UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

DOLBY LABORATORIES, INC.,
Petitioner,

v.

INTERTRUST TECHNOLOGIES CORPORATION
Patent Owner.

Case IPR2020-00660
Patent No. 6,785,815

PETITION FOR *INTER PARTES* REVIEW

TABLE OF CONTENTS

I. INTRODUCTION 1

II. MANDATORY NOTICES UNDER 37 C.F.R. § 42.8 5

A. Real Party-In-Interest 5

B. Related Matters 5

C. Lead and Back-up Counsel, and Service Information 6

III. PAYMENT OF FEES 6

IV. REQUIREMENTS FOR *INTER PARTES* REVIEW 7

A. Grounds for Standing 7

B. Identification of Challenge 7

V. BACKGROUND 10

A. Summary of the '815 Patent 10

B. Deficient Priority Claim 13

VI. LEVEL OF ORDINARY SKILL IN THE ART 18

VII. CLAIM CONSTRUCTION 18

VIII. GROUNDS OF UNPATENTABILITY 19

A. Ground 1: Claims 42 and 43 are rendered obvious by Hanna in view of Stefik 19

1. Overview of Hanna 19

2. Overview of Stefik 23

3. Motivation to Combine 26

4. Claim Charts: Claims 42 and 43 are rendered obvious by Hanna in view of Stefik 30

B. Ground 2: Claims 42 and 43 are rendered obvious by Gennaro 42

1. Overview of Gennaro 42

2. Claim Charts: Claims 42 and 43 are rendered obvious by Gennaro 48

IX. SECONDARY CONSIDERATIONS 59

X. CONCLUSION 60

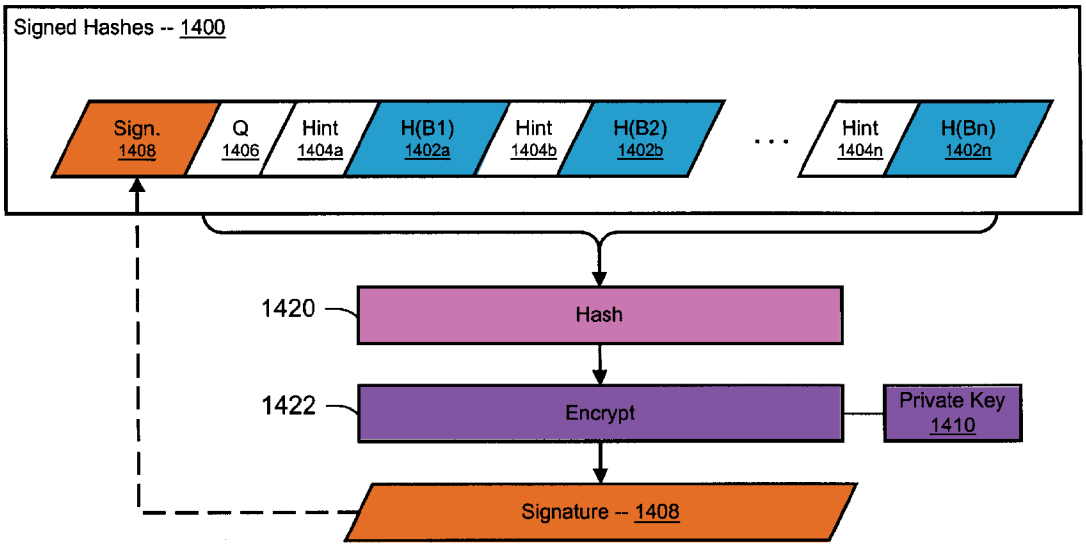
LIST OF EXHIBITS

Exhibit ("Ex-")	Description
1001	U.S. Patent No. 6,785,815 to Serret-Avila et al. (the "'815 patent")
1002	Declaration of Dr. Vijay K. Madisetti
1003	File History of U.S. Application No. 09/588,652, which led to U.S. Patent No. 6,785,815 (the "'652 Application")
1004	U.S. Provisional Application No. 60/138,171 to Serret-Avila et al. (the "Serret-Avila Provisional")
1005	Redline comparing Serret-Avila Provisional and the '815 patent
1006	International Publication No. WO 1999040702 to Hanna ("Hanna")
1007	U.S. Patent No. 5,629,980 to Stefik et al. ("Stefik")
1008	U.S. Patent No. 6,009,176 to Gennaro et al. ("Gennaro")
1009	Gennaro et al., How to Sign Digital Streams, I.B.M. T.J. Watson Research Center
1010	Declaration of Cornell University Library authenticating the Gennaro article (Ex. 1009)
1011	U.S. Patent No. 5,898,779 to Squilla et al.
1012	U.S. Patent No. 5,958,051 to Renaud et al.
1013	U.S. Patent No. 5,604,801 to Dolan et al.
1014	U.S. Patent No. 5,754,659 to Sprunk et al.
1015	U.S. Patent No. 5,907,619 to Davis
1016	U.S. Patent No. 6,697,948 to Rabin et al.
1017	Bruce Schneier, One Way Hash Functions, Dr. Dobb's Journal, September 1991, at 148-151.
1018	U.S. Patent No. 7,761,910 to Ransom et al.
1019	Declaration of Crena Pacheco ISO Petition

I. INTRODUCTION

Petitioner Dolby Laboratories, Inc. ("Petitioner") respectfully requests *inter partes* review of Claims 42 and 43 ("Challenged Claims") of U.S. Patent No. 6,785,815 (the "'815 patent") in accordance with 35 U.S.C. §§ 311-319 and 37 C.F.R. § 42.100 et seq.

The '815 patent discloses protecting data through encoding by way of a hash algorithm. Initially, the data is hashed to yield one or more hash ("check") values (*e.g.*, "1402a, 1402b, …, 1402n"). One or more of the check values is then hashed, and the resulting hash value is encoded (*e.g.*, "1420") with a sender's private key (*e.g.*, "1410") to create a digital signature (*e.g.*, "1408"), as shown in Figure 14 below. (*Id.*, 23:42-60).



(color-annotated Fig. 14); (Ex. 1002, ¶¶61-63).

Claims 42 and 43 of the '815 patent are directed to a method for managing use of a file of electronic data. (Ex. 1001, 32:13-14). The claimed method allows the integrity of the data to be checked before allowing it to be used. (*Id.*, 23:33-34). In operation, when a user requests to use a file of electronic data, *e.g.,* attempts to copy a file, the check value(s) (*e.g.,* "1402a, 1402b, …, 1402n") and corresponding digital signature (*e.g.,* "1408") are retrieved. (Ex. 1001, 23:1-5). The digital signature is decrypted using the sender's public key, which yields a first hash value. (*Id.*, 23:6-7). A hash function is applied to the check value(s), which yields a second hash value. (*Id.*, 23:7-10). The first hash value can be trusted because it comes from the sender's digital signature, which was encoded using the sender's private key. To verify the authenticity of the check value(s), the second hash value, which is derived from the check value(s), is compared against the first hash value, which is derived from the sender's digital signature. (*Id.*, 23:6-10). If the two hash values are the same, then the check value(s) are authentic, as any change to the transmitted check value(s) would cause the comparison to fail. (*Id.*) If the check values are verified to be authentic, they are used to verify the authenticity of the data in the transmitted file. (*Id.*, 23:10-14). The authenticity of the data is verified by hashing the data, and comparing the resulting hashes with the verified check values. (*Id.*) If the transmitted file is

authentic, then the request to use of the file is granted. (*Id.*, 23:14-16; Ex. 1002, ¶¶64-66).

Although the '815 patent claims priority to its provisional application that was filed on June 8, 1999, Claims 42 and 43 are not entitled to the benefit of the provisional application because the steps of the claims were not described in the provisional application. (Ex. 1002, ¶¶67-71). The claimed method was well-known in the art as of the June 7, 2000 filing date of the non-provisional application that led to the issuance of the '815 patent. (Ex. 1002, ¶¶49-60).

For example, Hanna describes a method and system for efficient authentication and integrity checking using hashes transmitted with the protected data. (Ex. 1006, *Title, Abstract*). It describes dividing a file into data packets, hashing the data packets, and applying a digital signature. (*Id.*, 3:6-17, 8:9-9:4, Fig. 2). The hashes and digital signature can be transmitted with the data. (*Id.*, 9:11-16, Fig. 2). The authenticity of the hashes can be verified using the digital signature (*Id.*, Fig. 2). The authenticity of the data packets can be verified by computing a hash function on the data packets, and comparing the resulting hash with hash values authenticated using the digital signature. (*Id.*, 9:21-10:16, 11:12-19, 11:24-27, Fig. 3; Ex. 1002, ¶¶75-78).

An authentication method for managing use of a file is also disclosed in Stefik. Stefik recognizes the same problem of unauthorized distribution and usage

of digital files. (Ex. 1007, 1:10-24). Stefik discloses a request mechanism in which a request to use a digital file is made, and granted if authenticity of the file is verified. (*Id.*, 15:33-36, 16:61-65, 42:26-27). Stefik discloses sending a digital certificate associated with the file, retrieving a check value (*e.g.,* "tamper-checking code") by decrypting the digital certificate, decrypting the file using "1-way hash function" to compute "a check code," and comparing the "check code" against the "tamper-checking code" to verify the authenticity of the file. (*Id.*, 42:38-62). Once the file has been authenticated, the request to use the file is granted. (*Id.*, 42:49-62; Ex. 1002, ¶¶79-85).

Even if Claims 42 and 43 are entitled to the June 8, 1999 filing date of the provisional application (they are not, as explained below), the claimed method was known in the art at least as of 1997. (Ex. 1002, ¶¶49-60). Gennaro describes a method of signing and authenticating digital streams. (Ex. 1008, *Abstract*). It describes dividing a digital stream into blocks, hashing each block, embedding the hash of a successor block in its preceding block to form a combined block, and finally signing the hash of the first combined block to create a digital signature. (Ex. 1008, 3:63-5:2). The hash of the first combined block and the digital signature are sent to a receiver of the digital stream. (Ex. 1008, 5:23-34). The authenticity of the hash is verified using the digital signature. (*Id.*) The authenticity of the first combined block is verified by computing a hash function

on the first combined block, and comparing the resulting hash with the hash value authenticated using the digital signature. (*Id.*; Ex. 1002, ¶¶124-136).

This Petition, and in particular the analysis in Section VIII, demonstrates that the Challenged Claims are unpatentable over the prior art and that Petitioner has a reasonable likelihood of prevailing with respect to the same.

II. MANDATORY NOTICES UNDER 37 C.F.R. § 42.8

A. Real Party-In-Interest

Pursuant to 37 C.F.R. § 42.8(b)(1), Petitioner identifies Dolby Laboratories, Inc., a Delaware corporation, and Dolby Laboratories, Inc., a California corporation, as real parties-in-interest. Petitioner further identifies Cinemark Holdings, Inc., AMC Entertainment Holdings, Inc., and Regal Entertainment Group (hereinafter "Cinemas"), as real parties-in-interest out of an abundance of caution, to avoid any dispute about such status based on any alleged business relationship between the Cinemas and Petitioner.

B. Related Matters

The '815 patent is currently the subject of four district court litigations: *Intertrust Technologies Corporation v. AMC Entertainment Holdings, Inc.*, Case No. 2:19-cv-00265 (E.D. Tex., filed August 7, 2019); *Intertrust Technologies Corporation v. Cinemark Holdings, Inc.*, Case No. 2:19-cv-00266 (E.D. Tex., filed August 7, 2019); *Intertrust Technologies Corporation v. Regal*

Entertainment Group, Case No. 2:19-cv-00267 (E.D. Tex., filed August 7, 2019); *Dolby Laboratories, Inc. v. Intertrust Corporation*, Case No. 3:19-cv-03371 (N.D. Cal., filed June 13, 2019).

C. Lead and Back-up Counsel, and Service Information

Lead Counsel	Backup Counsel
Scott A. McKeown Reg. No. 42,866 ROPES & GRAY LLP 2099 Pennsylvania Avenue, NW Washington, D.C. 20006-6807 Phone: 202-508-4740 Fax: 617-235-9492 scott.mckeown@ropesgray.com **Mailing address for all PTAB correspondence:** ROPES & GRAY LLP IPRM—Floor 43 Prudential Tower 800 Boylston Street Boston, Massachusetts 02199-3600	Mark Rowland Reg. No. 32,077 Keyna Chow *Pro Hac Vice Forthcoming* ROPES & GRAY LLP 1900 University Ave., 6th Floor East Palo Alto, CA 94303-2284 Phone: 650-617-4000 Fax: 617-235-9492 mark.rowland@ropesgray.com keyna.chow@ropesgray.com Leslie Spencer Reg. No. 47,105 Kilpatrick Townsend & Stockton LLP The Grace Building #21, 1114 6th Ave, New York, NY 10036 Phone: 212-775-8700 Fax: 212-775-8800 lspencer@kilpatricktownsend.com

III. PAYMENT OF FEES

The undersigned authorizes the Office to charge the fee required by 37 C.F.R. § 42.15(a) for this Petition for *inter partes* review to Deposit Account No. 18-1945. Any additional fees that might be due are also authorized.

IV. REQUIREMENTS FOR *INTER PARTES* REVIEW

A. Grounds for Standing

Pursuant to 37 C.F.R. § 42.104(a), Petitioner certifies that the '815 patent is available for *inter partes* review and that the Petitioner is not barred or estopped from requesting *inter partes* review challenging the claims of the '815 patent on the grounds identified herein.

B. Identification of Challenge

Pursuant to 37 C.F.R. §§ 42.104(b) and (b)(1), Petitioner requests *inter partes* review of the Challenged Claims and that the Board cancel the same as unpatentable. The '815 patent matured from U.S. Patent Application No. 09/588,652 filed June 7, 2000, and claims the benefit of Provisional Application No. 60/138,171 filed on June 8, 1999 (Ex. 1004). As discussed in Section V.B. below, the Challenged Claims are not entitled to the benefit of the Provisional Application. Accordingly, for the purpose of this Petition, the invention date of the Challenged Claims is June 7, 2000.

1. The Specific Art on Which the Challenge is Based

Petitioner relies upon the following prior art:

Exhibit 1006 – International Publication Number WO 1999040702 to Hanna ("**Hanna**"), which was published on August 12, 1999. Hanna is prior art to the Challenged Claims of the '815 patent under (Pre-AIA) 35 U.S.C. § 102(a) because

its publication date is before June 7, 2000. Hanna was not considered during prosecution of the '815 patent.

Exhibit 1007 – U.S. Patent No. 5,629,980 to Stefik et al. ("**Stefik**"), issued on May 13, 1997. U.S. Application No. 344,042, which led to Stefik, was filed on November 23, 1994. Stefik is prior art to the '815 patent under (Pre-AIA) 35 U.S.C. § 102(b), (a) & (e). Stefik was not considered during prosecution of the '815 patent.

Exhibit 1008 – U.S. Patent No. 6,009,176 to Gennaro et al. ("**Gennaro**"), issued on December 28, 1999 from U.S. Application No. 08/799,813, which was filed on February 13, 1997. Gennaro is prior art to the '815 patent under (Pre-AIA) 35 U.S.C. § 102(e). Gennaro is also prior art to the Challenged Claims of the '815 patent under (Pre-AIA) 35 U.S.C. § 102(a) because it was issued before June 7, 2000. Gennaro was not considered during prosecution of the '815 patent.

These references were not cited in an Information Disclosure Statement, identified by the Examiner, or applied in a rejection during the '815 patent's prosecution. The Examiner never considered the grounds presented herein or the testimony of Petitioner's expert Dr. Vijay Madisetti (Ex. 1002) regarding the prior art's scope and content. *See* Ex. 1003. The presented grounds are not cumulative of any prior art previously considered, and are not the same or substantially the same as prior art or arguments previously considered. Co-pending district court

proceedings also do not warrant the exercise of discretion under §314(a). *E.g.*, *Precision Planting, LLC v. Deere & Company*, IPR2019-01044, Paper 17, *9-18 (PTAB, Dec. 2, 2019). The district court proceeding includes multiple grounds of invalidity including §§101 and 112, and unique grounds under §§102 and 103 not at issue here, potentially enabling the district court to focus its limited trial time on different invalidity defenses. Moreover, Patent Owner only served its infringement contentions against Dolby and confirmed which claims it was asserting on February 27, 2020. The Board should not exercise its discretion to deny institution. §§314(a), 325(d).

2. Statutory Grounds on Which the Challenge is Based

Petitioner respectfully requests cancelation of the Challenged Claims on the following grounds under (Pre-AIA) 35 U.S.C. § 103(a):

Ground	Statute	Claim(s)	Prior Art
1	§ 103	42, 43	Hanna in view of Stefik
2	§ 103	42, 43	Gennaro

Pursuant to 37 C.F.R. § 42.204(b)(4), an explanation of how the Challenged Claims are unpatentable under the statutory grounds identified above, and that the Petitioner has at least a reasonable likelihood of prevailing on such grounds, including the identification of where each element of the claim is found in the prior art, is provided in Section VIII, below. Pursuant to 37 C.F.R. § 42.204(b)(5), the

exhibit numbers of the supporting evidence relied upon to support the challenges and the relevance of the evidence to the challenges raised, including identifying specific portions of the evidence that support the challenges, are provided in Section VIII, below.

This Petition is supported by the Declaration of Vijay Madisetti (Ex. 1002, ¶¶1-172), which discusses the disclosure of the provisional application of the '815 patent, and the scope and content of the prior art at the relevant time of the purported '815 patent invention.

V. BACKGROUND

A. Summary of the '815 Patent

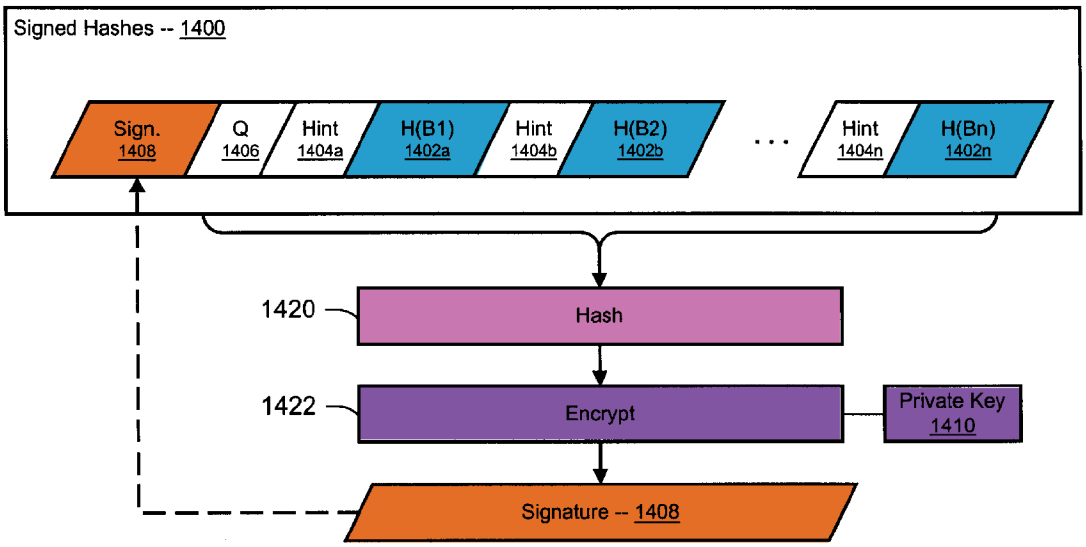
The '815 patent was filed on June 7, 2000, issued on August 31, 2004, and claims the benefit of a provisional application filed on June 8, 1999.

The '815 patent relates generally to methods and systems for encoding and protecting data using techniques including digital signatures. (Ex. 1001, Title, Abstract). The techniques protect the data against unauthorized use or modification, and facilitate using data "in a certain manner—such as copying, moving, viewing, or printing the data." (*Id.*, 1:23-28, 4:65-5:3; Ex. 1002, ¶61).

The '815 patent discloses an embodiment in which the use of electronic data is managed by retrieving a file containing one or more check values and a digital signature derived from the check values, using the digital signature to authenticate

the check values, and using the authenticated check values to authenticate at least a portion of the data. (Ex. 1001, 4:50-58, 22:63-23:14; Ex. 1002, ¶62).

Claims 42 and 43 of the '815 patent are directed to this method for managing use of a file of electronic data. (Ex. 1001, 32:13-34). The claimed method, which is illustrated in Figures 13 and 14, allows the integrity of the data to be checked before granting access to the file. (*Id.*, 23:33-34). Figure 14 illustrates how a signed hash file 1400 is created such that it can be provided to a user to verify the authenticity of a content file:



(color-annotated Fig. 14). The original content file is divided into different portions. Those portions are hashed to yield a plurality of **hash values H(Bn) 1402**. These hash values, along with other data (such as hints 1404 and quality indicator 1406), are further hashed to yield the **hash 1420**. The **hash 1420** is

encrypted using a **private key 1410** to create a **digital signature 1408**. Both the **digital signature 1408** and the **hash values 1402** of the original content file are included in the signed hash file 1400. (*Id.*, 23:40-60; Ex. 1002, ¶63).

Figure 13, color-annotated below, shows how the signed hash file is provided to a user who has made a request to use the content file. This request can refer to any attempt to use the file, such as copying, moving, viewing, or printing the data in the file. (Ex. 1001, 4:66-5:9; Ex. 1002, ¶64).

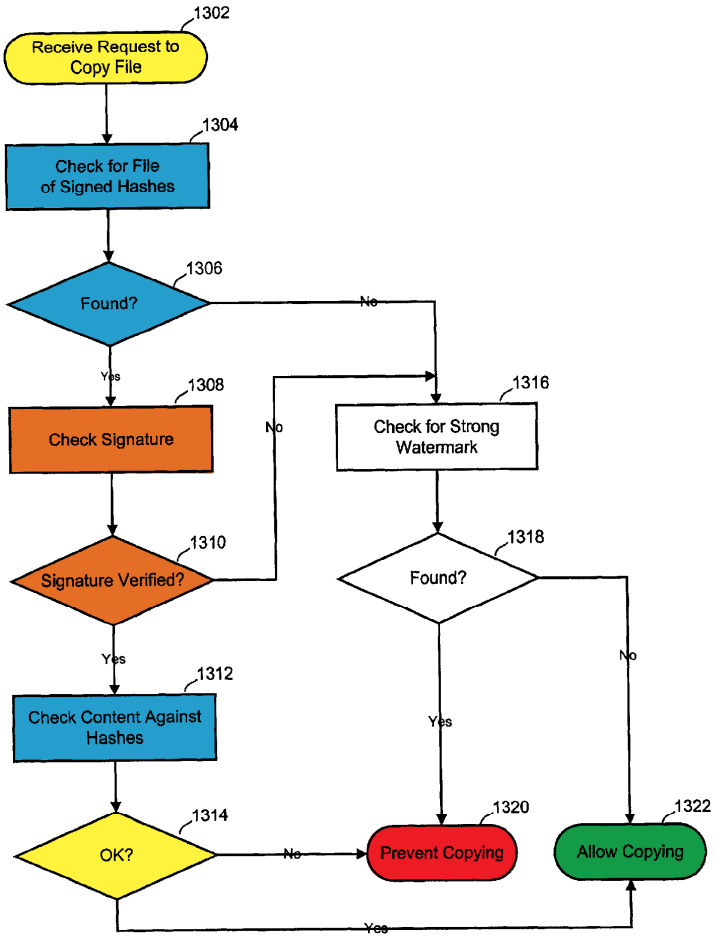


FIG. 13

In response to the request to copy the file in step 1302, the **check values ("signed hashes")** and the **digital signature** are retrieved as shown in **step 1304** and **step 1308**. (*Id.*, 23:1-5). In **step 1308**, the **digital signature** is decrypted using the sender's public key, which yields a **first hash value**. (*Id.*, 23:6-7). The **first hash value** can be trusted because it comes from the sender's **digital signature**, which was encoded using the sender's private key. A hash function is applied to the **check values**, which yields a second hash value. (*Id.*, 23:7-10). In **step 1310**, to verify the authenticity of the check values, the second hash value, which is derived from the **check values**, is compared against the **first hash value**, which is derived from the sender's **digital signature**. (*Id.*, 23:6-10). If the two hash values are the same, then the **check values** are authentic, as any change to the check values would cause the comparison to fail. (*Id.*; Ex. 1002, ¶65).

In **step 1312**, if the **check values** are verified to be authentic, they are used to verify the authenticity of the content file. (Ex. 1001, 23:10-14). This is done by hashing the appropriate portions of the file, and comparing the resulting hashes with the **check values**. (*Id.*) In step 1314, if the file is authentic, then the requested use of the file is granted (**step 1322**). (*Id.*, 23:14-16; Ex. 1002, ¶66).

B. Deficient Priority Claim

The Challenged Claims—Claim 42 and Claim 43 (that depends on Claim 42)—of the '815 patent are not entitled to the benefit of its U.S. provisional

application No. 60/138,171 to Serret-Avila et al. (Ex. 1004, "Serret-Avila Provisional") that was filed on June 8, 1999 because the Serret-Avila Provisional fails to provide adequate written description support for at least two limitations of Claim 42: (1) "retrieving at least one digital signature and at least one check value associated with the file"; and (2) "verifying the authenticity of the at least one check value using the digital signature."

1. Legal Standard

For a non-provisional application to be afforded the priority date of a provisional application, the provisional application must adequately describe the claimed invention of the non-provisional application in "full, clear, concise, and exact terms" to satisfy the written description requirement of 35 U.S.C. § 112. *New Railhead Mfg., L.L.C. v. Vermeer Mfg. Co.*, 298 F.3d 1290, 1294 (Fed. Cir. 2002). To satisfy the written description requirement of 35 U.S.C. § 112, the disclosure of the provisional application must "reasonably convey[] to those skilled in the art that the inventor had possession of the claimed subject matter as of the filing date." *Ariad Pharms., Inc. v. Eli Lilly & Co*., 598 F.3d 1336, 1351 (Fed. Cir. 2010). This is an "objective inquiry into the four corners of the specification from the perspective of a person of ordinary skill in the art." *Id.*

"[A]ll of the elements and limitations" of the claimed invention must be described to show possession. *Univ. of Rochester v. G.D. Searle & Co.*, 358 F.3d

916 (Fed. Cir. 2004) (quoting *Hyatt v. Boone*, 146 F.3d 1348, 1353 (Fed. Cir. 1998)). As such, mere descriptions of individual elements do not suffice if the claimed invention as a whole is not adequately described. *See Quake v. Lo*, 928 F.3d 1365, 1373-74 (Fed. Cir. 2019). Further, the application must actually describe the claimed subject matter, and not merely include a description that renders the invention obvious. *Ariad*, 598 F.3d at 1352 (citing *Lockwood v. Am. Airlines*, 107 F.3d 1565, 1571-72 (Fed. Cir. 1997)).

2. The Serret-Avila Provisional Fails to Provide Written Description Support for the Recited Steps of "retrieving at least one digital signature and at least one check value associated with the file" and "verifying the authenticity of the at least one check value using the digital signature."

The Serret-Avila Provisional fails to describe retrieving a check value associated with the file and verifying the authenticity of that check value using the digital signature. The portions of the specification that describe these limitations of Claim 42 do not appear in the Serret-Avila Provisional. Ex. 1005 (comparing the Serret-Avila Provisional with the '815 specification). Specifically, Serret-Avila Provisional does not include Figures 13 and 14, their corresponding disclosures (Ex. 1001, 23:4-10, 23:54-60), or the corresponding summary of the embodiment. (Ex. 1001, 4:50-58; Ex. 1002, ¶¶67-69).

Indeed, nothing in the Serret-Avila Provisional supports these limitations of Claim 42. The Serret-Avila Provisional describes and depicts conventional cryptographic signatures techniques in Figure 2A through Figure 2C[1] in which digital signatures are inserted into a data stream. As illustrated in Figure 2A, each digital signature corresponds to a block of data, and is either inserted into the stream adjacent to the corresponding block or its bits are distributed through the corresponding block as a watermark. (Ex. 1004, 10-11). Although Figure 2B depicts applying a hash function on the data and then encrypting the resulting hash value to create a digital signature, only the signature (and not the hash value) is inserted into the data stream. (Ex. 1004, 11). Figure 2C reinforces this point when it depicts that the content stream to be verified is composed of data blocks and signatures only. (Ex. 1004, 3). Accordingly, the description of the conventional techniques in the Serret-Avila Provisional at best suggests retrieval of the digital signature, and not the check/hash values associated with the data. (Ex. 1002, ¶70).

The Serret-Avila Provisional discloses modifying the conventional techniques by embedding a block's signature in the watermark of a following block, and by embedding an additional "strong" watermark. (Ex. 1004, 12, 21-23). Because watermarking cannot insert a large amount of data into a stream, the

[1] These figures do not appear in the '815 specification. They appear to be similar to Figs. 2A and 2B of the '815 specification.

Serret-Avila Provisional further discloses using a single "shared signature" formed by encrypting a concatenation of hashes corresponding to multiple blocks of data. (Ex. 1004, 20). Figure 6[2] depicts the "shared signature" technique. Like the other embodiments disclosed, the "shared signature" technique does not involve retrieving a check/hash value associated with a file, or verifying the authenticity of the check/hash value using the digital signature. (Ex. 1004, 20-21; Ex. 1002, ¶71).

Because the Serret-Avila Provisional fails to describe retrieving a check/hash value associated with the file, it fails to describe verifying the authenticity of the check/hash value as claimed. More specifically, Claim 42 requires "verifying the authenticity of the at least one check value *using the digital signature*." The Serret-Avila Provisional contains no disclosure where a check/hash value is verified using a digital signature.

The disclosures in the Serret-Avila Provisional would not suggest to a person of ordinary skill in the art (hereinafter "POSITA") that the inventor is in possession of the recited "retrieving" step and the first "verifying" step of Claim 42. Therefore, the Serret-Avila Provisional fails to provide adequate written description support for the claimed steps reciting retrieving and verifying the

[2] Figure 6 does not appear in the '815 specification, but it appears to be similar to Figure 8 of the '815 specification.

authenticity of at least one check value associated with the file using the digital signature. (Ex. 1002, ¶¶67-68).

Accordingly, the Challenged Claims 42 and 43 are not entitled to the priority date of June 8, 1999, which is the filing date of the Serret-Avila Provisional.

VI. LEVEL OF ORDINARY SKILL IN THE ART

The level of ordinary skill in the art is evidenced by the prior art. (*See In re GPAC Inc.*, 57 F.3d 1573, 1579 (Fed. Cir. 1995) (determining that the Board did not err in adopting the approach that the level of skill in the art was best determined by references of record.)) The prior art discussed herein, and in the declaration of Dr. Vijay K. Madisetti (Ex. 1002) demonstrates that a POSITA, at the time the '815 patent was filed, would have had a background in electronic data protection and distribution, a minimum of a bachelor of science's degree in computer science, electrical engineering, mathematics, or a related field, and approximately four years of professional experience or equivalent study in the design of electronic systems for data protection and distribution. Additional graduate education could substitute for professional experience, or significant experience in the field could substitute for formal education. (Ex. 1002, ¶¶1-48)

VII. CLAIM CONSTRUCTION

Only terms necessary to resolve the controversy need to be construed. *Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co. Ltd.*, 868 F.3d 1013, 1017

(Fed. Cir. 2017). Since the challenged claims are unpatentable over the prior art under any construction that may be reasonably proposed, Petitioner does not propose any terms for construction in this Petition.

VIII. GROUNDS OF UNPATENTABILITY

Although the '815 patent purports to have invented a system and method for encoding and protecting data, such system and method was known in the field of electronic content protection and distribution prior to the earliest possible priority date. As demonstrated below, the prior art references render the Challenged Claims unpatentable. (Ex. 1002, ¶¶74, 123).

A. Ground 1: Claims 42 and 43 are rendered obvious by Hanna in view of Stefik

1. Overview of Hanna

Hanna, entitled "Method and apparatus for efficient authentication and integrity checking using hierarchical hashing," is directed to a method and system for signing data and verifying data. (Ex. 1006, *Title*, *Abstract*). Like the '815 patent that seeks to detect unauthorized modification of electronic data (Ex. 1001, 2:6-10, 2:31-33), Hanna seeks to protect against unintentional data corruption and malicious acts that cause errors in data processing. (Ex. 1006, 3:1-4; Ex. 1002, ¶75).

Hanna explains that the prior art systems do not allow a single digital signature to apply to an arbitrary amount of data (but rather, assign a digital

signature to each individual packet of data) and do not allow the data to be verified as it is received. (Ex. 1006, 2:15-28). To solve this problem, Hanna uses a single signed hash block, allowing a single digital signature to protect the data set from both data corruption and malicious acts that cause errors in data processing. The hash block is retrieved with the data and the digital signature. (*Id.*, Abstract). Receiving the hashes before the data also allows the data packets to be quickly verified as they are received. (*Id.*, 3:6-12; Ex. 1002, ¶76).

Data signing involves dividing a data set into packets, hashing the data within each of the packets to produce a hash block of hash values, and applying a digital signature to the hash block. (Ex. 1006, 3:14-17). This approach of data signing is shown in Figure 2 (reproduced below):

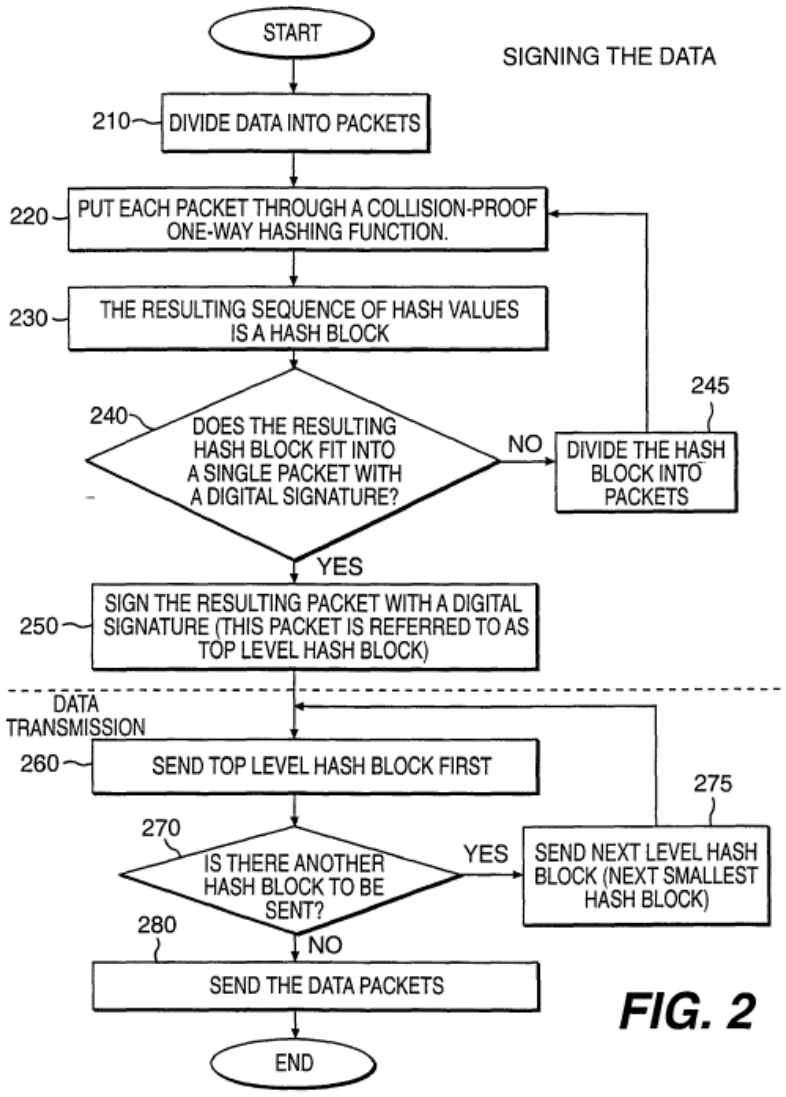


FIG. 2

In step 210, data signing begins by dividing a data set into small packets. In step 220, a one-way hash function is applied to each data packet. In step 230, the application of the hash function yields a sequence of hash values, *i.e.*, a hash block. In steps 240 and 250, the hash block is signed to create a digital signature if it can fit into a single packet with the digital signature; this packet is referred to as a top level hash block. (*Id.*, 8:9-9:7). In step 260, the top level hash block containing the

hash values and the digital signature are transmitted, followed by the transmission of the data packets in step 280. (*Id.*, 9:11-13; Ex. 1002, ¶77).

The verifying process involves receiving the top level hash block containing the hash values and the digital signature, verifying the hash values using the digital signature, and verifying the data packets using the verified hash values. (Ex. 1006, 3:18-22). Such verifying process is shown in the flowchart in Figure 3 (reproduced below):

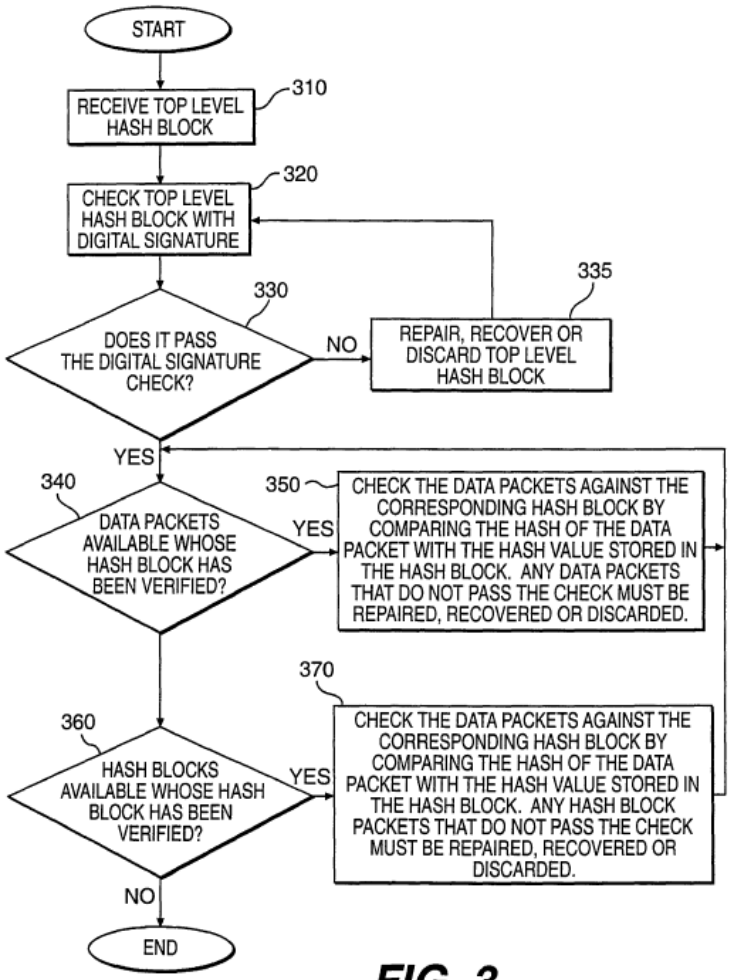


FIG. 3

Figure 3 shows that the verification starts with receiving the top level hash block containing the digital signature and the hash block (step 310). *See* steps 250 and 260 in Figure 2. The hash block is verified using the digital signature (step 320). If the digital signature check fails, then the hash block must be repaired, recovered, or discarded before verifying data packets (step 335). If the digital signature check passes, then it is determined whether data packets are available to be verified (steps 340). The data packets are verified by computing a hash function on the packets and comparing the resulting hash values with the hash values stored in the hash block (step 350). If this check fails, the data packet is corrupt and should be repaired, recovered, or discarded (step 350). (*Id.*, 9:22-10:16; Ex. 1002, ¶78).

2. Overview of Stefik

Stefik, entitled "System for Controlling the Distribution and Use of Digital Works," is directed to a method and system for controlling usage and distribution of digital works, such as software. (Ex. 1007, *Title, Abstract*). Stefik recognizes the problem of unauthorized distribution and usage of digital works. (*Id.*, 1:10-24). Stefik's invention allows the owner of a digital work to attach usage rights to the work that define how the work may be used and distributed. (*Id.*, 3:51-60; Ex. 1002, ¶79).

Stefik explains that a system responsible for the use and distribution of digital works must ensure the integrity of repositories where digital works are

exchanged. (Ex. 1007, 12:41-50). Such integrity has three parts: physical integrity, communications integrity, and behavior integrity. (*Id.*, 12:41-50) Physical integrity refers to the integrity of the physical devices themselves. (*Id.*, 12:51-52). Communications integrity refers to the integrity of the communications between repositories such that a repository can present proof that it is the intended repository and can detect "imposters" and malicious interference. (*Id.*, 13:11-23). Communications integrity is achieved by security measures involving encryption and exchange of digital certificates. (*Id.*, 13:19-23). Behavioral integrity refers to the integrity in what repositories do. Behavioral integrity is maintained by requiring software be distributed with proof that it is certified, *i.e.*, a digital certificate. Software will only be installed after verifying the authenticity of the software using the digital certificate. (*Id.*, 13:30-40; Ex. 1002, ¶80).

Stefik describes a rendering system having at least two components: a rendering device and a repository. A rendering device renders a digital work; examples of such rendering device are a computer system, a digital audio system, or a printer. (Ex. 1007, 8:24-28). A repository has a user interface that permits a user to request access to the digital work. (*Id.*, 16:61-67). When a user requests access to a digital work, the repository will initiate various transactions. (*Id.*, 26:38-39; Ex. 1002, ¶81).

In one transaction example, the repository receives a request from a user to install a player. (Ex. 1007, 42:25-27). Such player may be used to play digital movies/music/video game, run a computer program, or display a document. (*Id.*, 19:46-55). Stefik defines such a request as an "Install transaction"; it explains, "[i]n a typical case, the requester repository is a rendering repository and the software would be a new kind or new version of a player. Also in a typical case, the software would be copied to file system of the requester repository before it is installed." (*Id.*, 42:26-31; Ex. 1002, ¶82).

In operation, the requester repository sends a server a message indicating the software to be installed. (Ex. 1007, 42:26-35). The requester repository "extracts a copy of the digital certificate for the software." (*Id.*, 42:38-42). A digital certificate is similar to a digital signature in that both can be used to verify the source of the data as part of a security measure. (Ex. 1002, ¶ 83). To certify the software before installation, the requester repository decrypts the digital certificate using a public key to retrieve "a tamper-checking code" associated with the software, and a key for decrypting the software. (Ex. 1007, 42:42-48). The requester repository then verifies the authenticity of the software using the tamper-checking code. This is done by decrypting the software using the key from the digital certificate, and then using a "1-way hash function" to compute "a check code." (*Id.*, 42:49-54). This "check code" is then compared against the "tamper-

checking code" to "assure[] that the contents of the software … have not been tampered with." (*Id.*) If the "check code" matches with the "tamper-checking code," then the software is verified, and the requester repository grants the user's request to install the software by continuing with the installation process. (*Id.*, 42:49-43:2; Ex. 1002, ¶¶83-84).

When performing one or more transactions in response to a user request, a rendering repository may need to request access to a digital work from another repository, if the user's request requires such access. The request by the rendering repository is made for a stated purpose, such as to obtain a copy of the digital work, corresponding to a specific usage right. (Ex. 1007, 7:19-23). The repository receiving this request checks the usage rights associated with the digital work to determine if the access to the digital work may be granted. If access is granted, the digital work is transmitted to the rendering repository. (*Id.*, 7:23-32). Install rights, such as might be applied to a new type of player within the rendering repository, may be one of the usage rights associated with a digital work. (*Id.*, 20:48-53; Ex. 1002, ¶85).

3. Motivation to Combine

Hanna discloses a computer system that includes a bus or other communication mechanism for communicating information, and a processor coupled with the bus for processing information. (Ex. 1006, 5:2-5). It explains

that in response to the processor executing instructions, signed or verified data is provided by the computer system and the instructions cause the processor to perform the steps of signing or verifying the data. (Ex. 1006, 5:22-28). The computer system further includes a communication interface that provides a two-way data communication coupling to a network, such as a local network, or the Internet. (*Id.*, 6:26-7:7:14). The computer system can send messages and receive data, including program code transmitted from a server in response to a request, through such network. (*Id.*, 7:15-23). Accordingly, Hanna describes a computer system that can receive and sign or verify data that has been requested (e.g., program code transmitted from a server requested by another computer), and such requests are processed in accordance to its disclosed method and system of signing and verifying data. (Ex. 1002, ¶86).

Hanna discloses that certain types of business activities create the need to transfer information in a secure manner between systems that are remote from each other. Hanna provides an example that banks periodically "backup" their computer files to a remote central database and need to know that these files were successfully copied to the remote database without having been changed or corrupted during the process. (Ex. 1006, 1:10-13). Video conferencing is another example of an application that requires secure transmission of information, such as video, voice, and digital data. (*Id.*, 1:13-15; Ex. 1002, ¶87).

A POSITA would have recognized that in order to download an application program from a server over a network, copy and store computer files for backup to a remote database, or transmit data files in a video conference, as disclosed in Hanna, there would necessarily be communication to convey the electronic files for use in a predefined manner, such as copying and installing an application program, copying and storing or retrieving backup files, and playing video, audio or other digital data files. A POSITA also would have recognized from Hanna that a file may be checked when received or retrieved for such use. (Ex. 1006, 7:21-23). Thus, a POSITA would associate Hanna's process for signing and verifying data with a request to use that data, and granting that request if the data is verified. (Ex. 1002, ¶¶49-60, 88).

To the extent it may be argued that Hanna does not *explicitly* describe features claimed in the '815 patent, such as receiving a request to use a file of electronic data in a predefined manner, and granting a request to use the file in the predefined manner, such steps were disclosed by Stefik. A POSITA would have been motivated to look to and incorporate Stefik's disclosures of receiving and granting requests to use files of electronic data in a remote access environment (e.g., a repository different than the one in which the file is stored). (Ex. 1007, 16:43-67, 42:25-43:4). This is at least because, as disclosed in Hanna, the server and the computer downloading the application program, the bank and the remote

central database, and the systems of the participants in a video conference, are remote from each other. (Ex. 1002, ¶89).

In addition, both references are from the same field of endeavor (digital file security) and address the problem of unauthorized manipulation of digital files. Likewise, both use a digital authentication mechanism (signature/certificate) that identifies the source of the digital file, and hash functions that check the integrity of the data. (Ex. 1007, 1:10-24, 42:25-64; Ex. 1006, 1:6-8, 1:28-31, 3:1-4). Therefore, a POSITA would have been further motivated to modify Hanna to include a request/grant process as taught by Stefik because it would facilitate timely detection of data corruption in a file of electronic data, *i.e.*, when data that could be corrupted or hacked during transmission is about to be used. (Ex. 1002, ¶90).

A POSITA also would have been motivated to modify the data protection system of Hanna to include receiving a request to use a file of electronic data in a predefined manner, and granting that request, as taught by Stefik. Hanna and Stefik teach granting of the request based upon satisfying certain conditions, such as the requested use meeting predefined usage rights, and/or authenticating of at least a portion of the file. (Ex. 1007, 7:19-30; Ex. 1002, ¶91).

Accordingly, it would have been clear to a POSITA that there would be a reasonable expectation of success in utilizing the request/grant process of Stefik in

the data protection system of Hanna, in the manner for which each was designed, for managing use of an electronic file, because, as discussed above, the computer system in Hanna already has a communication mechanism in place to facilitate the request/grant process as taught in Stefik. (Ex. 1006, 5:2-5, 6:26-7:7:23; Ex. 1002, ¶¶49-60, 92).

4. Claim Charts: Claims 42 and 43 are rendered obvious by Hanna in view of Stefik

Claim 42	Hanna in view of Stefik
[42pre][3] A method for managing at least one use of a file of electronic data, the method including:	**Hanna** discloses a method for managing at least one use (e.g., "cop[ying]", "download[ing]", "retriev[ing]", "receiving", "execut[ing]", stor[ing]") of a file of electronic data (e.g., "computer files" or "a data set"). (Ex. 1006, Abstract, 1:10-13, 3:6-22, 7:15-26, 12:3-4, Figs. 2-5; Ex. 1002, ¶93). To the extent that it is argued that a method for managing a use of an electronic file is not already disclosed in Hanna, **Stefik** discloses a method for managing at least one use (*e.g.,* "install[ation]") of a file of electronic data (*e.g.,* "a digital work" or "software"). (Ex. 1002, ¶¶94-96). • "An Install transaction is ***a request to install a digital work*** as runnable software on a repository. In a typical case, the requester repository is a rendering repository and the software would be a new kind or new version of a player. Also in a typical case, the software would be copied to file system of the requester repository before it is installed." (Ex. 1007, 42:26-31) • "The requester sends the server an Install message. ***This message indicates the work to be installed***, the version

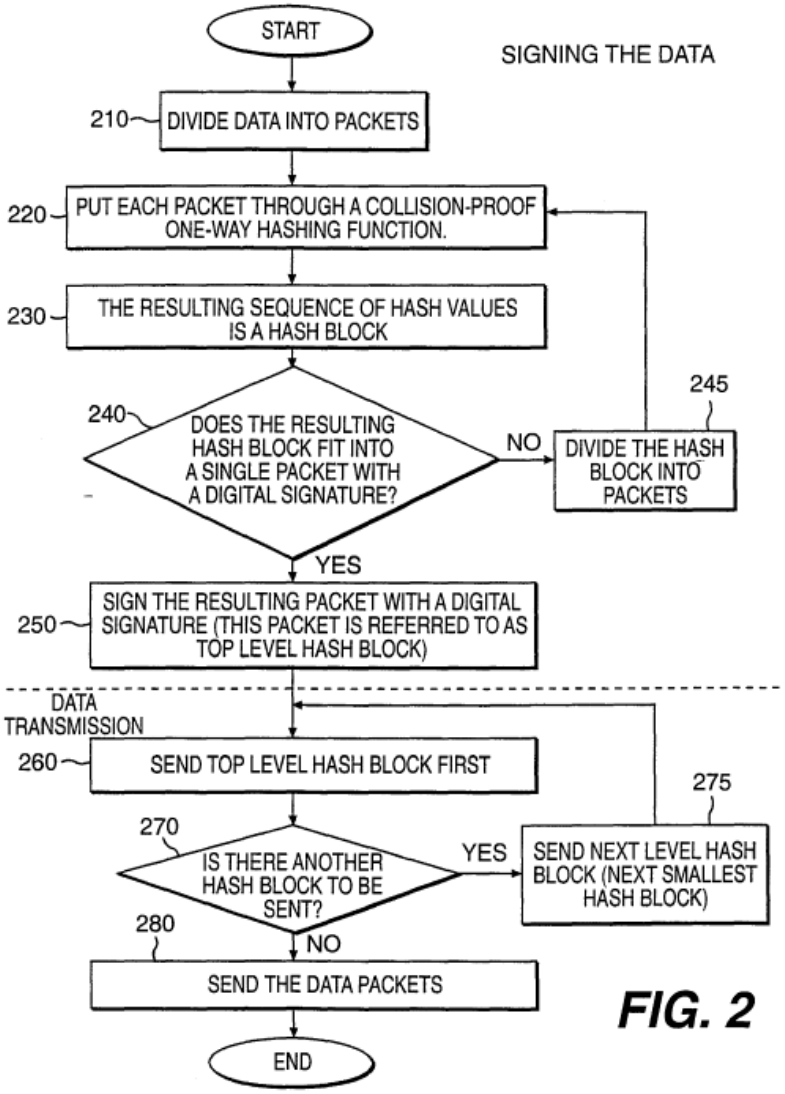
[3] To the extent they are limiting, the preambles of all Challenged Claims are nonetheless met by the art of record.

Claim 42	Hanna in view of Stefik
	of the Install right being invoked, and the file data for the work (including its size)." (*Id.*, 42:32-35) • *See also, e.g.,* Ex. 1007, *Abstract,* 3:51-61, 4:29-36, 6:36-56, 19:51-55, 51:1-5, 51:64-67; Ex. 1002, ¶¶49-50.
[42a] receiving a request to use the file in a predefined manner;	**Hanna** discloses "transmit[ting] a requested code for an application program." (Ex. 1006, 7:16-18, 7:24-26). A POSITA would recognize that a request to copy, store and/or execute the program has been made. Hanna also discloses "back[ing] up" computer files to a remote central database, and "secure transmission of information" in "video conferencing." (Ex. 1006, 1:10-15). A POSITA would recognize that in each case a request to transmit, copy and/or store files of electronic data is made. (Ex. 1002, ¶¶97-98). To the extent it is argued that a "request" to use the file in a predefined manner is not disclosed in Hanna, **Stefik** discloses receiving a request from a user via the user interface of its repository to use the file in a predefined manner (*e.g.,* "When a user requests access to a digital work … request specific transactions are performed"; "An Install Transaction is a request to install a digital work as runnable software on a repository"). (Ex. 1007, 26:37-47, 42:26-31; Ex. 1002, ¶99). Stefik further discloses a repository making a request to another repository to use the file in a predefined manner, such as to copy and install software (*e.g.*, "Repository 2 may then request access to the Digital Work for a stated purpose … for example… to obtain a copy of the digital work. The purpose will correspond to a specific usage right"; "Configuration-Code: = Install/Uninstall lists a category of rights for installing and uninstalling software on a repository (typically a rendering repository). This would typically occur for the installation of a new type of player within the rendering repository.") (Ex. 1007, 7:19-23, 20:48-52; Ex. 1002, ¶100). Such request by the rendering repository is to use a file in a predefined manner. Specifically, the request is made for a

Claim 42	Hanna in view of Stefik
	stated purpose, such as to obtain a copy of the digital work, corresponding to a specific usage right. (Ex. 1007, 7:19-23). The repository receiving this request checks the usage rights associated with the digital work to determine if access to the digital work may be granted. If access is granted, the digital work is transmitted to the rendering repository. (Ex. 1006, 7:23-32). Install rights, such as might be applied to a new type of player within the rendering repository, may be one of the usage rights associated with a digital work. (Ex. 1006, 20:48-53; Ex. 1002, ¶101). • "The digital work remains securely in Repository 1 until a request for access is received. The request for access begins with a session initiation by another repository. Here a Repository 2 initiates a session with Repository 1, step 103... Assuming that a session can be established, ***Repository 2 may then request access to the Digital Work for a stated purpose***, step 104. The purpose may be, for example, to print the digital work or ***to obtain a copy of the digital work***. ***The purpose will correspond to a specific usage right.***" (Ex. 1007, 7:13-23) • "At a minimum, the user interface must ***permit a user to input information such as access requests*** and alpha numeric data and provide feedback as to transaction status. The user interface will then cause the repository to initiate the suitable transactions to service the request." (*Id.*, 16:61-65). • "Grammar element 1508 'Configuration-Code:=***Install|Uninstall' lists a category of rights for installing and uninstalling software on a repository*** (typically a rendering repository.) This would typically occur for the installation of a new type of player within the rendering repository." (*Id.*, 20:48-52) • "REPOSITORY TRANSACTIONS: ***When a user requests access to a digital work, the repository will initiate various transactions.*** The combination of transactions invoked will depend on the specifications

Claim 42	Hanna in view of Stefik
	assigned for a usage right. There are three basic types of transactions, Session Initiation Transactions, Financial Transactions and Usage Transactions… Finally, ***request specific transactions are performed***. (*Id.*, 26:37-47) • "***An Install transaction is a request to install a digital work as runnable software on a repository***. In a typical case, the requester repository is a rendering repository and the software would be a new kind or new version of a player. Also in a typical case, the software would be copied to file system of the requester repository before it is installed." (*Id.*, 42:26-31) • "The requester sends the server an Install message. ***This message indicates the work to be installed***, the version of the Install right being invoked, and the file data for the work (including its size)." (*Id.*, 42:32-35) • *See also, e.g.,* Ex. 1007, *Abstract,* 3:51-61, 4:13-23, 6:36-56, 51:1-5, 51:64-67; Ex. 1002, ¶¶58-59.
[42b] retrieving at least one digital signature and at least one check value associated with the file;	**Hanna** discloses retrieving at least one digital signature (*e.g.,* "digital signature") and at least one check value (*e.g.,* "hash values" in a hash block) associated with the file (*e.g.,* "a data set"). Hanna explains that the verification begins with retrieving the top level hash block (*e.g.,* in step 260 of Fig. 2), which includes a digital signature (*e.g.,* "digital signature" in step 250 of Fig. 2) and hash values (*e.g.,* hash values that are part of the "top level hash block" in step 250 of Fig. 2). (Ex. 1006, 9:1-24). The hash values are associated with the data file because the hash values are computed by applying a one-way hash function to the data packets (*e.g.,* as described in steps 220 and 230 of Fig. 2). (Ex. 1006, 8:14-17; Ex. 1002, ¶¶102-104). • "Once the data is broken into packets, a collision-proof, one-way hash function is applied to each packet (step 220). Each application of the hash function will produce a single hash value. The application of the hash function to each of the data packets will produce ***a sequence of***

Claim 42	Hanna in view of Stefik
	hash values. This sequence is called a hash block (step 230)." (Ex. 1006, 8:14-17) • "If, however, the hash block originally created by the hashing of the data packets (step 230) is small enough to fit in a single packet with the digital signature, there is no need to go through the steps of breaking down the hash block and creating another higher level hash block and ***the top level hash block remains the only hash block***." (*Id.*, 9:1-4) • "Systems consistent with the present invention ***apply a digital signature to the top level hash block packet (step 250).*** No signature need be applied to any of the other hash blocks or data packets. The single digital signature applied to the top level packet is sufficient to verify all of the data." (*Id.*, 9:5-8) • "In accordance with the data signing procedure, ***the top level hash block packet with the single digital signature***, the hierarchy of lower level hash blocks, and the data ***are sent to a receiver***." (*Id.*, 9:18-20) • "To verify data, the digital signature on the top level packet must be verified first (step 320)." (*Id.*, 9:23-24)

Claim 42	Hanna in view of Stefik
	 **FIG. 2** • *See also, e.g.,* Ex. 1006, Figs. 3-5; Ex. 1002, ¶¶51-57.
[42c] verifying the authenticity of the at least one check value using the digital signature;	**Hanna** discloses verifying the authenticity of the at least one check value (*e.g.,* "hash values" in a hash block) using the digital signature (*e.g.,* "digital signature"). (Ex. 1002, ¶105). After the hash block containing the sequence of the hash values and the digital signature is received (step 310 in Fig. 3), Hanna discloses checking the hash block containing the sequence of the hash values using the digital signature (step 320 in Fig. 3). (Ex. 1002, ¶106).

Claim 42	Hanna in view of Stefik

- "Fig. 3 is a flowchart of the steps used in the data receiving and verification procedure for systems consistent with the present invention." (9:22-23)

The top level hash block is checked with the digital signature (step 320 in Fig. 3).

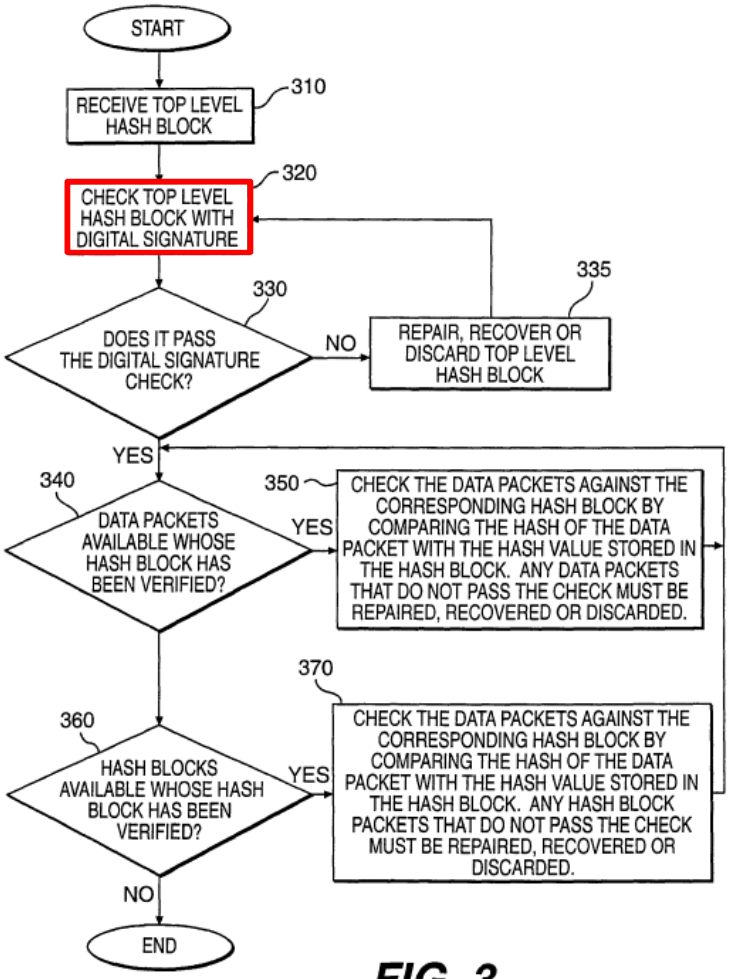


FIG. 3

(annotated with a red box); (Ex. 1002, ¶107).

- "If the signature fails this verification step or it is determined that the packet was corrupted during transmission, the top level packet must be repaired or recovered before checking the other packets (steps 330, 335)." (Ex. 1006, 9:24-26)

Claim 42	Hanna in view of Stefik
	• "***Data verification begins with checking the digital signature on the top level hash block 502***. If the top level hash block 502 passes this check, the next level hash block 501 is verified." (*Id.*, 11:12-13) • *See also, e.g.,* Ex. 1006, Fig. 5, 11:12-19; Ex. 1002, ¶54.
[42d] verifying the authenticity of at least a portion of the file using the at least one check value; and	**Hanna** discloses verifying the authenticity of at least a portion of the file (*e.g.,* a "data packet" of a data set) using the at least one check value (*e.g.,* "hash values" in a hash block). (Ex. 1002, ¶108). Hanna discloses that the authenticity of a portion of the file is verified using the verified hash values in the hash block. This is done by applying "[a] hash function" to each portion of the file*, i.e.,* data packet (*e.g.,* A_{10}, A_{11}, A_{12}), and comparing the resulting hash with the hash value (*e.g.,* B_4) in the hash block in the level about it. (Ex. 1002, ¶¶109-110). • "The hierarchical structure used in the method cryptographically protects ***individual portions of the data*** and makes it easier to recognize corruption." (Ex. 1006, 3:10-11) • "Finally, the data is checked against the hash block containing the hashes of the data set." (*Id.*, 4:26-27) • "Once a hash block is received and verified, data packets that depend on it may be verified (step 340) ***by computing the hash of the packet and comparing it with the hash value stored in the hash block*** (step 350). If this check fails, the data packet is corrupt and should be repaired or recovered (step 350)." (*Id.*, 10:1-4) • "The data packets are checked in the same manner. For example, ***the data packet A_{10}, A_{11}, A_{12} would be checked by comparing the hash of the data packet 500d with the value B_4***. If this check fails, the data packet A_{10}, A_{11}, A_{12}, is corrupt." (*Id.*, 11:24-27) • *See also, e.g.,* Ex. 1006, Abstract, 3:6-17, 4:15, 8:9-11, 9:1-4, 11:27-12:2, Figs. 2-5; Ex. 1002, ¶53.

Claim 42	**Hanna in view of Stefik**
[42e] granting the request to use the file in the predefined manner.	**Hanna** discloses that if the "check" (*i.e.,* verifying the data packets by computing the hash of the packet and comparing it with the hash value stored in the hash block) fails, then the data packet is corrupt and should be repaired, recovered, or discarded. (Ex. 1006, 10:1-4, Fig. 3). A POSITA would have recognized that corrupted data, or data not properly signed with a recognized/authenticated signature, would be undesirable to be used. As such, it would have been obvious to allow or prevent a requested use of a file depending on whether it passes the authentication process. (Ex. 1002, ¶¶111-112). To the extent it is argued that granting a request to use a file in in a predefined manner is not already disclosed in Hanna, **Stefik** discloses granting the request to use the file in the predefined manner. (Ex. 1002, ¶113). Stefik discloses performing transactions to "service the [user] request." (Ex. 1007, 16:61-67). Stefik further discloses granting a request by one repository to obtain a copy of software stored in another repository for a stated purpose corresponding to a specific usage right, including installation, or granting a request to install software after authentication. (Ex. 1007, 7:19-33). For example, if the software to be installed within a rendering repository is stored in a second repository, the second repository grants the request of the rendering repository for a copy of the software to install. (Ex. 1007, 20:48-54, 42:26-31). As another example, after the copied software has been verified using the tamper-checking code, the request to install the software is granted (*e.g.,* installation process continues in accord with instructions). If the software fails verification (*i.e.,* the two codes do not match), then the installation transaction ends with an error. (Ex. 1007. 42:49-43:2; Ex. 1002, ¶¶114-115). • "In any event, ***Repository I checks the usage rights associated with the digital work to determine if the access to the digital work may be granted***, step 105. The check of the usage rights essentially involves a

Claim 42	Hanna in view of Stefik
	determination of whether a right associated with the access request has been attached to the digital work and if all conditions associated with the right are satisfied. If the access is denied, repository 1 terminates the session with an error message, step 106. ***If access is granted, repository 1 transmits the digital work to repository 2***, step 107." (Ex. 1007, 7:23-33) • "At a minimum, the user interface must permit a user to input information such as access requests and alpha numeric data and provide feedback as to transaction status. ***The user interface will then cause the repository to initiate the suitable transactions to service the request.*** Other facets of a particular user interface will depend on the functionality that a repository will provide." (*Id.*, 16:61-67). • "If the check-code does not match the tamper-checking code from the certificate, the installation transaction ends with an error." (*Id.*, 42:51-53) • "***The requester retrieves the instructions in the installation script and follows them.*** If there is an error in this process (such as insufficient resources), then the transaction ends with an error." (*Id.*, 42:61-64) • "determining if ***a request for access*** to a digital work stored in said storage means ***may be granted***" (*Id.*, 52:35-37) • "if said particular usage right is one of said usage rights associated with said digital work, ***granting access to said digital work and performing usage transaction steps associated with said particular usage right***." (*Id.*, 56:6-9) • *See also, e.g.,* Ex. 1007, 6:36-56, 16:61-67, 42:26-31, 51:1-5, 51:64-67, Fig. 1; Ex. 1002, ¶¶58, 60.

Claim 43	**Hanna in view of Stefik**
[43pre] A method as in claim 42, in which the at least one check value comprises one or more hash values, and in which verifying the authenticity of at least a portion of the file using the at least one check value includes:	**Hanna** discloses a method as in claim 42, in which the at least one check value (*e.g.,* "hash block") comprises one or more hash values (*e.g.,* "hash values"), and in which verifying the authenticity of at least a portion of the file (*e.g.,* a "data packet" of a data set) using the at least one check value (*e.g.,* the "hash block" containing the "hash values"). (Ex. 1002, ¶¶116-117). • "The hierarchical structure used in the method cryptographically protects ***individual portions of the data*** and makes it easier to recognize corruption." (Ex. 1006, 3:10-11) • "Finally, the data is checked against ***the hash block containing the hashes of the data set***." (*Id.*, 4:26-27) • "Once a hash block is received and verified, data packets that depend on it may be verified (step 340) ***by computing the hash of the packet and comparing it with the hash value stored in the hash block*** (step 350). If this check fails, the data packet is corrupt and should be repaired or recovered (step 350)." (*Id.*, 10:1-4) • "The data packets are checked in the same manner. For example, ***the data packet A_{10}, A_{11}, A_{12} would be checked by comparing the hash of the data packet 500d with the value B_4***. If this check fails, the data packet A_{10}, A_{11}, A_{12}, is corrupt." (*Id.*, 11:24-27) • *See also, e.g.,* Ex. 1006, Abstract, 3:6-17, 4:15, 8:9-11, 9:1-4, 11:27-12:2, Figs. 2-5.
[43a] hashing at least a portion of the file to obtain a first hash value	**Hanna** discloses hashing (*e.g.,* "[a] hash function") at least a portion of the file (*e.g.,* a "data packet" of a data set) to obtain a first hash value. (Ex. 1002, ¶¶118-119). • "The hierarchical structure used in the method cryptographically protects ***individual portions of the data*** and makes it easier to recognize corruption." (Ex. 1006, 3:10-11) • "Once a hash block is received and verified, data packets that depend on it may be verified (step 340) ***by***

Claim 43	Hanna in view of Stefik
	computing the hash of the packet and comparing it with the hash value stored in the hash block (step 350)." (*Id.*, 10:1-3) • "***A hash function (not shown) is applied to each packet*** and compared with the corresponding hash value in the hash block in the level above it." (*Id.*, 11:14-15) • "The data packets are checked in the same manner. For example, the data packet A_{10}, A_{11}, A_{12} would be checked by comparing ***the hash of the data packet 500d*** with the value B_4." (*Id.*, 11:24-25) • *See also, e.g.,* Ex. 1006, Abstract, 3:6-17, 4:15, 8:9-11, 11:12-19, 11:27-12:2, Figs. 2-5.
[43b] comparing the first hash value to at least one of the one or more hash values.	**Hanna** discloses comparing the first hash value (*e.g.,* "the hash of the data packet") to at least one of the one or more hash values (*e.g.,* "the hash value stored in the hash block"). (Ex. 1002, ¶¶120-121). • "Finally, the data is checked against the hash block containing the hashes of the data set." (Ex. 1006, 4:26-27) • "Once a hash block is received and verified, data packets that depend on it may be verified (step 340) by computing the hash of the packet and ***comparing it with the hash value stored in the hash block*** (step 350). If this check fails, the data packet is corrupt and should be repaired or recovered (step 350)." (Ex. 1006, 10:1-4) • "The data packets are checked in the same manner. For example, ***the data packet A_{10}, A_{11}, A_{12} would be checked by comparing the hash of the data packet 500d with the value B_4***. If this check fails, the data packet A_{10}, A_{11}, A_{12}, is corrupt." (*Id.*, 11:24-27) • *See also, e.g.,* Ex. 1006, Abstract, 3:6-17, 4:15, 4:26-27, 8:9-11, 11:12-19, 11:27-12:2, Figs. 2-5.

B. Ground 2: Claims 42 and 43 are rendered obvious by Gennaro

1. Overview of Gennaro

In 1997, Rosario Gennaro and Pankaj Rohatgi disclosed their invention relating to signing and authenticating a stream of digital data in U.S. Patent No. 6,009,176 ("Gennaro"; Ex. 1008), entitled "How to Sign Digital Streams," which was filed on February 13, 1997. (Ex. 1002, ¶124).

Gennaro describes a technique for efficiently signing a digital audio, video, or data stream so that a receiver of the stream can authenticate and consume the stream at the same rate which the stream is being sent. (Ex. 1008, Abstract, 12:44-51). The digital stream can be a finite stream that is known in its entirety to a sender of the stream, such as a digital movie; it can also be a "potentially infinite" stream that is not known in advance, such as a live feed. (Ex. 1008, 2:64-3:10; Ex. 1002, ¶125).

For the finite streams, Gennaro describes a prior art solution that requires retrieving both a digital signature and a hash of the data, just as recited in claim 42 of the '815 patent. In this solution, when a sender prepares a data stream for transmission to a receiver, several steps are performed: first, the sender splits the data stream into blocks; second, the sender hashes each block and puts the hashes into a table; third, the sender signs the table. When the receiver asks for the data stream, the sender would first send the signed table of hashes, followed by the data

stream. The receiver verifies the signature, which in turn verifies that the hashes in the signed table are authentic. Each block is then verified using the corresponding hash in the signed table. (Ex. 1008, 1:23-34; Ex. 1002, ¶126).

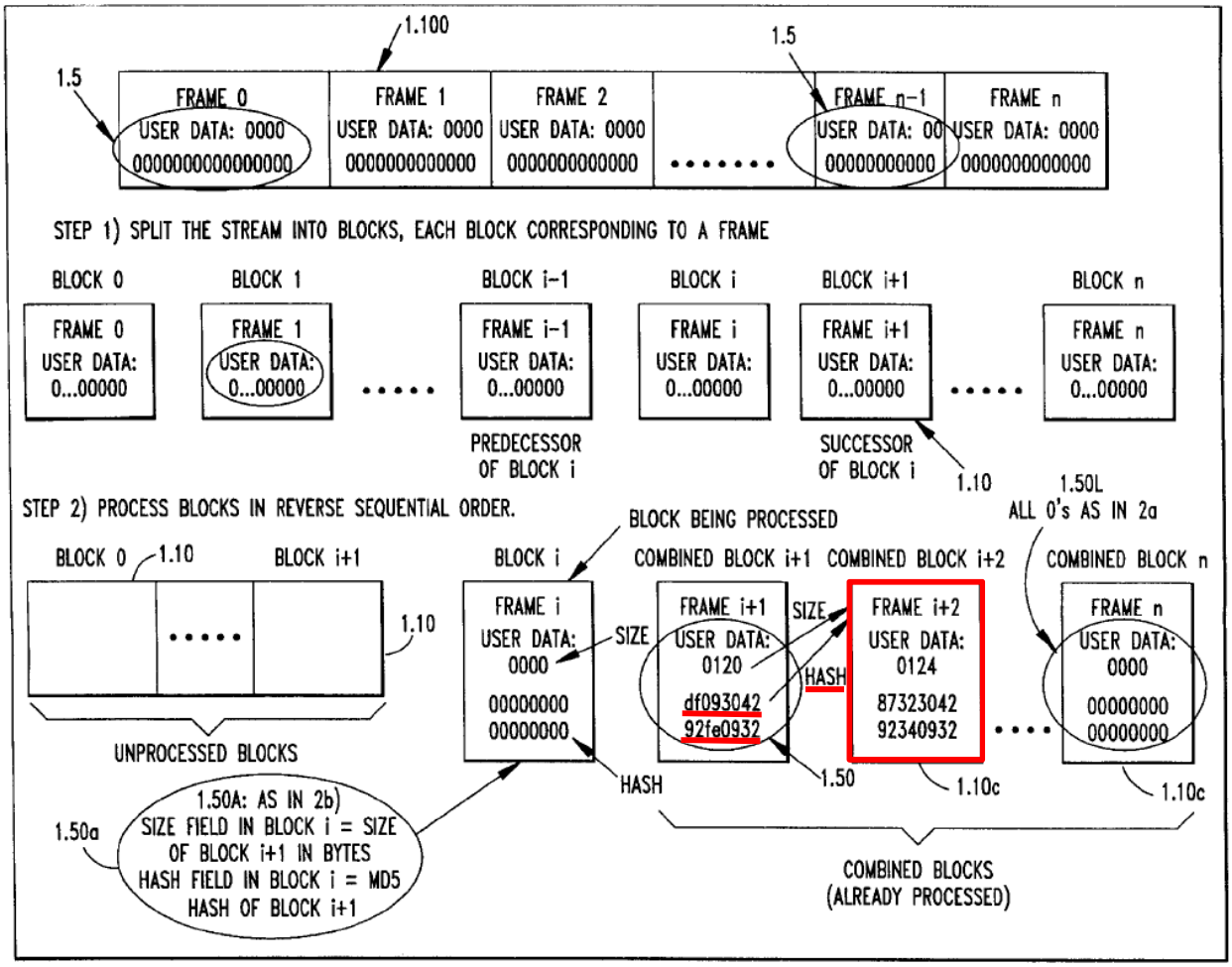
Gennaro presents an alternative way of signing and authenticating a data stream that would allow the receiver to consume the authenticated data from the stream at the same rate it receives the data from the stream. (Ex. 1008, 2:29-41). It explains:

> The basic idea of the present solution is to divide the stream into blocks and embed some authentication information in the stream itself. The authentication information placed in block i will be used to authenticate the following block i+1. This way the signer needs to sign just the first block and then the properties of this single signature will "propagate" to the rest of the stream through the authentication information.

(Ex. 1008, 2:55-63; Ex. 1002, ¶127).

The color-annotated Figures 1A and 1B in Gennaro illustrates the steps a sender would take to prepare a digital stream for transmission to a receiver.

FIG. 1A



In step 1, a digital stream (1.100) is split into blocks (block 0 to block n). (Ex. 1008, 4:14-23). In step 2, the blocks are processed in reverse order (from block n to block 0) such that the hash (*e.g.,* "df09304292fe0932") of a successor combined block (*e.g.,* combined block i+2) is embedded in its preceding block (*e.g.,* block i+1) to create a combined block (*e.g.,* combined block i+1); the stream of combined blocks is referred to as (1.100c). (Ex. 1008, 4:24-48; Ex. 1002, ¶128).

Figure 1B illustrates the last step of the process before the digital stream is transmitted to the receiver:

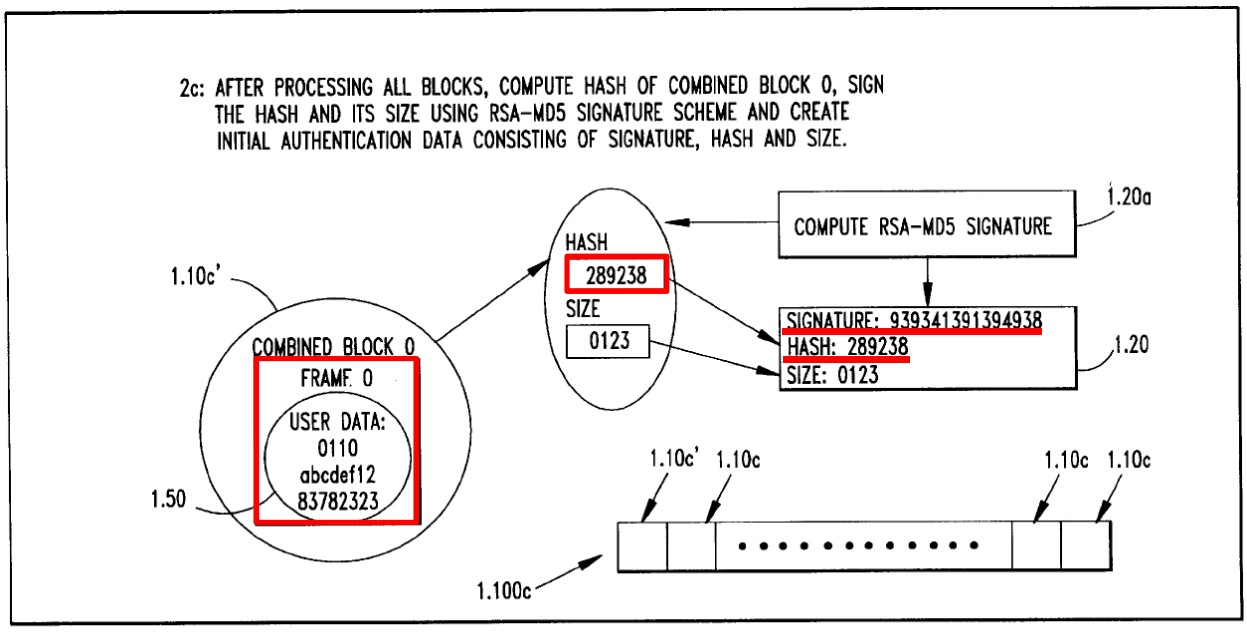
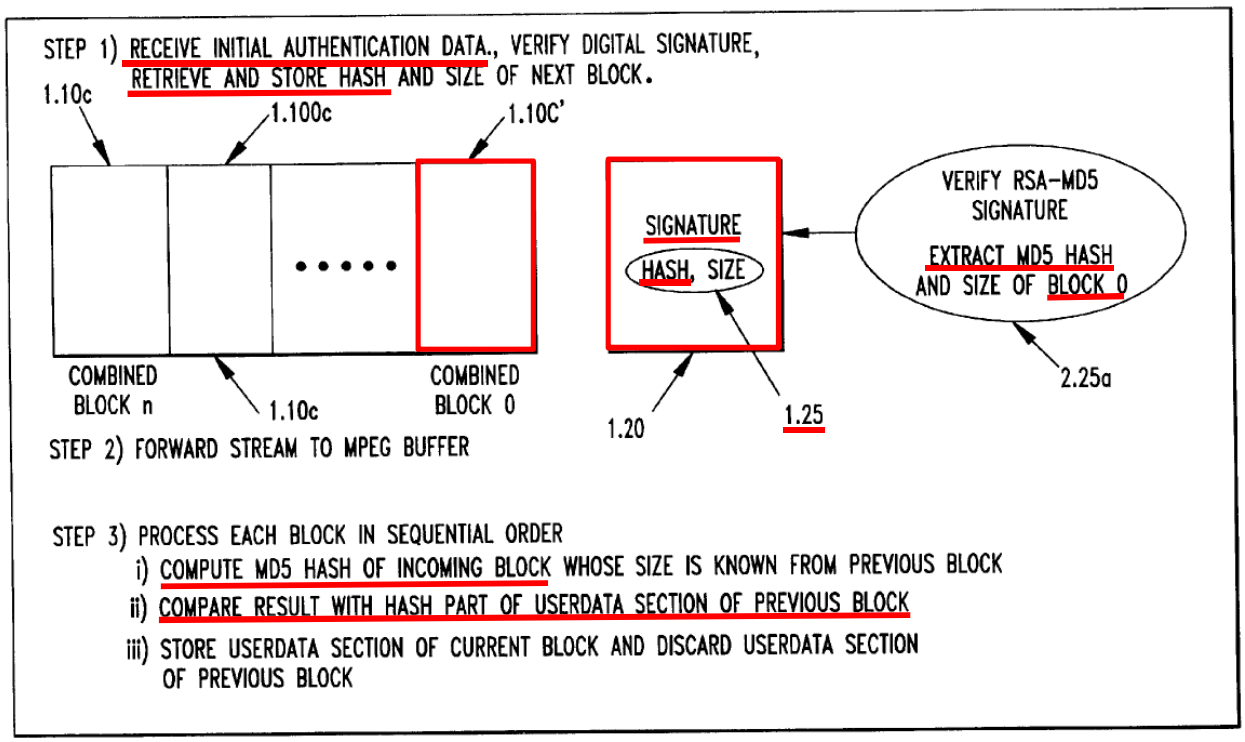


FIG.1B

The first combined block, depicted as the "Combined Block 0" in Figure 1B, is also referred to as the "combined distinguished block (1.10c')" in the specification. (Ex. 1008, 4:52-62). A hash of the first combined block is computed. This hash is referred to as "hash 1.25" (*Id.*, 4:59-61), and is depicted as having the value of "289238" in the example illustrated in Figure 1B. The hash 1.25 of the first combined block is then signed to create a digital signature. The digital signature and hash 1.25, along with other data, are combined to form the "initial authentication data (1.20)." (*Id.*, 4:59-65; Ex. 1002, ¶129).

When the receiver asks to play or consume the digital stream (Ex. 1008, *Abstract*, 1:28-29, 2:50-54), the sender sends the initial authentication data (1.20) first, and then the combined stream (1.100c). (*Id.*, 4:67-5:2). The color-annotated Figure 2A below illustrates how the combined stream is authenticated using the initial authentication data:

FIG.2A



In step 1, the receiver receives the initial authentication data, which includes the digital signature and hash 1.25 (that is the hash of the first combined block (1.10c')). (*Id.*, 4:63-5:2, 5:23-28). The authentication software at the receiver does the following: it first verifies the signature; if the signature is successfully verified, then the authenticity of hash 1.25 is verified. (*Id.*, 5:30-34). And if the

hash 1.25 is verified by the signature to be authentic, then it is extracted to verify the first combined block (1.10c'). (*Id.*, 5:30-34; Ex. 1002, ¶130).

In step 2, the receiver receives the combined stream (1.100c). The MPEG software at the receiver is prevented at this time from consuming the data in the stream because it has not been authenticated. (Ex. 1008, 5:35-39). In step 3, the authentication software splits the combined stream (1.100c) into blocks. As soon as a block is received, a hash of the block is computed. (*Id.*, 5:48-53). The hash of the first combined block is then compared against the hash 1.25 that has been verified using the digital signature in the initial authentication data. (*Id.*, 5:54-57). If the hashes are the same, the first combined block is verified; the request to play or consume the block is then granted and the MPEG software is permitted to process the block. (*Id.*, 5:57-62). If the hashes are different, then tampering has been detected and the processing of the first combined block would be halted. (*Id.*, 5:57-62). Subsequent blocks are authenticated in a similar manner, where the hash in a preceding block (*e.g.,* the hash in the first combined block) is used to authenticate its successor block (*e.g.,* the second combined block) by computing a hash on the successor block and comparing the two hash values. (*Id.*, 5:63-65; Ex. 1002, ¶¶131-136).

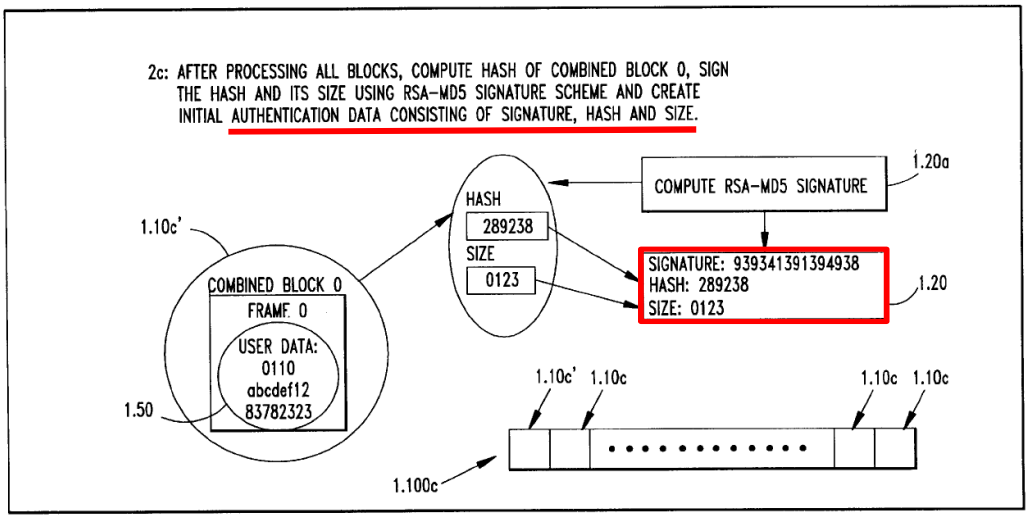
2. Claim Charts: Claims 42 and 43 are rendered obvious by Gennaro

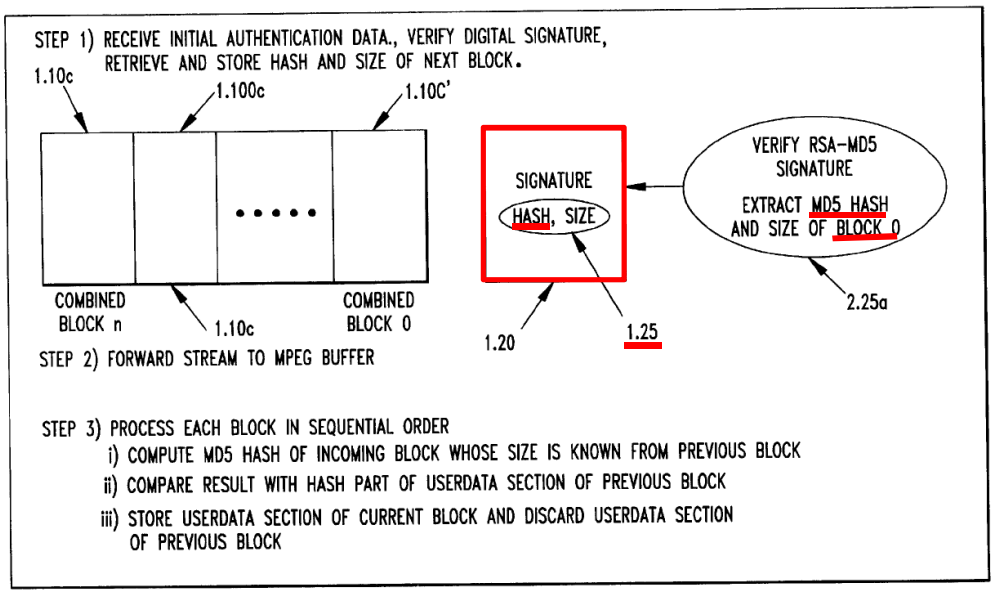
Claim 42	Gennaro
[42pre][4] A method for managing at least one use of a file of electronic data, the method including:	**Gennaro** discloses a method for managing at least one use (e.g., "play," "consume," "convert[] back to video") of a file of electronic data (e.g., "digital stream"). (Ex. 1002, ¶¶137-138). • "A method of signing digital streams so that ***a receiver of the stream can authenticate and consume the stream*** at the same rate which the stream is being sent to the receiver." (Ex. 1008, Abstract) • "Finally we assume that the receiver has processing power or hardware that can compute a small number of fast cryptographic c[h]ecksums faster than the incoming stream rate while still being able to ***play the stream*** in real-time." (*Id.*, 2:50-54) • "In the first scenario the stream is finite and is known in its entirety to the signer in advance. This is not a very limiting requirement since it covers most of ***the internet applications (digital movies, digital sounds, applets).***" (*Id.*, 2:64-67) • "The preferred embodiment for authenticating streams known in advance to the sender has been designed to work for ***MPEG video streams***." (*Id.*, 3:66-4:1) • "The function of this authentication software is to authenticate blocks from the incoming combined stream before ***passing them on*** to the MPEG processing software (2.200) ***to be converted back to video using the MPEG software***." (*Id.*, 5:7-14) • *See also, e.g.,* Ex. 1008, 12:44-51; Ex. 1002, ¶¶49-50.
[42a] receiving a request to use the file	**Gennaro** discloses receiving a request to use the file in a predefined manner (*e.g.,* consumer requests to play "digital movies, digital sounds, applets"; sender receives "request" from the receiver, or the receiver "asks for" the digital stream; the authentication software

[4] To the extent they are limiting, the preambles of all Challenged Claims are nonetheless met by the art of record.

Claim 42	Gennaro
in a predefined manner;	"controls how the MPEG software/hardware is informed of incoming data"). (Ex. 1002, ¶139). Gennaro discloses multiple requests to use a file in a predefined manner. For example, Gennaro discloses a user requesting to use a file in a predefined manner (e.g., playing "internet applications (digital movies, digital sounds, applets)"). (Ex. 1008, 2:64-67; Ex. 1002, ¶140). Gennaro further discloses that, when a user requests to play a file, a receiver requests transmission of the digital stream comprising the video for that purpose. (Ex. 1008, 1:28-29). It would have been obvious to a POSITA that the receiver would make similar requests for all file transmissions, and that such request would be received by a sender of the digital stream. (Ex. 1002, ¶141). Gennaro further discloses passing data in the file from the sender to an "authentication software" at the receiver. The authentication software controls the data flow to a "MPEG processing software" at the receiver. The authentication software makes sure only blocks that have been authenticated would be passed to the MPEG software for processing. (Ex. 1008, 5:4-12). The authentication software controls when and how the MPEG software/hardware is informed that it has data in its buffer to be processed. (Ex. 1008, 5:15-20, 5:35-39). After the authentication software has authenticated the data, such data would be passed to the MPEG software for processing to convert the data into video for playback. (Ex. 1008, 5:40-67). In view of the role of the authentication software in the receiver as a control for the MPEG software/hardware, it would have been obvious to a POSITA to configure the receiver so that, when it passes the data to the authentication software, it is in the context of a request for the MPEG software to convert the data into video for playback. (Ex. 1008, 5:8-12; Ex. 1002, ¶142). • "When ***the receiver asks for the authenticated stream***, the sender first sends the signed table followed by the stream." (Ex. 1008, 1:28-29)

Claim 42	Gennaro
	• "The invention requires additional authentication software to be added to any existing MPEG software (2.200) or hardware both of which currently do not do any authentication. ***The function of this authentication software is to authenticate blocks from the incoming combined stream before passing them on to the MPEG processing software (2.200) to be converted back to video using the MPEG standard***." (Ex. 1008, 5:4-14) • "The authentication software shares with the MPEG processing hardware/software 2.200, a bit buffer which is at least 1.8 M-bits in size. In addition ***this authentication software now controls how the MPEG processing software/hardware is informed of incoming data***." (*Id.*, 5:15-20). • "The receiver then receives the combined stream which is forwarded into the MPEG bit-buffer. However ***the MPEG software is not informed of incoming data before it is authenticated.*** This prevents the MPEG software to consuming unauthenticated data." (*Id.*, 5:35-39) • *See also, e.g.,* Ex. 1008, 14:3-15; Ex. 1002, ¶¶58-59.
[42b] retrieving at least one digital signature and at least one check value associated with the file;	**Gennaro** discloses retrieving at least one digital signature (*e.g.,* "digital signature") and at least one check value (*e.g.,* "hash 1.25," which is the "hash of the first combined block (1.10c')") associated with the file (*e.g.,* "digital stream"). (Ex. 1002, ¶143). Gennaro explains that the verification begins with receiving the "initial authentication data (1.20)," which includes a "digital signature," and a "hash 1.25," which is the hash of the "first of the combined blocks" in the "digital stream." (Ex. 1002, ¶¶144-145). • "The ***hash 1.25*** and the size of ***the combined distinguished block, which is the first of the combined blocks***, in the stream (1.10c') is calculated, and the ***said hash value*** together with the size of said block ***is signed*** using a RSA-MD5 signature scheme (1.20a). The ***signature, the hash*** and the size ***are combined to form initial authentication data (1.20)***. The combined blocks (1.10c) in sequence now constitute the

Claim 42	Gennaro
	combined stream (1.100c). ***The initial authentication data (1.20) will be sent to the receiver before the combined stream (1.100c) is sent.***" (Ex. 1008, 4:59-5:2) • "Before receiving the combined stream (1.100c) ***the receiver receives the initial authentication data (1.20) consisting of a digital signature*** on the hash of the first combined block (1.10c') and the size of first combined block, ***together with the purported values of the hash*** and size." (Ex. 1008, 5:23-29) The content of "initial authentication data (1.20)" is shown in Figure 1B of Gennaro. Figure 1B shows "Initial Authentication Data Consisting of Signature, Hash, and Size."  FIG.1B FIG.1 FIG.1A FIG.1B (color-annotated Fig. 1B of Gennaro)**;** (Ex. 1002, ¶146). Figure 2A of Gennaro shows that the hash value in the initial authentication data (1.20) is referred to as "hash 1.25"

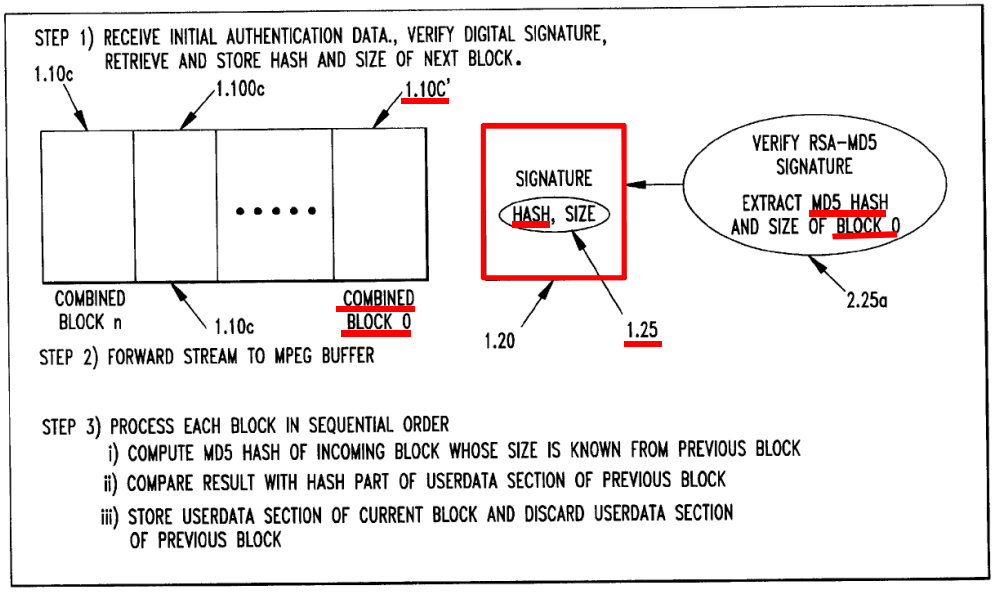
Claim 42	Gennaro
	FIG.2A  (color-annotated Fig. 2A of Gennaro); (Ex. 1002, ¶147). In the prior art embodiment, Gennaro describes retrieving a signature and a table of hashes from the sender. (Ex. 1002, ¶148). • "Instead of signing each block, the sender creates a table listing cryptographic hashes of each of the blocks. Then the sender signs this table. When the receiver asks for the authenticated stream, the sender first sends the signed table followed by the stream." (Ex. 1008, 1:24-29). • *See also, e.g.,* Ex. 1008, 4:34-48. *See also* Ex. 1002, ¶¶51-57.
[42c] verifying the authenticity of the at least one check value using the digital signature;	**Gennaro** discloses verifying the authenticity of the at least one check value (*e.g.,* "hash 1.25," which is "hash of the first combined block (1.10c')") using the digital signature (*e.g.,* "digital signature"). (Ex. 1002, ¶149). • "The hash 1.25 and the size of the combined distinguished block, which is the first of the combined blocks, in the stream (1.10c') is calculated, and the said ***hash value*** together with the size of said block is ***signed using a RSA-MD5 signature scheme*** (1.20a). (Ex. 1008, 4:59-63)"

Claim 42	Gennaro

- "Before receiving the combined stream (1.100c) the receiver receives the initial authentication data (1.20) consisting of ***a digital signature on the hash of the first combined block (1.10c')*** and the size of first combined block, together with the purported values of the hash and size. The authentication software at this stage does the following (2.25a): It verifies the signature. ***If the provided signature verifies, then the purported hash*** and size values ***of the first combined block (1.10c') are authentic***, and **the hash** and size values ***1.25*** are extracted out for use in authenticating the combined stream (1.100c)." (Ex. 1008, 5:23-34)

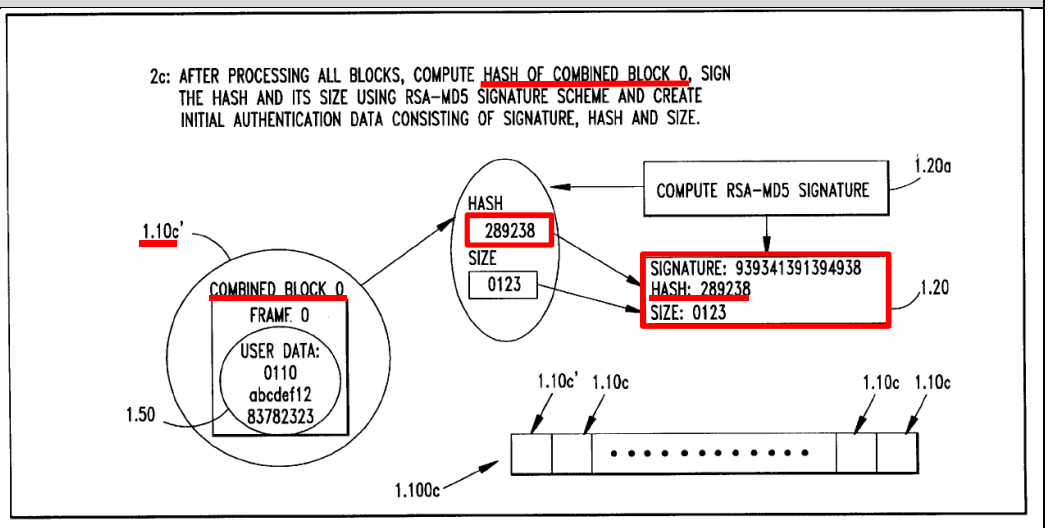
Figure 2A of Gennaro shows that the signature is used to verify "hash 1.25," which is the hash of combined block 0 (1.10c').

FIG.2A



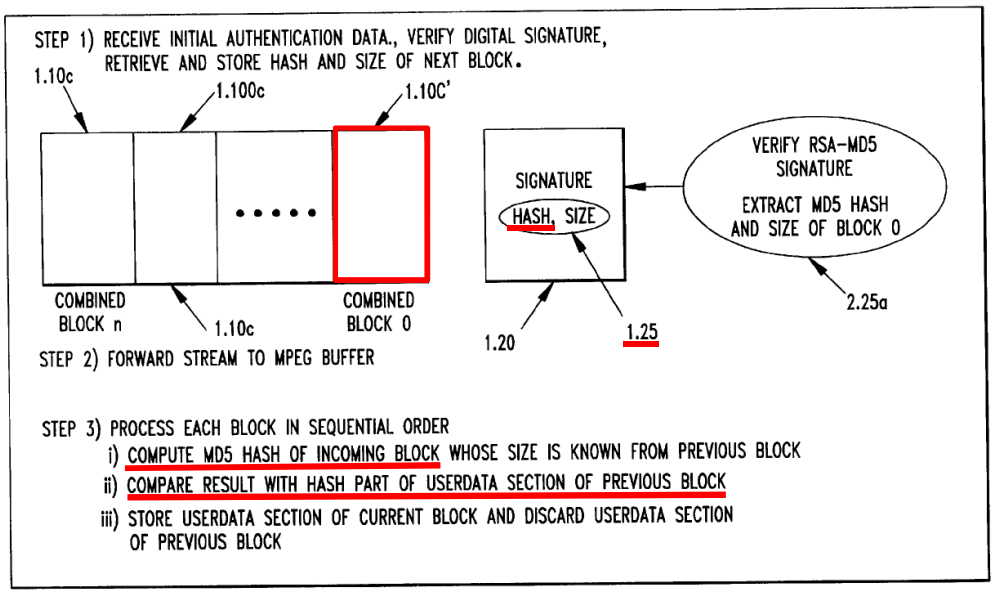
(color-annotated Fig. 2A of Gennaro); (Ex. 1002, ¶150).

Figure 1B of Gennaro shows the relationship between the combined block 0 (1.10c'), hash 1.25, and the signature. The hash 1.25, is the hash of the first combined block (*i.e.,* combined block 0 (1.10c')), which is depicted as having the value "289238," in Figure 2A. The hash 1.25 is signed to generate the signature. Both the signature and the hash 1.25 are part of the initial authentication data (1.20) that are sent to the receiver. Because the signature was created by signing the hash 1.25, the hash 1.25 could be verified using the signature.

Claim 42	Gennaro
	 FIG.1B FIG.1 FIG.1A FIG.1B (color-annotated Fig. 1B of Gennaro); (Ex. 1002, ¶151). In the prior art embodiment, Gennaro describes that the hashes in the signed table are verified once the signature is verified. (Ex. 1002, ¶152). • "The receiver first receives and stores this table and verifies the signature on it. If the signature matches then the receiver has the cryptographic hash of each of the following stream blocks." (Ex. 1008, 1:29-33). • *See also, e.g.,* Ex. 1008, 3:5-7. *See also* Ex. 1002, ¶54.
[42d] verifying the authenticity of at least a portion of the file using the at least one check value; and	**Gennaro** discloses verifying the authenticity of at least a portion of the file (*e.g.,* "the first combined block" in the digital stream) using the at least one check value (*e.g.,* "hash 1.25," which is the "hash of the first combined block (1.10c')"). (Ex. 1002, ¶153). Figure 2A of Gennaro illustrates that hash 1.25 from the initial authentication data is used to verify the first combined block (1.10c') in the digital stream. This is done by computing a hash on the block, and comparing it against hash 1.25. If they are the same, then the block is verified.

Claim 42	Gennaro

FIG.2A



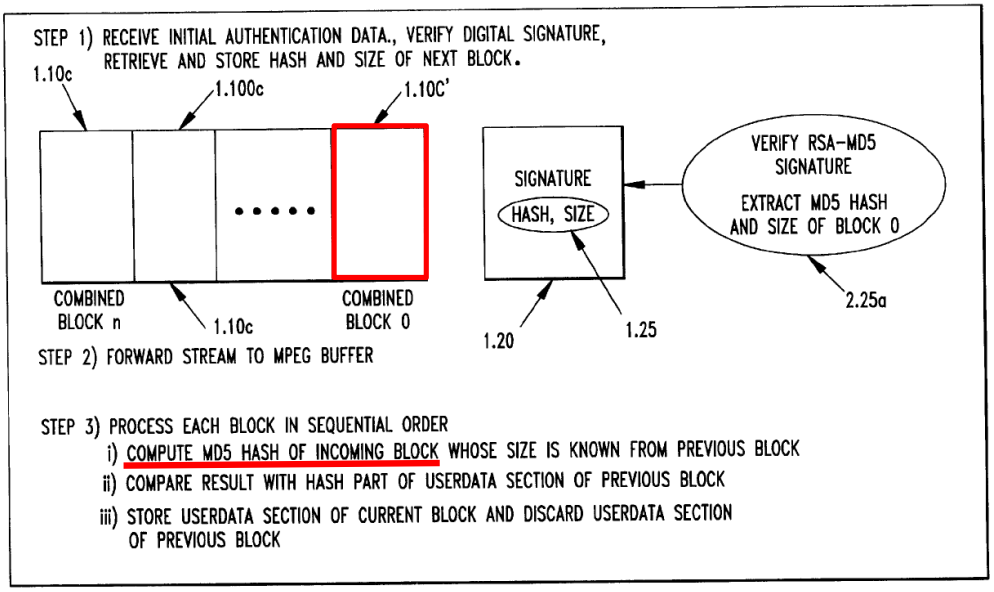
(color-annotated Fig. 2A of Gennaro); (Ex. 1002, ¶154).

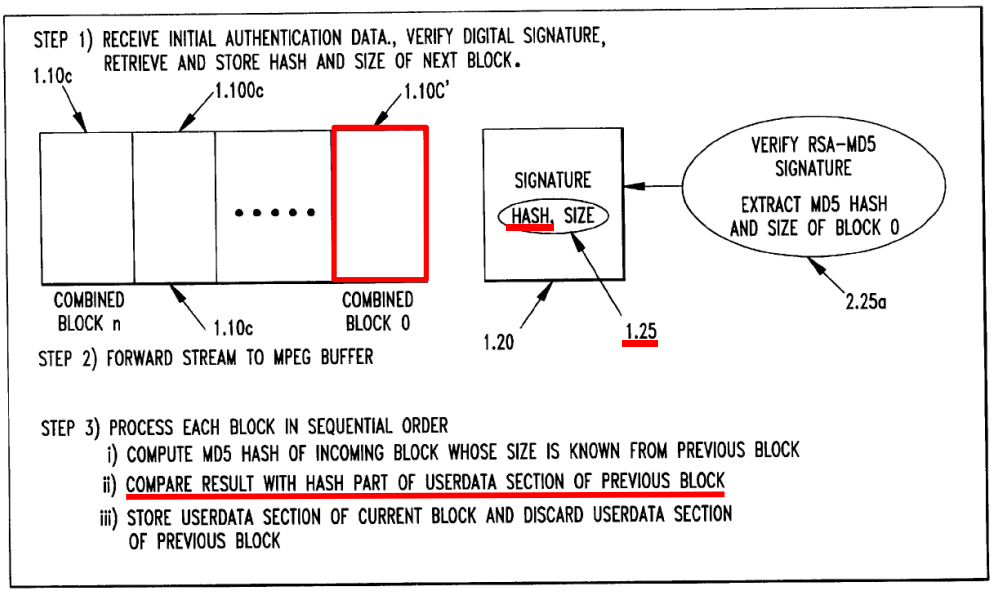
- "The next sixteen bytes of the 20 byte ancillary information placeholder is updated to hold the MD5 cryptographic hash of the successor block. ***This information can be used to authenticate the [] successor block*** which already has its ancillary information in place." (Ex. 1008, 4:44-48).
- "If the provided signature verifies, then ***the purported hash*** and size values ***of the first combined block (1.10c') are authentic***, and **the hash** and size values ***1.25 are extracted out for use in authenticating the combined stream (1.100c)***." (*Id.*, 5:30-34)
- "Compute the MD5 hash of the incoming bytes as they are transferred into the buffer of the receiver. ***As soon as a block comes in, its MD5 hash gets computed***, and computation of the MD5 hash of the next incoming block is started (although its length is not immediately known)." (*Id.*, 5:48-53)
- "***The MD5 hash of the recently arrived block is compared against what it should be according to the authentication chain emanating from the Digital Signature from (1.20)***." (*Id.*, 5:54-57)

Claim 42	Gennaro
	In the prior art embodiment, Gennaro describes verifying the blocks after the hashes have been verified. (Ex. 1002, ¶155). • "The receiver first receives and stores this table and verifies the signature on it. If the signature matches then the receiver has the cryptographic hash of each of the following stream blocks. Thus each block can be verified when it arrives." (Ex. 1008, 1:29-34). • *See also, e.g.,* Ex. 1008, 2:67-3:4, 4:51-55; *See also* Ex. 1002, ¶53.
[42e] granting the request to use the file in the predefined manner.	**Gennaro** discloses granting the request to use the file in the predefined manner (*e.g.,* "play the stream," "consume authenticated data," "it can proceed to process the incoming original block that was authenticated"). (Ex. 1002, ¶156). Gennaro discloses multiple ways in which a request to use the file is granted. For example, Gennaro discloses that the receiver plays a stream, thus granting the user's request to play an internet application. (Ex. 1008, 2:54, 2:65-67; Ex. 1002, ¶157). Gennaro further discloses that the receiver receives the stream from the sender in response to requests to use the data. (Ex. 1008, 1:28-29; Ex. 1002, ¶158). Gennaro further discloses that the authentication software grants a request to process received data from an electronic file when it passes authenticated data to the MPEG software/hardware. (Ex. 1008, 5:54-63). The MPEG software is prevented from consuming the data/block from the digital stream until it is authenticated. (Ex. 1008, 5:35-39). Once the data/block is authenticated by the authentication software, then the authentication software grants the request to use the file and passes it to the MPEG software for processing. (Ex. 1008, 5:8-12, 5:58-62). If the data/block is determined to be tampered, then the request is denied and the processing is halted. (Ex. 1008, 5:57-58; Ex. 1002, ¶159).

Claim 42	Gennaro
	• "As a consequence, ***the receiver can consume authenticated data from the stream*** at the same rate he receives such data from the stream." (Ex. 1008, 2:39-42) • "The receiver then receives the combined stream which is forwarded into the MPEG bit-buffer. However the MPEG software is not informed of incoming data before it is authenticated. ***This prevents the MPEG software to consuming unauthenticated data***." (*Id.*, 5:35-39) • "The MD5 hash of the recently arrived block is compared against what it should be according to the authentication chain emanating from the Digital Signature from (1.20). ***If it is different, then tampering has been detected and processing halted.*** Otherwise, the MPEG software/hardware is informed that a block of bits of a certain size representing a frame has arrived and ***it can proceed to process the incoming original block that was authenticated.***" (*Id.*, 5:54-63) *See also,* Ex. 1002, ¶¶58, 60.

Claim 43	Gennaro
[43pre] A method as in claim 42, in which the at least one check value comprises one or more hash values, and in which verifying the authenticity of at least a portion of the file using the at least one check	**Gennaro** discloses a method as in claim 42, in which the at least one check value comprises one or more hash values (*e.g.,* "hash 1.25," which is the "hash of the first combined block (1.10c')"), and in which verifying the authenticity of at least a portion of the file (*e.g.,* "the first combined block" in the digital stream) using the at least one check value. (Ex. 1002, ¶¶160-162). • *See* Disclosure in claim limitation [42d]

Claim 43	Gennaro
value includes:	
[43a] hashing at least a portion of the file to obtain a first hash value	**Gennaro** discloses hashing at least a portion of the file to obtain a first hash value (*e.g.*, "compute MD5 hash of incoming block"). (Ex. 1002, ¶163). Figure 2A of Gennaro illustrates that hash 1.25 from the initial authentication data is used to verify the first combined block (1.10c') in the digital stream. This is done by first computing a hash on the block. FIG.2A  (color-annotated Fig. 2A of Gennaro); (Ex. 1002, ¶164). • "***Compute the MD5 hash of the incoming bytes as they are transferred into the buffer of the receiver. As soon as a block comes in, its MD5 hash gets computed***, and computation of the MD5 hash of the next incoming block is started (although its length is not immediately known)." (Ex. 1008, 5:48-53)
[43b] comparing the first hash value to at least one of	**Gennaro** discloses comparing the first hash value (*e.g.*, "hash of the first combined block (1.10c')") to at least one of the one or more hash values (*e.g.*, "hash 1.25," which is the "hash of the first combined block (1.10c')"). (Ex. 1002, ¶165).

Claim 43	Gennaro
the one or more hash values.	Figure 2A of Gennaro illustrates that hash 1.25 from the initial authentication data is used to verify the first combined block (1.10c') in the digital stream. This is done by computing a hash on the block, and comparing it against hash 1.25. If they are the same, then the block is verified. FIG.2A  (color-annotated Fig. 2A of Gennaro); Ex. 1002, ¶166). • "***The MD5 hash of the recently arrived block is compared against what it should be according to the authentication chain emanating from the Digital Signature from (1.20)***." (Ex. 1008, 5:54-57) • *See also, e.g.,* Ex. 1008, 2:67-3:4, 4:44-48, 4:51-55, 5:29-34, 5:48-53.

IX. SECONDARY CONSIDERATIONS

Petitioner is unaware of evidence of secondary considerations relevant to the Challenged Claims at the date of this filing. (Ex. 1002, ¶¶168-169).

X. CONCLUSION

Substantial, new, and noncumulative technical teachings have been presented for each Challenged Claim, which are disclosed and/or rendered obvious for the reasons set forth above. (Ex. 1002, ¶¶170-172). There is a reasonable likelihood that Petitioner will prevail as to each of these Challenged Claims. *Inter partes* review of Claims 42 and 43 of the '815 patent is accordingly requested.

Respectfully submitted,

Dated: March 26, 2020

By: /Scott A. McKeown/
Scott A. McKeown
Registration No. 42,866

Customer Number
28120

Counsel for Petitioner Dolby Laboratories, Inc.

CERTIFICATE OF COMPLIANCE

Pursuant to 37 C.F.R. § 42.24(a) and (d), the undersigned hereby certify that the Petition For *Inter Partes* Review complies with the type-volume limitation of 37 C.F.R. § 42.24(a)(i) because, exclusive of the exempted portions, it contains 12,738 words as counted by the word processing program used to prepare the paper.

Dated: March 26, 2020

By: /Keyna Chow/
Keyna Chow

CERTIFICATE OF SERVICE

The undersigned certifies service pursuant to 37 C.F.R. §§ 42.6(e) and 42.105(b) on the Patent Owner by Fedex of a copy of this Petition for Inter Partes Review and supporting materials at the correspondence address of record for the '815 patent:

FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER LLP
901 NEW YORK AVENUE, NW
WASHINGTON DC 20001-4413

Dated: March 26, 2020

By: /Scott A. McKeown/
Scott A. McKeown
Registration No. 42,866