hayhurst-Provisional, 6); Bhattacharjee, ¶228.

To address that problem and provide other advantages, Hayhurst’s media files are segmented and “distributed among the network of subscriber units.” Hayhurst, [0006] (Hayhurst-Provisional, 6). Subscribers store different segments of a media file accessible to the subscriber base so that when a subscriber seeks access to media (e.g., to view a movie on-demand) the entire content is retrieved from the multiple subscribers that store different segments of the movie. Hayhurst, [0006] (Hayhurst-Provisional, 6); Bhattacharjee, ¶228.

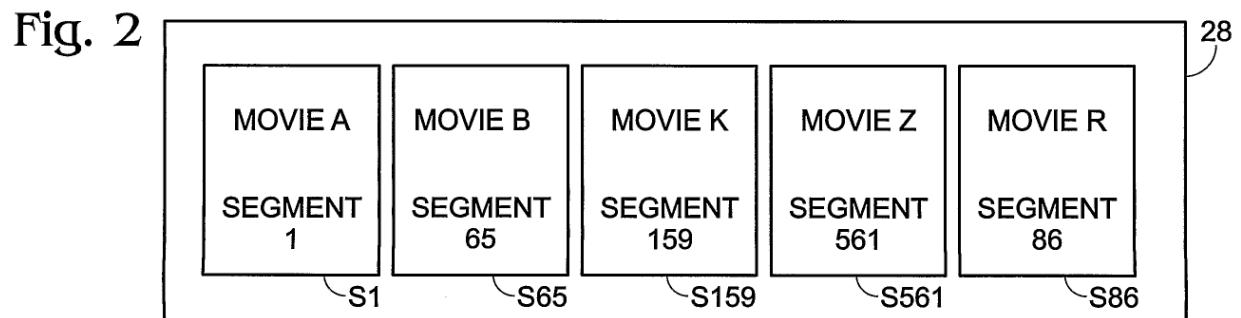
⁴ Citations to the Hayhurst-Provisional (EX1008) supporting the disclosure cited in Hayhurst are provided in parentheses.

Hayhurst's Fig. 1 (below) illustrates a system for "decentralized media delivery" (Hayhurst, [0012]), comprising subscriber units A-D representative of "thousands of subscribers" (Hayhurst, [0016]) "in communication with a main server 26," where each unit is a "conventional set-top box" (just like exemplary "set-top box" receivers at Foster's recipients 430) that includes "a storage device" for storing "segments of one or media files" (Hayhurst, [0018]). Hayhurst, [0016]-[0018] (Hayhurst-Provisional, 6-7, Figure 4); Bhattacharjee, ¶229.



Main server 26 segments a media file (e.g., "digitized data...represent[ing] a movie" or "any type of media file that can be stored and displayed," Hayhurst, [0018]) and distributes the segments to different subscriber units that each store only one segment of the media file. Hayhurst, [0006], [0008], [0018] (Hayhurst-Provisional, 6-7). Each segment "includes data representing a small portion of the movie" and "a header" that stores information about the segment. Hayhurst, [0020] (Hayhurst-Provisional, 7). Bhattacharjee, ¶230.

Hayhurst’s Fig. 2 (Hayhurst-Provisional, Figure 5) reproduced below illustrates the segment storage 28 of a subscriber unit 22 which stores segments from different movies—one segment from each of a plurality of movies. Hayhurst, [0008], [0020] (Hayhurst-Provisional, 7). Hayhurst’s “segments are each encrypted” before being distributed to the subscribers’ storage “to prevent unauthorized viewing of individual segments.” Hayhurst, [0007] (Hayhurst-Provisional, 6); Bhattacharjee, ¶¶231-232.



Hayhurst’s technique advantageously “shifts the bandwidth requirements away from the content provider and to the network of subscriber units” (Hayhurst, [0005]) and “provides a high level of security and a deterrence to piracy” because “[n]o single subscriber has more than a segment of a media file” (Hayhurst, [0008]). Hayhurst, [0005]-[0008] (Hayhurst-Provisional, 6-7); Bhattacharjee, ¶232.

Hayhurst also stores copies of segments on multiple subscriber units so that each segment is available even if “a given subscriber unit is unable to provide the requesting subscriber with the segment stored on that given unit.” Hayhurst, [0009], [0025] (Hayhurst-Provisional, 7); Bhattacharjee, ¶233.

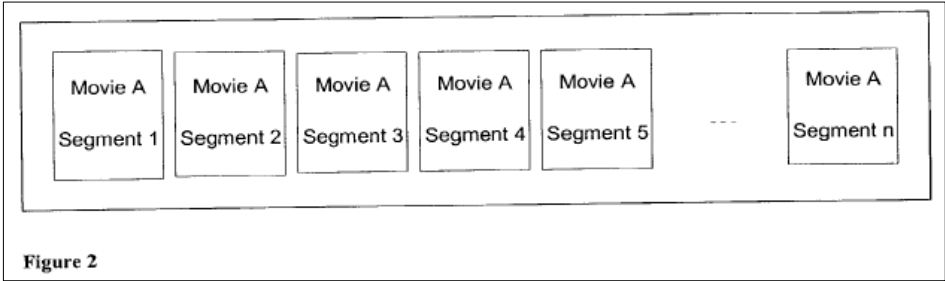
In response to a request from one of the subscribers, main server 26 instructs each subscriber unit storing a segment of the requested movie to send it to the requesting subscriber. Hayhurst, [0007], [0028] (Hayhurst-Provisional, 6). After receiving all movie segments, the requesting subscriber obtains from main server 26 “the encryption key to unlock the movie.” Hayhurst, [0007], [0028] (Hayhurst-Provisional, 6); Bhattacharjee, ¶234.

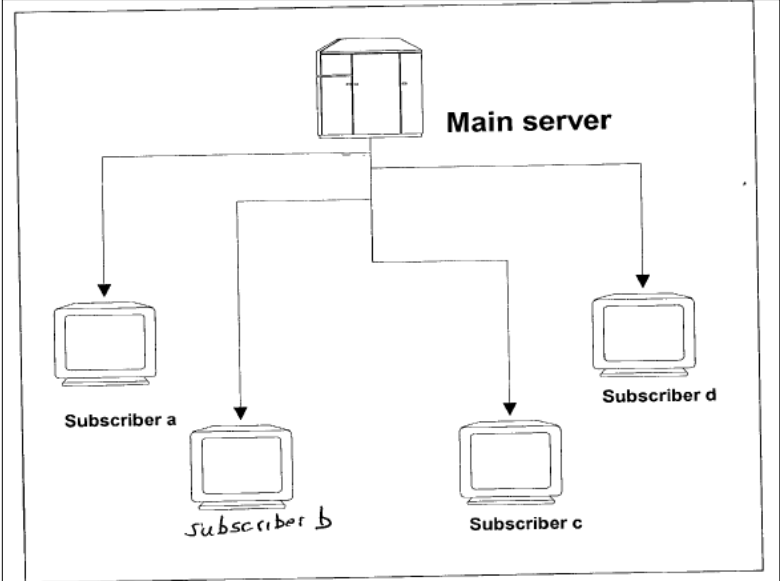
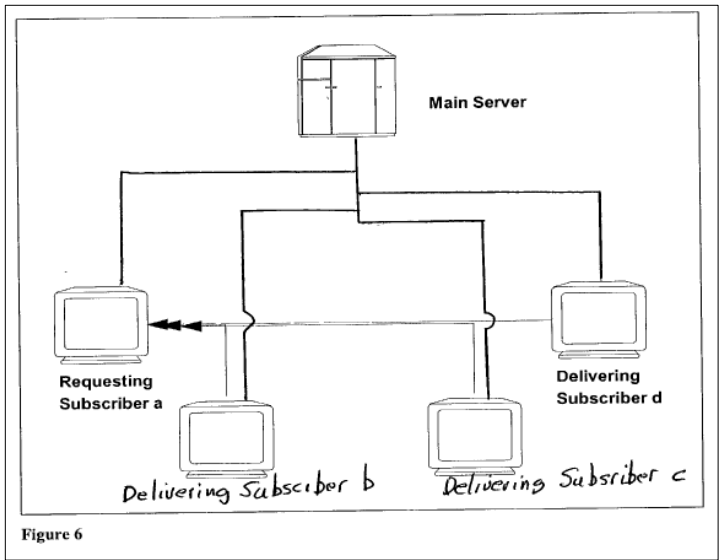
2. Hayhurst was effectively filed as of the Hayhurst-Provisional

Hayhurst is entitled to the July 17, 2002 filing date of the Hayhurst-Provisional because the Hayhurst-Provisional satisfies 35 U.S.C. §119(e). It supports: (1) “at least one claim” (i.e., claim 1) of Hayhurst, and (2) the portions of Hayhurst on which Petitioner relies. *Huawei v. Optis Cellular*, IPR2018-00807, Paper 10 at 14-15 (PTAB Nov. 5, 2018) (citing *Dynamic Drinkware v. Nat’l Graphics*, 800 F.3d 1375, 1378–79, 1383 (Fed. Cir. 2015)).

Hayhurst was filed by “inventors named in the provisional” (sole inventor William Hayhurst) within 12 months of the Hayhurst-Provisional (for which the applicable fee was paid). 35 U.S.C. §119; Hayhurst, (22), (30), (72); EX1009, (Hayhurst-Provisional coversheet); EX1010 (USPTO Application Data).

a. Hayhurst-Provisional Supports Hayhurst's Claim 1

Hayhurst's Claim 1	Hayhurst-Provisional
"A method of delivering a media file that is requested by a requesting subscriber of a network of subscribers, comprising the steps of:"	Title: "Decentralized Media Delivery" Abstract: "A media file is partitioned into small segments, and then distributed in pieces to subscribers' hard drives...The main server sends out a message to all subscribers to send any part of the media that is stored on their system directly to the subscriber who is requesting the media."
"segmenting the media file into a plurality of segments;"	Page 7: "A movie in its entirety is converted into a digital format. <i>The movie is split into segments FIG. 2</i>"  The diagram, labeled Figure 2, shows a horizontal sequence of boxes representing segments of a movie. Each box is divided into two parts: the top part is labeled 'Movie A' and the bottom part is labeled with a segment number. The segments are labeled 'Segment 1', 'Segment 2', 'Segment 3', 'Segment 4', 'Segment 5', followed by an ellipsis '...', and then 'Segment n'. The entire sequence is enclosed in a larger rectangular frame. Figure 2
"distributing the segments of the media file among the network of subscribers for storage on respective units of the subscribers; and"	Page 7: "A server attached through a high-speed connection to the Internet will <i>distribute a segment to hard drives located in the set-top box, or computer or other storage medium of various subscribers a,b,c,d FIG. 4.</i>" Each subscriber receives one movie segment.

Hayhurst's Claim 1	Hayhurst-Provisional
	 Figure 4
“responding to a request from the requesting subscriber by directing to that requesting subscriber all of the plurality of segments of the media file.”	Page 7-8: “When a subscriber, such as subscriber a desires to view a movie, a request is sent either over the network directly to other subscribers or to the main server which then sends a message to the subscribers b,c,d who are storing the segments of the movie on their hard drives. Copies of the movie segments are then sent directly to the requesting subscriber a from the subscribers b,c,d who have stored segments on their hard drives FIG. 6.”  Figure 6

b. Hayhurst-Provisional Supports Hayhurst’s Relied-Upon Subject-Matter

As the table *supra* §IX.B.2.a demonstrates, the Hayhurst-Provisional discloses *inter-alia*: (1) segmenting a media file; (2) storing media-file segments on “storage medium” of different subscribers; (3) and directing all of the media-file segments to a requesting subscriber (Hayhurst-Provisional, 6-8)—i.e., the same distributive-storage technique disclosed in Hayhurst. Bhattacharjee, ¶¶235-237; *see also* §IX.B.1 *supra* (citing Hayhurst-Provisional’s disclosure corresponding to Hayhurst’s).

As further demonstrated by citations to the Hayhurst-Provisional supporting Hayhurst’s relied-upon disclosure in (1) the Foster-Hayhurst combination (§IX.C *infra*) and (2) mapping Foster-Hayhurst to the challenged claims (§§IX.D-IX.H *infra*), the Hayhurst-Provisional supports all the relied-upon disclosure in Hayhurst. Bhattacharjee, ¶238.

C. Foster-Hayhurst

1. Reasons to Combine

Foster (§IX.A *supra*) discloses a system in which a content provider delivers encrypted digital content to consumers (e.g., to consumers’ set-top boxes), wherein Foster’s encryption techniques can be implemented in the content provider computer system. Foster, [0039], [0043], [0064], [0075]-[0080]; Bhattacharjee,

¶240. POSAs would have had numerous reasons to adopt Hayhurst’s distributive storage technique in Foster’s system so that Foster’s encrypted content units for a media file would be distributively stored on different consumer devices (e.g., consumer set-top boxes) as Hayhurst teaches. *Id.*

First, Foster’s content provider is responsible for delivering content to consumer devices. Bhattacharjee, ¶222-224, 241; §IX.A. Hayhurst teaches that such systems can be improved for the content provider by storing media-file segments (like Foster’s content units) on multiple subscriber units (like Foster’s receivers) because Hayhurst’s technique: (1) solves problems with on-demand delivery of a media library by shifting the bandwidth requirement in delivering digital content away from the content provider (Hayhurst, [0005]; Hayhurst-Provisional, 6); (2) “provides a high level of security and a deterrence to piracy” (Hayhurst, [0008], Hayhurst-Provisional, 6); and (3) supports redundancy in storage (Hayhurst, [0009], Hayhurst-Provisional, 7). Bhattacharjee, ¶¶241-242; §IX.B.1 *supra*. Thus, Hayhurst explicitly motivates a POSA to apply Hayhurst’s distributed-storage technique to Foster’s system to improve content delivery in multiple ways. Bhattacharjee, ¶¶241-242.

For example, in Foster’s embodiments in which the content service provider implements Foster’s encryption system (*supra* §IX.A), POSAs would have been motivated to configure Foster’s content provider to distributively store Foster’s

segmented content units on multiple devices as taught by Hayhurst to achieve Hayhurst's explicitly taught benefits (enumerated above) in improving delivery of digital content (e.g., "delivery-on-demand [of] a large number of movie selections" Hayhurst, [0004]-[0010]) and well-known benefits of distributive-storing techniques like Hayhurst's, including improvements to: (1) speed of accessing content by a subscriber because segments would be retrieved from multiple sources, thus avoiding single-source bottlenecks; and/or (2) efficiency by intelligently selecting storage device(s) that would provide the best delivery based on availability, location, etc. Bhattacharjee, ¶¶242-244 (*citing* as corroboration EX1028, [0022]-[0023]; EX1031, [0003], [0012]-[0013], [0015], [0018]-[0024]; EX1030, [0005]-[0010]; EX1029, 1:19-22, 2:31-38, 11:11-26, 15:39-16:-20).

Second, the Foster-Hayhurst combination is nothing but an obvious combination of prior-art elements—*i.e.*, Foster's encryption system to generate encrypted data segments, and Hayhurst's method for storing encrypted data segments on multiple distributed storage devices—according to known encryption and storage methods—to yield the predictable result of Foster's encrypted content units being stored on and retrievable from distributed storage devices as taught in Hayhurst. *KSR*, 550 U.S. at 416 ("The combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results."); Bhattacharjee, ¶245.

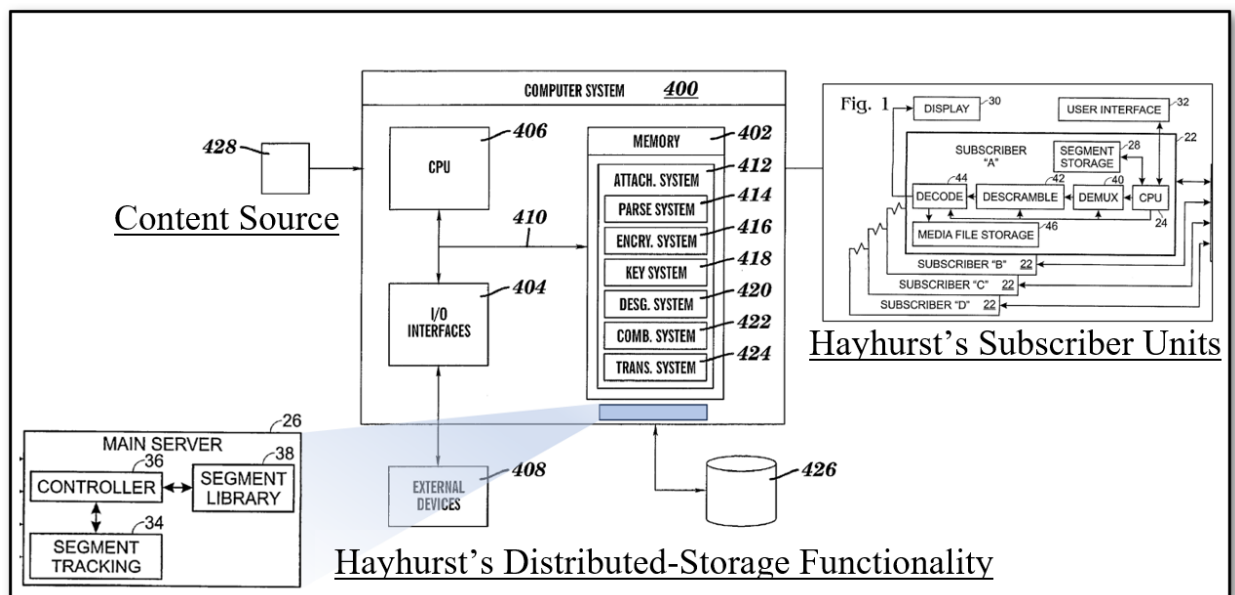
Third, Foster-Hayhurst is also nothing more than the obvious application of Hayhurst’s known distributed-storage technique to Foster’s system, which was ready for the improvement of distributed-storage techniques that addressed bandwidth issues, increased security, added redundancy and/or enabled economic on-demand delivery of large selections of content in the same Hayhurst achieved those benefits. *KSR*, 550 at 417-18 (“[I]f a technique has been used to improve one device, and a [POSA] would recognize that it would improve similar devices in the same way, using the technique is obvious unless its actual application is beyond his or her skill.”); *Intel Corp. v. PACT XPP Schweiz AG*, 61 F.4th 1373, 1380-81 (Fed. Cir. 2023); Bhattacharjee, ¶¶244, 246.

2. The Foster-Hayhurst Combination

The Foster-Hayhurst combination uses (1) Foster’s technique of securing digital content by encrypting parceled segments of the content with respective different encryption keys that are themselves encrypted with a KEK (Foster, [0064]-[0071]); (2) Hayhurst’s technique of storing encrypted-media segments in a distributed manner across a network of subscriber devices (Hayhurst, [0006]-[0008], [0016]-[0023]; Hayhurst-Provisional, 6-7); and (3) Hayhurst’s technique of redundantly storing media segments on the subscriber units (Hayhurst, [0025]-[0028]); Hayhurst-Provisional, 7). Bhattacharjee, ¶247.

One exemplary implementation of Foster-Hayhurst is depicted below (based on Foster's Figure 16 and Hayhurst's Figure 1) in which Foster's "recipients 430" are implemented on "subscriber units" as taught by Hayhurst, and Foster's content service provider implemented using "computer system 400" is augmented to include functionality of Hayhurst's main server 26 to control distribution of, and access to, Foster's encrypted content portions stored on the subscriber units.

Hayhurst, [0017], [0023]-[0024], [0026], FIGS. 1, 4 (Hayhurst-Provisional, 6-8, FIGS. 4, 6); Foster, [0040]-[0041], [0075]-[0080], FIG. 16; Bhattacharjee, ¶248.



Specifically, in Foster-Hayhurst's content provider (Foster's computer system 400 implementing the Hayhurst main server's distributive and redundant storage functionality), parse system 414 and encryption system 416 perform Foster's Fig. 8 (annotated below) technique of: **(1)** parceling digital content into individual content units 104, **(2)** encrypting each content packet ("payload") 106

with a different title key 108, (3) encrypting each title key 108 with key-encrypting key 110, and (4) attaching the resulting encrypted key 112 (e.g., incorporated in extension 118) to encrypted-content packet 114 with header 104 to generate encrypted content unit 116. Foster, [0064]-[0071], [0074]-[0079]; Bhattacharjee, ¶249.

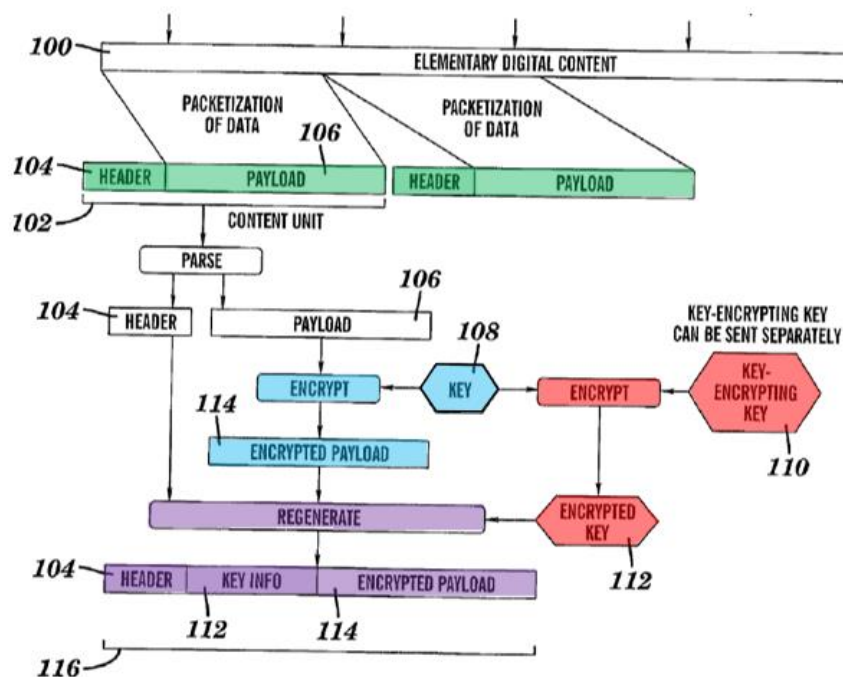


FIG. 8

The Foster-Hayhurst content provider then distributes the processed content units 116 to different respective Foster-Hayhurst subscriber units (Foster's recipients 430 adapted to send stored content units to a requesting subscriber as taught by Hayhurst), including making copies of each processed content unit 116 so that each processed content unit 116 is redundantly stored as taught by Hayhurst. Bhattacharjee, ¶¶250-251; §§IX.B.1, IX.C.1.

3. Reasonable Expectation of Success

POSAs would have reasonably expected success in implementing the above-discussed Foster-Hayhurst combination using ordinary computer programming skill to distribute Foster's encrypted content units to multiple storage devices based on Hayhurst's teachings. Bhattacharjee, ¶252.

Both references teach segmenting digital content (e.g., a video or movie), encrypting segments and delivering encrypted content from a content service provider to consumer devices (e.g., a consumer "set-top box," (Foster, [0040], Hayhurst, [0017] (Hayhurst-Provisional, 7). POSAs understood that Foster's content units are individually accessible and agnostic to where and how they are stored with respect to other stored encrypted content units, so Foster's system is compatible with storing the encrypted content in a distributed manner across multiple storage devices. Bhattacharjee, ¶¶252-253; Foster, [0040]-[0041], [0061]; Hayhurst, [0023]-[0029], [0031] (Hayhurst-Provisional, 6-8).

Foster-Hayhurst renders obvious claims 1-2, 5, 9 and 11.

D. Claim 1

1. 1[PRE]

Foster-Hayhurst (§IX.C *supra*) performs "*a method for securing a data set*" by encrypting digital content (i.e., a "*data set*" such as "video data," Foster, [0064]) using Foster's encryption technique that "provides enhanced security" by using multiple encryption keys (Foster, [0004], [0071]) and provides a "high level

of security and a deterrence to piracy” (Hayhurst, [0008], Hayhurst-Provisional, 6) by applying Hayhurst’s distributed storage technique to Foster’s “encrypted content.” Foster, [0064]-[0068], [0071]; Hayhurst, [0006]-[0008], [0020] (Hayhurst-Provisional, 6-7); Bhattacharjee, ¶254. Foster’s digital content is a “*data set*” under both parties’ constructions because POSAs understood that, e.g., a video file is a “collection of related information made up of separate elements [e.g., video frames] that can be treated as a unit [e.g., a file] in data handling” (Google; EX1042, 9-10) and “a collection of information for storage” (SFI; EX1043, 6-7). Bhattacharjee, ¶254.

2. [1A] distributing...

Foster-Hayhurst (§IX.C.2) performs Foster’s encryption method, which as illustrated in Foster’s Figure 8 (annotated below), generates a plurality of “payloads” by “parcel[ing]” digital content 100 (*i.e.*, the claimed “*data set*,” §IX.D.1 1[PRE]). Foster, [0064]. Specifically, digital content 110 (blue) is “parceled into content units 102” that each includes a “content packet” (*id.*) or “payload” 106 that is a portion of the digital content 100 and a header 104 that includes information about the payload (e.g., a timestamp) that can be used to reconstruct the digital content 100 from the content units 102 into which it is parceled. *Id.*; Bhattacharjee, ¶255.

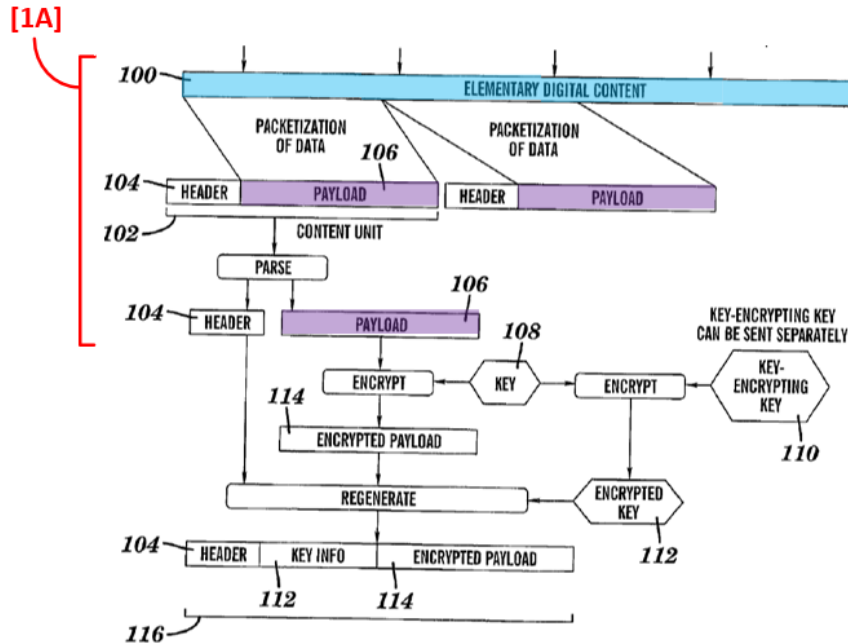


FIG. 8

Each content unit 102 is then parsed to separate the header 104 and content packet 106 (illustrated in Fig. 8 as payload 106 highlighted in **purple**). These “content packets”/“payloads” 106 are “a plurality of data chunks” as claimed because each includes only a portion of digital content 100 (i.e., a subset of the data set), which is all that is required of a data chunk as discussed in §V.B.2 above. Bhattacharjee, ¶256. Foster-Hayhurst thus meets both parties’ proposed constructions of “*distributing the data set...*” because Foster is “splitting” (Google) and “dividing and placing” (SFI) “*the data set into a plurality of chunks*” (EX1042, 2-5; EX1043, 2) that are each distributed to different subscribers after being encrypted (§IX.C.2). Bhattacharjee, ¶¶256-257. None of the payloads 106 “*are, by*

themselves, sufficient to reconstruct the data set,” as recited in [1A], because each payload is only a subset of the data set and not the entire digital content. *Id.*

3. [1B] encrypting...

Foster-Hayhurst (*supra* §IX.C.2) encrypts each payload 106 with a different title key 108. Foster, [0064]-[0065]. As illustrated in steps boxed-red in Fig. 8 below, Foster discloses that after parceling, “content packet 106 is **encrypted** with title key 108 to yield encrypted content packet 114.” Foster, [0064]. “[E]ach content packet can be encrypted with a **different** title key.” Foster, [0071], [0065]. Bhattacharjee, ¶258.

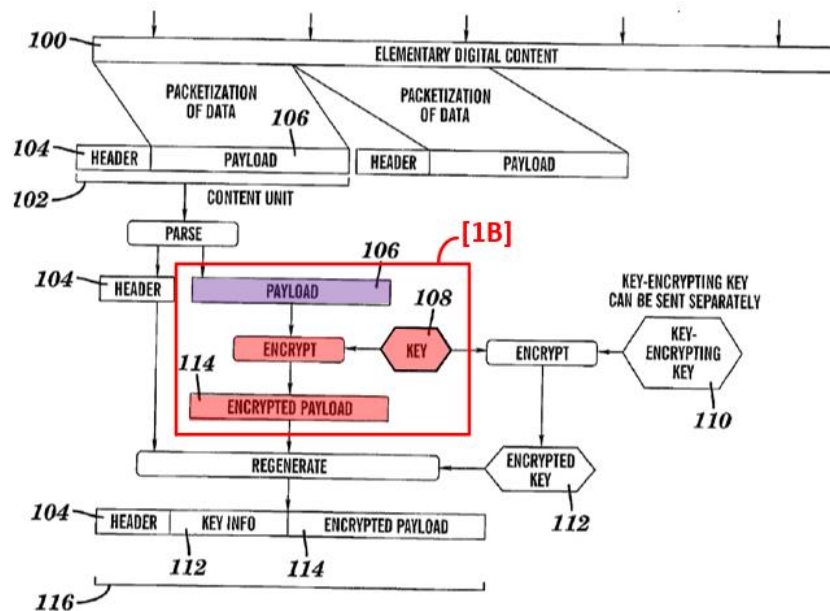


FIG. 8

Thus, Foster-Hayhurst performs “*encrypting each of the data chunks with a respective one of a plurality of encryption keys,*” as recited in [1B] because it uses

Foster's technique of encrypting each content packet (payload) with a different title key. Foster, [0064]-[0068], [0071]; Bhattacharjee, ¶259; §IX.C.2 *supra*.

4. [1C] obfuscating...

Foster-Hayhurst (§IX.C.2) uses Foster's technique of encrypting each title key 108 with a key encrypting key (KEK) 110 to yield encrypted key 112, as highlighted in **red** in Foster's Fig. 8 below. Foster, [0064]-[0065]; Bhattacharjee, ¶260.

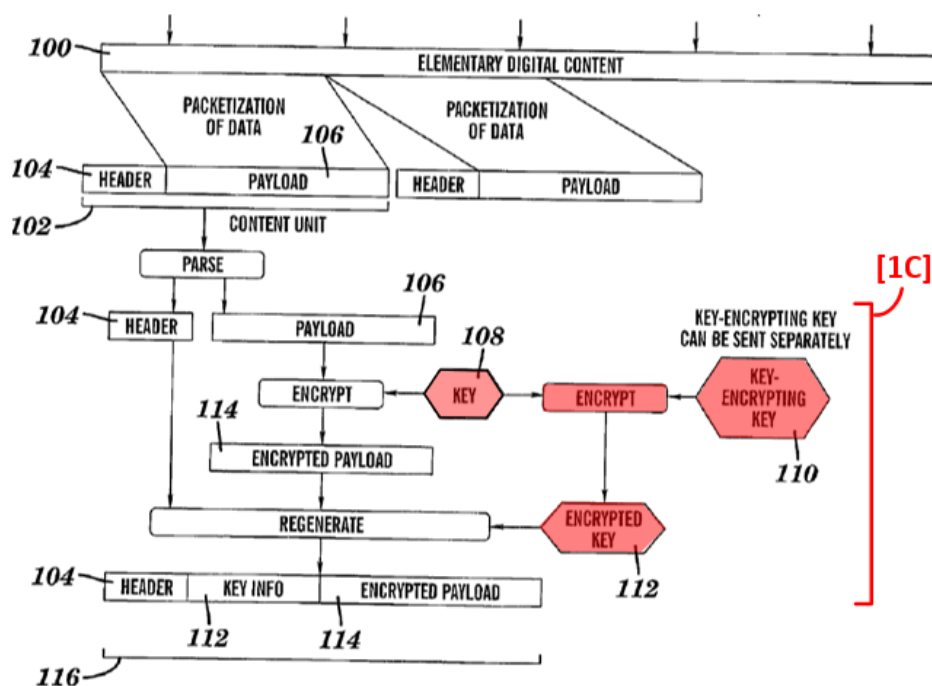


FIG. 8

As discussed above in §VI.F.2, SFI alleged that “*obfuscating each of the plurality of different encryption keys*” is met by encrypting the different encryption keys with a KEK (EX1012, 4-5; EX1011, ¶117). Thus, Foster-Hayhurst meets [1C] under SFI’s interpretation by encrypting each of the different title keys 108 with

KEK 110 (Foster, [0064]-[0071]), thereby also rendering the keys “unclear or obscure” as SFI construes an “obfuscated” encryption key (EX1043, 5).

Bhattacharjee, ¶¶261-262; §IX.C.2 *supra*.

5. [1D] separately storing...

Foster-Hayhurst (§IX.C.2 *supra*) meets [1D] because it applies Hayhurst’s distributive storing techniques to the encrypted content units 116 of Foster’s digital content. Hayhurst, [0006]-[0007], [0016]-[0020], [0025] (Hayhurst-Provisional, 6-8); Foster, [0043], [0064]-[0071]; Bhattacharjee, ¶263.

a. Foster-Hayhurst meets [1D] the same way the ’116 patent’s embodiments do

As illustrated in Foster’s Fig. 8 (annotated below), after encrypting each payload with a different title key 108 and then encrypting the title key with a KEK 110, Foster “regenerates” the parcel by attaching the encrypted key 112 (highlighted **red**) and the header (highlighted **green**) to the corresponding encrypted content package (highlighted **purple**) to produce the “processed content unit 116” (boxed **red**) that Foster discloses is stored by the content provider or transmitted to a receiver (e.g., a set-top box). Foster, [0039]-[0040], [0064]-[0068]; Bhattacharjee, ¶264.

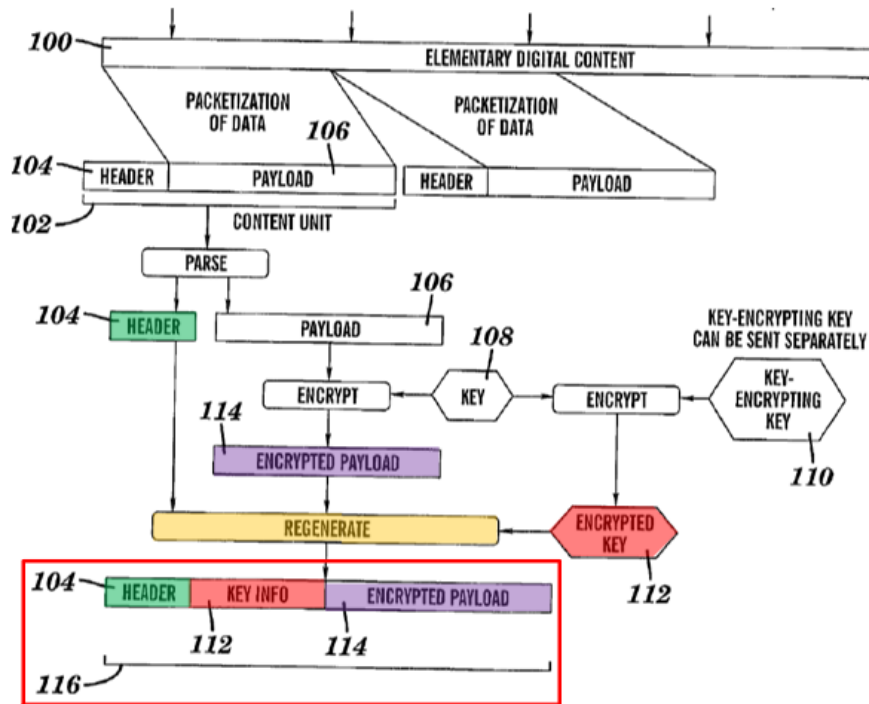


FIG. 8

In Foster-Hayhurst (§IX.C.2), each encrypted content unit (i.e., “processed content unit 116”) is stored on a different Foster-Hayhurst subscriber unit, as taught by Hayhurst, which “can be adapted from a conventional set-top box,” [0016]). Hayhurst, [0006]-[0007], [0016]-[0020], (Hayhurst-Provisional, 6-7); Bhattacharjee, ¶265; §IX.C.2 *supra*.

Thus, by storing each processed content unit 116, which includes the respective different encrypted keys 112 (i.e., the respective “*obfuscated different encryption key*”) attached to its corresponding encrypted content packet 114 (i.e., a “*data chunk*”) on a different subscriber unit, Foster-Hayhurst meets [1D] because

[1D] does not require redundant storing (see §§V.B.5.a-V.B.5.b *supra*; EX1043,5).
Bhattacharjee, ¶265.

**b. Foster-Hayhurst meets [1D] if construed to require
redundantly storing**

Foster-Hayhurst (§IX.C.2) uses Hayhurst’s redundant storage of encrypted segments on different subscriber units—i.e., *each* of Foster’s encrypted content units is “copied and distributed to a *plurality* of different subscribers to provide for redundant availability” in case a subscriber unit “becomes inaccessible” (Hayhurst, [0025]; Hayhurst-Provisional, 7)—and therefore meets [1D] even if construed to require redundantly storing each processed content unit 116. Bhattacharjee, ¶266.

E. Claim 2

Foster-Hayhurst (§IX.C.2) meets claim 2 by encrypting each encryption key with a KEK (Foster, [0064]-[0065]) under SFI’s interpretation that “*data indicative of*” a key may be the key itself and that encrypting an encryption key transforms it (changes its form) as discussed *supra* §VI.G. EX1012, 8 (SFI alleging that “Google performs a transform on each of the different encryption keys when it encrypts the DEKs using the KEKs”); EX1043, 3-4 (data indicative of a key “makes known” or “shows” the key), 5-6 (transforming is “changing the form”); Bhattacharjee, ¶267.

F. Claim 5:

Foster-Hayhurst (§IX.C.2) attaches each encrypted key 112 (each “*obfuscated different encryption key*”) to its corresponding encrypted content packet 114 (i.e., “*the plurality of data chunks*”) as taught by Foster (e.g., Foster, [0064]-[0071]) and stores each resulting processed content unit 116 on a different subscriber unit as taught by Hayhurst (e.g., Hayhurst, [0006]-[0007], [0020], Hayhurst-Provisional, 6-7). Foster-Hayhurst thus meets claim 5. Bhattacharjee, ¶268.

G. Claim 9

1. 9[PRE]-[9B]

9[PRE]-[9B] are identical to 1[PRE]-[1B] and are met by Foster-Hayhurst for the same reasons demonstrated in §§IX.D.1-IX.D.3. Bhattacharjee, ¶269.

2. [9C] transforming...

Foster-Hayhurst meets [9C] in two ways. First, Foster-Hayhurst (§IX.C.2) encrypts each different encryption key 108 with KEK 110, which meets [9C] under the interpretation advanced by SFI that encrypting an encryption key transforms it by changing its form. EX1012, 12 (alleging [9C] is met by “encrypt[ing] the DEKs using the KEKs”); EX1043, 5-6 (“changing the form” is “transforming”); Bhattacharjee, ¶270; *see also* §VI.H.2 *supra*.

Second, Foster-Hayhurst (§IX.C.2) transforms the encryption keys by masking the key information—i.e., modifying “part or all of the key information so

as to distinguish it from a known code that may have a specialized purpose.”

Foster, [0066]. Foster’s Fig. 9 annotated below shows header extension 118 with information 120 (shaded red) used to indicate that the key information has been modified (e.g., inverted) to avoid aliasing. *Id.*; Bhattacharjee, ¶271.

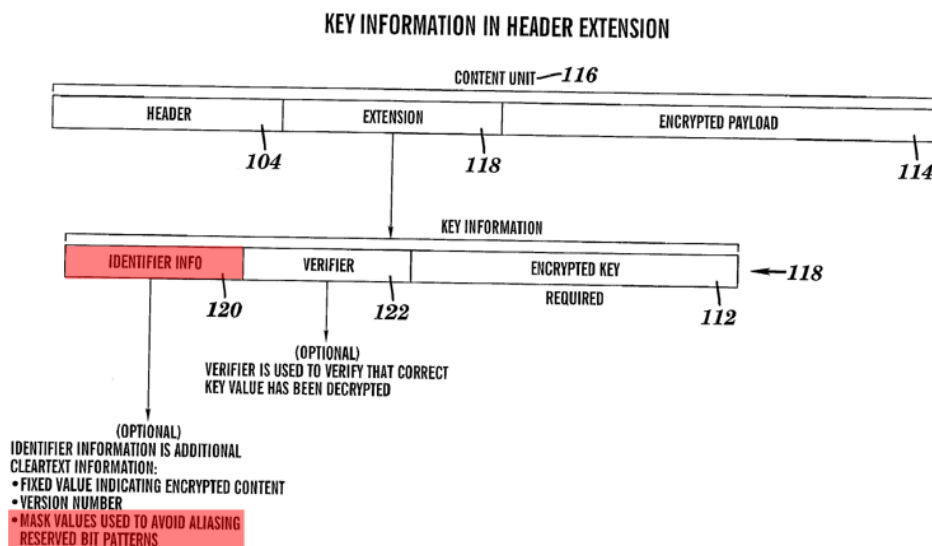


FIG. 9

Inverting the bits of the key information transforms Foster’s encryption key stored therein by “changing the form” of the encryption key under SFI’s construction (EX1043, 5-6) and by “changing without the use of a secret key the bits of the key” under Google’s construction (EX1042, 8). Bhattacharjee, ¶272.

3. [9D] separately storing...

Foster-Hayhurst meets [9D] for the same reasons Foster-Hayhurst meets [1D] (§IX.D.5 *supra*). Bhattacharjee, ¶273. While [9D] differs from [1D] in that the stored encryption keys are “transformed” encryption keys rather than “obfuscated,” Foster’s encrypted key 112 is both a “transformed” and an

“*obfuscated*” encryption key under SFI’s interpretation (see §§IX.D.4 and IX.G.2 *supra*) and is a “*transformed*” encryption key by inverting the key information (see §IX.G.2 *supra*), and the key/chunk pairs are distributively stored on respective Foster-Hayhurst subscriber units. Bhattacharjee, ¶273; §IX.C.2 *supra*.

Foster-Hayhurst’s use of Hayhurst’s redundant data storage technique (see §IX.C.2 *supra*) meets [9D] even if construed to require redundantly storing each processed content unit 116 as discussed above in connection with claim 1’s limitation [1D] (§IX.D.5.b *supra*). Bhattacharjee, ¶274.

H. Claim 11

Foster-Hayhurst (§IX.C.2 *supra*) attaches each encrypted key 112 (i.e., each “*transformed different encryption key*”) to its corresponding encrypted content packet 114 (i.e., “*the plurality of data chunks*”) and stores each resulting processed content unit 116 on a different subscriber unit. *See, e.g.*, Hayhurst, [0008] (Hayhurst-Provisional, 6). Foster-Hayhurst thus meets claim 11. Bhattacharjee, ¶275.

X. NO BASIS EXISTS FOR DISCRETIONARY DENIAL

A. Section 314(a)

“[T]he PTAB will not deny institution based on *Fintiv* if,” as here, “there is compelling evidence of unpatentability.” Director’s Interim Procedure for Discretionary Denials, 5 (June 21, 2022) (“Interim Procedure”) (discussing *Apple*

Inc. v. Fintiv, Inc., IPR2020-00019, Paper 11 (Mar. 20, 2020) (precedential)). That should conclude the *Fintiv* analysis.

If considered, the *Fintiv* factors also weigh against denial.

1. Stay Potential

Petitioner will seek a pre-institution stay. Without “specific evidence” how the court will rule, this factor is at worst neutral. *Sand Revolution II, LLC, v. Cont’l Intermodal Group–Trucking*, IPR2019-01393, Paper 24, 7 (June 16, 2020) (informative).

2. Trial Timing

The district-court trial is presently scheduled for July 2024. EX1038, 1. This factor is not dispositive especially where, as here, the petition presents strong merits. *See, e.g., PEAG LLC d/b/a JLab Audio v. VARTA Microbattery GmbH*, IPR2020-01214, Paper 8, 19-20 (Jan. 6, 2021).

3. Litigation Investment

Though there has been some activity in the district-court litigation, Petitioner’s diligence in filing this Petition about eight months from being served with a complaint and about five months after SFI identified which claims would be asserted weighs against discretionary denial. *Fintiv*, 11 (expeditious filing weighs against denial); *Dish Network LLC v. Broadband iTV, Inc.*, IPR2020-01332, Paper 14, 28-29 (Jan. 27, 2021) (finding petition diligently filed within three months of

receiving patent owner’s full set of asserted claims); EX1009 (SFI’s August 14, 2023 interrogatory response identifying asserted claims).

4. Issue Overlap

Given its early stage, the validity issues in the litigation are unknown. However, Petitioner challenges 13 claims, whereas only five claims are yet asserted in the litigation. *See* EX1012. This weighs against denial. *Markforged Inc. v. Continuous Composites*, IPR2022-00679, Paper 7, 32-33 (Oct. 25, 2022).

5. Litigation Defendants

That Petitioner is the defendant in the district-court proceeding “is far from an unusual circumstance.” *Sand Revolution II*, 12-13. This factor is neutral.

6. Other Circumstances

The Petition’s “strong showing on the merits” weighs against denial. Interim Procedure, 3-5.

B. Section 325(d)

No basis exists for denial under §325(d) for three independently sufficient reasons: (1) Grounds 1-2 and 5 present all new art (Tajima, Foster, Hayhurst) not considered during prosecution;⁵ (2) Grounds 3-4 present new art (Tajima, Shyu) in

⁵ *ShenZhen Apaltek Co., Ltd. v. Asetek Danmark A/S*, IPR2022-01317, Paper 7, 8-10 (Feb. 6, 2023)

combination with “previously considered” art (Rivest);⁶ and (3) Rivest was cited in an IDS (with 500+ references) but never discussed nor applied in any rejection.⁷

XI. CONCLUSION

The Board should institute review and cancel claims 1-3 and 5-14.

⁶ *Cisco Sys., Inc. v. Ramot at Tel Aviv Univ. Ltd.*, IPR2022-01283, Paper 12, 12-13 (Feb. 22, 2023)

⁷ *DJI Europe BV v. Textron Innovations, Inc.*, IPR2022-00453, Paper 10, 10-11 (Aug. 3, 2022)

Dated: 2023-11-24

Respectfully submitted,
Google LLC

/Elisabeth Hunt/

Elisabeth H. Hunt, Reg. No. 67,366
WOLF, GREENFIELD & SACKS, P.C

XII. APPENDIX: CLAIM LISTING

The following claim listing assigns element labels (*e.g.*, 1[PRE], [1A], etc.) to certain claims for convenience of reference.

Claim 1
1[PRE]. The method of claim 9, further comprising generating the data by encrypting primary data.
[1A] distributing the data set into a plurality of data chunks, wherein none of the data chunks are, by themselves, sufficient to reconstruct the data set;
[1B] encrypting each of the data chunks with a respective one of a plurality of encryption keys;
[1C] obfuscating each of the plurality of different encryption keys; and
[1D] separately storing each data chunk of the plurality of data chunks together with one of the plurality of obfuscated different encryption keys on a plurality of different storage devices.
Claim 2
2. The method of claim 1, wherein obfuscating each of the plurality of different encryption keys comprises performing a transform on data indicative of at least one of the plurality of different encryption keys.
Claim 3
3. The method of claim 2, wherein performing a transformation on data indicative of at least one of the plurality of different encryption keys comprises performing an all-or-nothing transform on the data indicative of at least one of the plurality of different encryption keys.
Claim 5
5. The method of claim 1, wherein storing the plurality of data chunks together with the plurality of obfuscated different encryption keys comprises storing each of the plurality of data chunks together with a respective one of the plurality of obfuscated different encryption keys.
Claim 6
6. The method of claim 1, wherein storing the plurality of data chunks together with the plurality of obfuscated different encryption keys comprises including, with a data chunk of the plurality of data chunks, data indicative of an encryption key that was used to encrypt a different data chunk of the plurality of data chunks.

Claim 7
7. The method of claim 1, wherein distributing the data set into a plurality of data chunks comprises using a random technique or pseudorandom technique.
Claim 8
8. The method of claim 1, further comprising generating the data set by encrypting primary data.
Claim 9
9[PRE]. A method for securing a data set, the method comprising:
[9A] distributing the data set into a plurality of data chunks, wherein none of the data chunks are, by themselves, sufficient to reconstruct the data set;
[9B] separately storing each data chunk of the plurality of data chunks together with one of the plurality of obfuscated different encryption keys on a plurality of different storage devices.
[9C] transforming each of the plurality of different encryption keys; and
[9D] separately storing each data chunk of the plurality of data chunks together with one of the plurality of transformed different encryption keys on a plurality of different storage devices.
Claim 10
10. The method of claim 9, wherein transforming each of the plurality of different encryption keys comprises performing an all-or-nothing transform on data indicative of at least one of the plurality of different encryption keys.
Claim 11
11. The method of claim 9, wherein storing the plurality of data chunks with the plurality of transformed different encryption keys comprises storing each of the plurality of data chunks with a respective one of the plurality of transformed different encryption keys.
Claim 12
12. The method of claim 9, wherein storing the plurality of data chunks with the plurality of transformed different encryption keys comprises including, with a data chunk of the plurality of data chunks, data indicative of an encryption key of the plurality of encryption keys that was used to encrypt a different data chunk of the plurality of data chunks.
Claim 13
13. The method of claim 9, wherein distributing the data set into a plurality of data chunks comprises using a random technique or pseudorandom technique.
Claim 14
14. The method of claim 9, further comprising generating the data by encrypting primary data.

CERTIFICATE OF SERVICE UNDER 37 C.F.R. § 42.6 (E)(4)

I certify that on November 24, 2023, I will cause a copy of the foregoing document, including any exhibits or appendices filed therewith, to be served via Overnight FedEx at the following correspondence address of record for the patent:

SECURITY FIRST INNOVATIONS, LLC
C/O FARJAMI & FARJAMI LLP
26522 LA ALAMEDA AVE., SUITE 360
MISSION VIEJO, CA 92691

Date: November 24, 2023

/MacAulay Rush/
MacAulay Rush
Paralegal
WOLF, GREENFIELD & SACKS, P.C.

CERTIFICATE OF WORD COUNT

Pursuant to 37 C.F.R. § 42.24, the undersigned certifies that the foregoing Petition for *Inter Partes* Review contains 13,990 words excluding a table of contents, a table of authorities, Mandatory Notices under § 42.8, a certificate of service or word count, or appendix of exhibits or claim listing. Petitioner has relied on the word count feature of the word processing system used to create this paper in making this certification.

Date: November 24, 2023

/MacAulay Rush/
MacAulay Rush
Paralegal
WOLF, GREENFIELD & SACKS, P.C.