

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

EXHIBIT 2031

DECLARATION OF SAM MALEK, PH.D.

TABLE OF CONTENTS

	Page
I. INTRODUCTION	1
II. QUALIFICATIONS	1
III. BASES OF OPINIONS	6
IV. APPLICABLE LEGAL STANDARDS	8
A. Level of Ordinary Skill in the Art	8
B. My Understanding Of Legal Issues.....	10
V. CLAIMS AT ISSUE (CLAIMS 1-27).....	12
VI. OPINIONS.....	13
A. Overview Of The '802 Patent, The Challenged Claims, And Prior Art References.....	13
1. The '802 And The Challenged Claims.....	13
2. Overview Of Torre	15
3. Overview Of Tajima.....	18
4. Overview Of Orsini.....	20
VII. PETITIONER'S TORRE-TAJIMA GROUNDS.....	23
A. The Petition 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).	23
1. Petitioner Fails To Demonstrate That "Torre/Tajima System-1" Teaches "Generating Data Splitting Information" (Torre/Tajima System 1, Limitation 1[C]/9[C]/19[D]).	25
2. Petitioner fails to Show that Torre/Tajima System-2 Teaches "Data Splitting Information" that is "Usable to Determine Into Which of a Plurality of Shares of Data ... Will Be Placed" (Torre-Tajima System 2, Limitation 1[C]/9[C]/19[D]).	29

3. 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” (Limitations 1[B], 10[B], 19[D]).33

B. Petitioner’s Orsini Grounds Fail (All Claims, Grounds V-VII).....41

1. The Claims of the ’802 Patent Are Supported by the ’802 Patent’s New Disclosures that Are Not in Orsini.....41

2. Petitioner Also Fails To Demonstrate That Orsini/Torre Discloses Or Renders Obvious Restoring The Data Set From “Less Than All” Of The Shares (Limitation 1[G]/9[G]/19[H]).45

3. Petitioner Fails to Demonstrate that Orsini Discloses or Renders Obvious “Generating Data Splitting Information” (Limitation 1[C]/9[C]/19[D]).49

4. 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” (Limitations 1[A]/9[A]/19[C], 1[E]/9[E]/19[G]).55

VIII. CONCLUSION 62

I. INTRODUCTION

1. My name is Sam Malek. I have been retained as an expert witness to provide my independent opinion in regards to the matters at issue in inter partes review of U.S. Patent No. 8,271,802 (“the ’802 Patent”) in IPR2025-01200. I have been retained by Patent Owner Security First Innovations, LLC (“SFI” or “Patent Owner”), in the above proceedings. Petitioner is International Business Machines Corporation (“IBM” or “Petitioner”).

2. I am being compensated \$800 per hour for my time spent working in connection with this case. My compensation is in no way related to the outcome of this litigation. If called as a witness, I would testify as to the statements and opinions contained in this report.

3. I am not a legal expert and offer no opinions on the law. However, I have been informed by counsel of the various legal standards that apply, and I have applied those standards in arriving at my conclusions.

II. QUALIFICATIONS

4. I have more than 25 years of experience in the field of computer science and software engineering. My professional career has been dedicated to issues relating to software design and architecture, including cloud computing architectures, distributed software systems, and software security.

5. I received a Bachelor of Science in Information and Computer Science from the University of California, Irvine in 2000, a Master of Science in Computer Science from the University of Southern California in 2004, and a Ph.D. in Computer Science from the University of Southern California in 2007. My Ph.D. dissertation, titled “A User-Centric Approach for Improving a Distributed Software System’s Deployment Architecture,” concerned distributed software systems and specifically resulted in new algorithms and an architectural middleware for effective deployment of a component-based software system among a distributed set of computing nodes.

6. I am a Professor of Software Engineering in the School of Information and Computer Sciences at the University of California, Irvine (“UCI”). I joined the Department of Informatics at UCI as an assistant professor in 2015 and was promoted to full professor with tenure in 2019. As a member of the faculty, I perform research, teach graduate and undergraduate courses in software architecture, design, and engineering, and supervise the research of postdoctoral associates and other students. I have taught software engineering and computer science courses in which I frequently cover topics related to the technologies at issue in this case, including data storage, distributed systems, software security, encryption authentication, and verification.

7. Before my role at UCI, I taught at George Mason University, where I was an Associate Professor in the Computer Science Department from 2007 to 2013.

Prior to that, I spent four years as a researcher at the University of Southern California, also focusing on software engineering and computer science.

8. Outside of teaching, I have held several leadership roles at various labs and research centers. For example, I currently serve as Director of the Software Engineering and Analysis Laboratory (“SEAL”) at UCI. SEAL is a lab engaged in research to automate software engineering activities. In particular, the lab focuses on the development of techniques and tools that aid with the construction, analysis, and maintenance of large-scale and dependable software systems. Additionally, from 2018 to 2022, I served as Director of the Institute for Software Research at UCI. The software systems and results produced in these research laboratories have been adopted for use by various private and public-sector organizations, including Boeing, Robert Bosch GmbH, the National Aeronautics and Space Administration (“NASA”), Department of Homeland Security (“DHS”), and the Federal Bureau of Investigations.

9. I have also served as the principal investigator of several federally funded research projects totaling more than 12 million dollars. For example, I led a team of researchers in a project sponsored by the United States Defense Advanced Research Projects Agency (“DARPA”) to develop a novel approach for protecting Android applications from Inter-Process Communication (“IPC”) security attacks. In another project sponsored by the Air Force Office of Scientific Research, I

developed a framework that monitors, organizes, and dynamically adapts distributed systems to ensure they continue operating effectively, even under conditions of impairment. A central focus of this project required ensuring the secure storage of mission-critical data in cloud environments. I also worked with NASA to develop a new distributed software system for sharing large amounts of data generated by NASA.

10. I have significant industry experience as a software engineer, software architect, and programmer at companies such as the Boeing Company (2005 to 2007), PriceWaterhouseCoopers Consulting (later acquired by IBM)(2000 to 2002), FieldCentrix (1999 to 2000), and Neural Computing Systems Lab (1998 to 1999). During my time in industry, I gained experience in the design, development, and management of large-scale software engineering efforts, including the development of systems that securely stored and retrieved large volumes of data in distributed environments. For example, while at Boeing, I was part of a team that designed autonomous aerial platforms, which were designed to provide real-time data transmission to remote operators and background servers. I also contributed to the development of solutions for securely transmitting and storing the data generated by such platforms on cloud-like military systems.

11. I am the author and co-author of over 100 journal articles and book chapters. These publications have been cited more than 12,000 times according to

Google Scholar (scholar.google.com). Many of those publications relate to security and software architecture, including: “Forecasting Architectural Decay from Evolutionary History,” *IEEE Transactions on Software Engineering (TSE)*, Vol. 32, No. 3, April 2023; *Security and Software Engineering in Handbook of Software Engineering*, Editors Kyo Chul Kang, Richard Taylor, and Sungdeok Cha, 2019; and “Introduction to Special Issue on the State of Art in Engineering Self-Adaptive Software Systems,” *Journal of Systems and Software (JSS)*, Vol. 85, No. 12, pages 2675-2677, December 2012.

12. I serve, or have served, as chair, committee member, or reviewer for more than 150 software engineering journals, magazines, and conferences. For example, I have served as associate editor of the *Institute of Electrical and Electronics Engineers (“IEEE”) Transactions on Software Engineering and Methodology*, *Association for Computing Machinery (“ACM”) Transactions on Autonomous and Adaptive Systems*, and the *Springer Journal of Computing*. I am currently the Deputy Editor-in-Chief of the *Springer Journal of Automated Software Engineering*, a role in which I have refereed several articles submitted by the scientific community that deal with topics related to the technologies at issue in this case, including cloud storage, data security, and encryption.

13. I have received numerous awards for my work and research in the field of software architecture and computer science, including: The ACM SIGSOFT

Distinguished Paper Award at the International Conference of Software Engineering (2025); the Test of Time Award from the Association for Computing Machinery (2020); the CAREER award from the National Science Foundation (2013); the Emerging Researcher/Scholar/Creator Award from George Mason University (2013); the Outstanding Faculty Research Award from George Mason University (2011).

14. I have spoken on information security and software architecture issues at more than 60 scientific conferences, workshops, and symposia. For example, in 2020, I was invited by the Association for Computing Machinery to give a presentation entitled “The Threat in Your Pocket: Trends, Challenges, and Solutions in Mobile Application Security.” I also presented talks such as “Tools for Automated Detection and Assessment of Security Vulnerabilities in Mobile Applications” to the DHS’s Cyber Security Division R&D Showcase and Technical Workshop.

15. A detailed record of my professional qualifications is set forth in the attached Exhibit A, which is my curriculum vitae, including a list of publications, awards, research grants, and professional activities. My curriculum vitae also lists the matters in which I have served as an expert.

III. BASES OF OPINIONS

16. In the course of conducting my analysis and forming my opinions, I have reviewed materials including those listed below:

- i. U.S. Patent No. 8,271,802 (Ex. 1001) (“’802 Patent”);
- ii. The Declaration signed by Erez Zadok, Ph.D. (Ex. 1002) (the “Zadok Declaration”);
- iii. U.S. Patent Publication No. 2003/0065656 (Ex. 1003) (“Torre”);
- iv. U.S. Patent Publication No. 2003/0028493 (Ex. 1004) (“Tajima”);
- v. U.S. Patent Publication No. 2004/0030921 (Ex. 1005) (“Aldridge”);
- vi. “Secret Sharing Made Short,” Hugo Krawczyk, *Advances in Cryptology - CRYPTO ’93*, 13th Annual International Cryptology Conference, LNCS 773, 136-146, 1993 (Ex. 1006) (“Krawczyk”);
- vii. U.S. Patent Publication No. 2003/0200176 (Ex. 1007) (“Foster”);
- viii. U.S. Patent Publication No. 2004/0049687 (Ex. 1008) (“Orsini”);
- ix. File History of U.S. Patent No. 8,271,802 (Ex. 1009) (“’802 File History”);
- x. File History of U.S. Patent Application No. 11/258,839 (Ex. 1010) (“’839 App. File History”);
- xi. File History of U.S. Patent Application No. 13/371,363 (Ex. 1011) (“’363 App. File History”);
- xii. Comparison Of The ’802 Patent’s Specification And Orsini (Ex. 1012);
- xiii. *Google, LLC v. Security First Innovations, LLC*, IPR2024-00212, Exhibit 1043 (Patent Owner’s Proposed Claim Constructions In *Security First Innovations, LLC v. Google, LLC*, No. 2:23-cv-00097 (E.D. Va.)) (“PO’s Google Litigation Construction”);
- xiv. U.S. Provisional Application No. 60/622,146 (Ex. 1014) (“Provisional Application No. 60/622,146”);
- xv. U.S. Provisional Application No. 60/718,185 (Ex. 1015) (“Provisional Application No. 60/718,185”);

- xvi. Microsoft Computer Dictionary, (5th ed., 2002), excerpts (Ex. 1016) (“Microsoft Computer Dictionary”);
- xvii. Bruce Schneier, APPLIED CRYPTOGRAPHY, (2d ed., 1996), excerpts (Ex. 1017) (“Schneier”);
- xviii. Curriculum Vitae of Dr. Erez Zadok (Ex. 1018);
- xix. Yitzhak Birk, Random RAIDs with Selective Exploitation of Redundancy for High Performance Video Servers, IEEE 1997 (Ex. 1019) (“Birk”);
- xx. U.S. Patent Publication No. 2003/0016596 (Ex. 1020) (“Chiquoine”);
- xxi. John Kubiawicz, et al., OceanStore: An Architecture for Global-Scale Persistent Storage, ACM 2000 (Ex. 1022) (“Kubiawicz”);
- xxii. U.S. Patent No. 6,226,618 (Ex. 1023) (“Downs”);
- xxiii. P. Venkat Rangan et al., *Designing File Systems for Digital Video and Audio*, ACM SIGOPS Operating Systems Review, Vol. 25, Issue 5, 81–94 (1991) (Ex. 1024) (“Rangan”);
- xxiv. The exhibits and other documents cited herein.

IV. APPLICABLE LEGAL STANDARDS

A. Level of Ordinary Skill in the Art

17. My opinions in this declaration are based on the understandings of a person of ordinary skill in the art, which I understand is sometimes referred to as an “ordinary artisan” or by the acronyms “POSITA” or “PHOSITA,” as of the time of the invention, which I understand for this IPR is assumed to be the effective filing date (“October 25, 2004”) of the provisional application from which the ’802 patent issued. I understand that the person of ordinary skill in the art is a hypothetical

person who is presumed to have known the relevant art at the time of the invention. By “relevant,” I mean relevant to the challenged claims of the ’802 patent.

18. I understand that the factual indicators of the level of ordinary skill in the art include the various prior art approaches employed, the types of problems encountered in the art, the rapidity with which innovations are made, the sophistication of the technology involved, and the educational background of those actively working in the field. I understand that, in assessing the level of skill of a person of ordinary skill in the art, one should consider the type of problems encountered in the art, the prior solutions to those problems found in the prior art references, the rapidity in which innovations are made, the sophistication of the technology, the level of education of active workers in the field, and my own experience working with those of skill in the art at the time of the invention.

19. In this case, Dr. Zadok has asserted in his declaration that a person of ordinary skill in the art at the time of the ’802 patent would have had “a Bachelor’s degree in Computer Science, Computer Engineering, Electrical Engineering, or an equivalent field, and about 2–3 years of experience in the fields of data storage and security.” Ex. 1002 [Zadok-Decl.], ¶ 155.

20. For the purposes of this declaration, I accept Dr. Zadok’s proposed qualifications of a POSITA. I reserve the right to revisit the issue should the Petition be instituted.

21. As further discussed below, my opinions as stated in this declaration are valid even if the Board adopts a slightly different level of ordinary skill in the art. For example, as will be discussed throughout my report, even a person with the level of knowledge or experience described by Dr. Zadok or adopted by the Board would not have a reasonable expectation of success in implementing certain aspects of the proposed combinations as of the priority date of the '802 Patent.

B. My Understanding Of Legal Issues

22. When considering the '802 patent and stating my opinions, I rely on the following legal standards as described to me by the attorneys for SFI.

23. I understand that a patent claim is unpatentable if the claimed invention was anticipated by a prior art reference or would have been obvious to a person of ordinary skill in the art at the time of the purported invention.

24. I understand that anticipation requires that every limitation of the claim at issue be disclosed, either expressly or inherently, in a single prior art reference.

25. I understand that an obviousness analysis involves comparing a claim to the prior art to determine whether the claimed invention would have been obvious to a person of ordinary skill in the art at the time of the invention in view of the asserted prior art references and in light of the general knowledge in the art as a whole. I also understand that obviousness is ultimately a legal conclusion based on underlying facts of four general types, all of which must be considered: (1) the scope

and content of the prior art; (2) the level of ordinary skill in the art; (3) the differences between the claimed invention and the prior art; and (4) any objective indicia of non-obviousness, including any praise of the invention.

26. I also understand that obviousness may be established under certain circumstances by combining or modifying the teachings of the prior art. Specific teachings, suggestions, or motivations to combine any first prior art reference with a second prior art reference can be explicit or implicit, but must have existed before the date of purported invention. I understand that prior art references themselves may be one source of a specific teaching or suggestion to combine features of the prior art, but that suggestions or motivations to combine art may come with the knowledge that a person of ordinary skill in the art would have had.

27. I understand that a reference may be relied upon for all that it teaches, including uses beyond its primary purpose but also including teachings that lead away from the invention. I understand that a reference may be said to teach away when a person of ordinary skill in the art, upon reading the reference, would be discouraged from following the path set out in the reference, although the mere disclosure of alternative designs does not teach away.

28. I further understand that whether there is a reasonable expectation of success in combining references in a particular way is also relevant to the analysis.

29. I understand that it is improper to use hindsight to combine references or elements of references to reconstruct the invention using the claims as a guide. My analysis of the prior art is made from the perspective of a person of ordinary skill in the art at the time of the invention.

30. I am not offering any legal opinions in this declaration nor am I qualified to do so. I only consider such legal standards in framing my opinions and conclusions as well as placing assertions made by Petitioner in the Petition into the proper context. Additionally, from a subject matter perspective, I understand that the petitioner always has the burden of persuasion regarding a challenge of patentability of an invention under inter partes review.

V. CLAIMS AT ISSUE (CLAIMS 1-27)

31. I understand that Petitioner has challenged the claims of the '802 Patent as follows:

- Independent claim 1 and its dependent claims 2, 3, 5, 9; independent claim 10 and its dependent claims 11-12, 14, 18; and independent claim 19 and its dependent claims 20-21, 23, 27 based on obviousness over Torre and Tajima (Ground I).
- Dependent claims 4, 13, 22 based on obviousness over Torre, Tajima, and Aldridge (Ground II).
- Dependent claims 6, 15, 24 based on obviousness over Torre, Tajima, and Krawczyk (Ground III).

- Dependent claims 7-8, 16-17, 25-26 based on obviousness over Torre, Tajima, and Foster (Ground IV).
- Independent claim 1 and its dependent claims 2-5, 9; independent claim 10 and its dependent claims 11-14, 18; and independent claim 19 and its dependent claims 20-23, 27 based on obviousness over Orsini and Torre (Ground V).
- Dependent claims 6, 15, 14 based on obviousness over Orsini, Torre, and Krawczyk (Ground VI).
- Dependent claims 7-8, 16-17, 25-26 based on obviousness over Orsini, Torre, and Foster (Ground VII).

VI. OPINIONS

A. Overview Of The '802 Patent, The Challenged Claims, And Prior Art References

1. The '802 And The Challenged Claims

32. 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], 2:30-35. An important feature of the method is that redundant 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.

33. 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.

34. 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 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.*, cls. 1[E], 10[E], 19[G].

35. 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 of the data set. . . using fewer than all the data portions”).

36. 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.

2. Overview Of Torre

37. In my opinion, Torre addresses different issues in a different way than the '802 Patent. Torre discloses a “shredding and deshredding system for storage and retrieval” that includes the use of redundancy to improve fault tolerance. Ex. 1003 [Torre], [0007], [0111]. This system differs from the '802 Patent in the way it encrypts, parses, stores and retrieves data. Torre also discloses a different

storage strategy, and is designed to achieve different goals, than the '802 Patent, as I explain below.

38. 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 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.

39. In my opinion, Torre also has different goals than the '802 Patent: while Torre emphasizes fault tolerance and efficiency, the '802 Patent is focused on data security and integrity. 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.

40. As part of its fault-tolerant system, Torre discloses that various optional data transformations can be implemented to “emphasiz[e] considerations such as redundancy, integrity, and security.” Ex. 1003 [Torre], [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:

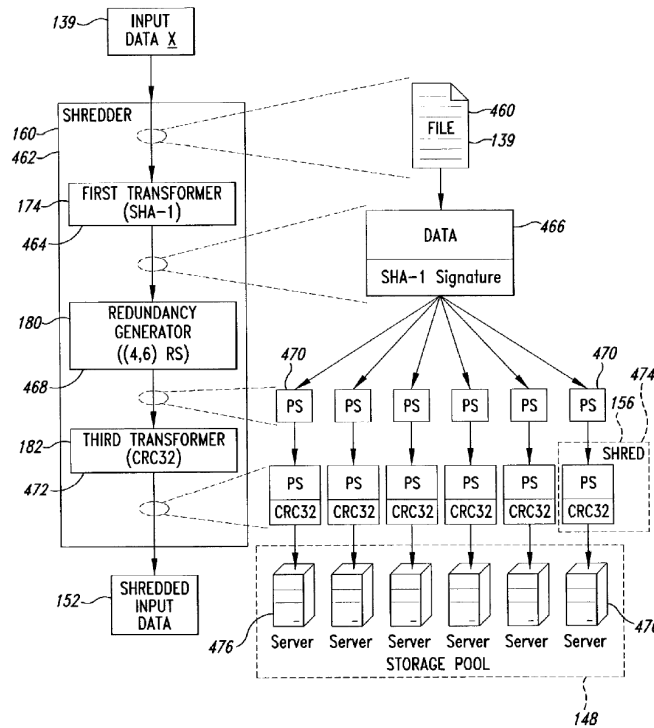


Fig. 32

Ex. 1003 [Torre] Fig. 32.

41. Unlike the '802 Patent, however, each transformer in Torre’s shredder also appends a tag to the transformed data or data shred to “help keep track” of the data “as the pieces . . . move through the shredder” *Id.*, [0078]. Each tag

contains “[i]nformation describing how the shreds were created” such that later, Torre’s deshredder can use the tag to identify the proper inverse transformation, which could be “decompression, decryption, or signature verification.” *Id.*, [0054], [0123]. Unlike the ’802 Patent, Torre’s deshredder uses these tags to guide properly “reassembl[y of] the original input data.” *Id.*, [0053].

42. 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 how to use a block cipher to encrypt a shred of any length or the “type of cipher” that identified the encryption algorithm 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 id.* [0084], [0085]. Torre recommends specific encryption techniques that are compatible with its use of tags and require a separate, external key management system T

43. 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], cls. 1, 10, 19.

3. Overview Of Tajima

44. 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]. It discloses a system and method for securing personal shopping information by dividing the personal information into “data portions” or “data fragments” and storing under “separate” control by different entities. *Id.*, [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].

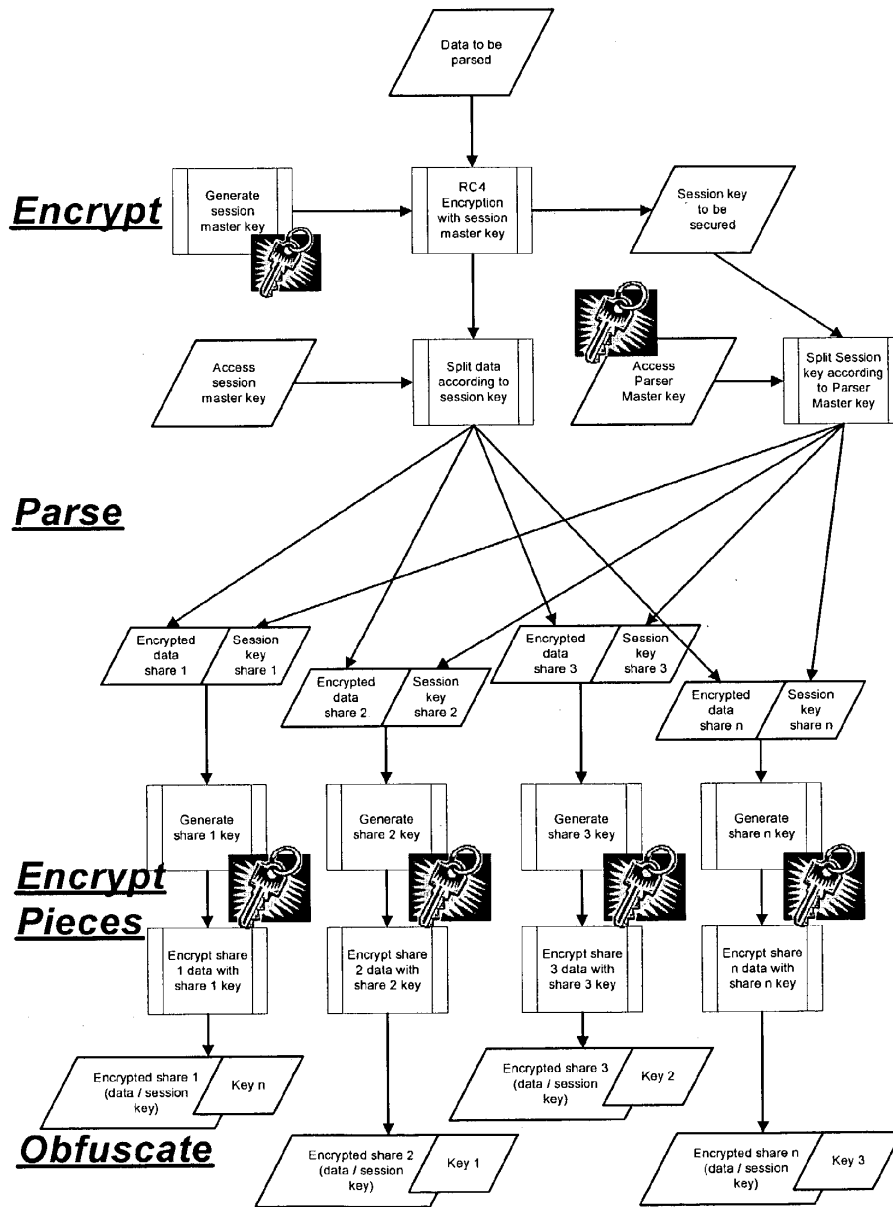
45. This is fundamentally different from the ’802 Patent, because the data security afforded by the ’802 Patent’s system does not depend on separating data among “separate” control, like in Tajima. *See* Ex. 1001 [’802 Patent], cls. 1, 10, 19; Ex. 1004 [Tajima], [0048]. The ’802 Patent does not parse data in that fashion. Instead, it 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.*, cls. 1, 10, 19

4. Overview Of Orsini

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

Figure 21



Ex. 1008 [Orsini], Fig. 21.

47. 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.*, [0297-98]), 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].

48. I understand that while the ’802 Patent shares several disclosures with Orsini, it also introduces new disclosures and figures and recites claims based on new material. Ex. 1001 [’802 Patent] Figs. 30-35; 69:39-75:8. In my opinion, these additions, which are not disclosed by Orsini, 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. In my opinion, Orsini does not disclose these limitations.

VII. PETITIONER’S TORRE-TAJIMA GROUNDS

A. The Petition 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).

49. All of the independent claims require:

- Limitation 1[C]: “generating data splitting information, wherein the data splitting information is usable to determine into which a plurality of shares of data a unit of data of the encrypted data set will be placed.”
Ex. 1001 [’802 Patent], cl. 1[C].
- Limitation 10[C]: “generating data splitting information, wherein the data splitting information is usable to determine into which a plurality of shares of data a unit of data of the encrypted data set will be placed.”
Ex. 1001 [’802 Patent], cl. 10[C].
- Limitation 19[C] “generating data splitting information, wherein the data splitting information is usable to determine into which a plurality of shares of data a unit of data of the encrypted data set will be placed.”
Ex. 1001 [’802 Patent], cl. 19[C].

50. I understand that for all of its Torre and Tajima grounds, Petitioner relies upon a combination of Torre and Tajima for this limitation, proposing two alternative systems—Torre/Tajima System-1 (“System-1) and Torre/Tajima System-2 (“System-2”). Pet., 36, 43; Ex 1002 [Zadok-Decl.] ¶¶ 185-92.. In my opinion,

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.

51. I understand that Petitioner's Torre/Tajima System-1 relies on Torre's "selection information" as corresponding to the claimed "data-splitting information." Pet., 36-39. In my opinion, that theory fails because the Petition does not show that Torre *generates* "selection information," and 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 VI.B.1.

52. I understand that Petitioner's Torre/Tajima System-2 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. In my opinion, Petitioner 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 VI.B.2.

1. Petitioner Fails To Demonstrate That “Torre/Tajima System-1” Teaches “Generating Data Splitting Information” (Torre/Tajima System 1, Limitation 1[C]/9[C]/19[D]).

53. I understand that Petitioner relies solely on Torre to disclose this limitation in Torre/Tajima System 1. Petitioner asserts that Torre discloses “generating data splitting information” through its references to “remapping” and “partitioning” operations. *See* Pet., 36. I understand that Petitioner also argues that it would have either been obvious to generate such information. I understand that Petitioner also argues that Torre’s disclosure of a Copy-N redundancy scheme discloses the limitation. *See id.*

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

55. In my opinion, this is incorrect. 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 id.*, 36-37. Petitioner’s contention that Torre “generat[es]” “selection information” appears to be entirely

based on Torre's disclosure that "specific remapping is dependent upon particular situations," and Dr. Zadok's conclusion that "[w]ithout . . . selection information, it would not have been possible to remap data." See *id.* (quoting Ex. 1003 [Torre], [0101]). In my opinion, Petitioner's expert is wrong because "selection information" is not data splitting information." 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].

56. Even assuming that Torre's "selection information" did correspond to the "data splitting information," in my opinion Torre does not suggest that selection information is "generated" by Torre's system as a data set that is processed for storage. To the contrary, Torre indicates that selection information pre-exists in the system. For example, Torre discloses that the "interleaving" functionality should be part of the system's pre-existing hardware. Ex. 1003 [Torre], [0100]. Petitioner attempts to analogize "selection information" to a "case statement" used to write computer programs, (Pet., 37), but its analogy merely confirms that "selection

information” is pre-existing and not generated. The Microsoft Computing Dictionary’s definition of “case statement,” on which Petitioner relies, defines it as comprising the execution different instructions based on constants “assigned by the programmer.” Ex. 1016 [Microsoft-Computer-Dictionary], 87. But in my opinion, 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 pre-existing, they are not “generat[ed].”

57. In my opinion, Petitioner’s obviousness theory also fails. I understand that Petitioner 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 the claim that “mode switching may be performed by generating selection information.” *Id.* (citing Ex. 1016 [Microsoft], 87). As discussed above, however, I believe this definition shows the opposite—that the constants in a “case statement” are “assigned by the programmer,” not “generated.” Ex. 1016 [Microsoft], 87.

58. 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 in my opinion, 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).

59. 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” that are then appended “with the data blocks 286.” Ex. 1003 [Torre] [0103].

60. Torre makes clear that tags do not prospectively determine future processing. Instead, they merely record completed operations. *Id.*, [0078] (“the tags 110 help keep track of *what [h]as been done* to the pieces of the input data as the pieces . . . move through the shredder.”); 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.”). Torre indicates that its 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]. All of these are retrospective operations.

61. In my opinion generating data-splitting information for encrypted data as required by the claims is a core aspect of a complex cryptographic process that

is lacking in the prior art on which petitioner relies, not an “unusually simple” feature.

2. Petitioner fails to Show that Torre/Tajima System-2 Teaches “Data Splitting Information” that is “Usable to Determine Into Which of a Plurality of Shares of Data ... Will Be Placed” (Torre-Tajima System 2, Limitation 1[C]/9[C]/19[D]).

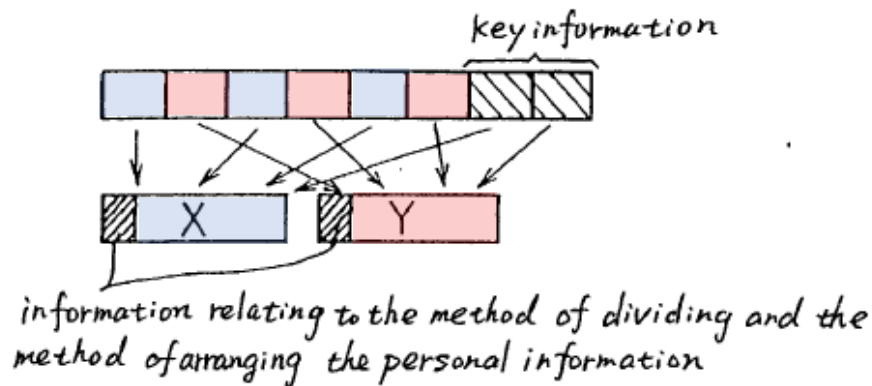
62. In System-2, Petitioner relies on Tajima’s method of splitting data, which divides a data set into portions of arbitrary length and then allocates each portion to a fragment in a “round-robin” fashion. Pet., 53. Petitioner asserts that the relevant process is Tajima’s method of dividing personal information. Pet., 40-42. Petitioner contends that Tajima’s “arbitrary-length” parameter constitutes the claimed “data-splitting information.” *Id.*, 24. 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.*

63. I understand that Petitioner relies on 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. In my opinion, Petitioner’s theory 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 lengths” is not data splitting information, and the information appended to 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.

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

Fig. 11



Ex. 1004 [Tajima], Fig. 11 (annotated).

65. Petitioner never identifies what the claimed “unit of data” is in System-2. In my opinion, the most natural reading is that the arbitrary-length portions themselves are the alleged “unit[s] of data.” 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. Thus, the calculated arbitrary lengths are not “usable to determine into which of a plurality of shares a unit of data ... will be placed.”

66. A POSITA would understand that 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. *Id.*, [0107]. This retrospective 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. 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.

67. I understand that Petitioner also 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. I also understand that Petitioner claims 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 “each shred . . . includes a unit of data from the corresponding data block. Pet., 39. I understand that Petitioner appears to contend that the decision to use Copy-N redundancy constitutes the required “data splitting information.” Pet., 39-40.

68. In my opinion, Petitioner’s theory is unsupported, and assumes that the system disclosed in Torre’s Figure 34 would be able to use a Copy-N redundancy transformation. Pet., 12-13. But 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”). 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.

69. In both 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 generated during processing of the data set. This is presumably why Petitioner invokes a different redundancy transformation (i.e., Copy-N redundancy transformation) to support its argument, rather the one that its proposed systems actually use.

3. 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” (Limitations 1[B], 10[B], 19[D]).

70. 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], cls. 1[B], 10[B,] 19[D]. I understand that 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.” See, e.g., Pet., 16–20. Petitioner argues 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. Pet., 18.

71. 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; 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.”

72. It is my opinion that Petitioner has failed to show a motivation to add an encryption key to Torre's tags to fulfill this claim limitation.

73. In my opinion, Petitioner's proposed combination is based on the errant premise that Torre already teaches including the encryption keys in the tags. *See* Pet. 17. But 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

74. 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 the

need to do so from Torre's lack of any disclosure that keys need to be included in the tags.

75. It is my opinion that Petitioner fails to demonstrate that Torre would benefit from adding the encryption key to its tag as allegedly taught by Tajima. Torre emphasizes that encryption security depends on "proper key management" but is generally 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).

76. My opinion is that Petitioner offers no reason why adopting Tajima's approach—embedding the key with the data—would improve upon Torre's existing "proper key management." Torre's system would gain nothing by incorporating Tajima to embed encryption keys within Torre's shreds. 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.

77. In my opinion, Tajima's alleged key-embedding approach is incompatible with Torre, and adding it to Torre would introduce serious security vulnerabilities.

78. Petitioner incorrectly states that the Torre-Tajima combination would include the encryption key within the "tag" created by Torre's first transformer.

Pet. 17-18. 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. *Id.*, [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. 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. 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. It is also my opinion that adding Tajima's alleged key-embedding approach to Torre would introduce significant security risks. Because the claim limitation 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 is attached to each shred, as Petitioner suggests, Pet. 46; Ex. 1002 [Zadok-Decl.], ¶¶283-84, an attacker who possessed even one the shreds would also possess the encryption key.

79. 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. So even if an attacker could get access to one fragment, he cannot use the key it includes to decrypt the fragment..

80. 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 of 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.

81. By contrast, Torre discloses no such separation keys or control. Tajima's solution would be incompatible with Torre.

82. 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.

83. Moreover, 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 . . . 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. *See id.* Petitioner likewise does not propose modifying Torre to introduce such separate control, instead relying on Torre alone as "disclos[ing] . . . storing the plurality of shares in respective separate storage locations." Pet., 47. Accordingly, in my opinion 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.

84. Torre also 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 "stream ciphers, such as RC4," and "block ciphers, such as the Advanced Encryption Standard (AES), the Data Encryption Standard (DES), and Triple-DES (3DES)." *Id.*, [0084]. Each suggested encryption technique employs external key management:

85. 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.

86. 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.

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

88. Petitioner does not address these drawbacks. A POSITA would recognize that embedding the encryption key within Torre’s shreds—thereby exposing both the data and the key to compromise—reflects precisely 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.

B. Petitioner’s Orsini Grounds Fail (All Claims, Grounds V-VII)

1. The Claims of the ’802 Patent Are Supported by the ’802 Patent’s New Disclosures that Are Not in Orsini.

89. I understand that 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, (Figs. 26–35; 64:3–75:8), which introduce new functionality addressing a different problem. Ex. 1001 [’802], Figs. 26–35, 64:3–75:8. It is my opinion that

Petitioner's assertion that the two disclosures are "substantially identical" overlooks the significance of this added material. Pet., 82–84.

90. It is my understanding that Petitioner's Grounds V-VII rely entirely on Orsini's Figure 21. *See, e.g.*, Pet., 81, 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 Patent], cls. 1, 10, 19; Ex. 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.

91. 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*").

92. 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.

93. For example, 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”); [0381]-[0382] (“[a]ll shares or portions of the data set” are utilized to restore the dataset). In contrast, figures (Fig. 33-Fig. 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], 72:30-72:33 (describing redundancy information “appended ... to allow for the restoration ... using fewer than all the data portions”).

94. The ’802 Patent also 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. 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 encryption and 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.

95. 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.* It is my understanding that Petitioner concedes that Orsini does not disclose any integrity-verification mechanism. Pet., 85; *see also* Ex. 1008 [Orsini], [0294]-[0300].

96. Thus, 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, in my opinion, 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 of 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 discussed below, it is

my opinion that combining Torre with Orsini does not eliminate its deficiencies and, importantly, does not transform the entire encryption scheme into the encryption scheme claimed in the '802 Patent.

2. Petitioner Also Fails To Demonstrate That Orsini/Torre Discloses Or Renders Obvious Restoring The Data Set From "Less Than All" Of The Shares (Limitation 1[G]/9[G]/19[H]).

97. Each of the claims require 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], cls. 1[G], 10[G], 19[I]. I understand that Petitioner contends that "Orsini discloses or renders obvious 1[G]," (Pet., 96; *see also id.*, 98-99), relying on Orsini's Figure 21 embodiment. In my opinion, Petitioner's argument is unsupported and contrary to what Orsini does disclose.

98. I understand that 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. I understand that 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.*

99. 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], [0384]-[0385]. In my opinion, nothing

in Figure 21 or its attendant description discloses redundant cryptosplitting or redundancy of any kind.

100. I understand that 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. Petitioner 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 a redundant cryptosplit 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.

101. It is my opinion that redundant cryptosplitting cannot be obvious to a POSITA in light of Figure 21 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] (“[Figure] 21 illustrates a process for parsing, splitting or separating data with encryption and storage of the encryption master key with the data.”).

102. Figure 21, which incorporates the session key by storing it with the data, is distinct from Figures 22 and 24, which do not “incorporate” 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] (“[Figure] 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*.”)(emphasis added); [0057] (“FIG. 24 illustrates . . . storing the encryption master key *separately from the data*”)(emphasis added).

103. Orsini then discloses in that in 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].

104. As a result, all four shares are necessary to reconstruct the session master key used in Orsini’s Figure 21 embodiment. *See* 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 data set.”) (emphasis added).

105. It is my understanding that 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.*, 81 (citing Ex. 1003 [Torre]). But this argument is undermined by Orsini’s express disclosure that its Figure 21 embodiment cannot restore a data set from “less than all” of the shares. In my opinion and for the same reason, a POSITA could not

have a reasonable expectation of success in trying to improve the Figure 21 embodiment by using redundant cryptosplitting—Orsini itself expressly discloses that it cannot be done.

3. Petitioner Fails to Demonstrate that Orsini Discloses or Renders Obvious “Generating Data Splitting Information” (Limitation 1[C]/9[C]/19[D]).

106. Each of the 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], cls. 1[B]-[C], 10[B]-[C], 19[D]-[E]. I understand that 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.

107. It is my opinion that Petitioner’s reading of the ’802 Patent is incorrect: To establish that those limitations are disclosed by Orsini alone, I understand that Petitioner is required to show that Orsini discloses them separately, as “arranged as in the claim.” I understand that, as a fallback, Petitioner also 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.

108. 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[B], 19[E]. Petitioner contends that both 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. In my opinion, Petitioner is incorrect. I understand that 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. My opinion is that Orsini fails this 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.

109. 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 Patent] Figs. 30-35; 64:3-75:8. The process disclosed for the first time involves creating and storing both integrity information and redundancy information.

110. 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.

111. 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. None of this is disclosed in Orsini.

112. 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.

113. Petitioner also fails to demonstrate that Orsini renders obvious “generating data splitting information.” Petitioner asserts “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 adds 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. Pet., 93. 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.” *Id.*

114. A POSITA would not view such a modification as desirable. Orsini teaches that its method provides a “comprehensive package for data security.” Ex. 1008 [Orsini], [0380]. It explains that “[t]he data shares are then encrypted individually, and the keys are stored safely with different encrypted shares,” and that “[w]hen combined, the entire process for securing data according to the methods disclosed herein becomes a comprehensive package for data security.” *Id.* 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. In my opinion, 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,” Pet., 93, is unsupported, and implausible in the absence of any explanation about how the second key would be secured. In fact, it is very possible that generating a new key for splitting would

decrease overall security provided by Orsini's Figure 21 embodiment by requiring additional complex key management.

115. Petitioner does not address 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. Key management is the most vulnerable point in any security system.

116. 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.

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

118. Petitioner's suggestion that it would be obvious to add a second independent key adds no benefit and introduces new risks. In my opinion, it should be rejected.

4. 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” (Limitations 1[A]/9[A]/19[C], 1[E]/9[E]/19[G]).

119. It is my opinion that, even if the combination of Orsini and Torre disclosed the claim limitations, Petitioner fails to show 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.”

120. The claims of the '802 Patent 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], cls. 1[A], 1[E], 10[A], 10[E], 19[C], 19[G].

121. Here, 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.”).

122. I understand that 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] 569-576 (Examiner’s Reasons for Allowance).

123. I understand that the Reasons for Allowance expressly distinguish the claims from Orsini, stating that, 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 (emphases added).

124. 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. I disagree.

125. It is my opinion that Orsini already achieves the purported benefit of Petitioner’s and Dr. Zadok’s proposed modification. 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].

126. 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.

127. Thus it is my opinion that there would be no motivation for a POSITA to add this modification to Orsini.

128. In my opinion, Petitioner’s motivation to combine Orsini/Torre misreads Orsini, because Orsini does not have the integrity problems purportedly addressed by Torre. 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, it is the use of redundant data that creates the need to check the integrity of the reconstructed data set:

Implementations using signature generation allows 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 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].

129. Orsini’s Figure 21 embodiment, on which I understand 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 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.”). 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].

130. Because Orsini’s Figure 21 embodiment does not include redundant information in its shares, in my opinion, 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.

131. It is also my opinion that Torre counsels away from the modification Petitioner proposes, because the proposed combination would offer no benefit and impose significant cost. 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. As Torre admits, “digital signal generation functions” 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 Patent], cls. 1[A], 10[A], 19[C].

132. 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.

133. In my opinion, Petitioner also fails to explain how Torre’s asymmetric encryption keys are created in the Orsini-Torre system. Petitioner contends that its combination implements Torre’s SHA-1 electronic signature for integrity verification in Orsini’s system, (Ex. 1008 [Orsini] [0054]), which uses asymmetric

encryption, but Orsini's Figure 21 only discloses the use of a "session master key," which necessitates the use of symmetric encryption.

134. 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.

135. 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.

136. 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.

137. 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 important 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.

138. In my opinion, Petitioner ignores the fundamental differences between these two encryption methods and fails to explain 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.

139. At this stage of the proceedings, my opinions are preliminary. I may further develop these opinions or offer additional opinions after a decision to institute, should the Board decide to do so.

VIII. CONCLUSION

140. Although my complete opinions are set forth above, for convenience I summarize several points of my opinions in conclusion. For the foregoing reasons, based on my expertise and experience and the record of this case that I have reviewed, it is my opinion that:

- Claims 1-3, 5, 9, 10-12, 14, 18-21, 23, 27 are not obvious over Torre and Tajima;
- Claims 4, 13, 22 are not obvious over Torre, Tajima, and Aldridge;
- Claims 6, 15, 24 are not obvious over Torre, Tajima, and Krawczyk;

- Claims 7-8, 16-17, 25-26 are not obvious over Torre, Tajima, and Foster;
- Claims 1, 2-5, 9-14, 18-23, 27 are not obvious over Orsini and Torre;
- Claims 6, 15, 14 are not obvious over Orsini, Torre, and Krawczyk; and
- Claims 7-8, 16-17, 25-26 are not obvious over Orsini, Torre, and Foster.

In signing this declaration, I recognize that the declaration will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I also recognize that I may be subject to cross-examination in the case and that cross-examination will take place within the United States. If cross-examination is required, I will appear for cross-examination within the United States during the time allotted.

I hereby declare that all statements made herein of my own knowledge are true and all statements made herein on information and belief were and are believed by me to be true, and that all statements herein were and are made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code and that any such willful false statements may jeopardize the validity of the application or any patents issued thereon.

Dated: October 17, 2025

A handwritten signature in black ink, appearing to read "Sam Malek", written in a cursive style.

Sam Malek, Ph.D.