

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

INTERNATIONAL BUSINESS MACHINES CORPORATION,
Petitioner,

v.

SECURITY FIRST INNOVATION, LLC,
Patent Owner.

Case IPR2025-01200
Patent 8,271,802

PATENT OWNER'S PRELIMINARY RESPONSE

TABLE OF CONTENTS

	Page
I. INTRODUCTION	1
II. BACKGROUND.....	4
A. Overview Of The '802 Patent And The Challenged Claims	4
B. Overview Of Torre	7
C. Overview Of Tajima.....	10
D. Overview Of Orsini	12
III. PETITIONER'S TORRE-TAJIMA GROUNDS FAIL (ALL CLAIMS, GROUNDS I-IV).	15
A. Petitioner Fails To Demonstrate That Its Proposed Torre/Tajima Grounds Teach "Generating Data Splitting Information" That Is "Usable To Determine Into Which Of A Plurality Of Shares Of Data A Unit Of Data ... Will Be Placed" (All Claims, Grounds I-IV).....	15
1. Petitioner Fails To Demonstrate That "Torre/Tajima System-1" Discloses "Generating Data Splitting Information."	16
a. Torre Does Not Inherently Disclose "Generating Data Splitting Information."	17
b. Petitioner's Obviousness Theory Also Fails.	20
2. Petitioner Fails To Demonstrate That "Torre/Tajima System-2" Teaches That The Alleged "Data Splitting Information" Is "Usable To Determine Into Which Of A Plurality Of Shares Of Data A Unit Of Data ... Will Be Placed."	23

a.	Tajima’s Calculated Arbitrary Length Determines Size, Not Placement.....	24
b.	Tajima’s Appended Information Retrospectively Describes How Data Was Placed, Not How It Will Be Placed As Required By The Claims.	26
3.	Petitioner’s Invocation Of The Copy-N Redundancy Transformation Does Not Meet This Limitation.	27
B.	Petitioner Fails To Demonstrate That A POSITA Would Be Motivated To Combine Torre And Tajima To Achieve “Including In The Plurality Of Shares Data Indicative Of [] The Encryption Key” (All Claims, Grounds I-IV).....	28
1.	Petitioner Incorrectly Assumes That Torre Itself Already Teaches Including The Encryption Key In Its Tags.	30
2.	Petitioner Fails To Show Torre Would Benefit From Attaching The Encryption Key To Its Tags.....	32
3.	Tajima’s Key-Embedding Approach Is Incompatible With Torre; Torre Discourages Adding The Key To The Tags, Which Entails Significant, Unaccounted-For Drawbacks.	33
IV.	PETITIONER’S ORSINI/TORRE GROUNDS FAIL (ALL CLAIMS, GROUNDS V-VII).	39
A.	The Claims Of The ’802 Patent Are Supported By New Disclosures That Are Not In Orsini (All Claims, Grounds V-VII).....	39
B.	Petitioner Fails To Demonstrate That Orsini/Torre Discloses Or Renders Obvious Restoring The Data Set From “Less Than All” Of The Shares (All Claims, Grounds V-VII).	43
1.	Contrary To Petitioner’s Contention, Orsini Makes Clear That Figure 21 Requires Retrieval Of All Four Shares To Reassemble The Data Set.....	43

C.	Petitioner Fails To Demonstrate That Orsini Discloses Or Renders Obvious “Generating Data Splitting Information” (All Claims, Grounds V-VII).	47
1.	Petitioner Fails To Demonstrate That Orsini Alone Discloses the Separate Steps Of “Encrypting The Data Set Based On An Encryption Key” And “Generating Data Splitting Information.”	48
2.	Petitioner Fails To Demonstrate That Orsini Renders Obvious “Generating Data Splitting Information.”	51
D.	Petitioner Fails To Demonstrate That A POSITA Would Be Motivated To Combine Orsini And Torre To Add “Creating Integrity Information Using The Data Set” And “Including In The Plurality Of Shares Data Indicative Of ... The Integrity Information” (All Claims, Grounds V-VII).	54
1.	Orsini Already Provides A Complete, Self-Verifying System.	56
2.	Orsini’s Figure 21 Embodiment Does Not Have The Integrity Problems Purportedly Addressed By Torre.	57
3.	The Proposed Combination Would Offer No Benefit And Impose Significant Cost.	58
4.	Petitioner Fails To Explain How Torre’s Asymmetric Encryption Keys Are Created In Orsini/Torre.	60
5.	Petitioner’s Theory Is Driven By Hindsight.	62
V.	CONCLUSION	63

TABLE OF AUTHORITIES

	Page(s)
CASES	
<i>Adasa v. Avery Dennison Corp</i> , 55 F.4th 900 (Fed. Cir. 2022)	48
<i>Apple Inc. v. Cellular Commc'ns Equipment, LLC</i> , IPR2015-00576, Paper 7 (June 12, 2015).....	55
<i>Arctic Cat Inc. v. Bombardier Recreational Prods.</i> , 876 F.3d 1350 (Fed. Cir. 2017)	36, 58
<i>Arctic Cat Inc. v. Polaris Indus.</i> , 795 Fed. App'x. 827 (Fed. Cir. 2019)	33, 58
<i>Arendi S.A.R.L. v. Apple, Inc.</i> , 832 F.3d 1355 (Fed. Cir. 2016)	21
<i>CRFD Research, Inc. v. Matal</i> , 876 F.3d 1339 (Fed. Cir. 2017)	47
<i>DSS Tech. Mgmt. v. Apple Inc.</i> , 885 F.3d 1367 (Fed. Cir. 2018)	21
<i>F5 Networks, Inc. v. Radware, Ltd.</i> , IPR2017-00124, Paper 48 (Apr. 23, 2018).....	55
<i>Google LLC v. Security First Innovations, LLC</i> , IPR2024-00212, Paper 42 (May 16, 2025).....	55
<i>Henny Penny Corp. v. Frymaster LLC</i> , 938 F.3d 1324 (Fed. Cir. 2019)	36, 58
<i>In re Kahn</i> , 441 F.3d 977 (Fed. Cir. 2006)	51
<i>In re Magnum Oil Tools Int'l, Ltd.</i> , 829 F.3d 1364 (Fed. Cir. 2016)	51

<i>In re NTP, Inc.</i> , 654 F.3d 1279 (Fed. Cir. 2011)	55
<i>Interconnect Planning Corp. v. Feil</i> , 774 F.2d 1132 (Fed. Cir. 1985)	37
<i>InTouch Techs., Inc. v. VGo Commc'ns., Inc.</i> , 751 F.3d 1327 (Fed. Cir. 2014)	38, 61
<i>KSR Int'l Co. v. Teleflex Inc.</i> , 550 U.S. 398 (2007).....	51, 53
<i>Otsuka Pharm Co. v. Sandoz, Inc.</i> , 678 F.3d 1280 (Fed. Cir. 2012)	61
<i>Ruiz v. A.B. Chance Co.</i> , 357 F.3d 1270 (Fed. Cir. 2004)	37, 53, 61
<i>TQ Delta, LLC v. Cisco Sys., Inc.</i> , 942 F.3d 1352 (Fed. Cir. 2019)	18
<i>Univ. of Strathclyde v. Clear-Vu Lighting LLC</i> , 17 F.4th 155 (Fed. Cir. 2021)	37, 53, 61
<i>Xerox Corp. v. Bytemark</i> , IPR2022-00624 Paper 12 (Aug. 25, 2022) (precedential).....	18

STATUTES

37 C.F.R. § 42.104(b)(4).....	24
37 C.F.R. § 42.65(a).....	18

EXHIBIT LIST	
2001	Lowenstein Declaration In Support of Notice of Intent
2002	Woo Declaration In Support of Notice of Intent
2003	Complaint in <i>Sec. First Innovations, LLC v. Int’l Bus. Machs. Corp.</i> , 1-25-cv-00514 (E.D. Va. Mar. 24, 2025), ECF No. 1 [Complaint]
2004	“Security First Corp. Presentation,” (March 4, 2015) [3/4/15-O’Hare Presentation] (CONFIDENTIAL)
2005	“IBM and Security First Corp to co-develop security capability for cloud computing,” PROACTIVE (last updated Nov. 9, 2010), https://www.proactiveinvestors.co.uk/companies/news/74758/ibm-and-security-first-corp-to-co-develop-security-capability-for-cloud-computing-9918.html [Proactive-Investors]
2006	“SFC – IBM Status Report” (September 16, 2011) (slip sheet omitted) [9/16/11 SFC-IBM Status Report] (CONFIDENTIAL)
2007	“IBM and Security First Corp. to Develop Integrated Security Technology,” SECURITY TODAY (Aug. 1, 2011), https://securitytoday.com/articles/2011/08/01/ibm-and-security-first-corp.-to-develop-integrated-security-technology.aspx [Security-Today]
2008	[8/24/09 Email] (slip sheet omitted) (CONFIDENTIAL)
2009	[11/11/13 Presentation] (CONFIDENTIAL)
2010	[2/15 Presentation] (CONFIDENTIAL)
2011	[3/11 IBM Presentation] (slip sheet omitted) (CONFIDENTIAL)
2012	Excerpts from File History of U.S. Patent Application No. 16/197,275 [’275-FH]
2013	U.S. Patent No. 8,447,695 [’695 patent]

2014	Excerpts from File History of U.S. Patent No. 8,447,695 ['695-FH]
2015	Licensed Works Agreement [LWA] (CONFIDENTIAL)
2016	2013 Statement of Work [2013 SOW] (CONFIDENTIAL)
2017	2015 Statement of Work [2015 SOW] (CONFIDENTIAL)
2018	“IBM Completes Acquisition of Cleversafe,” PRITZKER GROUP (Nov. 6, 2015), https://www.pritzkergroup.com/ibm-completes-acquisition-of-cleversafe/
2019	Jing Cao, “IBM Paid \$1.3 Billion to Acquire Cleversafe in Hybrid-Cloud Push,” BLOOMBERG (Feb. 24, 2016), https://www.bloomberg.com/news/articles/2016-02-24/ibm-paid-1-3-billion-to-acquire-cleversafe-in-hybrid-cloud-push (slip sheet omitted)
2020	[10/9/06 Email] (slip sheet omitted) (CONFIDENTIAL)
2021	[12/6/16 Email] (slip sheet omitted) (CONFIDENTIAL)
2022	Reserved
2023	Order Granting Motion to Stay in <i>Sec. First Innovations, LLC v. Int’l Bus. Machs. Corp.</i> , 1-25-cv-00514 (E.D. Va. Aug. 20, 2025), ECF No. 88 [Stay Decision]
2024	Chart of Settled Expectations Decisions [Chart]
2025	Scheduling Order in <i>DivX, LLC v. Amazon.com, Inc.</i> , No. 1:24-cv-2061 (E.D. Va. July 1, 2025), ECF No. 69 [Recent Scheduling Order]
2026	Opposition to Motion to Stay in <i>Sec. First Innovations, LLC v. Int’l Bus. Machs. Corp.</i> , 1-25-cv-00514 (E.D. Va. July 29, 2025), ECF No. 71 [Stay Opposition] (CONFIDENTIAL)
2027	U.S. Patent No. 11,178,116

2028	U.S. Patent No. 11,068,609
2029	U.S. Patent No. 10,452,854
2030	U.S. Patent No. 7,391,865 [Orsini-'865]
2031	Declaration of Sam Malek, PhD. [Malek-Decl.]

I. INTRODUCTION

Before reaching the merits of the Petition, the Deputy Director should exercise her discretion and deny institution. *First*, the patent at issue in this IPR has been in force for over a decade and Petitioner has known about this patent for years.

Second, the original examination by the USPTO Examiner was correct. In fact, one of Petitioner's key references, Orsini, was before the Examiner during prosecution, was thoroughly considered and addressed by the Examiner and the claims were expressly allowed by the Examiner over Orsini. The Examiner's Reasons for Allowance pointedly state that "[t]he closest prior art, Orsini et al (US 7,391,865)¹ discloses a conventional system for securing sensitive data that fails to anticipate or render obvious" the limitations directed to data encryption, integrity, splitting and redundancy recited in the claims of the '802 patent. Petitioner points to no error by the Examiner, and for the reasons discussed in this POPR, there was none. There is simply no reason for the Office to revisit this examination.

Institution should also be denied on the merits as the Petition fails to demonstrate a reasonable likelihood that any challenged claim of U.S. Patent No.

¹ The Petition relies on US2004/0049687, the published patent application that led to the '865 Patent.

8,271,802 (“’802”) patent is unpatentable. Petitioner relies upon two core combinations: Torre and Tajima (Grounds I-IV) and Orsini and Torre (Grounds V-VII). Petitioner’s Torre/Tajima combination fails because, even in combination, the references do not teach or render obvious the claimed step of “generating data splitting information” “usable to determine into which of a plurality of shares . . . a unit of data will be placed.” Ex. 1001 [’802 Patent], cl. 1[C]. In Petitioner’s proposed combination, Torre never “generat[es]” any data-splitting information, and Tajima’s calculated arbitrary-length parameter is not used to determine into which share each unit of data will be placed. *See* Section III.A.

Even if that could be overlooked, Petitioner fails to establish any motivation to combine Torre with Tajima to add “data indicative of [] the encryption key” as required by another limitation of the independent claims of the ’802 Patent Ex. 1001 [’802 Patent], cls. 1[E], 10[E,] 19[G]. Petitioner contends a POSITA would be motivated to append Tajima’s encryption key information to Torre’s tags but that theory relies on the unsupported assumption that that Torre already stores encryption keys with its shreds, when in fact Torre discourages doing so. *See* Section III.B.1. Petitioner neither demonstrates how Torre would benefit from combination with Tajima, *see* Section III.B.2, nor addresses the substantial security vulnerabilities that would be the result from their combination. *See* Section III.B.3.

Petitioner’s Orsini/Torre grounds fare no better; as noted above, the Examiner expressly allowed the claims of the ’802 Patent *despite* Orsini. Orsini and the ’802 Patent have inventors in common, including Mr. Orsini. Here, Petitioner relies entirely on Orsini’s Figure 21 embodiment but that is not the embodiment that supports the claims of the ’802 Patent. The ’802 Patent adds significant new material not in Orsini—ten figures and over twelve columns of specification—that disclose the various limitations of the claims in the ’802 Patent, including new ordered data-splitting, redundancy, and integrity-checking features that are found in the claim limitations. Petitioner treats Orsini’s Figure 21 embodiment as if it included the new features disclosed in the ’802 patent, when it clearly and expressly does not. *See* Section IV.A. The ’802 Patent claims an entirely different encryption scheme.

To take one example, Petitioner contends that Orsini’s Figure 21 employs a “redundant cryptosplit” process allowing the data set to be “restorable by accessing less than all, but at least a threshold number of the plurality of shares” as required by the claims. Pet., 80, 83. But Orsini itself states, expressly, that the Figure 21 embodiment requires *all four* shares for reconstruction. *See* Section IV.B. The claims also recite “encrypting the data set based on an encryption key” as separate from “generating data-splitting information” to allocate the data set into shares—in other words, the data splitting information cannot be the same as the encryption

key. But Petitioner contends, contrary to this requirement, that Orsini discloses these separate limitations by offering Orsini's "session master key" to meet both limitations simultaneously. *See* Section IV.C. Finally, Petitioner contends that a POSITA would be motivated to add Torre's data signature to Orsini to meet the "integrity information" limitation of the '802 Patent claims, but doing so would add no benefit to Orsini's system, is based upon an incorrect interpretation of Orsini's Figure 21, and would impose significant, unnecessary costs on Orsini. Ultimately, there is no plausible motivation for Petitioner's proposed combination beyond impermissible hindsight. *See* Section IV.D.

For these reasons and others detailed below, institution should be denied.

II. BACKGROUND

A. Overview Of The '802 Patent And The Challenged Claims

The '802 Patent relates to, *inter alia*, a cryptographic system for "securing virtually any type of data from unauthorized access or use" by means of a method comprising "encrypting the data to be secured" with an encryption key and then "parsing, splitting and/or separating the data to be secured" into a plurality of portions, or "shares," which may be stored in "respective, separate storage locations." Ex. 1001 ['802 Patent], 2:30-2:35, cl. 1[F]. One important feature of this method is that redundancy information is included in the shares, such that "the data set is restorable by accessing less than all, but at least a threshold number of"

the shares. *Id.*, cls. 1[G], 10[G], 19[I]. Another is that the system generates, and stores, integrity information to ensure that the data is properly reconstituted from the shares. The methods disclosed and claimed in the '802 Patent ensure that the data is protected against corruption or loss.

As Patent Owner's expert, Dr. Malek,² explains:

As a first step, the method comprises “creating integrity information using the data set.” Ex. 1001 ['802 Patent], cls. 1[A], 1[E], 10[A], 10[E], 19[C], 19[G]; *see id.* 64:3-9, 64:30-35. The next step involves “encrypting the data set based on an encryption key.” *Id.*, cls. 1[B], 10[B], 19[D]. An encryption key is used to transform the data set so that it cannot be read without the encryption key.

After encryption, the '802 Patent next transforms the encrypted data set by “separating the encrypted data set into a plurality of shares.” Ex. 1001 ['802 Patent], cls. 1[D], 10[D], 19[F]. This step entails splitting the encrypted data set by placing encrypted data units into multiple subsets or “shares.” To enable this step, prior to splitting the

² Dr. Sam Malek is a Professor in the Informatics Department within the School of Information and Computer Sciences at the University of California, Irvine. He is also the Director of the Software Engineering Analysis Laboratory. His credentials and relevant experience are included in Exs. 2031 [Malek-Decl.] and 2031 [Malek-CV].

data, the '802 Patent generates “data splitting information” that is used to determine into which share each “unit” of the encrypted data set will be distributed. *See id.*, cls. 1[C], 10[C], 19[E]; Figs. 33-35; 71:27-72:29 (explaining that encryption precedes hash-based generation of split control applied to the encrypted data, followed by separation into shares). Once the encrypted data is separated into the plurality of shares, the method of the '802 Patent also includes within the plurality of shares “data indicative of (a) the encryption key and (b) the integrity information.” *Id.*, cl. 1[E], 10[E], 19[G].

The claims also require that “the data set [be] restorable by accessing less than all, but at least a threshold number of, the plurality of shares.” *Id.* cls. 1[G], 10[G], 19[I]. This is accomplished by including redundant information in the shares that allow recovery of the complete dataset from fewer than all shares, to ensure that the data set can be recovered if some of the shares are lost or corrupted. *Id.* 71:39-72:33 (describing redundancy information “appended . . . to allow for the restoration . . . using fewer than all the data portions”).

Finally, once a plurality of shares has been created, including redundant data and data indicative of the encryption key and integrity information, the shares can be stored in “respective separate storage locations” to further protect the data set.

Ex. 2031 [Malek-Decl.] ¶¶ 33-36.

B. Overview Of Torre

Torre addresses different issues, in a different way, than the '802 Patent. Namely, Torre discloses a “shredding and deshredding system for storage and retrieval” that includes the use of redundancy to improve fault tolerance. Ex. 1003, [0007], [0111]. Among other things, Torre disclose a system that differs from the '802 Patent in the way it encrypts, parses, stores and retrieves data. Torre discloses a different storage strategy, and is designed to achieve different goals, than the '802 Patent.

Dr. Malek explains the differences:

Torre's shredder splits input data into “smaller pieces of data called shreds,” and “adds redundancy to the shreds so that the input data can be recovered from a fewer number of shreds than the total number stored.” *Id.*, [0051], [0052]. The shredder then stores the shreds, which “may involve writing the shredded input data to a single hard disk or single memory,” or may involve storage “in multiple storage units.” *Id.*, [0052]. The input data set can later be recovered with fewer than all of the shreds. Among other differences, Torre uses different encryption techniques and does not include or store “data indicative” of an encryption key or integrity information with its shreds.

Ex. 2031 [Malek-Decl.] ¶ 38. Torre also has different goals than the '802 Patent—Torre emphasizes fault tolerance and efficiency, while the '802 Patent is focused on data security and integrity. *Id.* Although Torre teaches that its various data

transformations can be arranged in several different orders, none teach a system with the same characteristics as the '802 Patent's claimed invention. In short, Torre claims a different data storage and retrieval system—one that, among other things, does not include the same type of encryption method or strategy, does not store encryption keys or integrity information with its shreds, does not include the generation of data-splitting information, and discloses a different method for reconstructing data. Ex. 2031 [Malek-Decl.] ¶ 39.

As part of its fault-tolerant system, Torre discloses that various optional data transformations can be implemented. Ex. 1003, [0049]. Example transformations include “compression, encryption, and digital signature generation.” *Id.*, [0084]. These transformations occur at one of several “transformers” as shown below on the left side of Torre's Figure 32:

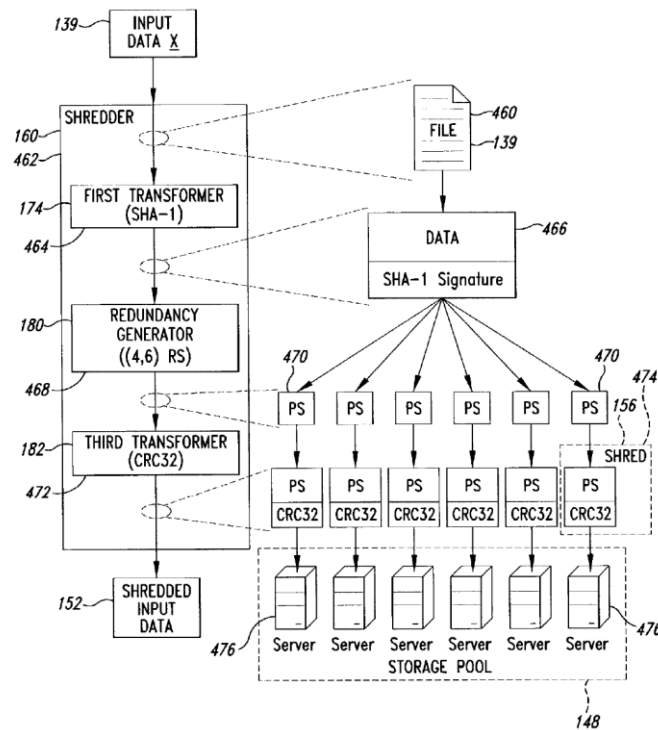


Fig. 32

Ex. 1003 [Torre], Fig. 32.

Unlike the '802 Patent, however, each transformation of data that occurs in Torre's system is recorded on a "tag." These tags are appended to the transformed data or data shred to "help keep track" of the data "as the pieces . . . move through the shredder." Ex. 1003, [0078]. Each tag contains "[i]nformation describing how the shreds were created" such that when the data is retrieved, Torre's deshredder can use the tag to identify the proper inverse transformation, which could be "decompression, decryption, or signature verification." *Id.*, [0053], [0123]. Unlike

the '802 Patent, Torre's deshredder uses these tags to guide properly "reassembl[y of] the original input data." *Id.*, [0053].

If, for example, a transformer used a block cipher to encrypt data shreds, the transformer would tag the encrypted shreds with "the cipher block chaining" that identified the "type of cipher" used to encrypt the shred. *Id.*, [0079]. To undo the encryption, Torre's deshredder would have to use the tag to locate the encryption key, then use it to decrypt the shreds. *See* Ex. 1003, [0084], [0085]; Ex. 2031 [Malek-Decl.], ¶ 42. Torre recommends specific encryption techniques that are compatible with its use of tags and require a separate, external key management system. Ex. 2031 [Malek-Decl.], ¶ 42. These are very different encryption methods than those disclosed and claimed in the '802 Patent—which, instead of an external key management, requires that data indicative of the encryption key be stored with the data shares. *See* Ex. 1001 ['802 Patent], cl. 1.

C. Overview Of Tajima

Tajima is directed to long-term "registration" (*e.g.*, storage) of personal information in databases connected to the Internet, particularly in the e-commerce context. Ex. 1004 [Tajima], [0044]; Ex. 2031 [Malek-Decl.], ¶ 44.

Tajima discloses a system and method for securing personal shopping information by dividing the personal information into "data portions" or "data fragments" and storing each fragment under "separate" control by different

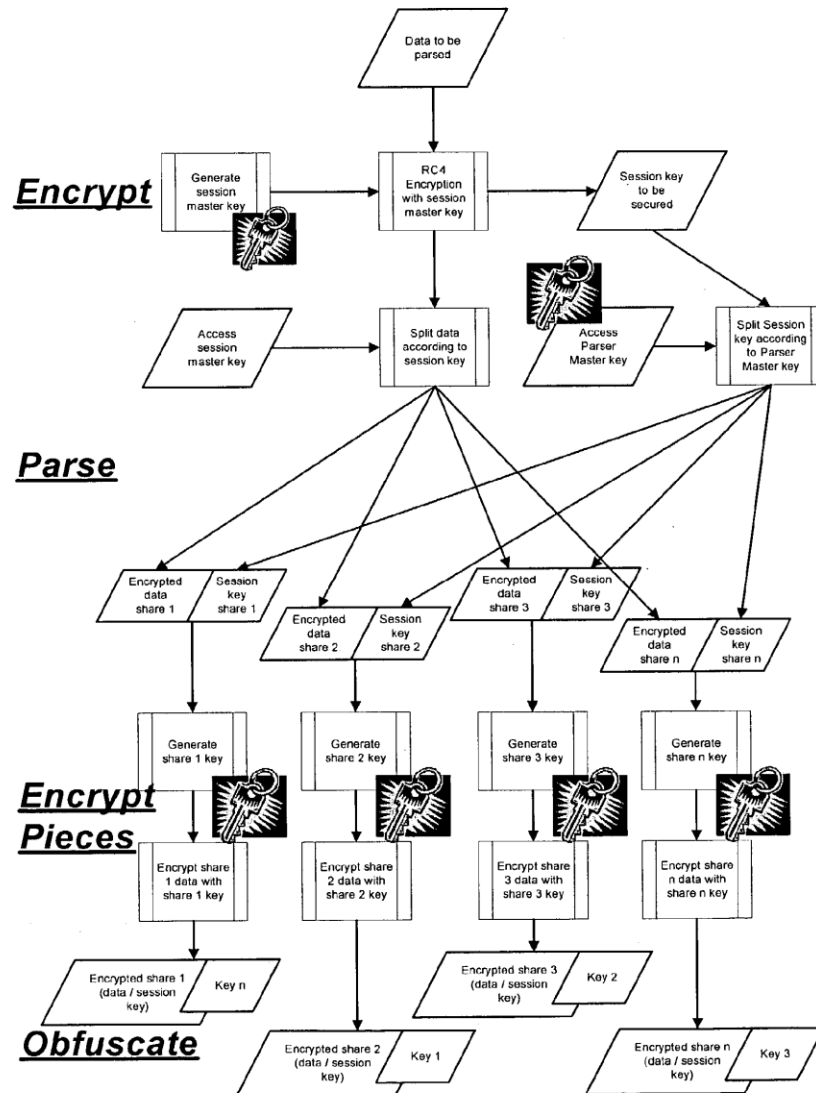
entities. Ex. 1004 [Tajima], [0046], [0072]. Separate control is required because, unlike the system disclosed and claimed in the '802 Patent, Tajima stores information regarding the encryption key used to encrypt one data fragment with a different fragment that the key cannot decrypt, and then relies the fact that the fragments are stored under separate control to ensure that a single adversary cannot gain access all of the fragments and thereby recover all of the needed keys. Importantly, Tajima states that its security stems from the fact that “the personal information cannot be viewed unless all of the areas in which data are registered are exposed” (in which case, of course, it can be). *Id.*, [0048].

This is fundamentally different from the '802 Patent. Ex. 2031 [Malek-Decl.] ¶ 45. The data security afforded by the '802 Patent's system does not depend on separating data among “separate” control, like in Tajima, and the '802 Patent does not parse data in the same way. *Id.* The '802 Patent performs its cryptographic operations such that the data set is encrypted, data splitting information is generated to determine the placement of each unit of that data, the data is split according to that information, and the resulting shares include integrity and key information that allow the original information to be reconstructed from only a threshold of those shares. *Id.*

D. Overview Of Orsini

The key disclosure of Orsini describes a process for data storage that includes encryption and allocation of the data into shares using a single key followed by storage of that encryption key with the stored shares. Ex. 1008 [Orsini], [0054], [0294], [0297]. Petitioner relies entirely on a single embodiment depicted in Orsini's Figure 21, shown below:

Figure 21



Ex. 1008 [Orsini], Fig. 21.

As Dr. Malek explains:

Figure 21 teaches a method of first generating a session master key to encrypt the data (*id.*, [0295]), then using it to separate the encrypted data into “four shares or portions of parsed data,” (*id.*, [0296]), storing the session master key “along with the secured data

shares” such that the four shares “contain encrypted portions of the original data and portions of the session master key,” (*id.*, [0297]-[0298]), and then “[e]ncrypting each share, then stor[ing] the encryption keys in different locations from the encrypted data portions or shares” (*id.*, [0299]). It additionally teaches that the single session master key both encrypts the dataset and dictates how the dataset is split into shares. *Id.* [0295]-[0299]; *see also* Pet., 92-93. Importantly, Orsini’s Figure 21 embodiment does not include redundancy operations or the creation of integrity information, and the shares do not include redundant information or integrity information. Thus, when the shares are retrieved to reconstruct the data set, ***all*** the shares are required to reassemble the dataset and there is no way to ensure that the reconstruction is correct. Ex. 1008 [Orsini], [0368]; [0384]-[0385].

Ex. 2031 [Malek-Decl.] ¶ 47.

Although ’802 Patent shares several disclosures with Orsini, the ’802 Patent introduces new disclosures and figures and recites claims based on this new material. *Id.* Specifically, the new disclosures in the ’802 Patent support at least the limitations in its claims directed to (i) restoring a data set with fewer than all the shares, (ii) generating data-splitting information to create the shares, and (iii) including information to ensure share integrity in the shares. Orsini does not disclose these additions. *Id.*

III. PETITIONER’S TORRE-TAJIMA GROUNDS FAIL (ALL CLAIMS, GROUNDS I-IV).

A. **Petitioner Fails To Demonstrate That Its Proposed Torre/Tajima Grounds Teach “Generating Data Splitting Information” That Is “Usable To Determine Into Which Of A Plurality Of Shares Of Data A Unit Of Data ... Will Be Placed” (All Claims, Grounds I-IV).**

Each of the independent claims requires “generating data splitting information” that is “usable to determine into which of a plurality of shares of data a unit of data . . . will be placed.” Ex. 1001 [’802 Patent], cls. 1[C], 10[C], 19[E]; Figs. 33-35; 71:27-72:29. Petitioner relies upon a combination of Torre and Tajima for this limitation, proposing two alternative systems—Torre/Tajima System-1 and Torre/Tajima System-2. Pet., 36, 43. But neither system discloses the generation of data splitting information usable to determine into which of a plurality of shares of data a unit of data will be placed. This failure to disclose the generation of data splitting information is another distinction between these systems and those disclosed and claimed in the ’802 Patent, fatal to the Petition.

Petitioner’s Torre/Tajima System-1 relies on what it calls Torre’s “selection information” as corresponding to the claimed “data-splitting information.” *Id.*, 36-39. That theory fails because at least because the Petition does not show that Torre *generates* “selection information,” and for the additional reason that “selection

information” is not used to determine where each unit of data is placed in Torre’s shreds, as required by the claims of the ’802 Patent. *See* Section III.A.1.

In its Torre/Tajima System-2, Petitioner relies on Tajima, which divides a data set into portions of arbitrary length and then allocates each portion to a fragment in a “round-robin” fashion. Pet., 53. Petitioner, however, fails to demonstrate that the “arbitrary length” information determines into which share a “unit of data” will be placed, as required by the claims of the ’802 Patent. *See* Section III.A.2.

1. Petitioner Fails To Demonstrate That “Torre/Tajima System-1” Discloses “Generating Data Splitting Information.”

Petitioner relies solely on Torre to disclose this limitation in Torre/Tajima System-1. *First*, Petitioner asserts—but fails to demonstrate—that Torre inherently discloses “generating data-splitting information” through its references to “remapping” and “partitioning” operations. *Second*, implicitly acknowledging that Torre never expressly discloses this limitation, Petitioner argues that it nonetheless would have been obvious to generate such information. *Third*, Petitioner argues that Torre’s disclosure of a Copy-N redundancy scheme discloses the limitation. All these theories fail for the reasons set out below. *See* Section III.A.1-III.A.3.

a. Torre Does Not Inherently Disclose “Generating Data Splitting Information.”

Petitioner first argues that Torre discloses “generating data splitting information” because Torre says it can use “various partitioning methods” to remap transformed data, and explains that “specific remapping is dependent upon particular situations.” Pet., 36 (quoting Ex. 1003 [Torre], [0101]). Petitioner claims that a “POSITA would have understood” that “selection information” in Torre corresponds to the “data splitting information” required by the claims of the ’802 Patent. *Id.*, 36-37.

Petitioner never explains what “selection information” consists of, how it is generated, or how (or even if) it is used to determine where a unit of data is placed into shreds. *See* Pet., 36-37. Petitioner’s contention that Torre “generat[es]” “selection information” is entirely based on Torre’s disclosure that “specific remapping is dependent upon particular situations” and Petitioner’s expert’s unsupported conclusion that “[w]ithout [] selection information, it would not have been possible to remap data.” *See id.* (quoting Ex. 1003 [Torre], [0101]). This argument fails for two reasons.

First, Petitioner’s expert is wrong because “selection information” is not “data splitting information.” The term “selection information” is not used in Torre. As Dr. Malek explains:

Torre itself describes two ways in which remapping can be carried out, “chunking” and “interleaving.” Neither requires “data splitting information” as the term is used in the ’802 Patent. Ex. 1003 [Torre], [0100]. “Chunking” simply takes relatively large blocks of data one at a time and sends each to a shred. *Id.* “Interleaving” simply “routes different pieces of data separately . . . in a round robin fashion.” *Id.* No data splitting information “usable to determine into which of a plurality of shares of data a unit of data . . . will be placed” is required to perform these functions. Ex. 1001 [’802 Patent], cls. 1[C], 10[C], 19[E].

Ex. 2031 [Malek Decl.] ¶ 55.

Second, even assuming for the sake of argument that Torre’s “selection information” did correspond to the “data splitting information,” nothing in Torre suggests that selection information is “generated” by Torres system as a data set is processed for storage. To the contrary, Torre indicates that it is relying on preexisting selection information. For example, Torre discloses that the “interleaving” functionality should be part of the system’s preexisting hardware. Ex. 1003 [Torre], [0100]. Petitioner also attempts to analogize “selection information” to a “case statement” used to write computer programs, but its analogy merely confirms that “selection information” is preexisting and not generated. *See* Pet., 37. The Microsoft Computing Dictionary’s definition of “case statement,” on which Petitioner relies, defines it as comprising the execution

of different instructions based on constants “assigned by the programmer.”

Ex. 1016 [Microsoft Computing Dictionary], 87. But if the “series of constants” is “assigned by the programmer,” those constants were established when the code was written, not generated as it is run. In other words, they are preexisting, they are not “generat[ed].” Ex. 2031 [Malek-Decl.] ¶ 56.

Dr. Zadok’s testimony regarding “selection information” in Torre is wholly conclusory and unsupported. *See* Ex. 1002 [Zadok-Decl.], ¶¶ 256-57. It is therefore entitled to “little or no weight.” 37 C.F.R. § 42.65(a); *Xerox Corp. v. Bytemark*, IPR2022-00624 Paper 12, 5 (Aug. 25, 2022) (precedential) (“[T]he Board was correct in giving little weight to Petitioner’s expert because the expert declaration merely offered conclusory assertions without underlying factual support and repeated, *verbatim*, Petitioner’s conclusory arguments.”); *see also TQ Delta, LLC v. Cisco Sys., Inc.*, 942 F.3d 1352, 1358 (Fed. Cir. 2019) (“This court’s opinions have repeatedly recognized that conclusory expert testimony is inadequate to support an obviousness determination on substantial evidence review.”).

Petitioner has failed to show that the “selection information” it alleges is disclosed in Torre corresponds to the required “data splitting information” required by the claims, or that it is “generated” in Torre/Tajima System-1.

b. Petitioner's Obviousness Theory Also Fails.

Tacitly acknowledging its failure to show that Torre discloses this limitation, Petitioner also asserts that “it would have been obvious to generate such selection information to switch between ‘various partitioning methods’ based on ‘particular situations.’” Pet., 37 (citing Ex. 1003 [Torre], [0100], [0102]). Petitioner relies again on the Microsoft Computing Dictionary’s “case statement” definition to support its claim that “mode switching may be performed by generating selection information.” *Id.* (citing Ex. 1016 [Microsoft], 87). As discussed above, however, that definition supports the opposite of Petitioner’s argument: The constants in a “case statement” are “assigned by the programmer,” not “generated.” Ex. 1016 [Microsoft], 87; Ex. 2031 [Malek-Decl.] ¶¶ 57.

Petitioner also invokes Torre’s statement that its allocator “‘generates information describing its processing,’” to argue that it would have been obvious to use that information prospectively to “select[] the remapping scheme.” Pet., 37 (quoting Ex. 1003 [Torre], [0103]). But Torre’s disclosure is retrospective; it describes information created after the allocator has completed processing that is later used to reconstruct data. *See* Ex. 1003 [Torre], [0078] (“tags [] keep track of what has been done to the pieces of the input data”); [0079] (stating that tags record functions “performed by the shredder” and their order). As Dr. Malek

explains, a POSITA would not equate that after-the-fact record with information used prospectively to determine future data placement:

Petitioner relies upon Torre's teaching of "generat[ing] information describing [the allocator's] processing." Pet., 37 (citing Ex. 1003 [Torre], [0103]). Torre teaches that this information describing the allocator's processing is "append[ed]" along with information describing previous tags "to generate tags 288" which are then appended "with the data blocks 286." Ex. 1003 [Torre], [0103].

Torre makes clear that tags do not prospectively determine future processing. Instead, they merely record completed operations. *Id.*, [0078] ("[t]he tags 110 help keep track of *what [has] been done* to the pieces of the input data as the pieces . . . move through the shredder" (emphasis added)); *see also id.* ("The tags 110 are used to indicate which functions should be performed in various stages of reconstructing the input data 139 from the shredded input data 132."). In accord, Torre's tags may include identifiers of functions, versions, and modes "performed by the shredder," along with sequencing information "call[ing] out the order of steps in which the functions *were performed.*" *Id.*, [0079] (emphasis added).

Ex. 2031 [Malek-Decl.] ¶ 59.

Petitioner's argument rests only on generalized assertions that it would have been "obvious" or "well known" to generate such information, without any supporting evidence or reasoning. Pet., 37. But such thinly veiled appeals to

“common sense” may supply a missing limitation “*only* when ‘the limitation in question [is] unusually simple and the technology particularly straightforward.’” *DSS Tech. Mgmt. v. Apple Inc.*, 885 F.3d 1367, 1374-75 (Fed. Cir. 2018) (internal citations omitted). Where, as here, a “missing limitation goes to the heart of an invention” a court’s review of any purported “reasoned basis for resort to common sense” must be especially “searching.” *Arendi S.A.R.L. v. Apple Inc.*, 832 F.3d 1355, 1363 (Fed. Cir. 2016). The claimed step—generating data-splitting information for encrypted data—is a core aspect of a complex cryptographic process found to be lacking in the prior art, not an “unusually simple” feature. *See* Ex. 1009 [’802 Patent Prosecution History], 569-576 (Examiner finding Orsini fails to render this limitation obvious); Ex. 1001 [’802 Patent], 71:39-72:29; Ex. 2031 [Malek-Decl.] ¶ 61.

Accordingly, Petitioner’s obviousness theory fails because its proposed modification inverts Torre’s teaching: Torre’s “information describing [the allocator’s] processing” is generated only after processing is complete and cannot be repurposed, through “common sense” or otherwise, into the claimed step of prospectively “generating data-splitting information usable to determine into which of a plurality of shares . . . a unit of data *will* be placed,” as required by the claims of the ’802 Patent.

2. Petitioner Fails To Demonstrate That “Torre/Tajima System-2” Teaches That The Alleged “Data Splitting Information” Is “Usable To Determine Into Which Of A Plurality Of Shares Of Data A Unit Of Data . . . Will Be Placed.”

For its Torre/Tajima System-2, Petition drops its theory that Torre’s alleged “selection information” relating to its partitioning process constitutes the required “data splitting information.” For System-2, Petitioner contends that the relevant process is Tajima’s method of dividing personal information. Pet., 40-42.

Petitioner contends that Tajima’s “arbitrary-length” parameter constitutes the required “data-splitting information.” *Id.*, 24, 26. It asserts that “the allocator [of Tajima] determines arbitrary length[s] according to a function of, for example, random numbers, time, or file capacity,” uses those lengths to divide the transformed data and encryption key into portions of the corresponding lengths, and then “assign[s] each data portion to one of the pre-shreds, as in Tajima’s Figure 11.” *Id.*, 24.

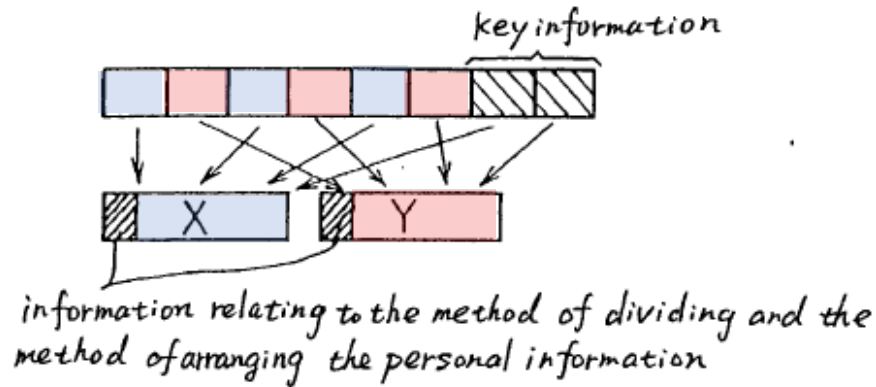
Yet in making that argument, Petitioner cites portions of Tajima (Ex. 1004 [Tajima], [0106]–[0108]) that discuss not only the calculated arbitrary lengths but also the information later appended to each fragment describing how the data was divided and arranged. Pet., 40-41. Petitioner’s theory thus blurs these two distinct disclosures and treats them as a single source of the claimed “data splitting information.” Notwithstanding this conflation, for the reasons discussed below

Tajima’s calculated “arbitrary length” is not data splitting information, and the information appended to the Tajima’s fragments is not “useable to determine into which of a plurality of shares of data a unit will be placed,” as the claims of the ’802 Patent require. *Id.*, 41.

a. Tajima’s Calculated Arbitrary Length Determines Size, Not Placement.

Tajima’s calculated arbitrary length merely determines the size of each unit of data, not its destination. *See* Ex. 1004 [Tajima], [0106]. Petitioner relies on the “method of dividing and arranging the received data to generate data fragments . . . as in Tajima’s Figure 11.” Pet., 24. According to Petitioner, Tajima “assign[s] data portions of ‘arbitrary length[s]’ to different pre-shreds in round-robin.” *Id.*, 26. But in Tajima’s Figure 11, the “round-robin” assignment simply alternates between fragments—**odd portions** are assigned to fragment X while **even portions** are assigned to fragment Y—without regard to the arbitrary length values:

Fig. 11



Tajima, Fig. 11 (annotated); Ex. 2031 [Malek-Decl.] ¶ 67.

Critically, Petitioner never identifies what the claimed “unit of data” is in System-2. This alone warrants denial as it is Petitioner’s burden to “specify where each element of the claim is found in the prior art patents or printed publications relied upon.” 37 C.F.R. § 42.104(b)(4). Even setting that aside, the most natural reading is that the arbitrary-length portions themselves are the alleged “unit[s] of data.” Ex. 2031 [Malek-Decl.] ¶ 68. Yet, by Petitioner’s own account, the fixed round-robin procedural rule—not the arbitrary length—determines placement. Regardless of their length, odd portions are assigned to fragment X and even portions to fragment Y. *Id.* Thus, the calculated arbitrary lengths are not “usable to determine into which of a plurality of shares of data a unit of data . . . will be placed” as required by the claims of the ’456 Patent.

b. Tajima's Appended Information Retrospectively Describes How Data Was Placed, Not How It Will Be Placed As Required By The Claims.

The information that Tajima appends to each data fragment likewise is not usable to determine placement. The appended information—like Torre's tags—is retrospective metadata that merely records how the data was divided and arranged. *See* Ex. 1004 [Tajima], [0107]. This retrospectively appended information enables reconstruction, by showing how “data fragment X and data fragment Y [are to be] combined based on the information relating to the method of dividing and the method of arranging that is attached to each of the data fragments X and Y.” *Id.*, [0108]. Even if this appended information is considered to include the previously calculated arbitrary length, such an arbitrary-length parameter itself contains no information about how fragments are arranged. *See id.*, [0106]-[0107]; Ex. 2031 [Malek-Decl.] ¶ 69. At best, therefore, the appended information facilitates reconstruction of a data set, not its original allocation into fragments: It describes how the data was divided when an existing fragment was created, but it is never used to determine how data will be divided to create new fragments. Ex. 2031 [Malek-Decl.] ¶ 69.

3. Petitioner's Invocation Of The Copy-N Redundancy Transformation Does Not Meet This Limitation.

Petitioner contends that in both Torre/Tajima System-1 and Torre-Tajima System-2, Torre's Figure 34 allocator uses the "selection information"/"arbitrary length information" to determine into which data block (and thus which shred) a unit of data of the encrypted data set will be placed. Pet., 39-42. Petitioner claims that the systems it proposes disclose "generating data splitting information" because after the encrypted data set is allocated into blocks, a Copy-N redundancy transformation *might* be used to create "multiple exact copies of existing data" that is then placed in the shreds, such that "[e]ach shred [] includes a unit of data from the corresponding data block." Pet., 38-39 (quoting Ex. 1003 [Torre], [0113]). Petitioner appears to contend that the decision to use Copy-N redundancy, which merely creates copies of existing data, allows the use of an allocator to satisfy the claim's requirement of "generating data splitting information." Pet., 39-40.

Petitioner's theory is based on the unsupported assumption that the system disclosed in Torre's Figure 34 would be able to use a Copy-N redundancy transformation. Pet., 12-13. Torre indicates that this redundancy transformation cannot be used with every embodiment. Ex. 1003 [Torre], [0113] (noting that copy-N redundancy can only be "used *by some* implementations" (emphasis added)). But the embodiment in Torre's Figure 34 is restricted to using the 4

input-6 output Reed Solomon redundancy transformation, and Torre does not disclose that other redundancy transformations could be implemented in this embodiment. *Id.*, [0131], [0135] (describing Figure 34 as containing an allocator in addition to the redundancy generator 468 depicted in Figure 32). Figure 34's use of a 4-input-6-output Reed Solomon redundancy transformation is a preexisting fact, and it does not require "generating" data splitting information. Ex. 2031 [Malek-Decl.] ¶ 71.

Moreover, in Torre/Tajima Systems-1 and -2, it is the 4-input-6-output Reed Solomon redundancy engine that determines into which shreds the data blocks are placed, not any "data splitting" information allegedly generated during the use of the allocator. Presumably, this is why Petitioner needed to invoke a different redundancy transformation (*i.e.*, Copy-N redundancy transformation) to support its argument, rather the one that its proposed systems actually use. Ex. 2031 [Malek-Decl.] ¶ 72.

B. Petitioner Fails To Demonstrate That A POSITA Would Be Motivated To Combine Torre And Tajima To Achieve "Including In The Plurality Of Shares Data Indicative Of [] The Encryption Key" (All Claims, Grounds I-IV).

The challenged claims require "including in the plurality of shares data indicative of [] the encryption key" used to encrypt the original data set. Ex. 1001 ['802 Patent], cls. 1[E], 10[E], 19[G]. Petitioner contends this limitation is

satisfied by combining Torre, which appends descriptive tags to encrypted data portions or “shreds,” with Tajima’s supposed disclosure of attaching the encryption key itself to encrypted “fragments.”³ *See, e.g.*, Pet., 16-20. Despite Torre’s lack of any express disclosure, Petitioner claims that a POSITA “would have been motivated to modify Torre to include the encryption key in the tag” so that each Torre shred would contain the key and thereby allegedly meet the “data indicative of [] the encryption key” limitation.⁴ Pet., 18.

³ As an initial matter, Petitioner appears inconsistent on whether it views this limitation as taught by Torre alone or as the result of a Torre/Tajima combination. In its “motivation to combine” section, but not in its element-by-element mapping, Petitioner suggests that a tag in Torre “identifying an encryption function is indicative of an encryption key.” Pet., 17; *see* Ex. 1002 [Zadok Decl.] ¶ 188. But Torre itself defines the tag as data “that identifies the transforms used to produce the particular shred.” Ex. 1003 [Torre], [0056]. It never describes the tag as including the encryption key—or any “data indicative of [] the encryption key.” To the extent Petitioner relies on Torre alone, its argument fails at the threshold.

⁴ In this regard, Petitioner implicitly assumes including the encryption key in Torre’s shreds satisfies the requirement that the encryption key itself qualifies as

Petitioner, however, fails to offer any plausible motivation to add an encryption key to Torre's tags, for at least three independent reasons. *First*, Petitioner incorrectly assumes that Torre already discloses including the encryption key with its tags. *See* Section III.B.1. *Second*, Petitioner fails to identify any way in which Torre would benefit from the proposed modification. *See* Section III.B.2. *Third*, Tajima's key-embedding approach is incompatible with Torre. Torre teaches away from adding the encryption key to the shares, which introduces significant and unaccounted for drawbacks that Petitioner ignores. *See* Section III.B.3.

1. Petitioner Incorrectly Assumes That Torre Itself Already Teaches Including The Encryption Key In Its Tags.

Before turning to those drawbacks, Petitioner's combination is based upon the errant premise that Torre already teaches including the encryption keys in the tags. Petitioner asserts that including the encryption key in Torre's tag would be necessary to "undo the encryption operation using the [deshredder] as taught by

"data indicative of . . . the encryption key." Petitioner does not attempt to justify this implicit construction, asserting instead that "no terms require construction for purposes of this IPR." Pet., 8.

Torre.” Pet., 17. But Torre nowhere suggests that its tags contain encryption keys, nor that such inclusion is required for decryption. As Dr. Malek explains:

Petitioner and Dr. Zadok fail to substantiate their assertion that Torre’s tag would have included an encryption key because “otherwise, it would not be possible to undo the encryption operation using the de-shredder as taught by Torre.” *See* Pet., 17; Ex. 1002 [Zadok-Decl.] ¶ 188. In fact, many prior-art systems decrypt data without ever storing within the data any “data indicative of the encryption key,” let alone the encryption key itself. For example, Aldridge teaches that the encryption keys may be “linked to distinctive features of the recipient’s hardware or user identification information.” Ex. 1005 [Aldridge], [0014] (“Access to the data is provided using key technology, where keys can be linked to distinctive features of the recipient’s hardware or user identification information.”); Ex. 1001 [’802 Patent], cl. 21 (similar). To decrypt the data, the recipient would not need “data indicative of [] the encryption key” or the “encryption key” itself to be included in the data because the recipient already has the encryption key.

Ex. 2031 [Malek-Decl.] ¶ 76.

In short, Torre does not render the limitation obvious because it does not disclose the inclusion of encryption keys in its tags, and a POSITA would not understand from the lack of any disclosure that keys need to be included in the tags.

2. Petitioner Fails To Show Torre Would Benefit From Attaching The Encryption Key To Its Tags.

Moreover, Petitioner fails to demonstrate that Torre would benefit from adding the encryption key to its tags as allegedly taught by Tajima. Torre emphasizes that encryption security depends on “proper key management” but is silent on any mechanism of key management used by its systems. *See* Ex. 1003 [Torre], [0085]. A POSITA would therefore understand Torre’s unmodified system already to employ a suitable key-management scheme (although he or she would not know what it is). Ex. 2031 [Malek-Decl.] ¶ 78.

Petitioner offers no reason why adopting Tajima’s approach—embedding the key with the data—would improve upon Torre’s existing “proper key management” and it would not. *See* Ex. 1003 [Torre], [0085].

Torre’s system would gain nothing by incorporating Tajima to embed encryption keys within Torre’s shreds. *See* Ex. 2031 [Malek-Decl.] ¶ 79. Adding Tajima would not improve security as compared to Torre’s approach because anyone obtaining a full set of shreds—including an attacker—could reconstruct the dataset directly. *Id.*; *see also Google LLC v. Security First Innovations, LLC*, IPR2024-002122, Paper 42 at 19 (May 16, 2025) (crediting expert testimony that “Tajima’s method of attaching ‘information regarding the key,’ rather than the key

itself, has the security benefit that an attacker with all of Tajima's data fragments still would not possess all the plaintext keys.”).

3. Tajima's Key-Embedding Approach Is Incompatible With Torre; Torre Discourages Adding The Key To The Tags, Which Entails Significant, Unaccounted-For Drawbacks.

Beyond offering no benefit to Torre's system, Tajima's alleged key-embedding approach is incompatible with Torre, and adding it to Torre would introduce serious security vulnerabilities.

Petitioner incorrectly contends that the Torre/Tajima combination would include the encryption key within the “tag” created by Torre's first transformer. Pet. 17-18. But Torre discloses that the tags do not include the output of the functions performed by the shredder that are included in the tags, but rather include the “identifiers of the particular functions.” Ex. 1003 [Torre], [0078]-[0079]. For encryption functions, Torre discloses that a function identifier could include the “type of cipher and the cipher block chaining” used to encrypt the data. *Id.* Torre teaches that each transformer within the shredder appends these identifiers of the particular function(s) to the tags generated by the previous transformers. Ex. 1003 [Torre], [0119]. Then, to reconstitute the shreds into the original data input, a “tag reader” reads a tag to “obtain a list of identifiers of transformations performed during the shredding process. *Id.*, [0127]. Even assuming the tags contained the encryption keys, Petitioner recognizes that Tajima's Figure 11 teaches that the

“key information” and encrypted data are each split into data fragments. Pet., 18 (Tajima’s Figure 11 “split[s] the collection of encrypted input data and the encryption key into data fragments.”). But for Torre’s tag reader to perform as described, the tags themselves must contain the full identity of the encryption method used—in this case, the encryption performed by the first transformer on the entire original data. Ex. 2031 [Malek-Decl.] ¶ 81. Including only a portion of the key in each tag would render the tags meaningless to both the transformers as they append subsequent tag information and the tag reader. *Id.* Thus, if Petitioner’s assertion that Tajima’s Figure 11 embodiment was included in Torre, each tag would contain only a fraction of the original encryption key, and Torre’s tag reader would be unable to obtain the identifier of the particular encryption method or reassemble the encryption key. *Id.*

The Torre/Tajima combination also poses significant security risks. Because the claim limitation of the ’802 Patent requires that the data be encrypted before the shares are created, only one key is needed to decrypt the entire data set. If that key, or a portion of the key, is attached to each shred within a tag as Petitioner suggests, Pet. 46; Ex. 1002 [Zadok-Decl.], ¶¶ 283-84, an attacker who possesses even one of the shreds or who has access to Torre’s centralized “storage pool” in which all the storage devices are under common control would also possess the encryption key or all portions of the encryption key. Ex. 2031 [Malek-Decl.] ¶ 87.

This risk is far greater in Torre than in Tajima because Tajima’s system is explicitly designed to mitigate such exposure through physical and administrative separation of storage locations. Tajima explains that if personal information “is registered with one area that is connectible to the Internet, there is the danger that . . . this information may be viewed” by unauthorized attackers. Ex. 1004 [Tajima], [0042]. Part of Tajima’s solution is to store each key used to encrypt its data fragments with a fragment that that key cannot decrypt. Ex. 2031 [Malek-Decl.] ¶ 83. So even if an attacker could get access to one fragment, he could not use the included key to decrypt the fragment. *Id.* The other aspect of Tajima’s solution is to “register[] the other data portions with other areas . . . under control that is separate from the authentication means or service provider,” rendering an attack that uncovers all the keys unlikely, as areas under separate control are likely to have different access policies. Ex. 1004 [Tajima], [0046]. The result is that Tajima’s architecture can safely associate key information with the data only because those portions are stored under separate control.

By contrast, Torre discloses no such separation of keys or control, and Tajima’s solution would be incompatible with Torre. *First*, to achieve the invention of the ’802 Patent, the Torre/Tajima combination secures only the single key used to encrypt the original data set, unlike the multiple keys that can be shuffled with the data fragments to provide security in Tajima. *Second*, Torre

discloses that its data is stored in a centralized “storage pool” in which all the storage devices are under common control. It discloses a storage architecture in which “the shreds 474 are then sent to the storage pool 148 . . . [which can be] made up of six servers 476,” or even “a single rack containing thousands of hard drives . . . [creating] a giant highly fault-tolerant drive.” Ex. 1003 [Torre], [0132], [0069]. Although Torre notes that its storage units could be physically separate, it never suggests distinct administrative control or isolation among them. Ex. 2031 [Malek-Decl.] ¶ 87. Petitioner likewise does not propose modifying Torre to introduce such separate control, instead relying on Torre alone as “causing the plurality of shares to be stored in respective separate storage locations.” Pet., 47. Accordingly, importing Tajima’s key-embedding feature into Torre—without also adopting Tajima’s key allocation and multi-authority control—would create a significant new vulnerability because an attacker with access to any storage node would gain both the encrypted data and its decryption key. *See* Ex. 2031 [Malek-Decl.] ¶ 87.

Lastly, Torre discourages this combination because Tajima’s internal key management scheme is contrary to one of Torre’s central teachings of external key management. Torre specifically recommends certain encryption methods that are compatible with its system, including “[s]tream ciphers, such as RC4,” and “[b]lock ciphers, such as the Advanced Encryption Standard (AES), the Data Encryption

Standard (DES), and Triple-DES (3DES).” Ex. 1003 [Torre], [0085]. Per Dr. Malek, each of these suggested encryption techniques examples employ external key management:

A POSITA would know that key management increases security of an encrypted data set by preventing unauthorized access to the key. Cryptographic operations that encrypt a data set with a key do not themselves obfuscate the key. This is true of typical stream ciphers (*e.g.*, RC4) and block ciphers, (*e.g.*, AES, DES, and 3DES), all of which return an encrypted data set and the unencrypted key. A POSITA would know that to increase security, these encryption algorithms should secure the key with a management system that obfuscates and stores the key separately from the encrypted data.

Such external key management systems offer several benefits. *First*, they minimize the amount of data passing through a given system. In general, the less data that moves through a system, the lower the computational demand on the system, and the faster the data can move. *Second*, external key management ensures that access to the key is not hindered by loss or corruption of the shares.

Ex. 2031 [Malek-Decl.] ¶¶ 89.

Tajima, in contrast, teaches an internal key management scheme by storing the keys alongside the data. Thus, Tajima’s key-embedding scheme would be contrary to one of Torre’s central teachings of external key management and fundamentally incompatible with Torre’s architecture.

The Petition does not address these substantial drawbacks. But the Federal Circuit has made clear that “[t]he Board must weigh the benefits and drawbacks of the modification against each other, to determine whether there would be a motivation to combine.” *Arctic Cat Inc. v. Polaris Indus.*, 795 Fed. App’x. 827, 833 (Fed. Cir. 2019); *Henny Penny Corp. v. Frymaster LLC*, 938 F.3d 1324, 1331-32 (Fed. Cir. 2019) (similar); *Arctic Cat Inc. v. Bombardier Rec. Prods.*, 876 F.3d 1350, 1363 (Fed. Cir. 2017) (“The potential problems” with combination supported finding that “a skilled artisan would not have been motivated” to make combination). A POSITA would recognize that embedding the encryption key within Torre’s shreds—thereby exposing both the data and the key to compromise—precisely reflects the absence of “proper key management” that Torre itself warns against. Such a modification would have dissuaded, not motivated, a POSITA from combining Torre and Tajima.

In short, Torre does not disclose this limitation, and Petitioner’s proposed Torre/Tajima combination provides no benefit over Torre’s already existing key management and instead introduces substantial security risks. Considering these issues, Petitioner’s motivation to combine is not plausible, and the only apparent rationale for the combination is that Petitioner needs the combination to cover certain claim limitations. This is classic hindsight reasoning—the very approach the Federal Circuit has repeatedly rejected. *Univ. of Strathclyde v. Clear-Vu*

Lighting LLC, 17 F.4th 155, 165 (Fed. Cir. 2021) (rejecting use of “[t]he inventor’s own path” to support “a conclusion of obviousness,” because “that is hindsight.”); *Ruiz v. A.B. Chance Co.*, 357 F.3d 1270, 1275 (Fed. Cir. 2004) (rejecting “hindsight reasoning, using the invention as a roadmap to find its prior art components” to show obviousness); *see also Interconnect Planning Corp. v. Feil*, 774 F.2d 1132, 1143 (Fed. Cir. 1985) (rejecting an obviousness claim that lacked “any suggestion in the prior art” to make a combination, other than “hindsight gleaned from the invention itself.”); *InTouch Techs., Inc. v. VGo Commc’ns., Inc.*, 751 F.3d 1327, 1351 (Fed. Cir. 2014) (rejecting expert’s obviousness theory that “relied on the [] patent itself as her roadmap for putting what she referred to as pieces of a ‘jigsaw puzzle’ together.”).

IV. PETITIONER’S ORSINI/TORRE GROUNDS FAIL (ALL CLAIMS, GROUNDS V-VII).

A. The Claims Of The ’802 Patent Are Supported By New Disclosures That Are Not In Orsini (All Claims, Grounds V-VII).

Orsini is Patent Owner’s application and shares three (of four) named inventors, including Mr. Orsini. Figures 1–25 and paragraphs 1–424 of Orsini were incorporated verbatim into the ’802 Patent’s specification. The ’802 Patent, however, adds ten new figures and over twelve columns of additional material, including Figures 26–35 and columns 64:3–75:8; the added material introduces new functionality addressing a different problem than Orsini. Ex. 1001 [’802

Patent], Figs. 26-35, 64:3-75:8. Petitioner’s assertion that the two disclosures are “substantially identical” overlooks the significance of this added material. Pet., 82-84. The claims in the ’802 Patent reflect the new material and represent an entirely different encryption scheme than what was disclosed in the Orsini reference.

Petitioner’s Grounds V-VII rely entirely on Orsini’s Figure 21. *See, e.g.*, Pet., 82, 91-94, 95-96. At a high level, Figure 21, and the independent claims of the ’802 Patent each describe encrypting a dataset, splitting it into shares, and storing those shares separately. *See* Ex. 1001 [’802], cls. 1, 10, 19; 1008 [Orsini], Fig. 21, [0294]-[0300]. But Orsini’s Figure 21 and the new ’802 material (and the claims of the patent) are directed to different scenarios and problems, and different solutions.

Orsini’s Figure 21 embodiment is directed to steps for storing data in a depository. *See* Ex. 1008 [Orsini], [0054] (“FIG. 21 illustrates a process for parsing, splitting or separating data with encryption and ***storage of the encryption master key with the data.***”); [0294] (“steps of the process performed . . . to store the session master key with the parsed data”); [0297] (“the session master key will be ***stored along with the secured data shares in a data depository***”).

The ’802 Patent adds features that appear in its claims but not in Orsini’s Figure 21, and these claims are supported by disclosures that first appear in the

specification of the '802 Patent. *See* Section IV.A. *First*, the claims require that “the data set [be] restorable by accessing less than all, but at least a threshold number of, the plurality of shares.” Ex. 1001 ['802 Patent], cls. 1[G], 10[G], 19[I]. Orsini’s Figure 21, however, teaches the opposite: “Unless all four shares are retrieved, the data cannot be reassembled according to this example.” Ex. 1008 [Orsini], [0368]; *see also id.*, [0294]-[0300] (stating that to restore, “steps are reversed”); *id.*, [0381]-[0382] (“[a]ll shares or portions of the data set” are utilized to restore the data set). In contrast, figures (Figs. 33-35) and text that appear in the '802 Patent but not in Orsini disclose splitting methods and redundancy logic that allow recovery of the complete dataset from fewer than all portions. Ex. 1001 ['802 Patent], 72:30-33 (describing redundancy information “appended ... to allow for the restoration ... using fewer than all the data portions”). *See* Section IV.B.

Second, the '802 Patent separates encrypting the dataset from another, distinct step of generating data splitting information that is used to determine, for the encrypted data, how each unit is distributed. *See* Ex. 1001 ['802 Patent], cls. 1[B]-[C]; Figs. 33-35; 71:27-28, 72:23-29 (encryption by encryption key is distinct from hash-based generation of splitting information used to separate followed by separation into shares). This enhances the flexibility and security of the storage method. Ex. 2031 [Malek-Decl.] ¶ 98. In contrast, Orsini’s Figure 21 embodiment uses a single session master key to both encrypt the dataset and to determine how it

is split; the key is created in a single step and reused for both functions. Ex. 1008 [Orsini], [0295]-[0299] (describing the generation of the session key and its use in encrypting *and* splitting a data set); Pet., 92-93 (treating the session key as both an encryption key and as a splitting key). The post-encryption generation and use of split information required by the claims appear only in the new '802 material, not in Orsini's Figure 21. *See* Section III.C.

Third, the '802 Patent adds “creating integrity information using the data set” and “including in the plurality of shares data indicative of . . . the integrity information.” Ex. 1001 ['802 Patent], cls. 1[A], 1[E], 10[A], 10[E], 19[C], 19[G]; *see also* 64:3-9, 64:30-35. In text that is not in Orsini, the specification of the '802 Patent describes an error-checking component to “assure the integrity of the data within a portion.” *Id.*, 64:2-9. Petitioner concedes that Orsini does not disclose any integrity-verification mechanism. Pet., 85; *see also* Ex. 1008 [Orsini], [0294]-[0300]. *See* Pet., 88-89; Section III.D.

In short, the '802 Patent introduces new disclosures and figures, as compared to Orsini, and recites claims based on this new material that amount to new and different ways to encrypt, secure, store and retrieve data. Specifically, the new disclosures in the '802 Patent support at least the limitations in its claims directed to (i) restoring a data set with fewer than all the shares, (ii) generating data-splitting information to create the shares, and (iii) including information to

ensure share integrity in the shares. These additions are not disclosed by Orsini's Figure 21 or its discussion of that figure, and amount to a fundamentally different way of securing data from unauthorized access. As explained below, combining Torre with Orsini does not eliminate Orsini's deficiencies and, importantly, does not transform the entire encryption scheme of the Orsini reference into that encryption scheme claims in the '802 Patent.

B. Petitioner Fails To Demonstrate That Orsini/Torre Discloses Or Renders Obvious Restoring The Data Set From “Less Than All” Of The Shares (All Claims, Grounds V-VII).

Each of the claims requires that the “data set is restorable by accessing *less than all*, but at least a threshold number of, the plurality of shares.” *See* Ex. 1001 ['802 Patent], cls. 1[G], 10[G], 19[I]. Petitioner contends that “Orsini discloses or renders obvious 1[G],” (Pet., 96; *see also id.*, 98-99), relying on Orsini's Figure 21 embodiment. Petitioner's argument, however, is unsupported and contrary to what Orsini does disclose.

1. Contrary To Petitioner's Contention, Orsini Makes Clear That Figure 21 Requires Retrieval Of All Four Shares To Reassemble The Data Set.

Petitioner first contends that Orsini's Figure 21 discloses the limitation requiring that the “data set is restorable by accessing *less than all*, but at least a threshold number of, the plurality of shares.” Pet., 96. Petitioner argues that Figure 21 employs a “redundant cryptosplit” process wherein redundant

information is added to the shares, so that fewer than all shares are needed to reconstruct the data set. *Id.*

Orsini makes clear, however, that in the Figure 21 embodiment, “[u]nless all four shares are retrieved, the data cannot be reassembled[.]” Ex. 1008 [Orsini], [0368]; *see also id.*, [0294]-[0300], [0381]-[0382]. Critically, Petitioner never demonstrates that Orsini’s Figure 21 employs (or can employ) redundant cryptosplitting. Certainly, nothing in Figure 21 or its attendant description discloses redundant cryptosplitting or redundancy of any kind.

Petitioner also argues that Orsini renders limitation 1[G] obvious, asserting that it would have been “obvious to use [a] redundant cryptosplit process” in Orsini’s Figure 21 embodiment based on Orsini’s disclosure that “[t]he parser software suite of the present invention . . . perform[s] a cryptographic split.” Pet., 96. As explained above, Petitioner’s claim of obviousness is inconsistent with Orsini’s express disclosure that the data set of Figure 21 *cannot* be restored with fewer than all the shares. Ex. 1008 [Orsini], [0368]. In addition, Petitioner’s argument improperly ignores the distinction between a cryptosplit—wherein a data set is merely divided into shares—and a **redundant** cryptosplit wherein the data set may be reassembled from fewer than all the shares. As Orsini explains, “[a] cryptographic split (cryptosplit) partitions the data into N number of shares.” Ex. 1008 [Orsini], [0281]. Orsini distinguishes redundant cryptosplitting as a

“process[] to provide sufficient redundancy in the shares such that only a subset of the shares is needed to reassemble or restore the data to its original or useable form.” *Id.*, [0283]. Thus, the “redundant cryptosplit” Petitioner relies on is not the “cryptosplit” used in Orsini’s Figure 21 embodiment. *See* Ex. 2031 [Malek-Decl.] ¶ 104.

In fact, redundant cryptosplitting cannot be obvious because it cannot be used with the Figure 21 embodiment. Orsini discloses that “the session master key” used to encrypt the data set of Figure 21 is “split into ‘n’ shares . . . (SK1, SK2, SK3, SKn)[.]” Ex. 1008 [Orsini], [0363]; *see also id.*, [0294] (“As shown in FIG. 21, this embodiment of the present invention shows the steps of the process performed by the parser software suite on data *to store the session master key with the parsed data*[.]” (emphasis added)); *see also id.*, [0297] (“In this embodiment of the method, *the session master key will be stored along with the secured data shares* in a data depository.” (emphasis added)), *see also id.*, [0054] (“FIG. 21 illustrates a process for parsing, splitting or separating data with encryption and storage of the encryption master key with the data.”).⁵ Orsini then discloses that in

⁵ In this regard, Figure 21, which incorporates the session key by storing it with the data, is distinct from, *e.g.*, Figures 22 and 24, which do not “incorporate” the

this example, a portion of a session master key (SK1, SK2, SK3, SKn) is stored with each share:

[0363] In the incorporated session key example described herein, the session master key is split into “n” shares according to the contents of the installation dependant Parser Master Key (SK1, SK2, SK3, SKn):

[0364] Depository 1: SD1, Kn, SK1

[0365] Depository 2: SD2, K1, SK2

[0366] Depository 3: SD3, K2, SK3

[0367] Depository n: SDn, K3, SKn.

Ex. 1008 [Orsini], [0363]-[0367].

Accordingly, all four shares are necessary to reconstruct the session master key. Ex. 2031 [Malek-Decl.] ¶ 108; *see also* Ex. 1008 [Orsini] [0368] (“***Unless all four shares are retrieved, the data cannot be reassembled according to this example.***”) (emphasis added); *see also id.*, [0381]-[0382] (“In order to restore the original data, the following items may be utilized: 1. ***All*** shares or portions of the

session master key with the data and, instead, “store the session master key data in one or more separate key management table[s].” *See id.*, [0302], [0305], [0323], [0329]; *compare* [0054] (“FIG. 21 illustrates . . . stor[ing] of the encryption master key ***with the data***”) *with* [0055] (“FIG. 22 illustrates . . . storing the encryption master key ***separately from the data.***”).

data set.”) (emphasis added). Petitioner contends that a POSITA would be motivated to use a redundant cryptosplit in the Figure 21 embodiment because “providing such redundancy is beneficial in case one of the storage devices fails.” Pet., 84 (citing Ex. 1003 [Torre]). But this argument is fatally undermined by Orsini’s disclosure that its Figure 21 embodiment cannot restore a data set from “less than all” of the shares. For the same reason, a POSITA could not have a reasonable (or any) expectation of success in trying to improve the Figure 21 embodiment by using redundant cryptosplitting—Orsini itself expressly discloses that it cannot be done.

C. Petitioner Fails To Demonstrate That Orsini Discloses Or Renders Obvious “Generating Data Splitting Information” (All Claims, Grounds V-VII).

Each of the independent claims requires a number of separate steps as limitations. Two of the limitations are “encrypting the data set based on an encryption key,” and “generating data splitting information . . . usable to determine into which of a plurality of shares . . . a unit of [the] encrypted data set will be placed.” Ex. 1001 [’802 Patent], cls. 1[B]-[C], 10[B]-[C], 19[D]-[E]. Petitioner contends that both elements are satisfied by Orsini alone, because Orsini’s session master key, it claims, serves simultaneously as the encryption key and as the “data splitting information.” Pet., 91–92. That reading fails: To establish that those

limitations are disclosed by Orsini alone, Petitioner is required to show that Orsini discloses them separately, as “arranged as in the claim.” *See* Sections III.C.1).

As a fallback, Petitioner asserts that Orsini “renders obvious” the limitation because a POSITA supposedly would have generated a second, independent sequence of random numbers to perform the split. Pet., 93. But Petitioner identifies no reason—other than hindsight based on the limitations of the ’802 Patent claims—why such a modification would have been made. *See* Section III.C.2.

1. Petitioner Fails To Demonstrate That Orsini Alone Discloses the Separate Steps Of “Encrypting The Data Set Based On An Encryption Key” And “Generating Data Splitting Information.”

Each of the claims recites a “method for securing a data set” comprising the separate steps of “encrypting the data set based on an encryption key to produce an encrypted data set” (Ex. 1001 [’802 Patent], cls. 1[B], 10[B], 19[D]); and “generating data splitting information, wherein the data splitting information is usable to determine into which of a plurality of shares of data a unit of data of the encrypted data set will be placed” (*id.*, cls. 1[C], 10[C], 19[E]). Petitioner contends that both of these limitations are disclosed by Orsini in a single feature. The Petition characterizes Orsini’s “session master key” as serving both as the encryption key used to encrypt the data set, and, at the same time, as the “data splitting information” that must be generated by the system.

Petitioner is incorrect. For a single reference like Orsini to invalidate a claim, it must either disclose the limitations “arranged as in the claim,” or the reference must “suggest[] some reason to modify it to obtain the claimed limitations.” *Adasa v. Avery Dennison Corp.*, 55 F.4th 900, 910 (Fed. Cir. 2022) (citing *CRFD Research, Inc. v. Matal*, 876 F.3d 1339, 1345 (Fed. Cir. 2017)). Orsini fails that test. It does not disclose the encryption key and data splitting information “arranged as in the claim.” The claim does not equate the encryption key and the data splitting key but rather requires two separate objects to perform these functions. But the Petition asserts Orsini only discloses a single object that, contrary to the structure of the claims, stands in for both limitations.

In text and figures that are not in Orsini, the ’802 Patent describes a process, very different from that disclosed in Orsini, for securing data by “parsing and splitting” it, storing the data in multiple locations, and restoring it when needed. *See* Ex. 1001 [’802] Figs. 30-35; 64:3-75:8. The process disclosed for the first time involves creating and storing both integrity information and redundancy information.

New text and figures in the ’802 Patent (but not in Orsini) disclose a process like that in the claims, using an encryption key to encrypt the data set and a separate splitting key to allocate the encrypted data into shares. These are separate steps, carried out using different keys or splitting information. For encryption, an

encryption key is used. *Id.*, 71:27-28. However, for data splitting “a hash [of the data set] may be used” *Id.* 71:39-42. As Dr. Malek explains:

The ‘802 Patent discloses that “a hash may be used” (e.g., as a function of the cipher feedback session key, as a function of any other suitable value) to determine a bit value at which to split each byte of data.” Ex. 1001 [’802 Patent], 71:64-67. “After a split point has been determined . . . , a determination may be made with regard to which data portions to append each of the left and right segments.” *Id.*, 72:8-12. Thus, for each byte of data, a hash function is used to determine at which bit to split the byte into two units. Then, the system must further determine into which share of a plurality of shares each resulting unit is appended to. To make this determination, the ’802 teaches:

[I]n one suitable approach, a table of all possible distributions (e.g., in the form of pairings of destinations for the left segment and for the right segment) may be created, whereby a destination share value for each of the left and right segment may be determined by using any suitable hash function on corresponding data For example, ***a hash function of a corresponding byte in the random or pseudo-random value may be made. The output of the hash function is used to determine which pairing of destinations*** (i.e., one for the left segment and one for the right segment) ***to select*** from the table of all the destination combinations. Based on this result, each segment of the split data unit is appended to the respective two shares indicated by the table value selected because of the hash function.

Id., 72:13-29; Ex. 2031 [Malek-Decl.] ¶ 115. None of this is disclosed in Orsini.

Accordingly, Petitioner’s “session master key” theory fails because Orsini does not disclose encrypting the data set using an encryption key separate from generating data splitting information using a hash. The encryption key and data splitting information cannot be the same thing which is all that Petitioner contends Orsini discloses.

2. Petitioner Fails To Demonstrate That Orsini Renders Obvious “Generating Data Splitting Information.”

Tacitly acknowledging the flaws in its theory, Petitioner shifts to an obviousness argument, asserting that “it would have been obvious to generate a sequence of random numbers—separate from the encryption key—for splitting the encrypted data.” Pet., 93; *see also* Ex. 1002 [Zadok-Decl.] ¶ 456. Petitioner contends that adding another key to Orsini would be a routine “modification” that “improves security” but provides no support for its claim that a POSITA would have been motivated to generate another key or that such a change would have been straightforward or beneficial. To the contrary, this is an entirely different scheme for securing the data. Orsini’s Figure 21 embodiment does not disclose the use of two different keys for encryption and splitting and does not disclose or suggest “generat[ing] a sequence of random numbers—separate from the encryption key—for splitting the encrypted data.” Pet., 93; *See* Ex. 2031 [Malek-Decl.] ¶ 118. Petitioner’s bare assertion that it does is precisely the kind of

conclusory reasoning that fails under *KSR* and *Magnum Oil Tools*. See *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 418 (2007) (citing *In re Kahn*, 441 F.3d 977, 988 (Fed. Cir. 2006)); *In re Magnum Oil Tools Int’l, Ltd.*, 829 F.3d 1364, 1380 (Fed. Cir. 2016).

Nor is there any reason a POSITA would view such a modification as desirable. Orsini teaches that its method provides a “***comprehensive package for data security***.” Ex. 1008 [Orsini], [0380] (“The data shares are then encrypted individually, and the keys are stored safely with different encrypted shares. When combined, the entire process for securing data according to the methods disclosed herein becomes a comprehensive package for data security.”). In that design, the same session master key both encrypts the data and defines the split pattern, so that only a single key must be secured. Petitioner’s claim that “using two separate keys instead of one improves security because an intruder would need to hack both keys to recover the original data” is unsupported, and implausible in the absence of any explanation about how the second key would be secured. See Pet., 93. In fact, generating a new key for splitting ***would decrease*** overall security provided by Orsini’s Figure 21 embodiment. As Dr. Malek explains:

Petitioner has not addressed how the additional key would be managed. A POSITA would not be motivated to add a cryptographic key without a clear key-management scheme, because even “small”

changes can introduce exploitable weaknesses.

If the new key were stored the same way as in Figure 21—where “the session master key [is stored] with the parsed data,” (Ex. 1008 [Orsini], [0294]), and each share’s key were likewise stored with the parsed data, *id.* [0299]—then compromising all shares would expose all keys, yielding no added security. If, instead, the keys were stored or transmitted separately, Petitioner has not explained where they would reside, how they would be exchanged, or how they would be protected. Either approach increases complexity without improving security and could weaken the overall system.

In practice, introducing a new key without a clear management framework adds complexity while creating new vulnerabilities rather than mitigating existing ones.

Ex. 2031 [Malek-Decl.] ¶¶ 120-122.

Petitioner’s suggestion that it would be obvious to add a second, independent key adds no benefit and introduces new risks. It is classic hindsight—starting from the claims and retrofitting the prior art to match. *See Univ. of Strathclyde*, 17 F.4th at 165; *KSR*, 550 U.S. at 418–19; *Ruiz*, 357 F.3d at 1275. Petitioner’s two-key proposal should be rejected.

D. Petitioner Fails To Demonstrate That A POSITA Would Be Motivated To Combine Orsini And Torre To Add “Creating Integrity Information Using The Data Set” And “Including In The Plurality Of Shares Data Indicative Of ... The Integrity Information” (All Claims, Grounds V-VII).

Even if, counterfactually, the combination of Orsini and Torre disclosed the claim limitations, Petitioner fails to show that a POSITA would be motivated to combine them. The claims require both “creating *integrity information* using the data set” and “including in the plurality of shares data indicative of . . . the *integrity information*.” Ex. 1001 [’802 Patent], cls. 1[A], 1[E], 10[A], 10[E], 19[C], 19[G].

Petitioner concedes that Orsini does not disclose “creating” the claimed integrity information, much less storing it in a plurality of shares. Pet., 85 (“Orsini does not expressly teach a mechanism to verify that the restored data set is identical to the original data set.”). During prosecution of the ’802 Patent, the Examiner reached the same conclusion, explicitly allowing the claims over Orsini because, among other reasons, it does *not* disclose generating integrity information

about a data set or including integrity information about the data set in its shares.

Ex. 1009 [Prosecution-History], 569-76 (Examiner's Reasons for Allowance).⁶

⁶ The Reasons for Allowance expressly distinguish the claims from Orsini, stating that: The present invention is directed to a method and system for preventing unauthorized access or use of sensitive data. Each independent claim identifies the uniquely distinct features of “creating integrity information using the data set; encrypting the data set based on an encryption key to produce an encrypted data set; generating data splitting information, wherein the data splitting information is usable to determine into which of a plurality of shares- of- data a unit of data of- the encrypted data set will be placed; separating the encrypted data set into the plurality of shares based on the data splitting information; including in the plurality of shares data indicative of (a) the encryption key and (b) the integrity information; and causing the plurality of shares to be stored in respective separate storage locations; wherein the data set is restorable by accessing less than all, but at least a threshold number of, the plurality of shares”. ***The closest prior art, Orsini et al (US 7,391,865), discloses a conventional system for securing sensitive data that fails to anticipate or render the above underlined limitations obvious.*** Ex. 1009 [Prosecution-History], 574-75 (emphasis added).

Instead, Petitioner argues that “a POSITA would have been motivated to incorporate, into Orsini, Torre’s teaching of computing a digital signature of input data and appending it to the encrypted data . . . so that Orsini’s system can use the digital signature to verify the integrity of the reconstructed data.” Pet., 88-89.

Petitioner is wrong.

1. Orsini Already Provides A Complete, Self-Verifying System.

Orsini’s Figure 21 provides a comprehensive—albeit different—system for encrypting, splitting, and reassembling data:

The data shares are then encrypted individually, and the keys are stored safely with different encrypted shares. When combined, the entire process for securing data according to the methods disclosed herein becomes a ***comprehensive package*** for data security.

Ex. 1008 [Orsini], [0380]. Orsini further makes clear that “[t]he data secured according to the methods of the present invention is readily retrievable and restored, reconstituted, reassembled, decrypted, or otherwise returned into its original or other suitable form for use.” *Id.*, [0381]. By its own terms, Orsini already provides integrity and verifiability—there is no function for a digital signature to fill.

Where, as here, a reference already achieves the purported benefit of the proposed modification, there is no motivation to add it. *See, e.g., In re NTP, Inc.*, 654 F.3d 1279, 1289 (Fed. Cir. 2011); *Apple Inc. v. Cellular Commc’ns*

Equipment, LLC, IPR2015-00576, Paper 7, 12 (June 12, 2015) (rejecting the petitioner’s argument that a combination of references rendered an invention obvious because Petitioner failed to provide a “specific explanation” of why a prior art reference didn’t accomplish that by itself.”); *F5 Networks, Inc. v. Radware, Ltd.*, IPR2017-00124, Paper 48, 64-65 (Apr. 23, 2018) (“Petitioner provides no persuasive reasons to modify [a prior art reference] to include the [components of another prior art reference]” and render Patent Owner’s invention obvious).

2. Orsini’s Figure 21 Embodiment Does Not Have The Integrity Problems Purportedly Addressed By Torre.

Petitioner premises its motivation-to-combine argument for Orsini/Torre on a fundamental misreading of Orsini. Torre explains that the need for integrity information exists in its system because errors can arise when redundant data is added to its shares, and the data set is subsequently reconstructed using that redundant data. Torre explains that it is the use of redundant data that creates the need to check the integrity of the reconstructed data set:

Implementations using signature generation allow[] for data integrity checking to verify the validity of reconstruction of the input data 139 from the shredded input data 156 including subsets of the shredded input data that has few[er] than the total number of shreds 156 found in the shredded input data, but enough, due to redundancy inherent with the shreds, to reconstruct the input data.

Ex. 1003 [Torre], [0086].

Orsini's Figure 21 embodiment, on which Petitioner solely relies for its Orsini/Torre combination, does not have this integrity problem because its shares do not include redundant data. Petitioner's motivation to combine fails because it relies on the incorrect assumption that Orsini's Figure 21 implements a redundant cryptosplit process in which redundant data is added to the shares "such that only a subset of the shares is needed to reassemble or restore the data." Pet., 83, 84 ("a POSITA would have understood that the redundant cryptosplit process may be used with Figure 21 to provide redundancy."). As explained in Section III.B, *supra*, the redundant cryptosplit is inapplicable to, and incompatible with, the Figure 21 embodiment, which requires all the shares to be retrieved to reconstruct the original data set. *See* Ex. 1008 [Orsini], [0294]-[0300], [0368], [0384]-[0395].

Because Orsini's Figure 21 embodiment does not include redundant information in its shares, there would be no motivation to modify it to add the integrity feature that Torre uses to address problems that arise from its use of redundant information in its shreds.

3. The Proposed Combination Would Offer No Benefit And Impose Significant Cost.

Even setting Petitioner's mistaken assumption aside, a POSITA would not be motivated to make the proposed modification—generating a signature for the entire original data set—because the admitted drawbacks of doing so outweigh any

hypothetical benefits. Torre acknowledges that including a digital signature of the original data set to the shreds would substantially decrease the speed and efficiency of its storage and retrieval system. Torre discloses that digital “signature generation function[s]” like SHA-1 are “resource intensive” because they “produce relatively large output” and require “a relatively large amount of processing capability.” Ex. 1003 [Torre], [0087]-[0088]. Torre therefore teaches an alternative approach in which its parallel “second transformers” generate signatures on the already-split shreds. *Id.*, [0087]. That approach does not produce “integrity information using the data set,” as the claims require, but instead operates on the shreds themselves. *See* Ex. 1001 [’802], cl. 1[A], 10[A], 19[C]. Torre thus counsels away from the very modification Petitioner proposes. Ex. 2031 [Malek-Decl.] ¶ 136.

A POSITA would not attempt to burden Orsini’s efficient, self-contained Figure 21 embodiment with Torre’s resource-intensive digital signature logic add-on—particularly when Orsini tells him or her it will not work. Adding Torre’s digital-signature processing would consume resources without improving functionality. *See Polaris Indus.*, 795 Fed. App’x. at 833 (“The Board must weigh the benefits and drawbacks of the modification against each other, to determine whether there would be a motivation to combine.”); *Henny Penny Corp.*, 938 F.3d at 1331-32 (similar); *Bombardier Rec. Prods.*, 876 F.3d at 1363 (“The potential

problems” with combination supported finding that “a skilled artisan would not have been motivated” to make combination).

4. Petitioner Fails To Explain How Torre’s Asymmetric Encryption Keys Are Created In Orsini/Torre.

Petitioner contends that its combination implements Torre’s SHA-1 electronic signature for integrity verification in Orsini’s system, (Pet., 87), which uses asymmetric encryption, but Orsini’s Figure 21 only discloses the use of a “session master key,” (Ex. 1008 [Orsini] [0054]), which necessitates the use of symmetric encryption. As Dr. Malek explains:

Asymmetric and symmetric encryption are fundamentally different from one other. In the case of asymmetric encryption, there are two keys: a private key and a public key. Data encrypted using the private key can be decrypted using the public key, and vice versa. This property is used to create electronic signatures, such as Torre’s SHA-1 electronic signature. SHA-1 electronic signature is computed in two steps. First the Secure Hash Algorithm 1 (SHA-1) is used to compute the hash of certain data (e.g., message, file). Second, the resulting hash is encrypted using the private key of the entity signing the data. Private keys are kept a secret, known only by the signer, while the public key is made available publicly.

The integrity of the data carrying an electronic signature can be verified by a recipient in two steps. First, the recipient decrypts the electronic signature using the public key of the signer to get back the SHA-1 hash of data. Second, the recipient computes the hash of the

data (using the same SHA-1 algorithm) and compares it with the hash obtained through decryption. If the two hash values match, the recipient can be assured the received data is coming from the signer and that it has not been modified. In contrast, a session key is a form of symmetric encryption, where a *single* secret key is used for both encrypting and decrypting data, such as the session master key in Figure 21 of Orsini. The parties using a session key for communication must keep the key a secret, because anyone with access to the key can decrypt and read the data.

The two encryption methods present distinct challenges and therefore must be managed differently. Because in symmetric encryption the same secret key is used for both encryption and decryption, the key must remain confidential at all times. To ensure this, organizations typically rely on external key management systems (KMS) or hardware security modules (HSMs) to securely generate, store, and rotate symmetric keys. In contrast, asymmetric encryption uses a pair of mathematically related keys: a public key that can be distributed openly and a private key that must be kept secret. And because public keys must be trusted as genuine, asymmetric systems employ a public key infrastructure (PKI) that uses digital certificates and certificate authorities to validate the authenticity of public keys.

Another key difference has to do with the performance tradeoffs. Symmetric algorithms, such as AES, are computationally efficient and therefore well-suited for encrypting large volumes of data. Asymmetric algorithms, such as RSA, are considerably slower and are typically used

for secure key exchange or digital signatures rather than bulk data encryption. Another significant difference lies in scalability: symmetric encryption requires a unique shared key for every pair of communicating parties, whereas asymmetric encryption scales more easily since each participant only needs one key pair.

Ex. 2031 [Malek-Decl.], ¶¶ 139-142.

Petitioner ignores the fundamental differences between these two encryption methods and fails to explain how the keys—*i.e.*, both a private and a public key—necessary to create Torre’s SHA-1 digital signature are created, managed, or used in the proposed Orsini-Torre combination.

5. Petitioner’s Theory Is Driven By Hindsight.

Finally, Petitioner’s rationale is pure hindsight. Petitioner explicitly uses the ’802 Patent itself as a roadmap, asserting that Torre’s teaching “is consistent with the ’802 Patent which also states that ‘[e]xamples of integrity information may include . . . digital signature information.’” Pet., 32-33 (citing Ex. 1001 [’802], 64:30-35). But the inventor’s own path cannot supply a motivation to combine. *Univ. of Strathclyde*, 17 F.4th at 165 (quoting *Otsuka Pharm. Co. v. Sandoz, Inc.*, 678 F.3d 1280, 1296 (Fed. Cir. 2012)); *InTouch Techs., Inc. v. VGo Commc’ns*, 751 F.3d at 1351. Petitioner’s theory thus merely retraces the inventor’s own path, using the claims as a blueprint to retrofit Orsini and Torre—exactly the sort of

hindsight reasoning the Federal Circuit has repeatedly condemned. *Univ. of Strathclyde*, 17 F.4th at 165; *Ruiz*, 357 F.3d at 1275.

In sum, Petitioner's proposed Orsini/Torre combination simply would not be pursued by a POSITA and has no other basis than hindsight. It should be rejected.

V. CONCLUSION

Petitioner fails to demonstrate a reasonable likelihood of success as to any of the challenged claims, and *inter partes* review should be denied.

Respectfully submitted,

/s/ Stephen J. Elliott
Stephen J. Elliott (Reg. No. 52,858)
Andrei Iancu (Reg. No. 41,862)
SULLIVAN & CROMWELL LLP

Kenneth J. Weatherwax (Reg. No. 54,528)
Nathan Lowenstein, *pro hac vice* pending
Colette Woo, *pro hac vice* pending
LOWENSTEIN & WEATHERWAX LLP

Date: October 17, 2025

CERTIFICATE OF COMPLIANCE WITH TYPE-VOLUME LIMITS

This Patent Owner Preliminary Response (the “POPR”) consists of 13,882 words, excluding table of contents, table of authorities, certificate of service, this certificate, or table of exhibits. The POPR complies with the type-volume limitation of 14,000 words as mandated in 37 C.F.R. § 42.24. In preparing this certificate, counsel has relied on the word count of the word-processing system used to prepare the paper (Microsoft Word).

Respectfully submitted,

/s/ Stephen J. Elliott

Date: October 17, 2025

CERTIFICATE OF SERVICE

The undersigned hereby certifies that the following documents were served
by electronic service, by agreement between the parties, on the date below:

PATENT OWNER PRELIMINARY RESPONSE

EXHIBIT 2031

The names and addresses of the parties being served are as follows:

Taeg Sang Cho	tcho@desmaraisllp.com
Kurt Fredrickson	kfredrickson@desmaraisllp.com
Lindsey Miller	lmiller@desmaraisllp.com
Laura Avena	lavena@desmaraisllp.com
	IBM-SFI-IPR-Service@desmaraisllp.com

Respectfully submitted,

/s/ Stephen J. Elliott

Date: October 17, 2025