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

U.S. Patent No. 6,785,815

---

# DECLARATION OF DR. VIJAY K. MADISETTI

# I. INTRODUCTION

## A. Engagement

**1.** I submit this report on behalf of Dolby Laboratories, Inc. in connection with their request for *inter partes* review of U.S. Patent No. 6,785,815 (the "'815 patent")—to review and to provide my opinion on the scope and content of "prior art" predating the application for the '815 patent and regarding the subject matter recited in the claims of the '815 patent. I understand that this Declaration relates to a Petition for the above-captioned *inter partes* review (IPR) of the '815 patent.

**2.** For my efforts in connection with the preparation of this declaration, I have been compensated at my standard hourly consulting rate of $550. My compensation is in no way contingent on the results of these or any other proceedings relating to the above-captioned patent. I have no expectation or promise of additional business with the Petitioner in exchange for the positions explained herein.

**3.** I make this declaration based on personal knowledge.

## B. Background and Qualifications

**4.** A detailed description of my professional qualifications, including a listing of my specialties/expertise and professional activities, is contained in my curriculum vitae, a copy of which is attached as Appendix A. In what follows, I provide a short summary of my professional qualifications.

**5.** I have over thirty years of experience as an electrical and computer engineer in industry, education, and consulting.

**6.** I am a Professor in Electrical and Computer Engineering at Georgia Tech. I have worked extensively in the field of information systems (including information security), digital communications and have studied telecommunications and systems engineering since 1981. I also have over 20 years of industry experience in computer engineering, secure information systems, distributed computer systems, networking, software engineering, signal processing, and telecommunications, including wireless communications and signal processing. Throughout this time, I have designed, implemented, and tested various products in the fields of electronics, computer engineering, and communications.

**7.** In 1984, I received a Bachelor of Technology in Electronics and Electrical Communications Engineering from the Indian Institute of Technology (IIT). In 1989, I received my Ph.D. in Electrical Engineering and Computer Sciences (EECS) from the University of California, Berkeley. That year, I also received the Demetri Angelakos Outstanding Graduate Student Award from the University of California, Berkeley, and the IEEE/ACM Ira M. Kay Memorial Paper Prize.

**8.** In 1989, I also joined the faculty of Georgia Tech. I began working at Georgia Tech as an assistant professor, became an associate professor in 1995, and have held my current position since 1998. As a member of the faculty at Georgia

Tech, I have been active in, among other technologies, cloud computing, distributed computing, image and video processing, computer engineering, information security, embedded systems, chip design, software systems, wireless networks and cellular communications.

**9.** I have been involved in research and technology in the area of computing and digital signal processing since the late 1980s, and I am the Editor-in-Chief the IEEE Press/CRC Press's 3-volume Digital Signal Processing Handbook (Editions 1 & 2) (1998, 2010).

**10.** Over the past three decades, I studied, used, and designed distributed systems, including one of the first published works in distributed secure content delivery networks, that included processing of multimedia signals over the internet, called BEEHIVE, in addition to image and video processing and wireless networking circuits for several applications, including digital and video cameras, mobile phones and networking products for leading commercial firms.

**11.** Between 1994-1998 as part of a research initiative in collaboration with the US Army Research Laboratory and Georgia Tech, I and my students developed one of the first published implementations of a distributed signal processing environment, where secure sensor data (from US Army) was accessed over the network and processed and rendered/displayed at distributed and protected server locations (Georgia Tech, Rice University, and Spelman College) utilizing Java-

based object broker technologies. This secure environment, BEEHIVE, is shown in the figure below, implemented within Java, utilized resource objects (including servers), data sources (such as cameras and sensors), sink sources (such as display), and service objects (representing Java applications), broker objects (e.g., schedulers), and user COE interfaces (e.g., GUIs) that download applications that run locally on secure data obtained from networked sources consistent with information policies and user privileges.

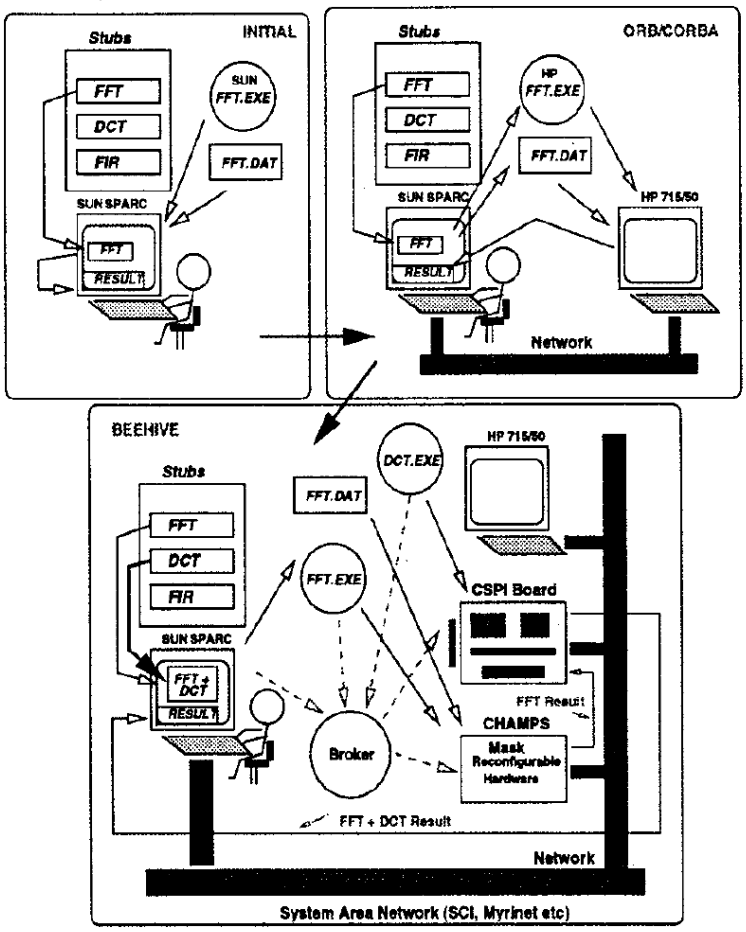



**Figure 1. Evolution of the BEEHIVE Project.**

**12.** A detailed description of the distributed environment is described in the Figure 4 below.

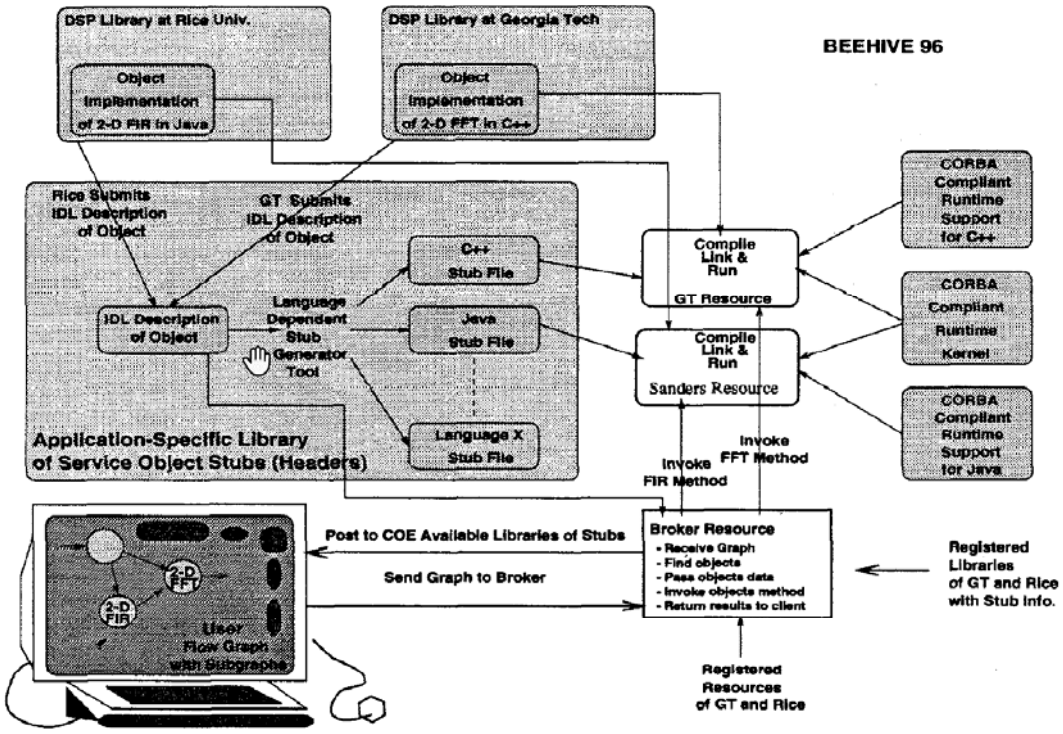



**Figure 4. BEEHIVE Environment v0.1. Next version plans to include seamless support for BEEHIVE-compliant DSP cards and FPGA boards.**

**13.** Prior to or around the timeframe of the filing the ’815 Patent, some of my significant work in the area of digital image processing, video processing, networking technologies, and software engineering include the following:

i. M. Romdhane, V. Madisetti, “All Digital Oversampled Front-End Sensors”, IEEE Signal Processing Letters, Vol 3, Issue 2, 1996;

ii. A. Hezar, V. Madisetti, “Efficient Implementation of Two-Band PR-QMF Filterbanks”, IEEE Signal Processing Letters, Vol 5, Issue 4, 1998; and

iii. R. Tummala, V. Madisetti, System on Chip or System on Package”, IEEE Design & Test of Computers, Vol 16, Issue 2, 1999

**14.** I have also designed code and compilation tools for chipsets used for advanced imaging and document cameras used in photocopying and other applications, such as the Intel MXP 5800 family of image processing chipsets that have been widely used in commercial document imaging and video products since the late 1990s. I have also implemented H.264 and AVC video compression and rendering codecs for a large commercial semiconductor vendor in the mid 2000 timeframe. I have published several papers on audio, video and image content processing in a distributed environment over the two decades.

**15.** I also have significant experience in designing and implementing electronic equipment using various source code languages, including C, assembly code, VHDL, and Verilog. In 2000, I published a book entitled “VHDL: Electronics Systems Design Methodologies.” In 1997, I was awarded the VHDL International Best PhD Dissertation Advisor for my contributions in the area of rapid prototyping.

**16.** Since 1995, I have authored, co-authored, or edited several books in the areas of communications, signal processing, chip design, and software engineering, including VLSI Digital Signal Processors (1995), Quick-Turnaround ASIC Design in VHDL (1996), The Digital Signal Processing Handbook (1997 & 2010), Cloud Computing: A Hands-On Approach (2013), Internet of Things: A Hands-On Approach (2014), Big Data Science & Analytics (2016).

**17.** I have authored over 100 articles, reports, and other publications pertaining to electrical engineering, and in the areas of computer engineering, communications signal processing, and communications. All of my publications, including the ones identified above, are set forth in my attached CV.

**18.** I have implemented several electronic devices, including audio and video codecs, echo cancellers, equalizers, and multimedia audio/video compression applications and modules for a leading commercial mobile phone manufacturer. I have also designed tools for porting mobile applications from one platform to another for a leading mobile phone manufacturer. I have also developed hardware and software designs for a leading set-top box manufacturer. I am familiar with the design, test and implementation of embedded systems, such as image and video processing systems, security systems, barrier control systems, and associated software and hardware architectures.

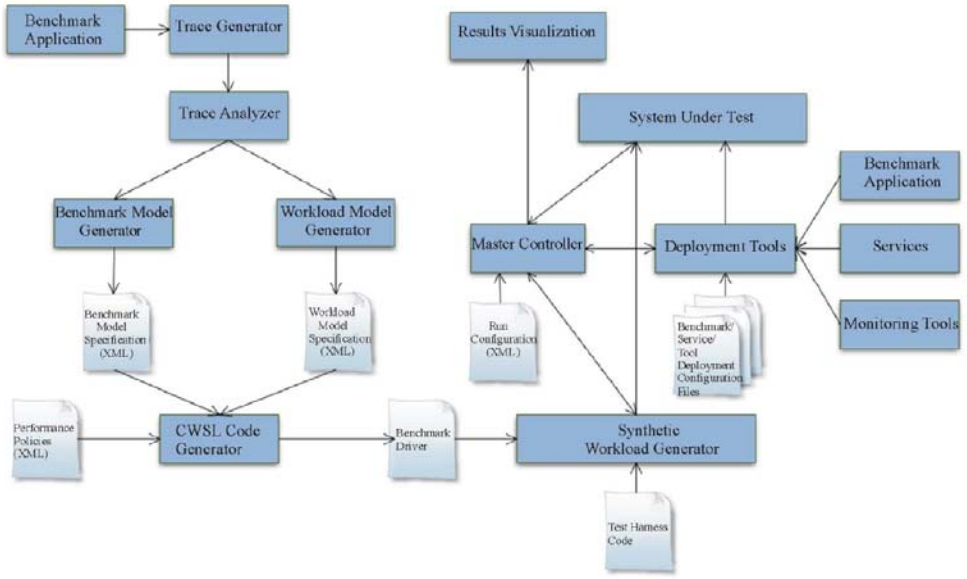
**19.** I have been elected a Fellow of the Institute of Electrical and Electronics Engineers ("IEEE") in recognition of my contributions to embedded computing systems. The IEEE is a worldwide professional body consisting of more than 300,000 electrical and electronic engineers. Fellow is the highest grade of membership of the IEEE, with only one-tenth of one percent of the IEEE membership being elected to the Fellow grade each year.

**20.** In 2006, I was awarded the Frederick Emmons Terman Medal from the American Society of Engineering Education (ASEE) and HP Corporation for my contribution to electrical engineering while under the age of 45.

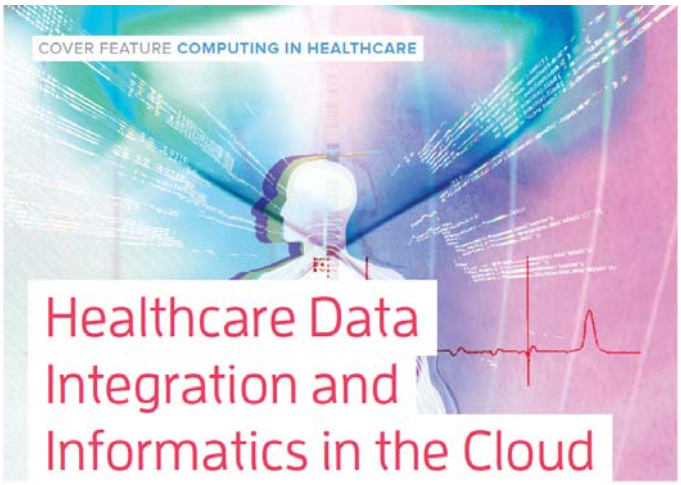
**21.** I have been working in the areas of secure cloud computing, workload modeling, virtualization, and benchmarking for over ten years.

**22.** In the area of characterization, modeling and generation of workloads for cloud computing applications I have created a synthetic workload generator for virtual environments that accepts benchmark and workload model specifications. I also developed a cloud workload specification language called, GT-CWSL, to provide a structured way for specification of application workloads. These approaches allow accurate characterization of virtualization and cloud performance, and have been published in "Synthetic Workload Generation for Cloud Computing Applications", Journal of Software Engineering and Applications, 2011, Vol 4, pp. 396-410.

**23.** Our methodology for analysis of virtual workloads on cloud platforms is shown in the figure below.




**24.** In other work related to distributed computing, virtualization, performance modeling, and data integration, I have developed cloud-based information integration and informatics framework, called CHISTAR, that allows integration of secure distributed and heterogeneous healthcare data into a common virtual environment (on commodity hardware running hypervisors)allow easier analysis and analytics to be performed. This research was presented as a cover feature in the Special Issue on Computing in Healthcare, IEEE Computer Magazine, in 2015.




**25.** In my work on rapid prototyping of cloud-based virtual services and systems, I proposed a new methodology using loosely coupled virtual components that are not restricted by architecture or programming styles. This approach, shown below, creates virtual components that are then integrated and deployed effectively to realize improvements in deployment efficiency and time to deployment, as shown below.

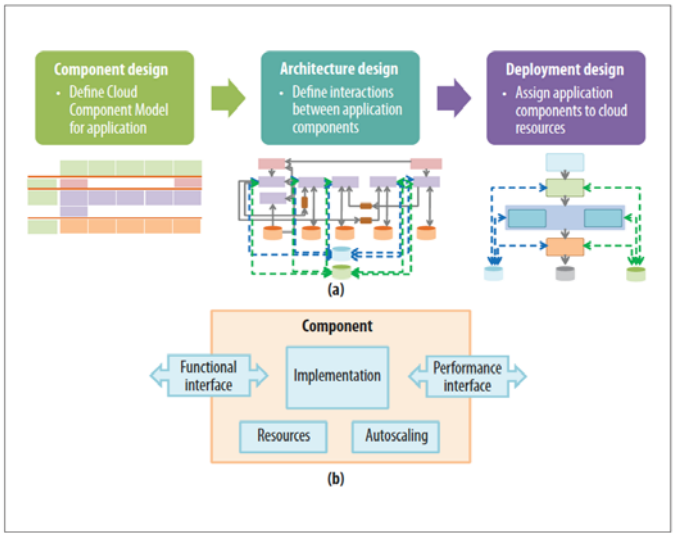



**Figure 2.** Cloud Component Model (CCM)-based development: (a) design methodology, which comprises component, architecture, and deployment design; and (b) CCM component architecture. CCM's component-based approach is suitable for both Web-based and mobile applications.

**26.** Use of our CCM-based virtualization and distributed cloud component methodology improved throughput and response times as shown below. These results were published in flagship journal, IEEE Computer Magazine, in November 2013.

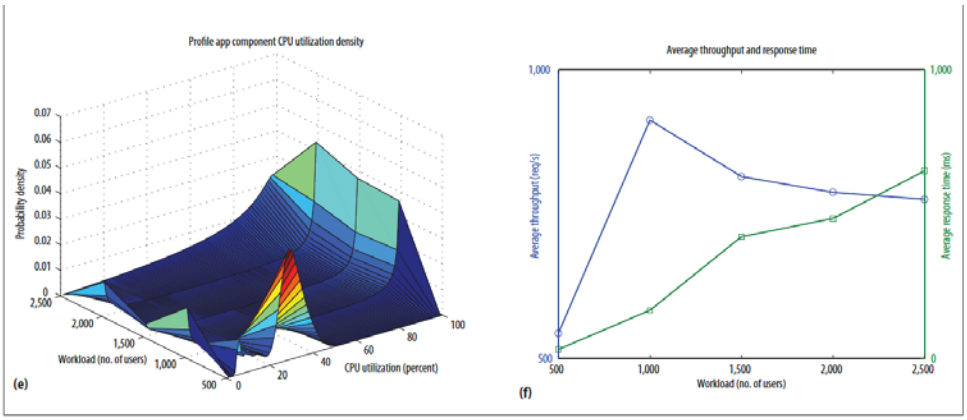
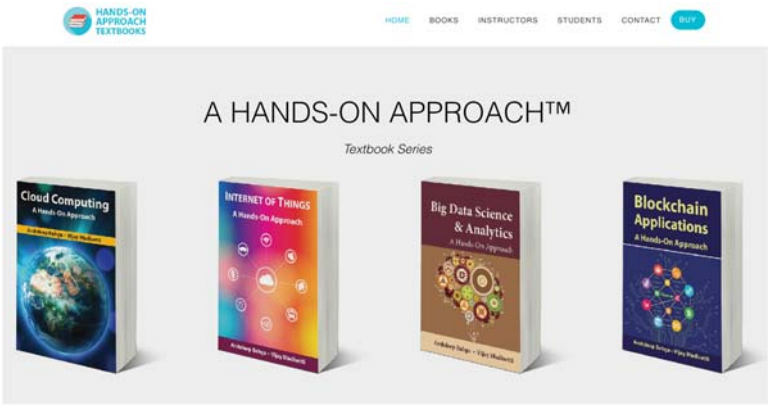



**Figure 4.** Results of performance monitoring in Experiments 1 and 2. (a) Average CPU use in Experiment 1. (b) CPU use density of the catalog component in Experiment 1. (c) Throughput and response time in Experiment 1. (d) Average CPU use in Experiment 2. (e) CPU use density of the customer profile component in Experiment 2. (f) Throughput and response time in Experiment 2. The two experiments show that scaling up the catalog component can increase throughput and decrease response time.

**27.** In addition to research and commercialization projects in secure computing, in virtualization and cloud computing and advanced cloud applications and data

analytics, I have also taught courses and written books in these areas. These books are widely used by dozens of universities all over the world.

**28.** I also have nearly thirty issued or pending applications for US Patents in the past twenty years in areas that include content management and digital rights management, encryption and crypto-currency applications.




### C. Basis of My Opinion and Materials Considered

**29.** I have reviewed the '815 patent and the prior art and other documents and materials cited herein. For ease of reference, the full list of documents that I have considered is included in Appendix B.

**30.** My opinions in this declaration are based on my review of these documents, as well as upon my education, training, research, knowledge, and experience.

**31.** I have reviewed, had input into, and endorse as set forth fully herein the discussions in the accompanying Petition.

### D. Legal Standards for Patentability

**32.** I am not an attorney and I offer no opinions on the law itself. My understanding of the relevant law principles this section is based on information provided to me by counsel.

#### 1. Priority

**33.** I was informed and understand that for a patent to claim the benefit of an earlier provisional application, each application in the chain leading back to the provisional must comply with the written description requirement of 35 U.S.C. § 112. To satisfy the written description requirement, the specification of the provisional must contain a written description of the claimed invention and the manner and process of making and using it, in such full, clear, concise, and exact terms to enable an ordinarily skilled artisan to practice the invention. Moreover, the provisional application must describe the later claimed invention in sufficient detail that a person of ordinary skill in the art can clearly conclude that the inventor had possession of the claimed invention as of the provisional filing date.

#### 2. Obviousness

**34.** I have been informed that a claim may be unpatentable under 35 U.S.C. § 103(a) if the subject matter described by the claim as a whole would have been obvious to a hypothetical person of ordinary skill in the art in view of a prior art reference or in view of a combination of references at the time the alleged

invention was made. I have been informed that obviousness is determined from the perspective of a hypothetical person of ordinary skill in the art and that the asserted claims of the patent should be read from the point of view of such a person at the time the alleged invention was made. I have been informed that a hypothetical person of ordinary skill in the art is assumed to know and to have all relevant prior art in the field of endeavor covered by the patent in suit, and would thus have been familiar with each of the references cited herein, as well as the background knowledge in the art discussed in Section VII, and the full range of teachings they contain.

**35.** I have been informed that there are two criteria for determining whether prior art is analogous and thus can be considered prior art: (1) whether the art is from the same field of endeavor, regardless of the problem addressed, and (2) if the reference is not within the field of the patentee's endeavor, whether the reference still is reasonably pertinent to the particular problem with which the patentee is involved. I have also been informed that the field of endeavor of a patent is not limited to the specific point of novelty, the narrowest possible conception of the field, or the particular focus within a given field. I have also been informed that a reference is reasonably pertinent if, even though it may be in a different field from that of the patentee's endeavor, it is one which, because of the matter with which it

deals, logically would have commended itself to a patentee's attention in considering his problem.

**36.** I have also been informed that an analysis of whether an alleged invention would have been obvious should be considered in light of the scope and content of the prior art, the differences (if any) between the prior art and the alleged invention, and the level of ordinary skill in the pertinent art involved. I have been informed as well that a prior art reference should be viewed as a whole.

**37.** I have also been informed that in considering whether an invention for a claimed combination would have been obvious, I may assess whether there are apparent reasons to combine known elements in the prior art in the manner claimed in view of interrelated teachings of multiple prior art references, the effects of demands known to the design community or present in the market place, and/or the background knowledge possessed by a person having ordinary skill in the art. I have been informed that other principles may be relied on in evaluating whether an alleged invention would have been obvious, and that these principles include the following:

- A combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results;
- When a device or technology is available in one field of endeavor, design incentives and other market forces can prompt variations of it, either in the

same field or in a different one, so that if a person of ordinary skill can implement a predictable variation, the variation is likely obvious;

- If a technique has been used to improve one device, and a person of ordinary skill in the art would recognize that it would improve similar devices in the same way, using the technique is obvious unless its actual application is beyond his or her skill;
- An explicit or implicit teaching, suggestion, or motivation to combine two prior art references to form the claimed combination may demonstrate obviousness, but proof of obviousness does not depend on or require showing a teaching, suggestion, or motivation to combine;
- Market demand, rather than scientific literature, can drive design trends and may show obviousness;
- In determining whether the subject matter of a patent claim would have been obvious, neither the particular motivation nor the avowed purpose of the named inventor controls;
- One of the ways in which a patent's subject can be proved obvious is by noting that there existed at the time of invention a known problem for which there was an obvious solution encompassed by the patent's claims;

- Any need or problem known in the field of endeavor at the time of invention and addressed by the patent can provide a reason for combining the elements in the manner claimed;
- "Common sense" teaches that familiar items may have obvious uses beyond their primary purposes, and in many cases a person of ordinary skill will be able to fit the teachings of multiple patents together like pieces of a puzzle;
- A person of ordinary skill in the art is also a person of ordinary creativity, and is not an automaton;
- A patent claim can be proved obvious by showing that the claimed combination of elements was "obvious to try," particularly when there is a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions such that a person of ordinary skill in the art would have had good reason to pursue the known options within his or her technical grasp; and
- One should be cautious of using hindsight in evaluating whether an alleged invention would have been obvious.

**38.** I have further been informed that, in making a determination as to whether or not the alleged invention would have been obvious to a person of ordinary skill, the Board may consider certain objective factors if they are present, such as: commercial success of products practicing the alleged invention; long-felt but

unsolved need; teaching away; unexpected results; copying; and praise by others in the field. These factors are generally referred to as "secondary considerations" or "objective indicia" of nonobviousness. I have been informed, however, that for such objective evidence to be relevant to the obviousness of a claim, there must be a causal relationship (called a "nexus") between the claim and the evidence and that this nexus must be based on what is claimed and novel in the claim rather than something in the prior art. I also have been informed that even when they are present, secondary considerations may be unable to overcome primary evidence of obviousness (e.g., motivation to combine with predictable results) that is sufficiently strong.

**39.** I have been asked to consider the patentability of Claims 42 and 43 (the "Challenged Claims") of the '815 patent that are challenged in the Petition. I have been informed that for *inter partes* reviews, unpatentability must be shown under a preponderance of the evidence standard. I have been informed that to establish something by a preponderance of the evidence one needs to prove it is more likely true than not true. I have concluded that each of the Challenged Claims is unpatentable under 35 U.S.C. § 103 based on **Hanna** in view of **Stefik**, or under 35 U.S.C. § 103 based on **Gennaro**, as described below, under both the preponderance of the evidence standard as well as the higher standard of clear and convincing evidence.

### 3. Claim Construction

**40.** I have been informed that patent claims are construed from the viewpoint of a person of ordinary skill in the art at the time of the alleged invention. I have been informed that patent claims generally should be interpreted consistent with their plain and ordinary meaning as understood by a person of ordinary skill in the art in the relevant time period (*i.e.*, at the time of the purported invention, or the so called "effective filing date" of the patent application), after reviewing the patent claim language, the specification and the prosecution history (*i.e.*, the intrinsic record).

**41.** I have further been informed that a person of ordinary skill in the art must read the claim terms in the context of the claim itself, as well as in the context of the entire patent specification. I understand that in the specification and prosecution history, the patentee may specifically define a claim term in a way that differs from the plain and ordinary meaning. I understand that the prosecution history of the patent is a record of the proceedings before the U.S. Patent and Trademark Office, and may contain explicit representations or definitions made during prosecution that affect the scope of the patent claims. I understand that an applicant may, during the course of prosecuting the patent application, limit the scope of the claims to overcome prior art or to overcome an examiner's rejection, by clearly and unambiguously arguing to overcome or distinguish a prior art reference, or to clearly and unambiguously disavow claim coverage.

**42.** In interpreting the meaning of the claim language, I understand that a person of ordinary skill in the art may also consider "extrinsic" evidence, including expert testimony, inventor testimony, dictionaries, technical treatises, other patents, and scholarly publications. I understand this evidence is considered to ensure that a claim is construed in a way that is consistent with the understanding of those of skill in the art at the time of the alleged invention. This can be useful for technical terms whose meaning may differ from its ordinary English meaning. I understand that extrinsic evidence may not be relied on if it contradicts or varies the meaning of claim language provided by the intrinsic evidence, particularly if the applicant has explicitly defined a term in the intrinsic record.

## II. DESCRIPTION OF THE RELEVANT FIELD AND THE RELEVANT TIMEFRAME

**43.** I have carefully reviewed the '815 patent and its prosecution history.

**44.** I understand that the '815 patent issued from U.S. Patent Application 09/588,652, filed on June 7, 2000, and claims the benefit of U.S. Provisional Patent Application 60/138,171, filed on June 8, 1999.

**45.** I have been instructed by counsel to assume the relevant timeframe for my analysis in this declaration to be on or before June 7, 2000. Although the '815 patent claims the benefit of priority of provisional application No. 60/138,171 ("Serret-Avila Provisional," Ex. 1004), as I mention in Sections V.B. below, Claims 42 and 43 are not described in the Serret-Avila Provisional from the prospective of person

of one skilled in the art, so it is my understanding that the priority claim may be determined improper under the law. To the extent that it may be argued that Claims 42 and 43 can claim priority to the Serret-Avila Provisional, my analysis in this declaration also applies to the timeframe on or before June 8, 1999.

**46.** Based on my review of this material, I believe that the relevant general fields for the purposes of the ’815 patent are electronic data protection and distribution.

## III. THE PERSON OF ORDINARY SKILL IN THE RELEVANT FIELD IN THE RELEVANT TIMEFRAME

**47.** My opinions are provided based on what a person of ordinary skill in the art in the technical field of the invention would have understood at the time of the invention of the Challenged Claims. As I explained above, the relevant timeframe in this Declaration is on or before June 7, 2000. I have been informed and understand that the content of a patent and prior art should be interpreted the way a person of ordinary skill in the art (“POSITA”) would have interpreted those references at the time of the invention of the patent using the ordinary and customary meanings of the claim terms. I understand that the POSITA is a hypothetical concept referring to one who thinks along the lines of conventional wisdom at the time.

**48.** I was asked to provide an opinion as to the level of a POSITA pertinent to the subject matter set forth in the Challenged Claims of the ’815 patent at the time

of the priority date. I have been informed that the level of one of ordinary skill in the art is evidenced by prior art references. In the relevant time period, it is my opinion that a POSITA in the field of the Challenged Claims would have been someone with a good working knowledge of electronic data protection and distribution. A POSITA to whom the patent is addressed would have a Bachelor of Science 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. The basis for my familiarity with the level of ordinary skill is my own technical experience and my own interaction with large numbers of students and my employees in the computing field who were at this level of skill. In reaching this opinion as to the hypothetical POSITA, I have considered the types of problems encountered in the art, the prior art solutions to those problems, the rapidity with which innovations are made, the sophistication of the technology, and the educational level and professional capabilities of workers in the field.

## IV. TECHNICAL BACKGROUND AND STATE OF THE ART

**49.** The ’815 patent describes “methods and systems for encoding and protecting data using digital signature and watermarking techniques.” (Ex. 1001, Title). While the listed inventors seem to believe they have described and claimed a novel technique, the method of verifying the authenticity of a portion of a file, the use of digital signature and comparison of hash values, and receiving and granting a request to use the file in a predefined manner, has been previously practiced, and in the same manner as claimed in Claims 42 and 43 of the ’815 patent:

> 42. A method for managing at least one use of a file of electronic data, the method including:
>
> receiving a request to use the file in a predefined manner;
>
> retrieving at least one digital signature and at least one check value associated with the file;
>
> verifying the authenticity of the at least one check value using the digital signature;
>
> verifying the authenticity of at least a portion of the file using the at least one check value; and
>
> granting the request to use the file in the predefined manner.

> 43. 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:
>
> hashing at least a portion of the file to obtain a first hash value; and
>
> comparing the first hash value to at least one of the one or more hash values.

Below I provide some background information related to Claims 42 and 43 of the ’815 patent.

**50.** The ’815 patent recognizes that the distribution and use of unauthenticated, unauthorized, corrupted, and/or modified files is a problem, especially in the Internet age. (Ex. 1001, 1:55-14). It further recognizes that there is “a need for systems and methods for protecting electronic content and/or detecting unauthorized use or modification thereof.” (Ex. 1001, 2:32-34). Such need has been recognized in the prior art well before 1999 and 2000. For example, Hanna is motivated by “a need for a system that protects against unintentional data corruption and malicious acts that cause errors in data processing and allows data to be checked as it is received” (Ex. 1006, 3:1-4); and Stefik recognizes “[a] fundamental issue … is how to prevent the unauthorized and unaccounted distribution or usage of electronically published materials” (Ex. 1007, 1:10-24). *See also, e.g.*, Ex. 1008, 5:57-58 (halting processing of electronic data when tampering is detected); Ex. 1009, 188 (“stops playing the stream”); Ex. 1011, 1:19-20 (expressing concern that digital image data is “extremely susceptible to unauthorized altering”); Ex. 1012, 1:25-32 (identifying problem with data corruption or alteration during transmission).

**51.** To address such need, methods that increase the security of data transmitted over computer networks have been developed. For example, it was known to “include encrypting all or part of the data prior to sending it, and likewise decrypting the received data prior to using it.” (Ex. 1012: 1:33-42). It was also known to use

signature algorithms, such as Message Digest, RSA, and DSA. (*Id.* at 1:42-61). *See also* Ex. 1006, 1:28-31, Ex. 1013, 1:15-26.

**52.** A signature algorithm often involves the use of a key pair. For example, a person sending a data file can use a "private" key unique to that person to create a digital signature for the data file. A corresponding "public" key, which can be used to verify the digital signature created using the private key, is available to the recipient of the data file. Anyone possessing the public key can use it to verify the digital signature created using the corresponding private key, but no one can create the digital signature without access to the private key. As Dolan et al. explained in a patent issued in 1997, "[t]he principle behind these techniques is the creation of a public digital value—the signature—which depends on a message to be transmitted and the signing user, so the receiving user can be sure that the sending user, and no other user, could create the signature value, and that the user created the signature value for this message and no other." (Ex. 1013, 1:28-41). *See also, e.g.* Ex. 1011, 1:42-61; Ex. 1012, 1:62-2:17; Ex. 1018, 26:25-59.

**53.** The digital signature may include information useful for checking the integrity of the data file. Therefore, a recipient of the "signed" data file can be sure that the data has not been tampered with when the digital signature is verified. This information may include one or more values generated by applying a hash function to data in the file. (Ex. 1012, 2:18-33). Applying a publicly-known mathematical

function to the data, a sender of the data can generate a "hash," which is a smaller representation of the data. Each "hash" is unique in that changing even a single bit in the original data would produce a very different hash output. An example of such technique was explained by Squilla et al. in a patent issued in early 1999:

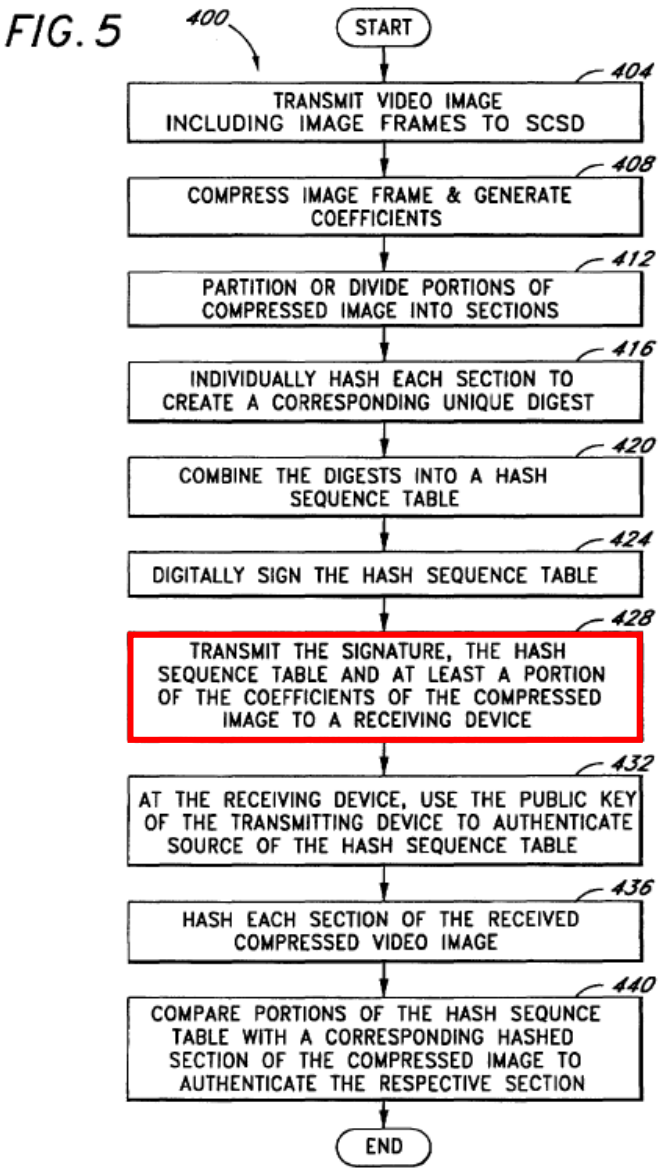
> In practice, a "hash" is created by the sender from the message by using a publicly-known mathematical function which maps values from a large domain into a smaller range. For example, a checksum is a simple kind of hash that produces a simple hash file which is much smaller than the original file yet is particularly unique to it. (For example of another algorithm see "One Way Hash Functions" by Bruce Schneier, Dr. Dobb's Journal, September 1991, pp. 148-151). [M]any more complex and secure transformations are well known to those of skill in this art. The hash file makes it possible to encrypt a smaller representation of the message (the "hash") in order to verify the integrity and source of the larger message. Another advantageous characteristic of a hash is that changing a single bit in the original message input would produce a very different hash output if subjected to the same mathematical hash function.

(Ex. 1011, 1:62-2:10; Ex. 1017; *see also* Ex. 1018, 26:60-27:10).

**54.** Thus, by including a hash value in the digital signature, the sender can choose to encrypt only the hash value, rather than the data itself. This typical use is described in the Dolan patent: "Typically, the signer produces a hash value from the message using a strong hash algorithm, such that the chance of another message resulting in the same value is extremely low. The means of calculating this value is public knowledge but there is no feasible way to determine a different message which results in the same value. The signer encrypts the value using the private key, and sends the message and the encrypted value to the recipient." (Ex. 1013, 1:42-

58, 6:1-15). *See also*, e.g., Ex. 1006, 4:19-21; Ex. 1008, 4:59-5:2; Ex. 1009, 184-185, 188.

**55.** It was also known to apply a hash function to a portion of the data in a file resulting in the generation of multiple hash values for the file, and to use those hash values to generate a digital signature. The underlying hash values are sent with the digital signature to the recipient of the data file. (Ex. 1014, 4:30-5:24). It was also known to transmit a hash sequence table (comprising hashes of the data portions), along with the digital signature of the table, to a receiving device. The transmission of the signature, along with the hash sequence table, was illustrated by Davis in a patent issued in early 1999:




(color-annotated Fig. 5 of Ex. 1015; *see also*, *id.* at 6:20-49). Similar techniques were described in other references in the relevant timeframe. *See, e.g.*, Ex. 1006, 3:18-22, Fig. 3; Ex. 1008, 1:24-45, 4:63-5:2, 5:23-28, Fig. 2A; Ex. 1009, 182.

**56.** There are advantages to combining multiple hash values together to generate a digital signature, and sending those hash values with the digital signature. One advantage is that once the digital signature is verified, each of the hash values can

be used to verify the authenticity of the corresponding data portion. Thus, this can involve less computation than generating a digital signature from each hash value and having to verify multiple digital signatures for the data file. Where the portions are individual packets, this "allows the data packets to be quickly verified as they are received." (Ex. 1006, 3:8-10).

**57.** In other words, if both the digital signature and hashes are sent, along with the data file, then a receiving system can use the signature to verify the hashes. (Ex. 1006, 4:22-26). Because the hashes are signed using a private key to create the digital signature, and further if the digital signature is verified, then the receiver would know the hashes are also verified and authentic. (Ex. 1013, 1:42-58; Ex. 1006, 9:22-10:16; Ex. 1008, 5:30-34). Once a hash is verified, then it can be used to verify its corresponding data block. This allows faster processing on verifying the data block. (Ex. 1006, 3:6-13, 4:22-26; Ex. 1008, 2:39-41).

**58.** Processes for receiving and granting a request to use a digital file were also known. A computer system was known to send and receive messages in a network, including communication with another remote system. (Ex. 1006, 6:26-7:14). For example, Hanna discloses a bus 102 for internal communication and a communication interface 118 for external communication with a network:

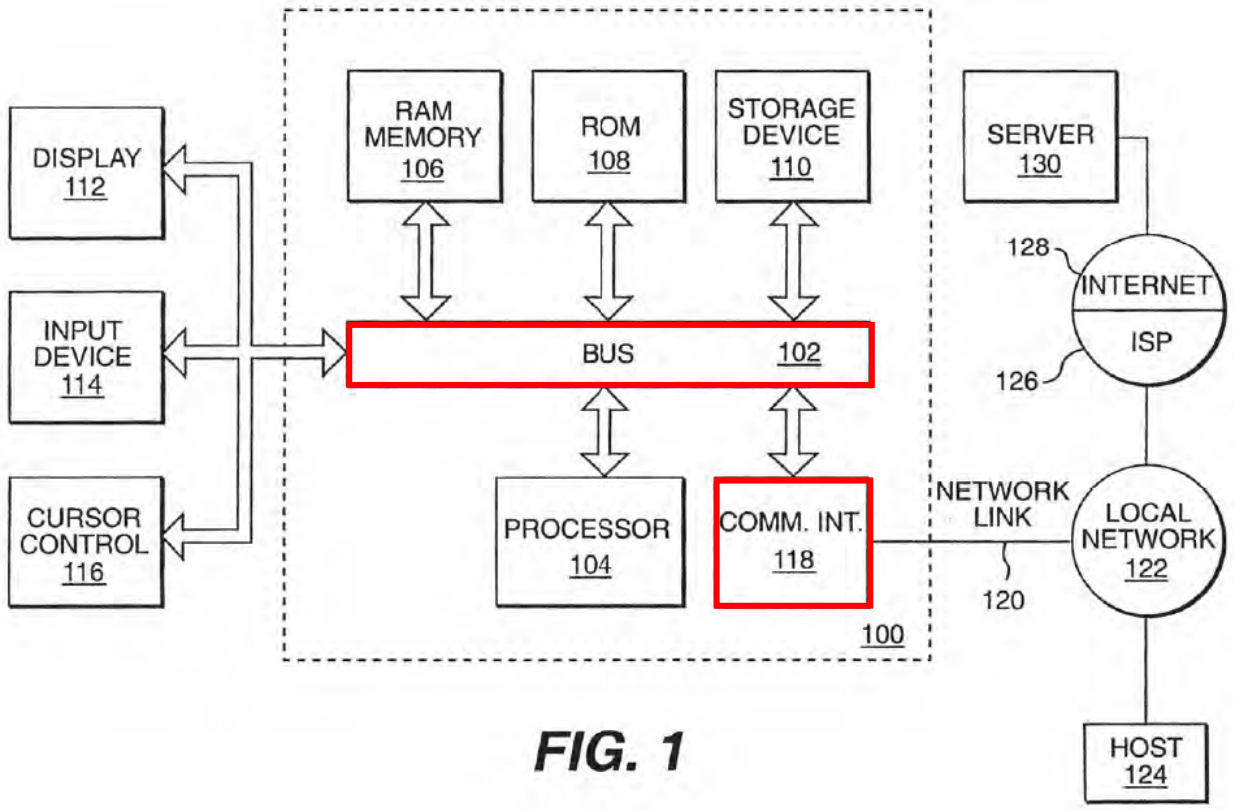



FIG. 1

(color-annotated Fig. 1 of Ex. 1006). Such communication was known to involve making a request, receiving the request and granting the request to use an electronic file. (Ex. 1006, 1:10-15, 7:15-23; Ex. 1007, 7:19-30, 26:37-47, 42:26-31; Ex. 1016, 8:13-26)

**59.** Receiving a request to use an electronic file in a predefined manner was known to take many forms. One way to receive a request is from a user via an interface to use the file for a certain purpose, such as playing a digital movie, or installing a digital work. (Ex. 1007, 16:61-67; 26:37-47, 42:26-31; Ex. 1008, 2:64-67). When a user requests to use a file comprising a bit stream, a receiver of the user's request may in turn request transmission of such file from a sender, and the

sender would thus also receive a request to use the file for a certain purpose. (Ex. 1008, 1:28-29; Ex. 1009, 181-182). Another way is for component of a system to receive a request from another component of the system to use the file for a certain purpose, such as obtaining a copy of the file. (Ex. 1007, 7:19-23, 20:48-52). The receiver of such request may have the file in its storage, or have the ability to allow the file to be used as requested. For example, it was known for authentication software to receive a request from the component to further process the file once the file has been authenticated. (Ex. 1008, 5:4-20, 5:35-67).

**60.** Granting a request to use an electronic file in a predefined manner was also known to take many forms. One way to grant a request is simply by servicing the request; for example, by playing a digital stream as requested by the user (Ex. 1007, 16:61-67; Ex. 1008, 2:54, 2:65-67; Ex. 1009, 188). Another way to grant a request is by allowing the requested use of the electronic file; for example, by passing the file to the requester for further processing. (Ex. 1007, 7:19-33, 20:48-54, 42:26-43:2; Ex. 1008, 5:4-20, 5:35-67). An example of such request receiving and granting process is further described:

> Another embodiment of the invention is a user device that includes an input port that receives an instance of software and receives a tag uniquely associated with that instance of software and also ***receives a request to use the instance of software.*** A processor included in the user device executes a supervising program. The supervising program ***detects the request to use the instance of software*** and verifies the authenticity of the tag associated with the instance of software ***before allowing use of the instance of software by the user device***. The supervising program also verifies the authenticity of the

> tag and stores the tag in a tag table and maintains the instance of software if the tag is authentic and rejects the instance of software if the tag associated with the software is not authentic.

(Ex. 1016, 8:13-26)(emphasis added).

## V. THE ’815 PATENT

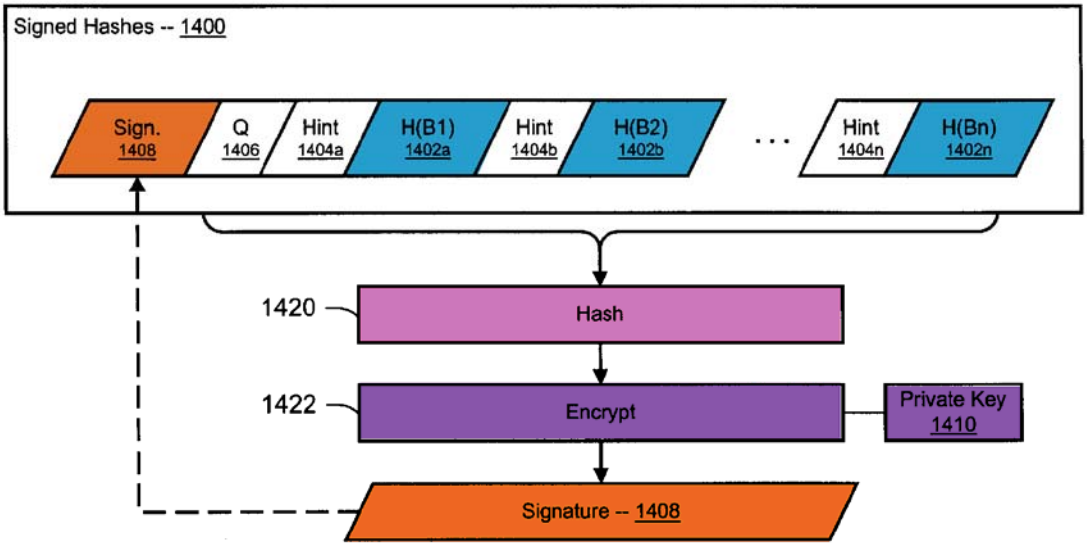
### A. Overview

**61.** The ’815 patent is directed to a system and method for encoding and protecting data using digital signatures. (Ex. 1001, Title, Abstract). The encoding and protection is to facilitate using data “in a certain manner—such as copying, moving, viewing, or printing the data.” (Ex. 1001, 4:65-5:3) To address a problem that files are modified without authorization during data transmission, the ’815 patent presents a system and method for detecting unauthorized modification of electronic data. (Ex. 1001, 2:6-10, 2:31-33).

**62.** One embodiment disclosed in the ’815 patent is managing the use of electronic data 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). Such method is recited in Claims 42 and 43 of the ’815 patent. (Ex. 1001, 32:13-34).

**63.** The claimed method allows the integrity of the data to be checked before allowing it to be used. (Ex. 1001, 23:33-34). Figures 13 and 14 illustrate such

claimed method. The color-annotated 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:




The original content file is divided into different portions. Those portions are hashed to yield a plurality of **hash values 1402**. These hash values, along with other data (such as hints 1404 and quality indicator 1406), are further hashed to yield the **hash value 1420**. The **hash value 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. (Ex. 1001, 23:40-60).

**64.** The color-annotated Figure 13 illustrates that the signed hash file is provided to a user who has made a request to use the content file; in this example, it is a request

to copy a file. But the 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).

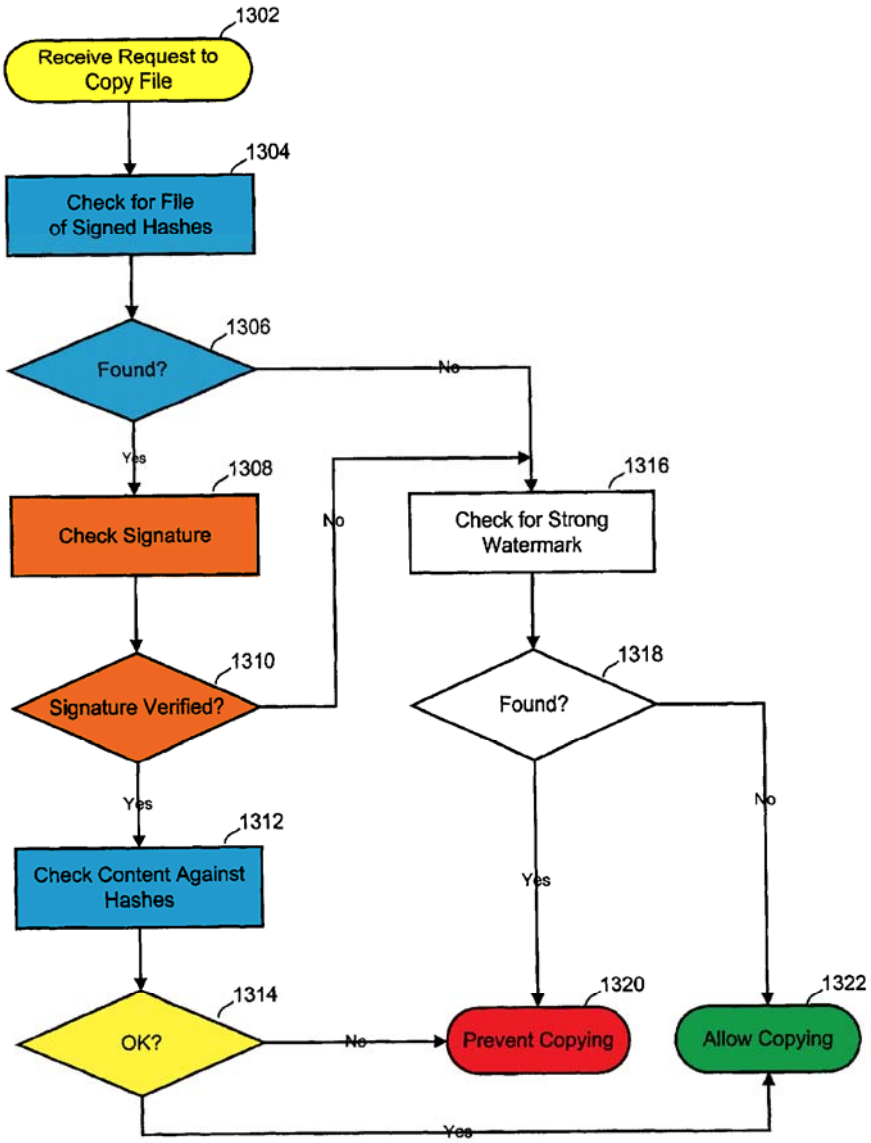



FIG. 13

**65.** 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**. (Ex. 1001, 23:1-5). In **step 1308**, the **digital signature** is decrypted

using the sender's public key, which yields a **first hash value**. (Ex. 1001, 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. (Ex. 1001, 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**. (Ex. 1001, 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.*)

**66.** In **step 1312**, if the **check values** are verified to be authentic, they are used to verify the authenticity of the content in the 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 request to use of the file is granted (**step 1322**). (Ex. 1001, 23:14-16).

### B. The Serret-Avila Provisional

**67.** I have reviewed the provisional application No. 60/138,171 ("Serret-Avila Provisional, Ex. 1004).

**68.** In the Serret-Avila Provisional, I do not find a technical disclosure of the limitations of Claim 42 that would indicate to a POSITA that the inventors were in possession of such an alleged invention. For example, the 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." Ex. 1005.

**69.** Specifically, Serret-Avila Provisional does not include the corresponding summary of the embodiment (Ex. 1001, 4:50-58), the corresponding disclosures (Ex. 1001, 23:4-10, 23:54-60), and Figures 13 and 14 in the '815 specification. Indeed, I have reviewed the entirety of the Serret-Avila Provisional for this technical content, perhaps conveyed in different words or figures, and am unable to find such written description support that would indicate to a POSITA that the inventors were in possession of such an alleged invention.

**70.** The Serret-Avila Provisional describes and depicts conventional cryptographic signatures techniques in Figure 2A through Figure 2C. These descriptions do not mention retrieving the check/hash value associated with the data. Figures 2A, 2B and 2C do not appear in the '815 specification. They appear to be similar to Figures 2A and 2B of the '815 specification. Figure 2A illustrates that 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). Figure 2B depicts applying a hash function on the data and then encrypting the resulting hash value to create a digital signature, but only the signature (and not the hash value) is inserted into the

data stream. (Ex. 1004, 11). Figure 2C depicts that the content stream to be verified is composed of data blocks and signatures only. (Ex. 1004, 3).

**71.** The Serret-Avila Provisional discloses the "shared signature" technique in Figure 6. Figure 6 does not appear in the '815 specification, but it appears to be similar to Figure 8 of the '815 specification. This technique modifies 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, a single "shared signature" is formed by encrypting a concatenation of hashes corresponding to multiple blocks of data. (Ex. 1004, 20). 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, that would indicate to a POSITA that the inventors were in possession of such an alleged invention. (Ex. 1004, 20-21)

## VI. CLAIM INTERPRETATION

**72.** I have been informed that for purposes of this *Inter Partes* Review, the standard for claim construction is the same as the standard used in the federal district courts: claim terms should generally be given their ordinary and customary meaning as understood by one of ordinary skill in the art at the time of the invention and after reading the patent and its prosecution history.

**73.** I have also been informed that only terms necessary to resolve the controversy need to be construed. For the purposes of the present declaration, I have applied the legal principles outlined above. I reserve the right to offer opinions on claim construction in response to arguments that may be made by Patent Owner.

## VII. DISCUSSION OF RELEVANT PRIOR ART

### A. Ground 1: Claims 42 and 43 are Rendered Obvious by Hanna in view of Stefik

**74.** It is my opinion that the Challenged Claims 42 and 43 are rendered obvious by **Hanna** in view of **Stefik**. Below, I provide an overview of Hanna, an overview of Stefik, and why a POSITA would modify Hanna in view of Stefik. I further provide an analysis of claim 42, followed by claim 43, and a claim chart of relevant prior art disclosures to explain how the limitations of claim 42 and 43 are met.

#### 1. Overview of Hanna

**75.** 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*). The objective of Hanna is to protect against unintentional data corruption and malicious acts that cause errors in data processing. (Ex. 1006, 3:1-4).

**76.** Prior art systems, as Hanna explains, 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. They also do not allow the data to be verified as it is

received. (Ex. 1006, 2:15-28). Hanna uses a single signed hash block to address this problem. Hanna's solution allows a single digital signature to protect the data set from both data corruption and malicious acts that cause errors in data processing. In addition to the digital signature, the hash block is retrieved with the data. (Ex. 1006, Abstract). Receiving the hashes before the data also allows the data packets to be quickly verified as they are received. (Ex. 1006, 3:6-12).

**77.** To prepare the data for transmission, the data is signed. This data signing process 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). Figure 2 illustrate this process:

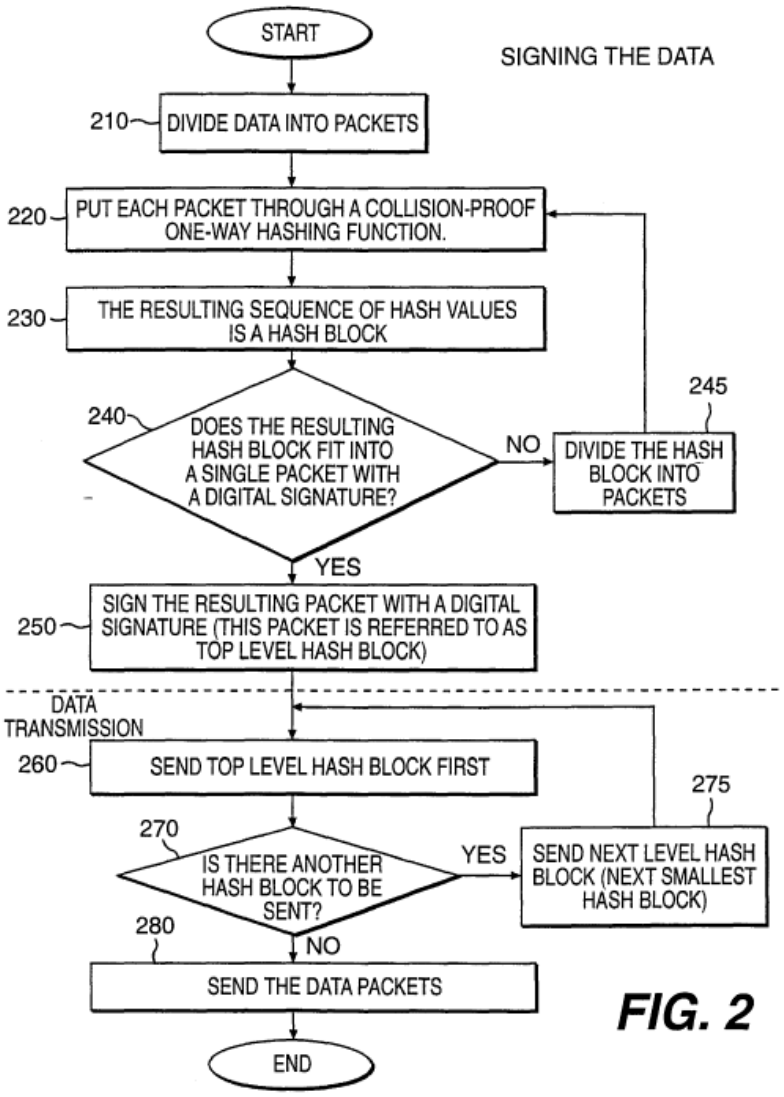



FIG. 2

Step 210 begins data signing by dividing a data set into small packets. Step 220 applies a one-way hash function to each data packet. Step 230 shows that the application of the hash function yields a sequence of hash values, *i.e.*, a hash block. Steps 240 and 250 show that 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 top level hash block. (Ex. 1006, 8:9-9:7). Step 260 sends the top level hash block

containing the hash values and the digital signature, followed by sending the data packets in step 280. (Ex. 1006, 9:11-13).

**78.** On the receiving end, 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). Figure 3 illustrates such verifying process:

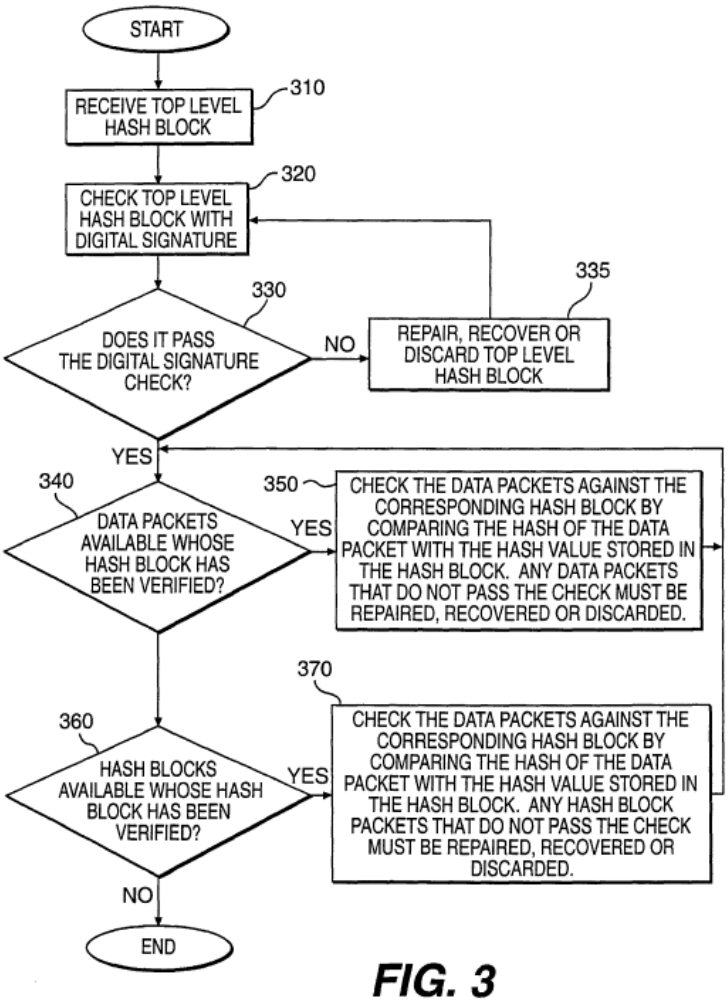



FIG. 3

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). (Ex. 1006, 9:22-10:16).

### 2. Overview of Stefik

**79.** 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. (Ex. 1007, *Title*, *Abstract*). To address the problem of unauthorized distribution and usage of digital works (Ex. 1007, 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. (Ex. 1007, 3:51-60).

**80.** A system responsible for the use and distribution of digital works, as Stefik explains, 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. (Ex. 1007, 12:41-50) Physical integrity refers to the integrity of the physical devices themselves. (Ex. 1007, 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. (Ex. 1007, 13:11-23). Communications integrity is achieved by security measures involving encryption and exchange of digital certificates. (Ex. 1007, 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. A software will only be installed after verifying the authenticity of the software using the digital certificate. (Ex. 1007, 13:30-40).

**81.** In Stefik's rendering system, there are two components: a rendering device and a repository. The rendering device, such as a computer system, a digital audio system, and a printer, renders a digital work. (Ex. 1007, 8:24-28). A user requests access to the digital work on the user interface of a repository. (Ex. 1007, 16:61-67). In response to such a request, the repository would initiate various transactions. (Ex. 1007, 26:38-39)

**82.** For example, on the user interface of the repository, a user makes a request to install a player. (Ex. 1007, 42:25-27). A player may be used to play digital movies/music/video game, run a computer program, or display a document. (Ex. 1007, 19:46-55). Such a request is referred to as an "Install transaction" in Stefik, which is "a request to install a digital work as runnable software on a repository."

(Ex. 1007, 42:26-27). Stefik 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." (Ex. 1007, 42:27-31).

**83.** The Install transaction begins when the requester sends a server a message indicating the software to installed. (Ex. 1007, 42:26-35). The requester repository "extracts a copy of the digital certificate for the software." (Ex. 1007, 42: 29-31, 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. For example, both may rely on key pairs to allow a recipient of data to confirm that the source is authentic. A digital signature may be generated using a private key of the pair, and the recipient can check using the corresponding public key of the pair whether the signature was generated by the private key. *See supra* ¶ 52. A digital certificate may associate the public key with a particular entity, and may be used to distribute the public key. The digital certificate itself may be signed using another key pair associated with the issuer. (Ex. 1018, 27:11-28:5).

**84.** 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 an "1-way hash function" to compute "a check code." This "check code" is then compared against the "tamper-checking code" to "assure[] that the contents of the software … have not been tampered with." (Ex. 1007, 42:49-54). If the "check code" matches with the "tamper-checking code," then the software is verified and the request to install the software is granted. (Ex. 1007, 42:49-62). The installation of the software is allowed to proceed.

**85.** Further, in response to a user request, a rendering repository may need to request access to a digital work from another repository. For example, if the request to access the digital work is for a stated purpose, such as to obtain a copy of the digital work, then the repository responding to this request would check for the corresponding specific usage right. (Ex. 1007, 7:19-23). The responding repository would determine if such request to access the digital work may be granted, such as determining whether all conditions associates with the rights are satisfied. (Ex. 1007, 7:23-30). If the request is granted, then the responding repository transmits the digital work to the rendering repository. (Ex. 1007, 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. 1007, 20:48-53).

### 3. Motivation to Modify Hanna In view of Stefik

**86.** In Hanna, the disclosed computer system 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). In response to the processor executing instructions, Hanna explains that 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. (Ex. 1006, 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. (Ex. 1006, 7:15-23). 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 a host or the computer system itself, and such requests are processed in accordance to its disclosed method and system of signing and verifying data.

**87.** Hanna further 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. For example, Hanna discloses 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). Hanna discloses another example—video conference—an application that requires secure transmission of information, such as video, voice, and digital data. (Ex. 1006, 1:13-15).

**88.** 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 be communication to convey the electronics 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. Therefore, a POSITA would have recognized from Hanna that a file may be checked to verify its authenticity when received or retrieved for use in a predefined manner. (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 in a predefined manner, and granting that request if the data is verified. As discussed in Section IV above, it was well known in the art to receive a request to use an electronic file in a predefined manner, and upon authentication of the file, grant the request accordingly.

**89.** To the extent that it may be argued that Hanna does not *explicitly* describe 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 after authentication, these

steps were disclosed by Stefik. A POSITA would have been motivated to look to and incorporate Stefik's disclosure of receiving and granting requests to use files of electronic data in a remote access environment because Stefik discloses receiving and granting a request from a repository different than the one in which the file is stored. (Ex. 1007, 16:43-67, 42:25-43:4). Similarly, Hanna describes a remote access environment, such as 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. (Ex. 1006, 1:10-15, 7:15-23).

**90.** Hanna and Stefik are from the same field of endeavor, digital file security; and both address the problem of unauthorized manipulation of digital files. Also, they both propose using a digital authentication mechanism, *i.e.* a digital signature or certificate, to identify the source of a 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 motivated to modify the system of Hanna to include a request/grant process as taught by Stefik. Doing so would be advantageous because it would facilitate timely detection to determine whether a portion of the electronic file was corrupted or hacked during transmission before it is used.

**91.** A POSITA would have been further 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, because Hanna is directed to protecting transmitted data, and the modification would improve control over use and distribution of valuable data. (Ex. 1006, 1:6-21, 3:1-4). Granting of the request could be based upon satisfying certain conditions, such as whether the requested use meets a predefined usage right, or whether at least a portion of the file has been authenticated. (Ex. 1007, 7:19-30, 42:43-55).

**92.** A POSITA would recognize 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 it was designed, for managing use of an electronic file. This is because the computer system in Hanna already has a communication mechanism in place to facilitate the request/grant process as taught in Stefik, as discussed above. (Ex. 1006, 5:2-5, 6:26-7:7-23).

#### 4. Claim 42

##### a) [42pre] "A method for managing at least one use of a file of electronic data"

**93.** **Hanna** discloses a method for managing at least one use (e.g., "cop[ying]," "download[ing]," "retriev[ing]," "receiving," "execut[ing]," or "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, 4:22-28, 7:15-26, 12:3-4, Figs. 2-5). For example, Hanna discloses "cop[ying]" files to a remote database for "backup" purposes (Ex. 1006, 1:10-13); Hanna further discloses "download[ing]" an application program from the Internet

(Ex. 1006, 7:15-26); Hanna further discloses "retriev[ing]" or "receiv[ing]" data from a source, such as a remote device (Ex. 1006, *Abstract*, 2:27-28, 3:1-3, 3:18-22, 7:22-24, 12:3-4); and Hanna further discloses "storing" or "execut[ing]" instructions (Ex. 1006, 5:5-9, 5:25-29, 6:3-4, 7:24-25).

**94.** To the extent it may be argued that 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.*, "installation") of a file of electronic data (*e.g.*, "a digital work" or "software").

**95.** Stefik describes a rendering system having at least two components: a repository and a rendering device. A repository has a user interface that permits a user to make access requests of the digital work. (Ex. 1007, 16:61-67). 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). Such rendering is one way of using the digital work.

**96.** One example of such use of a digital work would be to install a player. (Ex. 1007, 42:25-27). Stefik discloses an "Install transaction" for installing "a new kind or new version of a player." (Ex. 1007, 42:26-31). In such a transaction, "the software would be copied to file system of the requester repository before it is installed." (Ex. 1007, 42:26-31).

**b) [42a] "receiving a request to use the file in a predefined manner"**

**97.** **Hanna** discloses "transmit[ing] a requested code for an application program." (Ex. 1006, 7:16-18). Such transmission is made through the Internet, local network, and communication interface. (Ex. 1006, 7:16-18). When the received code is received by a processor, the processor performs various actions in response to such code, such as downloading, storing, executing, or transmitting the program. (Ex. 1006, 7:18-26). A POSITA would recognize that a request to copy, store, and/or execute the program has been made because the requested code transmitted from the server is "for an application program," and that such code would be used for its intended application. (*Id.*)

**98.** Hanna further discloses "back[ing] up" computer files to a remote central database, and "secure transmission of information" in video conferencing." (Ex. 1006, 1:10-15). In order to facilitate transmission of electronic files in the remote access environment described in Hanna, a POSITA would recognize that there would be communication for requesting such file to be used in a predefined manner. *See supra* ¶¶ 58-59.

**99.** To the extent it may be argued that a "request" to use the file in a predefined manner is not already disclosed in Hanna, **Stefik** discloses receiving a request from a user to use the file in a predefined manner. For example, the user "requests access to a digital work" and "request[s] specific transactions" to be performed. (Ex. 1007,

26:38-47). Stefik discloses that such user request can be made via a user interface of a repository. (Ex. 1007, 16:61-67).

**100.** Stefik further discloses a repository receiving a request from another repository to use an electronic file in a predefined manner. For example, Stefik discloses that a digital work is stored securely in "Repository 1". "Repository 2," which may be a rendering repository, may request access to the digital work "for a stated purpose." Repository 1 receives the request from Repository 2 to use the digital work for the stated purpose, such as printing or copying the digital work. In response to such request, Repository 1 checks the usage rights associated with the digital work to determine if the request to use the digital work for the stated purpose may be granted. (Ex. 1007, 7:19-32).

**101.** Stefik further discloses that a request to install a new type of player within a rendering repository. Because installation is a usage right associated with a digital work, such request is to use an electronic file in a predefined manner. (Ex. 1007, 20:48-53, 42:26-31).

**c) [42b] "retrieving at least one digital signature and at least one check value associated with the file"**

**102. 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").

**103.** Figure 2 of Hanna shows that both the digital signature and the hash values associated with the data file are retrieved:

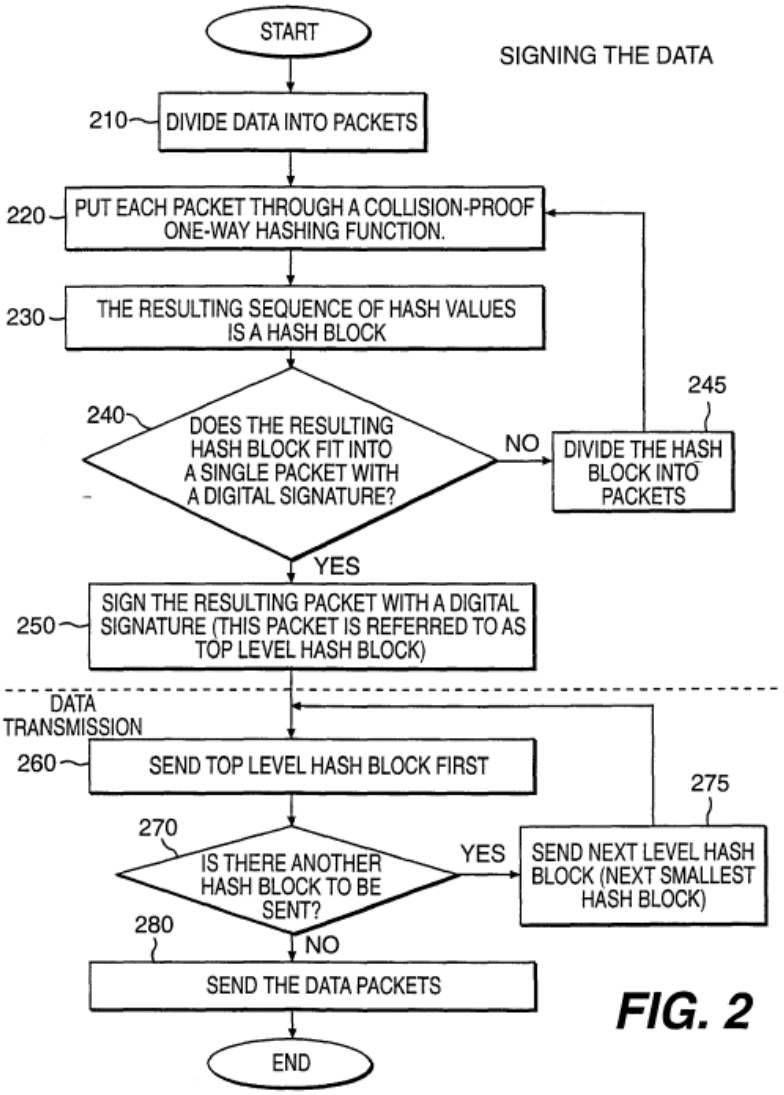



FIG. 2

**104.** In step 260 of Figure 2, Hanna explains that the verification begins with retrieving the top level hash block. From step 250, Hanna discloses that such top level hash block includes a digital signature and hash values that are part of the "top level hash block." (Ex. 1006, 9:1-24). 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). A POSITA would understand that the hash values are associated with the file because they are computed based on the data from the file. (Ex. 1006, 8:14-17); *see also supra* ¶¶ 53-57.

**d) [42c] "verifying the authenticity of the at least one check value using the digital signature"**

**105.** **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").

**106.** Figure 3 of Hanna shows that the hash values in the hash block are verified using the digital signature:

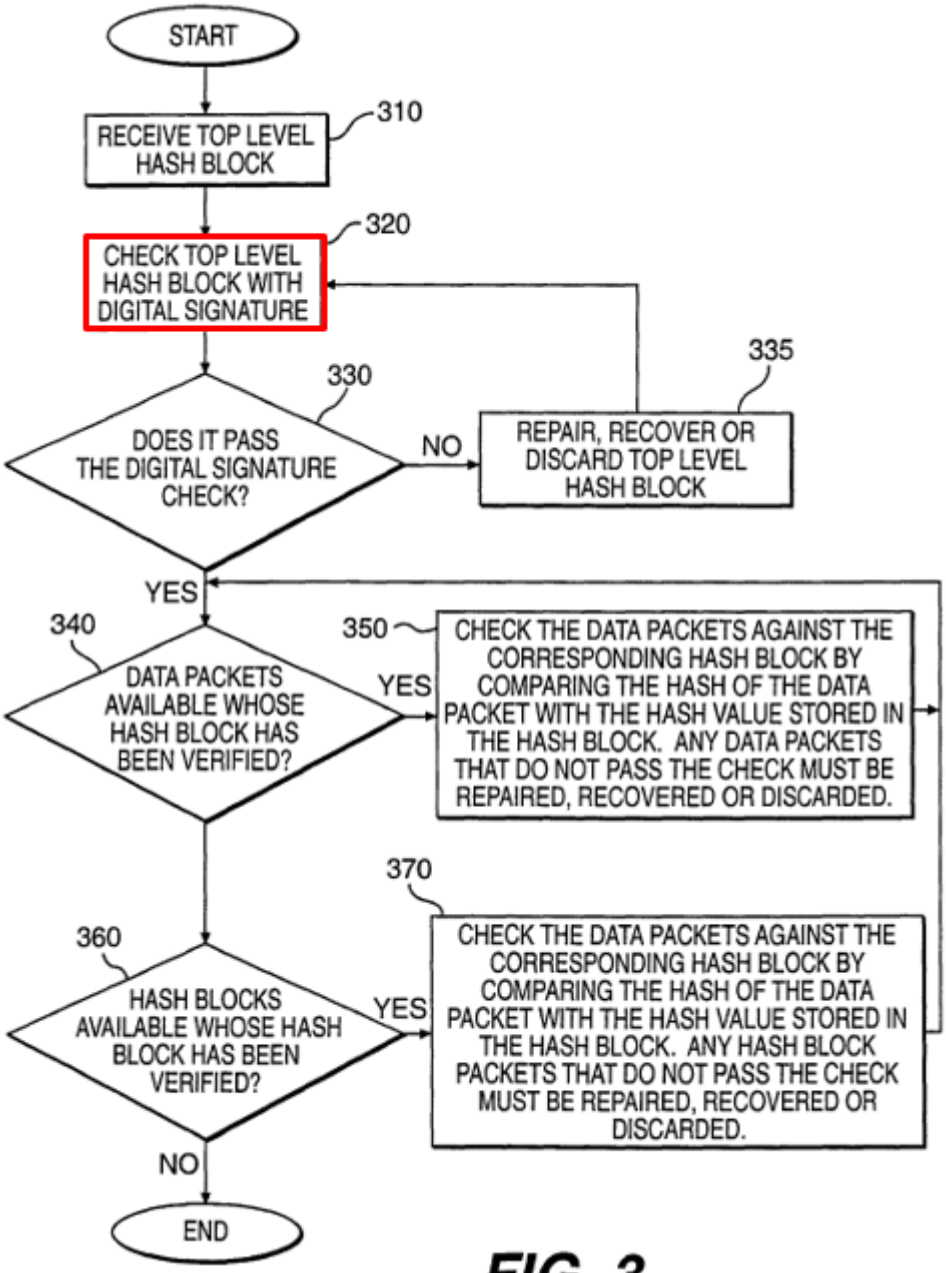



FIG. 3

**107.** Figure 3 is a flowchart that illustrates the steps of receiving and verifying the data. In step 310, the hash block containing the digital signature and the hash values are received. In step 320, Hanna discloses "check[ing] top level block with digital signature." A POSITA would understand this step to disclose verifying the hash

values in the hash block using the digital signature also provided in the hash block. (Ex. 1006, 11:12-13); *see also supra* ¶¶ 53-57.

**e) [42d] "verifying the authenticity of at least a portion of the file using the at least one check value"**

**108.** **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).

**109.** After the "hash values" in the hash block are verified using the digital signature, Hanna discloses using the verified hash values to verify the corresponding data portions of the file. For example, in step 350 of Figure 3, Hanna discloses "[c]heck[ing] the data packets against the corresponding hash block by comparing the hash of the data packet with the hash value stored in the hash block":

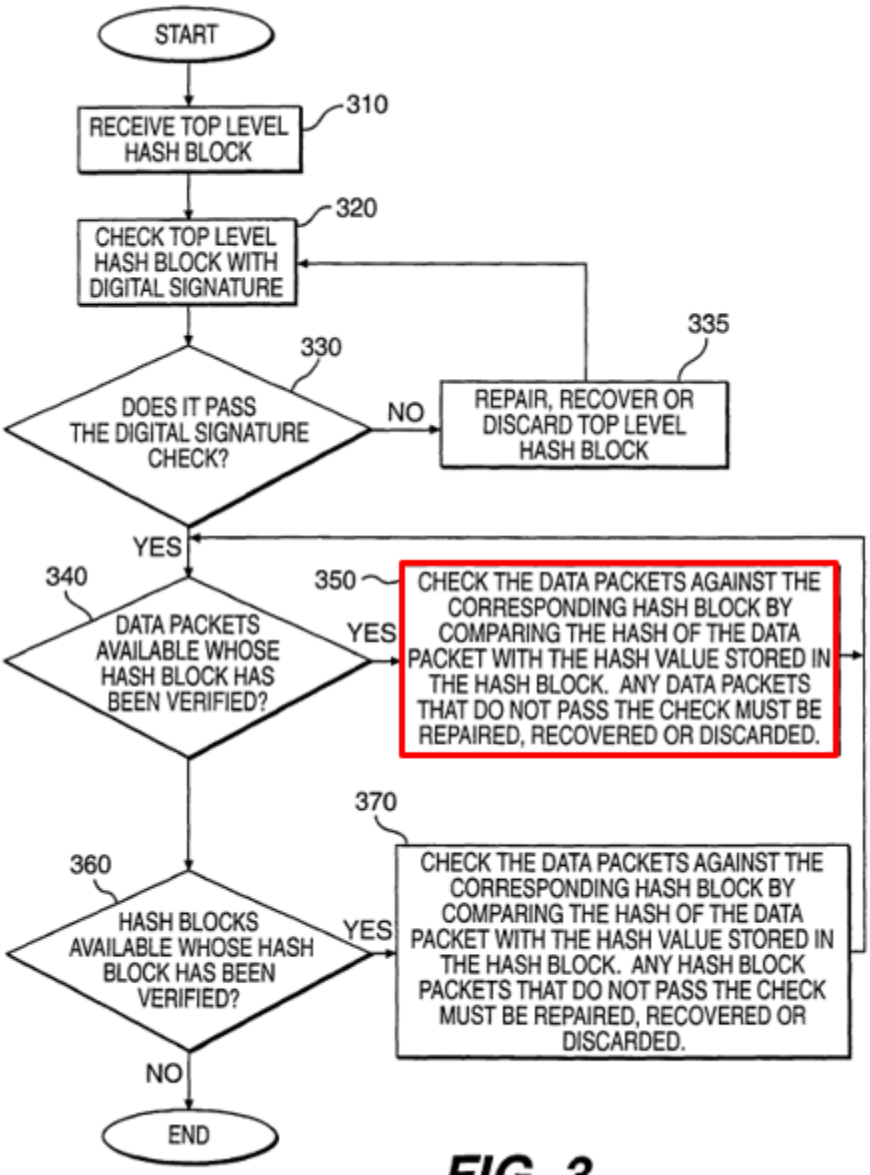



FIG. 3

**110.** Hanna provides an example of such verification in the specification. It describes first applying "[a] hash function" to each portion of the file, *i.e.*, data packet (*e.g.*, $A_{10}$, $A_{11}$, $A_{12}$), and then comparing the resulting hash with the hash value (*e.g.*, $B_4$) in the hash block in the level about it. (Ex. 1006, 11:24-27). A POSITA would have understood that the data packets, which are portions of the data file, are

verified using the hash values in the hash block that have been previously verified using the digital signature. *See supra* ¶¶ 53-57, 105-107.

**f) [42e] "granting the request to use the file in the predefined manner"**

**111. 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). This is shown in step 350 of Figure 3:

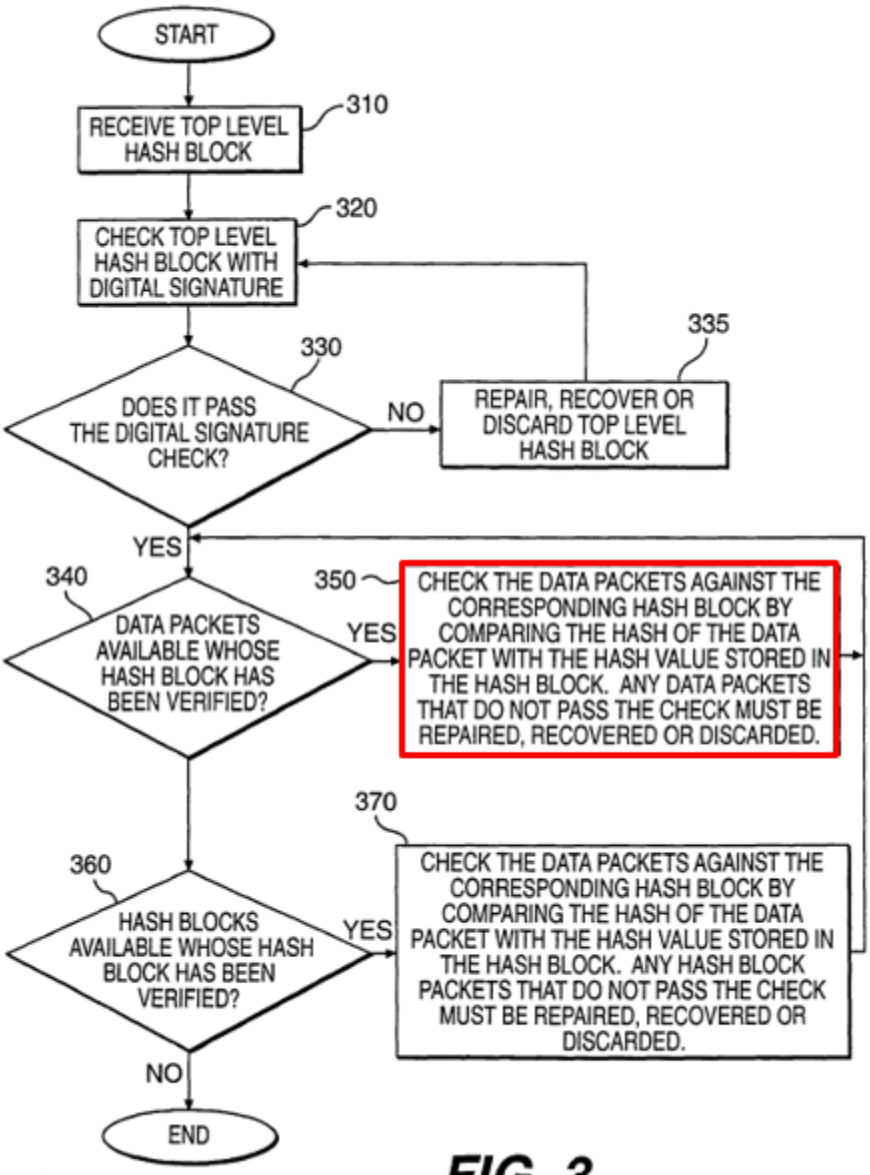



FIG. 3

**112.** Based on this disclosure, a POSITA would have understood that the verified data packets are authenticated for the use for which they were transmitted; otherwise, as disclosed in step 350, they would have been "repaired, recovered, or discarded." Because Hanna's system is designed to combat data corruption (Ex. 1006, 1:6-8, 3:6-13), it would be obvious to a POSITA to allow or prevent a requested use of a file depending on whether it passes the authentication process because corrupted

data, or data not properly signed with a recognized/authenticated signature, would be undesirable to be used. *See supra* ¶¶ 58-60.

**113.** To the extent it may be argued that granting a request to use a file in a predefined manner is not already disclosed in Hanna, **Stefik** discloses granting the request to use the file in the predefined manner.

**114.** Stefik discloses at least two embodiments of granting a request to use the file in the predefined manner. First, Stefik discloses granting a request simply by servicing the request. When a user requests to access a digital work using the interface of a repository, Stefik discloses "initiat[ing] suitable transactions to service the request." (Ex. 1007, 16:61-67). A POSITA would have understood that taking steps to service the request is granting the request. *See supra* ¶ 60.

**115.** Second, Stefik discloses granting a request from a repository to use an electronic file for a stated purpose. As discussed in [42a] above, the stated purpose corresponds to a specific usage right. For example, when a rendering repository makes a request to install a player, the repository receiving this request provides a copy of the player software to be installed. (Ex. 1007, 7:19-33, 20:48-54, 42:26-31). A POSITA would have understood that providing a copy of the player software to be installed is granting a request to install the player. As another example, after the copied software has been verified to be authenticated using the tamper-checking code, the installation process is permitted to continue following the installation

instructions. Stefik discloses that only software that has been verified is allowed to proceed with the installation process. Stefik explains that "this step assures that the contents of the software, including the various scripts, have not been tampered with." (Ex. 1006, 42:53-55). A POSITA would have understood that permitting the installation to proceed is granting a request to install the player. (Ex. 1007. 42:49-43:2); *see also supra* ¶ 60.

**5. Claim 43**

**a) [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"**

**116.** **Hanna** discloses that 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").

**117.** As discussed above, the "hash block" comprises one or more hash values. And such hash values were explicitly disclosed to verify the data in the file. For example, Hanna discloses that "the data is checked against the hash block containing the hashes of the data set." (Ex. 1006, 4:26-27). As another example, Hanna discloses that "the data packet $A_{10}$, $A_{11}$, $A_{12}$ would be checked by comparing the hash of the data packet 500d with the value $B_4$." (Ex. 1006, 11:24-27).

**b) [43a] "hashing at least a portion of the file to obtain a first hash value"**

**118.** **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.

**119.** Hanna teaches applying "a hash function" to a data packet to "comput[e] the hash of the packet." (Ex. 1006, 10:1-3, 11:14-15).

**c) [43b] "comparing the first hash value to at least one of the one or more hash values"**

**120.** **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").

**121.** Hanna teaches comparing two hash values together. The first hash value is the hash of the packet computed by applying a hash function to the data packet, as discussed above. (Ex. 1006, 10:1-3, 11:14-15). The second hash value(s), which is "the one or more hash values," are the hashes in the received hash block. Hanna discloses comparing "the hash of the packet" with "the hash value stored in the hash block." (Ex. 1006, 10:1-3).

**6. Claim Charts**

**122.** I have reviewed, had input to, and endorse as described below the claim charts of the accompanying Petition showing that each element of Challenged Claims 42 and 43 of the '815 patent are met by Hanna in view of Stefik, and that, as explained

above, a POSITA would have been motivated to combine the teachings of the references as claimed. Any similarities between the claim charts below and those in the accompanying Petition are intentional.

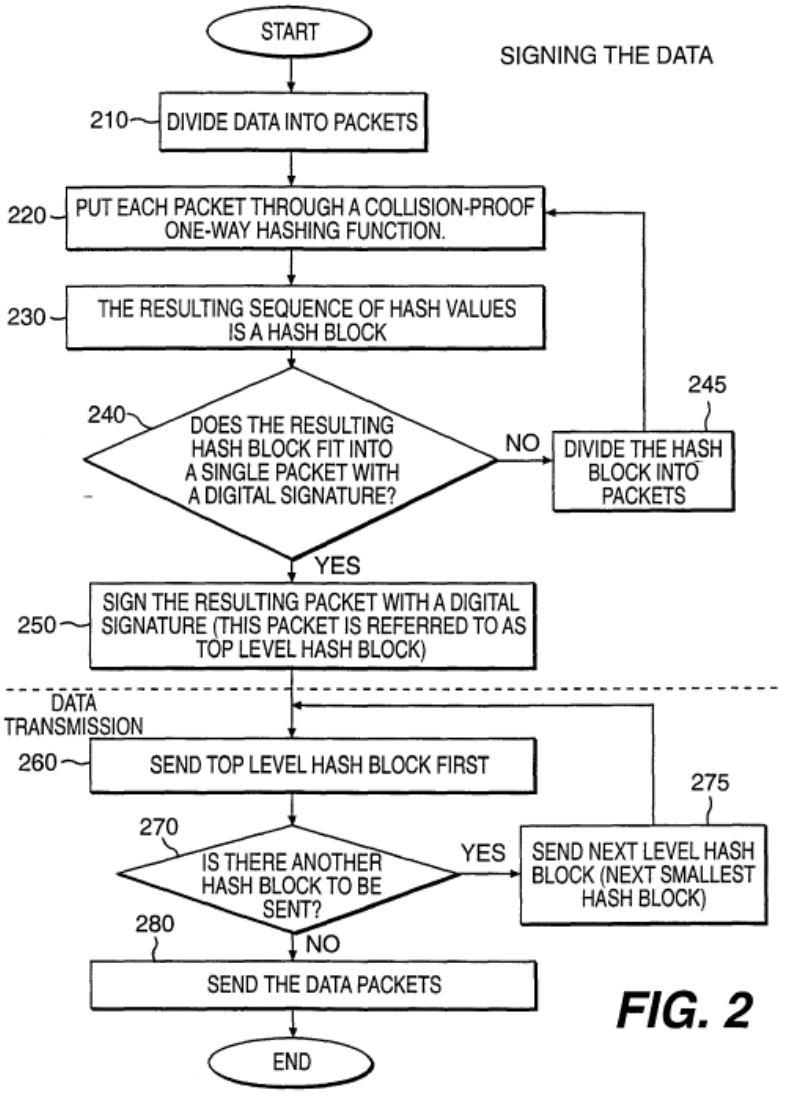
| **Claim 42** | **Hanna in view of Stefik** |
|---|---|
| [42pre][1] A method for managing at least one use of a file of electronic data, the method including: | **Hanna** discloses a method for managing at least one use (e.g., "cop[ying]", "download[ing]", "retriev[ing]", "receiving", "execut[ing]", 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).<br><br>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).<br><br>• "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)<br>• "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)<br>• *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 |

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

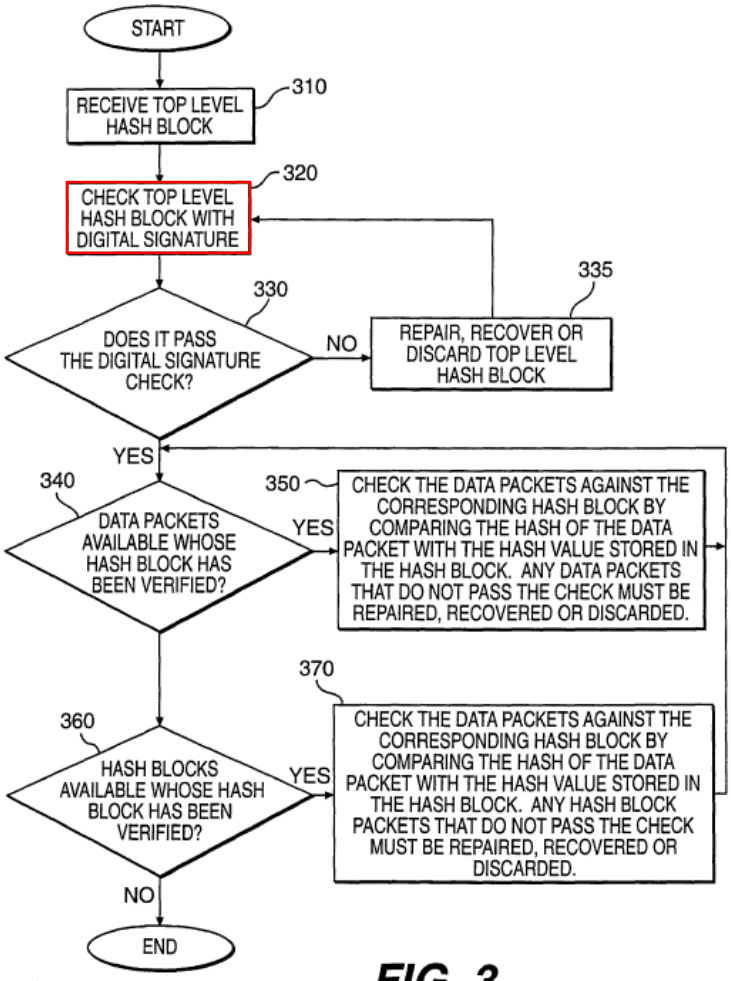
| Claim 42 | Hanna in view of Stefik |
|---|---|
| | transmission of information" in "video conferencing." (Ex. 1006, 1:10-15). A POSITA would recognize that in each case a request to transmit, copy and/or store files of electronic data is made. (Ex. 1002, ¶¶97-98).<br><br>To the extent it is argued 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).<br><br>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).<br><br>Such request by the rendering repository is to use a file in a predefined manner. Specifically, the request 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 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). |

| Claim 42 | Hanna in view of Stefik |
|---|---|
| | • "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)<br>• "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).<br>• "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)<br>• "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 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)<br>• "***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) |

| Claim 42 | Hanna in view of Stefik |
|---|---|
| | • "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)<br>• *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).<br><br>• "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 hash values. This sequence is called a hash block*** (step 230)." (Ex. 1006, 8:14-17)<br>• "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)<br>• "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 |

| Claim 42 | Hanna in view of Stefik |
|---|---|
| | applied to the top level packet is sufficient to verify all of the data." (*Id.*, 9:5-8)<br>• "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)<br>• "To verify data, the digital signature on the top level packet must be verified first (step 320)." (*Id.*, 9:23-24)<br>  <br><br>**FIG. 2**<br>• *See also, e.g.*, Ex. 1006, Figs. 3-5; Ex. 1002, ¶¶51-57. |

| Claim 42 | Hanna in view of Stefik |
|---|---|
| [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).<br><br>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).<br><br>• "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)<br><br>The top level hash block is checked with the digital signature (step 320 in Fig. 3). |

| Claim 42 | Hanna in view of Stefik |
|---|---|
| |  <br>***FIG. 3***<br>(annotated with a red box); (Ex. 1002, ¶107).<br>• "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)<br>• "***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)<br>• *See also, e.g.*, Ex. 1006, Fig. 5, 11:12-19; Ex. 1002, ¶54. |
| [42d] verifying the authenticity | **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 |

| Claim 42 | Hanna in view of Stefik |
|---|---|
| of at least a portion of the file using the at least one check value; and | one check value (*e.g.*, "hash values" in a hash block). (Ex. 1002, ¶108).<br><br>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).<br><br>• "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)<br>• "Finally, the data is checked against the hash block containing the hashes of the data set." (*Id.*, 4:26-27)<br>• "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)<br>• "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)<br>• *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. |
| [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 |

| Claim 42 | Hanna in view of Stefik |
|---|---|
| | prevent a requested use of a file depending on whether it passes the authentication process. (Ex. 1002, ¶¶111-112).<br><br>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).<br><br>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).<br><br>• "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 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) |

| Claim 42 | Hanna in view of Stefik |
|---|---|
| | • "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).<br>• "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)<br>• "***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)<br>• "determining if ***a request for access*** to a digital work stored in said storage means ***may be granted***" (*Id.*, 52:35-37)<br>• "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)<br>• *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 | **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). |

| Claim 43 | Hanna in view of Stefik |
|---|---|
| authenticity of at least a portion of the file using the at least one check value includes: | • "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)<br>• "Finally, the data is checked against ***the hash block containing the hashes of the data set***." (*Id.*, 4:26-27)<br>• "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)<br>• "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)<br>• *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).<br>• "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)<br>• "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)." (*Id.*, 10:1-3)<br>• "***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) |

| Claim 43 | Hanna in view of Stefik |
|---|---|
| | • "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)<br>• *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).<br>• "Finally, the data is checked against the hash block containing the hashes of the data set." (Ex. 1006, 4:26-27)<br>• "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)<br>• "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)<br>• *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

**123.** It is my opinion that the Challenged Claims 42 and 43 are rendered obvious by **Gennaro**. Below, I provide an overview of Gennaro. I further provide an

analysis of claim 42, followed by claim 43, and a claim chart of relevant prior art disclosures to explain how the limitations of claim 42 and 43 are met.

### 1. Overview of Gennaro

**124.** 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"), entitled "How to Sign Digital Streams," which was filed on February 13, 1997.

**125.** 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). This technique applies to both finite and "potentially infinite" digital streams. When a digital stream is finite, it is known in its entirety to a sender of the stream, such as a digital movie; and when a digital stream is "potentially infinite," then it is not known in advance, such as a live feed. (Ex. 1008, 2:64-3:10).

**126.** Gennaro describes a prior art solution for the finite streams. This solution requires sending both a digital signature and a hash of the data to a receiver, just as recited in claim 42 of the '815 patent. Several steps are performed when a sender wishes to send a data stream to a receiver: the sender first splits the data stream into blocks; the sender then hashes each block and puts the hashes into a table; and last, the sender signs the table. When the receiver asks for the data stream, the sender

sends the signed table of hashes, and then the data stream. The receiver verifies the signature, which in turn verifies that the hashes in the signed table are authentic. The verified hash in the signed table is then used to verify its corresponding block. (Ex. 1008, 1:23-34).

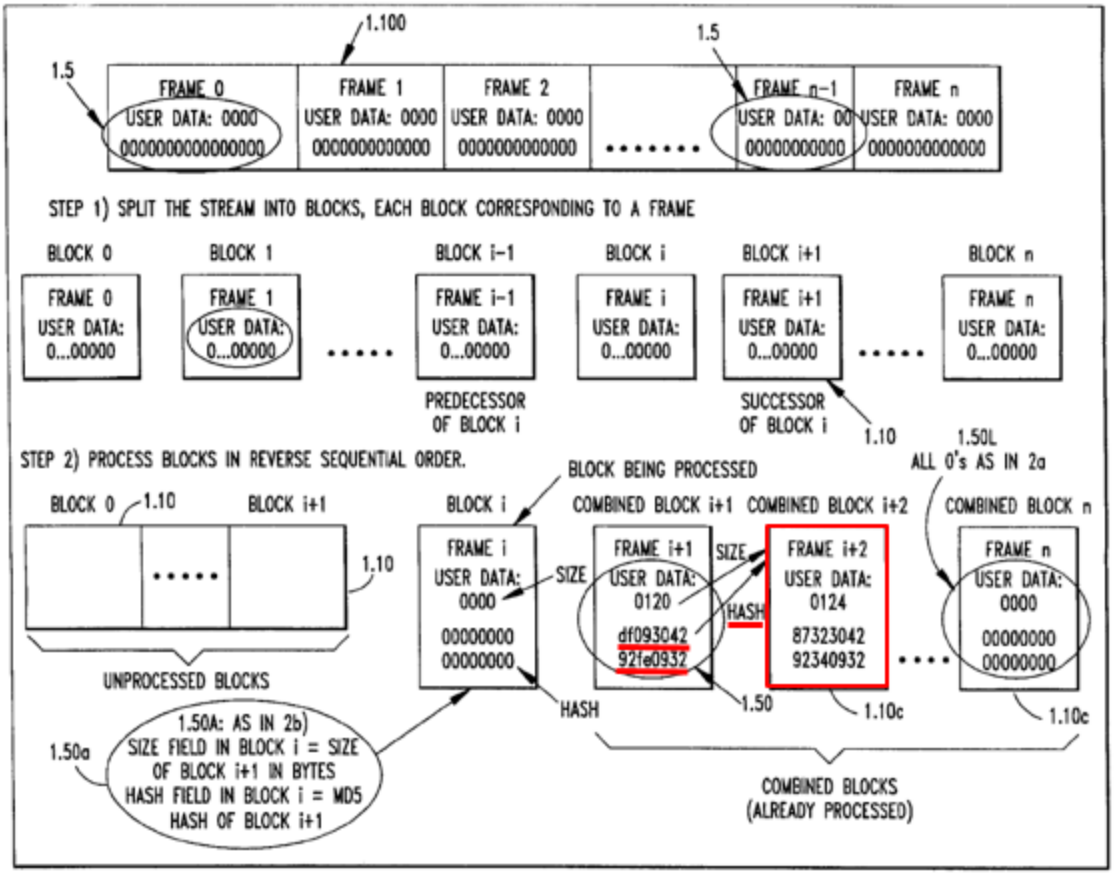
**127.** Gennaro presents an alternative way of signing and authenticating a data stream. This approach allows the receiver to consume the authenticated data from the stream at the same rate it receives the data from the stream. (Ex. 1008, 2:29-41). Gennaro explains:

> 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).

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

FIG.1A




As shown in Figure 1A, in step 1, a digital stream (1.100) is split into blocks that are represented as 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).

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

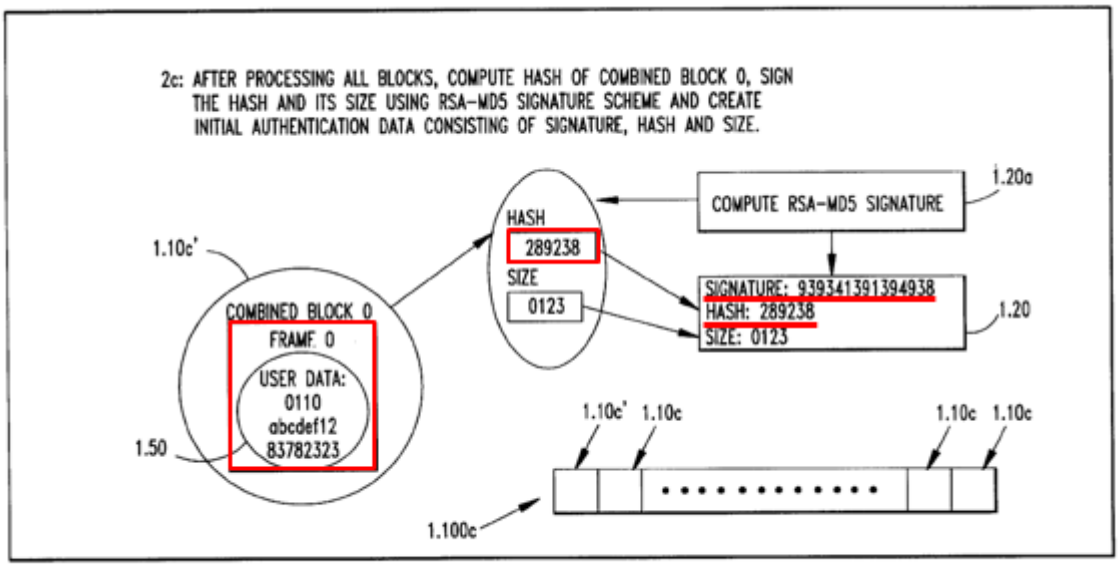


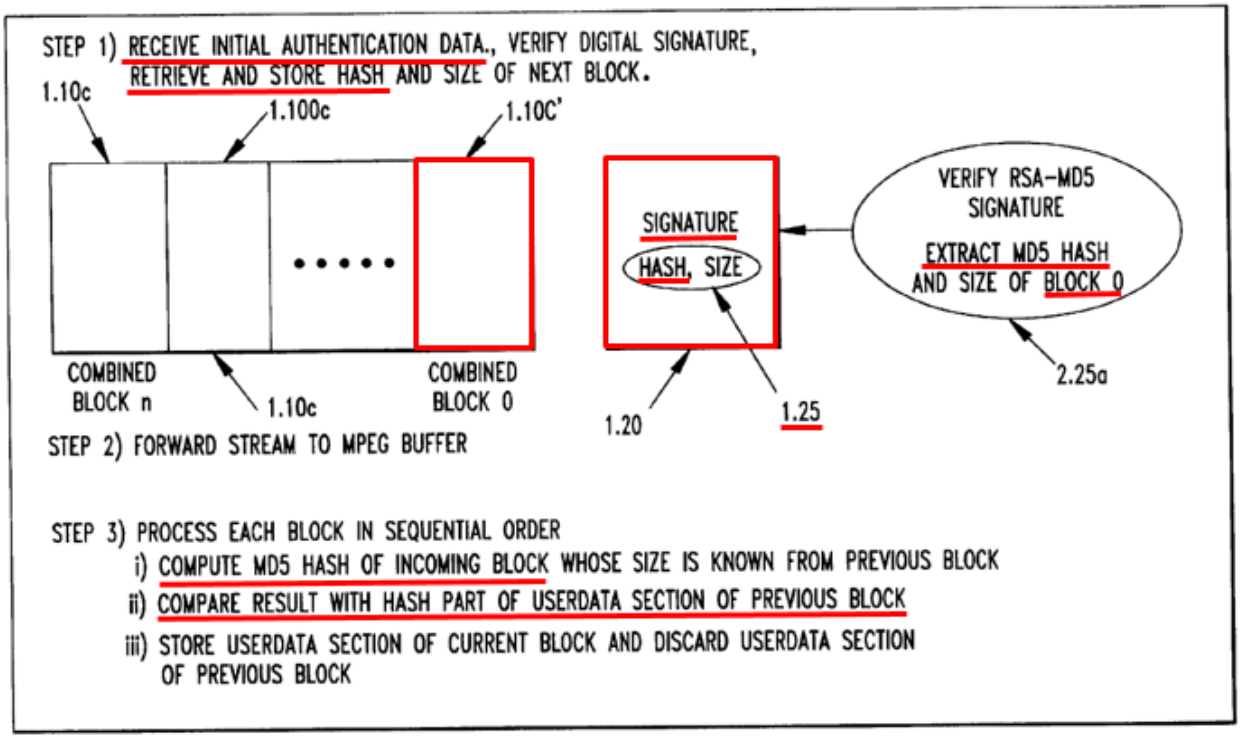

FIG.1B

In Figure 1B, the first combined block (referred to as the "combined distinguished block (1.10c')" in the specification (Ex. 1008, 4:52-62)), is depicted as the "Combined Block 0." A hash of the first combined block is computed, and is depicted as having the value of "289238" in the example shown in Figure 1B. In the specification, this hash is referred to as "hash 1.25." (Ex. 1008, 4:59-61). 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)," as shown in Figure 1B. (Ex. 1008, 4:59-65).

**130.** 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). (Ex. 1008, 4:67-5:2). The color-

annotated Figure 2A in Gennaro 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'). (Ex. 1008, 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 it is used to verify the authenticity of hash 1.25. (Ex. 1008, 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'). (Ex. 1008, 5:30-34).

**131.** In step 2, the receiver receives the combined stream (1.100c). The combined stream (1.100c) is passed to an authentication software at the receiver. The function of the authentication software is to authenticate data before passing them to the MPEG processing software for further processing, such as converting the data back to video using the MPEG standard. (Ex. 1008, 5:8-14). 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).

**132.** 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. (Ex. 1008, 5:48-53).

**133.** 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. (Ex. 1008, 5:54-57).

**134.** 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. (Ex. 1008, 5:57-62).

**135.** If the hashes are different, then tampering has been detected and the processing of the first combined block would be halted. (Ex. 1008, 5:57-62).

**136.** 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. (Ex. 1008, 5:63-65).

### 2. Claim 42

#### a) [42pre] "A method for managing at least one use of a file of electronic data"

**137. Gennaro** discloses a method for managing at least one use (e.g., "play," "consume," or "convert[] back to video") of a file of electronic data (e.g., "digital stream").

**138.** For example, Gennaro discloses "play[ing]" a digital stream in real time. (Ex. 1008, 2:50-54). Gennaro further discloses "consum[ing] the stream" after authenticating it using its disclosed method. (Ex. 1008, Abstract). Gennaro further discloses using the MPEG processing software to convert the data stream back to video using the MPEG standard. (Ex. 1008, 5:8-12). The digital stream can be any stream of digital data, such as digital movies, digital sounds, or applets. (Ex. 1008, 2:64-67). A POSITA would understand from these disclosures as describing using a file of electronic data. *See supra* ¶¶ 49-50, 58-60.

#### b) [42a] "receiving a request to use the file in a predefined manner"

**139. Gennaro** discloses receiving a request to use the file in a predefined manner. Gennaro discloses multiple requests to use a file in a predefined manner.

**140.** In one example, Gennaro discloses rendering or playing "internet applications (digital movies, digital sounds, applets)." (Ex. 1008, 2:64-67). A POSITA would have understood such action as a user requesting to use a digital file in a predefined manner (*e.g.*, play). *See supra* ¶¶ 58-59.

**141.** In another example, Gennaro discloses that when a user requests to use data, a receiver "asks for" the transmission of the data from a sender. (Ex. 1008, 1:28-29). A POSITA would have understood that such ask is a request to use the data in a predefined manner, and that such request is received by the sender. It would have been obvious to a POSITA that the receiver would make similar requests for all transmission of an electronic file from the sender. *See supra* ¶¶ 58-59.

**142.** As another example, Gennaro discloses passing the data from a sender to an "authentication software" at the receiver. An "authentication software" can be added to any existing MPEG software to control the data flow to the MPEG processing software. (Ex. 1008, 5:4-12). The authentication software controls when and how the MPEG software is informed that it has data in its buffer to be processed. The "authentication software" ensures only blocks that have been authenticated would be passed to the MPEG software for processing. (Ex. 1008, 5:4-12). When a user requests to use a file (*e.g.*, to play or consume it), the data in the file is first passed from the sender to the "authentication software" at the receiver. And only after the "authentication software" has authenticated the data, such data would be passed to

the MPEG software for processing, *e.g.*, converting the data back to video using the MPEG standard. (Ex. 1008, 5:40-67). Because the authentication software in the receiver controls the data flow to the MPEG software (Ex. 1008, 5:8-12), it would have been obvious to a POSITA to configure the receiver such that the receiver passes the data to the authentication software as a request for the MPEG software to convert the data into video for playback; and such request would subsequently be granted by the authentication software once the data has been authenticated. *See supra* ¶¶ 58-59.

**c) [42b] "retrieving at least one digital signature and at least one check value associated with the file"**

**143.** **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").

**144.** Gennaro describes two embodiments that disclose retrieving both a digital signature and hash value.

**145.** In the first embodiment, 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. 1008, 4:59-5:2, 5:23-29).

**146.** Color-annotated Figure 1B of Gennaro shows the content of "initial authentication data (1.20)" as "consisting of signature [and] hash":

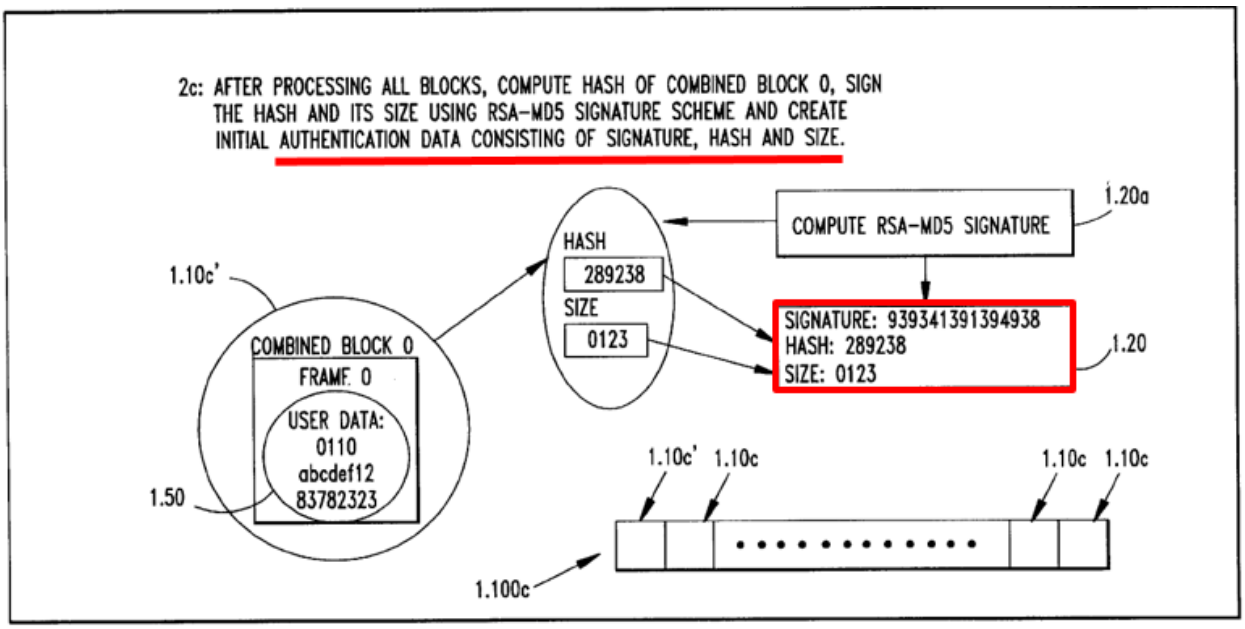
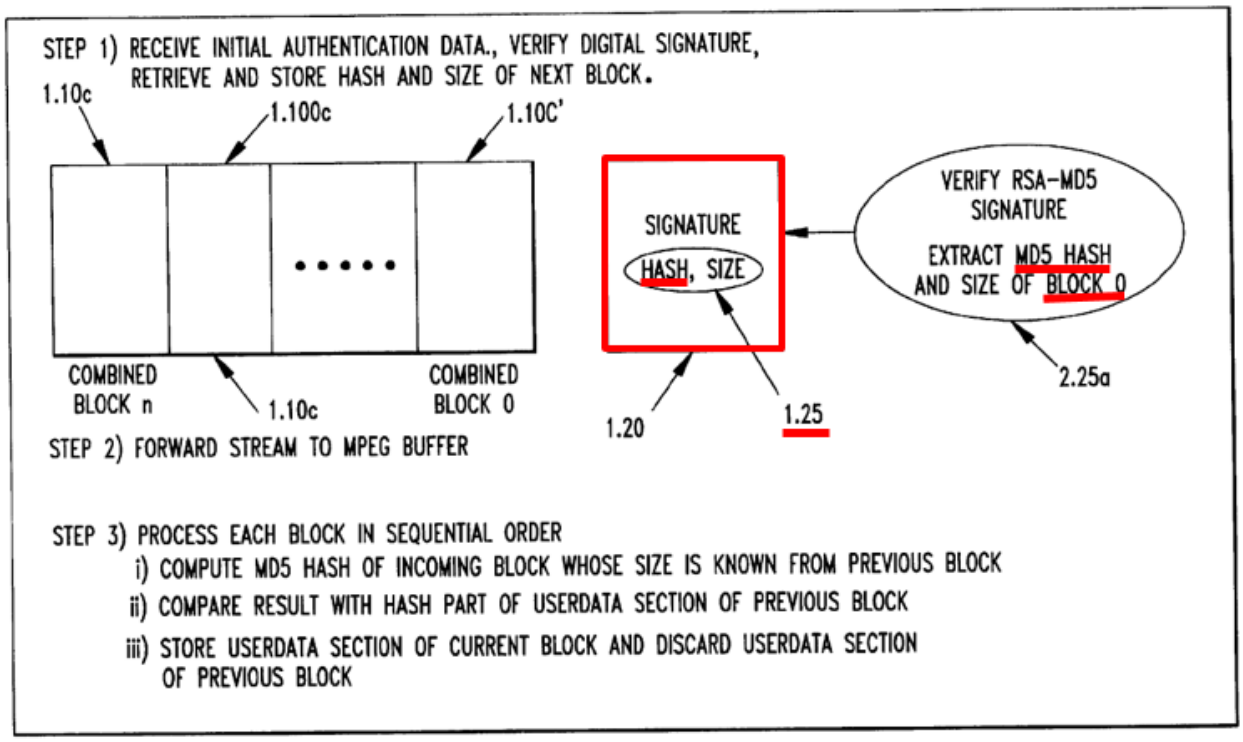



FIG. 1B

FIG. 1 FIG.1A FIG.1B

**147.** The hash value in the initial authentication data (1.20) is referred to as "hash 1.25" in the color-annotated Figure 2A of Gennaro:

FIG.2A




The specification explicitly discloses that "the receiver receives the initial authentication data (1.20) consisting of a digital signature on the hash of the first combined block (1.10c')…, together with the purported values of the hash…" (Ex. 1008, 4:59-5:2).

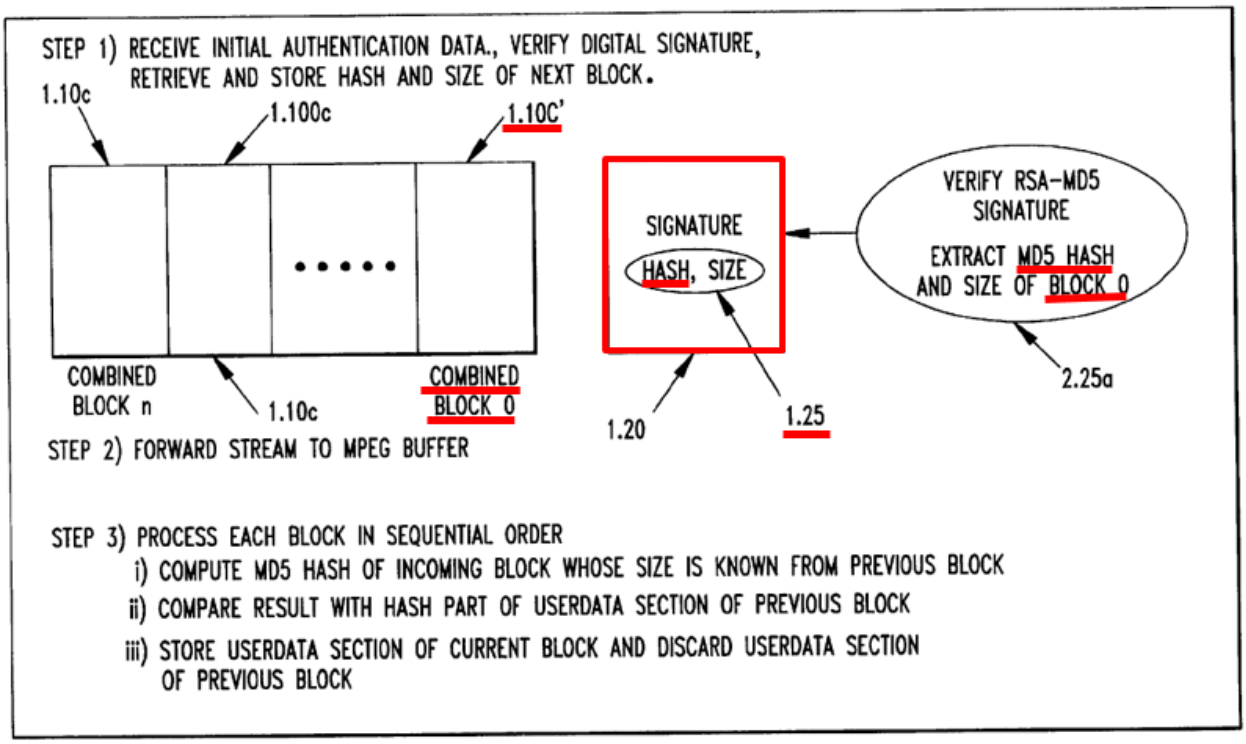
**148.** The prior art embodiment described in Gennaro also discloses retrieving a digital signature and at least a hash associated with the file. Specifically, Gennaro describes retrieving a signature and a table of hashes from the sender. (Ex. 1008, 1:24-29). The hashes in the table are hashes of each data block of a file, and are thus associated with the file.

**d) [42c] "verifying the authenticity of the at least one check value using the digital signature"**

**149.** **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").

**150.** Color-annotated 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




In connection with Figure 2A, the specification explains that if the signature is verified, then the hash of the first combined block (1.10c')(also referred to as hash 1.25) is verified to be authentic. (Ex. 1008, 5:23-34).

**151.** Figure 1B of Gennaro shows the relationship between the combined block 0 (1.10c'), hash 1.25, and the signature:

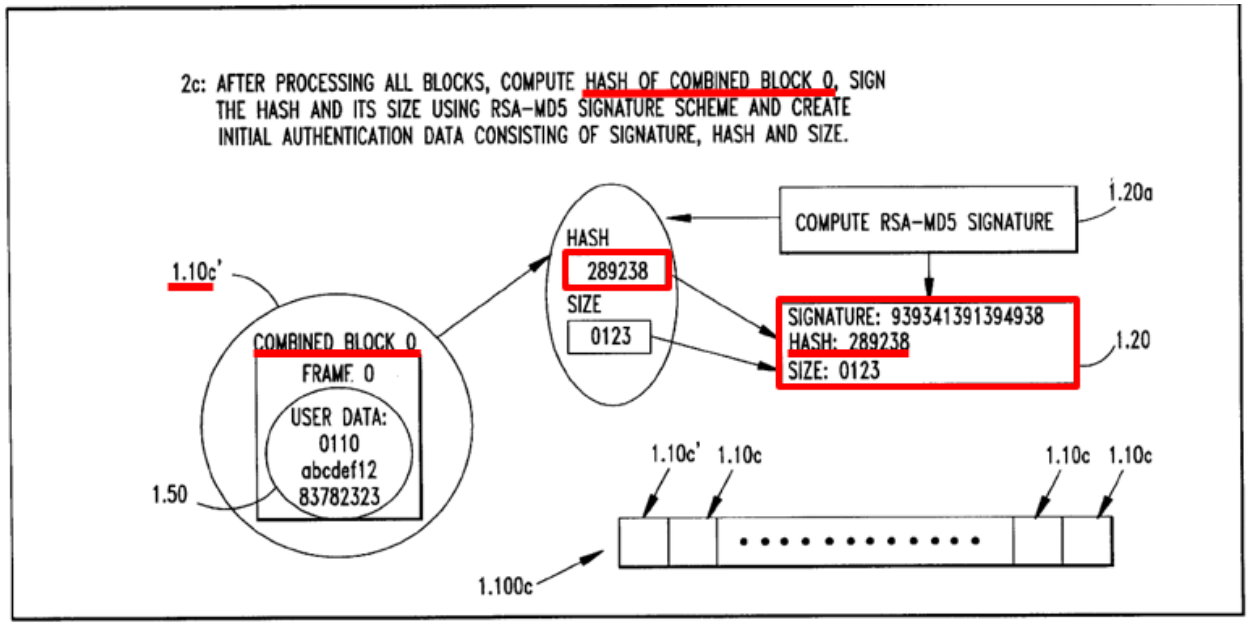



FIG. 1B

FIG. 1 | FIG.1A / FIG.1B

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. A POSITA would have understood that because the signature was created by signing the hash 1.25, the hash 1.25 could be verified using the signature.

**152.** In the prior art embodiment, Gennaro discloses that, "[t]he 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."

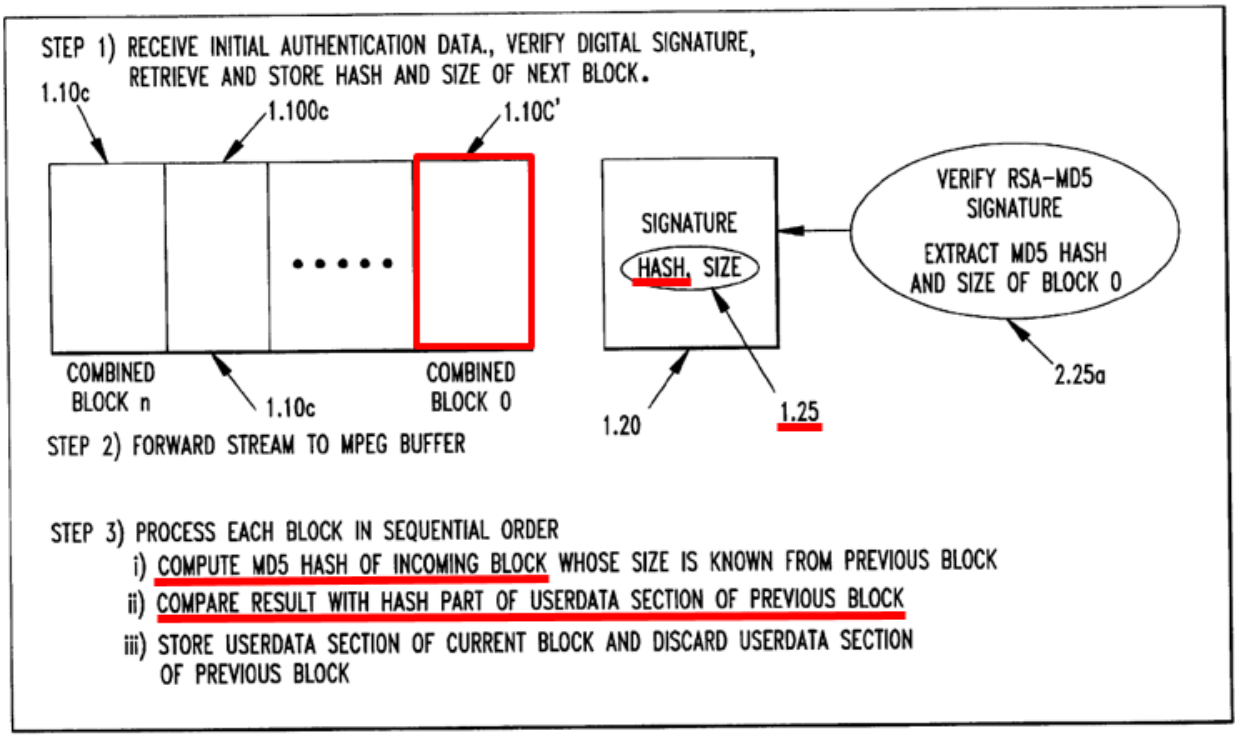
(Ex. 1008, 1:29-33). Because the hashes in the signed table are verified once the signature is verified, a POSITA would have understood that the hashes in the signed table are verified using the signature after the signature has been verified. *See supra* ¶¶ 53-57.

**e) [42d] "verifying the authenticity of at least a portion of the file using the at least one check value"**

**153.** **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')").

**154.** Color-annotated 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.

FIG.2A




The verification 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.

**155.** In the prior art embodiment, the verified hashes are also used to verify the corresponding data blocks. Specifically, Gennaro discloses, "[t]he 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).

**f) [42e] "granting the request to use the file in the predefined manner"**

**156. Gennaro** discloses granting the request to use the file in the predefined manner. Gennaro discloses multiple ways in which such request is granted.

**157.** For example, Gennaro discloses the receiver plays a stream. (Ex. 1008, 2:54, 2:65-67). A POSITA would have understood such action as granting the user's request to play an internet application, such as a digital movie, sound or applet.

**158.** In another example, Gennaro discloses the receiver receives the data stream from the sender in response to "ask[ing] for" the data stream. (Ex. 1008, 1:28-29, 4:67-5:2). A POSITA would have understood that providing the data stream the receiver has asked for is granting its request to transmit the stream. *See supra* ¶¶ 58-60.

**159.** In another example, Gennaro discloses the authentication software passing the data stream to the MPEG software for further processing (*e.g.*, converting to video using the MPEG standard) after the data stream has been authenticated. (Ex. 1008, 5:8-12, 5:58-62). The MPEG software is initially prevented from consuming the data from the stream until it is authenticated. (Ex. 1008, 5:35-39). Once the authentication software authenticates the data, then the MPEG software is informed of the arrival of the data and can proceed with its processing. (Ex. 1008, 5:58-62). If the authentication process reveals that tampering has been detected, then the data processing is halted. (Ex. 1008, 5:57-58). A POSITA would have understood that permitting the data to be further processed by the MPEG software is granting a request from the receiver to pass the data stream to the MPEG software for further processing. *See supra* ¶¶ 58-60.

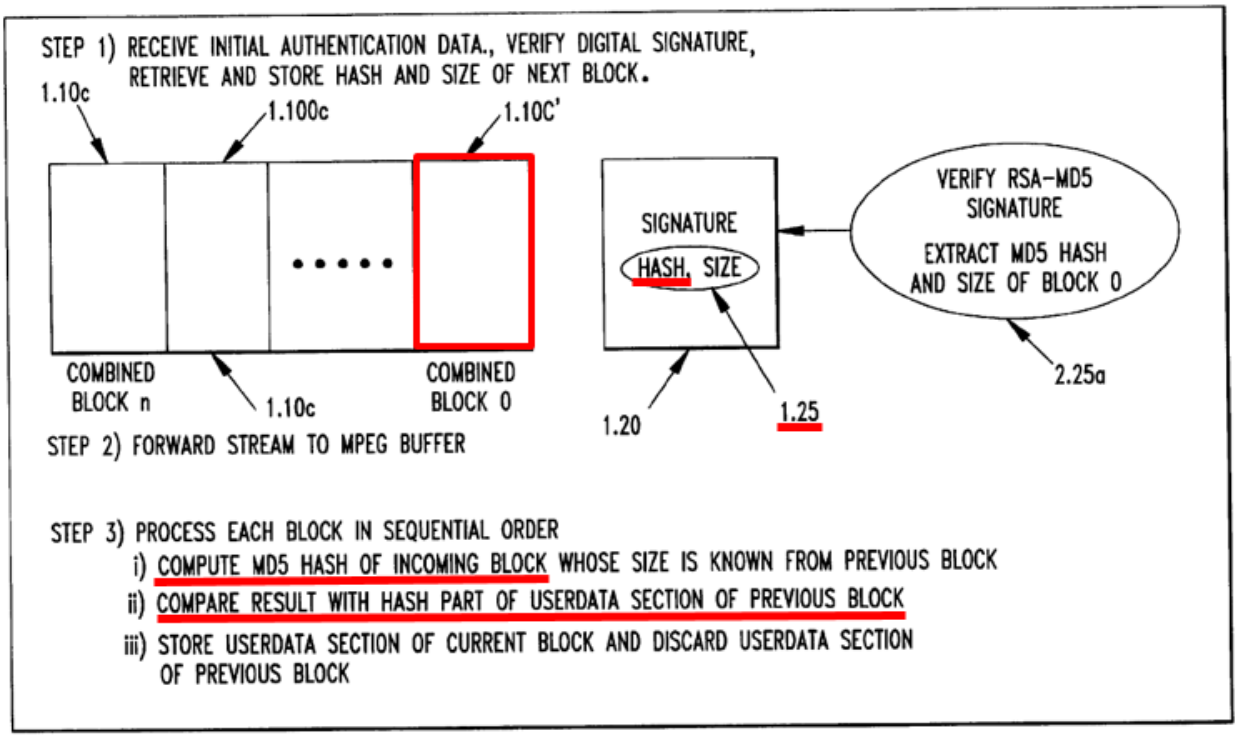
### 3. Claim 43

#### a) [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"

**160.** **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.

**161.** Color-annotated 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.

FIG.2A




The verification 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.

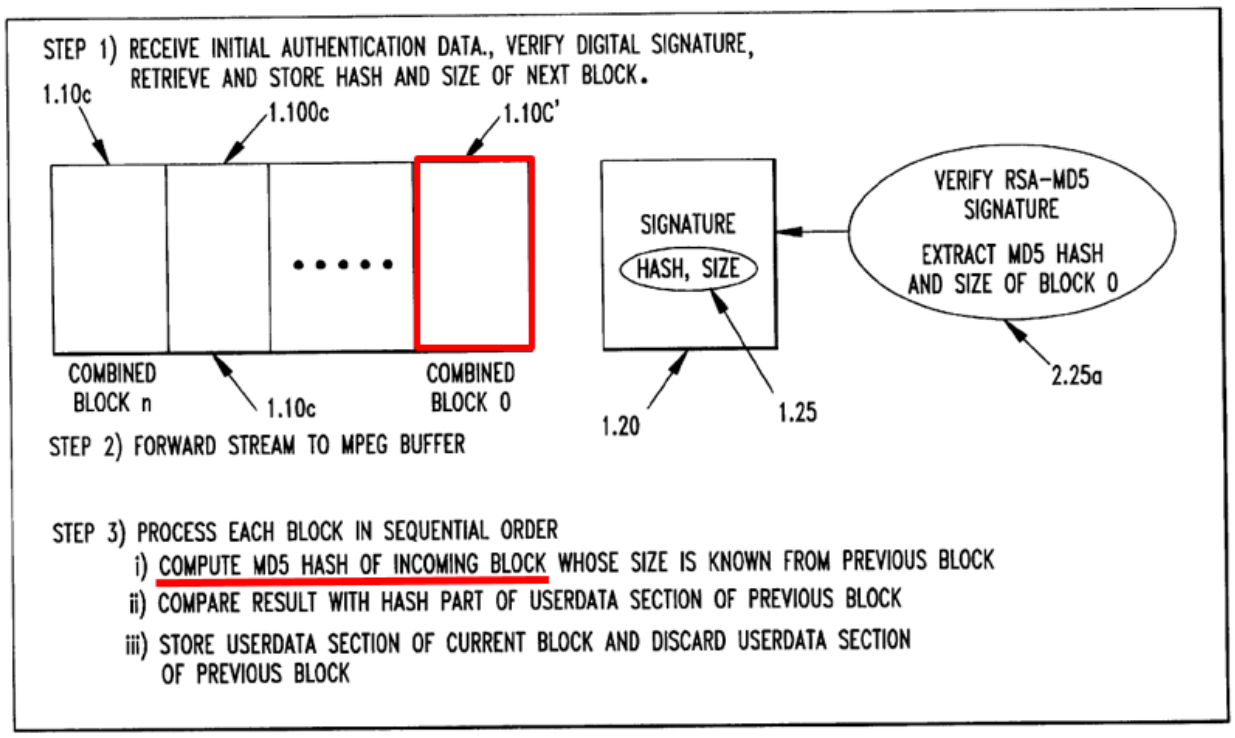
**162.** In the prior art embodiment, the verified hashes are also used to verify the corresponding data blocks. Specifically, Gennaro discloses, "[t]he receiver first receives and stores this table and verifies the signature on it. If the signature matches then the receiver has the cryptographic hash of each of the following stream blocks. Thus each block can be verified when it arrives." (Ex. 1008, 1:29-34).

**b) [43a] "hashing at least a portion of the file to obtain a first hash value"**

**163.** **Gennaro** discloses hashing at least a portion of the file to obtain a first hash value (*e.g.*, "compute MD5 hash of incoming block").

**164.** Color-annotated 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




(Ex. 1008, 5:48-53).

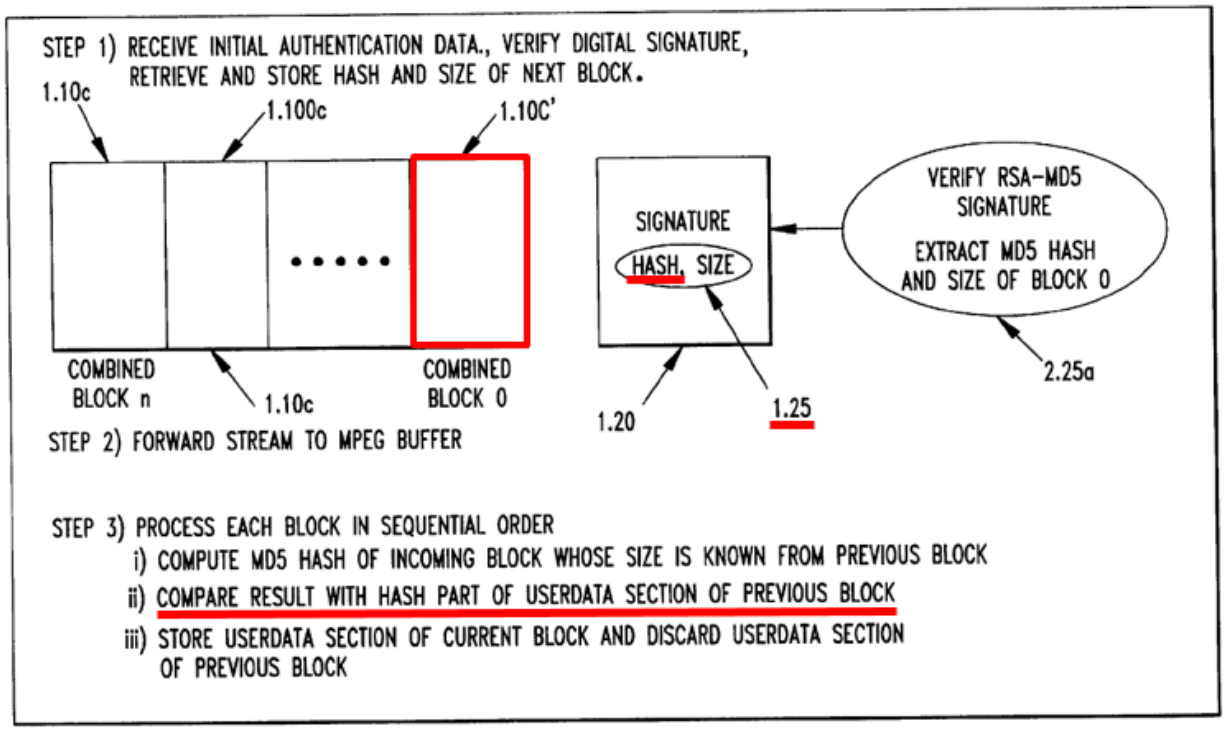
**c) [43b] "comparing the first hash value to at least one of the one or more hash values"**

**165.** **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')").

**166.** Color-annotated 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




(Ex. 1008, 5:54-57).

### 4. Claim Charts

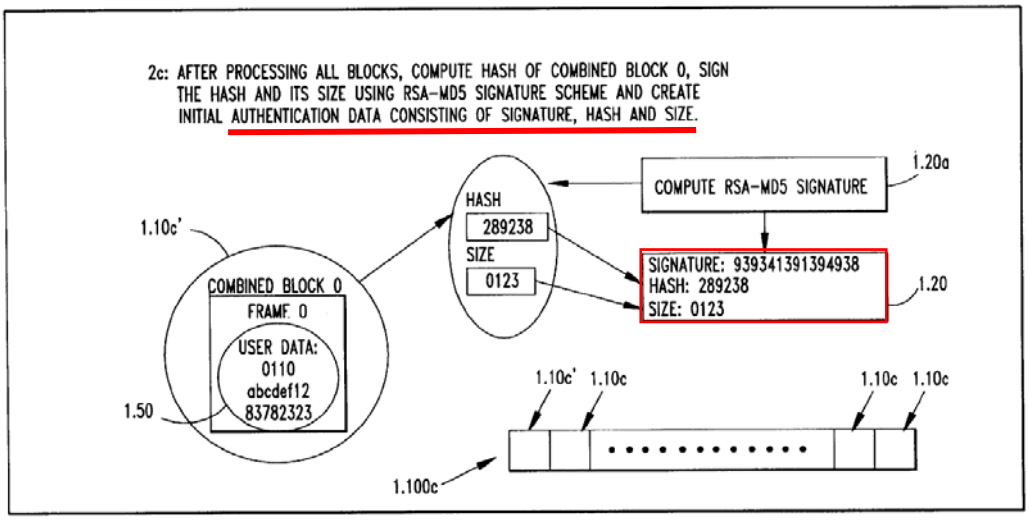
**167.** I have reviewed, had input to, and endorse as described below the claim charts of the accompanying Petition showing that each element of the Challenged Claims 42 and 43 of the ’815 patent are met by Gennaro, and that, as explained above, a POSITA would have been motivated to combine the teachings of the references as claimed. Any similarities between the claim charts below and those in the accompanying Petition are intentional.

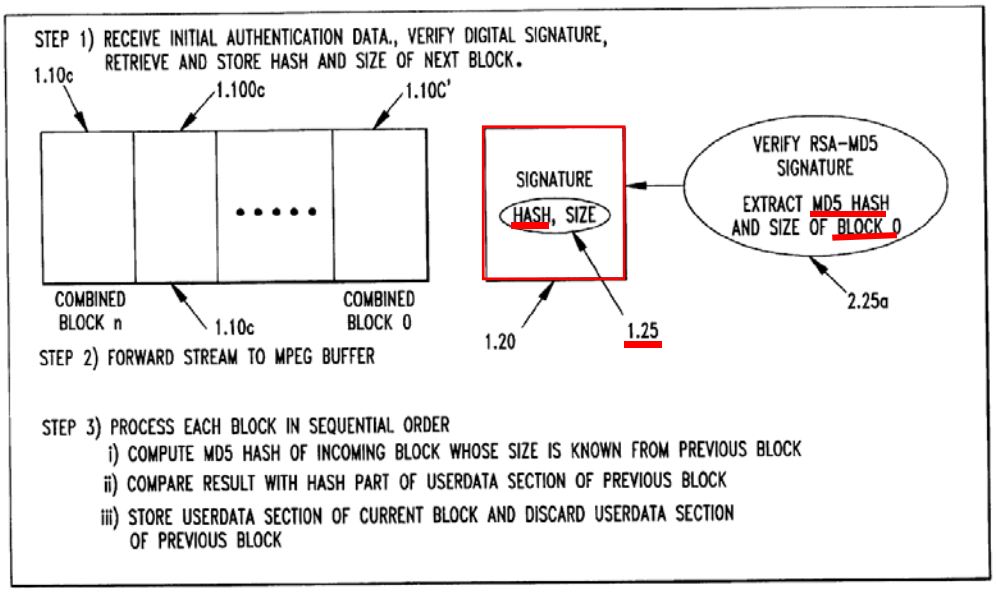
| Claim 42 | Gennaro |
|---|---|
| [42pre][2] A method for managing at least one use of a file of electronic data, the method including: | **Gennaro** discloses a method for managing at least one use (e.g., "play," "consume," "convert[] back to video") of a file of electronic data (e.g., "digital stream"). (Ex. 1002, ¶¶137-138).<br><br>• "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)<br>• "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)<br>• "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)<br>• "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)<br>• "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)<br>• *See also, e.g.*, Ex. 1008, 12:44-51; Ex. 1002, ¶¶49-50. |
| [42a] receiving a request to use the file in a predefined manner; | **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 "controls how the MPEG software/hardware is informed of incoming data"). (Ex. 1002, ¶139).<br><br>Gennaro discloses multiple requests to use a file in a predefined manner. For example, Gennaro discloses a user requesting to use a |

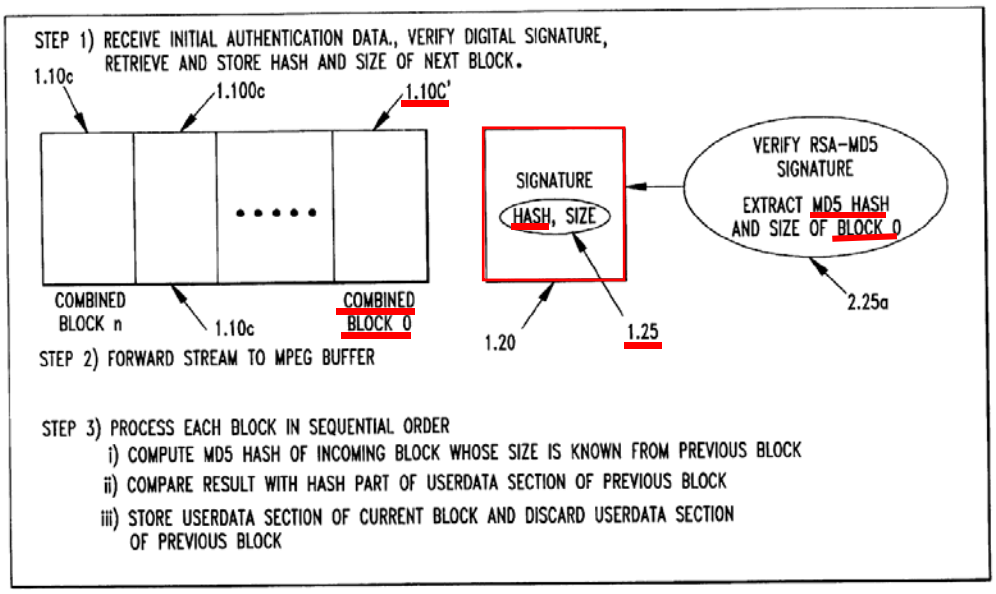
[2] To the extent they are limiting, the preambles of all Challenged Claims are nonetheless met by the art of record.

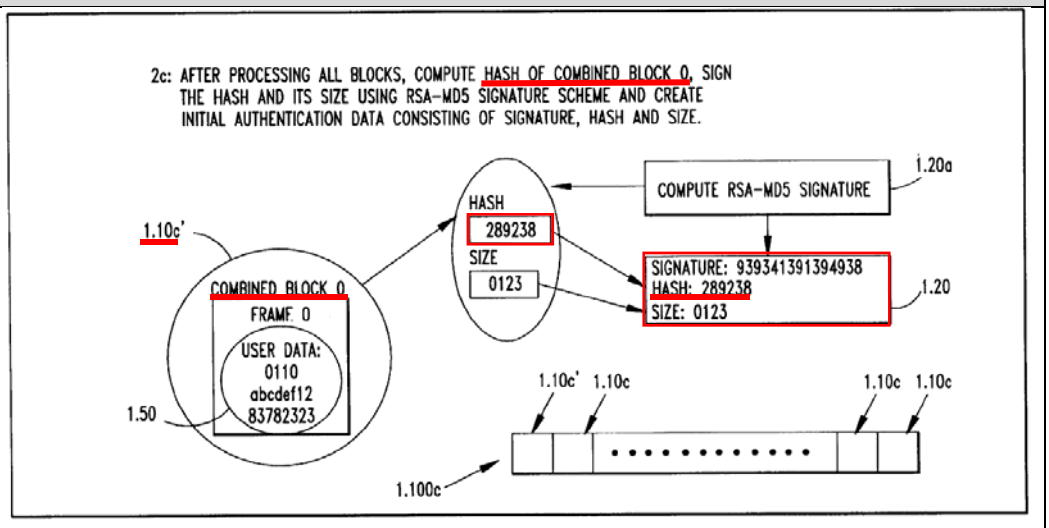
| Claim 42 | Gennaro |
|---|---|
| | file in a predefined manner (e.g., playing "internet applications (digital movies, digital sounds, applets)"). (Ex. 1008, 2:64-67; Ex. 1002, ¶140).<br><br>Gennaro further discloses that, when a user requests to play a file, a receiver requests transmission of the 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).<br><br>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).<br><br>• "When ***the receiver asks for the authenticated stream***, the sender first sends the signed table followed by the stream." (Ex. 1008, 1:28-29)<br>• "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*** |

| Claim 42 | Gennaro |
|---|---|
| | ***converted back to video using the MPEG standard***." (Ex. 1008, 5:4-14)<br>• "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).<br>• "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)<br>• *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).<br><br>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).<br>• "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 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)<br>• "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 |

| Claim 42 | Gennaro |
|---|---|
| | (1.10c') and the size of first combined block, ***together with the purported values of the hash*** and size." (Ex. 1008, 5:23-29)<br><br>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."<br><br>  <br><br><br>FIG.1B<br><br>FIG.1 FIG.1A FIG.1B<br><br>(color-annotated Fig. 1B of Gennaro)**;** (Ex. 1002, ¶146).<br><br>Figure 2A of Gennaro shows that the hash value in the initial authentication data (1.20) is referred to as "hash 1.25" |

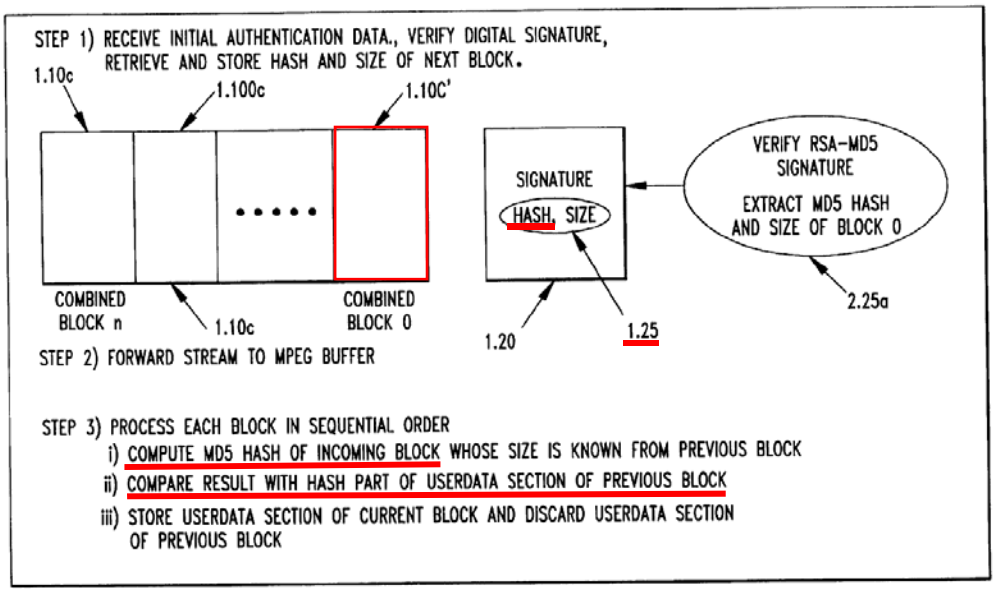
| Claim 42 | Gennaro |
|---|---|
| | FIG.2A<br>  <br><br>(color-annotated Fig. 2A of Gennaro); (Ex. 1002, ¶147).<br>In the prior art embodiment, Gennaro describes retrieving a signature and a table of hashes from the sender. (Ex. 1002, ¶148).<br>• "Instead of signing each block, the sender creates a 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).<br>• *See also, e.g.*, Ex. 1008, 4:34-48.<br>*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).<br>• "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)<br><br>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').<br><br>FIG.2A<br>  <br><br>(color-annotated Fig. 2A of Gennaro); (Ex. 1002, ¶150).<br><br>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 |
|---|---|
| |  <br>FIG.1B<br>FIG.1 FIG.1A FIG.1B<br>(color-annotated Fig. 1B of Gennaro); (Ex. 1002, ¶151).<br>In the prior art embodiment, Gennaro describes that the hashes in the signed table are verified once the signature is verified. (Ex. 1002, ¶152).<br>• "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).<br>• *See also, e.g.*, Ex. 1008, 3:5-7.<br>*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).<br>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



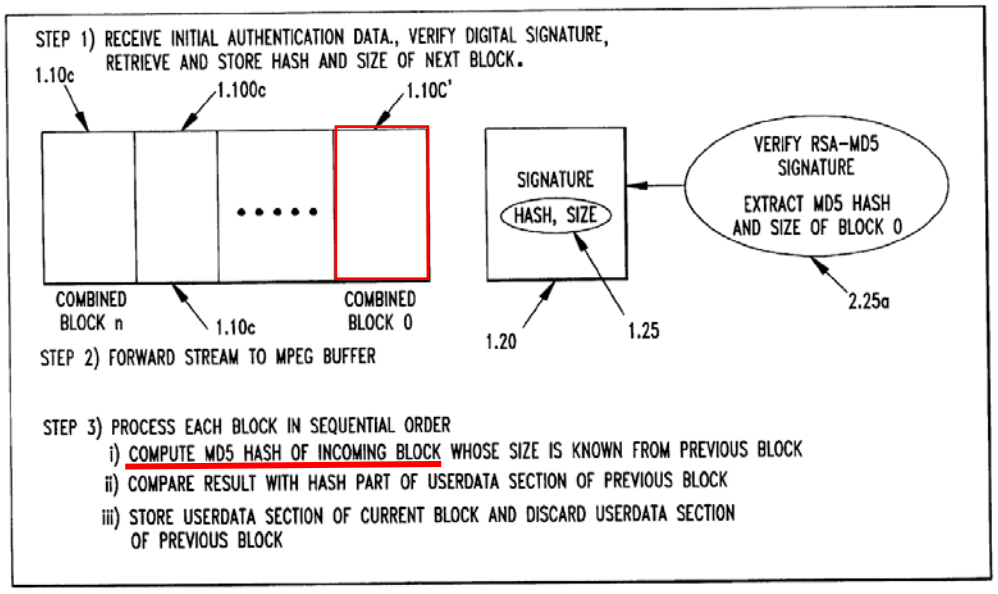

(color-annotated Fig. 2A of Gennaro); (Ex. 1002, ¶154).

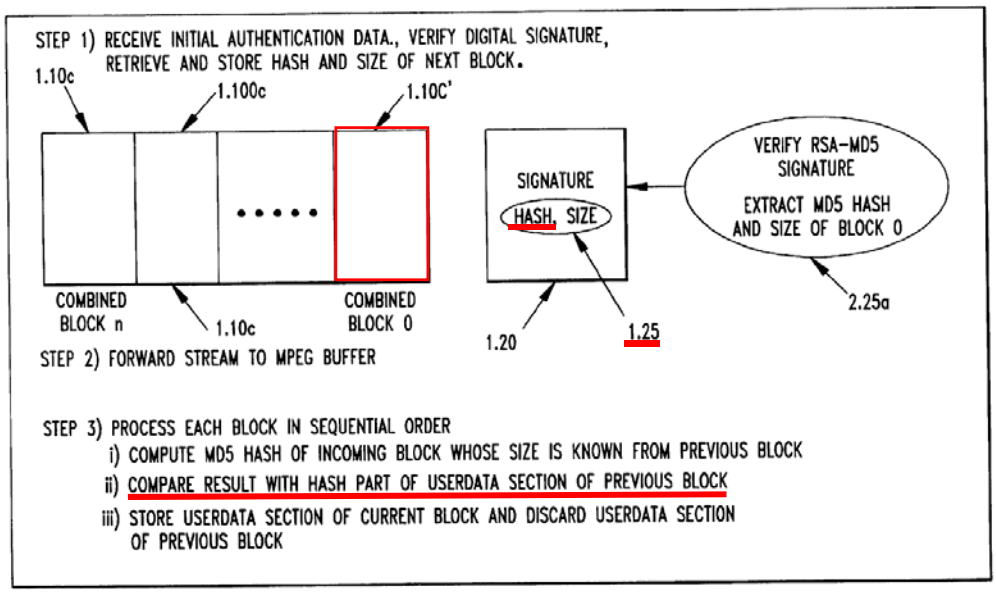
- "The next sixteen bytes of the 20 byte ancillary information placeholder is updated to hold the MD5 cryptographic hash of the successor block. ***This information can be used to authenticate the [] successor block*** which already has its ancillary information in place." (Ex. 1008, 4:44-48).
- "If 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).<br>• "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).<br>• *See also, e.g.*, Ex. 1008, 2:67-3:4, 4:51-55;<br>*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).<br><br>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).<br><br>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).<br><br>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)<br>• "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)<br>• "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)<br>*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).<br>• *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).<br><br>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.<br><br>FIG.2A<br>  <br><br>(color-annotated Fig. 2A of Gennaro); (Ex. 1002, ¶164).<br>• "***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.<br>FIG.2A<br>  <br><br>(color-annotated Fig. 2A of Gennaro); Ex. 1002, ¶166).<br>• "***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)<br>• *See also, e.g.*, Ex. 1008, 2:67-3:4, 4:44-48, 4:51-55, 5:29-34, 5:48-53. |

## VIII. SECONDARY CONSIDERATIONS

**168.** Prior to working on this declaration, I have no recollection of hearing of the inventors on the '815 patent or of the '815 patent. I have never heard anyone offer praise for the '815 patent, nor am I aware of any commercial success attributable to

the ’815 patent. I am also unaware of any copying of the alleged invention of the ’815 patent.

**169.** I have found no evidence of secondary considerations or objective factors of non-obviousness, and it is my opinion that Challenged Claims of the ’815 patent are obvious over the prior art. To the extent Patent Owner alleges secondary considerations in this proceeding, it is my opinion that they would not outweigh the strong evidence of obviousness of the ’815 patent Claims, as set forth herein.

## IX. CONCLUSION

**170.** As explained above, the Challenged Claims 42 and 43 of the ’815 patent are disclosed and/or rendered obvious by the combinations set forth therein, and the Challenged Claims 42 and 43 are unpatentable under 35 U.S.C. § 103.

**171.** In signing this declaration, I understand that the declaration will be filed as evidence in a review proceeding before the Patent Trial and Appeal Board of the U.S. Patent and Trademark Office. I acknowledge that I may be subject to cross examination in the case and that cross examination will take place within the United States. If cross examination is required of me, I will appear for cross examination within the United States during the time allotted for cross examination.

**172.** I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false

statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of the Title 18 of the United States Code and that such willful false statements may jeopardize the results of these proceedings.

Executed on: March 20, 2020

Vijay Madisetti, Ph.D.

3/20/20

# APPENDIX

# A

**Dr. Vijay K. Madisetti**
**Fellow, IEEE**
vkm@madisetti.com
Cell: 770-527-0177
**Address:**
**56 Creekside Park Drive**
**Johns Creek, GA 30022**

## Employment:

- 1984-1989: Post Graduate Researcher (UC Berkeley),
- 1989-present: Full Professor of Electrical & Computer Engineering (**Georgia Tech, Atlanta, GA 30332**).

**Areas of Technical Interest –** Wireless & Mobile Communications, Computer Engineering, Circuit Design (Analog/Digital), Software Engineering, Digital Signal Processing, Wireline & Wireless Computer Networks, Software Systems**,** Control Systems, Cloud Computing.

## Startup Companies:

Director, **VP Technologies, Inc**. (1995- ): A startup commercialized through Georgia Tech's Advanced Technology Development Corporation (ATDC) focusing on digital software and hardware design services for military market. http://www.vptinc.net

Director, **Soft Networks, LLC** (2001-2007): A startup commercialized through Georgia Tech support focusing on software development tools and compilers for Cellular/WiFi/VOIP/telecommunication products. http://www.soft-networks.com

Director, **Elastic Video Inc.** (2007- 2009): A startup commercialized through Georgia Tech's VentureLab (http://venturelab.gatech.edu) development image and video processing software for wireless & IP networking.

## Litigation Experience (2011-2019) With Testimony

**(Note: There may be multiple cases between the parties, e.g., District Court v. ITC, US versus Foreign Cases)**

**Case Name: HTC v. IPCOM**
Case No: 1:2008-cv-01897 (District of Columbia)
Expert for IPCOM
(3G Standards: 2009 – 2012)
Testified by deposition

**Case Name: Apple v. Kodak**
Case No. ITC 337-TA-717 (ITC)
Expert for Kodak
(Digital Image Processing & UI: 2008-2011)
Testified at trial

**Case Name: Harkabi v. Sandisk,**
Case No: 1:08-cv-08203-WHP (SDNY)
Expert for Harkabi
(Digital Rights Management for Flash Devices: 2010-2012)
Testified at trial

**Case Name: Yangaroo Inc. v. Destiny Media Technologies, Inc.**
Case No: 09-C-0462 (ED Wisconsin)
Expert for Yangaroo.
(Digital Rights Management Streaming: 2010-2011)
Testified by deposition

**Case Name: Motorola v. Microsoft**,
Case No: ITC 337-TA-752
Expert for Motorola
(Peer to Peer Gaming: 2011-2013)
Testified at trial

**Case Name: Motorola v. Apple,**
Case No: ITC 337-TA-745 (ITC)
Expert for Motorola
(Mobile Applications & UI: 2011-2012)
Testified at trial

**Case Name: Innovative Sonic Ltd. vs. RIM**
Case No: 3:11-cv-00706-K-BF (ND Dallas)
Expert for Innovative Sonic Ltd
(3G Standards – Encryption, HSDPA: 2010-2013)
Testified at trial

**Case Name: Interdigital v. ZTE et al (JDA)**
Case No: ITC 337-TA-800
Expert for JDA
(3G Standards – HSDPA: 2012-2013)
Testified at trial

**Case Name: Kodak v. Apple, HTC**
Case No: ITC 337-TA-831
Expert for Kodak
(Digital Image Processing & UIs: 2011-2012)
Submitted reports

**Case Name: Calypso v. T-Mobile**
Case No: 2:08-CV-441-JRG-RSP [ED Texas]
Expert for T-Mobile
(Unified Communications: 2012-2013)
Testified by deposition

**Case Name: TracBeam v. AT&T**
Case No: 6:11-cv-00096-LED [ED Texas]
Expert for AT&T
(GPS Services: 2011-2012)
Testified by deposition

**Case Name: BT v. Cox/Comcast**
Case No: 10-658 (SLR) (District of Delaware)
Expert for Cox and Comcast
(VOIP, Network Management: 2012-2014 )
Testified by deposition

**Case Name: Ericsson v. Samsung**
Case No: 337-TA-862 (ITC)
Expert for Ericsson
(RF Receivers, EDGE Standards: 2012- 2013)
Testified at trial

**Case Name: IPR – ContentGuard v. ZTE**
Case No: (PTAB)
Expert for ZTE
(DRM for Digital Devices: 2012-2014)
Testified by deposition

**Case Name: Emblaze v. Apple**
Case No: 5:11-cv-01079-PSG (ND Cal)
Expert for Emblaze
(Digital Video/Audio Streaming: 2012-2014 )
Testified at trial

**Case Name: Emblaze v. Microsoft**
Case No: 3:12-cv-05422-JST (ND Cal)
Expert for Emblaze
(Digital Video/Audio Streaming: 2012 - Present)
Ongoing

**Case Name: MMI v. RIM**
Case No: 2:10-cv-00113-TJW-CE (ED Texas)
Expert for MMI
(Area: Mobile Devices/User Interfaces: 2012- 2013)
Testified by deposition

**Case Name: Wi-LAN v. Apple**
Case No: 13-cv-0790 DMS (ED Texas)
Case No: 3:14-cv-010507-DMS-BLM
Expert for Wi-LAN
(Area: 4G/3G Wireless Communications: 2013 –2018 )
Testified by Deposition

**Case Name: Sentius LLC v. Microsoft**
Case No: 5:13-cv-00825-PSG (ND Cal)
Expert for Sentius
(Area: Enterprise Software Systems: 2014-2015)
Testified by deposition

**Case Name: Medius Eagle Harbor v. Ford**
Case No: 3:1-cv-05503-BHS (WD Washington)
Expert for Medius Tech
(Area: Automotive Multimedia Systems: 2014-2016)
Testified at Trial

**Genband US LLC v. Metaswitch Networks**
No 2:14-cv-33 (E.D. Texas)
Expert for Metaswitch Networks
Technology: Voice & Data Over IP Networks (2014-2015)
Submitted declarations & deposition

**Enterprise - Systems Technologies S.a.r.l v. Samsung Electronics Co. Ltd**
Case No: 6:14-cv-555-MHS (ED Texas) and ITC Inv. No. 337-TA-925
Expert for Samsung
Technology: Android Operating System (2014-2015)
Testified by deposition

**Ericsson Inc. v. Apple Inc.**
Case No: 2:15-cv-287 (ED Texas) and ITC-337-952/953
Expert for Ericsson
Technology: 4G Wireless Systems (2015 – present)
Testified by deposition & Trial

**Intellectual Ventures LLC v. Motorola Mobility LLC (Google)**
Case No: 13-cv-61358-RST (S.D. Florida)
Expert for Google
Technology: Wireless Systems (2013- present)
Testified by deposition (IPR)

**Intellectual Ventures LLC v. Nikon Corp**
C.A No. 11-1025-SLR (District of Delaware)
Expert for Nikon
Technology: Wireless Systems (WiFi) (2014-2015)
Submitted declarations

**Masimo v. Mindray Biomedical Electronics Co. / Philips**
Case No: SACV-12-02206 CJC (JPRx) (C.D. California)
8:12-cv-02206-CJC-JPR
Expert for Masimo
Technology: Pulse Oximetry (2014 – 2016)
Submitted reports, testified by deposition

**Samsung v. Nvidia**
Case No: 3:14-cv-757-REP ED Virginia
Expert for Samsung
Technology: Microprocessors & Memories (2014 – 2016)
Submitted reports & Deposition & Trial

**Chrimar v. HP/Cisco/Alcatel/Dell/Adtran/AeroHive/D-Link/TP-Link/TrendNet/Juniper Networks/CoStar/Accton/AMX**
Case No: 4:13-cv-1300-JSW, Case 6:15-cv-163 (ED Texas)
Case No: 6:15-cv-00618-JRG-JDL (ED Texas)
Expert for Chrimar
Technology: Power over Ethernet (2015-2017)
Submitted reports & Deposition & Trial

**Chamberlain v. Ryobi/TTI**
Case No: 1:16-cv-06097 (ND Illinois)
Expert for Ryobi
Technology: Wireless/IoT/Barrier Movement (2016 – present)
Submitted reports & deposition & trial testimony

**IOEngine v. IMC/Imation**
Case No: cv-14-1572-GMS (US Delaware)
Expert for IMC/Imation
Technology: Networked Storage Device (2016-2017)
Submitted reports & deposition & trial testimony

**Huawei v. Samsung**
Case No: 3:16-cv-2787-WHO (ND Cal)
Expert for Samsung
Technology: 4G/LTE Random Access Protocols (2016-present)
Submitted reports & deposition

**Hitachi Maxell v. ZTE/Huawei**
Case No: 5:16-cv-00178-RWS (ED Texas)
Expert for Hitachi Maxell
Technology: Digital Cameras
Submitted reports & deposition (2017 – 2018)

**Qualcomm v. Apple**
Case No: 17-ccv-0108-GPC-MDD (SD Cal) Also, Related ITC/FTC Matters
Expert for Qualcomm
Technology: 4G/Wireless Communications/Smartphones (2017-2019)
Submitted Reports and Deposition

**Qualcomm v. Apple**
Case No: 3:17-cv-01375-DMS-MDD (SD Cal)
Expert for Qualcomm
Technology: 4G/Wireless Communications/Smartphones (2017-2019)
Submitted Reports and Deposition

**Optis v. Huawei**
Case No: 2:17-cv-123 (E.D. Texas)
Expert for Optis Wireless
Technology: 4G/Video (2017-2019)
Submitted reports & deposition

**Broadcom v. Sony**
Case No: 8:16-cv-1052 (PTAB and Central Cal)
Expert for Broadcom
Technology: Wi Fi
No activity - settled (2017 – 2017)

**Ameranth v. Hyatt Corporation**
Case No: 3:11-cv-1810 (SD Cal)
Expert for Hyatt
Technology: Wireless eCommerce Applications (2017-2018)
Expert Technical consulting

**Beckman Coulter v. Sysmex**
Case No: 1:17-cv-24049-DPG (ND Illinois)
Expert for Sysmex
Technology: Medical Instrumentation Automation (2017-present)
Testifying Expert

**TQ Delta**
Expert for TQ Delta
Technology: DSL Technologies (2018-present)
Testifying Expert

**3GL/KPN v. LG, Blackberry, HTC**
Expert for 3GL/KPN
Technology: 4G/LTE Protocols (2018-present)
Reports and Deposition / Testifying Expert

**Cirba/Densify v. VMWare**
Case No: 1:19-cv-00742-LPS
Expert for Cirba/Densify
Technology: Virtualization (2019-present)
Reports Deposition/PI Hearing / Testifying Expert

**Power Integrations v. On Semiconductor**
Case No: 17-cv-03189-BLF
Expert for On Semiconductor
Technology: Power Electronics (2018-present)
Reports & Deposition/Testifying Expert.

**Wilan v. Apple**
Case No: 3:14-cv-2235
Expert for WiLan
Technology: Voice over LTE/LTE Protocols
Reports and Testified through Deposition/Trials

Additional matters include declarations supporting IPRs at the PTAB for Google (US Patent 8,601,154), On Semiconductor (US Patent 6,212,079), Ubisoft (US Patent 5,490,216), Broadcom (WiFi), Sony (US Patent 6,101,534), Kia, (US Patent 5,530,431), Qualcomm, Fortinet (US patent 6,195,587), and ZTE (US Patent 7,523,072), and Ericsson, Amazon, Ring, Digital Ally, On Semiconductor, United Patents, Lenovo, BMC Software.

## Earned Degrees

1. **B. Tech (Hons), Electronics & Electrical Comm. Engineering**
   *Indian Institute of Technology (IIT), Kharagpur, India*
   1984.

2. **Ph.D., Electrical Engineering & Computer Sciences (EECS)**
   *University of California (UC), Berkeley. CA*
   1989.

## Books

1. **VLSI Digital Signal Processors**
   *Madisetti, V.K.;*
   Boston: MA, IEEE Press: Butterworth Heinemann, 1995, 525 pp.

2. **Quick-Turnaround ASIC Design in VHDL**
   *Romdhane, M., Madisetti, V.K., Hines, J.*
   Boston: MA, Kluwer Academic Press, 1996, 190 pp.

3. **The Digital Signal Processing Handbook (First Edition)**
   *Madisetti, V. K., Williams, D. (Editors)*
   CRC Press, Boca Raton, Fla, 1998, 2500 pp.

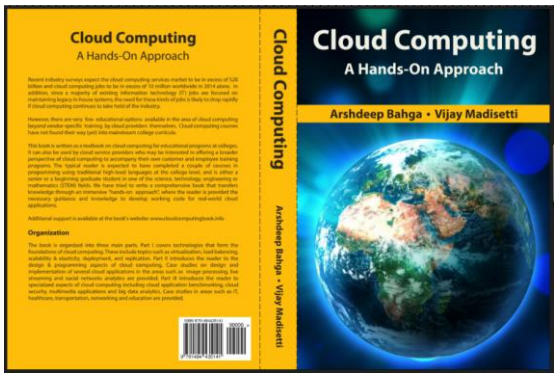
4. **VHDL: Electronics Systems Design Methodologies**.
   *Madisetti, V. K. (Editor)*
   Boston: MA, IEEE Standards Press, 2000, ISBN 0-7381-1878-8.

5. **Platform-Centric Approach to System-on-Chip (SoC) Design.**
   *Madisetti, V. K., Arpnikanondt, A.*
   Springer, Boston: MA, Springer, 2004, 280 pp.

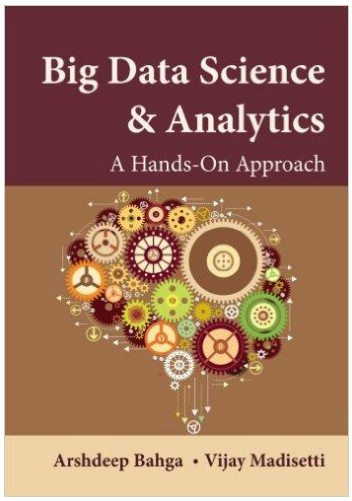
6. **The Digital Signal Processing Handbook – Second Edition.**
   Madisetti, V. K. (2009)
   CRC Press, Boca Raton, Fla.

7. **Cloud Computing: A Hands-On Approach**
   *A Bahga, V. Madisetti (2013)*
   Amazon CreateSpace Publishing, 2013, 454 pp.

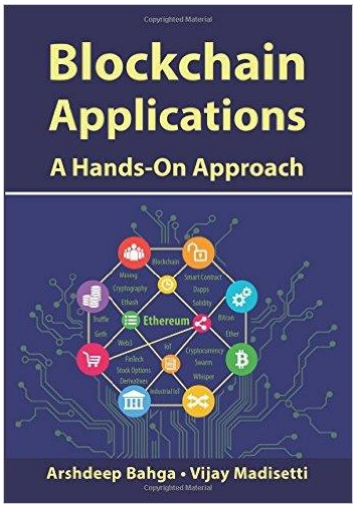
8. **Internet of Things: A Hands-On Approach**
   *A Bahga, V. Madisetti (2014)*
   Amazon CreateSpace Publishing, 2014, 450 pp.

9. **Big Data Science & Analytics: A Hands-On Approach**
   *A Bahga, V. Madisetti (2016)*
   Amazon CreateSpace Publishing, 2016, 542 pp.




10. **Blockchain Applications: A Hands-On Approach**
    *A Bahga, V. Madisetti (2017)*
    Amazon CreateSpace Publishing, 2017, 380 pp.




## Edited Books & Collection of Papers

1. **Advances in Parallel & Distributed Simulation**
   *Madisetti, V.K.;Nicol, D., Fujimoto, R. (Editors)*
   San Diego, CA: SCS Press, 1991, 200 pp.

2. **Modeling, Analysis, Simulation of Computer & Telecommunications Systems**
   *Madisetti, V., Gelenbe, E., Walrand, J. W. (Editors)*
   Los Alamitos: CA, IEEE Computer Society Press, 1994, 425 pp.

3. **Modeling & Simulations on Microcomputers**
   *Madisetti, V.K. (Editor)*
   San Diego, CA: SCS Press, 138 pp. 1990.

## Editorship of Journals & Transactions

1. **IEEE Design & Test of Computers**
   Special Issue: Reengineering Digital Systems
   April – June 1999 (Vol 16, No 2)
   *Madisetti, V.K (Editor)*
   Los Alamitos: CA, IEEE Computer Society Press, 1999.

2. **IEEE Design & Test of Computers**
   Special Issue: Rapid Prototyping of Digital Systems
   Fall 1996 (Vol 13, No 3)
   *Madisetti, V., Richards, M. (Editors)*
   Los Alamitos: CA, IEEE Computer Society Press, 1994, 425 pp.

3. **IEEE Transactions on Circuits & Systems II**
   *Associate Editor*: 1993-1995.

4. **International Journal in Computer Simulation**
   *Associate Editor: 1990-1993*

5. **International Journal in VLSI Signal Processing**
   *Editorial Board: 1995 - Present*

## Issued US Patents

[1] 10,503,927 Method and system for securing cloud storage and databases from insider threats and optimizing performance, Issued Dec 10, 2019

[2]. 10,460,283 Smart contract optimization for multiparty service or product ordering

system, Issued Oct 29, 2019

[3]. 10,459,946 Method and system for tuning blockchain scalability, decentralization, and security for fast and low-cost payment and transaction processing, Issued Oct 29, 2019

[4]. 10,402,589 Method and system for securing cloud storage and databases from insider threats and optimizing performance, Issued Sep 3, 2019

[5]. 10,394,845 Method and system for tuning blockchain scalability for fast and low-cost payment and transaction processing, Issued Aug 27, 2019

[6]. 10,289,631 Method and system for tuning blockchain scalability for fast and low-cost payment and transaction processing, Issued May 24, 2019

[7]. 10,255,342 Method and system for tuning blockchain scalability, decentralization, and security for fast and low-cost payment and transaction processing, Issued April 9, 2019

[8]. 10,243,743 Tokens or crypto currency using smart contracts and blockchains, Issued March 26. 2019

[9]. 10,204,339 Method and system for blockchain-based combined identity, ownership, integrity and custody management, Issued February 12, 2019

[10]. 10,204,148 Method and system for tuning blockchain scalability, decentralization, and security for fast and low-cost payment and transaction processing, Issued Feb 12, 2019

[11]. 10,121,143 Method and system for blockchain-based combined identity, ownership, integrity and custody management, Issued Nov 6, 2018

[12]. 10,102,526 Method and system for blockchain-based combined identity, ownership, integrity and custody management, October 16, 2018

[13]. 10,102,265 Method and system for tuning blockchain scalability for fast and low-cost payment and transaction processing, October 16, 2018

[14]. 9,935,772 Methods and systems for operating secure digital management aware applications, April 3, 2018

[15]. 9,769,213 Method and system for secure digital object management, Issued Sep 19, 2017

## Refereed Journal Publications

1. **Trends in the Electronic Control of Mine Hoists**
   *Madisetti, V. and Ramlu, M.*,
   *IEEE Transactions on Industry Applications, Vol IA-22, No. 6, November/December 1986. Pages 1105-1112*

2. **Multilevel range/NEXT performance in digital subscriber loops**
   *Brand, G.; Madisetti, V.; Messerschmitt, D.G.;*
   Communications, Speech and Vision, IEE Proceedings I [see also IEE Proceedings-Communications] ,Volume: 136 , Issue: 2 , April 1989
   Pages:169 – 174

3. **Seismic migration algorithms on parallel computers**
   *Madisetti, V.K.; Messerschmitt, D.G.;*
   Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on] ,Volume: 39 , Issue: 7 , July 1991
   Pages:1642 – 1654

4. **Asynchronous algorithms for the parallel simulation of event-driven dynamical systems**
   *Madisetti, V.K.; Walrand, J.C.; Messerschmitt, D.G.:*
   ACM Transactions on Modeling and Computer Simulation, v 1, n 3, July 1991, Pages: 244-74

5. **Synchronization mechanisms for distributed event-driven computation**
   *Madisetti, V.K.; Hardaker, D.:*
   ACM Transactions on Modeling and Computer Simulation, v 2, n 1, Jan. 1992, Pages: 12-51

6. **Efficient VLSI Architectures for the Arithmetic Fourier Transform (AFT)**
   *Kelley, B.T.; Madisetti, V.K.;*
   Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on] ,Volume: 41 , Issue: 1 , January 1993
   Pages:365-378

7. **The fast discrete Radon transform. I. Theory**
   *Kelley, B.T.; Madisetti, V.K.;*
   Image Processing, IEEE Transactions on ,Volume: 2 , Issue: 3 , July 1993
   Pages:382 – 400

8. **The Georgia Tech digital signal multiprocessor**
   *Barnwell, T.P., III; Madisetti, V.K.; McGrath, S.J.A.;*
   Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on] ,Volume: 41 , Issue: 7 , July 1993
   Pages:2471 – 2487

9. **The MIMDIX Environment for Parallel Simulation**

*Madisetti, V.K.; Hardaker, D.; Fujimoto, R.M.:*
Journal of Parallel and Distributed Computing, v18, no. 4, August 1993,
Pages: 473-83.

**10. LMSGEN: a prototyping environment for programmable adaptive digital filters in VLSI**
*Romdhane, M.S.B.; Madisetti, V.K.;*
Chapter in VLSI Signal Processing, VII, 1994.,
Pages:33 – 42

**11. Fixed-point co-design in DSP**
*Egolf, T.W.; Famorzadeh, S.; Madisetti, V.K.;*
Chapter in VLSI Signal Processing, VII, 1994.,
Pages:113 - 126

**12. A fast spotlight-mode synthetic aperture radar imaging system**
*Madisetti, V.K.;*
Communications, IEEE Transactions on ,Volume: 42 , Issue: 234 , February-April 1994
Pages:873 – 876

**13. Rapid prototyping on the Georgia Tech digital signal multiprocessor**
*Curtis, B.A.; Madisetti, V.K.;*
Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on] ,Volume: 42 , Issue: 3 , March 1994
Pages:649 – 662

**14. Low-power signaling in asymmetric noisy channels via spectral shaping**
*Sipitca, M.; Madisetti, V.K.;*
Signal Processing Letters, IEEE, Volume: 1 , Issue: 8 , Aug 1994
Pages:117 – 118

**15. A quantitative methodology for rapid prototyping and high-level synthesis of signal processing algorithms**
*Madisetti, V.K.; Curtis, B.A.;*
Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on] ,Volume: 42 , Issue: 11 , Nov. 1994
Pages:3188 – 3208

**16. Computer Simulation of Application-Specific Signal Processing Systems**
*Casinovi, G.; Madisetti, V.K.;*
International Journal in Computer Simulation, Vol. 4, No. 4, Nov 1994.

**17. System partitioning of MCMs for low power**
*Khan, S.A.; Madisetti, V.K.;*
Design & Test of Computers, IEEE ,Volume: 12 , Issue: 1 , Spring 1995
Pages:41 – 52

**18. Error correcting run-length limited codes for magnetic recording**
*Jaejin Lee; Madisetti, V.K.;*
Magnetics, IEEE Transactions on ,Volume: 31 , Issue: 6 , Nov. 1995
Pages:3084 – 3086

**19. Virtual prototyping of embedded microcontroller-based DSP systems**
*Madisetti, V.K.; Egolf, T.W.;*
Micro, IEEE ,Volume: 15 , Issue: 5 , Oct. 1995
Pages:9 – 21

**20. Constrained multitrack RLL codes for the storage channel**
*Lee, J.; Madisetti, V.K.;*
Magnetics, IEEE Transactions on ,Volume: 31 , Issue: 3 , May 1995
Pages:2355 – 2364

**21. Rapid digital system prototyping: current practice, future challenges**
*Madisetti, V.K.;*
Design & Test of Computers, IEEE ,Volume: 13 , Issue: 3 , Fall 1996
Pages:12 – 22

**22. Conceptual prototyping of scalable embedded DSP systems**
*Dung, L.-R.; Madisetti, V.K.;*
Design & Test of Computers, IEEE ,Volume: 13 , Issue: 3 , Fall 1996
Pages:54 – 65

**23. Advances in rapid prototyping of digital systems**
*Madisetti, V.K.; Richards, M.A.;*
Design & Test of Computers, IEEE ,Volume: 13 , Issue: 3 , Fall 1996
Pages:9

**24. Combined modulation and error correction codes for storage channels**
*Jaejin Lee; Madisetti, V.K.;*
Magnetics, IEEE Transactions on ,Volume: 32 , Issue: 2 , March 1996
Pages:509 – 514

**25. Model-based architectural design and verification of scalable embedded DSP systems-a RASSP approach**
*Dung, L.-R.; Madisetti, V.K.; Hines, J.W.;*

Chapter in VLSI Signal Processing, IX, 1996.
Pages:147 – 156

**26. Low-power digital filter implementations using ternary coefficients**
*Hezar, R.; Madisetti, V.K.;*
Chapter in VLSI Signal Processing, IX, 1996.,
Pages:179 – 188

**27. All-digital oversampled front-end sensors**
*Romdhane, M.S.B.; Madisetti, V.K.;*
Signal Processing Letters, IEEE, Volume: 3 , Issue: 2 , Feb. 1996
Pages:38 – 39

**28. Modeling COTS components in VHDL**
*Calhoun, S., Reese, R; Egolf, T., Madisetti, V.K.;*
Journal of VLSI Signal Processing, Volume: 14 , Issue: 2 , Nov 1996
Pages: 24 – 31

**29. VHDL-Based Rapid Systems Prototyping**
*Egolf, T.; Madisetti, V.K.;*
Journal of VLSI Signal Processing, Volume: 14 , Issue: 2 , Nov 1996
Pages: 40-52

**30. Interface design for core-based systems**
*Madisetti, V.K.; Lan Shen;*
Design & Test of Computers, IEEE ,Volume: 14 , Issue: 4 , Oct.-Dec. 1997
Pages:42 - 51

**31. Incorporating cost modeling in embedded-system design**
*Debardelaben, J.A.; Madisetti, V.K.; Gadient, A.J.;*
Design & Test of Computers, IEEE ,Volume: 14 , Issue: 3 , July-Sept. 1997
Pages:24 – 35

**32. On homomorphic deconvolution of bandpass signals**
*Marenco, A.L.; Madisetti, V.K.;*
Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on] ,Volume: 45 , Issue: 10 , Oct. 1997
Pages:2499 – 2514

**33. A case study in the development of multi-media educational material: the VHDL interactive tutorial**
*Gadient, A.J.; Stinson, J.A., Jr.; Taylor, T.C.; Aylor, J.H.; Klenke, R.H.; Salinas, M.H.; Madisetti, V.K.; Egolf, T.; Famorzadeh, S.; Karns, L.N.; Carter, H.W.;*

Education, IEEE Transactions on ,Volume: 40 , Issue: 4 , Nov. 1997
Pages:17 pp.

**34. Adaptive mobility management in wireless networks**
*Jeongwook Kim; Madisetti, V.K.;*
Electronics Letters ,Volume: 34 , Issue: 15 , 23 July 1998
Pages:1453 – 1455

**35. Efficient implementation of two-band PR-QMF filterbanks**
*Hezar, R.; Madisetti, V.K.;*
Signal Processing Letters, IEEE ,Volume: 5 , Issue: 4 , April 1998
Pages:92 – 94

**36. On fast algorithms for computing the inverse modified discrete cosine transform**
*Yun-Hui Fan; Madisetti, V.K.; Mersereau, R.M.;*
Signal Processing Letters, IEEE ,Volume: 6 , Issue: 3 , March 1999
Pages:61 – 64

**37. System on chip or system on package?**
*Tummala, R.R.; Madisetti, V.K.;*
Design & Test of Computers, IEEE ,Volume: 16 , Issue: 2 , April-June 1999
Pages:48 – 56

**38. Reengineering legacy embedded systems**
*Madisetti, V.K.; Jung, Y.-K.; Khan, M.H.; Kim, J.; Finnessy, T.;*
Design & Test of Computers, IEEE ,Volume: 16 , Issue: 2 , April-June 1999
Pages:38 – 47

**39. Reengineering digital systems**
*Madisetti, V.K.;*
Design & Test of Computers, IEEE ,Volume: 16 , Issue: 2 , April-June 1999
Pages:15 – 16

**40. Parameter optimization of robust low-bit-rate video coders**
*Sangyoun Lee; Madisetti, V.K.;*
Circuits and Systems for Video Technology, IEEE Transactions on, Volume: 9
Issue: 6 , Sept. 1999
Pages:849 – 855

**41. Closed-form for infinite sum in bandlimited CDMA**
*Jatunov, L.A.; Madisetti, V.K.;*
Communications Letters, IEEE ,Volume: 8 , Issue: 3 , March 2004
Pages:138 – 140

**42. A new protocol to enhance path reliability and load balancing in mobile ad hoc networks**
*Argyriou, A.; Madisetti, V.K.;*
Journal of Ad Hoc Networks, Elsevier Press, 2004

**43. Closed-form analysis of CDMA systems using Nyquist pulse**
*Jatunov, L.A.; Madisetti, V. K.;*
Communications Letters, IEEE (Under Revision), 2005.

**44. Systematic Design of End-to-End Wireless Mobility Management Prototocols,**
*Argyriou, A.; Madisetti, V. K.;*
ACM/Springer Wireless Networks (WINET), Accepted 2005.

**45. A Novel End-to-End Approach for Video Streaming Over the Internet,**
*Argyriou, A.; Madisetti, V. K.;*
Kluwer Telecommunications Systems, Vol. 28, No. 2, Pages 133-150, Jan 2005. *Special Issue on Multimedia Streaming.*

**46. An Analytical Framework of RD Optimized Video Streaming with TCP,**
*Argyriou, A.; Madisetti, V. K.;*
IEEE Transactions on Multimedia, Submitted for review in March 2005.

**47. Modeling the Effect of Handoffs on Transport Protocol Performance,**
*Argyriou, A.; Madisetti, V. K.;*
IEEE Transactions on Mobile Computing, Submitted for review in March 2005

**48. Throughput Models for Transport Protocols with CBR and VBR Traffic Workloads",**
*Argyriou, A.; Madisetti, V. K.;*
ACM Transactions on Multimedia Computing, Communications & Applications, Submitted for review in April 2005.

**49. "Electronic System, Platform & Package Codesign",**
*Madisetti, V. K.*
*IEEE Design & Test of Computers*, Volume 23, Issue 3, June 2006. pages 220-233.

**50. "The Design of an End-to-End Handoff Management Protocol",**
*A. Argyriou, Madisetti, V. K.*
*Wireless Networks, Springer, May 2006.*

**51."A Soft-Handoff Transport Protocol for Media Flows in Heterogeneous Mobile Networks ",**
*A. Argyriou, Madisetti, V. K.*
*Computer Networks, Vol 50, Issue 11, Pages 1860-1871, August 2006.*

**52. "Computationally Efficient SNR Estimation for Bandlimited WCDMA Systems"**
*L. Jatunov, Madisetti, V. K.*
*IEEE Transactions on Wireless Communications, Volume 5, Issue 13, December 2006, Pages 3480-3491.*

**53."Space-Time Codes for Wireless & Mobile Applications",**
*M. Sinnokrot, Madisetti, V.K.*
*DSP Handbook, Second Edition, 2009 (to be published)*

**54.**A. Bahga, V. Madisetti, "**Rapid Prototyping of Advanced Cloud-Based Systems**", *IEEE Computer*, vol. 46, no. 11, Nov 2013, pp 76-83, 2013

**55.**A. Bahga, V. Madisetti, "**Cloud-Based Information Integration & Informatics Framework for Healthcare Application**s", *IEEE Computer*, February 2015.

**56.**A. Bahga, V. Madisetti, "**A Cloud-based Approach for Interoperable EHRs**", *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 5, Sep 2013, pp. 894-906, 2013

**57.**A. Bahga, V. Madisetti, "**Cloud-Based Information Technology Framework for Data Driven Intelligent Transportation Systems**", *Journal of Transportation Technologies*, vol.3 no.2, April 2013

**58.**A. Bahga, V. Madisetti, **"Performance Evaluation Approach for Multi-tier Cloud Applications"**, *Journal of Software Engineering and Applications*, vol. 6, no. 2, pp. 74-83, Mar 2013.

**59.**Yusuf, A., V. Madisetti, **"Configuration for Predicting Travel Time Using Wavelet Packets and Support Vector Regression**", ***Journal of Transportation Technologies***, vol 3, no. 3, June 2013.

**60.**A. Bahga, V. Madisetti, **"Analyzing Massive Machine Maintenance Data in a Computing Cloud"**, *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 10, pp. 1831 - 1843, 2012.

**61.** N. Radia. Y. Zhang, M. Tatimapula, V. Madisetti, **"Next Generation Applications on Cellular Networks: Trends, Challenges, and Solutions**," *Proceedings of the IEEE*, Vol 100, Issue 4, pp. 841-854, 2012.

**62.** A. Bahga, V. Madisetti, "**Rapid Prototyping of Advanced Cloud-Based Systems**", *IEEE Computer*, vol. 46, no. 11, Nov 2013, pp 76-83, 2013

**63.** A. Bahga, V. Madisetti, "**A Cloud-based Approach for Interoperable EHRs**", *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 5, Sep 2013, pp. 894-906, 2013

**64.** A. Bahga, V. Madisetti, "**Cloud-Based Information Integration & Informatics Framework for Healthcare Application**s", *IEEE Computer*, February 2015.

## Peer Reviewed Conference Publications

1. **Dynamically-reduced complexity implementation of echo cancelers**
   *Madisetti, V.; Messerschmitt, D.; Nordstrom, N.;*
   Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '86. ,Volume: 11 , Apr 1986

2. **Seismic migration algorithms using the FFT approach on the NCUBE multiprocessor**
   *Madisetti, V.K.; Messerschmitt, D.G.;*
   Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on , 11-14 April 1988

3. **Seismic migration algorithms on multiprocessors**
   *Madisetti, V.K.; Messerschmitt, D.G.;*
   Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on , 11-14 April 1988
   Pages:2124 - 2127 vol.4

4. **WOLF: A rollback algorithm for optimistic distributed simulation systems**
   *Madisetti, V.; Walrand, J.; Messerschmitt, D.;*
   Simulation Conference Proceedings, 1988 Winter , December 12-14, 1988
   Pages:296 – 305

5. **Efficient distributed simulation**
   *Madisetti, V.; Walrand, J.; Messerschmitt, D.;*
   Simulation Symposium, 1989. The 22nd Annual , March 28-31, 1989
   Pages:5 - 6

6. **High speed migration of multidimensional seismic data**
   *Kelley, B.; Madisetti, V.;*
   Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on , 14-17 April 1991
   Pages:1117 - 1120 vol.2

7. **Performance of a fast analog VLSI implementation of the DFT**
   *Buchanan, B.; Madisetti, V.; Brooke, M.;*
   Circuits and Systems, 1992., Proceedings of the 35th Midwest Symposium on , 9-12 Aug. 1992
   Pages:1353 - 1356 vol.2

8. **Task scheduling in the Georgia Tech digital signal multiprocessor**
   *Curtis, B.A.; Madisetti, V.K.;*
   Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on ,Volume: 5 , 23-26 March 1992
   Pages:589 - 592 vol.5

9. **The fast discrete Radon transform**
   *Kelley, B.T.; Madisetti, V.K.;*
   Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on ,Volume: 3 , 23-26 March 1992
   Pages:409 - 412 vol.3

10. **Yield-based system partitioning strategies for MCM and ASEM design**
    *Khan, S.; Madisetti, V.;*
    Multi-Chip Module Conference, 1994. MCMC-94, Proceedings., 1994 IEEE,15-17 March 1994
    Pages:144 – 149

11. **Multitrack RLL codes for the storage channel with immunity to intertrack interference**
    *Lee, J.; Madisetti, V.K.;*
    Global Telecommunications Conference, 1994. GLOBECOM '94. 'Communications: The Global Bridge'., IEEE,Volume: 3 , 28 Nov.-2 Dec. 1994
    Pages:1477 - 1481 vol.3

12. **A parallel mapping of backpropagation algorithm for mesh signal processor**
    *Khan, S.A.; Madisetti, V.K.;*
    Neural Networks for Signal Processing [1995] V. Proceedings of the 1995

IEEE Workshop , 31 Aug.-2 Sept. 1995
Pages:561 – 570

**13. Virtual prototyping of embedded DSP systems**
*Madisetti, V.K.; Egolf, T.; Famorzadeh, S.; Dung, L.-R.;*
Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on ,Volume: 4 , 9-12 May 1995
Pages:2711 - 2714 vol.4

**14. Assessing and improving current practice in the design of application-specific signal processors**
*Shaw, G.A.; Anderson, J.C.; Madisetti, V.K.;*
Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on ,Volume: 4 , 9-12 May 1995
Pages:2707 - 2710 vol.4

**15. Introduction to ARPA's RASSP initiative and education/facilitation program**
*Corley, J.H.; Madisetti, V.K.; Richards, M.A.;*
Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on ,Volume: 4 , 9-12 May 1995
Pages:2695 - 2698 vol.4

**16. DSP design education at Georgia Tech**
*Madisetti, V.K.; McClellan, J.H.; Barnwell, T.P., III;*
Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on ,Volume: 5 , 9-12 May 1995
Pages:2869 - 2872 vol.5

**17. Rapid prototyping of DSP systems via system interface module generation**
*Famorzadeh, S.; Madisetti, V.K.;*
Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on ,Volume: 2 , 7-10 May 1996
Pages:1256 - 1259 vol. 2

**18. Rapid prototyping of DSP chip-sets via functional reuse**
*Romdhane, M.S.B.; Madisetti, V.K.;*
Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on ,Volume: 2 , 7-10 May 1996
Pages:1236 - 1239 vol. 2

**19. A constructive deconvolution procedure of bandpass signals by homomorphic analysis**
*Marenco, A.L.; Madisetti, V.K.;*
Geoscience and Remote Sensing Symposium, 1996. IGARSS '96. 'Remote Sensing for a Sustainable Future.', International ,Volume: 3 , 27-31 May 1996
Pages:1592 - 1596 vol.3

**20. BEEHIVE: an adaptive, distributed, embedded signal processing environment**
*Famorzadeh, S.; Madisetti, V.; Egolf, T.; Nguyen, T.;*
Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on ,Volume: 1 , 21-24 April 1997
Pages:663 - 666 vol.1

**21. Target detection from coregistered visual-thermal-range images**
*Perez-Jacome, J.E.; Madisetti, V.K.;*
Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on ,Volume: 4 , 21-24 April 1997
Pages:2741 - 2744 vol.4

**22. Variable block size adaptive lapped transform-based image coding**
*Klausutis, T.J.; Madisetti, V.K.;*
Image Processing, 1997. Proceedings., International Conference on ,Volume: 3 , 26-29 Oct. 1997
Pages:686 - 689 vol.3

**23. A Rate 8/10 (0, 6) MTR Code And Its Encoder/decoder**
*Jaejin Lee; Madisetti, V.K.;*
Magnetics Conference, 1997. Digests of INTERMAG '97., 1997 IEEE International , 1-4 April 1997
Pages:BS-15 - BS-15

**24. VHDL models supporting a system-level design process: a RASSP approach**
*DeBardelaben, J.A.; Madisetti, V.K.; Gadient, A.J.;*
VHDL International Users' Forum, 1997. Proceedings , 19-22 Oct. 1997
Pages:183 – 188

**25. A performance modeling framework applied to real time infrared search and track processing**
*Pauer, E.K.; Pettigrew, M.N.; Myers, C.S.; Madisetti, V.K.;*
VHDL International Users' Forum, 1997. Proceedings , 19-22 Oct. 1997
Pages:33 – 42

**26. System design and re-engineering through virtual prototyping: a temporal model-based approach**
*Khan, M.H.; Madisetti, V.K.;*
Signals, Systems & Computers, 1998. Conference Record of the Thirty-Second Asilomar Conference on ,Volume: 2 , 1-4 Nov. 1998
Pages:1720 - 1724 vol.2

**27. A debugger RTOS for embedded systems**
*Akgul, T.; Kuacharoen, P.; Mooney, V.J.; Madisetti, V.K.;*
Euromicro Conference, 2001. Proceedings. 27th , 4-6 Sept. 2001
Pages:264 - 269

**28. Adaptability, extensibility and flexibility in real-time operating systems**
*Kuacharoen, P.; Akgul, T.; Mooney, V.J.; Madisetti, V.K.;*
Digital Systems, Design, 2001. Proceedings. Euromicro Symposium on , 4-6 Sept. 2001
Pages:400 – 405

**29. Effect of handoff delay on the system performance of TDMA cellular systems**
*Turkboylari, M.; Madisetti, V.K.;*
Mobile and Wireless Communications Network, 2002. 4th International Workshop on , 9-11 Sept. 2002
Pages:411 – 415

**30. Enforcing interdependencies and executing transactions atomically over autonomous mobile data stores using SyD link technology**
*Prasad, S.K.; Bourgeois, A.G.; Dogdu, E.; Sunderraman, R.; Yi Pan; Navathe, S.; Madisetti, V.;*
Distributed Computing Systems Workshops, 2003. Proceedings. 23rd International Conference on , 19-22 May 2003
Pages:803 – 809

**31. Performance evaluation and optimization of SCTP in wireless ad-hoc networks**
*Argyriou, A.; Madisetti, V.;*
Local Computer Networks, 2003. LCN '03. Proceedings. 28th Annual IEEE International Conference on , 20-24 Oct. 2003
Pages:317 - 318

**32. Implementation of a calendar application based on SyD coordination links**
*Prasad, S.K.; Bourgeois, A.G.; Dogdu, E.; Sunderraman, R.; Yi Pan; Navathe, S.; Madisetti, V.;*
Parallel and Distributed Processing Symposium, 2003. Proceedings.

International , 22-26 April 2003
Pages:8 pp.

**33. Bandwidth aggregation with SCTP**
*Argyriou, A.; Madisetti, V.;*
Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE Volume: 7 , 1-5 Dec. 2003
Pages:3716 - 3721 vol.7

**34. Software streaming via block streaming**
*Kuacharoen, P.; Mooney, V.J.; Madisetti, V.K.;*
Design, Automation and Test in Europe Conference and Exhibition, 2003 , 2003
Pages:912 – 917

**35. Frequency-dependent space-interleaving for MIMO OFDM systems**
*Mohajerani, P.; Madisetti, V.K.;*
Radio and Wireless Conference, 2003. RAWCON '03. Proceedings , Aug. 10-13, 2003
Pages:79 - 82

**36. A media streaming protocol for heterogeneous wireless networks**
*Argyriou, A.; Madisetti, V.;*
Computer Communications, 2003. CCW 2003. Proceedings. 2003 IEEE 18th Annual Workshop on , 20-21 Oct. 2003
Pages:30 – 33

**37. Realizing load-balancing in ad-hoc networks with a transport layer protocol**
*Argyriou, A.; Madisetti, V.;*
Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE ,Volume: 3 , 21-25 March 2004
Pages:1897 - 1902 Vol.3

**38.Streaming H.264/AVC video over the Internet**
*Argyriou, A.; Madisetti, V.;*
Consumer Communications and Networking Conference, 2004. CCNC 2004. First IEEE , 5-8 Jan. 2004
Pages:169 – 174

## Other Publications

1. **A Transport Layer Technology for Improving the QoS of Networked Multimedia Applications <draft-madisetti-arguriou-qos-sctp-00.txt).**
   *Madisetti, V., Argyriou, A.*
   IETF Internet-Draft, Jul 25, 2002.

2. **Voice & Video over Mobile IP Networks <draft-madisetti-argyriou-voice-video-mip-00.txt>**
   *Madisetti, V., Argyriou, A.*
   IETF Internet-Draft, Nov 20, 2002.

3. **Enhancements to ECRTP with Applications to Robust Header Compression for Wireless Applications**. <**draft-madisetti-rao-suresh-rohc-00.txt>**
   *Madisetti, V.; Rao, S., Suresh, N.*
   IETF Internet-Draft, June 30, 2003.

## Ph.D. Students Graduated

1. **Brian T. Kelley, 1992**
   *VLSI Computing Architectures for High Speed Signal Processing*
   Member of Technical Staff, Motorola.

   Winner of Dr. Thurgood Marshall Dissertation Fellowship Award

2. **Bryce A. Curtis, 1992**
   *Special Instruction Set Multiple Chip Computer for DSP*
   Member of Technical Staff, IBM

3. **Jaejin Lee, 1994**
   *Robust Multitrack Codes for the Magnetic Channel*
   Professor, Yonsei University, Korea

4. **Mohamed S. Ben Romdhane, 1995**
   *Design Synthesis of Application-Specific IC for DSP*
   Director of IP, Rockwell.

5. **Shoab A. Khan, 1995**
   *Logic and Algorithm Partitioning on MCMs*
   Professor, National University of Science & Technology, Pakistan

6. **Lan-Rong Dung, 1997**
   *VHDL-based Conceptual Prototyping of Embedded DSP Architectures*
   Professor, National Chaio Tung University, Taiwan.

Winner of VHDL International Best PhD Thesis Award, 1997

**7. Thomas W. Egolf, 1997**
*Virtual Prototyping of Embedded DSP Systems*
Distinguished Member of Technical Staff, Agere

**8. Alvaro Marenco, 1997**
*On Homomorphic Deconvolution of Bandpass Signals*
Professor, Texas A&M University.

Winner of GIT ECE Outstanding Teaching Assistant Award

**9. Shahram Famorzadeh, 1997**
*BEEHIVE: A Distributed Environment for Adaptive Signal Processing*
Member of Technical Staff, Rockwell.

**10. Timothy J. Klausutis, 1997**
*Adaptive Lapped Transforms with Applications to Image Coding.*
US Air Force/Univ. of Florida.

**11. Lan Shen, 1998**
*Temporal Design of Core-Based Systems*
Member of Technical Staff, IBM

**12. James DeBardelaben, 1998**
*Optimization Based Approach to Cost Effective DSP Design*
Research Scientist, Johns Hopkins University

Georgia Tech ECE Faculty Award

**13. Sangyoun Lee, 1999**
*Design of Robust Video Signal Processors*
Professor, Yonsei University

US Army Sensors Lab Research Excellence Award, 1999

**14. Rahmi Hezar, 2000**
*Oversampled Digital Filters*
Member of Technical Staff, Texas Instruments

**15. Yong-kyu Jung, 2001**
*Model-Based Processor Synthesis*
Professor, Texas A&M University

**16. Mustafa Turkboylari, 2002**
*Handoff Algorithms for Wireless Applications*
Member of Technical Staff, Texas Instruments

**17. Yun-Hui Fan, 2002**
*A Stereo Audio Coder with Nearly Constant Signal to Noise Ratio*
Post-Doctoral Research Associate, Northeastern University

**18. Subrato K. De, 2002**

*Design of a Retargetable Compiler for DSP*
Member of Technical Staff, Qualcomm

US Army Sensors Lab Research Excellence Award, 1999

**19. Chonlameth Aripnikanondt, 2004**
*System-on-Chip Design with UML*
*Professor, King Mongkut's University, Thailand.*

US Army Sensors Lab Research Excellence Award, 1999

**20. Loran Jatunov, 2004**
*Performance Analysis of 3G CDMA Systems*
Senior Research Scientist, Soft Networks, LLC.

**21. Antonios Argyriou, 2005**, Serving in Hellenic Army.

**22. Pilho Kim, 2009, Scientist, VP Technologies, Inc.**

**23. M. Sinnokrot, 2009, Staff Engineer, Qualcomm.**

## Awards & Honors

1. **Jagasdis Bose National Science Talent Fellowship**, Indian Institute of Technology, Kharagpur, 1980-1984.

2. **General Proficiency Prize**, Indian Institute of Technology, Kharagpur, 1984.

3. **Demetri Angelakos Outstanding Graduate Student Award**, Univ. of California, Berkeley, 1989

4. **Ira Kay IEEE/ACM Best Paper Award** for Best Paper presented at IEEE Annual Simulation Symposium, 1989

5. **IBM Faculty Development Award** 1990

6. **Technical Program Chair**, IEEE Workshop on Parallel and Distributed Simulation. 1990.

7. **Technical Program Chair**, IEEE MASCOTS'94

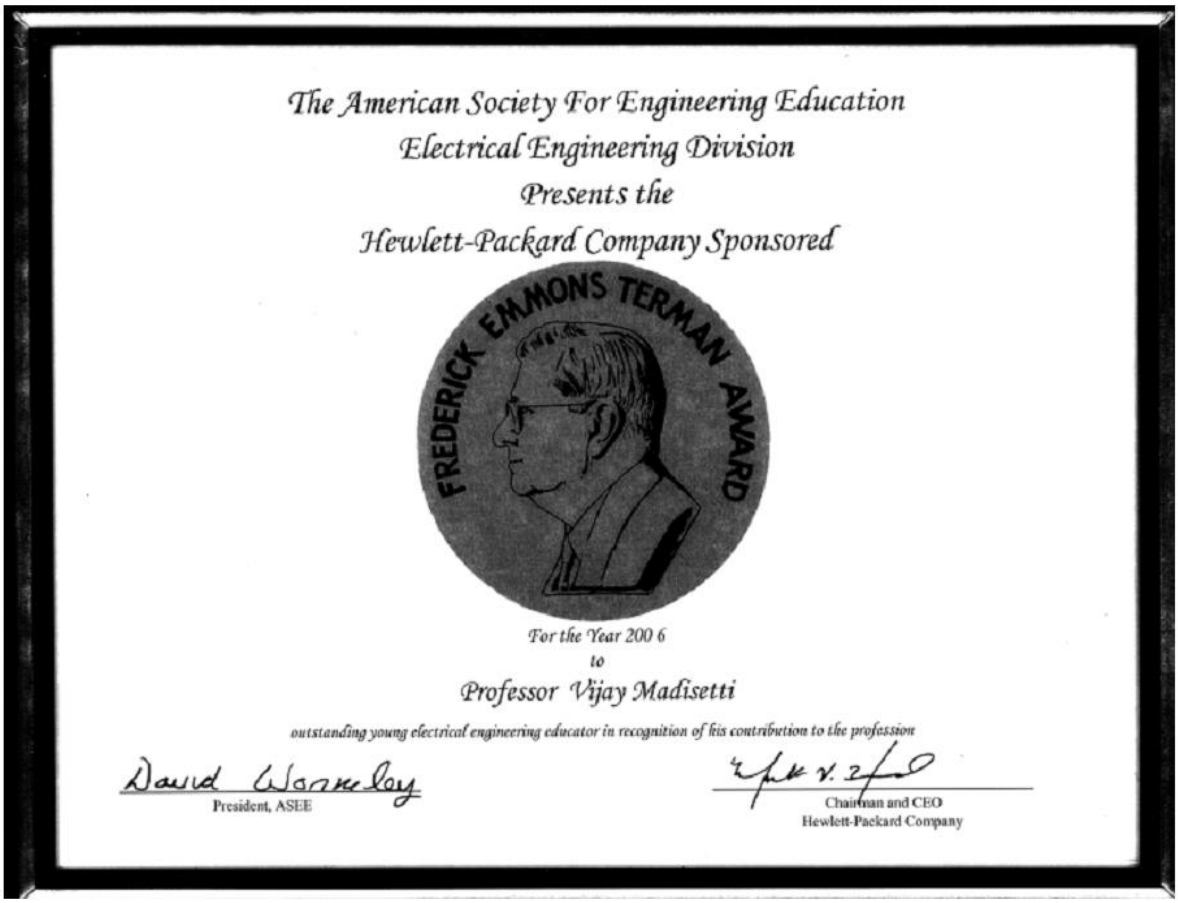
8. **NSF RI Award**, 1990

9. **VHDL International Best PhD Dissertation Advisor Award**, 1997

10. **Georgia Tech Outstanding Doctoral Dissertation Advisor Award**, 2001.

11. **ASEE 2006 Frederick Emmons Terman Medal, 2006.**

12. **Fellow of IEEE**




## Intellectual Property Disclosures (Georgia Tech)

| Patent | Date | Description |
|---|---|---|
| 2843 | 2004 | Method and Apparatus for Improving the Performance of Wireless LANs |
| 2825 | 2003 | Method and Apparatus for Optimal Partitioning and Ordering of Antennas for Layered Space-Time Block Codes in MIMO Communications Systems |
| 2815 | 2003 | How to Rapidly Develop a SyD Application |
| GSU-023 | 2003 | Rapid Development of SyD Applications |

| | | |
|---|---|---|
| 2810 | 2003 | System on Mobile Devices Middleware Design |
| 2718 | 2003 | A Transport Layer Algorithm for Improved Anycast Communication |
| 2717 | 2003 | A Novel Transport Layer Load-Balancing Algorithm |
| 2716 | 2003 | A Transport Layer QoS Algorithm |
| 2715 | 2003 | A Novel Transport Layer Algorithm for MPLS Performance |
| 2659 | 2002 | A New Algorithm and Technology for Implementing Mobile IP with Applications to Voice and Video over Mobile IP |
| 2656 | 2002 | Debugging with Instruction-Level Reverse Execution |
| 2655 | 2002 | Embedded Software Streaming |
| 2539 | | System of Databases: An Enabling Technology for Programming |
| 2517 | 2002 | A Dynamic Instantiated Real-Time Operating System Debugger |
| 2516 | 2002 | A Dynamic Real-Time Operating System |
| GSU-009 | 2001 | System of Databases: Architecture,, Global Queries, Triggers and Constraints |
| 2480 | 2001 | Mobile Fleet Application based on SyD Technology |
| 2479 | 2001 | System of Databases: A model with coordination links and a calendar application |
| 1893 | 1999 | Beehive |
| 1726 | 1995 | Very High Scale Integrated Circuit Hardware Description Language Models (VHDL Models) |
| 1401 | 1995 | Self-Compensation Receiver (SCR) |

**Issued Patents**: 9,935,772, 9,769,213

# APPENDIX

# B

| 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. |