

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent of: Ranganath G. Iyengar
U.S. Patent No.: 7,904,686 Attorney Docket No.: 39898-0050IP1
Issue Date: March 8, 2011
Appl. Serial No.: 12/152,595
Filing Date: May 13, 2008
Title: DATA SECURITY FOR USE WITH A FILE SYSTEM

Mail Stop Patent Board

Patent Trial and Appeal Board
U.S. Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

**PETITION FOR *INTER PARTES* REVIEW OF UNITED STATES PATENT
NO. 7,904,686 PURSUANT TO 35 U.S.C. §§ 311–319, 37 C.F.R. § 42**

TABLE OF CONTENTS

I.	REQUIREMENTS FOR IPR—37 C.F.R. §42.104.....	1
A.	Grounds for Standing—37 C.F.R. §42.104(a).....	1
B.	Challenge and Relief Requested—37 C.F.R. §42.104(b)	1
II.	THE '686 PATENT	2
A.	Pertinent Disclosure	2
B.	Prosecution History.....	3
III.	LEVEL OF ORDINARY SKILL	4
IV.	CLAIM CONSTRUCTION—37 C.F.R. §42.104(b)(3)	4
V.	THE CHALLENGED CLAIMS ARE UNPATENTABLE.....	5
A.	GROUND 1A: Claims 1-2, 5-10, 13-18 Are Obvious Over Madany- Locasto	5
1.	Madany (EX1004).....	5
2.	Locasto (EX1005)	9
3.	The Combination.....	14
4.	Motivation to Combine	18
5.	Reasonable Expectation of Success	21
6.	Claim Element Analysis.....	21
B.	GROUND 1B: Claims 1-2, 5-10, 13-18 Are Obvious Over Madany- Locasto	42
1.	Madany (EX1004).....	42
2.	Locasto (EX1005)	42
3.	The Combination.....	42

4.	Motivation to Combine and Reasonable Expectation of Success.	44
5.	Claim Element Analysis.....	45
C.	GROUND 2: Claims 1-18 Are Obvious Over Madany-Locasto-Leffler ...	
		58
1.	Leffler (EX1006).....	59
2.	Claim Element Analysis.....	63
D.	GROUND 3: Claims 1-8 Are Obvious Over Leffler-Locasto.....	74
1.	Leffler (EX1006).....	74
2.	Locasto (EX1005)	75
3.	The Combination.....	75
4.	Motivation to Combine	79
5.	Reasonable Expectation of Success	82
6.	Claim Element Analysis.....	82
VI.	PAYMENT OF FEES—37 C.F.R. §42.103.....	97
VII.	MANDATORY NOTICES—37 C.F.R § 42.8(a)(1)	97
A.	Real Party-In-Interest—37 C.F.R. § 42.8(b)(1).....	97
B.	Related Matters—37 C.F.R. § 42.8(b)(2).....	98
C.	Lead And Back-Up Counsel—37 C.F.R. § 42.8(b)(3).....	98
D.	Service Information	98

LIST OF EXHIBITS

EX1001	U.S. Patent No. 7,904,686 to Reese et al. (“the ’686 patent”)
EX1002	Excerpts from the Prosecution History of the ’686 patent (“the Prosecution History”)
EX1003	Declaration of Dr. Nenad Medvidovic
EX1004	U.S. Patent No. 5,774,715 (“Madany”)
EX1005	Locasto, <i>et al.</i> , 2004. “CamouflageFS: Increasing the effective key length in cryptographic filesystems on the cheap.” In Applied Cryptography and Network Security: Second International Conference, ACNS 2004, Yellow Mountain, China, June 8-11, 2004. Proceedings 2 (pp. 1-15). Springer Berlin Heidelberg (“Locasto”)
EX1006	Leffler, et al., “The Design and Implementation of the 4.3 BSD UNIX Operating System,” Addison-Wesley, 1989-90 (“Leffler”)
EX1007	Downing, et. al., “Dictionary of Computer and Internet Terms,” Ninth Edition, Barron’s Educational Series, Inc., 2006 (“Downing”).
EX1008	U.S. Pat. App. Pub. No. 2006/0184673 (“Liebman”)
EX1009	U.S. Pat. App. Pub. No. 2012/0124121 (“Pope”)
EX1010	Custer, “Inside the Windows NT File System,” Microsoft Press, 1994 (“Custer”)
EX1011	Garfinkel, “Practical Unix Security,” O’Reilly & Associates, Inc., 1992 (“Garfinkel”)
EX1012	U.S. Pat. App. Pub. No. 2005/0091487 (“Cross”)

EX1013	U.S. Pat. App. Pub. No. 2007/0168678 (“Meenakshisundaram”)
EX1014	Rivest, et al, 1998, August. “The RC6™ Block Cipher.” In First advanced encryption standard (AES) conference (p. 16) (“Rivest”)
EX1015	Arnold, “Unix Security: a practical tutorial” McGraw-Hill, 1992 (“Arnold”)
EX1016	Tanenbaum, et. al, “Modern Operating Systems, Second Edition,” Prentice Hall, 2001 (“Tanenbaum”)
EX1017	Kelsey, 2002, February, “Compression and information leakage of plaintext,” in International Workshop on Fast Software Encryption (pp. 263-276). Berlin, Heidelberg: Springer Berlin Heidelberg (“Kelsey”)
EX1018	U.S. Pat. App. Pub. No. 2008/0034264 (“Kaabouch”)
EX1019 – EX1029	RESERVED
EX1030	Applied Cryptography and Network Security: Second International Conference, ACNS 2004, Yellow Mountain, China, June 8-11, 2004. Proceedings 2 (full). Springer Berlin Heidelberg.
EX1031	Library of Congress. https://lccn.loc.gov/2004106759 (accessed August 20, 2025).
EX1032	Declaration of June Munford
EX1033	“Pearson: History.” https://web.archive.org/web/20060711035850/http://www.pearson.com/index.cfm?pageid=14 (accessed August 20, 2025)

- EX1034 “Addison-Wesley.” Wikipedia, The Free Encyclopedia, May 10, 2025, <https://en.wikipedia.org/wiki/Addison-Wesley> (accessed August 20, 2025)
- EX1035 “Books by Marshall Kirk McKusick.” <https://web.archive.org/web/19971224220211/https://www.mckusick.com/books.html> (accessed August 20, 2025)
- EX1036 Library of Congress. <https://lcn.loc.gov/88022809> (accessed August 20, 2025).
- EX1037 Marsh, et al. “First-Class User-Level Threads,” Association for Computing Machinery, 1991.
- EX1038 U.S. Patent No. 6,477,560

LISTING OF CLAIMS

Claim 1	
[1.pre]	A method of providing data security for use with a file system, the method comprising:
[1.1.1]	applying a mapping function to data block numbers that are associated with a file,
[1.1.2]	wherein the data block numbers are contained in an index node associated with said file; and
[1.2.1]	obtaining mapped data block numbers after applying the mapping function,
[1.2.2]	wherein the mapped data block numbers are addresses of data of the file in a storage device.
Claim 2	
[2]	The method of claim 1, further comprising: storing a mapping of the data block numbers and the mapped data block numbers.
Claim 3	
[3.1]	The method of claim 1, wherein a read operation for the file comprises: obtaining the data block numbers of the file;
[3.2]	obtaining the mapped data block numbers of the file based on the mapping function that is applied to the data block numbers; and
[3.3]	reading the data in the mapped data block numbers in the storage device.
Claim 4	
[4.1]	The method of claim 1, wherein a write operation for the file comprises: allocating data block numbers of free data blocks in the storage device;

[4.2]	linking the data block numbers to a file name of the file;
[4.3]	applying the mapping function to the data block numbers in order to obtain the mapped data block numbers; and
[4.4]	writing the data of the file into the mapped data block numbers in the storage device.
Claim 5	
[5]	The method of claim 1, wherein the mapping function is an equation that is applied to each data block number of the file.
Claim 6	
[6]	The method of claim 1, further comprising: selecting a complexity level of the mapping function in order to set a security level and efficiency of the mapping function.
Claim 7	
[7]	The method of claim 1, wherein the data block numbers are addresses of data of the file in the storage device if the mapping function is not applied to the data block numbers.
Claim 8	
[8]	The method of claim 1, wherein the mapping function prevents copying of data of the file in the storage device.
Claim 9	
[9.pre]	An apparatus for providing data security for use with a file system, the apparatus comprising:
[9.1.1]	a block distribution engine configured to apply a mapping function to data block numbers that are associated with a file,
[9.1.2]	wherein the data block numbers are contained in an index node associated with said file, and

[9.1.3]	obtain mapped data block numbers after applying the mapping function;
[9.2]	wherein the mapped data block numbers are addresses of data of the file in a storage device.
Claim 10	
[10]	The apparatus of claim 9, wherein the block distribution engine is configured to store a mapping of the data block numbers and the mapped data block numbers.
Claim 11	
[11.1]	The apparatus of claim 9, wherein for a read operation for the file, the block distribution engine obtains the data block numbers of the file from a file system, and
[11.2]	obtains the mapped data block numbers of the file based on the mapping function that is applied to the data block numbers;
[11.3]	and wherein a block device driver is configured to read the data in the mapped data block numbers in the storage device.
Claim 12	
[12.1]	The apparatus of claim 9, wherein for a write operation for the file, the block distribution engine is configured to receive allocated data block numbers of free data blocks in the storage device from a file system,
[12.2]	link the data block numbers to a file name of the file, and
[12.3]	apply the mapping function to the data block numbers in order to obtain the mapped data block numbers;
[12.4]	and wherein a block device driver is configured to write the data of the file into the mapped data block numbers in the storage device.

Claim 13	
[13]	The apparatus of claim 9, wherein the mapping function is an equation that is applied to each data block number of the file.
Claim 14	
[14]	The apparatus of claim 9, wherein a complexity level of the mapping function is selected in order to set a security level and efficiency of the mapping function.
Claim 15	
[15]	The apparatus of claim 9, wherein the data block numbers are addresses of data of the file in the storage device if the mapping function is not applied to the data block numbers.
Claim 16	
[16]	The apparatus of claim 9, wherein the mapping function prevents copying of data of the file in the storage device.
Claim 17	
[17.pre]	An apparatus for providing data security for use with a file system, the apparatus comprising:
[17.1.1]	means for applying a mapping function to data block numbers that are associated with a file,
[17.1.2]	wherein the data block numbers are contained in an index node associated with said file; and
[17.2.1]	means for obtaining mapped data block numbers after applying the mapping function,
[17.2.2]	wherein the mapped data block numbers are addresses of data of the file in a storage device.
Claim 18	
[18.pre]	Article of manufacture, comprising:

[18.1]	a machine-readable medium having stored thereon instructions to:
[18.2.1]	apply a mapping function to data block numbers that are associated with a file,
[18.2.2]	wherein the data block numbers are contained in an index node associated with said file; and
[18.3.1]	obtain mapped data block numbers after applying the mapping function,
[18.3.2]	wherein the mapped data block numbers are addresses of data of the file in a storage device.

CONVENTIONS OF THE PETITION

- All emphases are added unless noted otherwise;
- Bold-italic emphases correlate to claim language;
- Quotations are generally from exhibits, not claim language;
- Citations to non-patent literature reference the absolute page number added to the exhibit (as opposed to the original pagination of the document), except for Leffler (EX1006) which refers to the document's original pagination; and
- The phrase “as discussed” and equivalent phrases incorporate fully the analysis of the cross-cited portion of the Petition.

FedEx Corporation and FedEx Office and Print Services, Inc. petition for *Inter Partes* Review of Claims 1-18 (“Challenged Claims”) of U.S. Patent No. 7,904,686.

I. REQUIREMENTS FOR IPR—37 C.F.R. §42.104

A. Grounds for Standing—37 C.F.R. §42.104(a)

Petitioners certify that the ’686 patent is available for IPR, and that Petitioners are not barred or estopped from requesting IPR.

B. Challenge and Relief Requested—37 C.F.R. §42.104(b)

Petitioners request IPR and cancellation of the ’686 patent’s Challenged Claims based on the grounds below. Petitioners submit that the cited prior art references are themselves sufficient to establish obviousness pursuant to the analysis in Section V. The Section V analysis further cites to the Declaration of Dr. Medvidovic for complementary corroboration and explanation. *See* EX1003.

Asserted Grounds		
ID	Claims	Prior Art Basis
1A	1-2, 5-10, 13-18	§103—Madany (EX1004), Locasto (EX1005) ¹
1B	1-2, 5-10, 13-18	§103—Madany, Locasto
2	1-18	§103—Madany, Locasto, Leffler (EX1006)
3	1-8	§103—Leffler, Locasto

¹ The Madany-Locasto Combination yields two distinct obviousness mappings. Petitioners present these mappings in separate grounds for clarity and convenience.

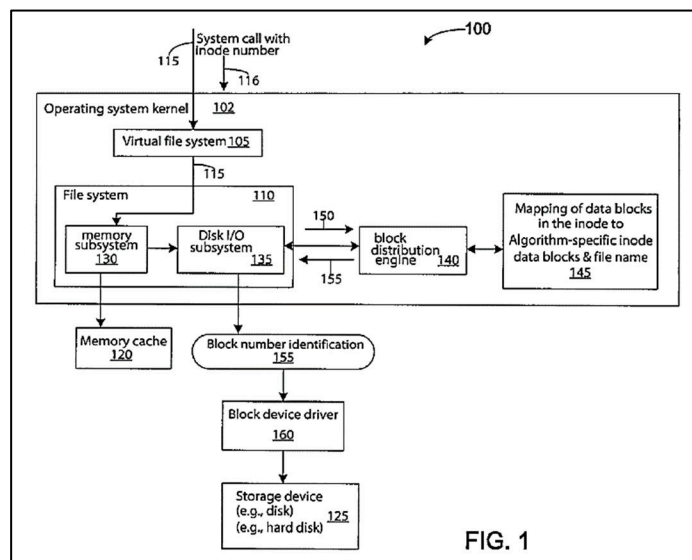
As shown below, each of the above identified references pre-dates the '686 patent's earliest claimed priority date of July 31, 2007 ("Critical Date").

Reference	Prior Art Date	Status
Madany (EX1004)	Issued Jun. 30, 1998	§102(b)
Locasto (EX1005)	Published 2004	§102(b)
Leffler (EX1006)	Published 1989-1990	§102(b)

II. THE '686 PATENT

A. Pertinent Disclosure

The '686 patent "relate[s] generally to data security for use with a file system." EX1001, 1:6-7; EX1003, ¶38. Figure 1 of the '686 patent shows "a block diagram of an apparatus (system)...implemented in, for example, a computer." EX1001, 2:1-14. The block diagram shows an operating system 102 including a file system 110 (e.g., one or more "Unix file systems") that "will assign a unique inode for each file." EX1001, 2:15-28. "As known to those skilled in the art, an inode (index node) is a data structure that contains information about a file," including "the addresses of the file's data blocks on the disk." *Id.* Where, "[i]n previous systems, the data blocks of a file are linearly placed on a storage device and can be obtained by



accessing the inode of the file,” the ’686 patent uses a block distribution engine 140 to perform “a block distribution algorithm...[that] will vary the distribution of the data blocks on the storage device [] in order to provide security to the data blocks.” EX1001, 2:62-3:8.

B. Prosecution History

The ’686 patent issued from U.S. Patent Application No. 12/152,595 (“the ’595 application”). EX1001, Cover. The Examiner rejected the ’595 application’s claims over a single prior art reference—US 2008/0034264 (“Kaabouch”)—in a single office action. EX1002, 315-318 The Applicant responded by amending the claims to include the limitation of “wherein the data block numbers are contained in an index node associated with said file” and arguing that Kaabouch teaches “memory addresses” but “is silent” regarding “data block numbers.” EX1002, 275-284. The Examiner subsequently issued a notice of allowance, and the ’686 patent issued on March 8, 2011. EX1002, 262-267; EX1001, Cover.

Clearly, the Examiner erred. For one, the Kaabouch reference is hardly pertinent to the subject matter at hand. The ’686 patent is about upgrading file system security via an algorithm that distributes data blocks derived from a file’s index node. *Supra* §II.A. Indeed, the title is “Data Security for Use With a File System.” EX1001, Cover. By contrast, Kaabouch is directed to the concept of protecting the program memory of a microcontroller and does not describe a file

system at all. EX1018, [0001]-[0008]. It is no surprise that Kaabouch does not disclose the amended feature of “wherein the data block numbers are contained in an index node associated with said file.” But that does not mean the ’686 patent’s claims are valid. Data block numbers and index nodes were well-known components of the very “Unix file systems” that the ’686 patent concedes were prior art. EX1001, 2:1-14; *see also* 1:25-32; *see also infra*, §V.C.1 (summarizing Leffler’s disclosure regarding the 4.3BSD Unix operating system). Had the Examiner performed a more thorough search—or even just read the ’686 patent’s specification—the Challenged Claims would not have issued.

III. LEVEL OF ORDINARY SKILL

A person of ordinary skill in the art (“POSITA”) relating to the subject matter of the ’686 patent would have had (1) at least a master’s degree in computer science, computer engineering, or a related field, and (2) at least two years of professional experience involving the study and/or development of operating systems, file systems, and/or storage devices. EX1003, ¶¶39-40; *see also* ¶¶41-49 (discussing a POSITA’s background knowledge of file systems and data security).

IV. CLAIM CONSTRUCTION—37 C.F.R. §42.104(b)(3)

The analysis below in Section V proves the Challenged Claims obvious under any interpretation reasonably advanced according to the *Phillips* standard. Moreover, consistent with the requirements of 37 C.F.R. §42.104(b)(3), this Petition

identifies the “specific portions of the specification” that describe structures corresponding to the recited function of each §112, ¶6 element of the ’686 patent’s Claim 17. Nevertheless, Petitioners submit that the Board need not resolve any express constructions. Should any subsequent discussion of claim interpretation arise here or in the parallel district court proceeding, Petitioners reserve the right to address and respond to any construction advanced by Patent Owner, the Board, and/or the District Court.

V. THE CHALLENGED CLAIMS ARE UNPATENTABLE

A. GROUND 1A: Claims 1-2, 5-10, 13-18 Are Obvious Over Madany-Locasto

1. Madany (EX1004)

Madany describes a “method, apparatus, and computer-usable medium for compressing data in a file system utilizing the concept of ‘holes’.” EX1004, Abstract; EX1003, ¶¶51-55. Madany’s “FIG. 1 shows in general a structure for a file system 200” that contains “a hierarchy of directory structures 201, a file table 301 and a mapping table 305...for each entry in the file table 301.” EX1004, 5:29-35. The directory contains “entries of file names within the file system and corresponding file numbers that allow files to be indexed to an entry in the file table.” EX1004, 5:32-34. In Madany, “[e]ach entry 302 in the file table 301 identifies a file of the file system and is commonly known as an Inode.” EX1004, 5:35-37.

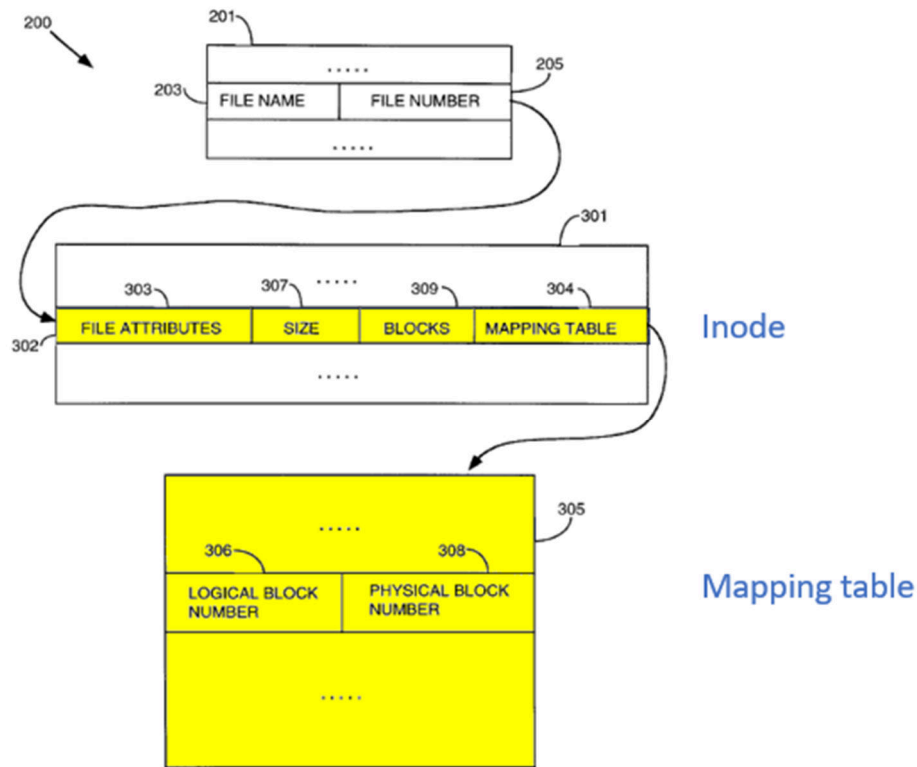


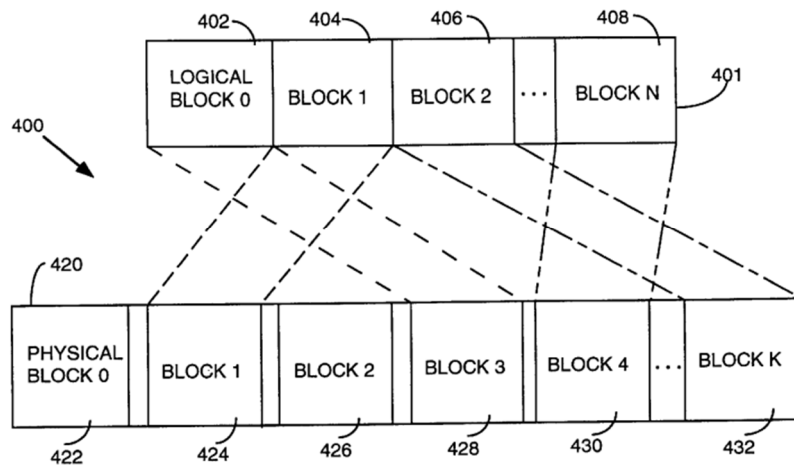
Figure 1

EX1004, Figure 1 (annotated)

Madany explains that the “mapping table 305 contains information on how the logical blocks within a file are mapped to physical blocks on a disk” and the “physical block number identifies the location on the disk where the logical block is to be stored.” EX1004, 5:52-54, 5:63-64. For example, Madany explains that the first entry in the mapping table 305 “maps to the physical block number 308 for logical block number 0 of the file.” EX1004, 5:58-60.

Madany further describes the mapping process in reference to Figure 2, which “shows symbolically how the logical blocks of the file may be mapped to physical blocks on a hard disk.” EX1004, 6:3-4. As shown below, Figure 2 depicts a logical

file 401 and a hard disk 420. EX1004, 6:5-7. The “hard disk is typically separated into physical blocks, but it is not always true that the logical blocks of the file are stored in contiguous physical blocks on a hard disk,” e.g., “the logical blocks of logical file 400 are shown being mapped to noncontiguous physical blocks of the hard disk 420,” so “when the logical file 401 is written to the hard disk the logical blocks will be written to noncontiguous physical blocks.” EX1004, 6:7-15. “In this example, logical block zero 402 is written to physical block three 428;” “[l]ogical block one 404 is written to physical block one 424;” etc. EX1004, 4:15-16. Therefore, Madany’s “mapping table 305 of Figure 1 is used to indicate to which physical blocks the logical blocks are written.” EX1004, 6:25-29.

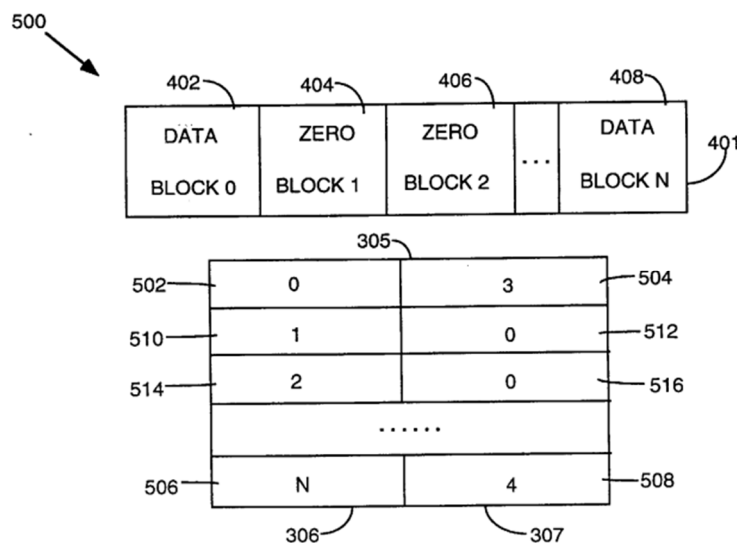


EX1004, Figure 2.

Madany also explains that “a logical block within a file [can] contain[] very little data or may in fact be completely empty, that is, the logical block is full of zeroes,” e.g., “if logical block one 404 was completely full of zeroes then this zero

block would be written in its entirety to physical block one 424.” EX1004, 6:33-35. This approach “is not desirable as a physical block completely full of zeroes is a waste of space.” EX1004, 6:35-37. Madany calls blocks containing only zeroes “holes.” EX1004, 6:38-39.

To avoid such waste, in the mapping table, Madany represents holes using a special address, which is “not an actual physical block address where data will be written but a single address that indicates that the corresponding logical blocks are in fact full of zeroes.” EX1004, 6:59-61; EX1003, ¶55. Madany’s Figure 3 shows an example in which “logical block zero 402 and logical block N 408 each contain non-zero data,” but “logical blocks one and two both contain nothing but zeroes.” EX1004, 6:43-46. In this example, “a hole 512 or 516 is represented by the number zero.” EX1004, 6:57-58.

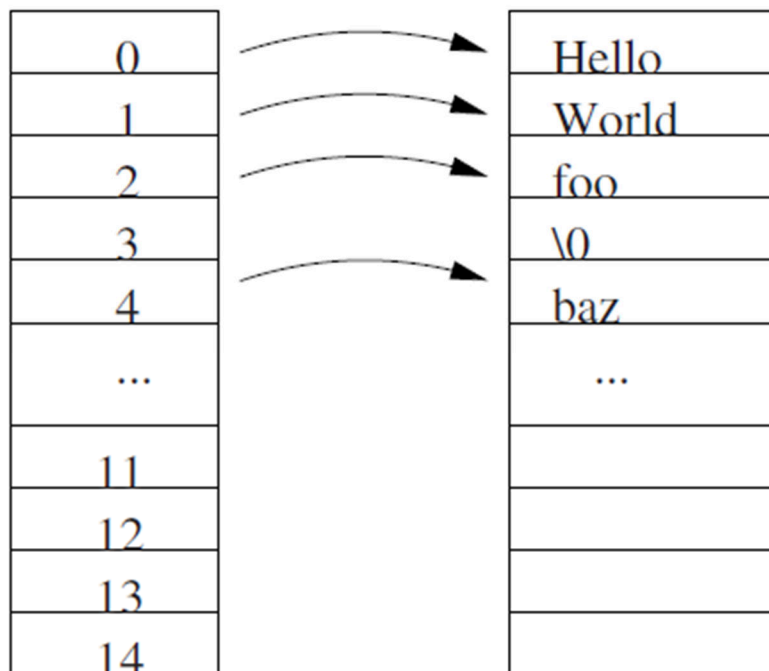


EX1004, Figure 3.

2. Locasto (EX1005)

(a) Summary of Locasto's Pertinent Disclosure

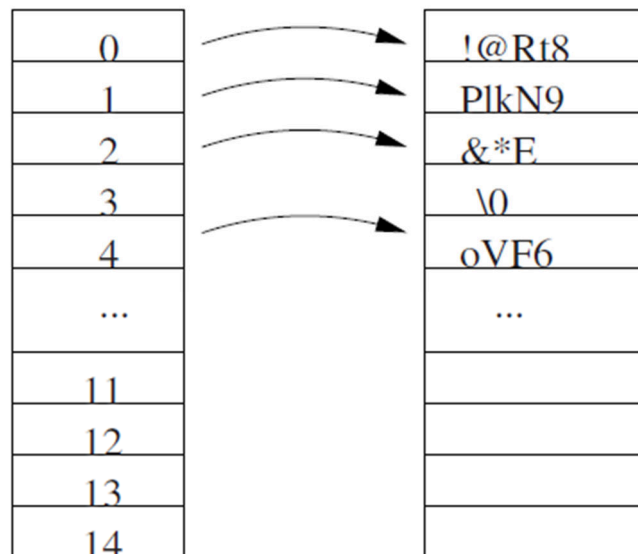
Locasto is a technical paper titled “ComouflageFS: Increasing the Effective Key Length in Cryptographic Filesystems on the Cheap.” EX1005, 1; EX1003, ¶¶56-65. Locasto explains that file systems can use indexed allocation in which “the operating system maintains an index of entries per file that can quickly address any given block of that file.” EX1005, 3. Locasto’s Figure 2 shows “a single-level direct-mapped index,” where “[t]he file index points directly to blocks with plaintext data.” EX1005, 3.



EX1005, Figure 2.

Like Madany, Locasto explains that “[h]oles in the file may exist; reading data from such holes returns zeroed-out blocks. EX1005, 3. The index of Figure 2 “has a hole at the third block position.” EX1005, 4.

Locasto further explains that “[c]ryptographic file systems encrypt the stored data, which leaves the index structure identical but protects the contents of the data blocks, as shown in Figure 3.” EX1005, 3. In this example, “[t]he indexing has not changed, merely the contents of the data blocks,” which are encrypted. EX1005, 3, 5. As with Figure 1, “the file hole at block three is present.” EX1005, 5.



EX1005, Figure 3.

To improve security “without incurring the usual performance penalty,” Locasto introduces a “spreading schedule” that “randomly distribute[s] the real file data throughout a large address space so that an attacker would have to first guess

the spreading schedule before he attempts a brute force search on the rest of the key.”

EX1005, 2 (altered emphasis), 4.

Locasto explains that in conventional systems, “the number of the index entry is calculated by taking the floor of the current file position ‘pos’ divided by the block size,” such that “*index = pos / blocksize*.” EX1005, 4. The computed “index number is then used to derive the logical block number (the block on disk) where the data at ‘pos’ resides,” such that “*lbn = get from index(index)*,” where “lbn” is the logical block number. EX1005, 4. To be clear, when Locasto refers to a “logical block number” as “the block on the disk,” a POSITA would understand this to mean the number of the physical disk block where the data resides. EX1003, ¶61. To avoid confusion with respect Locasto’s terminology and Madany’s use of the terms “logical block number” and “physical block number,” this Petition will use the generic term “block number” when referencing and quoting from Locasto.

Moving on, Locasto alters the conventional two-step approach by introducing a spreading schedule such that the “initial calculation of the index is performed, but before the [] block number is derived, a pseudo-random permutation (PRP) function takes the calculated index and the bits of the spreading seed to return a new index value,” which Locasto calls a “map” function in the second equation below. EX1005, 4-5; EX1003, ¶63. Thus, Locasto “calculat[es] a new entry in the file index, rather than using the strictly sequential ordering” of the blocks. EX1005, 5;

EX1003, ¶63. Locasto calls the mapped virtual data blocks “an implicit *virtual index*” which is used “to spread the file data blocks throughout the file’s address space.” EX1005, 6.

$$index = pos / blocksize$$

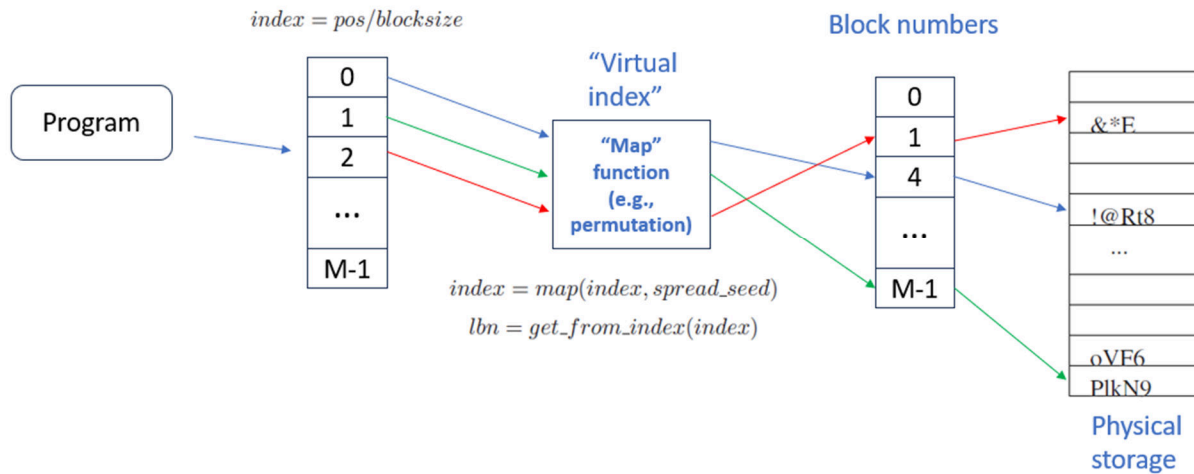
$$index = map(index, spread_seed)$$

$$lbn = get_from_index(index)$$

EX1005, 5.²

Locasto’s process is illustrated in the visual aid below. EX1003, ¶65. In the first step, an “index entry is calculated by taking the floor of the current file position ‘pos’ divided by the block size.” EX1005, 4. In the second step, an index (e.g., 0) is “spread” to a block number (e.g., 4, shown with the blue arrow) using a “map” function, e.g., a pseudo-random permutation (e.g., $index = map(index, spread\ seed)$) followed by an identification of the block number (e.g., $lbn = get_from_index(index)$). EX1005, 5; EX1003, ¶65. Using the same process, index number 1 is mapped (spread) to block number M-1 (green arrow) and block number 2 is mapped to mapped block number 1 (red arrow). EX1003, ¶65.

² While Locasto uses “spread_seed” as a variable in this function, Locasto describes it as a “spreading seed.” This Petition uses “spreading_seed” in the formulas.



Locasto: Visual Aid

(b) Analysis of Locasto's Public Availability

Locasto was published by Springer-Verlag Berlin Heidelberg in 2004 in a conference proceedings collection titled *"Applied Cryptography and Network Security: Second International Conference, ACNS 2004; Yellow Mountain, China, June 2004; Proceedings."* See EX1005, 1; EX1030, 1-5, 15-29. The collection's front matter lists a Library of Congress Control Number, an ISSN, an ISBN, and a 2004 copyright date. EX1030, 5. The Library of Congress's catalog includes an entry matching the details provided in the collection's front matter and corroborating the 2004 publication year. Compare EX1031, 1-2, with EX1030, 5. Cataloging dates extracted from various other library MARC records further corroborate the 2004 public availability of Locasto in the ACNS conference proceedings collection. See EX1032, ¶¶4-5, ¶¶14-23.

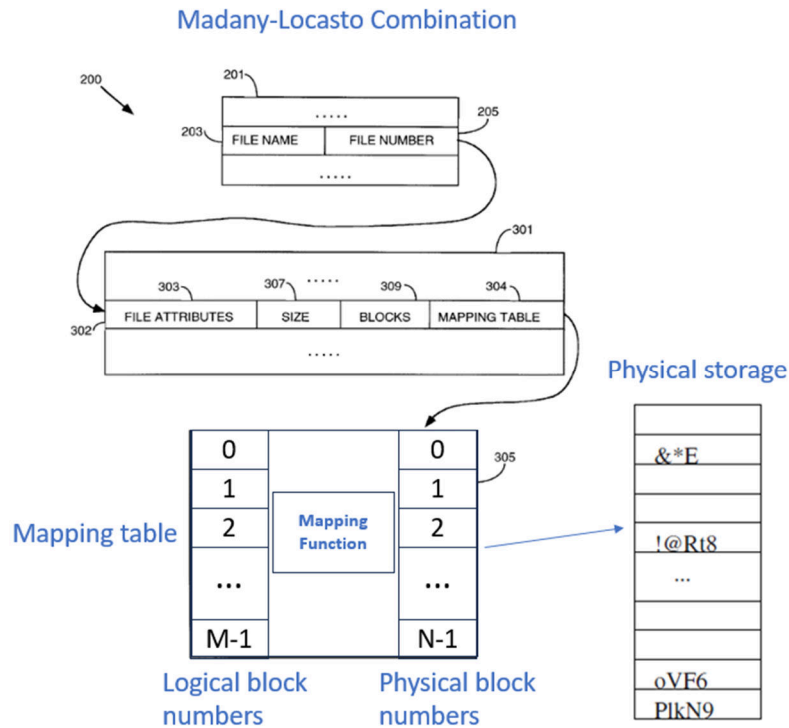
3. The Combination

As discussed, Madany describes a “method, apparatus, and computer-usable medium for compressing data in a file system utilizing the concept of ‘holes’.” EX1004, Abstract. Madany explains that “mapping table 305 contains information on how the logical blocks within a file are mapped to physical blocks on a disk” and the “physical block number identifies the location on the disk where the logical block is to be stored.” EX1004, 5:52-54, 5:63-64; §V.A.1, *supra*. Similar to Madany, Locasto describes a file system in which “indices” are mapped to block numbers. EX1005, 3-4; §V.A.2, *supra*; EX1003, ¶66. Locasto introduces two security features: [i] a cryptographic file system that encrypts the stored data and a mapping function that uses a permutation function and [ii] a spreading seed to randomly distribute real file data throughout a large address space. EX1005, 3-6, Figures 3-4.

In the Madany-Locasto Combination, the file system Madany teaches is enhanced with security features like those taught in Locasto, (e.g., adding a mapping function and encrypting the stored data). EX1003, ¶67. As described by Locasto, such “[c]ryptographic file systems provide data confidentiality by employing encryption to protect files against unauthorized access.” EX1005, 1; EX1003, ¶67. Thus, as explained further below in Section V.A.4, a POSITA would have appreciated the predictable benefits of incorporating Locasto’s security features with Madany’s efficient file system and associated data structures. EX1003, ¶67 (citing

EX1010; EX1011). Moreover, the POSITA would have understood the compatibility of Locasto's teachings with Madany, such that Locasto's technical improvements could be implemented while preserving the fundamental aspects of Madany's disclosure. EX1003, ¶68. For example, Madany describes Unix file systems and Locasto describes the "Linux Ext2fs" file system. EX1004, 1:45-50, 3:14-18, 5:41-42; EX1005, 3, 8. A POSITA would have understood that Linux is a Unix-like operating system. EX1003, ¶68 (citing EX1008; EX1009).

A POSITA would have understood that Madany's file system architecture presents two apparent options for incorporating Locasto's mapping function. EX1003, ¶69. First, it would have been apparent to the POSITA that Locasto's mapping function would be deployed effectively as a mechanism to convert Madany's logical block numbers to physical block numbers, as shown in the visual aid below. *Id.* This first version of the Madany-Locasto Combination is the subject of Ground 1A. *Id.* Second, it would have been apparent to the POSITA that Locasto's mapping function also would be deployed effectively as a mechanism to map the physical block numbers in Madany's mapping table to addresses on the physical storage disk. *Id.* This second version of the Madany-Locasto Combination is the subject of Ground 1B. *Infra*, §V.B.

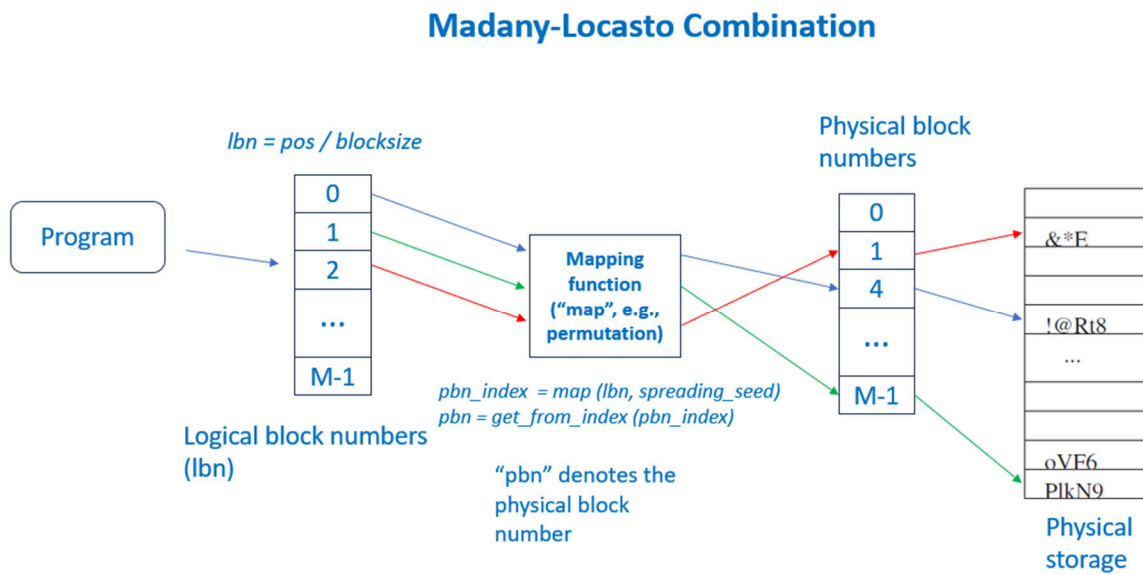


Madany-Locasto Combination: Visual Aid³

In operation, when a program accesses a file at a given position, the Madany-Locasto Combination of Ground 1A determines the inode and then uses a mapping table pointer in the inode to determine the mapping table. EX1004, 5:38-40, 48-49; EX1003, ¶70. Next, as shown in the visual aid above, the Combination employs a process like Locasto's (*see* §V.A.2) to facilitate the conversion of logical block numbers in Madany's mapping table to physical block numbers (*see* §V.A.1). More

³ This visual aid, adapted from Madany and Locasto, is not proposing bodily incorporation but merely showing an example of the Combination for the Board's convenience. The same is true of all visual aids in this Petition.

specifically, logical block numbers are “spread” to physical block numbers using Locasto’s “map” function, e.g., a pseudo-random permutation function and a “spreading seed.” EX1005, 4-5; EX1003, ¶70. The mapping function “randomly distribute[s] the real file data...throughout a large address space so that an attacker would have to first guess the spreading schedule before he attempts a brute force search on the rest of the key.” EX1005, 4.



Madany-Locasto Combination: Visual Aid

As shown in the visual aid above, the Madany-Locasto Combination applies Locasto’s mapping function (*map*) to the logical block numbers (*lbn*) in Madany’s mapping table and the bits of a *spreading seed* to create an index from which the physical block number (*pbn*) is obtained:

pbn_index = map(lbn, spreading_seed)

pbn = get_from_index(pbn_index)

EX1005, 5-6; EX1003, ¶71. These two functions can be expressed equivalently as the single mapping function:

pbn = get_from_index (map (lbn, spreading_seed))

EX1003, ¶71.

4. Motivation to Combine

The following reasons, taken separately or collectively, support the above-described Madany-Locasto Combination. EX1003, ¶¶72-78.

First, a POSITA would have been motivated to pursue the Madany-Locasto Combination for the purpose of maintaining confidentiality of the data in Madany's file system. EX1004, Abstract; EX1005, 3-6; EX1003, ¶73; *see also* ¶¶44-49. As Locasto explains, “[c]ryptographic file systems provide data confidentiality by employing encryption to protect files against unauthorized access.” EX1005, 1. The POSITA would have appreciated the importance of protecting sensitive data against unauthorized access and, thus, would have been prompted to bolster Madany's file system disclosure with data encryption teachings from a supplemental reference like Locasto. EX1003, ¶73.

Second, a POSITA would have been motivated to pursue the Combination to achieve in Madany the specific benefits espoused in Locasto. EX1003, ¶¶74-75. For example, Locasto's approach to encryption "reduc[es] the performance penalty associated with using larger key sizes without decreasing the level of security." EX1005, 2. As Locasto explains, by "using only part of the key for cryptographic operations" and "[t]he rest of the key...to unpredictably spread the data into the file's address space," "a user may decrease the performance penalty they pay for employing a cryptographic file system." *Id.*; see also 1 ("improved performance relative to a cipher with a larger key-schedule"), 12 ("a negligible impact on performance") ("we have increased the effective cryptographic key length by 28 bits, correspondingly increasing an attacker's work factor by 2^{28}). From the POSITA's perspective, Locasto's balance of security and performance would have been an advantageous addition to Madany's file system. EX1003, ¶75.

Third, "design incentives" and "market forces" would have prompted a POSITA to pursue a combination of Madany and Locasto. EX1003, ¶76. For example, a POSITA would have known that data security solutions were incorporated into "popular file systems," such as the "WINDOWS NT file system" that Madany expressly identified. EX1004, 1:44-52; EX1003, ¶76 (citing and quoting EX1010, 5 (a WINDOWS NT reference book explaining that "Data security is of primary importance...")). Madany also references and applies aspects of the

UNIX operation system. EX1004, 1:44-52, 5:29-51. And it was well known that skilled artisans and market participants were actively pursuing secure variations of the UNIX operating system, some of which included built-in encryption programs. EX1003, ¶76 (citing EX1015, 6 (“The UNIX operating system has been augmented by many firms in order to gain better [security] ratings.”); EX1011, 14 (noting that a Sun Microsystems’ UNIX-based operating system incorporates data encryption algorithms)). Accordingly, the POSITA would have sought data security improvements to Madany’s file system in an effort to compete and distinguish over competitor products in the marketplace. EX1003, ¶76; *see also* ¶¶44-49.

Fourth, by the time of the ’686 patent in 2007, the scientific and patent literature was replete with proposals to incorporate data encryption techniques into existing file systems. EX1003, ¶77 (citing EX1012; EX1013). A POSITA would have been motivated by this contemporaneous design and research trend to incorporate Locasto’s encryption-focused teachings into a file system like Madany’s. EX1003, ¶77; *see also* ¶¶44-49.

Fifth, a POSITA would have been led to the Madany-Locasto Combination because it merely involves the application of a known technique (Locasto’s data encryption) to a known system (Madany’s file system) to achieve predictable results (a file system with improved data security). EX1003, ¶78. Each element in combination merely performs the same function as it does separately. *Id.* “It’s

enough for [the Petition] to show that there was a known problem of [data security], that [Locasto’s encryption technique] helped address that issue, and that combining the teachings of [Madany and Locasto] wasn’t beyond the skill of an ordinary artisan. Nothing more is required to show a motivation to combine under *KSR*.” *Intel Corp. v. PACT XPP Schweiz AG*, 61 F.4th 1373, 1380-81 (Fed. Cir. 2023).

5. Reasonable Expectation of Success

A POSITA would have had a reasonable expectation of success in creating a combination like the Madany-Locasto Combination, which would have required only routine programming knowledge well within the skill of a POSITA. EX1003, ¶¶79-81. For example, the Madany-Locasto Combination includes components of a file system and known encryption techniques to enhance the security of such file systems. *Supra* §§V.A.1–3; EX1003, ¶79. Integrating these aspects of Madany and Locasto would have been a routine programming task for a POSITA. EX1003, ¶80.

6. Claim Element Analysis

(a) Claim 1

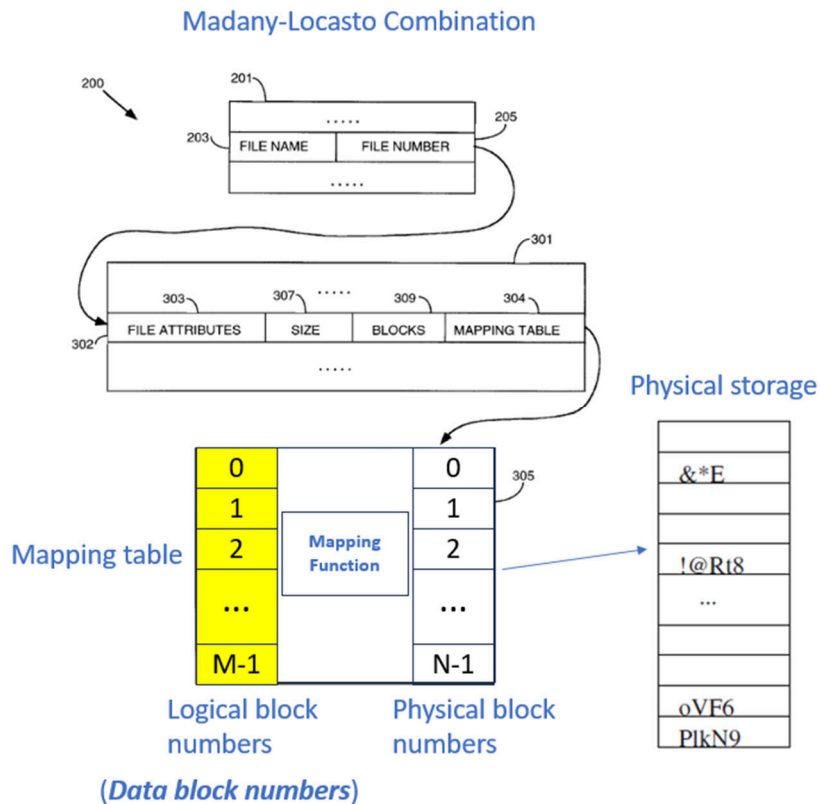
Element_[1.pre]:

The Madany-Locasto Combination renders obvious Element_[1.pre]. EX1003, ¶82. For example, the Combination includes Madany’s description of a “*method*, apparatus, and computer-usable medium for compressing data in a *file system* utilizing the concept of ‘holes’.” EX1004, Abstract. The Combination

additionally includes Locasto's explanation that "[c]ryptographic file systems provide **data** confidentiality by employing encryption to protect files against unauthorized access." EX1005, 1; *supra* §§V.A.1-3. Locasto then describes a cryptographic file system that "offers improved performance relative to a cipher with a larger key-schedule, while providing the same **security** properties." EX1005, 1.

Element_[1.1.1]:

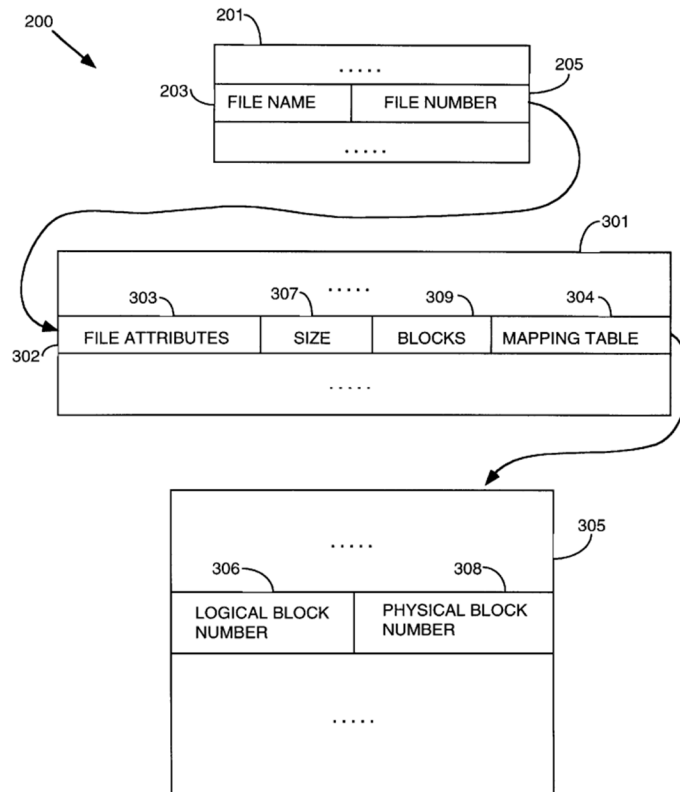
The Madany-Locasto Combination renders obvious Element_[1.1.1]. EX1003, ¶¶83-86. For example, the Combination includes **data block numbers** in the form of Madany's logical block numbers. EX1003, ¶83.



Madany-Locasto Combination: Visual Aid

A POSITA would have understood or found obvious that logical block numbers are *data block numbers* at least since logical block numbers correspond to data managed by a file system and stored on a physical disk. EX1003, ¶84. In other words, as would have been clear to a POSITA, Madany's logical *block numbers* are used by the file system to store and access *data*. EX1003, ¶84.

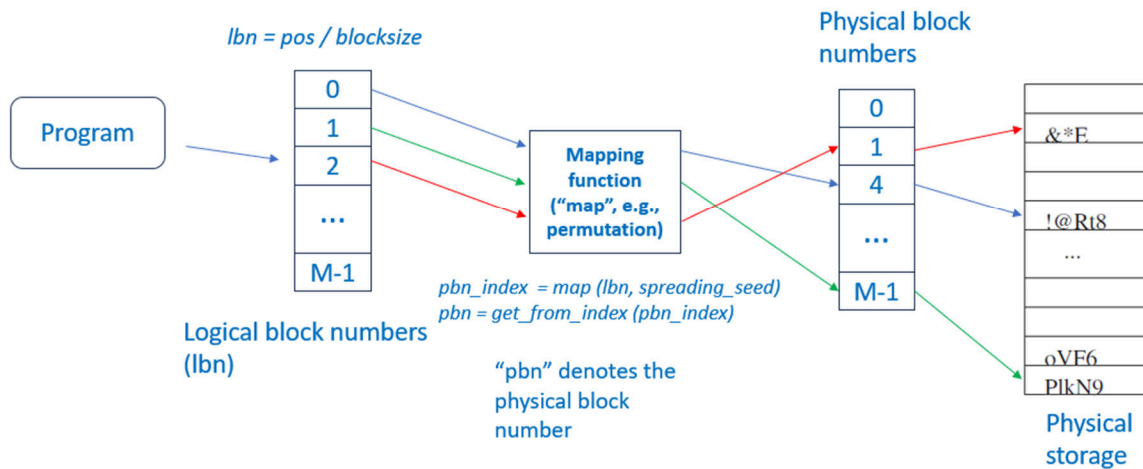
The Combination further teaches or renders obvious that the logical block numbers (*block numbers*) are *associated with a file*. EX1003, ¶85. For example, Madany explains that “[t]o effectively support block I/O, a file system typically divides the array of bytes in a given file into a set of fixed size logical blocks.” EX1004, 2:33-35. Thus, it would have been understood by or obvious to a POSITA that the logical *block numbers are associated with a file* since the file is assigned to the logical blocks corresponding to those logical block numbers. EX1003, ¶85. Madany depicts the association in Figure 1 (below), which shows information about the file (e.g., file name 203, file number 205) linked to the file's file table 301, which is linked to the file's mapping table 305 containing logical block numbers 306.



EX1004, Figure 1.

As discussed at Sections V.A.1-3, the Madany-Locasto Combination *applies a mapping function* to logical block numbers (*data block numbers*). EX1003, ¶86. For example, Locasto explains that “a pseudo-random permutation (PRP) function takes the calculated” logical block number “and the bits of the spreading seed to return a” physical block number. EX1005, 4-5. Locasto provides the formula in which the physical block number is computed by applying two functions, *map* and *get_from_index*, such that the results of *map(logical_block_number, spread seed)* are used in the function *get_from_index()*. EX1005, 5; EX1003, ¶86. The process is illustrated in the visual aid below. EX1003, ¶86.

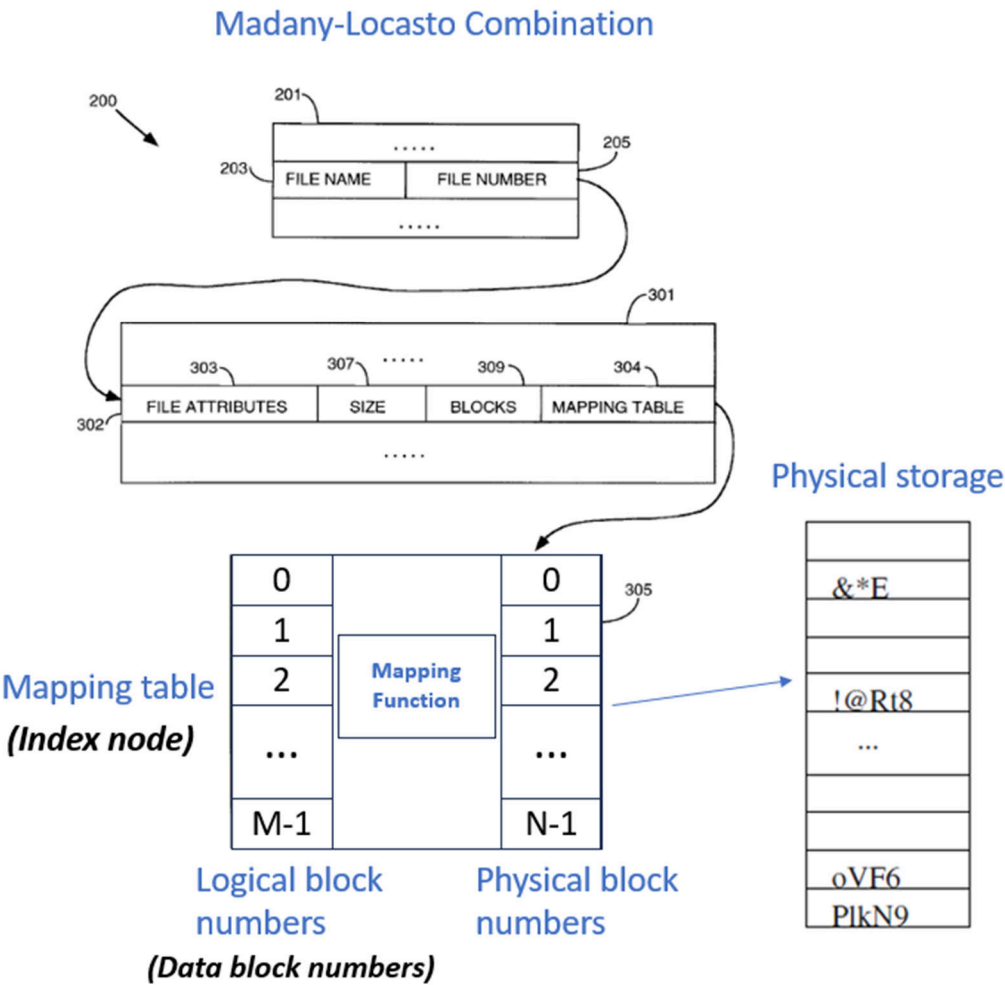
Madany-Locasto Combination



Madany-Locasto Combination: Visual Aid

Element_[1.1.2]:

The Madany-Locasto Combination renders obvious Element_[1.1.2]. EX1003, ¶¶87-92. As discussed at Element_[1.1.1], the Combination includes logical block numbers (*data block numbers*) *contained* in a mapping table. EX1003, ¶87. “The mapping table *contains* information on how the *logical blocks* within a file are mapped to physical blocks on a disk.” EX1004, 5:52-54 (w/o reference numbers); *see also* 5:29-7:65, Figures 1-4. For example, Madany explains, the first entry in the mapping table “maps to the physical block number for logical block number 0 of the file,” and so on. EX1004, 5:58-66 (w/o reference numbers).



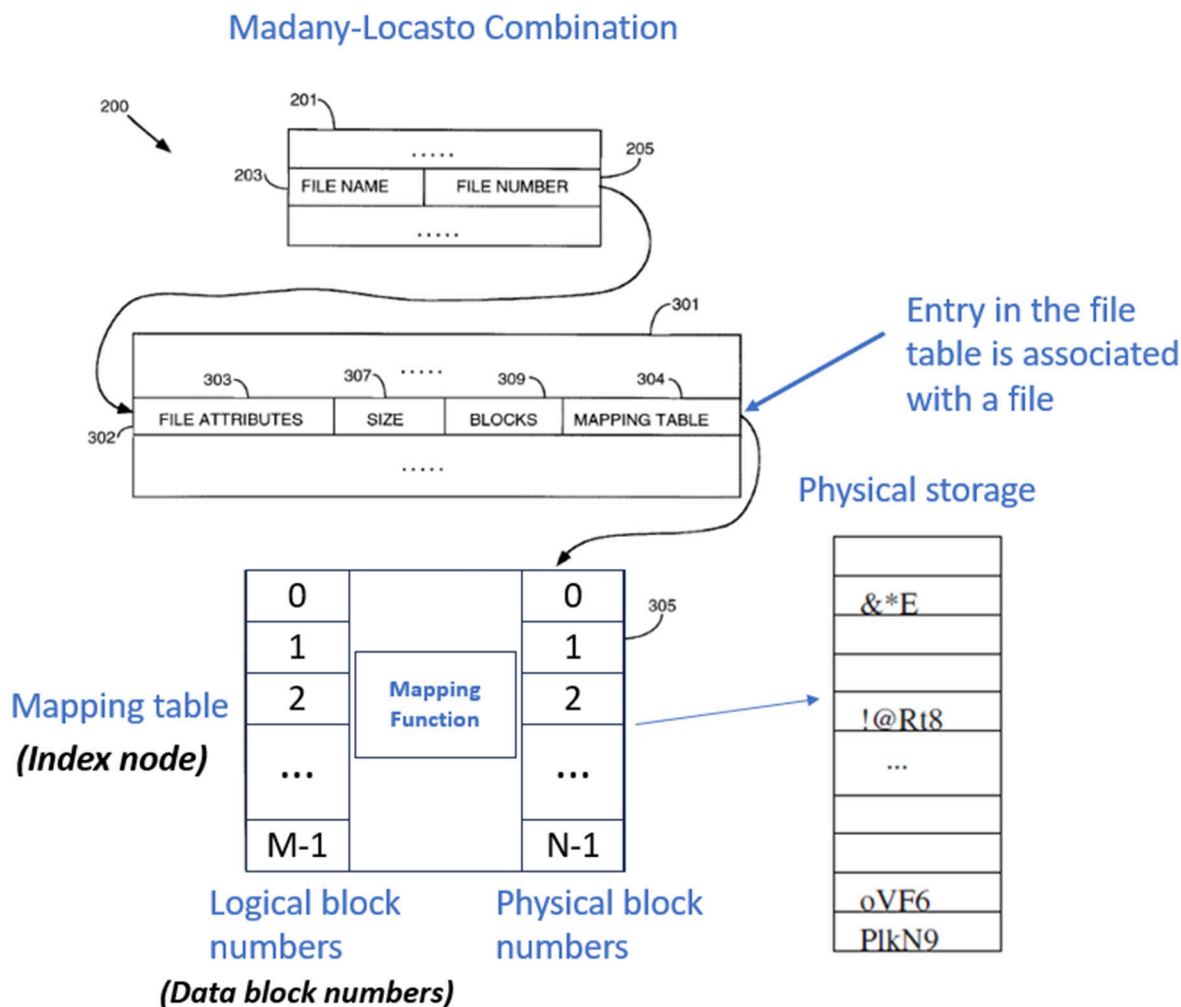
Madany-Locasto Combination: Visual Aid

As would have been understood by or obvious to a POSITA from Madany’s teachings, the mapping table serves as *an index node* by containing a mapping from a first index (logical block numbers) to a second index (physical block numbers). EX1003, ¶88.

Alternatively, while Madany explains that the inode 302 contains a pointer 304 to the mapping table 305, it would have nevertheless been understood by or

obvious to a POSITA that the inode can include the mapping table itself (e.g., as implemented in the 4.3BSD Unix operating system), increasing efficiency by eliminating the need to follow a pointer to access the mapping table. EX1003, ¶¶89-90 (citing EX1006, Figure 7.6). This predictable variation is consistent with Madany's explanation that "[i]t should be appreciated that other structures are possible for a file system, and that many file systems may be suitable for use with the present invention." EX1004, 5:47-51; EX1003, ¶91. In this alternative, the inode (*index node*) contains the logical block numbers (*data block numbers*) since the inode contains the mapping table 305. EX1003, ¶91.

Further, as would have been understood by or obvious to a POSITA from Madany's teachings, the mapping table (as well as the inode) is *associated with a file*. EX1003, ¶92. For example, Madany explains that "[e]ach entry 302 in the file table 301 identifies a *file* of the file system and is commonly known as an Inode." EX1004, 5:35-37; *see also* 5:29-51. Madany also explains that "[t]here is a mapping table 305 for each entry [Inode 302] in the file table 301." EX1004, 5:31-32.



Madany-Locasto Combination: Visual Aid

Element_[1.2.1]:

The Madany-Locasto Combination renders obvious Element_[1.2.1]. EX1003, ¶¶93-95. As discussed at Sections V.A.2-3, the Madany-Locasto Combination applies Locasto’s *mapping function*, e.g., “a pseudo-random permutation (PRP) function [that] takes [Madany’s logical block number] and the bits of the spreading seed to return a new index value.” EX1005, 4-5.

pbn_index = map(lbn, spreading_seed)

The physical “block number is then derived from this new index.” EX1005, 5; EX1003, ¶¶93-94.

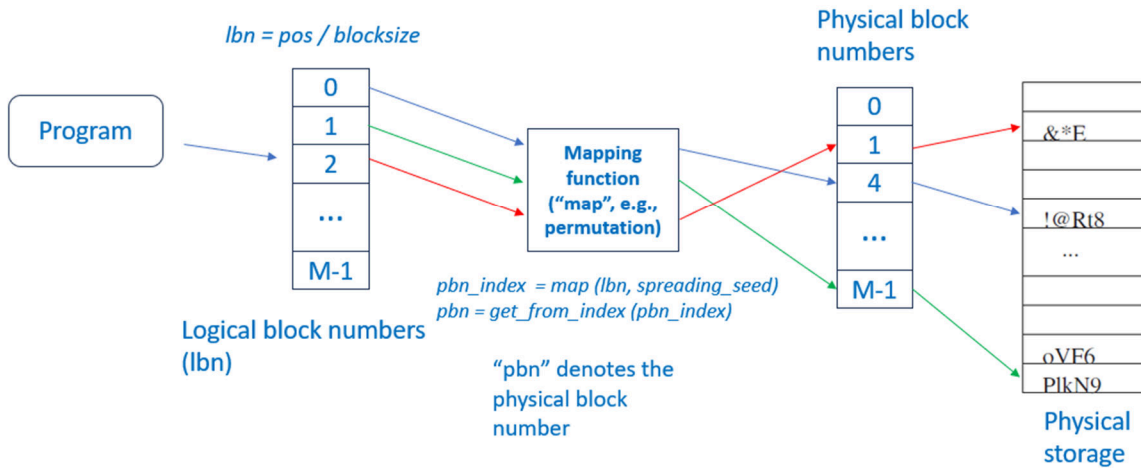
pbn = get_from_index(pbn_index)

Id. Thus, in the Madany-Locasto Combination, the physical block numbers (***mapped data block numbers***) ***are obtained after applying*** the “map” (***mapping***) ***function***. EX1003, ¶95.

Element_[1.2.2]:

The Madany-Locasto Combination renders obvious Element_[1.2.2]. EX1003, ¶¶96-97. For example, in the Combination, “the physical block number [(***mapped data block numbers***)] identifies the location [(***addresses of data of the file***)] on the disk [(***storage device***)] where the logical block is to be stored.” EX1004, 5:63-64; *see also* 2:34-39 (a file system “divides the array of bytes in a given ***file*** into a set of fixed size logical blocks), 2:40-46 (“The ***data*** in each logical block is stored in a physical block on the disk,” and “[a] physical block is the location on the disk to which the file system maps a logical block[.]”), 2:47–51 (“logical to physical block mappings” provide “the physical block ***address*** for [the corresponding] logical block”); EX1003, ¶97; *see also* EX1005, 4-5.

Madany-Locasto Combination



Madany-Locasto Combination: Visual Aid

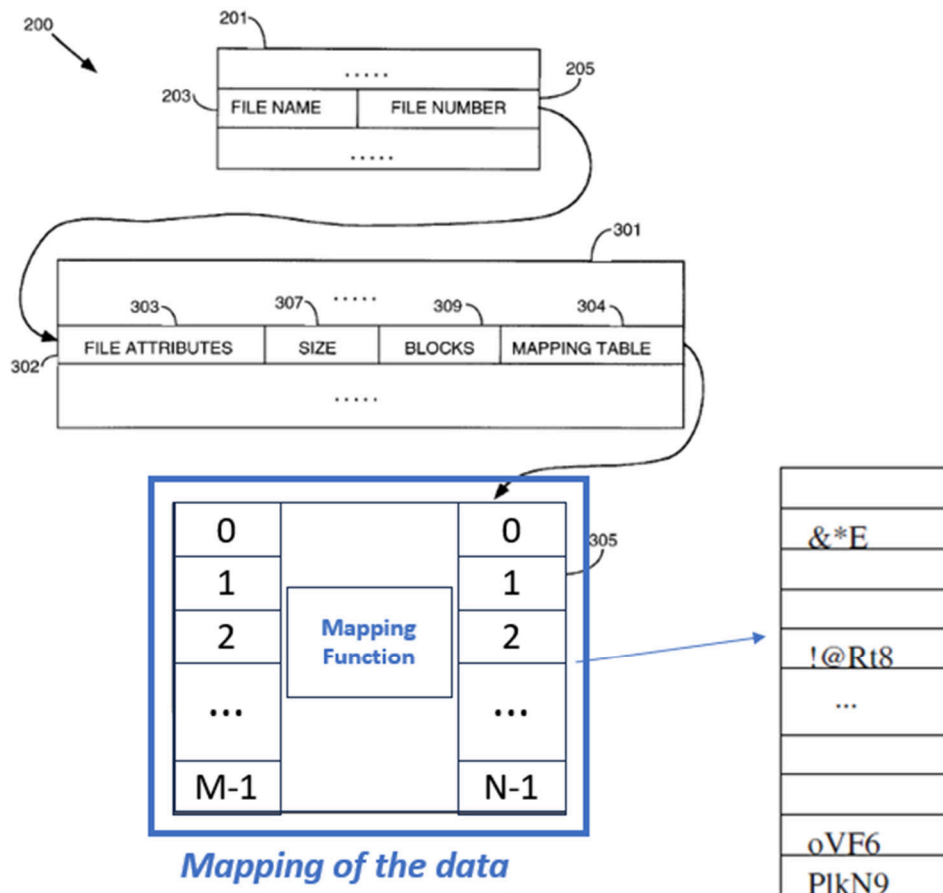
(b) Claim 2

Claim_[2]:

The Madany-Locasto Combination renders obvious Claim_[2]. EX1003, ¶¶98-100. For example, the notion of *storing a mapping* would have been obvious to a POSITA in view of Madany's teachings pertaining to an updatable "*mapping table*" where the Madany-Locasto Combination maintains (*stores*) a *mapping* between logical block numbers (*data block numbers*) and physical block numbers (*mapped data block numbers*). EX1004, Abstract, 5:52-60, Figure 1; EX1003, ¶98. Moreover, as discussed, in the Madany-Locasto Combination, the conversion from logical to physical block numbers involves applying Locasto's *mapping* function,

which a POSITA would have understood to be *stored* as coded instructions for performing the spreading algorithm. EX1005, 4-5; EX1003, ¶¶99-100 (citing EX1001; EX1014).

Madany-Locasto Combination



*Mapping of the data
block numbers and the
mapped data block
numbers*

Madany-Locasto Combination: Visual Aid

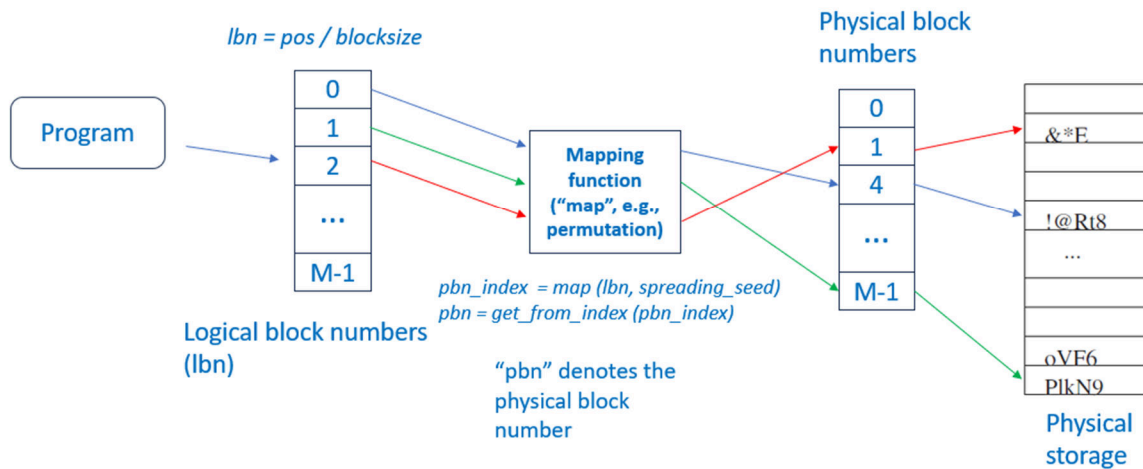
(c) **Claim 5**

Claim_[5]:

The Madany-Locasto Combination renders obvious Claim_[5]. EX1003, ¶¶101-103. For example, as discussed at Sections V.A.3, V.A.6 (Element_[1.1.1]), the Madany-Locasto Combination *applies a mapping function* to logical block numbers (*data block numbers*) *of a file*. EX1005, 4-5. The rudimentary concept of performing this technique at *each* logical block number would have been understood by or obvious to a POSITA. EX1003, ¶101. First, when each data block of a file is written, the mapping is performed on the data block number of that data block. EX1005, 4; EX1003, ¶101. Therefore, since all of the data blocks are written, the mapping function is applied for each data block number of the file. EX1003, ¶101. Second, the mapping function is applied to each data block number for the purpose of retrieving the complete set of data blocks for the file and, at least in some instances, entire files will be read.⁴ EX1003, ¶101.

⁴ To the extent Patent Owner argues Madany does not map each logical block number to a physical block number during compression, any such argument fails because Madany teaches that some files are not compressed. EX1004, 10:57-11:4.

Madany-Locasto Combination



Madany-Locasto Combination: Visual Aid

As discussed, the **mapping function** corresponds to the following **equation**:

$$pbn = get_from_index(map(lbn, spreading_seed))$$

EX1003, ¶102; §V.A.3, *supra*. Moreover, Locasto teaches that the mapping function includes a pseudo-random permutation function based on the well-known cipher algorithms “RC6” and “RC4,” both of which include multiple mathematical operations. EX1005, 4-5, 10; EX1003, ¶103 (citing EX1014).

(d) Claim 6

Claim [6]:

The Madany-Locasto Combination renders obvious Claim [6]. EX1003, ¶¶104-106. For example, regarding the **mapping function** Locasto explains that a “variable length block cipher is utilized as a pseudo-random permutation (PRP) to

map sequential block indices to ostensibly random indices.” EX1005, 10. By “variable length,” Locasto refers to a “variable-key length,” where “larger key lengths imply higher resistance against specific types of attacks” (higher *security level*) and a higher *complexity level* by “requiring more rounds of processing to spread the influence of the key.” EX1005, 1; EX1003, ¶104. At the same time, Locasto says, variations that increase the security level often present a “trade-off [with] performance” (*efficiency*). EX1005, 1, 11 (explaining that “performance” is “measured by the amount of time to read or write a file”); EX1003, ¶104. For the CamouflageFS prototype, Locasto *selected* a “28-bit” key and noted that “RC6 was chosen [(*selected*)]...because its construction makes it applicable to small block sizes and RC4 was utilized due to its simplicity.” EX1005, 10. Locasto also indicates the further design choice of *selecting* different key lengths—e.g., “128 bit keys” or “56 bit keys”—and different cipher algorithms. EX1005, 10 (“any existing block cipher and stream cipher”).

Thus, Locasto teaches the POSITA that key length and cipher algorithms are *selectable* variables that define *the complexity level* of the *mapping function* and thus *set the security level and efficiency of the mapping function*. EX1003, ¶106.

(e) **Claim 7**

Claim_[7]:

The Madany-Locasto Combination renders obvious Claim_[7]. EX1003, ¶¶107-109. As a threshold matter, because this dependent claim merely recites a conditional method step (“if”), the Board can find it obvious for the same reasons as the base independent claim. *See Cybersettle, Inc. v. Nat’l Arbitration Forum, Inc.*, 243 F. App’x 603, 607 (Fed. Cir. 2007).

Regardless, even under a narrower interpretation, Claim_[7] is obvious. EX1003, ¶107. For example, Locasto makes clear that its data-security spreading and encryption techniques “can coexist with normal file operations and does not require any extra work to use regular Ext2 files.” EX1005, 8. Thus, it would have been understood by or obvious to a POSITA that some operations proceed normally—i.e., without spreading or encryption (*if the mapping function is not applied to the data block numbers*). EX1005, 8; EX1003, ¶107. The POSITA would also have understood that, when a computer is first used, a file will be assigned to contiguous data blocks. EX1003, ¶108 (citing EX1006, 200). In other words, *the data block numbers are addresses of data of the file in the storage device*. EX1003, ¶109.

(f) **Claim 8**

Claim_[8]:

The Madany-Locasto Combination renders obvious Claim_[8]. EX1003, ¶¶110-111. As discussed, the Madany-Locasto Combination ***prevents copying*** because the file data are stored in mapped data blocks. EX1003, ¶110; *supra* §V.A.3. For example, per Locasto, the Combination uses “the extra bits of the cryptographic key to spread the file data throughout its address space and use the primary key material to encrypt that data.” EX1005, 3. “By combining this spreading function with random data for unallocated blocks,” the Combination “prevent[s] an attacker from knowing which blocks to perform a brute force search on.” *Id.* In addition, “[t]o maintain this illusion of a larger file without actually allocating it on disk, [the Combination] return[s] consistently random data on read() operations that are not accompanied by the proper cryptographic key.” *Id.* Therefore, the Combination’s ***mapping function prevents copying of data of the file in the storage device***. EX1001, 4:34-38; EX1003, ¶111.

(g) **Claim 9**

Element_[9.pre]:

The Madany-Locasto Combination renders obvious Element_[9.pre]. *Supra* §V.A.6, Element_[1.pre] (***data security for use with a file system***); EX1004, Abstract (“a method, ***apparatus***, and computer-usable medium”); 16:66-17:30

(describing “a typical computer system” including a CPU, ROM, RAM, etc.); EX1003, ¶112.

Element_[9.1.1]:

The Madany-Locasto Combination renders obvious Element_[9.1.1]. *Supra* §V.A.6, Element_[1.1.1] (discussing *apply a mapping function to data block numbers*). Additionally, it would have been understood by or obvious to a POSITA that the Madany-Locasto Combination provides *a block distribution engine*. EX1003, ¶113. Indeed, an “engine” in this context merely connotes “the part of a program that implements a special technique.” EX1003, ¶113 (citing EX1007, 3). Here, the technique is *block distribution*, which the Madany-Locasto Combination provides via Locasto’s mapping function, which spreads (*distributes*) blocks across an address space. *Supra* §V.A.6, Elements_[1.1.1]-[1.2.2]; EX1003, ¶113.

Element_[9.1.2]:

Supra §V.A.6, Element_[1.1.2]; EX1003, ¶114.

Element_[9.1.3]:

Supra §V.A.6, Element_[1.2.1]; EX1003, ¶115.

Element_[9.2]:

Supra §V.A.6, Element_[1.2.2]; EX1003, ¶116.

(h) Claim 10

Claim_[10]:

Supra §V.A.6, Claim_[2]; EX1003, ¶117.

(i) Claim 13

Claim_[13]:

Supra §V.A.6, Claim_[5]; EX1003, ¶118.

(j) Claim 14

Claim_[14]:

Supra §V.A.6, Claim_[6]; EX1003, ¶119.

(k) Claim 15

Claim_[15]:

Supra §V.A.6, Claim_[7]; EX1003, ¶120.

(l) Claim 16

Claim_[16]:

Supra §V.A.6, Claim_[8]; EX1003, ¶121.

(m) Claim 17

Element_[17.pre]:

Supra §V.A.6, Element_[9.pre]; EX1003, ¶123.

Element_[17.1.1]:

The Madany-Locasto Combination renders obvious Element_[17.1.1]. EX1003, ¶¶124-127. To start, the structure provided in the '686 patent's specification and described as corresponding to the claimed function of ***applying a mapping function to data block numbers that are associated with a file*** is a “a computer” executing an “operating system” that includes a “block distribution engine” performing a “distribution algorithm,” such as “ $f(x)=5x$,” to block numbers. EX1001, 2:1-14, 2:62-4:3, Figures 1-2; EX1003, ¶125; *see also* ¶122.

As discussed at Element_[1.1.1], the Madany-Locasto Combination describes and/or renders obvious the function of ***applying a mapping function to data block numbers that are associated with a file***. And, as discussed at Element_[9.1.1], the Madany-Locasto Combination describes and/or renders obvious the aspect of the corresponding structure that involves a “block distribution engine.” Moreover, like the '686 patent, Madany and Locasto each describe implementations in the context of a computer operating system. *E.g.*, EX1004, 1:45-52, 5:29-51, 16:66-17:46 EX1005, 1, 8-11; EX1003, ¶126.

As for the “distribution algorithm,” Locasto describes “a pseudo-random permutation (PRP) function” based on the well-known cipher algorithms “RC6” and “RC4,” while the '686 patent describes an example using the simple algebraic expression “ $f(x)=5x$.” *Compare* EX1005, 4-5, 10, *with* EX1001, 3:50-4:3.

Nevertheless, the differences between Locasto's algorithm and the '686 patent's algorithm are insubstantial. EX1003, ¶127. *See Odetics, Inc. v. Storage Tech. Corp.*, 185 F.3d 1259, 1268 (Fed. Cir. 1999). What matters is that Locasto's algorithm (within the Combination) and the '686 patent's algorithm work at least substantially the same way—i.e., by converting block numbers derived from an inode into mapped block numbers spread throughout an address space—to achieve at least substantially the same result—i.e., hiding the location of data on the storage disk from anyone unable to repeat the algorithm. EX1003, ¶127.

Element_[17.1.2]:

Supra §V.A.6, Element_[1.1.2]; EX1003, ¶128.

Element_[17.2.1]:

The Madany-Locasto Combination renders obvious Element_[17.2.1]. EX1003, ¶¶129-130. To start, the structure provided in the '686 patent's specification and described corresponding to the claimed function of ***obtaining mapped data block numbers after applying the mapping function*** is a “a computer” executing an “operating system” performing that function. EX1001, 2:1-14, 2:62-4:3, Figures 1-2; EX1003, ¶129; *In re Katz Interactive Call Processing Patent Litigation*, 639 F.3d 1303, 1316 (Fed. Cir. 2011) (a patent need not disclose anything beyond “the general purpose processor” for basic functions like “processing,” “receiving,” and “storing” data).

As discussed at Element_[1.2.1], the Madany-Locasto Combination describes and/or renders obvious the function of *obtaining mapped data block numbers after applying the mapping function*. And, as discussed at Elements_[9.1.pre] and [17.1.1], the Madany-Locasto Combination describes and/or renders obvious the aspect of the corresponding structure that involves a computer with a processor executing an operating system.

Element_[17.2.2]:

Supra §V.A.6, Element_[1.2.2]; EX1003, ¶131.

(n) Claim 18

Elements_[18.pre]-[18.1]:

The Madany-Locasto Combination renders obvious Elements_[18.pre]-[18.1]. For example, the Combination includes Madany’s description of a “method, apparatus ([*article of manufacture*]), and computer-usable [*machine-readable*]) medium for compressing data in a file system utilizing the concept of ‘holes’.” EX1004, Abstract; EX1003, ¶132.

Element_[18.2.1]:

Supra Element_[1.1.1]; EX1003, ¶133.

Element_[18.2.2]:

Supra Element_[1.1.2]; EX1003, ¶134.

Element_[18.3.1]:

Supra Element_[1.2.1]; EX1003, ¶135.

Element_[18.3.2]:

Supra Element_[1.2.2]; EX1003, ¶136.

B. GROUND 1B: Claims 1-2, 5-10, 13-18 Are Obvious Over Madany-Locasto

1. Madany (EX1004)

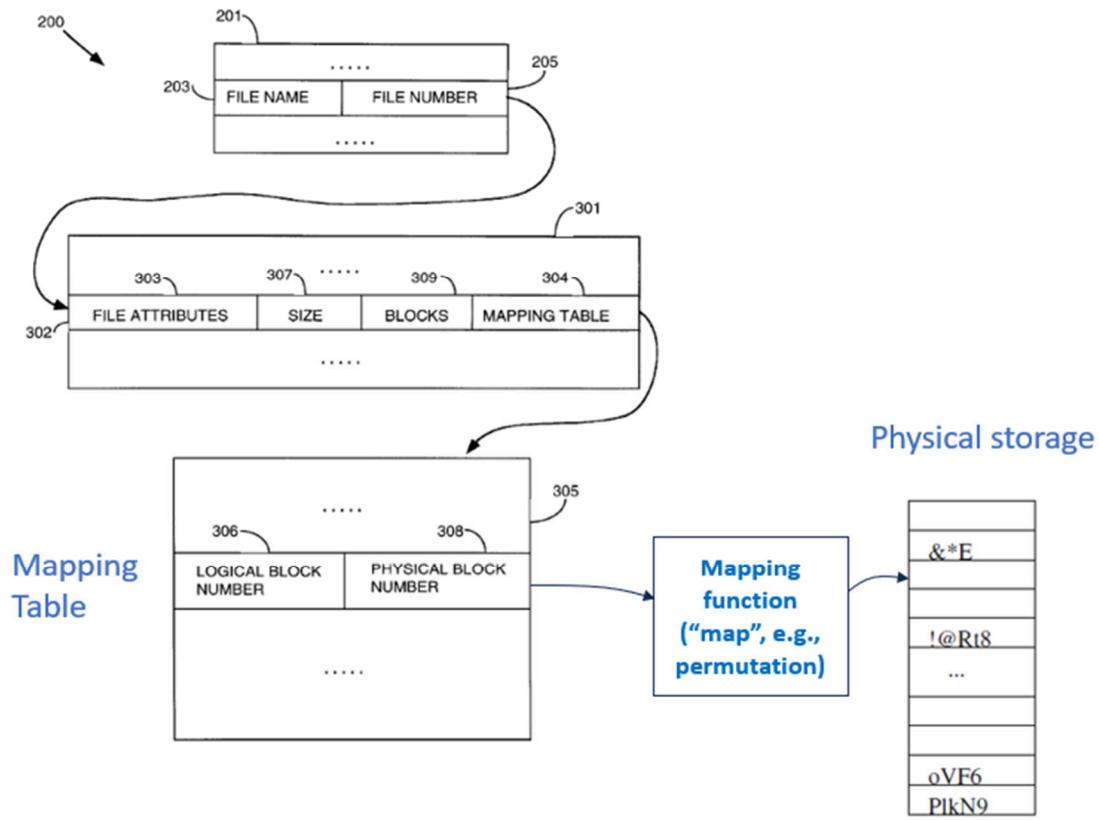
Supra §V.A.1; EX1003, ¶137.

2. Locasto (EX1005)

Supra §V.A.2; EX1003, ¶138.

3. The Combination

As discussed at Section V.A.3, a POSITA would have understood that Madany's file system architecture presents two apparent options for incorporating Locasto's mapping function, including using Locasto's techniques to map the physical block numbers stored in Madany's mapping table to addresses on the physical storage disk, e.g., as illustrated in the visual aid below and as described here in Ground 1B. EX1003, ¶¶139-143.



Madany-Locasto Combination: Visual Aid

In operation, when a program accesses a file at a given position, the Madany-Locasto Combination of Ground 1B determines the inode and then uses a mapping table pointer in the inode to determine the mapping table. EX1004, 5:38-40, 48-49; EX1003, ¶140. Per Madany, the Combination next maps logical block numbers to physical block numbers. EX1004, 5:58-64; EX1003, ¶141. After that, as shown in the visual aid above, the Combination converts the physical block numbers in Madany's mapping table to physical block numbers that are spread, thus providing security. EX1005, 5-6; EX1003, ¶141. More specifically, Madany's physical block

numbers are spread using a mapping function, e.g., Locasto's "map" function which can be a pseudo-random permutation function and a "spreading seed" (*spread_seed*). EX1005, 4-5; EX1003, ¶141. The function is shown below:

$$\textit{spread_pbn_index} = \textit{map}(\textit{pbn}, \textit{spreading_seed})$$
$$\textit{spread_pbn} = \textit{get_from_index}(\textit{spread_pbn_index})$$

EX1005, 5; EX1003, ¶141. These two functions can be expressed equivalently as the single mapping function:

$$\textit{spread_pbn} = \textit{get_from_index}(\textit{map}(\textit{pbn}, \textit{spreading_seed}))$$

EX1003, ¶142. Thus, the mapping function "randomly distribute[s] the real file data...throughout a large address space so that an attacker would have to first guess the spreading schedule before he attempts a brute force search on the rest of the key." EX1005, 4.

4. Motivation to Combine and Reasonable Expectation of Success

As discussed above at Ground 1A (§§V.A.4-5), a POSITA would have had multiple reasons (taken independently or collectively) to pursue the Madany-Locasto Combination with a reasonable expectation of success. All of those motivation rationales apply whether Locasto's mapping function is employed to

convert Madany's logical block numbers to physical block numbers, as above in Ground 1A, or to map the physical block numbers in Madany's mapping table to addresses on the physical storage disk, as here in Ground 1B. EX1003, ¶144.

5. Claim Element Analysis

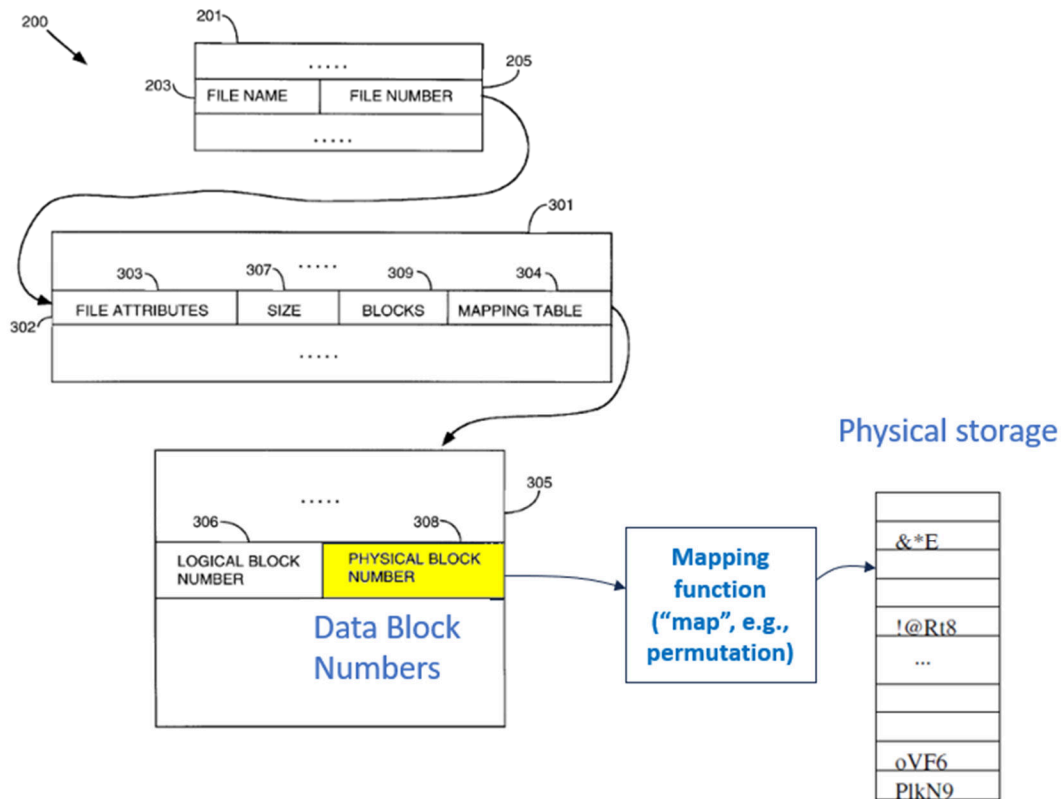
(a) Claim 1

Element_[1.pre]:

Supra §V.A.6, Element_[1.pre]; EX1003, ¶145.

Element_[1.1.1]:

The Madany-Locasto Combination renders obvious Element_[1.1.1]. EX1003, ¶¶146-148. For example, the Madany-Locasto Combination *applies a mapping function* to the physical block numbers (*data block numbers*) in the mapping table to obtain the physical addresses of the data blocks on the physical storage disk, as illustrated in the visual aid below. *Supra* §V.B.3; EX1003, ¶146.



Madany-Locasto Combination: Visual Aid

The Combination further teaches or renders obvious that the physical block numbers (*block numbers*) are *associated with a file*. EX1003, ¶147. For example, Madany explains that the “mapping table 305 contains information on how the logical blocks within a file are mapped to physical blocks on a disk,” and that the “physical block number identifies the location on the disk where the logical block [of the file] is to be stored.” EX1004, 5:52-54, 5:63-64. Thus, a POSITA would have understood that the physical *block numbers are associated with a file* since they are associated with the file’s logical blocks (via the mapping table) and with the file’s data stored on physical storage. EX1003, ¶147. Madany illustrates the

association in Figure 1, which shows information about the file (e.g., file name 203, file number 205) linked to the file's file table 301, which is linked to the file's mapping table 305. *Id.*

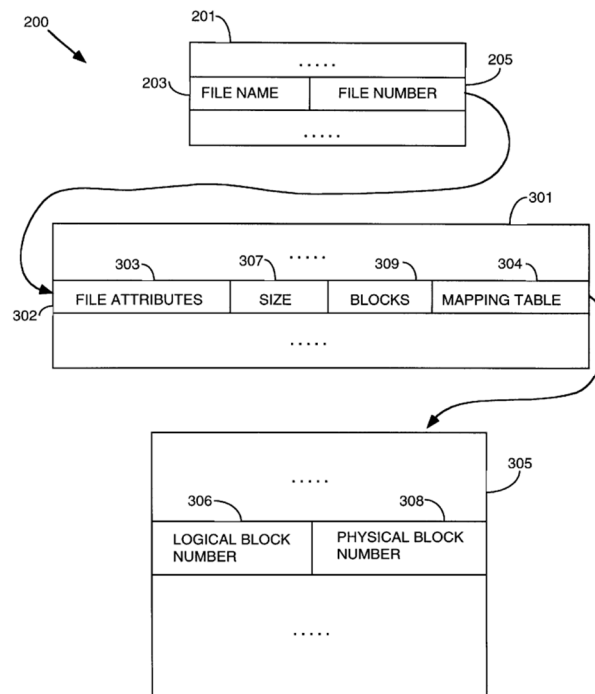


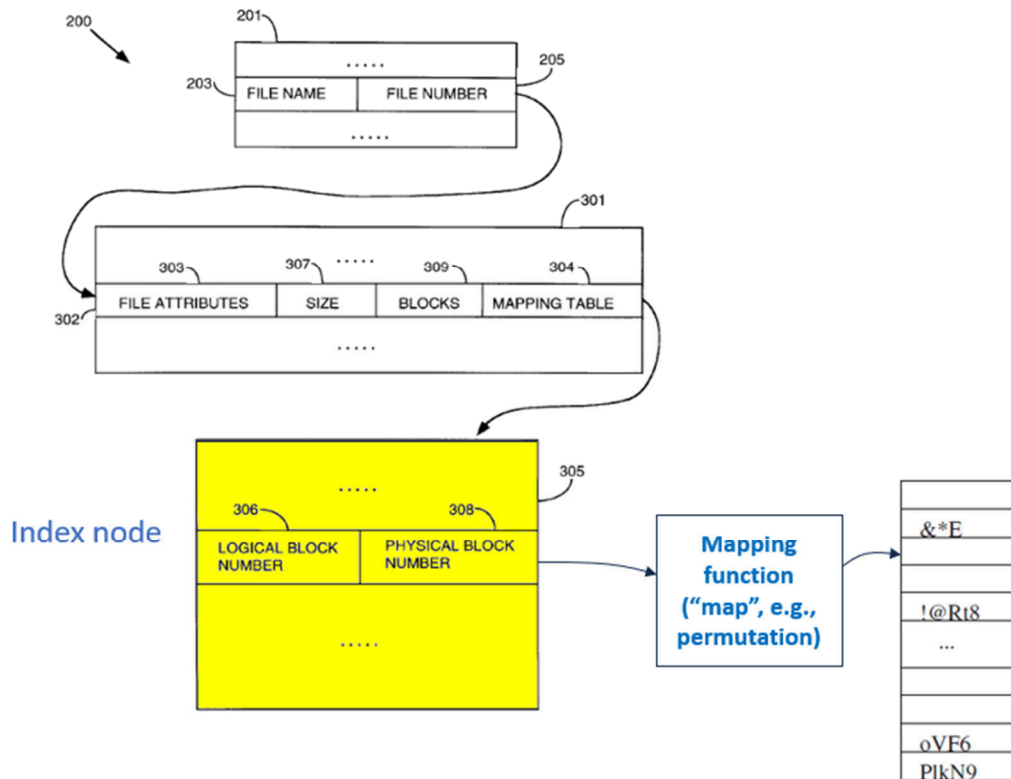
Figure 1

EX1004, Figure 1.

As discussed at Sections V.B.1-3, the Madany-Locasto Combination in Ground 1B *applies a mapping function* to physical block numbers (*data block numbers*). EX1003, ¶148. Specifically, the Combination’s “pseudo-random permutation (PRP) function takes the” physical block number “and the bits of the spreading seed to return a” spread physical block number, which is the physical address of the data block on the physical storage disk. EX1005, 4-5; EX1003, ¶148.

Element_[1.1.2]:

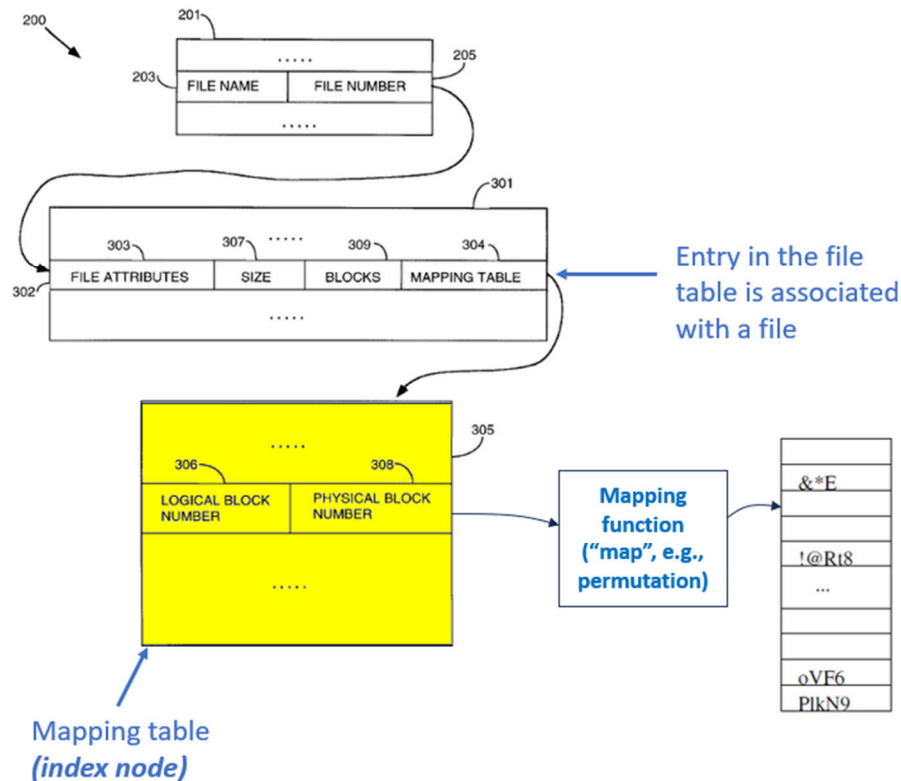
The Madany-Locasto Combination renders obvious Element_[1.1.2]. EX1003, ¶¶149-154. Madany makes clear that the physical block numbers (***data block numbers***) are ***contained*** in a mapping table (***index node***). EX1003, ¶149; *supra*, §V.A.6, Element [1.1.2]. For example, Madany explains that the first entry in the mapping table 305 “maps to the physical block number 308 for logical block number 0 of the file.” EX1004, 5:58-60.



Madany-Locasto Combination: Visual Aid

Madany further explains that “[e]ach entry 302 in the file table 301 identifies a ***file*** of the file system and is commonly known as an Inode.” EX1004, 5:35-37.

And “[t]here is a mapping table 305 for each entry in the file table 301.” EX1004, 5:31-32. Thus, as would have been understood by or obvious to a POSITA, the mapping table (*index node*) is *associated with a file*. EX1003, ¶150.



Madany-Locasto Combination: Visual Aid

Alternatively, while Madany explains that the Inode 302 contains a pointer 304 to the mapping table 305, it would have nevertheless been understood by or obvious to a POSITA that the inode can include the mapping table itself (e.g., as implemented in the 4.3BSD Unix operating system), increasing efficiency by eliminating the need to follow a pointer to access the mapping table. EX1003, ¶¶151-153 (citing EX1006, Figure 7.6). This predictable variation is consistent with

Madany's explanation that "[i]t should be appreciated that other structures are possible for a file system, and that many file systems may be suitable for use with the present invention." EX1004, 47-51; EX1003, ¶154. In this alternative, the inode (*index node*) contains the physical block numbers (*data block numbers*) since the inode contains the mapping table 305. EX1003, ¶154.

Element_[1.2.1]:

The Madany-Locasto Combination renders obvious Element_[1.2.1]. EX1003, ¶¶155-157. As discussed at Section V.B.3, the Combination applies "a pseudo-random permutation (PRP) function [that] takes the [Madany's physical block number in Madany's mapping table/inode] and the bits of the spreading seed to return a new index value." EX1005, 4-5.

spread_pbn_index = map (pbn, spreading_seed)

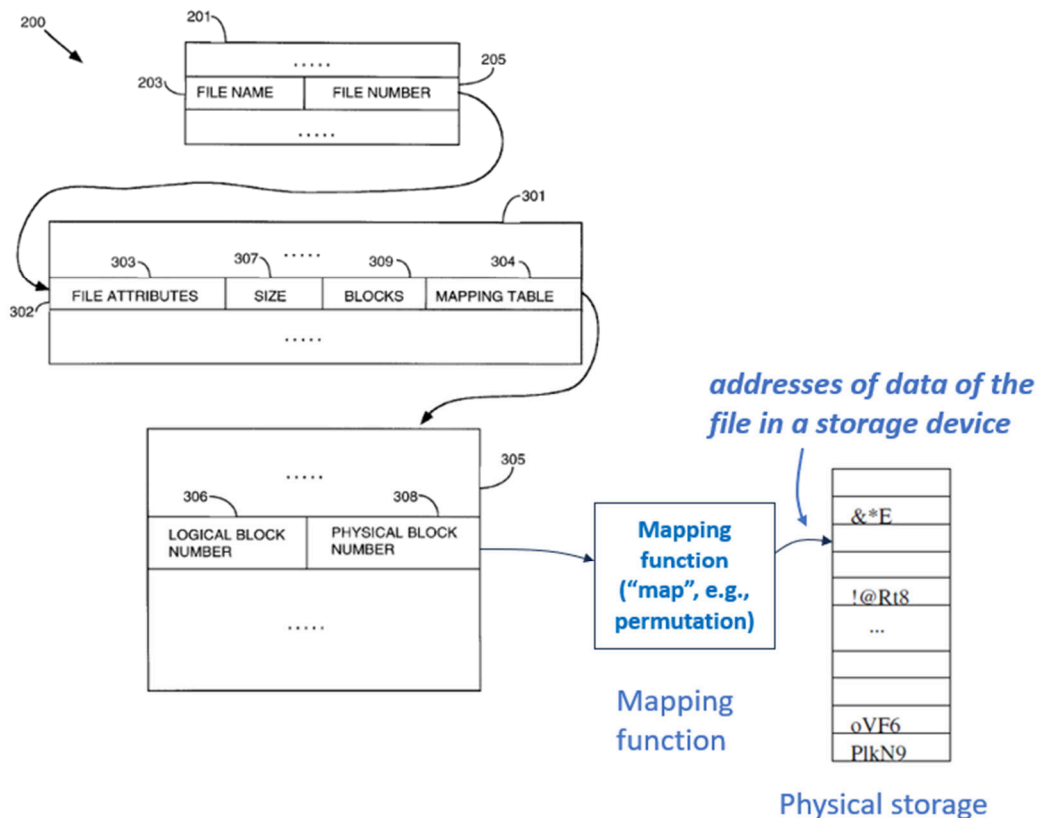
The spread physical "block number is then derived from this new index." EX1005, 5; EX1003, ¶156.

spread_pbn = get_from_index (spread_pbn_index)

Id. Thus, in the Madany-Locasto Combination, the spread physical block numbers (*mapped data block numbers*) are obtained after applying the "map" (*mapping*) *function*. EX1003, ¶157.

Element_[1.2.2]:

The Madany-Locasto Combination renders obvious Element_[1.2.2]. EX1003, ¶158. Specifically, it would have been understood by or obvious to a POSITA that the spread physical ***data block numbers are addresses*** of physical blocks on the disk (***storage device***) storing ***data of the file***. *Supra* §V.B.3; EX1003, ¶158. For example, Locasto explains that a block number determined by its mapping function is “the block on the disk” where the data at a particular file position resides. EX1005, 5; EX1003, ¶158.

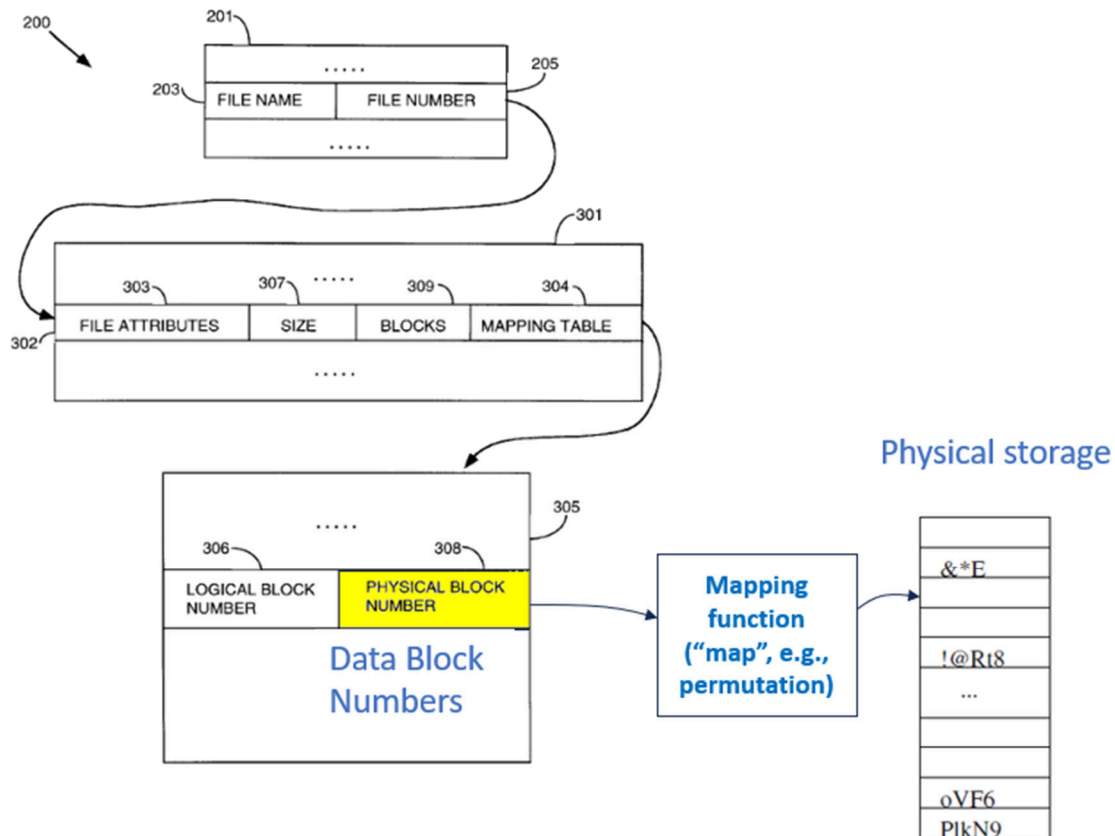


Madany-Locasto Combination: Visual Aid

(b) **Claim 2**

Claim [2]:

The Madany-Locasto Combination renders obvious Claim_[2]. EX1003, ¶¶159-162. To start, as discussed at Element_[1.1.1], the Madany-Locasto Combination applies a mapping function to the physical block numbers (***data block numbers***) to obtain the physical addresses of the data blocks on the physical storage disk, as illustrated in the visual aid below. See §V.B.3, *supra*; EX1003, ¶159.



Madany-Locasto Combination: Visual Aid

As discussed at Section V.B.3, in the Madany-Locasto Combination, spread (*mapped*) data block numbers are determined from data block numbers using the equation:

$$spread_pbn = get_from_index(map(pbn, spreading_seed))$$

EX1003, ¶160. Since the Madany-Locasto Combination computes the spread physical block numbers (*mapped data block numbers*) from the physical block numbers (*block numbers*), a POSITA would have understood that the Madany-Locasto Combination *stores the mapping* of physical block numbers (*block numbers*) to spread physical block numbers (*mapped block numbers*), e.g., at the conclusion of the spreading calculation. EX1003, ¶¶161-162.

(c) **Claim 5**

Claim_[5]:

The Madany-Locasto Combination renders obvious Claim_[5]. EX1003, ¶¶163-165. For example, as described in Sections V.B.3, V.B.5 (Element_[1.1.1]), the Madany-Locasto Combination *applies a mapping function* to physical block numbers (*data block numbers*) *of a file*. EX1005, 4-5. The rudimentary concept of performing this technique at *each* logical block number would have been understood by or obvious to a POSITA. EX1003, ¶163. First, when each data block of a file is written, the mapping is performed on the data block number of that data block.

EX1005, 4; EX1003, ¶163. Therefore, since all of the data blocks are written, the mapping function is applied for each data block number of the file. EX1003, ¶163. Second, the mapping function is applied to each data block number for the purpose of retrieving the complete set of data blocks for the file and, at least in some instances, entire files will be read.⁵ EX1003, ¶163.

As discussed at Section V.B.3, the *mapping function* corresponds to the following composite *equation*:

$$spread_pbn = map(get_from_index(pbn, spreading_seed))$$

EX1005, 5; EX1003, ¶164. Moreover, Locasto teaches that the mapping function includes a pseudo-random permutation function based on the well-known cipher algorithms “RC6” and “RC4,” both of which include multiple mathematical operations. EX1005, 4-5, 10; EX1003, ¶165.

(d) **Claim 6**

Claim_[6]:

Supra §V.A.6, Claim_[6]; EX1003, ¶166.

⁵ To the extent Patent Owner argues Madany does not map each physical block number to a physical block number during compression, such an argument fails because Madany’s teaches that some files are not compressed. EX1004, 10:57-11:4.

(e) **Claim 7**

Claim_[7]:

Supra §V.A.6, Claim_[7]; EX1003, ¶167.

(f) **Claim 8**

Claim_[8]:

Supra §V.A.6, Claim_[8]; EX1003, ¶168.

(g) **Claim 9**

Element_[9.pre]:

The Madany-Locasto Combination renders obvious Element_[9.pre]. *Supra* §V.B.5, Element_[1.pre] (*data security for use with a file system*); EX1004, Abstract (“a method, *apparatus*, and computer-usable medium”), 16:66-17:30 (describing “a typical computer system” including a CPU, ROM, RAM, etc.); EX1003, ¶169.

Element_[9.1.1]:

The Madany-Locasto Combination renders obvious Element_[9.1.1]. *Supra* §V.B.5, Element_[1.1.1] (discussing *apply a mapping function to data block numbers*). Additionally, it would have been understood by or obvious to a POSITA that the Madany-Locasto Combination provides *a block distribution engine*. EX1003, ¶170. An “engine” in this context merely connotes “the part of a program that implements a special technique.” EX1003, ¶170 (citing EX1007, 3). Here, the

special technique is ***block distribution***, which the Madany-Locasto Combination provides via Locasto's mapping function, which spreads (***distributes***) blocks across an address space. *Supra* §V.B.5, Elements_[1.1.1]-[1.2.2]; EX1003, ¶170.

Element_[9.1.2]:

Supra §V.B.5, Element_[1.1.2]; EX1003, ¶171.

Element_[9.1.3]:

Supra §V.B.5, Element_[1.2.1]; EX1003, ¶172.

Element_[9.2]:

Supra §V.B.5, Element_[1.2.2]; EX1003, ¶172(a).

(h) Claim 10

Claim_[10]:

Supra §V.B.5, Claim_[2]; EX1003, ¶173.

(i) Claim 13

Claim_[13]:

Supra §V.B.5, Claim_[5]; EX1003, ¶174.

(j) Claim 14

Claim_[14]:

Supra §V.B.5, Claim_[6]; EX1003, ¶175.

(k) Claim 15

Claim_[15]:

Supra §V.B.5, Claim_[7]; EX1003, ¶176.

(l) Claim 16

Claim_[16]:

Supra §V.B.5, Claim_[8]; EX1003, ¶177.

(m) Claim 17

Elements_[17.pre]-[17.1.1]:

Supra §V.B.5, Element_[1.1.1] (*applying a mapping function to data block numbers that are associated with a file*); *supra* §V.A.6, Elements_[17.pre]-[17.1.1] (*means for*); EX1003, ¶¶178-179.

Element_[17.1.2]:

Supra §V.B.5, Element_[1.1.2]; EX1003, ¶180.

Element_[17.2.1]:

Supra §V.B.5, Element_[1.2.1] (*obtaining mapped data block numbers after applying the mapping function*); *supra* §V.A.6, Element_[17.2.1] (*means for*); EX1003, ¶181.

Element_[17.2.2]:

Supra §V.B.5, Element_[1.2.2]; EX1003, ¶182.

(n) **Claim 18**

Elements_[18.pre]-[18.1]:

The Madany-Locasto Combination renders obvious Elements_[18.pre]-[18.1]. EX1003, ¶183. For example, the Combination includes Madany’s description of a “method, apparatus ([*article of manufacture*]), and computer-usable [*machine-readable*]) medium for compressing data in a file system utilizing the concept of ‘holes’.” EX1004, Abstract; EX1003, ¶183.

Element_[18.2.1]:

Supra §V.B.5, Element_[1.1.1]; EX1003, ¶184.

Element_[18.2.2]:

Supra §V.B.5, Element_[1.1.2]; EX1003, ¶185.

Element_[18.3.1]:

Supra §V.B.5, Element_[1.2.1]; EX1003, ¶186.

Element_[18.3.2]:

Supra §V.B.5, Element_[1.2.2]; EX1003, ¶187.

C. GROUND 2: Claims 1-18 Are Obvious Over Madany-Locasto-Leffler

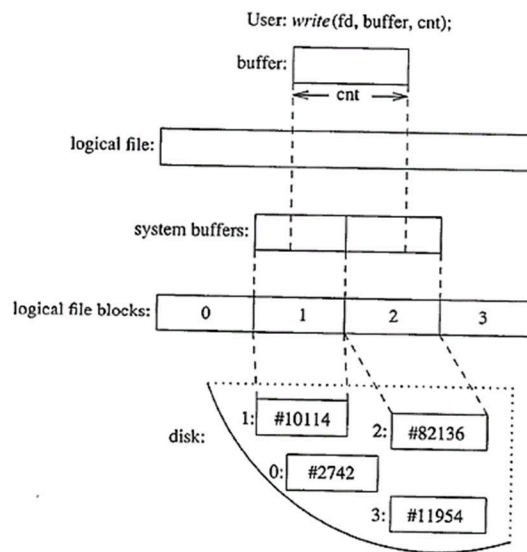
Leffler supplements Madany and Locasto with respect to certain rudimentary features regarding conventional file system operations and data structures. Integration of Leffler’s teachings does not disturb the aspects of Madany and Locasto mapped to Dependent Claims 2, 5-8, 10, and 13-16. To avoid repetition,

the analysis of these dependent claims is not repeated below because it is identical to Grounds 1A-1B and fully incorporated here in Ground 2. From Grounds 1A-1B, only Independent Claims 1, 9, 17, and 18 are addressed below. Dependent Claims 3-4 and 11-12, which are not challenged in Grounds 1A-1B, are also addressed below based on Leffler's complementary teachings. EX1003, ¶¶188-223.

1. Leffler (EX1006)

(a) Summary of Leffler's Pertinent Disclosure

Leffler is a textbook entitled "*The Design and Implementation of the 4.3BSD UNIX Operating System.*" Leffler provides a "full-length description of the design and implementation of the research versions of the UNIX system," with "[m]ost detail [] given about 4.3BSD." EX1006, vii; EX1003, ¶¶188-195. Leffler's Figure 7.5 shows the "work required to access the filesystem on a disk." EX1006, 192.



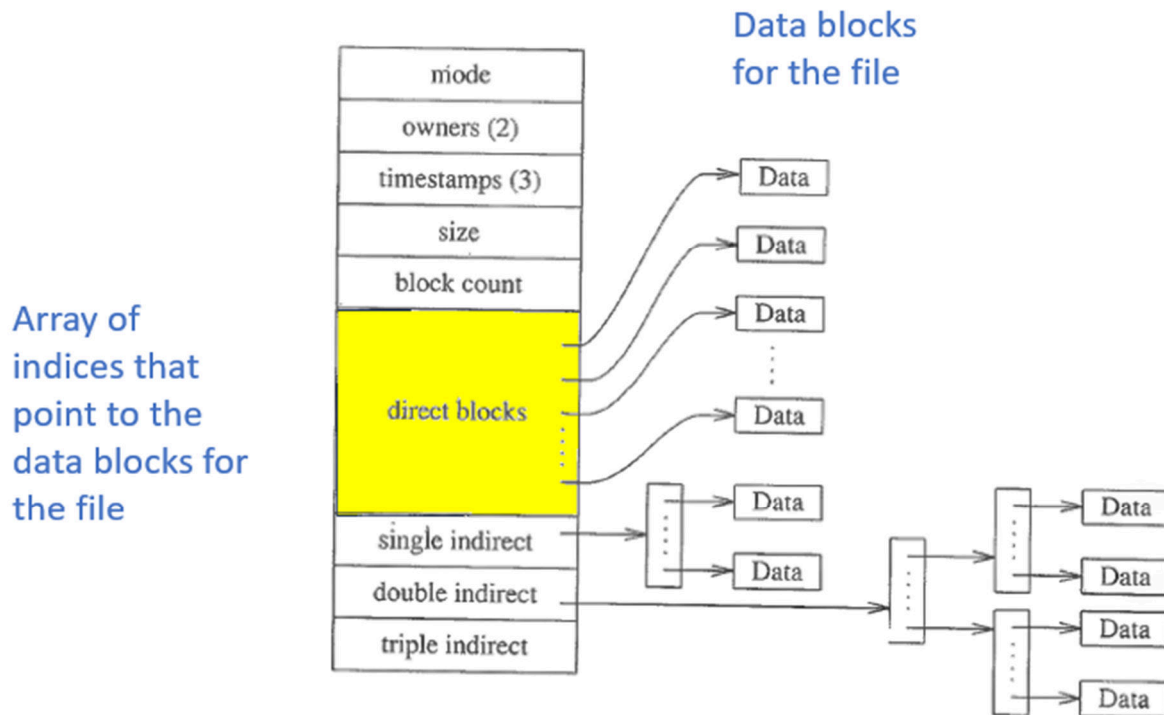
EX1006, Figure 7.5

The user creates an array of bytes (“buffer”) and issues a “write” command that indicates the number of bytes (“cnt” for “count”) from the buffer to be written. EX1006, 192, Figure 7.5; EX1003, ¶190. The user also provides a file descriptor (“fd,”) indicating the file to which the “write” command applies. *Id.*

To service the “write” operation, “the system breaks the user’s request into a set of operations to be done on each logical block of the file,” where “[l]ogical blocks describe block-sized pieces of a file.” EX1006, 192. “The system calculates the logical blocks by dividing the array of bytes into filesystem-sized pieces,” e.g., “if a filesystem’s block size is 8192 bytes, then logical block 0 would contain bytes 0 to 8191, logical block 1 would contain bytes 8192 to 16,383, and so on.” *Id.*

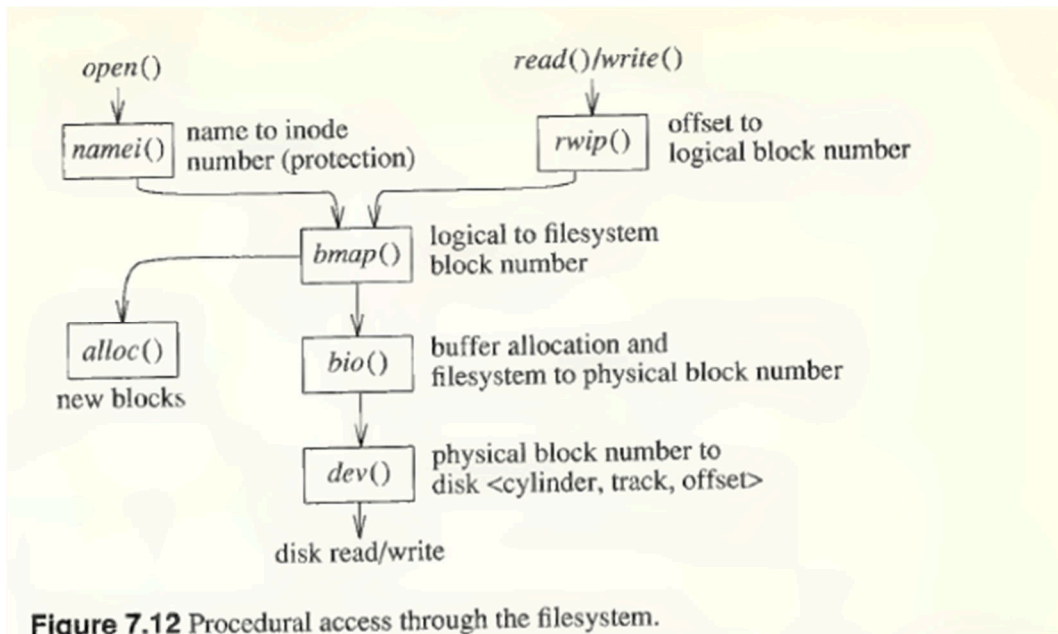
“The data in each logical block is stored in a physical block on the disk,” which “is the location on the disk to which the system maps a logical block.” EX1006, 193. “Although the contents of a logical block are contiguous on disk, the logical blocks of the file need not be laid out contiguously.” EX1006, 193. To enable mapping of logical blocks to physical blocks, “[e]very file has an associated inode that contains information describing access permissions, ownership of the file, timestamps marking last modification and access times for the file, and an array of indices that point to the data blocks for the file.” EX1006, 189; EX1003, ¶193. As described above, the system calculates the logical block number for the file operation and uses that logical block number as an index into the “array of indices that point

to the data blocks for the file” to determine the physical block. EX1006, 189, 192; EX1003, ¶193.



EX1006, Figure 7.6 (annotated); EX1003, ¶193.

When a program reads or writes data from disk, the operating system performs the operations illustrated in Figure 7.12, reproduced below and detailed in Dr. Medvidovic’s declaration. See EX1003, ¶¶194-195.



EX1006, 207 (Figure 7.12).

(b) Analysis of Leffler's Public Availability

Leffler was originally published in 1989 and re-published with corrections in 1990 by Addison-Wesley Publishing Company, a well-known and longstanding publisher of math, science, and computer science literature at the time. *See* EX1033; EX1034; EX1035. Leffler's front matter lists Library of Congress cataloging data and a 1989 copyright date. EX1006, 1-4. The Library of Congress's catalog includes an entry matching the details provided in Leffler's front matter and corroborating the initial 1989 publication year. *Compare* EX1006, 4, *with* EX1036, 1-2. Cataloging dates extracted from various other library MARC records further corroborate the public availability of Leffler, as do various technical papers and patents that cite to Leffler. *E.g.*, EX1032, ¶¶4-13; EX1037, 11 (reference [13]);

EX1038, 1:39-48; EX1004, 1:47-50 (Madany citing Leffler). Notably, whether Leffler is deemed publicly accessible as of 1989 or 1990 is not dispositive because either date is several years before the 2007 Critical Date.

2. Claim Element Analysis

(a) Claims 1, 9, 17-18

The following analysis applies Leffler's supplemental teachings to provide an alternative basis for satisfying specific elements of Independent Claims 1, 9, 17-18. Ground 2 incorporates the analysis of Grounds 1A-1B for all other elements.

Element_[1.1.2]:

Element_[9.1.2]:

Element_[17.1.2]:

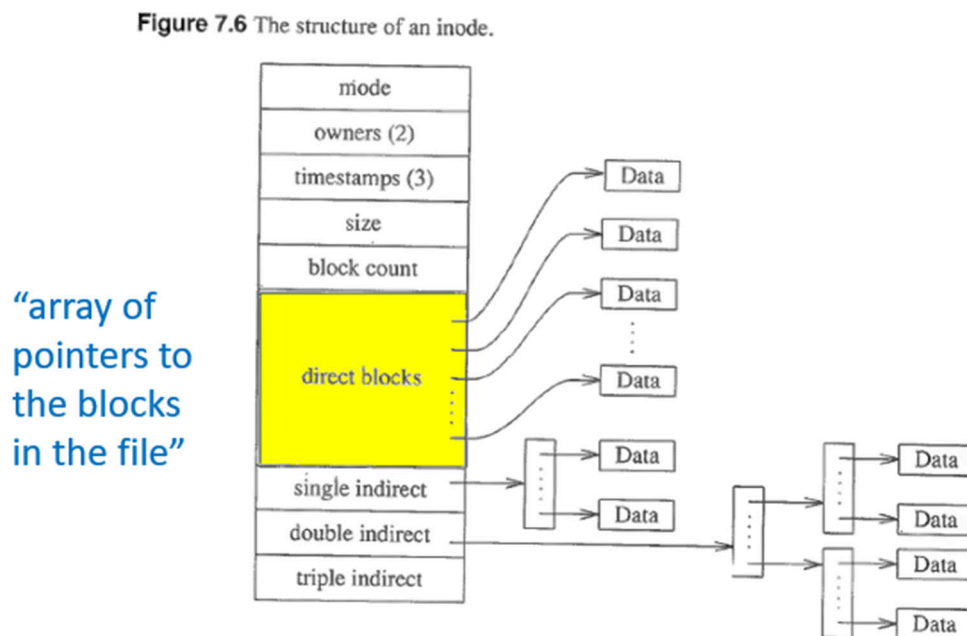
Element_[18.2.2]:

The Madany-Locasto-Leffler Combination renders obvious Elements_[1.1.2], [9.1.2], [17.1.2], and [18.2.2]. EX1003, ¶¶196-202.

In each of Grounds 1A-1B, the analysis points to *data block numbers* (e.g., logical block numbers or physical block numbers) *contained* in Madany's mapping table. *Supra* §V.A.6, Element_[1.1.2]; §V.B.5, Element_[1.1.2]. Moreover, it would have been understood by or obvious to a POSITA in view of Leffler's file system teachings that Madany's mapping table is an integral part of the file's inode,

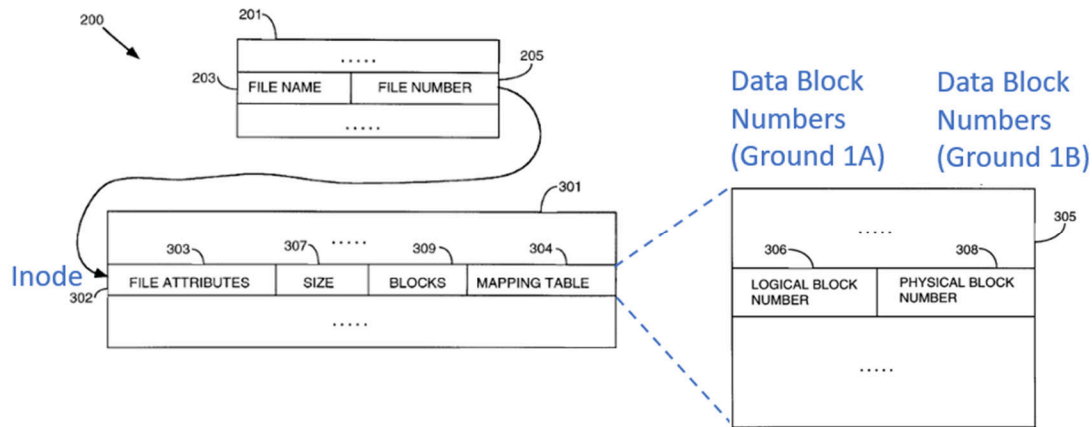
meaning the *data block numbers* are *contained* in an *index node* (the inode including the mapping table). *Supra* §V.C.1; EX1003, ¶¶198-199.

For example, as discussed, Leffler’s “Figure 7.6 shows that the inode contains an array of pointers to the blocks in the file,” allowing the system to “convert from a logical block number to a physical sector number by indexing into the array using the logical block number.” EX1006, 194; *supra* §V.C.1. Since Madany explains that its “mapping table 305 contains information on how the logical blocks within a file are mapped to physical blocks on a disk” (EX1004, 5:52-54), a POSITA would have understood that Leffler’s “array of pointers to blocks in the file” (EX1006, 194) is an example of Madany’s mapping table 305. EX1003, ¶200.



EX1006, 194 (Figure 7.6) (annotated).

A POSITA, therefore, would have understood and/or found obvious the notion of integrating Madany's mapping table in the inode of the Madany-Locasto-Leffler Combination, as illustrated below. EX1003, ¶201. Accordingly, in the Combination the *data block numbers* are *contained* in an Inode 302 (*index node*).



Madany-Locasto-Leffler Combination: Visual Aid

A POSITA would have been motivated to incorporate this rudimentary teaching of Leffler with the Madany-Locasto Combination for multiple reasons. EX1003, ¶202. To start, Madany explicitly references Leffler's description concerning the structure of an inode, writing that the "inode 302, or row of the file table 301, contains information about the given file. Inodes are described in 'The Design and Implementation of the 4.3BSD UNIX Operating System' [Leffler], Chapter 7." EX1004, 5:40-44. Since Leffler includes the mapping of logical to physical blocks in the inode itself, a POSITA would have been motivated to apply that technique in the context of the Madany-Locasto Combination. EX1006, 194,

Figure 7.6. Further, including the mapping in the inode itself increases system efficiency by eliminating the need to follow a pointer before accessing the mapping table for direct blocks. EX1006, 194; EX1003, ¶202. Additionally, a POSITA would have understood that there are effectively only two viable options: include the mapping table in the inode or include a reference to the mapping table in the inode. EX1003, ¶202. Both options would have been obvious. EX1003, ¶202.

(b) Claim 3

Elements_[3.1]-[3.3]:

The Madany-Locasto-Leffler Combination renders obvious Elements_[3.1]-[3.3]. EX1003, ¶¶203-208. As discussed at Section V.C.1, Leffler provides a full-length discussion of the 4.3BSD UNIX operating system, including a breakdown of the internal file system and a description of the calls and operations involved in reading and writing to the storage disk. *See generally* EX1006, 169-223. A POSITA contemplating the Madany-Locasto Combination of Grounds 1A-1B would have been motivated to incorporate Leffler's conventional file system teachings regarding disk reads and writes for multiple reasons. EX1003, ¶203. First, as discussed above at Independent Claims 1, 9, and 17-18, Madany explicitly references Leffler's Chapter 7 on the 4.3BSD file system, which would have prompted the POSITA to consult Leffler for supplemental implementation details on relevant subjects like the flow of calls and procedures for reading and writing to a storage disk. EX1003,

¶203; EX1004, 5:40-44. Second, even without Madany’s express reference to Leffler, the POSITA would have been motivated to pursue Leffler’s supplemental file system teachings as a well-known solution to the rudimentary design problem of “convert[ing] from the user abstraction of a file as an array of bytes to the structure imposed by the underlying physical medium [of the storage disk].” EX1006, 191; EX1003, ¶203.

The Board should reject any argument from Patent Owner that seeks to undermine the Madany-Locasto-Leffler Combination based on the fact that Leffler’s file system teachings on basic read/write calls do not involve Madany’s compression techniques. For one, Madany makes clear that not every file is compressed and that it is “desirable for both compressed and expanded files to coexist” in the same system. EX1004, 3:37-43; *see also* 4:32-36 (similar), 11:1-4 (“compression would not be useful and will not be used”). Even if, hypothetically, Madany’s sole focus was to create a system that compressed every file (which is not true), it would still have been obvious to rely on Madany’s description of, and explicit reference to, the well-known 4.3BSD UNIX file system taught by Leffler. *See Intel Corp. v. Qualcomm Inc.*, 21 F.4th 784, 799-800 (Fed. Cir. 2021) (rejecting a “complaint” that the petitioner “relied on the prior-art [device] described as background in [the primary reference]”). “[Madany] need not be limited to the invention claimed therein; it may be cited for all that it teaches.” *In re Theresa*, 720 F. App’x 634, 638

(Fed. Cir. 2018). Lastly, although it is not required for the reasons discussed, the Petition’s analysis of Claims 3 and 4 is applicable to implementations that employ Madany’s compression techniques. EX1003, ¶204; EX1004, 7:8-65 (“mapping table 305”). In short, Madany utilizes an inode and mapping table like the one in Leffler’s 4.3BSD Unix file system, even with compression, meaning the rudimentary steps of obtaining logical/physical block numbers are still performed during reading and writing operations. EX1003, ¶204.

Applying Leffler’s Teachings to the Ground 1A Madany-Locasto Combination

When reading or writing, the Madany-Locasto-Leffler Combination *obtains* the logical block number (*data block number*, per Ground 1A, Element_[1.1.1]) of the data to be *read*/written. See EX1006, 192 (“The system calculates the logical blocks...”); *supra* §V.C.1; EX1004, 2:47-51 (“a table containing N logical to physical block mappings”), 5:29-7:65 (Madany referencing Leffler and discussing “mapping table 305”); EX1003, ¶205.

Next, as discussed, the Combination *applies* Locasto’s *mapping function* to the logical block number to *obtain* a physical block number (*mapped data block number*) that is the address on the disk where the logical block data is stored. *Supra* §V.A.6 (Ground 1A, Elements_[1.1.1], [1.2.1], [1.2.2]); EX1003, ¶206. The Combination then completes the *read* or write request on the physical disk using the

physical block number (*reading the data in the mapped data block numbers*).
EX1006, 206-207 (Figure 7.12); §V.C.1; EX1003, ¶206.

Applying Leffler's Teachings to the Ground 1B Madany-Locasto Combination

When reading or writing, the Madany-Locasto-Leffler Combination obtains the logical block number of the data to be *read*/written and uses that logical block number, via the mapping table, to *obtain* the corresponding physical block number (*data block number*, per Ground 1B, Element_[1.1.1]). See EX1006, 192, 194 (“The system can convert from a logical block number to a physical [block] number...”), 195 (“logical-to-physical block mapping”); *supra* §V.C.1; EX1004, 2:47-51 (“a table containing N logical to physical block mappings”), 5:29-7:65 (Madany referencing Leffler and discussing “mapping table 305”); EX1003, ¶207.

Next, as discussed, the Combination *applies* Locasto’s *mapping function* to the physical block number to determine a spread physical block number (*mapped data block number*) that is the address on the disk where the logical block data is stored. *Supra* §V.B.5 (Ground 1B, Elements_[1.1.1], [1.2.1], [1.2.2]); EX1003, ¶208. The Combination then completes the *read* or write request on the physical disk using the spread physical block number (*reading the data in the mapped data block numbers*). EX1006, 206-207 (Figure 7.12); §V.C.1; EX1003, ¶208.

(c) **Claim 4**

Element_[4.1]:

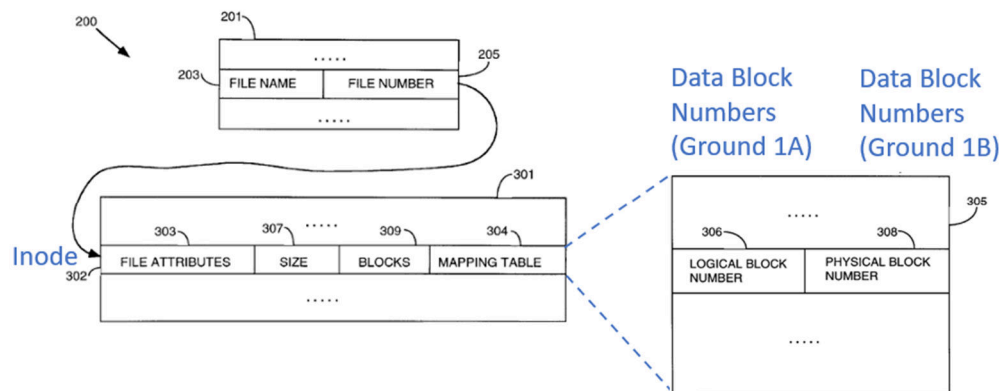
The Madany-Locasto-Leffler Combination renders obvious Element_[4.1]. EX1003, ¶¶209-211. Leffler explains that the operating system maintains a record of ***free blocks***. EX1006, 195, 197; EX1003, ¶209; *see also* §V.C.2, Claim 3 (motivation-to-combine Leffler’s conventional file system teachings with Madano-Locasto). For example, a “free list” contains “a list of all the ***free blocks*** in the filesystem.” EX1006, 195. Alternatively, the system can maintain “a bit map describing available [(***free***)] ***blocks*** in the cylinder group.” EX1006, 197.

As files grow, the “filesystem [] handles the ***allocation*** of new blocks to files.” EX1006, 193. Leffler continues, “[e]ach time a process does a ***write*** system call, the system checks to see whether the size of the file has increased,” and if “the file needs to be expanded, the request is rounded up to the next fragment size and only that much space is ***allocated***.” EX1006, 199. Since a single file system block can be divided into one or more fragments, a POSITA would have understood and/or found obvious that a block can be exactly one fragment. EX1006, 198; EX1003, ¶210. When a file is ***allocated*** a new block, a POSITA would have understood that the block would be assigned from among the ***free data blocks on the storage device***. EX1003, ¶210.

Leffler further teaches that an inode contains a file's "logical-to-physical block mapping." EX1006, 194-195. Therefore, a POSITA would have understood and/or found obvious that since the inode stores the block numbers of the physical blocks allocated during the *write operation, data block numbers of free data blocks* are also *allocated*. EX1003, ¶211. For example, since Leffler teaches that the "free list" contains "a list of all the free blocks in the file system," a POSITA would have understood and/or found obvious that when a block is removed from the free list, its corresponding data block number is removed from the free list ("list of all the free blocks in the filesystem") and added to the inode. EX1006, 194-195; EX1003, ¶211.

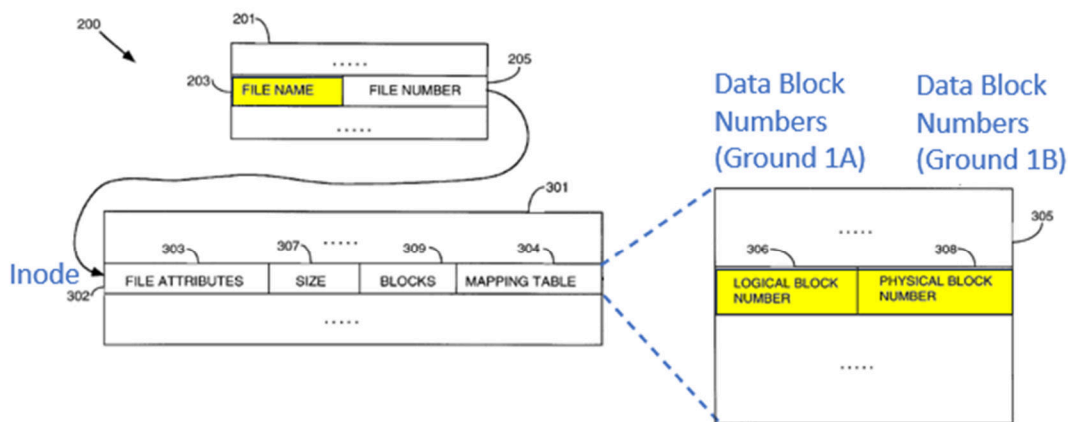
Element [4.2]:

The Madany-Locasto-Leffler Combination renders obvious Element_[4.2]. EX1003, ¶¶212-214. As explained above in reference to Element_[1.1.2], in the Madany-Locasto-Leffler Combination, the inode contains the *data block numbers of a file*, as illustrated in the visual aid below. *Supra* §V.C.1.



Madany-Locasto-Leffler Combination: Visual Aid

Further, per Madany, the “file system has a hierarchy of directory structures 201” and “[w]ithin the directory 201 are entries of file names within the file system and corresponding file numbers that allow files to be indexed to an entry in the file table.” EX1004, 5:30-35. “Each entry 302 in the file table 301 identifies a file of the file system and is commonly known as an Inode.” EX1004, 5:35-37. Since the directory 201 contains a file name that is linked to the inode, and the inode contains the data block numbers, a POSITA would have understood and/or found obvious that *the data block numbers are linked to a file name of the file*. EX1003, ¶213. Further, since the data block numbers are added to the inode during the write operation, the write operation comprises linking the data block numbers to a file name of the file. EX1003, ¶¶213-214.



Madany-Locasto-Leffler Combination: Visual Aid

Element_[4.3]:

The Madany-Locasto-Leffler Combination renders obvious Element_[4.3].
Supra §V.C.2, Elements_[3.1]-[3.3] (describing the ***application of the mapping function*** in both read and write operations); EX1003, ¶215.

Element_[4.4]:

The Madany-Locasto-Leffler Combination renders obvious Element_[4.4].
EX1003, ¶215. For example, per Leffler, the Combination uses the “*bio*” and “*dev*” calls to complete the read or ***write*** request on the physical disk using the physical (***mapped***) ***block number***. EX1006, 207; §V.C.1, §V.C.2, Elements_[3.1]-[3.3]; EX1003, ¶216.

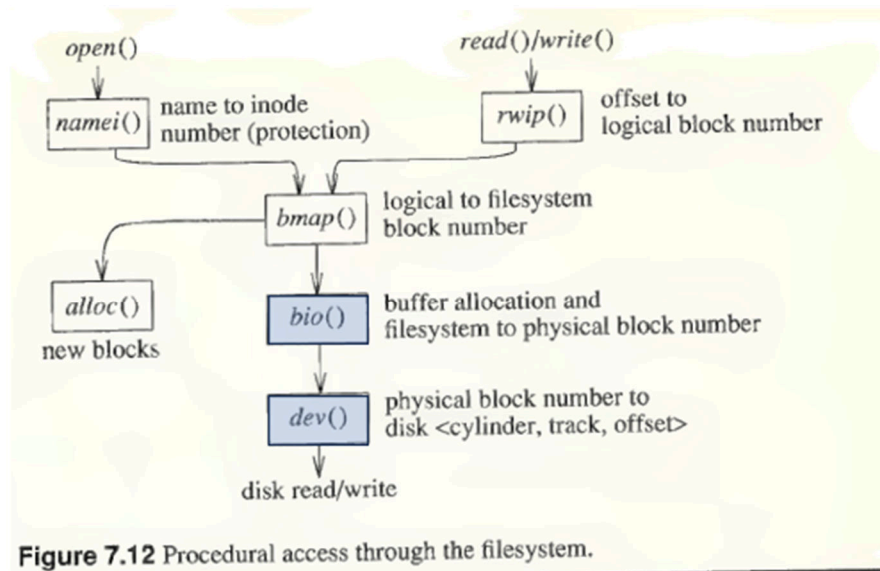


Figure 7.12 Procedural access through the filesystem.

EX1006, 207 (Figure 7.12) (annotated).

(d) **Claim 11**

Element_[11.1]:

Supra §V.C.2, Element_[3.1]; EX1003, ¶217.

Element_[11.2]:

Supra §V.C.2, Element_[3.2]; EX1003, ¶218.

Element_[11.3]:

Supra §V.C.2, Element_[3.3]; EX1003, ¶219.

(e) **Claim 12**

Element_[12.1]:

Supra §V.C.2, Element_[4.1] (*receive allocated data block numbers of free data blocks in the storage device from a file system*); *see also* §V.A.6, Element_[9.1.1] (*block distribution engine*); EX1003, ¶220.

Element_[12.2]:

Supra §V.C.2, Element_[4.2]; EX1003, ¶221.

Element_[12.3]:

Supra §V.C.2, Element_[4.3]; EX1003, ¶222.

Element_[12.4]:

Supra §V.C.2, Element_[4.4]; EX1003, ¶223.

D. GROUND 3: Claims 1-8 Are Obvious Over Leffler-Locasto

1. Leffler (EX1006)

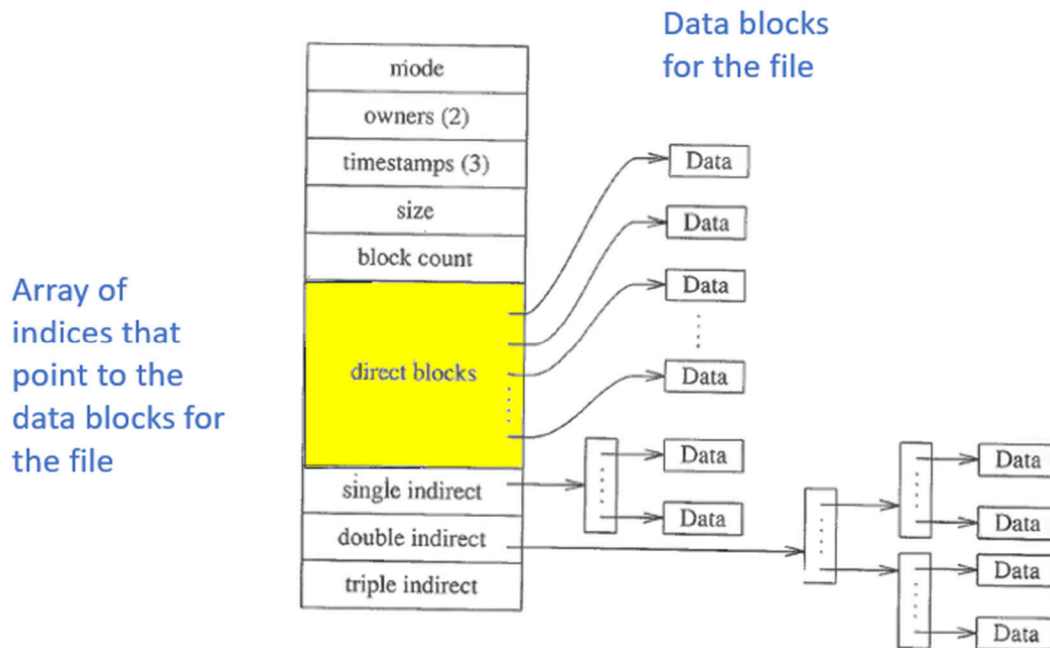
Supra §V.C.1; EX1003, ¶224.

2. Locasto (EX1005)

Supra §V.A.2; EX1003, ¶225.

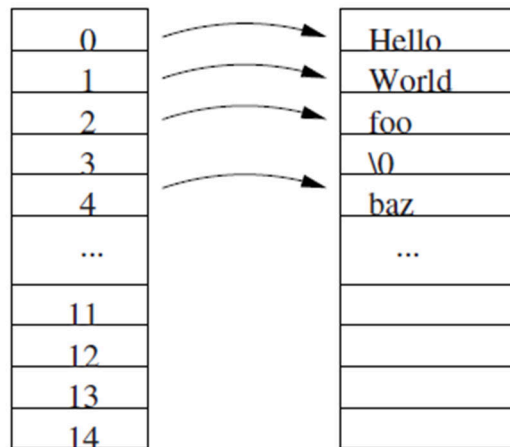
3. The Combination

As discussed, Leffler is a textbook that provides a “full-length description of the design and implementation of the research versions of the UNIX system,” including a discussion of the I/O system. EX1006, ix-x. Leffler further explains that, to enable mapping of logical blocks to physical blocks, “[e]very file has an associated inode that contains...an array of indices that point to the data blocks for the file.” EX1006, 189; EX1003, ¶226.



Leffler-Locasto-Combination: Visual Aid

Similar to Leffler, Locasto describes a file system in which “indices” are mapped to block numbers, as illustrated in Figure 2 (below). EX1005, 3-4; §V.A.2, *supra*; EX1003, ¶227.



EX1005, Figure 2.

Locasto introduces two security features: [i] a cryptographic file system that encrypts the stored data and a mapping function that uses a permutation function and [ii] a spreading seed to randomly distribute real file data throughout a large address space. EX1005, 3-6, Figures 3-4; §V.A.2, *supra*.

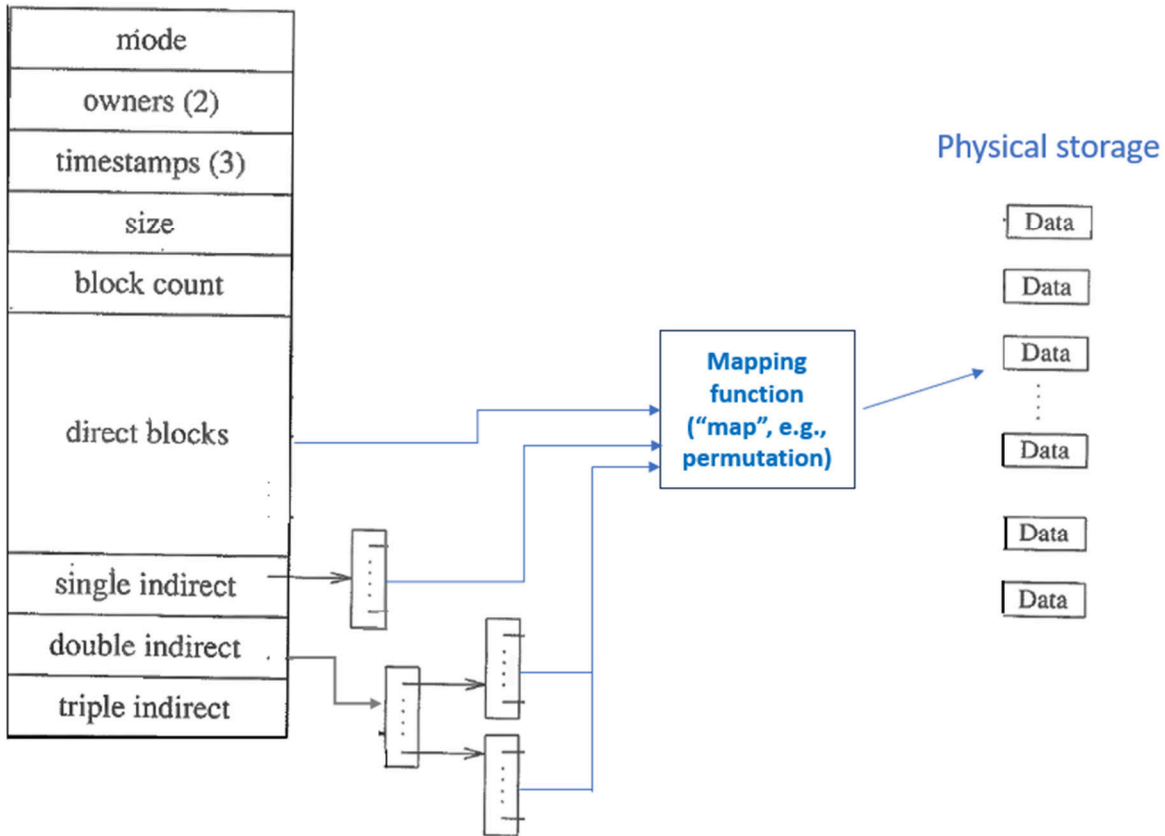
In the Leffler-Locasto Combination, the file system Leffler teaches is enhanced with security features like those taught in Locasto, (e.g., adding a mapping function and encrypting the stored data, as illustrated below). EX1003, ¶¶228-229. As described by Locasto, such “[c]ryptographic file systems provide data confidentiality by employing encryption to protect files against unauthorized access.” EX1005, 1; EX1003, ¶229. Thus, as explained further below in Section

V.D.4, a POSITA would have appreciated the predictable benefits of incorporating Locasto's security features with Leffler's efficient file system and associated data structures. EX1003, ¶229. Moreover, the POSITA would have understood the compatibility of Locasto's teachings with Leffler, such that Locasto's technical improvements could be implemented while preserving the fundamental aspects of Leffler's disclosure. EX1003, ¶229. For example, Leffler describes the BSD (a Unix variant) file system and Locasto describes the Linux (another Unix variant) "Ext2fs" file system. EX1004, 1:45-50, 3:14-18, 5:41-42; EX1005, 3, 8; EX1003, ¶229 (citing EX1008, EX1009).

In operation, when a program accesses a file at a given position, the Leffler-Locasto Combination determines the inode (e.g., as illustrated in Leffler's Figures 7.3, 4.4, 7.6) and uses the block mappings in the inode to determine the (unspread) physical blocks. EX1006, 194, Figure 7.6; EX1003, ¶230. Then, as shown in the visual aid below, the Combination employs a process like Locasto's (*see* §V.A.2) to facilitate the conversion of (unspread) physical block numbers in Leffler's inode to determine the spread physical block numbers. EX1005, 5; EX1003, ¶230. More specifically, the unspread physical numbers are converted to spread physical block numbers using Locasto's "map" function, e.g., a pseudo-random permutation function and a "spreading seed." EX1005, 4-5; EX1003, ¶230. The mapping function "randomly distribute[s] the real file data...throughout a large address space

so that an attacker would have to first guess the spreading schedule before he attempts a brute force search on the rest of the key.” EX1005, 4.

Leffler-Locasto Combination



Leffler-Locasto-Combination: Visual Aid

As shown in the visual aid above, the Leffler-Locasto Combination applies Locasto's mapping function (*map*) to the block numbers determined using Leffler's inode (e.g., by determining the block using the list of "direct blocks") and the bits of a *spreading seed* to determine the spread physical block number (*spread_pbn*):

$$spread_pbn_index = map(pbn, spreading_seed)$$

spread_pbn = get_from_index(spread_pbn_index)

EX1005, 5-6; EX1003, ¶231.

A POSITA would have understood that these two functions can be expressed equivalently as the single mapping function:

spread_pbn = get_from_index(map(pbn, spreading_seed))

EX1003, ¶232.

4. Motivation to Combine

The following reasons, taken separately or collectively, support the above-described Leffler-Locasto Combination. EX1003, ¶¶233-238.

First, a POSITA would have been motivated to pursue the Leffler-Locasto Combination for the purpose of maintaining confidentiality of the data in Leffler's file system. EX1005, 3-6; EX1003, ¶234. As Locasto explains, "[c]ryptographic file systems provide data confidentiality by employing encryption to protect files against unauthorized access." EX1005, 1. The POSITA would have appreciated the importance of protecting sensitive data against unauthorized access and, thus, would have been prompted to bolster Leffler's file system disclosure with data encryption teachings from a supplemental reference like Locasto. EX1003, ¶234 (citing EX1010; EX1011); *see also* ¶¶44-49.

Second, a POSITA would have been motivated to pursue the Combination to achieve in Leffler the specific benefits espoused in Locasto. EX1003, ¶235. For

example, Locasto's approach to encryption "reduc[es] the performance penalty associated with using larger key sizes without decreasing the level of security." EX1005, 2. As Locasto explains, by "using only part of the key for cryptographic operations" and "[t]he rest of the key...to unpredictably spread the data into the file's address space," "a user may decrease the performance penalty they pay for employing a cryptographic file system." *Id.*; *see also* 1 ("improved performance"), 12 ("negligible impact on performance") ("increasing an attacker's work factor by 2²⁸"). From the POSITA's perspective, Locasto's balance of security and performance would have been an advantageous addition to Leffler's file system. EX1003, ¶235.

Third, "design incentives" and "market forces" would have prompted a POSITA to pursue a combination of Leffler and Locasto. EX1003, ¶236. For example, a POSITA would have known that data security solutions were incorporated into popular file systems, such as the WINDOWS NT file system. EX1003, ¶236 (citing and quoting EX1010, 5 (a WINDOWS NT reference book explaining that "Data security is of primary importance...")). It was likewise well known that skilled artisans and market participants were actively pursuing secure variations of the UNIX operating system that Leffler describes, some of which included built-in encryption programs. EX1003, ¶236 (citing EX1015, 6; EX1011, 14). Accordingly, the POSITA would have sought data security improvements to

Leffler's file system in an effort to compete and distinguish over competitor products in the marketplace. EX1003, ¶236; *see also* ¶¶44-49.

Fourth, by the time of the '686 patent in 2007, the scientific and patent literature was replete with proposals to incorporate data encryption techniques into existing file systems. EX1003, ¶237 (citing EX1012; EX1013). A POSITA would have been motivated by this contemporaneous design and research trend to incorporate Locasto's encryption-focused teachings into a file system like Leffler's. EX1003, ¶237; *see also* ¶¶44-49.

Fifth, a POSITA would have been led to the Leffler-Locasto Combination because it merely involves the application of a known technique (Locasto's data encryption) to a known system (Leffler's file system) to achieve predictable results (a file system with improved data security). EX1003, ¶238. Each element in combination merely performs the same function as it does separately. *Id.* "It's enough for [the Petition] to show that there was a known problem of [data security], that [Locasto's encryption technique] helped address that issue, and that combining the teachings of [Leffler and Locasto] wasn't beyond the skill of an ordinary artisan. Nothing more is required to show a motivation to combine under *KSR*." *Intel*, 61 F.4th at 1380-81.

5. Reasonable Expectation of Success

A POSITA would have had a reasonable expectation of success in creating a combination like the Leffler-Locasto Combination, which would have required only routine programming knowledge well within the skill of a POSITA. EX1003, ¶¶239-240. For example, the Leffler-Locasto Combination includes components of a file system and known encryption techniques to enhance the security of such file systems. *Supra* §§V.D.1-3; EX1003, ¶239. Integrating these aspects of Leffler and Locasto would have been a routine programming task for a POSITA. EX1003, ¶240.

6. Claim Element Analysis

(a) Claim 1

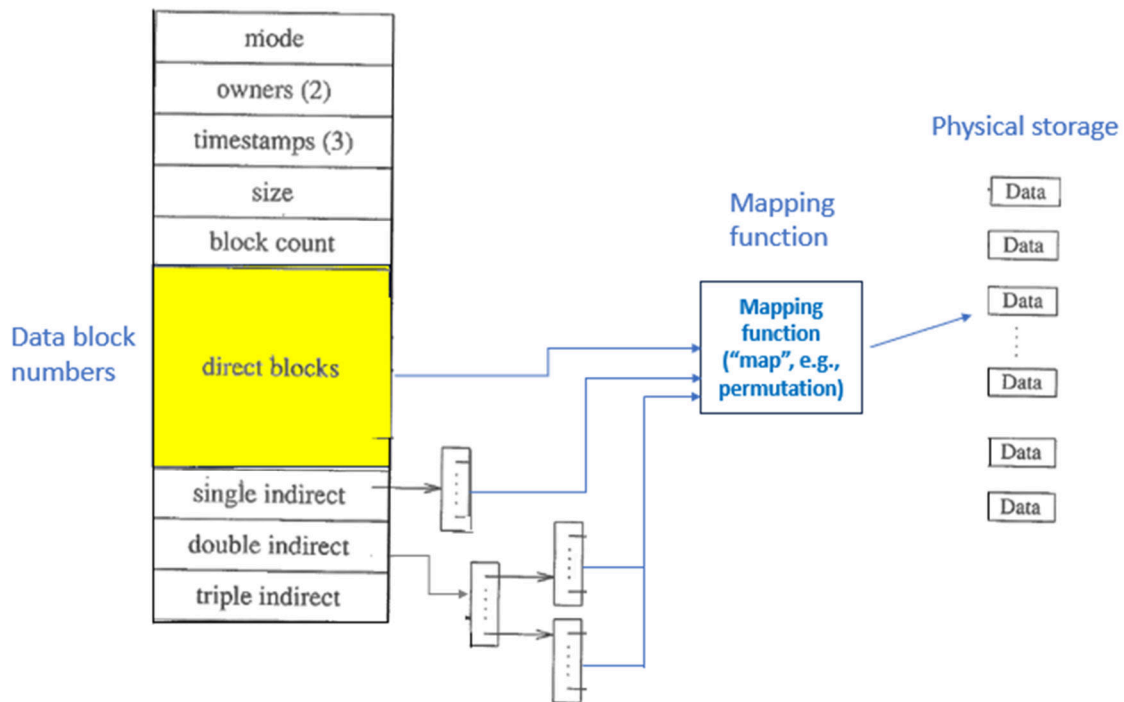
Element_[1.pre]:

The Leffler-Locasto Combination renders obvious Element_[1.pre]. EX1003, ¶241. For example, the Combination includes Leffler’s description of a “block-device interface” including “the *filesystem*” that converts “from the user abstraction of a disk as an array the structure imposed by the physical medium.” EX1006, 171; *see also* 187, 191-192. The Combination additionally includes Locasto’s explanation that “[c]ryptographic file systems provide *data* confidentiality by employing encryption to protect files against unauthorized access.” EX1005, 1; §§V.D.1-3, *supra*. Locasto then describes the Camouflage cryptographic file system

that “offers improved performance relative to a cipher with a larger key-schedule, while providing the same *security* properties.” EX1005, 1.

Element_[1.1.1]:

The Leffler-Locasto Combination renders obvious Element_[1.1.1]. EX1003, ¶¶242-245. For example, Leffler explains that “most UNIX systems use the concept of an index node, or inode” that “contains an array of pointers to the blocks in the file.” EX1006, 194. “The system can convert from a logical block number to a physical [block] number by indexing into the array using the logical block number.” EX1006, 194-195 (“In addition to the logical-to-physical block mapping, the inode contains other information about the contents of the file.”). Thus, the Combination includes *data block numbers* (e.g., Leffler’s physical block numbers) as illustrated below. EX1003, ¶242. A POSITA would have understood or found obvious that physical block numbers are *data block numbers* at least since physical block numbers correspond to data managed by a file system and stored on a physical disk. EX1003, ¶242. In other words, as would have been clear to a POSITA, Leffler’s physical *block numbers* are used by the file system to store and access *data*. EX1003, ¶242.



Leffler-Locasto-Combination: Visual Aid

Patent Owner could not reasonably contest this mapping because Leffler’s disclosure of an “inode” in the context of a Unix operating system matches the description of the ’686 patent’s preferred embodiment. *See* EX1001, 2:1-28 (“The inode data structure is used in, for example, a Unix file system[.]”), 3:50-4:24 (describing an example where “an inode 210 of file 205 points to four (4) data block numbers”), Figures 1-2; EX1003, ¶243.

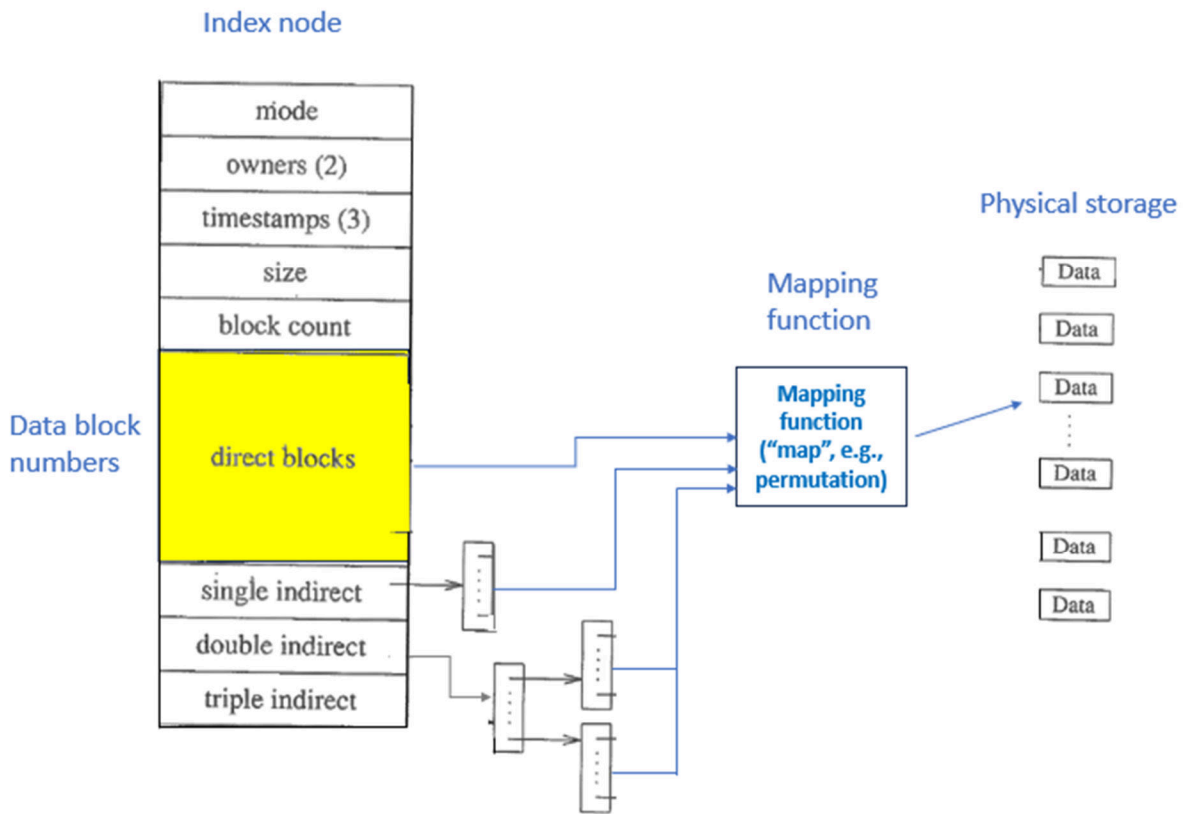
The Combination further teaches or renders obvious that the physical block numbers (***block numbers***) are ***associated with a file***. EX1003, ¶244. For example, Leffler explains that an “inode contains an array of pointers to the blocks in the file.” EX1006, 194. Thus, it would have been understood by or obvious to a POSITA that

the physical **block numbers** contained in Leffler's inode *are associated with a file* since the inode contains an array of pointers to the blocks in the file. EX1003, ¶244 (citing EX1006, 187).

As discussed at Section V.D.3, the Leffler-Locasto Combination *applies a mapping function* to physical block numbers (**data block numbers**). EX1003, ¶245. Specifically, the Combination's "pseudo-random permutation (PRP) function takes the" physical block number "and the bits of the spreading seed to return a" spread physical block number, which is the physical address of the data block on the physical storage disk. EX1005, 4-5; EX1003, ¶245.

Element_[1.1.2]:

The Leffler-Locasto Combination renders obvious Element_[1.1.2]. EX1003, ¶¶246-247. As discussed above at Element_[1.2.1], the Combination includes physical (**data**) **block numbers** that are *contained* in an inode (**index node**). *E.g.*, EX1006, 194 ("index node, or inode"); EX1003, ¶246. Here again, Patent Owner could not reasonably contest this mapping because Leffler's disclosure of an "inode" in the context of a Unix operating system matches the description of the '686 patent's preferred embodiment. *See* EX1001, 2:1-28, 3:50-4:24, Figures 1-2; EX1003, ¶246.



Leffler-Locasto-Combination: Visual Aid

Further, as would have been understood by or obvious to a POSITA from Leffler's teachings, the inode is *associated with a file*. EX1003, ¶247. For example, Leffler explains that "the inode contains an array of pointers to the blocks in the *file*." EX1006, 194; *see also* 189 ("Every *file* has an *associated inode*..." and "Each *file* has a single *inode*...").

Element [1.2.1]:

The Leffler-Locasto Combination renders obvious Element_[1.2.1]. EX1003, ¶¶248-250. As discussed, the Combination applies "a pseudo-random permutation

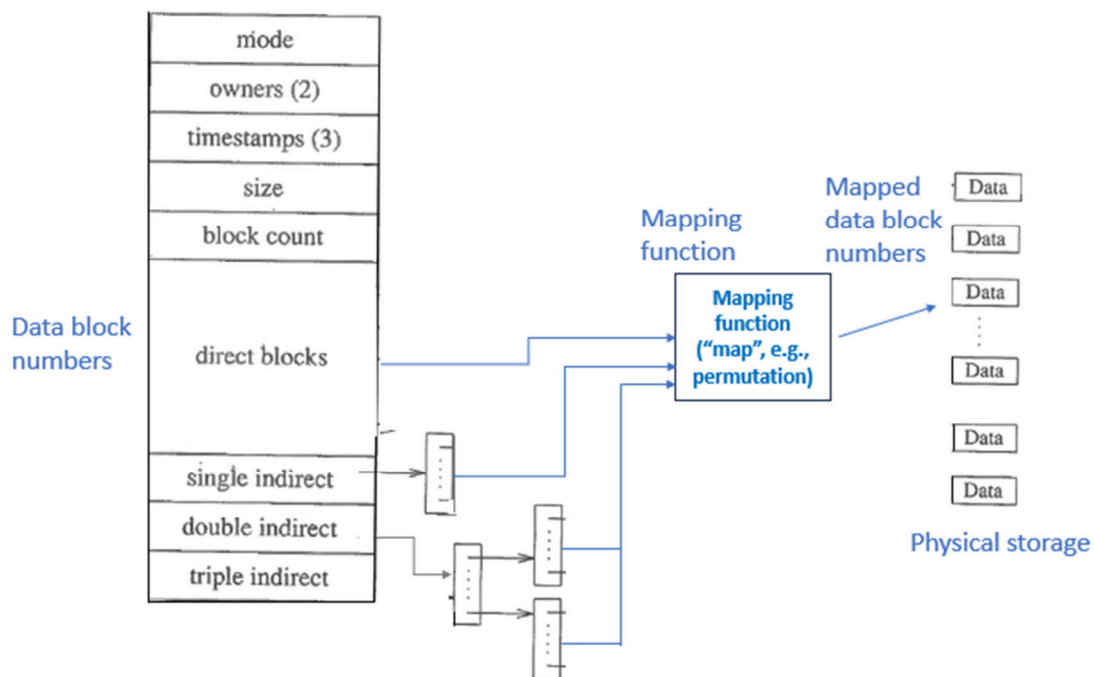
(PRP) function [that] takes the [physical block number in Leffler's inode] and the bits of the spreading seed to return a new index value.” EX1005, 4-5.

$$\text{spread_pbn_index} = \text{map}(\text{pbn}, \text{spreading_seed})$$

The spread physical “block number is then derived from this new index.” EX1005, 5; EX1003, ¶249.

$$\text{spread_pbn} = \text{get_from_index}(\text{spread_pbn_index})$$

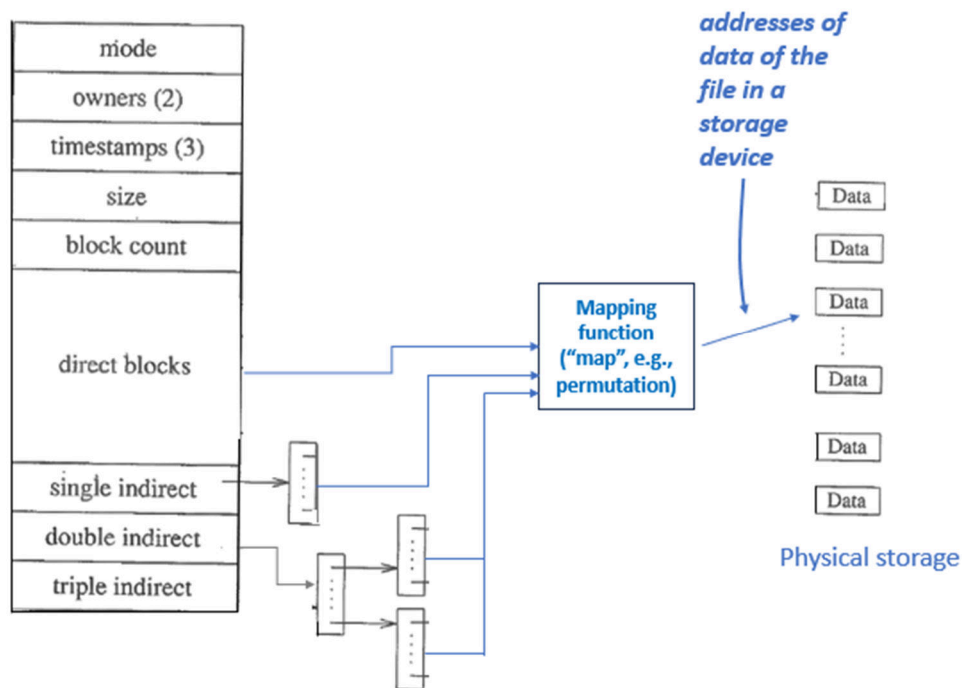
Id. Thus, in the Leffler-Locasto Combination, the spread physical block numbers (***mapped data block numbers***) are obtained after applying the “map” (***mapping***) ***function***. EX1003, ¶250.



Leffler-Locasto-Combination: Visual Aid

Element_[1.2.2]:

The Leffler-Locasto Combination renders obvious Element_[1.2.2]. EX1003, ¶251. Specifically, it would have been understood by or obvious to a POSITA that the spread physical ***data block numbers are addresses*** of physical blocks on the disk (***storage device***) storing ***data of the file***. *Supra* §V.D.3; EX1003, ¶251. For example, Locasto explains that a block number determined by its mapping function is “the block on the disk” where the data at a particular file position resides. EX1005, 5; EX1003, ¶251.

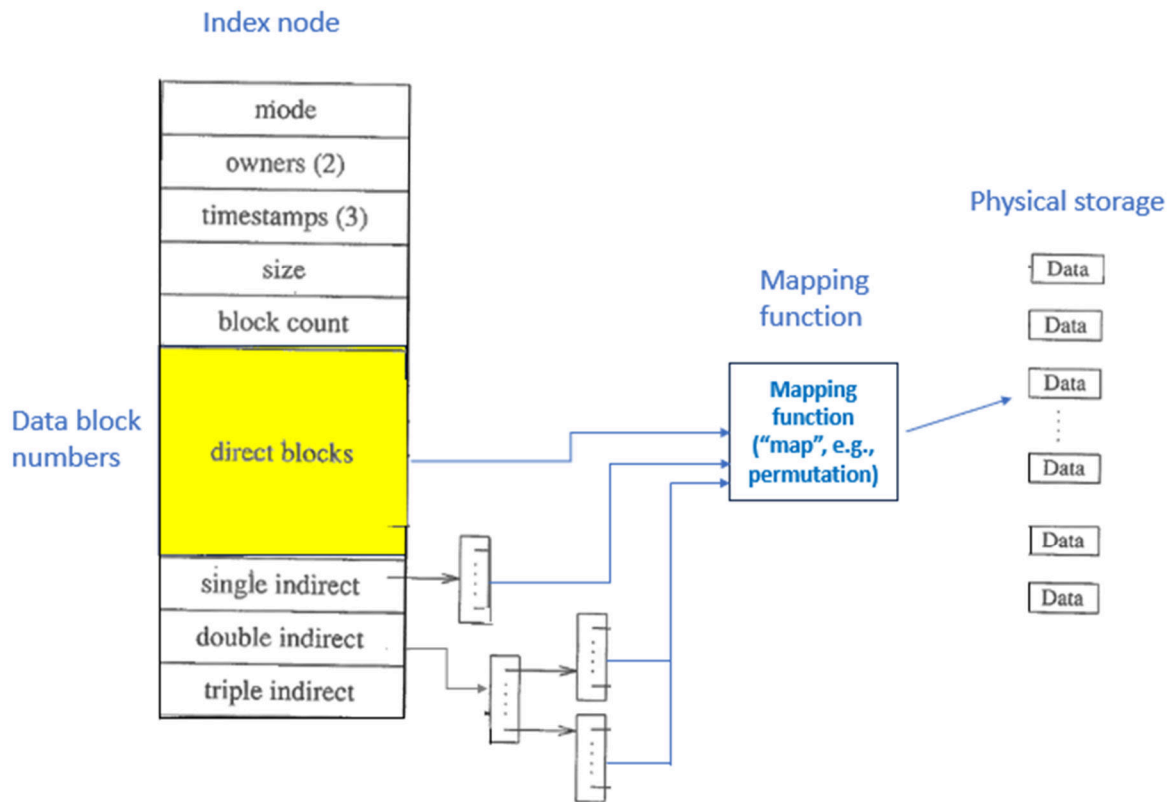


Leffler-Locasto-Combination: Visual Aid

(b) **Claim 2**

Claim_[2]:

The Leffler-Locasto Combination renders obvious Claim_[2]. EX1003, ¶¶252-256. For example, as discussed above at Element_[1.1.1], the Combination applies a mapping function to the physical block numbers (***data block numbers***) to obtain the physical addresses of the data blocks on the physical storage disk, as illustrated in the visual aid below. *Supra* §V.D.3; EX1003, ¶252.



Leffler-Locasto-Combination: Visual Aid

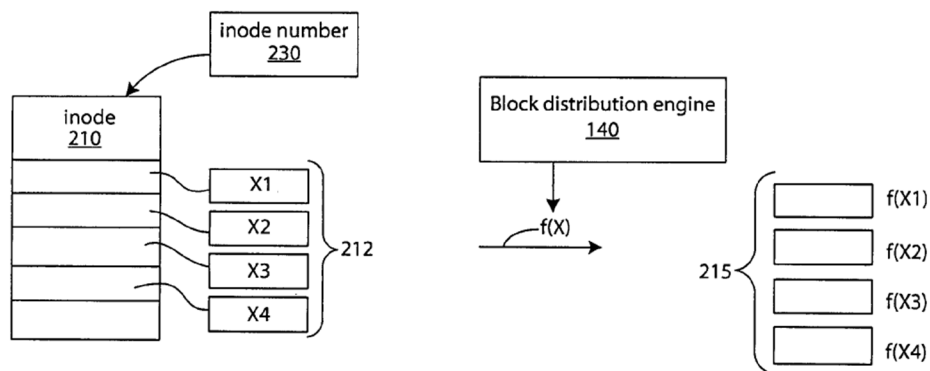
As discussed at Section V.D.3, in the Leffler-Locasto Combination, spread (mapped) data block numbers are determined from data block numbers using the equation:

$$\text{spread_pbn} = \text{get_from_index}(\text{map}(\text{pbn}, \text{spreading_seed}))$$

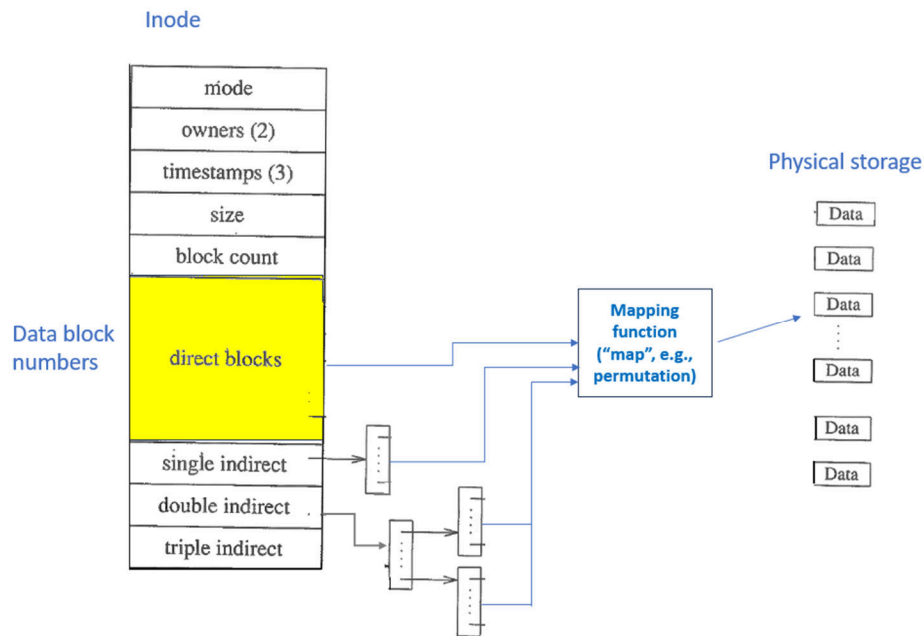
EX1003, ¶253.

Since the Leffler-Locasto Combination computes the spread physical block numbers (*mapped data block numbers*) from the physical block numbers (*block numbers*), a POSITA would have understood that the Leffler-Locasto Combination *stores the mapping* of physical block numbers (*block numbers*) to spread physical block numbers (*mapped block numbers*) at least at the conclusion of the calculation of the spread physical block numbers. EX1003, ¶¶254-255.

Notably, the Leffler-Locasto Combination has an analogous structure to '686 patent's structure, as illustrated below. EX1003, ¶256.



EX1001 ('686 patent), Figure 2 (excerpt).



Leffler-Locasto Combination: Visual Aid

(c) **Claim 3**

Elements_[3.1]-[3.3]:

The Leffler-Locasto Combination renders obvious Elements_[3.1]-[3.3]. EX1003, ¶¶257-258. As discussed at Section V.C.1, Leffler provides a full-length discussion of the 4.3BSD UNIX operating system. *See generally* EX1006, 169-223. Consistent with how 4.3BSD works, when reading or writing, the Leffler-Locasto Combination obtains the logical block number of the data to be *read*/written and uses that logical block number, via the mapping table, to *obtain* the corresponding physical block number (*data block number*, per Element_[1.1.1]). *See* EX1006, 192, 194, 195; EX1003, ¶257; *supra* §V.C.2 (Ground 2, addressing Elements_[3.1]-[3.3] in view of Leffler's teachings).

Next, as discussed at Elements_[1.1.1], [1.2.1], [1.2.2], the Combination *applies* Locasto's *mapping function* to the physical block number to determine a spread physical block number (*mapped data block number*) that is the address on the disk where the logical block's data is stored. EX1003, ¶258. The Combination then completes the *read* or write request on the physical disk using the spread physical block number (*reading the data in the mapped data block numbers*). EX1006, 206-207 (Figure 7.12); §V.C.1; EX1003, ¶258.

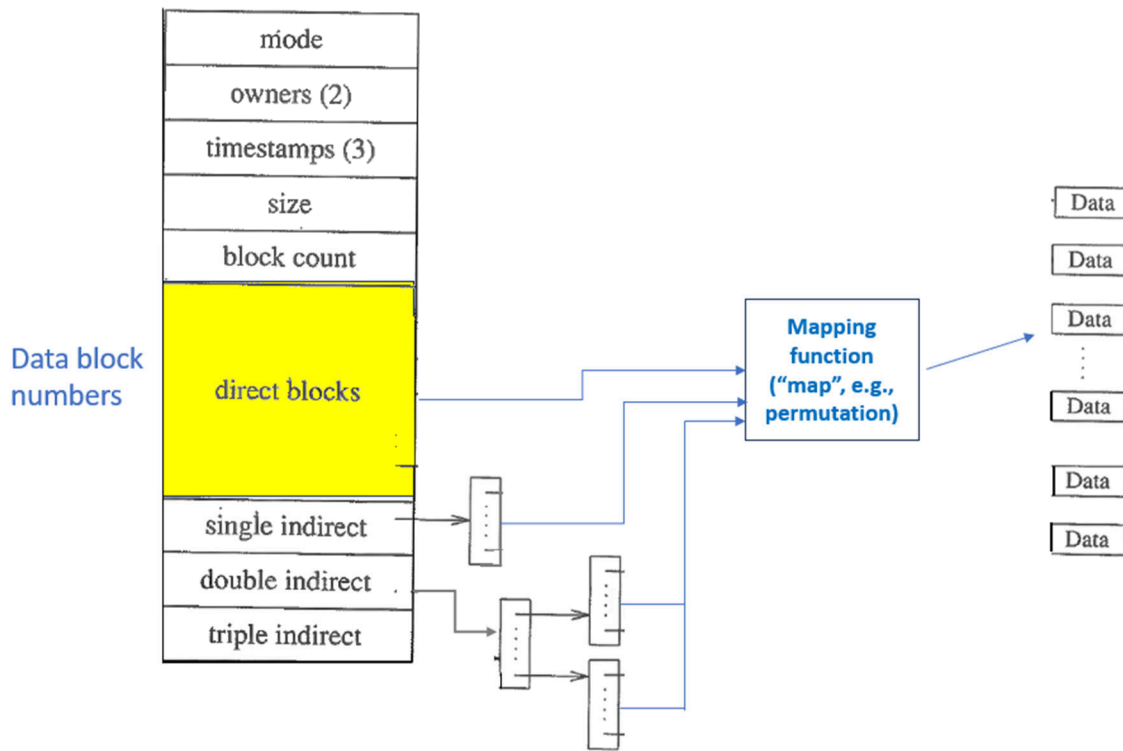
(d) **Claim 4**

Element_[4.1]:

The Leffler-Locasto Combination renders obvious Element_[4.1]. *Supra* §V.C.2, Element_[4.1] (detailing how Leffler teaches *a write operation for the file* that includes *allocating data block numbers*); EX1003, ¶¶259-261.

Element_[4.2]:

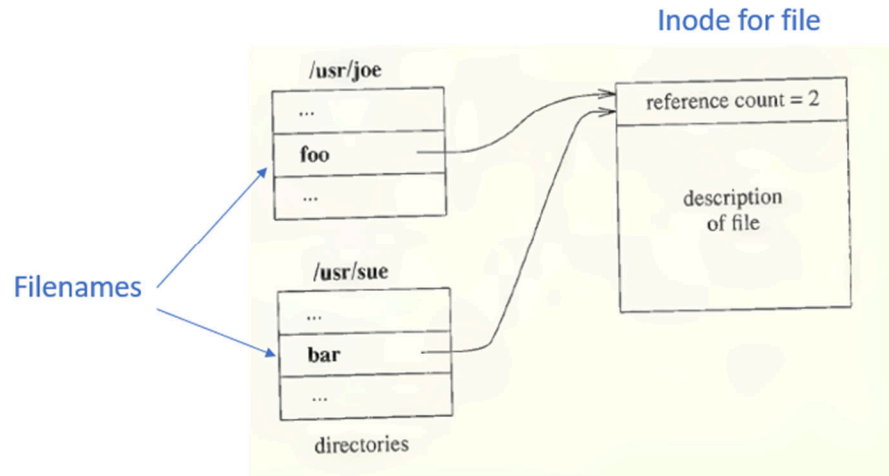
The Leffler-Locasto Combination renders obvious Element_[4.2]. EX1003, ¶¶262-266. For example, as explained above at Element_[1.1.2], in the Leffler-Locasto Combination, the inode contains the *data block numbers of a file*, as illustrated in the visual aid below. *Supra* §V.D.3.



Leffler-Locasto-Combination: Visual Aid

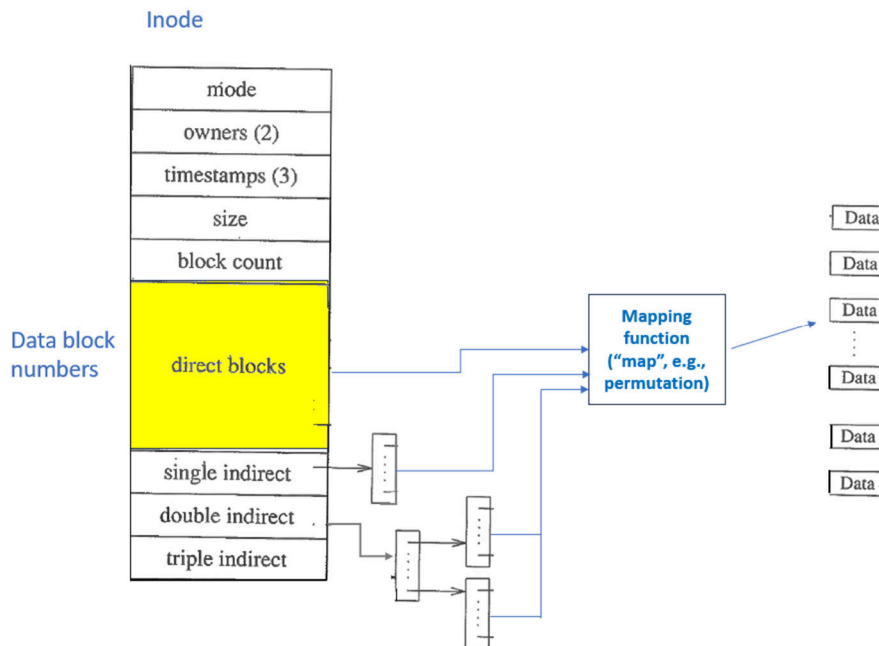
Further, per Leffler, “[f]ilenames are maintained in directories rather than in inodes because a file may have many names, or links, and the name of a file may be large (up to 255 bytes in length).” EX1006, 195. Directories further contain directory entries, and each entry contains a pointer to the file’s inode. EX1006, 187.

Leffler provides an example in which two different directory entries reference the same file, so the inode has a reference count of two. EX1006, 189, Figure 7.3; EX1003, ¶264.



EX1006, 189 (Figure 7.3) (annotated).

The inode further contains the data block numbers, as illustrated in the visual aid below. EX1006, 194; EX1003, ¶265.



Leffler-Locasto-Combination: Visual Aid

Thus, a POSITA would have understood or found obvious that when data are written to a new block for a file, the allocated physical block numbers (*data block numbers*) are added to the inode, which are *linked* to at least one directory entry containing the *filename*. EX1006, 187, 189, Figures 7.3, 7.6; EX1003, ¶266.

Element_[4.3]:

The Leffler-Locasto Combination renders obvious Element_[4.3]. See §V.D.6, Elements_[3.1]-[3.3] (describing the *application of the mapping function* in both read and write operations); EX1003, ¶267.

Element_[4.4]:

The Leffler-Locasto Combination renders obvious Element_[4.4]. See §V.C.2, Element_[4.4] (detailing how Leffler teaches *writing the data of the file into the mapped data block numbers*); EX1003, ¶268.

(e) Claim 5

Claim_[5]:

The Leffler-Locasto Combination renders obvious Claim_[5]. EX1003, ¶¶269-271. As described in Sections V.D.3 and V.D.6 (Element_[1.1.1]) the Leffler-Locasto Combination *applies a mapping function* to physical block numbers (*data block numbers*) *of a file*. EX1005, 4-5. The rudimentary concept of performing this technique at *each* logical block number would have been understood by or obvious to a POSITA. EX1003, ¶269. First, when each data block of a file is

written, the mapping is performed on the data block number of that data block. EX1005, 4 (“The purpose of the spreading schedule is to randomly distribute the real file data throughout a large address space”); EX1003, ¶269. Therefore, since all of the data blocks are written, the mapping function is applied for each data block number of the file. EX1003, ¶269. Second, the mapping function is applied to each data block number for the purpose of retrieving the complete set of data blocks for the file and, at least in some instances, entire files will be read. EX1003, ¶269.

As discussed at §V.D.3, the *mapping function* corresponds to the following composite *equation*:

$$spread_pbn = get_from_index(map(pbn, spreading_seed))$$

EX1005, 5; EX1003, ¶270.

Moreover, Locasto teaches that the mapping function includes a pseudo-random permutation function based on the well-known cipher algorithms “RC6” and “RC4,” both of which include multiple mathematical operations. EX1005, 4-5, 10; EX1003, ¶271 (citing EX1014).

(f) **Claim 6**

Claim_[6]:

The Leffler-Locasto Combination renders obvious Claim_[6]. EX1003, ¶272; *supra* §V.A.6, Claim_[6] (discussing Locasto’s *mapping function*).

(g) Claim 7

Claim_[7]:

The Leffler-Locasto Combination renders obvious Claim_[7]. EX1003, ¶273; *supra* §V.A.6, Claim_[7] (explaining, *inter alia*, that Locasto’s *mapping function* is optional).

(h) Claim 8

Claim_[8]:

The Leffler-Locasto Combination renders obvious Claim_[8]. EX1003, ¶274; §V.A.6, Claim_[8] (discussing Locasto’s *mapping function*).

VI. PAYMENT OF FEES—37 C.F.R. §42.103

Petitioners authorize the Patent and Trademark Office to charge any fees to Deposit Account No. 06-1050.

VII. MANDATORY NOTICES—37 C.F.R. § 42.8(a)(1)

A. Real Party-In-Interest—37 C.F.R. § 42.8(b)(1)

Petitioners, FedEx Corporation and FedEx Office and Print Services, Inc., are the real parties-in-interest. Additionally, even though Petitioners do not believe that these additional entities are real parties-in-interest, Petitioners further identify Federal Express Corporation and FedEx Dataworks in the interest of completeness.

B. Related Matters—37 C.F.R. § 42.8(b)(2)

Petitioners are not aware of any disclaimers, reexamination certificates or petitions for inter partes review for the '686 patent. The '686 patent is the subject of the following civil actions: *Valtrus Innovations Ltd. et al v. FedEx Corporation et al*, 2-25-cv-00516 (EDTX), filed May 9, 2025; *Valtrus Innovations Ltd. et al v. NetApp, Inc.*, 2-25-cv-00517 (EDTX), filed May 9, 2025; *NetApp, Inc. v. Valtrus Innovations Ltd. et al*, 1-25-cv-06138 (SDNY), filed July 25, 2025; *Valtrus Innovations Ltd. et al v. The Home Depot, Inc. et al*, 2-25-cv-00081 (EDTX), filed January 27, 2025.

C. Lead And Back-Up Counsel—37 C.F.R. § 42.8(b)(3)

Petitioners provides the following designation of counsel.

Lead Counsel	Backup Counsel
Kenneth W. Darby, Jr., Reg. No. 65,068 Fish & Richardson P.C. 60 South Sixth Street, Suite 3200 Minneapolis, MN 55402 Tel: 202-783-5070 Fax: 877-769-7945 Email: IPR39898-0050IP1@fr.com	David M. Hoffman, Reg. No. 54,174 Fish & Richardson P.C. 60 South Sixth Street, Suite 3200 Minneapolis, MN 55402 Tel: 202-783-5070 Fax: 877-769-7945 IPR39898-0050IP1@fr.com

D. Service Information

Please address all correspondence and service to the address listed above.

Petitioners consent to electronic service by email at IPR39898-0050IP1@fr.com.

Respectfully submitted,

Dated August 29, 2025

/Kenneth Wayne Darby Jr./
Kenneth W. Darby, Jr., Reg. No. 65,068
David M. Hoffman, Reg. No. 54,174
Fish & Richardson P.C.
60 South Sixth Street, Suite 3200
Minneapolis, MN 55402
T: 202-783-5070
F: 877-769-7945

(Control No. IPR2025-01491)

Attorneys for Petitioners

CERTIFICATION UNDER 37 CFR § 42.24

Under the provisions of 37 CFR § 42.24(d), the undersigned hereby certifies that the word count for the foregoing Petition for *Inter partes* Review totals 13,837 words, which is less than the 14,000 allowed under 37 CFR § 42.24.

Dated August 29, 2025

/Kenneth Wayne Darby Jr./
Kenneth W. Darby, Jr., Reg. No. 65,068
David M. Hoffman, Reg. No. 54,174
Fish & Richardson P.C.
60 South Sixth Street, Suite 3200
Minneapolis, MN 55402
T: 202-783-5070
F: 877-769-7945

Attorneys for Petitioners

CERTIFICATE OF SERVICE

Pursuant to 37 CFR §§ 42.6(e)(4)(i) *et seq.* and 42.105(b), the undersigned certifies that on August 29, 2025, a complete and entire copy of this Petition for *Inter partes* Review, Power of Attorney, and all supporting exhibits were provided via Federal Express, to the Patent Owner by serving the correspondence address of record as follows:

Hewlett Packard Enterprise
3404 E. Harmony Road
Mail Stop 79
Fort Collins, CO 80528
UNITED STATES
(888) 342-2156

/Crena Pacheco/
Crena Pacheco
Fish & Richardson P.C.
60 South Sixth Street, Suite 3200
Minneapolis, MN 55402
pacheco@fr.com