

Filed on behalf of Google LLC by:

Paper No. ____

Elisabeth H. Hunt, Reg. No. 67,336
Gregory S. Nieberg, Reg. No. 57,063
Nathan R. Speed (admitted *pro hac vice*)
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. IPR2024-00214
Patent No. 11,068,609

PETITIONER'S REPLY TO PATENT OWNER RESPONSE

TABLE OF CONTENTS

I.	Ground 2: All Claims Are Unpatentable over Foster.....	1
	A. Foster Stores Encrypted Content.....	1
	B. Foster’s Payloads Are <i>Data Chunks</i>	3
	C. Foster Discloses and Renders Obvious Executing Code To Store a Data Chunk on a Memory Coupled to its Processor.....	6
	1. Anticipation	6
	2. Obviousness	11
	D. Foster Discloses and Renders Obvious “Coupl[ing] to the Processor” Storage-Device Memory.....	12
	1. Anticipation	12
	2. Obviousness	14
II.	Ground 1: All Claims Are Unpatentable over Orsini.....	18
	A. POSAs Would Have Encrypted Orsini’s Share Keys Using a Conventional KEK.....	20
	1. The Board Misunderstood the Petition’s Argument.....	22
	2. Petitioner Does Not Rely on Common Sense To Supply a Missing Limitation.....	23
	3. SFI’s Challenges to Petitioner’s Motivation-to-Combine Theory Fail.....	24
	a. POSAs Would Have Wanted to Increase Orsini’s Security.	24
	b. The Alleged Drawbacks to KEKs Were Easily Overcome and Are Legally Irrelevant.	27
	c. The Identified Motivation To Use a KEK in Orsini Is Grounded in Evidence, Not Hindsight.....	29
	B. SFI’s Reasonable Expectation of Success Challenge Fails.	29

TABLE OF AUTHORITIES

Cases

<i>Arendi S.A.R.L. v. Apple, Inc.</i> , 832 F.3d 1355 (Fed. Cir. 2016)	23
<i>Axonics, Inc. v. Medtronic, Inc.</i> , 75 F.4th 1374 (Fed. Cir. 2023)	4
<i>Cole Kepro Int’l, LLC v. VSR Indus., Inc.</i> , 2017 WL 2627982 (Fed. Cir. 2017)	26
<i>Corephotonics, Ltd. v. Apple Inc.</i> , 2021 WL 4944471 (Fed. Cir. Oct. 25, 2021)	18, 28
<i>CyWee Grp. Ltd. v. ZTE (USA), Inc.</i> , 90 F.4th 1358 (Fed. Cir. 2024)	28
<i>DSS Tech. Mgm’t v. Apple Inc.</i> , 885 F.3d 1367 (Fed. Cir. 2018)	23
<i>Fleming v. Cirrus Design Corp.</i> , 28 F.4th 1214 (Fed. Cir. 2022)	24
<i>Google LLC v. Nobots LLC</i> , IPR2022-00940, Paper 38 (Nov. 23, 2023)	30
<i>In re Acad. of Sci. Tech Ctr.</i> , 367 F.3d 1359 (Fed. Cir. 2004)	14
<i>In re Cree, Inc.</i> , 818 F.3d 694 (Fed. Cir. 2016)	28
<i>In re Epstein</i> , 32 F.3d 1559 (Fed. Cir. 1994)	11
<i>Intel Corp. v. PACT XPP Schweiz AG</i> , 61 F.4th 1373 (Fed. Cir. 2023)	21-22, 28
<i>Keynetik, Inc. v. Samsung Elecs. Co., Ltd.</i> , 2023 WL 2003932 (Fed. Cir. 2023)	12

<i>Koninklijke Philips N.V. v. Google LLC</i> , 948 F.3d 1330 (Fed. Cir. 2020)	24
<i>Promptu Sys. Corp. v. Comcast Corp.</i> , 92 F.4th 1372 (Fed. Cir. 2024)	12
<i>Randall Mfg. v. Rea</i> , 733 F.3d 1355 (Fed. Cir. 2012)	23
<i>Teva Pharm. Indus. Ltd. v. AstraZeneca Pharms. LP</i> , 661 F.3d 1378 (Fed. Cir. 2011)	5
<i>Unwired Planet, LLC v. Google Inc.</i> , 841 F.3d 995 (Fed. Cir. 2016)	15
<i>Voda v. Cordis Corp.</i> , 536 F.3d 1311 (Fed. Cir. 2008)	5
<i>Wasica Finance GmbH v. Continental Automotive Sys., Inc.</i> , 853 F.3d 1272 (Fed. Cir. 2017)	7

APPENDIX LISTING OF EXHIBITS

Exhibit	Description
1001	U.S. Patent No. 11,068,609
1002	Prosecution History of U.S. Patent No. 11,068,609
1003	Declaration of Samrat Bhattacharjee, Ph.D. (“Bhattacharjee”)
1004	Curriculum Vitae of Samrat Bhattacharjee, Ph.D.
1005	U.S. Patent Application Publication 2004/0049687A1 (“Orsini”)
1006	U.S. Patent Application Publication 2003/0200176A1 (“Foster”)
1007	RESERVED
1008	RESERVED
1009	Security First Innovation’s 2023-08-14 Infringement Contentions, Chart B in <i>Security First Innovations, LLC v. Google LLC</i> , 2-23-cv-00097 (E.D. Va.)
1010	Redline Comparison of U.S. Patent Application Publication 2004/0049687A1 to U.S. Patent No. 11,068,609
1011	U.S. Patent No. 5,991,414
1012	U.S. Patent Application Publication No. 2002/0169842
1013	U.S. Patent No. 6,393,568
1014	RESERVED
1015	U.S. Patent Application Publication No. 2006/0053112
1016	U.S. Patent Application Publication No. 2004/0005061
1017	U.S. Patent No. 5,412,722
1018	U.S. Patent No. 7,689,829
1019	U.S. Patent Application Publication No. 2003/0081784
1020	RESERVED
1021	RESERVED
1022	U.S. Patent Application Publication No. 2002/0126850
1023	U.S. Patent No. 5,812,671
1024	RESERVED

Exhibit	Description
1025	RESERVED
1026	RESERVED
1027	U.S. Patent No. 7,251,828
1028	U.S. Patent Application Publication No. 2003/0167314
1029	U.S. Patent No. 6,598,161
1030	U.S. Patent Application Publication No. 2005/0198490
1031	U.S. Patent Application Publication No. 2002/0169808
1032	U.S. Patent No. 6,345,101
1033	U.S. Patent No. 8,707,034
1034	U.S. Patent Application Publication No. 2003/0105719
1035	U.S. Patent No. 4,238,853
1036	U.S. Patent No. 6,819,766
1037	RESERVED
1038	RESERVED
1039	U.S. Patent No. 7,885,405
1040	U.S. Patent Application Publication No. 2005/0210269
1041	U.S. Patent Application Publication No. 2005/0120065
1042	U.S. Patent No. 6,363,481
1043	RESERVED
1044	U.S. Patent Application Publication No. 2003/0191938
1045	U.S. Patent Application Publication No. 2003/0028493
1046	U.S. Patent Application Publication No. 2003/0016820
1047	U.S. Patent Application Publication No. 2002/0042859
1048	U.S. Patent Application Publication No. 2003/0188153
1049	PCT Application Publication No. WO 2004/057461
1050	U.S. Patent No. 7,529,834
1051	U.S. Patent Application Publication No. 2006/0034457

Exhibit	Description
1052	U.S. Patent Application Publication No. 2005/0256972
1053	U.S. Patent Application Publication No. 2004/0049700
1054	U.S. Patent No. 5,675,653
1055	U.S. Patent Application Publication No. 2006/0062384
1056	U.S. Patent No. 6,490,353
1057	U.S. Patent Application Publication No. 2003/0044020
1058	U.S. Patent No. 7,092,524
1059	PCT Application Publication No. WO 2005/024686
1060	U.S. Patent Application Publication No. 2006/0206889
1061	U.S. Patent Application Publication No. 2006/0059347
1062	RESERVED
1063	Rule 16(b) Scheduling Order in Security First Innovations, LLC v. Google, LLC, 2:23-cv-00097-JWK-RJK, Dkt. 54 (July 20, 2023)
1064	Google LLC's Proposed Claim Terms for Construction in Security First Innovations, LLC v. Google, LLC, 2:23-cv-00097-JWK-RJK (Nov. 11, 2023)
1065	Letter from A. Mayron to D. Perlson re Preliminary Proposed Constructions in Security First Innovations, LLC v. Google, LLC, 2:23-cv-00097-JWK-RJK (Nov. 11, 2023)
1066	U.S. Patent Application Publication No. 2004/0228479
1067	RESERVED
1068	Declaration of Nathan Speed ISO PHV Motion
1069	Telephone Hearing, Google v. SFI, IPR2024-00214, IPR2024-00215 (Mar. 21, 2024)
1070	Supplemental Declaration of Samrat Bhattacharjee, PH.D. [SUPPLMENTAL EVIDENCE – NOT FILED]
1071	Security First Innovations, LLC's Responses and Objections to Google LLC's First Set of Interrogatories (REDACTED by SFI) with Attached Charts A-D (Aug. 14, 2023) in Security First Innovations, LLC v.

Exhibit	Description
	Google LLC, 2-23-cv-00097 (E.D. Va.) [SUPPLMENTAL EVIDENCE – NOT FILED] [SUPPLMENTAL EVIDENCE – NOT FILED]
1072	U.S. Patent Application Publication No. 2004/0019900
1073	U.S. Patent Application Publication No. 2002/0083438
1074	U.S. Patent No. 7,076,661
1075	U.S. Patent No. 6,771,657
1076	U.S. Patent No. 6,389,473
1077	U.S. Patent No. 6,662,329
1078	International Application Publication No. WO 2003/026220
1079	Trappe, Key Management and Distribution for Secure Multimedia Multicast (IEEE Dec. 2003)
1080	D. McGovern, "Pushing the internet direct to the user - security issues," IEE Seminar on IP Over Satellite - The next Generation: MPLS, VPN and DRM Delivered Services, 2003., 2003, pp. 122-138, doi: 10.1049/ic:20030063.
1081	U.S. Patent No. 7,277,941
1082	Mike Knee, MPEG Video.
1083	U.S. Patent No. 7,266,613
1084	U.S. Patent Application Publication No. 2004/0202320
1085	U.S. Patent No. 7,823,009
1086	U.S. Patent No. 8,098,819
1087	U.S. Patent Application Publication No. 2004/0236958
1088	International Application Publication No. WO 2001/015162
1089	U.S. Patent Application Publication No. 2004/0107356
1090	U.S. Patent No. 7,739,501
1091	U.S. Patent Application Publication No. 2004/0162871
1092	U.S. Patent Application Publication No. 2002/0129383
1093	U.S. Patent Application Publication No. 2004/0005061
1094	U.S. Patent Application Publication No. 2005/0160292

Exhibit	Description
1095	U.S. Patent No. 6,959,086
1096	RFC 3394: Advanced Encryption Standard (AES) key wrap algorithm
1097	U.S. Patent No. 7,257,227
1098	U.S. Patent No. 6,993,508
1099	Deposition of Aviel D. Rubin, Ph.D. (Dec. 12, 2024) (“Rubin-Depo”)
1100	U.S. Patent Application Publication No. 2003/0140257
1101	U.S. Patent Application Publication No. 2004/0199771
1102	U.S. Patent Application Publication No. 2005/0177740
1103	U.S. Patent Application Publication No. 2004/0199667
1104	U.S. Patent Application Publication No. 2005/0050218
1105	U.S. Patent Application Publication No. 2003/0187799
1106	National Institute of Standards and Technology, NIST Special Publication 800-57, Recommendations for Key Management – Part 1: General (Aug. 2005)
1107	U.S. Patent No. 6,662,329
1108	U.S. Patent No. 6,169,843
1109	MPEG-2 - a tutorial introduction to the systems layer (IEEE 1995)
1110	U.S. Patent Application Publication No. 2003/0145329
1111	Reply Declaration of Samrat Bhattacharjee, Ph.D. (“Bhattacharjee-Reply”)
1112	U.S. Patent Application Publication No. 2004/0010699
1113	European Patent Application Publication No. EP1,564,686
1114	U.S. Patent Application Publication No. 2004/0010699
1115	U.S. Patent Application Publication No. 2004/0187014
1116	U.S. Patent Application Publication No. 2004/0216164 (“Hayhurst”)
1117	U.S. Patent Application Publication No. 2006/0143350

I. GROUND 2: ALL CLAIMS ARE UNPATENTABLE OVER FOSTER

The Board instituted based on anticipation by and obviousness over Foster. Paper 16 (“DI”), 19. Foster’s processor divides a dataset (e.g., movie file) into segments, encrypts the segments with respective keys, encrypts those keys with a key-encryption key (KEK), and stores encrypted segments+keys on storage devices. Foster, [0005], [0040]-[0041], [0064]. SFI’s response argues Foster’s processor does not (and would not) execute code to store *data chunks* in a storage device *coupled to* the processor. Paper 22 (“POR”), 32-55. SFI made similar arguments pre-institution. Paper 10 (“POPR”), 39-45, 54-62. SFI’s arguments still fail.

A. Foster Stores Encrypted Content.

SFI argues Foster is “merely...for encryption and **synchronized transmission**.” POR, 52-53.¹ However, Foster also **stores** its encrypted content. Foster attaches an encryption key to encrypted data to facilitate “synchronization.” Foster, [0002]-[0007]. Synchronization reduces the risk of applying the wrong key to the wrong piece of data and time needed to locate the correct key, thereby facilitating “random access” to portions of “protected content” (e.g., starting a stored movie at any point). Foster, [0064] (“access[ing]...**at any packet**”). EX1111 (“Bhattacharjee-Reply”), ¶¶4-14.

¹ Emphases added; internal quotations, citations, and original alterations omitted.

Foster repeatedly says synchronization improves data “transmission...**or storage.**” Foster, Abstract; *see* [0007]-[0008], [0048], [0064], [0079], claim 11. The Board properly recognized Foster “synchronously **stor[es]** the encrypted content.” DI, 25. SFI admits Foster “**teaches**...recipient 430 can....**store** the encrypted content.” POR, 44.

Foster’s synchronization reduces latency for storage-based purposes, like for Digital Rights Management Systems (“DRMS”). Foster, [0064]; EX1099 (“Rubin-Depo.”), 83-85, 96-99. In Foster’s DRMS, “content is packaged and **loaded** [stored] onto a web server” and subsequently “**downloaded** [stored] **to** [a] personal computer” for on-demand accessing at “rendering time” (i.e., viewing stored content whenever desired). Foster, [0005], [0049] (accessing “content on [the] media server”); FIG. 1 (“stream[ed] **or stored**” content); DI, 25-26; Bhattacharjee-Reply, ¶¶15-17; EX2039, 6-10.

Foster similarly processes content for an MPEG “program stream,” which is intended for use “**in fixed storage.**” Foster, [0073]; EX2014, 1 (“[p]rogram streams are **designed for...storage**”); EX1108, 1:65-2:31; EX1109, 1-9, 13); Bhattacharjee-Reply, ¶¶18-24.

Even Foster’s alternative implementation in conditional access systems (“CASs”) includes more than “live” broadcasting. POSAs knew content service providers used CAS in video-on-demand (VOD) systems, like Foster describes

([0005], [0049]-[0052]). In VOD systems, “content **is stored pre-encrypted** on hard drives in the VOD server” (EX1110, [0008]) and again on “**storage** media in which the **delivered content can be stored**” (EX1072, [0043]). Bhattacharjee-Reply, ¶¶25-35 (citing EX1073, [0001]-[0004] (VOD uses “conditional access”), [0043] (VOD “pre-encrypt[s]” and “stores”), [0098], FIG. 3; EX1097, 2:31-53, FIG. 1; EX1075, 1:51-67; EX1076, 9:53-10:5; EX1077, 1:12-21, 1:48-65, 3:56-5:8, FIG. 1; EX1092, [0079], FIG. 9; EX1116, [0004]-[0006]). Rubin-Depo., 85-94.

Foster stores encrypted data and SFI does not genuinely dispute this.

B. Foster’s Payloads Are *Data Chunks*

Claim 1 requires *generating a plurality of data chunks based on a data set* and encrypting each *data chunk with a respective second key*. SFI acknowledges the Petition “map[ped] the claimed ‘plurality of data chunks’ to Foster’s” “MPEG packet payloads” and “the claimed ‘data set’ to Foster’s [elementary digital] content.” POR, 34-35 (citing Petition, 65-66).

SFI (again) argues “MPEG packet payloads” cannot be *data chunks*. POR, 34-42; POPR, 40-45. The Board should (again – DI, 27-29) reject SFI’s narrow interpretation of *data chunk* because nothing in the intrinsic record supports it.

SFI says the Petition “is conclusory” for not explaining “what ‘data chunks’” are or why Foster’s “packet ‘payload’ would be deemed a chunk.” POR, 37-38; POPR, 40-41. Neither party construed *data chunk* because its meaning is readily

apparent. *See* EX1064-EX1065. *Axonics, Inc. v. Medtronic, Inc.*, 75 F.4th 1374, 1380-81 (Fed. Cir. 2023).

Claim 1’s plain language says a *data set* is the set of data to be securely stored and a *data chunk* is a portion (less than all) of that *data set*. SFI’s expert acknowledges “the share is the thing that something gets split into” (Rubin-Depo., 17) and “share”/“chunk” just “refer[] to the pieces that have been split off from the original data set” (Rubin-Depo., 157-158). Each of Foster’s payload’s is data split off from the *data set* being secured, and thus is a *data chunk*. Petition, 65-66. Bhattacharjee-Reply, ¶¶36-40.

The ’609 patent’s specification places no limit on the relationship between a *data set* and a *data chunk* so long as it is a subset of the *data set*. The specification says, e.g., POSAs “will recognize...a wide number of alternatives for the data splitting process,” including splitting “into any desired” “size unit.” 21:9-25; *see* 15:51-16:3 (similar); 21:27-37 (similar); 74:56-75:10 (similar); Bhattacharjee-Reply, ¶41.

SFI suggests a *data chunk* must be cryptographically split from the *data set* so the splitting itself makes the data less susceptible to “attack.” POR, 39-40. But the specification says “data units may be distributed into...shares” in different ways, including “**fixed**...size”; random/cryptographic splitting is **optional**. EX1001, 21:33-37. Moreover, dependent claims 4-6 limit claim 1’s *data chunks* to being

generated using a “random technique.” That confirms the independent claim is broader and does not require increasing security based on how *data chunks* are generated. *Voda v. Cordis Corp.*, 536 F.3d 1311, 1320 (Fed. Cir. 2008). Bhattacharjee-Reply, ¶¶42-49.

Furthermore, Foster’s payloads “are generated to further secure the data” (POR, 39-40), because Foster encrypts each payload with a different key to “provide **enhanced security**” (Foster, [0004], [0071]). Bhattacharjee-Reply, ¶49.

That Foster calls split portions of its *data set* “payloads” rather than *data chunks* (POR, 37) is immaterial because prior art “does not need” to “us[e] the same words as the patentee.” *Teva Pharm. Indus. Ltd. v. AstraZeneca Pharms. LP*, 661 F.3d 1378, 1384 (Fed. Cir. 2011). SFI does not dispute that Foster’s payloads each comprise less than all of the corresponding content file (*data set*). That is all the claimed *data chunks* require.

It is also irrelevant that some “literature” says a “payload” or “packet” can be “divided into ‘chunks’ or ‘shares.’” POR, 37-38. The intrinsic evidence does not preclude a packet from having one share/*chunk* or being divisible into multiple shares/*chunks*. The ’609 patent only requires that a *data chunk* contain less than its entire *data set*; Foster’s payloads are each less than their corresponding *data set*. Bhattacharjee, ¶¶184-187; Bhattacharjee-Reply, ¶¶50-57.

C. Foster Discloses and Renders Obvious Executing Code To Store a Data Chunk on a Memory Coupled to its Processor

1. Anticipation

Claim 1 recites “*executing code by a processor to perform*” ([1A]) “*storing, in a memory coupled to the processor, at least one data chunk...on at least one storage device*” ([1F]). SFI argues Petitioner “does not attempt to show that Foster’s CPU 406, executes any code **relating to storing content** units on recipient 430 **or instructing** recipient 430 **to store content.**” POR, 43. SFI’s argument fails.

SFI does not dispute that, as Petitioner argued, Foster’s processor “execut[es] ‘a computer program’” to implement Foster’s methods. Petition, 62 (quoting Foster, [0074]-[0075]). SFI’s expert testified that the claimed *code* need only “be code executed on the processor **that results in storing** something on – **in memory that’s coupled to the processor.**” Rubin-Depo, 189-191. *Infra* §I.D (discussing *coupled to* limitation). The Board properly found Foster’s processor does that because Foster’s methods store encrypted content in memories of storage devices (*e.g.*, on servers and at recipients) *coupled to* the processor. DI, 24-25; Petition, 72-76. Thus, Foster’s processor executes code that is “relat[ed] to storing content units on” (POR, 43) those storage devices. Bhattacharjee, ¶¶200-206; Bhattacharjee-Reply, ¶¶58-60.

SFI argues Foster’s processor is “agnostic as to what the recipient does with” content and fails to “instruct the recipient to store or perform some action related to storing.” POR, 43-44. This is a new construction (EX1060) and SFI identifies no

intrinsic evidence requiring the processor to instruct a storage device. *Wasica Finance GmbH v. Continental Automotive Sys., Inc.*, 853 F.3d 1272, 1281-82 (Fed. Cir. 2017). Claim 1 only requires executing code *to perform storing* in memory coupled to the processor. As discussed, Petitioner shows that Foster’s processor does that because it executes code that encrypts content for storing in storage-device memories *coupled to* Foster’s processor. DI, 25-26; Bhattacharjee-Reply, ¶¶61.

But even under SFI’s new, narrowing construction, Foster anticipates, in several ways.

First, Foster’s processor executes code that encrypts and stores data on storage-device memory (RAM) directly *coupled to* it. Bhattacharjee-Reply, ¶¶62-67 (citing EX1117, [0003]-[0005]); Foster, [0075]-[0078], FIG. 16 (annotated below); EX1001, 80:63-81:2 (“RAM” is “local [storage] hardware”); DI, 25 (rejecting SFI’s pre-institution argument – which SFI’s POR does not repeat – that claim 1 requires long-term storage devices).

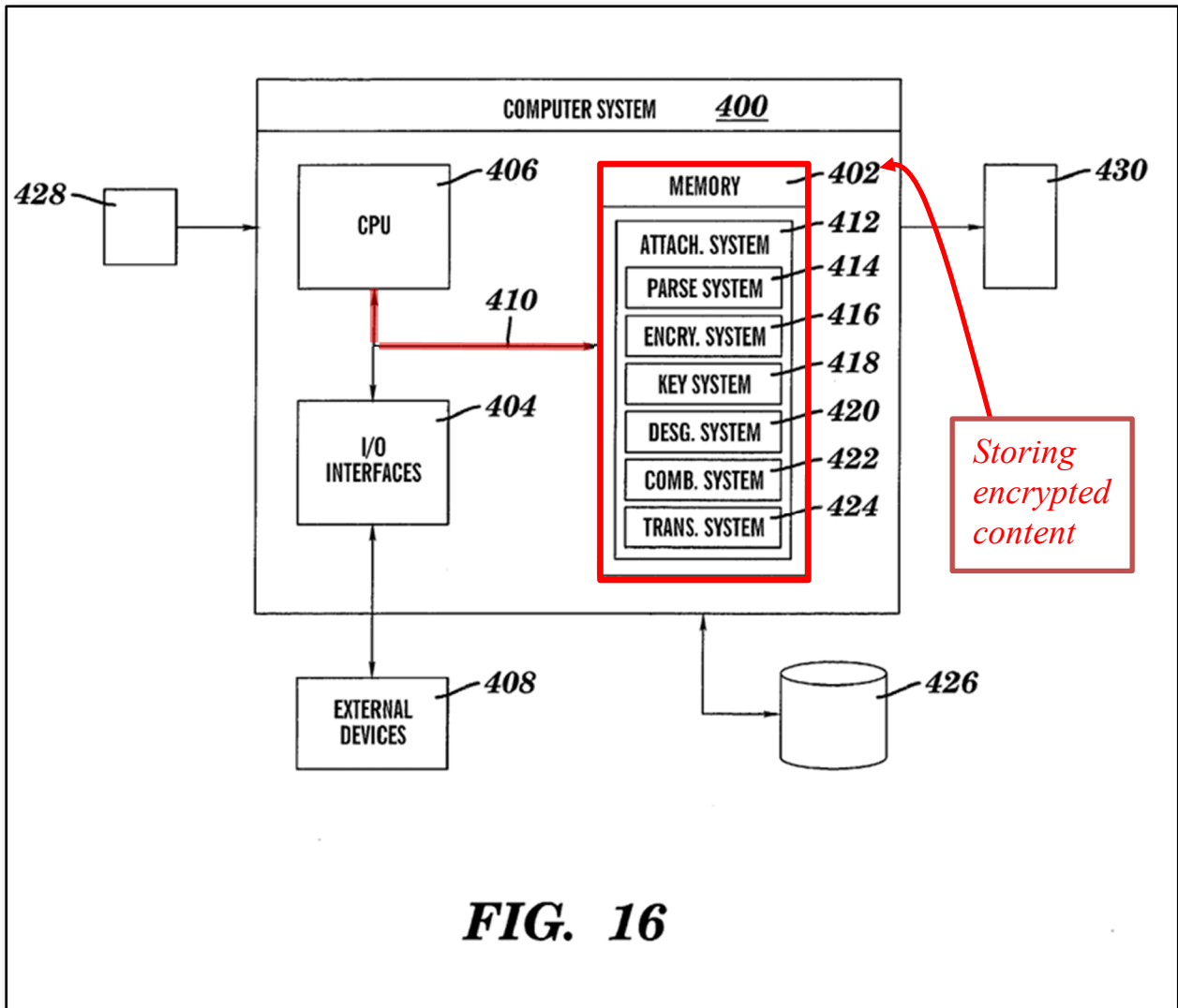
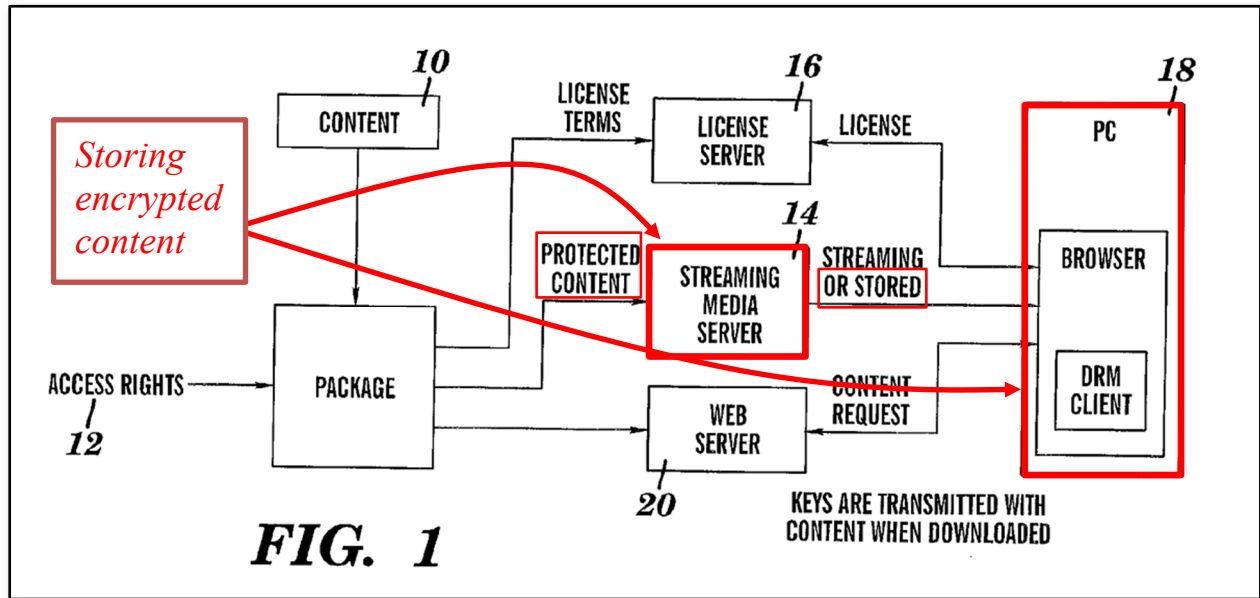


FIG. 16

Then, for Foster’s DRMS (*see supra* §I.A), Foster’s processor executes code to “package[]...and **load**[encrypted data] onto media server 14,” which is later “**downloaded** to PC 18.” Foster, [0049], FIG. 1 (annotated below); DI, 25 (“Foster teaches a PC...**storing**...the data set....”).



Thus, Foster’s processor is also executing code that instructs the media server to load (**store**) encrypted content on the media server’s storage device *coupled to* Foster’s processor, for the purpose of subsequent **storing** to (downloading to) another storage device *coupled to* Foster’s processor (i.e., FIG. 1’s “protected content” is “stored” at Foster’s server **and** PC). Bhattacharjee-Reply, ¶¶68-75 (citing EX1098, 5:44-65); DI, 23-25; Rubin-Depo., 83-89, 96-99 (admitting DRM “encrypt[s]” data “before it’s transmitted to the consumer”).

Additionally, Foster’s processor executes code to encrypt content for one of “two different system options”: “transport stream” (FIG. 14) and “program stream” (FIG. 15). Foster, [0073]. POSAs knew these were two processing options for the MPEG-2 standard. Rubin-Depo., 82-83 (admitting these are “different ways of encoding MPEG data,” but he did not “discuss[] the distinctions”). Foster explicitly says the program stream “is generally used in more reliable transmissions such as those...**in fixed storage.**” Foster, [0073]; EX2014, 1 (“[p]rogram streams are designed for...**local file storage**”). SFI’s expert similarly admits “Foster’s system focuses upon streamed content [] or **content stored on a DVD**” (IPR2024-00215, EX2033, ¶140), which the ’609 patent says is a storage device (EX1001, 66:36-45 (“optical disk”); Paper 3, 56). Bhattacharjee-Reply, ¶¶76-77. Thus, when Foster’s processor executes code to produce encrypted program-stream segments it is **not** “agnostic” to whether that data is for storage (POR, 43-44) because Foster says the program stream is “used in...**fixed storage**” (Foster, [0073]).

And, as discussed above, after encoding and encrypting a program stream, Foster’s processor then executes code to instruct a storage device (e.g., Foster’s “media server”/DVD-drive) to store that Foster-encrypted data. Foster, [0005]; Rubin-Depo., 195-196; Bhattacharjee-Reply, ¶¶78-80.

2. Obviousness

Petitioner also demonstrated POSAs had reason to implement Foster's processor to execute code to store encrypted content in "conventional storage devices coupled to [the] processor," *e.g.*, to beneficially "allow[] Foster's system to store copies of encrypted data," for "backup" and VOD/subsequent "distribution," etc., "without having to repeat the encryption processing" multiple times for one *dataset*. Petition, 77. SFI does not dispute that this implementation meets [1F]'s *storing* limitation.

SFI argues "Petitioner presents no argument as to how (or why) a POSITA would be motivated to modify" Foster's processor "to execute code to store encrypted content units on" storage devices. POR, 47. Not so. The Petition (76-78) identified POSAs' motivation, supported by expert testimony and corroborating evidence. *See also infra* §I.D.2. As to how the modification would work, SFI's patent relies on conventional storage devices and "did not provide the type of detail...[SFI] now argues is necessary in the prior art," which reinforces that POSAs "would have known how to implement the feature[]" in Foster. *In re Epstein*, 32 F.3d 1559, 1568 (Fed. Cir. 1994). The Petition's obvious implementation of Foster relied on conventional storage devices *coupled to* Foster's processor (*e.g.*, via Foster's bus and/or network). Petition, 77-78. Moreover, Foster's processor is already executing code to store data at storage devices *coupled to* it. Foster, [0075]-

[0078] (e.g., “memory 402,” “database 426”), FIG. 14. Furthermore, [1F] is (explicitly) software-based functionality, so “once the function to be performed...has been identified, writing code to achieve that function is within [POSAs’] skill.” *Keynetik, Inc. v. Samsung Elecs. Co., Ltd.*, 2023 WL 2003932, *2 (Fed. Cir. 2023). Bhattacherjee-Reply., ¶¶81-87.

For [1F]’s *code*, SFI further relies on its arguments regarding [1F]’s *coupled to* language (POR, 47), which fail as discussed below (§I.D).

D. Foster Discloses and Renders Obvious “Coupl[ing] to the Processor” Storage-Device Memory

1. Anticipation

SFI argues Foster’s processor is not *coupled to* memory at recipients because “there is no indication that [Foster’s] CPU tells the receiver what to do with th[e received] content,” which “alone [] decides whether to store the information.” POR, 50-51. SFI’s argument fails.

The Federal Circuit has “several times held variants of ‘coupled to’...mean simply ‘connected to.’” *Promptu Sys. Corp. v. Comcast Corp.*, 92 F.4th 1372, 1381 (Fed. Cir. 2024) (citations omitted). SFI’s expert similarly testified that “*coupled to*” means “it’s connected to it.” Rubin-Depo., 190-191, 199 (admitting he “ha[s] not come up with” “an opinion” on what “coupled to the processor require[s]”). SFI identifies no intrinsic evidence that says the processor is only *coupled to* devices it controls/instructs, and SFI ignores Petitioner’s evidence that the patent “uses

‘coupled to’ broadly to include a connection using any communication interface that allows two components to exchange information.” Petition, 42, 75 (citing EX1001, 66:49-55; Bhattacharjee, ¶¶124, ¶204).

As discussed above (§I.C.1), Foster’s processor is executing code to store encrypted content in storage devices within Foster’s system, e.g., local memory, Foster’s media server, DVD drives, **and** recipient storage devices. Those storage devices are *coupled to* Foster’s processor via a bus and (if not local storage) another communication link *coupled to* Foster’s processor. Foster, [0049], [0074]-[0078], FIGs. 1-2, 16; Bhattacharjee, ¶124; Bhattacharjee-Reply, ¶¶88-90; Rubin-Depo., 190-191 (admitting “content gets from Foster’s processor to the memory and [sic: in] the receiver”).

SFI admits “Foster’s CPU [*processor*]...encrypt[s] information...for distribution to receivers,” by “sending content units from Foster’s system to the recipient[s],” which “store the transmitted information.” POR, 50-51. That information is stored in memories (at a server and end-users) via their connection to Foster’s processor. Rubin-Depo., 190-194. Thus, Foster’s processor is *coupled to* those memories. Petition, 75. Bhattacharjee-Reply, ¶¶91-93.

SFI says Petitioner argued Foster’s coupling is only via “bus 410.” POR, 51. Not so. Petitioner argued coupling to Foster’s processor includes “**via ‘transmission system 424’ and ‘bus 410.’**” Petition, 75. SFI ignores transmission

system 424, which couples Foster’s processor to storage devices outside Foster’s computer; bus 410 is a conventional component for moving data **within** Foster’s computer, e.g., from the processor to local memory and the transmission system (which *couples to* storage devices like the media server, end-user storage) or I/O interfaces (which *couples to* additional, local *storage devices*). Bus 410 is only **part** of the *coupling* between Foster’s processor and storage-devices. *See* Foster, FIG. 1 (coupling the unit encrypting content to various units storing content). Bhattacharjee, ¶¶204-206; Bhattacharjee-Reply, ¶¶94-101.

2. Obviousness

Petitioner demonstrated POSAs also had reason to store Foster’s encrypted content in “storage devices coupled to Foster’s processor” for reasons of “backup,” “video-on-demand,” and/or “distribution to later-identified recipients.” Petition, 77-78 (citing corroboration); Rubin-Depo., 195-196 (“a hard disk” attached to a computer is “coupled to” its processor). SFI does not dispute those were known reasons to store encrypted content, but argues they would not “solve any problem[s]” for Foster or content owners because the Petition-identified “benefits are” allegedly already “provided by the content owner.” POR, 52-54. SFI arguments fail.

SFI only supports those arguments with uncorroborated expert testimony, which should be accorded little-to-no weight. POR, 52; *In re Acad. of Sci. Tech Ctr.*, 367 F.3d 1359, 1368 (Fed. Cir. 2004). Moreover, Petitioner did “not need to show

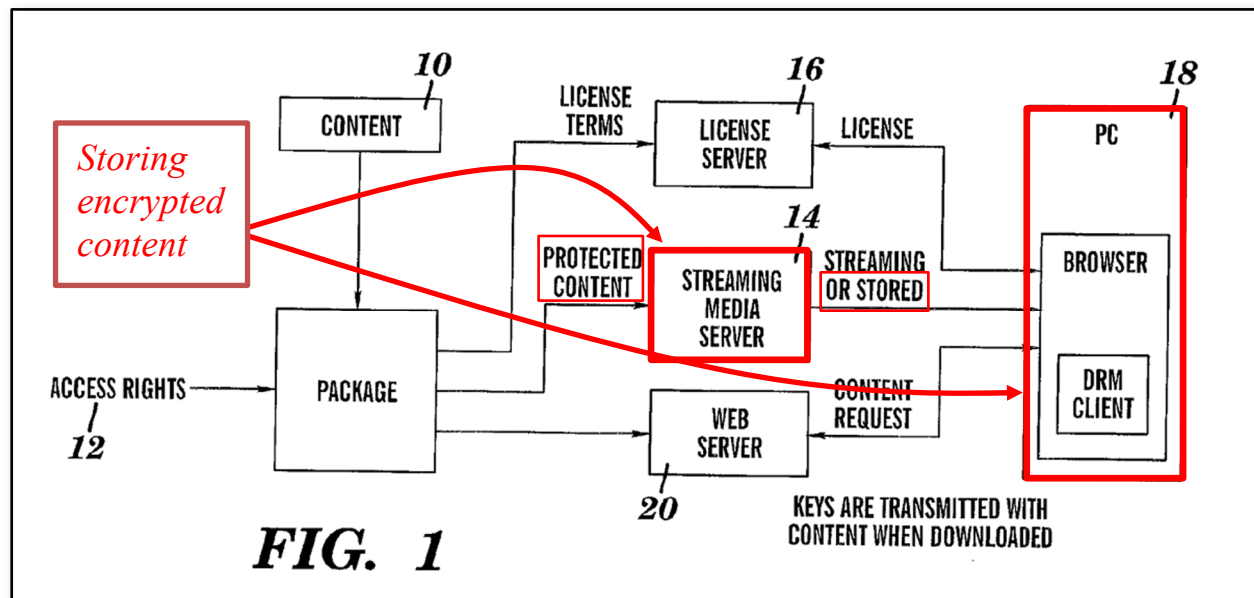
that there was a known problem with the prior art system in order to” demonstrate obviousness. *Unwired Planet, LLC v. Google Inc.*, 841 F.3d 995, 1003 (Fed. Cir. 2016).

Hypothetical content-owner storage would not negate benefits of content-service providers (“CSPs”) storing Foster-encrypted content. POSAs knew CSPs (e.g., cable-TV companies) often purchased/licensed content from content owners and then securely stored it before delivering to recipients. Rubin-Depo., 84-94. CSPs typically and beneficially encrypted and stored that content for repeated use (e.g., downloads by different customers) – rather than repeatedly encrypting it on-the-fly – and for backup purposes. Petition, 77-78 (citing EX1045, EX1050, EX1059-EX1060; Bhattacharjee, ¶¶206-213); Bhattacharjee-Reply, ¶¶102-114 (citing EX1073, [0043] (VOD “pre-encrypt[s]” and “stores the content”); EX1097, 2:31-53 (“real-time encryption [for VOD] poses much greater cost and space issues”); EX1075, 1:51-67; EX1105, [0034]-[0037]; EX1080, 2-7).

As discussed above (§I.A, §I.C.1), Foster encrypts content using an MPEG program stream, which Foster says “is generally used...**in fixed storage.**” Foster, [0073]; EX2014, 1 (“[p]rogram streams are designed for...**local file storage**”); EX1109, 1-6 (“programme stream is **intended for...storage**”); EX1001, 66:36-67:11; Rubin-Depo., 82-85. For example, as discussed above (§I.A, §I.C.1), Foster’s

encrypted content is “loaded onto media server 14” then “download[ed] to PC 18.”

Foster, [0049], FIG. 1 (annotated below).



Thus, Foster itself is (at least) suggesting/motivating having its processor execute code for storing on storage devices *coupled to* Foster’s processor, to support content-on-demand (like VOD) and/or backup, as the Petition identified. Petition, 76-77. Bhattacharjee, ¶209 (POSAs would have also encrypted “content stored by the content provider”); EX1059, 2:15-24 (prior art included “on-demand streaming download of content from the content provider”) (cited Petition, 77); Bhattacharjee-Reply, ¶¶115-119; Foster, [0037]-[0041]).

SFI argues Petitioner’s corroborating evidence does not “teach[] storage of content by an intermediary.” POR, 54. Not so. EX1060, e.g., describes prior-art VOD systems where CSP intermediary “servers...store all movies.” EX1060,

[0007]-[0008], FIG. 6 (cited Petition, 77), [0005]-[0016], [0176]-[0187] (service provider's "server" stores content "transported" from the "source provider" (e.g., "movie publisher")); EX1073, [0043]; EX1097, 2:31-53; EX1101, FIG. 1, [0003]-[0005], [0020]-[0027], [0043]-[0045]; EX1102, [0025]-[0032], FIGs. 1-2. Bhattacharjee-Reply, ¶¶120-130.

SFI's expert admitted prior-art conditional-access/DRM systems included intermediaries storing content-owner content. Rubin-Depo, 86-93 (testifying, e.g., content owners used intermediaries to "focus on what they were good at" (i.e., making rather than storing/distributing content)).

SFI argues "Petitioner's proposed modifications would" "compromise" Foster's security "or require significant resources to prevent" compromising security because Foster could be "hacked," whereas content owners "already securely store[]" their content. POR, 55. SFI argues modifying Foster would require "significantly more storage," which "would have to be as secure as the storage at the content provider/owner." *Id.*

Beyond a conclusory expert declaration, SFI offers no evidence that hacking risks or overall storage requirements would increase by storing content at a CSP/intermediary rather than a content owner. Nor does SFI allege, let alone support with evidence, that CSP/intermediary storage were less secure than content-owner storage. And even if it required "increased cost and complexity" to make

CSP storage more secure, that does “not show that [the technique] was unknown or could not have practical applications,” and thus does not support non-obviousness. *In re Cree*, 818 F.3d 694, 700-701 (Fed. Cir. 2016). SFI is also wrong because content owners often had intermediaries storing their content, demonstrating that SFI’s criticisms are illusory. EX1103, [0175]; EX1104, [0002]-[0004]; EX1059, 5:4-23; Rubin-Depo, 86-93; Bhattacharjee-Reply, ¶¶131-138.

SFI’s (unsupported) allegation that content owners securely stored content also ignores that Foster discusses content owners supplying **non**-encrypted content to CSPs. Foster, [0079]. Thus, Foster’s encryption **adds** security, and so Foster’s processor executing code to store Foster-encrypted content would **not** “make Foster’s system worse” (POR, 55-56). But even if there were tradeoffs, storing encrypted content on devices *coupled to* Foster’s processor would have provided other Petition-identified **benefits** (as discussed above) and “it is a commonplace fact that design decisions entail making tradeoffs among multiple objectives.” *Corephotonics, Ltd. v. Apple Inc.*, 2021 WL 4944471, at *6 (Fed. Cir. Oct. 25, 2021). Bhattacharjee-Reply, ¶¶139-140.

II. GROUND 1: ALL CLAIMS ARE UNPATENTABLE OVER ORSINI

For Ground 1, SFI only disputes limitation [1E]: “*a cryptographic operation based on the first key to further secure...data chunks.*”

Petitioner advanced several arguments demonstrating how claim 1 is unpatentable over Orsini. Petition, 30-42. The Board preliminarily found those arguments unpersuasive. DI, 35-51.

While Petitioner believes those arguments have merit, the focus below is on one obviousness argument to simplify the trial. Specifically, Petitioner's argument that POSAs would have encrypted Orsini's share keys with a key-encryption-key ("KEK") to safeguard Orsini's share keys from detection, a technique multiple corroborating references demonstrated was known and desirable in the art. Petition, 40-42 (citing corroboration).

SFI does not dispute that encrypting keys with KEKs was known or that encrypting Orsini's share keys with a KEK meets [1E] under SFI's constructions.

POSAs had motivation to use the known KEK-technique in Orsini because, as even SFI's expert admits, POSAs would "not put [] security in anything other than the keys," but the way Orsini allegedly protects its share keys – by obfuscating which *data chunk* each share key is stored with – "is not putting the protection in the keys, so...this **does not on its own provide security.**" Rubin-Depo., 155. That candid admission is consistent with the Petition's evidence demonstrating that POSAs knew "[s]ecurity' comes from 'encrypt[ing] the encryption keys themselves' with 'a key encryption key.'" Petition, 40 (quoting EX1036). On the fully developed record, claim 1 is plainly obvious over Orsini.

A. POSAs Would Have Encrypted Orsini's Share Keys Using a Conventional KEK

The parties agree Orsini's Figure 21 results in an encrypted share (the claimed *data chunk*) being stored with a plaintext (unencrypted) share-encryption key. Petition, 40-41; POR, 2, 20-22. Each plaintext key is stored with a share, but not the share it unlocks, which even SFI recognizes means an attacker can still "reconstruct [Orsini's] encrypted dataset if the attacker has access to each of the shares" (because the share keys are unencrypted). POR, 4; EX2033, ¶41.

Petitioner demonstrated, supported by expert testimony **and corroborating evidence**, that POSAs would have encrypted Orsini's share keys with a KEK to prevent reconstructing the dataset even when an attacker possessed every encrypted share. Petition, 40-41. Petitioner's corroborating evidence was direct in its teaching: "cryptographic keys should **never** occur in clear [*i.e.*, unencrypted] form outside the cryptographic device." EX1035, 10:42-62 (cited Petition, 40).

Petitioner's prior art explains POSAs knew that when an encryption key is transmitted or stored, the system should "encrypt the encryption keys themselves...[with] a key encryption key." EX1036, 1:66-2:6 (cited Petition, 43). That was true whether "encryption keys are included with the encrypted message" (*id.*) or stored separately from and/or obfuscated with encrypted data-portions (EX1035, 10:42-62; EX1036, 1:66-2:6; EX1086, 7:3-22; EX1087, [0017]-[0025]). Petition, 40-41. Bhattacharjee-Reply, ¶¶141-156 (citing EX1112, [0022]-[0027]).

POSAs, therefore, would have recognized that Orsini's obfuscation provides (negligible) non-cryptographic security that depends almost exclusively on the attacker not possessing the encrypted shares. Conversely, POSAs would have recognized that encrypting Orsini's keys with a KEK provided cryptographic (stronger) security that continued **even when** all encrypted shares were obtained (unlike Orsini's obfuscation). Bhattacharjee, ¶¶120-123; Bhattacharjee-Reply, ¶¶157-172; EX1100, [0010]. Rubin-Depo., 137-139, 145-147, 150-156.

Thus, POSAs knew leaving Orsini's share keys unencrypted was undesirable because, even with Orsini's obfuscation, the keys (and thus the shares they unlocked) remained exposed and subject to an attack that KEKs were known and designed to prevent.² EX1003, ¶132; Bhattacharjee-Reply, ¶¶173-180 (citing EX1106).

Therefore, encrypting Orsini's share keys with a KEK was obvious because it would have been using POSAs' ordinary skill to perform a known technique (encrypting encryption keys) to solve a known problem (protecting encryption keys from discovery), with a reasonable expectation of success and no unexpected results.

² The Board, when instituting IPR against the related '116 patent, recognized that POSAs knew KEKs beneficially "provided even further security" in systems (like Orsini's) that store encrypted data portions with unencrypted keys of other data portions. IPR2024-00212, Paper 11, 17

Intel Corp. v. PACT XPP Schweiz AG, 61 F.4th 1373, 1379-81 (Fed. Cir. 2023) (obvious to use known technique even if reference “already address[ed] the same problem” in another way). Petitioner’s motivation-to-combine theory was well-reasoned and well-supported.

1. The Board Misunderstood the Petitioner’s Argument

At institution, the Board said the “reasoning” behind Petitioner’s KEK-based obviousness argument was the same reasoning behind “Petitioner’s argument about [Orsini’s] public-key encryption key.” DI, 50-51. The Board found “Petitioner’s argument about the public-key encryption key” combined distinct embodiments. *Compare* DI, 44-45 (“Petitioner combines the features from multiple embodiments to arrive at the claimed invention.”) *with* DI, 50-51 (similar critique).

However, the Petition here did not argue Orsini’s public-key disclosures anticipate (or render obvious) any claims. The Petition noted Orsini public-key encrypts the session key that encrypts the entire *data set*, not Orsini’s share keys. Petition, 40. That demonstrates Orsini was already capable of using KEKs. However, the obviousness of using a conventional KEK to encrypt Orsini’s share keys advanced by Petitioner is premised on POSAs’ corroborated knowledge that encryption keys—like Orsini’s share keys—“**should never**” (EX1035, 12:15-23) be electronically stored without encrypting them. *See* Petition, 40-41. Bhattacharjee-Reply, ¶¶181-183.

In preliminarily rejecting that obviousness argument based on reasoning Petitioner did not offer, the Board did not mention or discuss Petitioner’s evidence of POSAs’ background knowledge or Dr. Bhattacharjee’s corroborating testimony concerning KEK-based security. DI, 40-41. Respectfully, the Board must consider the evidence supporting obviousness. *Randall Mfg. v. Rea*, 733 F.3d 1355, 1362 (Fed. Cir. 2012).

2. Petitioner Does Not Rely on Common Sense To Supply a Missing Limitation.

SFI argues Petitioner’s KEK-based obviousness argument relies on common sense to supply a missing limitation. POR, 5-7. This is incorrect.

[1E] is not missing in the prior art. As discussed (*supra* §II.A-§II.A.1), the Petition cited EX1035’s **and** EX1036’s disclosures of the known desirability of using KEKs to encrypt encryption keys for storage. Petition, 40. As the Board found, Foster also discloses [1E]. DI, 30. Other prior art similarly teach using a KEK to encrypt keys, including keys that (like Orsini) encrypt “partition[s]” of data. EX1088, 2:20-3:16; EX1080, 2-7; Bhattacharjee-Reply, ¶¶184-193.

Moreover, Petitioner does not rely on only common sense. Petitioner provides reasoned analysis — supported by multiple pieces of corroborating evidence and detailed expert testimony — detailing why POSAs had reason to further secure Orsini’s share keys by using the well-known (not SFI-invented) technique of encrypting encryption keys with KEKs. Bhattacharjee-Reply, ¶¶194-197.

The Federal Circuit has explained that *Arendi* and *DSS* (cited POR, 18) do not preclude obviousness based on different teachings in the art to meet different elements of a claim limitation. *Fleming v. Cirrus Design Corp.*, 28 F.4th 1214, 1222 (Fed. Cir. 2022) (“That the proposed *combination*...—rather than one of the individual references—discloses the disputed claim limitations does not defeat...obviousness.”) (emphasis original); *Koninklijke Philips N.V. v. Google LLC*, 948 F.3d 1330, 1339 (Fed. Cir. 2020) (obviousness is not based on “common sense” alone when supported by “corroborated” “expert evidence,” establishing that the limitation “was not only in the prior art, but also within the general knowledge of” POSAs).

3. SFI’s Challenges to Petitioner’s Motivation-to-Combine Theory Fail.

SFI offers three challenges to Petitioner’s KEK-based argument: (1) Orsini already provides sufficient security, (2) KEK-based encryption introduces complications, and (3) the modification is hindsight driven. Each argument fails.

a. POSAs Would Have Wanted to Increase Orsini’s Security.

SFI argues POSAs would not further secure Orsini’s system because Orsini itself was satisfied with its security. POR, 21-24. SFI’s argument misunderstands how POSAs would read Orsini and the law of obviousness.

Orsini's use of **non**-cryptographic obfuscation—storing a share's key with a different share—confirms that even Orsini recognized a need to add security by making it harder to determine the correct share key. Bhattacharjee-Reply, ¶¶198-203. But POSAs knew that Orsini's non-cryptographic obfuscation technique was **less secure** than **also** (or alternatively) encrypting Orsini's share keys using a conventional KEK because, unlike Orsini's obfuscation, a KEK would **cryptographically** protect share keys, even when all shares were collected. Bhattacharjee-Reply, ¶¶204-205; EX1100, [0010]; EX1089, [0569]; EX1090, 2:39-41 (“The more difficult it is for an unauthorized entity to obtain a valid key, the more secure the cryptographic scheme.”). SFI does not – because it cannot – dispute that encrypting Orsini's share keys would make Orsini more secure.

Orsini's Figure 21 divides data into four shares/chunks and stores each with a share encryption key. Orsini, [0298]-[0301]. That allows (at most) twenty-four different share-key pairings. POSAs understood that computers could easily overcome Orsini's obfuscation by trying all twenty-four possibilities. Bhattacharjee-Reply, ¶¶206-208; Rubin-Depo., 22-25, 154-156; EX2007, 11 (even “26 possible keys” is subject to “a *brute-force attack*,” but “any secure encryption **must not be vulnerable to such brute-force attack**”) (italics original).

SFI's expert admitted that describing Orsini's obfuscation "is a pretty simple math problem" because "it wouldn't take long" to write out the combinations by hand. Rubin-Depo., 22-25, 154-156.

Orsini's obfuscation also disregards industry-wide best practices. The National Institute of Standards and Technology ("NIST") advised POSAs that when, as in Orsini, a "key exists external to the cryptomodule, **confidentiality protection shall be provided either by encryption (e.g., key wrapping) [i.e., a KEK]**" or physically protecting keys (like using a safe, which would be inconsistent with Orsini's **electronic** storage of keys). EX1106, 34, 61, 123-125. Bhattacharjee-Reply, ¶¶171-176.

Thus, SFI's assertion that Orsini's non-cryptographic obfuscation "achieves the benefit...served by" a KEK's cryptographic protection (POR, 22) is incorrect. Bhattacharjee-Reply, ¶¶209-210; Rubin-Depo., 22-25, 154-156.

Moreover, even if SFI were correct that Orsini said obfuscation was "sufficient" for some data (POR, 24), that would not make using the known KEK technique non-obvious in Orsini's system for all *data* and/or risk levels. Rubin-Depo., 113-115. *Cole Kepro Int'l, LLC v. VSR Indus., Inc.*, 2017 WL 2627982, *3 (Fed. Cir. 2017) (that a prior art reference discloses a certain level of a solution to a problem "does not discourage," by that fact alone, seeking a greater level of solution to the same problem). Bhattacharjee-Reply, ¶¶211-212. Rubin-Depo., 145-147 ("in

security you want to provide defense in depth, meaning **the more layers of security you have, the better**"; opining that SFI's related '854 patent "talks about" that at 65:61, which is Orsini, [0423]).

b. The Alleged Drawbacks to KEKs Were Easily Overcome and Are Legally Irrelevant.

SFI argues "drawbacks associated with managing and securing" KEKs would have dissuaded POSAs from using KEKs in Orsini. POR, 25-32. But SFI does not allege that addressing these alleged "drawbacks" was beyond POSAs' skill.

Orsini **already** taught techniques for managing and securing other KEKs, or (at least) could have been modified in a simple, straightforward way to use conventional techniques for managing KEKs. For example, Orsini supports "Triple DES and AES" (Orsini, [0378]), which taught using and managing KEKs. *Infra* §II.B; Bhattacharjee-Reply, ¶¶213-215. Orsini also supports "a wide number of alternative key management systems." Orsini, [0107].

Petition-cited prior art demonstrates that using/managing KEKs was conventional. Petition, 43 (citing EX1035-EX1036). EX1036, e.g., discloses a "system for **managing keys**," including KEKs (2:29-61) and says the "prior art" included "security infrastructure to solve the key management problem[]" (2:7-22). *See also* EX2032, 683-84 ("key management" for "KEKs"); EX1017, 1:50-1:68 (similar); EX1093, [0013], [0035]-[0039] (similar), [0057]-[0060]. As discussed above (§II.A.3.b), NIST urged POSAs to use KEKs and discussed related key-

management issues. EX1106, 34, 61, 123-125. The evidence is clear that any drawbacks in managing KEKs were (at most) a conventional tradeoff and not beyond POSAs' skill, and thus do not defeat obviousness. *Intel*, 61 F.4th at 1381; Rubin-Depo., 185-186. Bhattacharjee-Reply, ¶¶216-221.

Indeed, SFI does not dispute that using the known KEK technique in Orsini's system would have increased security because Orsini stores its share keys outside of Orsini's encryption device in unencrypted form, which prior art taught "should **never** occur" for security reasons. EX1035, 12:15-23.

POSAs also knew that securing data often involves tradeoffs in "convenience and security, speed," etc. and knew how to address those tradeoffs. EX1081, 15:6-25; Bhattacharjee-Reply, ¶¶222-223.

Thus, even if using KEKs to encrypt Orsini's share keys added complexity, that does not defeat obviousness because it "does not show that [using KEKs] was unknown or could not have practical applications." *In re Cree, Inc.*, 818 F.3d 694, 700-01 (Fed. Cir. 2016); *Corephotonics*, 2021 WL 4944471, at *6.

Encrypting Orsini's share keys was not beyond POSAs' "capability" and was nothing more than using a "known technique [KEKs] to address a known problem [the risk of storing unencrypted encryption keys] using prior art elements according to their established function" and thus was obvious. *CyWee Grp. Ltd. v. ZTE (USA), Inc.*, 90 F.4th 1358, 1366 (Fed. Cir. 2024) (cleaned up).

c. The Identified Motivation To Use a KEK in Orsini Is Grounded in Evidence, Not Hindsight.

SFI contends Petitioner’s KEK-based argument is hindsight. POR, 29-32. This is plainly incorrect.

Petitioner’s combination of Orsini with a KEK is based on a specific motivation—supported by multiple pieces of **corroborating evidence** and Petitioner’s expert—to use a KEK for its known purpose and benefit of securing encryption keys being transmitted/stored outside the cryptographic device, rather than leave them unencrypted, as Orsini does. Petition, 40 (citing **EX1035**, 12:15-23, 10:42-62; **EX1036**, 1:66-2:6; Bhattacharjee, ¶132). That motivation is grounded in evidence and reasoning, not hindsight. Again, Orsini **itself** recognized a need to protect its keys, but elected non-cryptographic obfuscation, rather a conventional KEK technique. Bhattacharjee-Reply, ¶¶224-225.

B. SFI’s Reasonable Expectation of Success Challenge Fails.

SFI asserts POSAs had no “reasonable expectation of success in modifying Orsini to achieve claim 1.” POR, 4. But SFI never argues any specific reason **why** POSAs allegedly would not have reasonably expected success in using a conventional KEK to encrypt encryption keys (like Orsini’s) as was known, let alone

submit evidence to substantiate any such unexplained reason. Bhattacharjee-Reply, ¶¶226-227; Petition, 41-42 (citing Bhattacharjee, ¶123).³

As discussed above (§II.A.3.b), using KEKs and related key-management systems were known. Orsini itself encrypts shares using “Triple DES and AES” standardized techniques (Orsini, [0378]), which taught securing their keys by encrypting them with KEKs, and relied on existing key-management concepts. Bhattacharjee-Reply, ¶¶228-239; EX1017, 1:50-68; EX1094, [0056]-[0057] (“Triple-DES Key Wrap”); EX1090, 19:47-61, 21:1-22:54 (“AES[] key wrapping”); EX1095, 1:32-47 (“key managements schemes are built on **commonly used**...concepts, such as key wrapping”), 3:10-15; EX1096, 1-3.

Implementing a KEK in Orsini was within POSAs’ capabilities.

Dated: 12-23-2024

Respectfully submitted,

/Gregory Nieberg/

Gregory Nieberg, Reg. #57,063

³ *Google LLC v. Nobots LLC*, IPR2022-00940, Paper 38, 38 (Nov. 23, 2023)

(“present[ing] additional evidence” “directed specifically to arguments” in POR challenging POSAs’ expectation of success is proper).

CERTIFICATE OF WORD COUNT

Pursuant to 37 C.F.R. § 42.24, the undersigned certifies that the foregoing **PETITIONER'S REPLY** contains 5,591 words excluding; a table of contents, a table of authorities, 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: December 23, 2024

/Dara Del Rosario/

Dara Del Rosario

Paralegal

WOLF, GREENFIELD & SACKS, P.C.

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

I certify that on December 23, 2024, I will cause a copy of the foregoing document, including any exhibits filed therewith, to be served via electronic mail, as previously consented to by Patent Owner, upon the following:

Stephen J. Elliott

elliotts@sullcrom.com

Kenneth Weatherwax

Weatherwax@lowensteinweatherwax.com

Nathan Lowenstein

Lowenstein@lowensteinweatherwax.com

Colette Woo

Woo@lowensteinweatherwax.com

Dustin F. Guzior

guziord@sullcrom.com

Andrei Iancu

iancua@sullcrom.com

SFI_PTAB@sullcrom.com

Date: December 23, 2024

/Dara Del Rosario/

Dara Del Rosario

Paralegal

WOLF, GREENFIELD & SACKS, P.C.