(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
23 May 2019 (23.05.2019)

WIPO | PCT

(10) International Publication Number
WO 2019/098895 A1

(51) International Patent Classification:
*G06F 21/64* (2013.01) *H04N 21/854* (201 1.01)
*G06F3/06* (2006.01) *H04W 12/10* (2009.01)

(21) International Application Number:
PCT/SE20 17/05 1145

(22) International Filing Date:
17 November 2017 (17. 11.2017)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **TELEFONAKTIEBOLAGET LM ERICSSON (PUBL)** [SE/SE]; Torshamnsgatan 23, 164 83 Stockholm (SE).

(72) Inventor: **MEIROSU, Catalin;** Jarvstigen 6, 170 7 1 SOL-NA (SE).

(74) Agent: **BLIDEFALK, Jenny;** Ericsson AB, Patent Unit Kista - Device, Service, Media, Torshamnsgatan23, 164 83 Stockholm (SE).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ,

(54) **Title:** METHOD AND ARRANGEMENT FOR DETECTING DIGITAL CONTENT TAMPERING

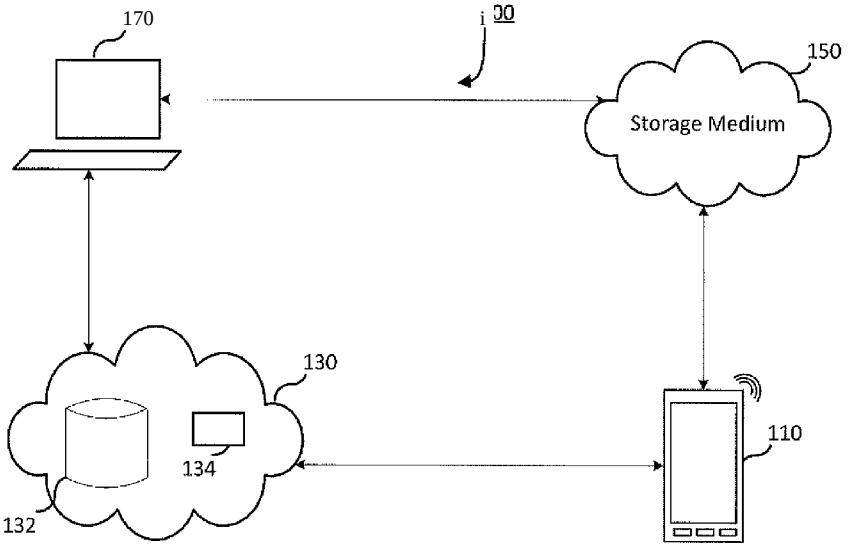


FIG. l a

**(57) Abstract:** A method for detecting tampering of a digital content is provided. The method includes receiving a portion of the digital content at a computing device (710). The method further includes encrypting the portion of the digital content by the computing device when the portion of the digital content contains proof-of-verification (720). The method further includes computing a first hash value for the encrypted portion of the digital content by the computing device (730). The method further includes comparing the computed first hash value with a second hash value associated with the portion of the digital content in the computing device, wherein the second hash value is obtained by the computing device from a distributed ledger system (740). The method further includes identifying the evidence of tampering of the digital content when the computed first hash value is not equal to the second hash value (750). The method further includes providing an alert message to a user or an output medium upon detecting tampering of the digital content (760). A method for creating a tamper-evident digital content is also provided.

WO 2019/098895 A1

*[Continued on next page]*

UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

**Published:**

— *with international search report (Art. 21(3))*

# METHOD AND ARRANGEMENT FOR DETECTING DIGITAL CONTENT TAMPERING

## TECHNICAL FIELD

The invention relates generally to detecting digital content tampering, specifically methods and arrangements for creating tamper-evident digital content and detecting tampering of digital content using the created tamper-evident digital content.

## BACKGROUND

Recent advancements in areas like face rendering, image and sound processing using techniques such as Artificial Intelligence (AI) allows to modify digital content such as videos, both in real-time, and for stored content; for example by inserting or deleting visuals of people and sounds that were not present in the original video footage. These advancements have made it extremely easy to alter digital content, and such modified digital content is appearing with increasing frequency in the public and private domains. The process of detecting whether digital content has been tampered with is extremely difficult and expensive. Different techniques are known and currently used in the area of digital content verification and detection of tampering. While useful in many different contexts, these techniques may provide only a partial solution. No foolproof solutions are available that allow for establishing whether the video has been tampered with, except in the context of detecting copyrighted content. Such solutions do however require exhaustive infrastructure such as encryption keys or alternately steganography.

Blockchain-based solutions are being used in the area of digital content verification and tamper detection. While cryptocurrencies are the most prevalent application today in blockchain-based solutions, the proof-of-creation for documents is an emerging application of blockchain techniques. Currently, the creation of a document can be registered with a blockchain transaction wherein the document is described in the metadata associated to the transaction. Once the transaction is recorded, the blockchain protects against modifications of that transaction information. This

information can be retrieved from the blockchain at a later point in time, which provides a proof of when the document was originally registered. So, a simple way of establishing whether the video has been tampered with would be to store a checksum or hash value on the blockchain at the time when the proof of existence is created. While such a solution could be automated, for example via a script that calculates the value and then verifies it against the blockchain, it is rather time consuming because of the manual work involved in determining the correct parameters for executing the script. Also, these solutions could not be applied in scenarios where verification of live or real-time transmission of digital content needs to be performed.

A simple solution for timestamping video on a proof-of-work blockchain (Bitcoin) was proposed in [1]. The solution focused on trusted timestamping and addressed the video tampering by including a hash in the blockchain transaction; however, the smallest amount of transactional currency needs to be included in the transaction. A method for logging and authenticating content using blockchains was proposed in [3], allowing for the case of video content to register hashes to a blockchain at pre-defined intervals. A method adapted to authenticate individual digital images has been proposed in [4] - it includes storing a hash value in a blockchain to create a timestamped digital record. Furthermore, the proposal in [1] was developed for a marketable cryptocurrency (Bitcoin) that requires proof-of-work. Given the current exchange rates of currencies such as Bitcoin, it is rather expensive to use such solution for example for surveillance cameras - not only would the price of proving the authenticity of the video soon exceed the cost of the camera, but the camera also needs to be re-supplied with coins, which needs to be done in small increments such that it would not become a target for thieves. Improvements suggested in [2], e.g. the aggregation of the transactions on 24 hour batches and submitting them in one aggregate transaction on the blockchain would address some of the cost issues but at the same time would not be applicable to real-time transmissions. Solutions such as [3] and [4] depend on a blockchain, which require proof of work therefore have a high computational overhead and low limits associated with storing data in the blockchain.

In the solutions explained above which are used in the area of digital content verification and tamper detection, none of them cover trusted computing - e.g. in case the device originating the video is compromised and the hash value is intercepted before being stored in the blockchain, an attacker could replace this value with a wanted value for a modified video and register the modified copy as the untampered original video. So, there is need to identify a solution to enable trusted computing in a decentralized manner, thus making it more difficult for attackers to register tampered versions.

## SUMMARY

It is an object of the invention to address at least some of the limitations, problems and issues outlined above. More specifically, it is an object of the invention to create tamper-evident digital content and detect tampering of digital content using it.

These and other objects of the invention are achieved by means of different aspects of the invention, as defined by the independent claims. Embodiments of the invention are characterized by the dependent claims.

According to a first embodiment, a method for creating a tamper-evident digital content is provided. The method comprises the steps of receiving a portion of the digital content at a computing device; encrypting the portion of the digital content by the computing device when the portion of the digital content is selected for proof-of-verification; sending the encrypted portion of the digital content from the computing device to a distributed ledger system; retrieving, by the computing device, hash identification data associated with the encrypted portion of the digital content from the distributed ledger system; and creating, by the computing device, an updated portion of the digital content using the retrieved hash identification data. The method may further comprise the step of storing the updated portion of the digital content in a storage device by the computing device.

The method may further comprise the steps of receiving one or more additional portions of the digital content at the computing device; encrypting one of the additional portions of the digital content by the computing device when that additional portion of

the digital content is selected for proof-of-verification; sending the encrypted additional portion of the digital content from the computing device to the distributed ledger system; retrieving, by the computing device, hash identification data associated with the encrypted additional portion of the digital content from the distributed ledger system; and creating, by the computing device, an updated additional portion of the digital content using the retrieved hash identification data.

In some variants, the computing device is a media encoding device.

In some variants, the digital content comprises media data wherein the media data comprises media frames; and each said portion of the digital content comprises at least one media frame.

In some variants, the digital content comprises MPEG encoded data; and each said portion of the digital content comprises at least one MPEG I-frame.

In embodiments, creating the updated portion of the digital content may further comprise the step of embedding the retrieved hash identification data with metadata associated with the portion of the digital content.

In embodiments, determining if the portion of the digital content is to be selected for proof-of-verification may further comprise the step of using a predefined technique to determine if the first portion of the digital content is to be selected for proof-of-verification wherein the technique is any of: batch sampling technique or simple sampling technique.

In embodiments, sending the encrypted portion of the digital content may further comprise the steps of creating a hash value for the encrypted portion of the digital content received at the distributed ledger system wherein the hash value is created using a smart contract or a stored procedure in the distributed ledger ; and storing the hash value in the distributed ledger system wherein storing the hash value in the distributed ledger system comprises storing the hash value in at least one of the data storage units of the distributed ledger system.

In embodiments, the hash identification data associated with the encrypted first portion of the digital content may further comprise at least a location identifier of the uploaded first hash value; and a type information of the distributed ledger system.

Embodiments of the first aspect may work with real-time digital content streams (e.g., real-time video) as well as regular files. Embodiments may also utilize real-time encrypting, hashing and uploading, which may contribute to a high level of integrity of the uploaded hashes. Embodiments may also utilize immutable databases together with smart contracts, which may also contribute to a high level of integrity of the uploaded hashes

According to second embodiment, a method for detecting tampering of a digital content is provided. The method comprises the steps of receiving a portion of the digital content at the computing device; encrypting the portion of the digital content by the computing device when the portion of the digital content contains proof-of-verification; computing a first hash value for the encrypted portion of the digital content by the computing device; comparing the computed first hash value with a second hash value associated with the portion of the digital content in the computing device, wherein the second hash value is obtained by the computing device from a distributed ledger system; and identifying the evidence of tampering of the digital content when the computed first hash value is not equal to the second hash value. The method may further comprise the step of providing an alert message to a user or an output medium upon detecting tampering of the digital content.

The method may further comprise the steps of receiving one or more additional portions of the digital content at the computing device; encrypting one of the additional portions of the digital content by the computing device when that additional portion of the digital content contains proof-of-verification; computing a first hash value for the encrypted additional portion of the digital content by the computing device; comparing the computed first hash value with a second hash value associated with the additional portion of the digital content in the computing device, wherein the second hash value is obtained by the computing device from the distributed ledger system; and identifying the

evidence of tampering of the digital content when the computed first hash value is not equal to the second hash value.

In some variants, the computing device is a media decoding device.

In some variants, the digital content comprises media data wherein the media data comprises media frames; and each said portion of the digital content comprises at least one media frame.

In some variants, the digital content comprises MPEG encoded data; and each said portion of the digital content comprises at least one MPEG I-frame.

In embodiments, determining if the portion of the digital content contains proof-of-verification may further comprise the steps of retrieving metadata information associated with the portion of the digital content; and verifying the retrieved metadata to identify whether the portion of the digital content contains proof-of-verification.

In embodiments, the second hash value associated with the portion of the digital content is obtained from the distributed ledger system using the metadata information associated with the portion of the digital content.

In some variants, the encrypting may be done using either a symmetric cryptography or an asymmetric cryptography.

In some variants, the distributed ledger system may be a block chain or a database.

According to third embodiment, a computing device for creating a tamper-evident digital content is provided. The computing device comprises: one or more processing circuits; at least one communication interface for communicating with a distributed ledger system and a storage device; and an encoding unit. The encoding unit is configured to: receive a portion of the digital content at the computing device; encrypt the portion of the digital content by the computing device when the portion of the digital content is selected for proof-of-verification; send the encrypted portion of the digital content from the computing device to the distributed ledger system; retrieve hash

identification data associated with the encrypted portion of the digital content from the distributed ledger system; and create an updated portion of the digital content using the retrieved hash identification data. The encoding unit may be further configured to store the updated portion of the digital content in the storage device.

The encoding unit may be further configured to: receive one or more additional portions of the digital content at the computing device; encrypt one of the additional portions of the digital content by the computing device when that additional portion of the digital content is selected for proof-of-verification; send the encrypted additional portion of the digital content from the computing device to the distributed ledger system; retrieve hash identification data associated with the encrypted additional portion of the digital content from the distributed ledger system; and create an updated additional portion of the digital content using the retrieved hash identification data.

In embodiments, the computing device may be a media encoding device.

According to a fourth embodiment, a computing device for creating a tamper-evident digital content is provided. The computing device comprises: one or more processing circuits; at least one communication interface for communicating with a distributed ledger system, an output medium or a user; and a decoding unit. The decoding unit is configured to: receive a portion of the digital content at the computing device; encrypt the portion of the digital content by the computing device when the portion of the digital content contains proof-of-verification; compute a first hash value for the encrypted portion of the digital content by the computing device; compare the computed first hash value with a second hash value associated with the portion of the digital content in the computing device, wherein the second hash value is obtained by the computing device from the distributed ledger system; and identify the evidence of tampering of the digital content when the computed first hash value is not equal to the second hash value. The decoding unit may be further configured to provide an alert message to a user or an output medium upon detecting tampering of the digital content.

The decoding unit may further configured to: receive one or more additional portions of the digital content at the computing device; encrypt one of the additional

portions of the digital content by the computing device when that additional portion of the digital content contains proof-of-verification; compute a first hash value for the encrypted additional portion of the digital content by the computing device; compare the computed first hash value with a second hash value associated with the additional portion of the digital content in the computing device, wherein the second hash value is obtained by the computing device from the distributed ledger system; and identify the evidence of tampering of the digital content when the computed first hash value is not equal to the second hash value.

In embodiments, the computing device may be a media decoding device.

According to fifth embodiment, a computer program product which comprises an embodiment of a computer program according to the first aspect and a computer readable means on which the computer program is stored.

According to sixth embodiment, a computer program product which comprises an embodiment of a computer program according to the second aspect and a computer readable means on which the computer program is stored.

Embodiments allow a user to verify whether a digital content (for e.g. a video) is taken out of context, out of place, or if the digital content (e.g. video) has been tampered with, and provide a user with a tool to test if the digital content (e.g. video) is trustable.

Generally, all terms used in the claims are to be interpreted according to their ordinary meaning in the technical field, unless explicitly defined otherwise herein. All references to "a/an/the element, apparatus, component, means, step, etc." are to be interpreted openly as referring to at least one instance of the element, apparatus, component, means, step, etc., unless explicitly stated otherwise. The steps of any method disclosed herein do not have to be performed in the exact order disclosed, unless explicitly stated. The use of “first”, “second” etc. for different features/components of the present disclosure are only intended to distinguish the features/components from other similar features/components and not to impart any order or hierarchy to the features/components.

Further possible features and benefits of this solution will become apparent from the detailed description below.

## BRIEF DESCRIPTION OF THE DRAWINGS

The invention will now be described in more detail by means of some example embodiments and with reference to the accompanying drawings, in which:

FIG. 1a is a schematic block diagram of an environment where the embodiments presented herein may be applied.

FIG. 1b is a schematic illustration of an environment where the embodiments presented herein may be applied.

FIG. 2 is a schematic flowchart in accordance with some embodiments of the invention.

FIG. 3 is a schematic flowchart in accordance with some embodiments of the invention.

FIG. 4 is a schematic sequence diagram of a method for creating a tamper-evident digital content in accordance with some embodiments of the invention.

FIG. 5 is a schematic sequence diagram of a method for detecting tampering of a digital content in accordance with some embodiments of the invention.

FIG. 6 is a schematic flowchart in accordance with some embodiments of the invention.

FIG. 7 is a schematic flowchart in accordance with some embodiments of the invention.

FIG. 8 is a schematic block diagram of a computing device for creating a tamper-evident digital content, in accordance with an embodiment of the invention.

FIG. 9 is a schematic block diagram of a computing device for detecting tampering of a digital content, in accordance with an embodiment of the invention.

FIG. 10 is a diagram showing functional modules of a computing device for creating a tamper-evident digital content, in accordance with some embodiments of the invention.

FIG. 11 is a diagram showing functional modules of a computing device for detecting tampering of a digital content, in accordance with some embodiments of the invention.

FIG. 12 is a schematic illustration of an embodiment of a computer program product in accordance with some embodiments.

FIG. 13 is a schematic illustration of an embodiment of a computer program product in accordance with some embodiments.

## DETAILED DESCRIPTION

The invention will now be described more fully hereinafter with reference to the accompanying drawings, in which certain embodiments of the invention are shown. However, other embodiments in many different forms are possible within the scope of the present disclosure. Rather, the following embodiments are provided by way of example so that this disclosure will be thorough and complete. Like numbers refer to like elements throughout the description.

More and more mobile phones, tablets, laptop computers, and other portable devices are being equipped with cameras, internet connections, GPS, and various other sensors and accessories. The increasing capabilities of today's smartphones enable the end user to produce and distribute digital content more easily than before. Video, for example, may be taken nearly anywhere, uploaded, and consumed by others in locations remote from the site of the content creation. One issue that has arisen from this is the lack of ability to identify whether the digital content (e.g., video) has been falsified, altered, doctored, modified, and/or forged as compared to the originally created digital content. It is getting increasingly difficult for the viewer to know what to believe, who to believe, and the tools to help counter these issues are scarce and limited. There is a need to identify whether any tampering was done to the original digital content. The

ability to detect any tampering with digital content may be important, for example, to protect individuals or businesses from deceptively edited video, to allow the public to have confidence in videos of public significance (e.g., news events, politics, and so on), and numerous other examples.

Tampering involves the deliberate altering or meddling of the digital content. Tamper-evident describes a device or process or technique that makes unauthorized access or alteration done to the protected object easily detected. Tamper-evident digital content are generally used to detect any unauthorized access or deliberate altering to the digital content and ascertain whether the digital content was altered or not after its creation. Thus, a tamper-evident digital content typically guarantees that any physical or software tampering that can alter the behavior of a digital content is detected. In some variants of tamper-evident digital content cryptographic hash functions and cryptographic signatures are used to add a tamper-evident layer of protection to digital content. In FIG. 1a a schematic block diagram of an environment 100 where the invention presented herein may be applied is illustrated. The environment 100 comprises an input computing device 110 (hereafter referred as “input device”) which is typically a digital content-creation device such as a smartphone, tablet, personal computer, and so forth. The input device 110 may also receive digital content from external resources. The digital content created or received by the input device 110 is processed by an encoding unit. The encoding unit could be present locally on the input device 110 or externally on other devices and/or servers (e.g., a trusted server containing the encoding unit receiving data in real-time from input device 110).

On receiving a portion of the digital content at the input device 110, the encoder unit using any of the configured sampling techniques, such as batch sampling technique or simple sampling technique, identifies whether the portion of the digital content is selected for proof-of-verification. All portions of the digital content received by the encoder unit of the input device 110 are not used for creating the tamper-evident digital content. So, identifying whether the portion of the digital content is selected for proof-of-verification means to determine whether the received portion of the digital content should be stored as part of the proof (after certain processing as explained in the below

steps) on the distributed ledger system or not, while creating the tamper-evident digital content.

If the portion of the digital content is selected for proof-of-verification then the encoding unit encrypts the portion of the digital content using the encryption keys preconfigured in it. Generally, the purpose of encryption is to protect digital data confidentiality as it is stored on computer systems and transmitted using the internet or other computer networks. Encryption is the process of encoding a message or information in such a way that only authorized parties can access it. Data encryption translates data into another form, or code, so that only entities with access to a secret key or password can read it. Encrypted data is commonly referred to as ciphertext, while unencrypted data is called plaintext. Data is encrypted with an encryption algorithm and an encryption key to create ciphertext, which can be viewed in its original form only if it is decrypted with the correct key. Encryption algorithms provide confidentiality of message and guarantee the authentication and integrity of a message. Authentication allows for the verification of a message's origin, and integrity provides proof that a message's contents have not changed since it was sent. Two main types of data encryption exist - asymmetric encryption, also known as public-key encryption, and symmetric encryption, also known as private-key encryption. Symmetric-key cryptography uses the same secret key for encrypting and decrypting a message or file. An asymmetric cryptography uses two different keys, one public and one private. The public key, as it is named, may be shared with everyone, but the private key must be protected. The Rivest-Sharmir-Adleman (RSA) algorithm is a cryptosystem for public-key encryption that is widely used to secure sensitive data, especially when it is sent over an insecure network like the internet.

The encrypted portion of the digital content is then sent from the input device 110 to a distributed ledger system 130 for further processing and storage. A distributed ledger is a database that is consensually replicated, shared and synchronized across network geographically spread across multiple sites, institutions or countries. It can be described as a ledger of any transactions or contracts maintained in decentralized form across different locations and people, eliminating the need of a central authority to keep

a check against manipulation. The participant at each node of the network can access the recordings shared across that network and can own an identical copy of it. Further, any changes or additions made to the ledger are reflected and copied to all participants in a matter of seconds or minutes. All the information on it is securely and accurately stored using cryptography and can be accessed using keys and cryptographic signatures. Once the information is stored, it becomes an immutable database and is governed by the rules of the network. The distributed ledgers are inherently harder to attack because instead of a single database, there are multiple shared copies of the same database, so a cyber-attack would have to attack all the copies simultaneously to be successful.

The distributed ledger system 130 may have more than one data storage unit implemented as blockchain databases or distributed databases. One form of distributed ledger system 130 implementation is the blockchain system, which can be either public or private. Another variant of distributed ledger system 130 implementation is using a trusted database system. The distributed ledger system 130 if implemented as a blockchain ledger, it may also include distributed blockchain databases 132 and smart contracts 134. The distributed ledger system 130 if implemented as a trusted database system, it may also include a database 132 and stored procedures 134. Distributed ledger system 130, blockchain ledger 132 and smart contract 134 may be immutable, i.e. data is not modified after being written, so that data is not changed (a/k/a mutated) after being loaded into database. A smart contract 134 operating with blockchain ledger 132 would hash the encrypted portion of the digital content received from the input device 110 and then store the hashed data in the blockchain ledger 130. In a trusted database 130, a stored procedure 134 operating with database 132 would hash the encrypted portion of the digital content received from the input device 110 and then store the hashed data in the trusted database 130.The hash value generated is stored in at least one of the data storage units 132 of the distributed ledger system 130.

For example, by using a lightweight blockchain infrastructure based on a proof-of-stake or proof-of-authority algorithm or alternately on directed acyclic graphs, the solution provides the advantage of low-overhead and high transaction rate support

during real-time video broadcasting. It enables to perform trusted computing in a decentralized manner, thus making it more difficult for attackers or hackers to register tampered versions of the digital content as an original version as no central repository exists. The use of a private blockchain removes the need to re-supply digital currencies to the content creation devices, thus improving the security of the owners. Furthermore, a lightweight proof of work or proof of stake may allow the content creation device itself to validate transactions and execute smart contracts because of the low demands on computational resources.

Generally, the hashing of data is used to guarantee the integrity and authentication of a message and uniquely represent the data. Hashing is the process of producing hash values for the purpose of accessing data and for security reasons in communication systems. An important characteristic about a hash value is that it is nearly impossible to derive the original input number without knowing the data used to create the hash value Hashing with regard to security is the technique of taking data, encrypting it, and creating unpredictable, irreversible output. Hashing will take arbitrary input and produce a string with a fixed length. Typically hashing techniques will have the following attributes:

(i) A given known input must always produce one known output;

(ii) Once hashing has been done, it should be impossible to go from the output to the input;

(iii) Different multiple inputs should give a different output;

(iv) Modifying an input should mean a change in the hash.

Message Digest 2 (MD2), Message Digest 6 (MD6), Secure Hash Algorithm (SHA), Hashed Message Authentication Code (HMAC) etc. are examples of hashing techniques.

After the hash value of the encrypted portion of the digital content is stored in the distributed ledger system 130, the input device 110 through the encoding unit then retrieves the hash identification data associated with the encrypted portion of the digital content from the distributed ledger system 130. The hash identification data may

comprise at least a location identifier of the uploaded hash value in the distributed ledger system 130; and a type information of the distributed ledger system. The location identifier may be a blockchain identifier i.e. a blockchain-specific address associated to the hash value of the encrypted portion of the digital content. Generally, new location identifiers are assigned for each encrypted portion of the digital content stored in the distributed ledger system 130. The type information of the distributed ledger system 130 may be the network-reachable address of the lightweight chain infrastructure i.e. the distributed ledger system 130 with which the input device 110 is linked with for storing the digital content. The type information could be provided as an HTTP address, IP address and port number, or any other identifier that is reachable over a communication network.

After retrieving the hash identification data associated with the encrypted portion of the digital content from the distributed ledger system 130, the encoding unit creates an updated portion of the digital content by the embedding the retrieved hash identification data with metadata associated with the portion of the digital content. The created updated portion of the digital content is then uploaded to a storage medium 150. The storage medium 150 could be database or a content distribution network (CDN). For example, video content may be uploaded to CDN, a social media site, distributed as live media, or otherwise shared with users. The solution may support real-time digital content (e.g. video) transmission by calculating the hash values online, and performing the distributed ledger transactions while the digital content is being transmitted.

The digital content together with its associated metadata may then be accessed by an end user on an output computing device 170 (hereafter referred as “output device”) from the storage medium 150 (e.g., database, social media platform or content distribution system etc.), and identification of evidence of tampering is done using a decoding unit. The decoding unit could be either executing internally within the output device 170 e.g., through the web client on the output device 170 or the decoding unit could be executing external to the output device 170 e.g., a trusted server containing

the decoding unit receiving digital content together with its associated metadata in real-time from the storage medium 150.

The decoding unit essentially mirrors most of the process performed for the tamper-evident digital data creation on receiving the content at the input device 110 (e.g., as performed by the encoding unit on the input device 110). Because verification mirrors most of the process performed during the time of tamper-evident digital data creation, the decoding unit may need to know certain parameters that were used to encrypt the digital content, hash the encrypted digital content and associated metadata. As one example, one parameter may be to identify the encrypting function or functions to be used (e.g. asymmetric or symmetric cryptographic keys). Another example, one parameter may be to identify the hashing function or functions to be used (e.g., SHA-256, MD6, Pearson hashing, Fletcher checksum). Another example may be to identify the technique used for selecting the digital content for the proof-of verification (e.g. simple sampling or batch sampling). These parameters may be configured on the decoding unit, provided to the decoding unit by the user, provided to the decoding unit as part of the metadata, or otherwise known to the decoding unit. By embedding the above mentioned requisite parameters /identifiers in the metadata of the digital content (e.g. video), the solution enables the automation of the tamper detection process without any human intervention.

The detection of tampering verification by the output device 170 involves communication with the distributed ledger system 130 (e.g., to receive verification data such as hash value by which the digital content may be verified for tampering). The decoding unit on the output device 170 may be connected to the distributed ledger system 130. To verify the digital content, the digital content containing the associated metadata is provided to the decoding unit (e.g., via the storage medium 150).

The decoding unit on receiving a portion of the digital content containing the associated metadata retrieves the metadata associated with the portion of the digital content. The decoding unit on receiving the digital content may involve communication with the distributed ledger system 130 (e.g., to receive hash data by which the digital content may be verified for tampering). Using the retrieved metadata information it

identifies whether the portion of the digital content contains proof-of-verification. If the digital content contains or was selected for proof-of-verification then the decoding unit retrieves the hash value associated with the portion of the digital content from the distributed ledger system 130. During the tamper detection process, the decoding unit encrypts the portion of the digital content and creates a hash value of the encrypted portion of the digital content. The decoding unit then compares the newly created hash value of the encrypted portion of the digital content with the hash value retrieved from the distributed ledger system 130. If the newly created hash value is not equal to the retrieved hash value then evidence of tampering is detected. If evidence of tampering of the digital content is identified then the decoding unit provides an alert message to an external user through the output device 170. If the newly created hash value is equal to the retrieved hash value then no tampering is detected. The output device 170 such as a smartphone, tablet, personal computer, and so forth renders the digital content if no tampering is detected by the decoding unit.

In FIG. 1b, a schematic illustration of an environment 105 in which the embodiments presented herein may be applied is shown. The environment 105 shown in FIG. 1b provides a sample illustration of the implementation of the environment 100 explained above with reference to FIG. 1a.

The video encoder application 120 (hereafter referred as “encoder”) corresponds to the input device 110 in the environment 100. The encoder 120 is a content-creation device for creating the digital content i.e. the video content in this illustration. The encoder 120 captures and transmits video. The encoder 120 can be a video camera (for example it could be a smartphone camera, video recorder, car dashboard camera or surveillance camera, and so forth). The encoder 120 comprises an encoding unit which is present locally on the encoder 120 in this illustration.

The blockchain database 140 (hereafter referred as “blockchain”) corresponds to the distributed ledger system 130 in the environment 100. The blockchain 140 is of lightweight blockchain infrastructure which could be either public blockchain or private blockchain. A public blockchain infrastructure is a blockchain where the nodes 142 that maintain the consensus belong to many administrative domains that have no

agreements between them. Example: Ethereum, NEO, IOTA etc. A private blockchain infrastructure is a blockchain where the nodes 142 that maintain the consensus belong to one administrative domain, or to several administrative domains that have well-defined agreements regarding the blockchain infrastructure. Example: Hyperledger community software deployed within an enterprise, a consortium of enterprises or as-a-Service. A smart contract 144 is deployed in the blockchain 140. Typically, the smart contract 144 cannot perform extensive computational tasks; therefore, its functionality is limited to computing hash values, using cryptographic functions such as SHA-256 etc. The usage of lightweight blockchain infrastructure enables the entire solution to be highly scalable by design. The lightweight blockchain infrastructure typically uses a proof-of-stake technique or a directed acyclic transaction graph with a lightweight proof-of-work and a short time-wise block size, thus enabling the solution to have fast transaction times and high transaction frequency.

The content distribution network (CDN) such as CDN 160 corresponds to the storage medium 150 in the environment 100. For example, video content may be uploaded to CDN 160, a social media site, or distributed as live media etc.

The video decoder application 180 (hereafter referred as “decoder”) corresponds to the output device 170 in the environment 100. The decoder 180 is a content-verification and display device for verifying the digital content i.e. the video content for tampering in this illustration. The decoder 180 receives the video and identifies if tampering is done with respect to the original video. The decoder 180 may be any device that renders the video content for visualization, the decoder 180 may be executed in any platform that allows for visually presenting video frames (for example it could be a smartphone, tablet, personal computer, video display unit and so forth). The decoder 180 comprises a decoding unit which is present locally on the decoder 180 in this illustration.

In FIG. 4, a schematic sequence diagram of a method 400 for creating a tamper-evident digital content in accordance with an embodiment of the invention is illustrated. The method 400 implements the input device 110, the distributed ledger system 130 and the storage medium 150 described above with reference to FIG.1

The method in FIG. 4 comprises the steps of the encoding unit on the input device 110 receiving (step 405) a portion of the digital content. The method further comprises the encoding unit on the input device 110 determining (step 410) whether the portion of the digital content is to be selected for proof-of-verification. After determining, when the portion of the digital content is selected for proof-of-verification, the method further comprises, the encoding unit on the input device 110 encrypting (step 415) the portion of the digital content. The method further comprises, the encoding unit on the input device 110 sending (step 420) the encrypted portion of the digital content to a distributed ledger system 130.

On receiving the encrypted portion of the digital content at the distributed ledger system 130 from the input device 110 the method further comprises, the steps of creating (step 425) a hash value for the encrypted portion of the digital content received at the distributed ledger system 130 wherein the hash value is created using a smart contract 134 or a stored procedure 134 in the distributed ledger system 130. The hash value created is then stored (step 430) in the distributed ledger system wherein storing the hash value in the distributed ledger system comprises storing the hash value in at least one of the data storage units 132 of the distributed ledger system 130.

After sending the encrypted portion of the digital content, the method further comprises the encoding unit on the input device 110 retrieving (step 435) the hash identification data associated with the encrypted portion of the digital content from the distributed ledger system 130. After retrieving the hash identification data, the method further comprises, the encoding unit on the input device 110 creating (step 440) an updated portion of the digital content using the retrieved hash identification data. The method further comprises the step of the encoding unit on the input device 110 storing (step 445) the updated portion of the digital content in a storage medium 150.

In FIG. 8, a schematic block diagram of a computing device 110 for creating a tamper-evident digital content in accordance with method 600 in FIG. 6. The computing device 110 is typically implemented as a content creation device i.e. the input device 110 which comprises an encoding unit, wherein the encoding device could be present locally on the input device 110 or externally on other devices and/or servers as

described above with reference to FIG. 1. The computing device 110 for creating a tamper-evident digital content may comprise:

- one or more processing circuits 830;
- at least one communication interface 810 which is for facilitating the communication with a distributed ledger system 130 and a storage device/medium 150 in the environment 100; and
- an encoding unit 820 is typically configured to perform the different processing functions of the computing device 110

Using the environment 105 (illustrated in FIG. 1b and explained above) and the schematic flowchart 200 (illustrated in FIG. 2), along with reference to the environment 100 (as shown in FIG. 1a and explained above) and the input device 110 (as shown in FIG. 8 and explained above), an example of an implementation of the above method 400 (as shown in FIG. 4 and explained above) for creating a tamper-evident digital content will now be described.

The encoder 120 shown in the environment 105 which is implemented as an input device 110 in the environment 100 (as described above with reference to FIG. 1) may comprise at least one communication interface 810, one or more processing circuits 830 and an encoding unit 820 (as described above with reference to FIG. 8).

As part of the initialization process, the different systems in the environment 105 such as the blockchain 140, the smart contract 144, the encoder 120, and the CDN 160 may be configured (Step 205) initially with the requisite parameters. For example, the address of block chain 140, the address of smart contract 144, the address of CDN 160, the type of encryption and the encryption key used for encryption, the technique used for proof-of-verification, etc. may be pre-configured in the encoder 120.

For example let us consider the scenario wherein, the encoder 120 is a video camera which is recording a video. One or more video frames which form a portion of the video content is received (Step 210) at the encoding unit from the encoder 120.

For each portion of the video content received, the encoding unit needs to determine (Step 215) whether the portion of the video content needs to be selected for proof-of-verification. If the portion of the video content is to be selected for proof-of-verification then the portion of the video content should be stored as proof on the blockchain 140. There are many techniques available to determine whether the portion of the video content is to be selected for proof-of-verification, and some examples in this respect are:

- Simple sampling, wherein every $N^{th}$ frame is selected (where N is either pre-configured as an arbitrary constant, or determined at run-time using a formula provided by the user of the encoder 120)
- Batch sampling, wherein M frames are selected for every N frames (where M and N are either pre-configured as arbitrary constants, or determined at run-time using a formula provided by the user of the encoder 120)
- A pre-learned machine learning or artificial intelligence model is used to identify significant changes in the video frame according to a criteria provided by the user of the encoder 120.

If the portion of the video content is selected for proof-of-verification then the one or more frames selected from the portion of the video content are encrypted (Step 220) by the encoding unit using the encryption keys preconfigured in encoder 120. The encryption is done to preserve the privacy or secrecy of the video content. The encryption may be done using symmetric or asymmetric encryption as configured by the user of the encoder 120.

After the selected portion of the video content is encrypted by the encoding unit, the encrypted portion of the video content is sent (Step 225) by the encoder 120 to the address of the smart contract 144 deployed in the blockchain 140. The smart contract 144 typically cannot perform extensive computational tasks by itself; therefore, its functionality is limited to computing (Step 235) a hash value on the received encrypted portion of the video content, using a cryptographic function such as SHA-256, and store (Step 240) the obtained hash value in the blockchain 140 along with the blockchain specific address associated with the encrypted portion of the video content. After the

computed hash value and the blockchain specific address are stored on the node 142 where the smart contract 144 was executed, the values are distributed to the other nodes 142 in the blockchain 140 for storage through the consensus protocol. The smart contract operating in the blockchain 140 may append a time- stamp to the values stored in the blockchain 140, thus enabling the proof of timestamp verification concept done using the blockchain databases. The solution also enables real-time video transmission by calculating online the hash values and performing the blockchain transactions while the video is being transmitted from the encoder 120.

The encoding unit in the encoder 120 then using the well-known blockchain traversing techniques retrieves (Step 245) from the blockchain 140 the hash identification data associated with the encrypted portion of the video content (encrypted video content which was sent to the smart contract in the previous step). The hash identification data associated with the encrypted portion of the video content may comprise at least a location identifier of the uploaded hash value in the blockchain 140 and a type information of the blockchain 140. Using the hash identification data retrieved from the blockchain 140, the encoding unit in the encoder 120 creates (Step 250) an updated portion of the video content. The encoding unit creates the updated portion of the video content by embedding the retrieved hash identification data with the metadata associated with the portion of the video content (video content which was selected for proof-of-verification in the previous step). The encoding unit during the creation of the updated portion of the video content may also include in the metadata fields an indication on which frame(s) of the portion of the video content were selected for proof-of-verification operation. Thus, the requisite information is embedded by the encoding unit in the metadata fields associated with the portion of the video content to create the updated portion of the video content. Typically, a potential place for this is the Program Element Descriptor Tags in the MPEG standard (tags 234-254 are currently available for such new descriptors). The information generally to be embedded in the metadata fields are:

- type information of the blockchain 140 may be the network-reachable address of the lightweight blockchain infrastructure i.e. the blockchain 140 with which

the encoder 120 is linked with for storing the video content. The type information could be provided as an HTTP address, IP address and port number, or any other identifier that is reachable over a communication network.

- location identifier may be a blockchain identifier i.e. a blockchain-specific address associated to the hash value of the encrypted portion of the video content stored in the blockchain 140 and potentially new identifiers are assigned for each encrypted portion of the video content stored in the blockchain 140

The created updated portion of the video content is then sent by the encoder 120 to the CDN 160 and the updated portion of the video content is stored (Step 255) in the CDN 160.

In the abovementioned step of determining (Step 215) whether the portion of the video content needs to be selected for proof-of-verification, if the portion of the video content is not selected for proof-of-verification then the portion of the video content is sent (it is sent in the same form as it was received at the encoding unit) by the encoder 120 to the CDN 160. The portion of the video content is stored (Step 260) in the CDN 160.

After storing, either the updated portion of the video content (Step 255) or the initial received portion of the video content (Step 260) in the CDN 160, the encoding unit in the encoder 120 determines (Step 265) whether the end of the video content has been reached i.e. it checks if any more additional portions of the video content are available in the encoder 120. If more additional portions of the video content are available in the encoder 120, then the above mentioned steps (Step 210 to 265) are performed for each of the additional portions of the video content available in the encoder 120. So, the method 200, after receiving the initial portion of the video content (Step 210) at the encoding unit from the encoder 120 enters into a loop, during which the video content is processed until there is some indication that the end of the video content has been reached, e.g. an end-of-file (EOF) bit (loop at Step 210-265). If no

further portions of the video content are available in the encoder 120, then the method 200 for creating a tamper-evident video content is ended.

In FIG. 5, a schematic sequence diagram of a method 500 for detecting tampering of a digital content in accordance with an embodiment of the invention is illustrated. The method 500 implements the output device 170, the distributed ledger system 130 and the storage medium 150 described above with reference to FIG.1

The method in FIG. 5 comprises the steps of the decoding unit on the output device 170 receiving (step 505) a portion of the digital content from the storage medium 150. The method further comprises the decoding unit on the output device 170 retrieving (step 510) the metadata associated with the received portion of the digital content. The method further comprises the decoding unit on the output device 170 determining (step 515) whether the portion of the digital content contains proof-of-verification. After determining, when the portion of the digital content is selected for proof-of-verification, the decoding unit on the output device 170 retrieves (step 520) the hash value associated with the portion of the digital content from the distributed ledger system 130

The method further comprises, the decoding unit on the output device 170 encrypting (step 525) the portion of the digital content. After encrypting the portion of the digital content, the method further comprises, the decoding unit on the output device 170 computing (step 530) a hash value for the encrypted portion of the digital content. The method further comprises, the decoding unit on the output device 170 comparing 535 the computed hash value of the encrypted portion of the digital content with the retrieved hash value associated with the portion of the digital content. The method further comprises, when the computed hash value of the encrypted portion of the digital content is not equal to the retrieved hash value associated with the portion of the digital content then identifying (step 540) the evidence of tampering of the digital content and method may further comprise the step of providing an alert message to a user or an output medium about detecting tampering of the digital content.

In FIG. 9, a schematic block diagram of a computing device 170 for detecting tampering of a digital content in accordance with in accordance with method 700 in FIG. 7. The computing device 170 is typically implemented as a content output/display device i.e. the output device 170 which comprises of a decoding unit, wherein the decoding unit which could be present locally on the output device 170 or externally on other devices and/or servers as described above with reference to FIG. 1. The computing device 900 for detecting tampering of a digital content may comprise:

- one or more processing circuits 930;
- at least one communication interface 910 which is for facilitating the communication with an output medium/device 170, a distributed ledger system 130 and a storage device/medium 150 in the environment 100; and
- a decoding unit 920 is typically configured to perform the different processing functions of the computing device 170 .

Using the environment 105 (illustrated in FIG. 1b and explained above) and the schematic flowchart 300 (illustrated in FIG. 3), along with reference to the environment 100 (as shown in FIG. 1a and explained above) and the output device 170 (as shown in FIG. 9 and explained above), an example of an implementation of the above method 500 (as shown in FIG. 5 and explained above) for detecting tampering of a digital content will now be described.

The decoder 180 shown in the environment 105 which is implemented as an output device 170 in the environment 100 (as described above with reference to FIG. 1) may comprise at least one communication interface 910, one or more processing circuits 930 and a decoding unit 920 (as described above with reference to FIG. 9).

The decoder 180 in effect mirrors most of the steps performed at the encoder 120 for creating the tamper-evident video data. Since for verification the decoder 180 executes many of the process performed during the time of tamper-evident video data creation, the decoding unit in the decoder 180 may need to know certain parameters that were used to encrypt the video content, hash the encrypted video content, the technique used for proof-of-verification, etc.

As part of the initialization process, the different systems in the environment 105 such as the blockchain 140, the decoder 180, and the CDN 160 may be configured (Step 305) initially with the requisite parameters. For example, the address of block chain 140, the address of CDN 160, the type of encryption and the encryption key used for encryption, the type of hashing, the technique used for proof-of-verification, etc. may be pre-configured in the decoder 180.

For example let us consider the scenario wherein the decoder 180 is a video display unit (VDU) which obtains the video from the CDN 160 and displays it to a user of the system. One or more video frames which form a portion of the video content is received (Step 310) at the decoding unit in the decoder 120 from the CDN 160.

For each portion of the video content received from the CDN 160, the decoding unit in the decoder 180 retrieves (Step 315) the metadata associated with the portion of the video content. The metadata retrieved along with the usual video metadata information may include the details about the type information of the blockchain 140, the location identifier may be a blockchain identifier etc. which were embedded into the video content by the encoder 120 before storing into the CDN 180, during the tamper-evident video content creation process as explained above.

For each portion of the video content received, the decoding unit in the decoder 180 needs to determine (Step 320) whether the portion of the video content was selected for proof-of-verification. The decoding unit in the decoder 180 determines whether the portion of the video content was selected for proof-of-verification by using the retrieved metadata associated with the portion of the video content. The metadata may contain an indication on whether frame(s) of the portion of the video content were selected for proof-of-verification operation. The metadata may contain the details about the technique that was used in the encoder 120 to determine whether the portion of the video content is to be selected for proof-of-verification. Then, using the same technique details available in the retrieved metadata, the decoding unit may determine whether the portion of the video content was selected for proof-of-verification. Alternatively, the metadata may contain identification information which indicates the portion of the video content was selected for proof-of-verification.

If the portion of the video content was selected for proof-of-verification then the hash value associated with the portion of the video content is retrieved (Step 325) from the blockchain 140 by the decoding unit using the information available in the retrieved metadata associated with the portion of the video content.

After retrieving the hash value associated with the portion of the video content, the decoding unit in the decoder 180 encrypts (Step 330) the one or more frames of the portion of the video content which were selected for proof-of-verification. The encryption is done by the decoding unit using the same encryption process and the encryption keys which was used in the encoder 120 while creating the tamper-evident video content. The encryption details may be available as part of the metadata associated with the portion of the video content or it may be configured by the user of the decoder 120 and so forth.

The decoding unit in the decoder 180 then computes (Step 335) the hash value of the encrypted portion of the video content using the same hashing technique as used in the smart contract 144 during the tamper-evident video content creation process. The hashing details may be available as part of the metadata associated with the portion of the video content or it may be configured by the user of the decoder 120 and so forth.

The decoding unit in the decoder 180 then compares (Step 340) the newly created hash value of the encrypted portion of the video content with the hash value associated with the portion of the video content retrieved from the blockchain 140.

If the newly created hash value is equal to the retrieved hash value then no tampering is detected. The decoder 180 (such as a smartphone, tablet, personal computer, and so forth) renders (Step 345) the portion of the video content to a user or on an output display, if no tampering is detected by the decoding unit.

If the newly created hash value is not equal to the retrieved hash value then evidence of tampering is detected. On identifying an evidence of tampering of the video content then the decoding unit in the decoder 180 provides (Step 355) an alert message to an external user through the decoder 180.

The decoder 180 while rendering the video content may display a visual cue that corresponds to whether the video content was tampered (when the created hash value is not equal to the retrieved hash value). This cue could be overlaid on the video content, or it could be appearing somewhere in the display of the decoder i.e. the graphical user interface. When the created hash value is not equal to the retrieved hash, the decoder 180 may also enter into an error logging device the details about the exact frames from the video content for which the created hash value is not equal to the retrieved hash value. Thus, by defining and identifying the length of the frame(s) that are selected for the proof-of verification using the different techniques (such as simple sampling, batch sampling etc.) the solution enables in determining exactly which portion of the video content was tampered with.

Also, in the abovementioned step of determining (Step 320) whether the portion of the video content was selected for proof-of-verification, if the portion of the video content was not selected for proof-of-verification then the portion of the video content (as it was received at the decoding unit) is rendered (Step 345) on to an user or on an output display by the decoder 180.

After rendering the portion of the video content to a user or on an output display, the decoding unit in the decoder 180 determines (Step 350) whether the end of the video content has been reached i.e. it checks if any more additional portions of the video content are available in the decoder 180. If more additional portions of the video content are available in the decoder 180, then the above mentioned steps (Step 310 to 355) are performed for each of the additional portions of the video content available in the decoder 180. So, the method 300, after receiving the initial portion of the video content (Step 310) at the decoding unit in the decoder 180 from the CDN 160 enters into a loop, during which the video content is processed until there is some indication that the end of the video content has been reached (e.g., an end-of-file (EOF) bit) (loop at Step 310-355). If no further portions of the video content are available in the decoder 180, then the method 300 for detecting tampering of the video content is ended (Step 360).

After the decoding unit in the decoder 180 provides an alert message about the video tampering identified, the decoding unit in the decoder 180 may either (i) terminate/end (Step 360) the method 300 for detecting tampering of the video content or, (ii) determine (Step 350) whether the end of the video content has been reached i.e. it checks if any more additional portions of the video content are available in the decoder 180 and continue with the tamper detection (as explained above) for the additional portions of the video content if any are available.

FIG. 6 is a schematic flowchart illustrating a process 600 in accordance with some embodiments of the invention. Process 600 is a method for creating tamper-evident digital content. The method includes receiving a portion of the digital content at a computing device (step 610). The computing device may be a media encoding device. The digital content may include media data wherein the media data may comprise media frames; and each said portion of the digital content may comprise at least one media frame. The media data may comprise video data, which could be encoded in various formats, one example being MPEG. So, the digital content may include MPEG encoded data; and each said portion of the digital content may include at least one MPEG I-frame. The method further includes encrypting the portion of the digital content by the computing device when the portion of the digital content is selected for proof-of-verification (step 620). The step of determining if the portion of the digital content is to be selected for proof-of-verification may further include the step of using a predefined technique to determine if the portion of the digital content is to be selected for proof-of-verification wherein the technique is any of: batch sampling technique or simple sampling technique. The encrypting may be done using either a symmetric cryptography or an asymmetric cryptography. The method further includes sending the encrypted portion of the digital content from the computing device to a distributed ledger system (step 630). The distributed ledger system may be a block chain or a database. The method further includes retrieving, by the computing device, hash identification data associated with the encrypted portion of the digital content from the distributed ledger system (step 640). The hash identification data associated with the encrypted first portion of the digital content may further include at least a location identifier of the uploaded first hash value; and a type information of the distributed ledger system. The

method further includes creating, by the computing device, an updated portion of the digital content using the retrieved hash identification data (step 650). The step of creating the updated portion of the digital content may further include the step of embedding the retrieved hash identification data with metadata associated with the portion of the digital content. The method further includes storing the updated portion of the digital content in a storage device by the computing device (step 660).

In some embodiments, the method further includes (1) receiving one or more additional portions of the digital content at the computing device; (2) encrypting one of the additional portions of the digital content by the computing device when that additional portion of the digital content is selected for proof-of-verification; (3) sending the encrypted additional portion of the digital content from the computing device to the distributed ledger system; (4) retrieving, by the computing device, hash identification data associated with the encrypted additional portion of the digital content from the distributed ledger system; and (5) creating, by the computing device, an updated additional portion of the digital content using the retrieved hash identification data.

.In some embodiments, sending the encrypted portion of the digital content may further include the steps of (1) creating a hash value for the encrypted portion of the digital content received at the distributed ledger system wherein the hash value is created using a smart contract or a stored procedure in the distributed ledger ; and (2) storing the hash value in the distributed ledger system wherein storing the hash value in the distributed ledger system comprises storing the hash value in at least one of the data storage units of the distributed ledger system.

.

FIG. 7 is a schematic flowchart illustrating a process 700 in accordance with some embodiments of the invention. Process 700 is a method for detecting tampering of a digital content. The method includes receiving a portion of the digital content at a computing device (step 710). The computing device may be a media decoding device. The digital content may include media data wherein the media data may comprise media frames; and each said portion of the digital content may comprise at least one

media frame. The media data may comprise video data, which could be encoded in various formats, one example being MPEG. So, the digital content may include MPEG encoded data; and each said portion of the digital content may include at least one MPEG I-frame. The method further includes encrypting the portion of the digital content by the computing device when the portion of the digital content contains proof-of-verification (step 720). The step of determining if the portion of the digital content contains proof-of-verification may further include the steps of (1) retrieving metadata information associated with the portion of the digital content; and (2) verifying the retrieved metadata to identify whether the portion of the digital content contains proof-of-verification. The encrypting may be done using either a symmetric cryptography or an asymmetric cryptography. The method further includes computing a first hash value for the encrypted portion of the digital content by the computing device (step 730). The method further includes comparing the computed first hash value with a second hash value associated with the portion of the digital content in the computing device, wherein the second hash value is obtained by the computing device from a distributed ledger system (step 740). The second hash value associated with the portion of the digital content is obtained from the distributed ledger system using the metadata information associated with the portion of the digital content. The distributed ledger system may be a block chain or a database. The method further includes identifying the evidence of tampering of the digital content when the computed first hash value is not equal to the second hash value (step 750). The method further includes providing an alert message to a user or an output medium upon detecting tampering of the digital content (step 760).

In some embodiments, the method further includes (1) receiving one or more additional portions of the digital content at the computing device; (2) encrypting one of the additional portions of the digital content by the computing device when that additional portion of the digital content contains proof-of-verification; (3) computing a first hash value for the encrypted additional portion of the digital content by the computing device; (4) comparing the computed first hash value with a second hash value associated with the additional portion of the digital content in the computing device, wherein the second hash value is obtained by the computing device from the

distributed ledger system; and (5) identifying the evidence of tampering of the digital content when the computed first hash value is not equal to the second hash value.

FIG. 10 is a diagram showing functional modules of a computing device 110 for creating a tamper-evident digital content, in accordance with some embodiments of the invention. As shown in FIG. 10, a computing device 110 includes a receiving module 1010, an encrypting module 1020, a verifying module 1030, a sending module 1040, a retrieving module 1050 and a creating module 1060. Receiving module 1010 is configured to receive a portion of the digital content. Encrypting module 1020 is configured to encrypt the portion of the digital content when the portion of the digital content is selected for proof-of-verification. Sending module 1040 is configured to send the encrypted portion of the digital content from the computing device 1000 to a distributed ledger system. Retrieving module 1050 is configured to retrieve hash identification data associated with the encrypted portion of the digital content from the distributed ledger system. Creating module 1060 is configured to create an updated portion of the digital content using the retrieved hash identification data. Sending module 1040 is further configured to store the updated portion of the digital content in a storage device.

FIG. 11 is a diagram showing functional modules of a computing device 170 for detecting tampering of a digital content, in accordance with some embodiments of the invention. As shown in FIG. 10, a computing device 170 includes a receiving module 1110, an obtaining module 1120, an encrypting module 1130, a verifying module 1140, a computing module 1150 and a sending module 1160. Receiving module 1110 is configured to receive a portion of the digital content at the computing device. Encrypting module 1130 is configured to encrypt the portion of the digital content when the portion of the digital content contains proof-of-verification. Computing module 1150 is configured to compute a first hash value for the encrypted portion of the digital content Verifying module 1140 is configured to compare the computed first hash value with a second hash value associated with the portion of the digital content in the computing device, wherein the obtaining module 1120 is configured to obtain the second hash value from the distributed ledger system. Verifying module 1140 is further configured to

identify the evidence of tampering of the digital content when the computed first hash value is not equal to the second hash value. Sending module 1160 is configured to provide an alert message to a user or an output medium upon detecting tampering of the digital content.

In FIG. 12, a schematic illustration of an embodiment of a computer program product is illustrated. It shows one example of a computer program product 1210 comprising a computer readable storage medium 1230. On this computer readable storage medium 1230, a computer program 1220 can be stored, which computer program 1220 can cause the computing device 110, i.e. the input device 110 and thereto operatively coupled entities and devices, such as the encoding unit 820, to execute methods 200, 600 described herein. The computer program 1220 and/or computer program product 1210 may thus provide means for performing any steps as herein disclosed.

In the example of FIG. 12, the computer program product 1210 is illustrated as an optical disc, such as a CD (compact disc) or a DVD (digital versatile disc) or a Blu-Ray disc. The computer program product 1210 could also be embodied as a memory, such as a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM), or an electrically erasable programmable read-only memory (EEPROM) and more particularly as a non-volatile storage medium of a device in an external memory such as a USB (Universal Serial Bus) memory or a Flash memory, such as a compact Flash memory. Thus, while the computer program 1220 is here schematically shown as a track on the depicted optical disk, the computer program 1220 can be stored in any way which is suitable for the computer program product 1210.

In FIG. 13, a schematic illustration of an embodiment of a computer program product is illustrated. It shows one example of a computer program product 1310 comprising computer readable storage medium 1330. On this computer readable storage medium 1330, a computer program 1320 can be stored, which computer program 1320 can cause the computing device 170 i.e. the output device 170 and thereto operatively coupled entities and devices, such as the decoding unit 920, to

execute methods 300, 700 described herein. The computer program 1320 and/or computer program product 1310 may thus provide means for performing any steps as herein disclosed.

In the example of FIG. 13, the computer program product 1310 is illustrated as an optical disc, such as a CD (compact disc) or a DVD (digital versatile disc) or a Blu-Ray disc. The computer program product 1310 could also be embodied as a memory, such as a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM), or an electrically erasable programmable read-only memory (EEPROM) and more particularly as a non-volatile storage medium of a device in an external memory such as a USB (Universal Serial Bus) memory or a Flash memory, such as a compact Flash memory. Thus, while the computer program 1320 is here schematically shown as a track on the depicted optical disk, the computer program 1320 can be stored in any way which is suitable for the computer program product 1310.

The invention has mainly been described above with reference to a few embodiments. However, as is readily appreciated by a person skilled in the art, other embodiments than the ones disclosed above are equally possible within the scope of the inventive concept, as defined by the appended patent claims.

## REFERENCES

1. B. Gipp, N. Meuschke, and A. Gernandt, “Decentralized Trusted Timestamping using the Crypto Currency Bitcoin,” inProceedings of the iConference 2015, Newport Beach, California, 2015.
2. Breitinger, Corinna, Gipp, Bela. Virtual Patent: Enabling the Traceability of Ideas Shared Online Using Decentralized Trusted Timestamping in Proceedings of the 15th International Symposium on Information Science (ISI2017), pp. 89-95, http://dx.doi.Org/10.18452/1442
3. US20160283920 - Authentication and verification of digital data utilizing blockchain technology

4. US201 60379330 - Method and apparatus for protecting digital photos from alteration

## CLAIMS

1. A method (600) for creating a tamper-evident digital content, wherein the method comprises:
    - Receiving (610) a portion of the digital content at a computing device (110);
    - Encrypting (620) the portion of the digital content by the computing device (110) when the portion of the digital content is selected for proof-of-verification;
    - Sending (630) the encrypted portion of the digital content from the computing device (110) to a distributed ledger system (130);
    - Retrieving (640) hash identification data associated with the encrypted portion of the digital content from the distributed ledger system (130); and
    - Creating (650) an updated portion of the digital content using the retrieved hash identification data.
2. The method of claim 1, further comprising storing (660) the updated portion of the digital content in a storage device (150).
3. The method according to claim 1 or 2, wherein creating the tamper-evident digital content further comprises :
    - Receiving one or more additional portions of the digital content at the computing device (110) ;
    - Encrypting one of the additional portions of the digital content by the computing device (110) when that additional portion of the digital content is selected for proof-of-verification;
    - Sending the encrypted additional portion of the digital content from the computing device (110) to the distributed ledger system (130);
    - Retrieving hash identification data associated with the encrypted additional portion of the digital content from the distributed ledger system (130); and
    - Creating an updated additional portion of the digital content using the retrieved hash identification data.

4. The method according to claim 1-3, wherein the computing device (110) is a media encoding device.
5. The method according to any of the preceding claims, wherein the digital content comprises media data wherein the media data comprises media frames; and each said portion of the digital content comprises at least one media frame.
6. The method according to any of the preceding claims, wherein the digital content comprises MPEG encoded data; and each said portion of the digital content comprises at least one MPEG I-frame.
7. The method according to any of the preceding claims, wherein creating the updated portion of the digital content further comprises embedding the retrieved hash identification data with metadata associated with the portion of the digital content.
8. The method according to any of the preceding claims, further comprising determining if the portion of the digital content is to be selected for proof-of-verification using a predefined technique to determine if the first portion of the digital content is to be selected for proof-of-verification wherein the technique is any of: batch sampling technique or simple sampling technique.
9. The method according to any of the preceding claims, wherein sending the encrypted portion of the digital content further comprises :
    - Creating a hash value for the encrypted portion of the digital content received at the distributed ledger system (130) wherein the hash value is created using a smart contract (134) or a stored procedure (134) in the distributed ledger system (130); and
    - Storing the hash value in the distributed ledger system (130) wherein storing the hash value in the distributed ledger system (130) comprises storing the hash value in at least one of the data storage units (132) of the distributed ledger system (130).
10. The method according to any of the preceding claims, wherein the hash identification data associated with the encrypted first portion of the digital content further comprises at least a location identifier of the uploaded first hash value; and a type information of the distributed ledger system (130).

11. A method (700) for detecting tampering of a digital content, wherein the method comprises:
    - Receiving (710) a portion of the digital content at a computing device (170);
    - Encrypting (720) the portion of the digital content by the computing device (170) when the portion of the digital content contains proof-of-verification;
    - Computing (730) a first hash value for the encrypted portion of the digital content by the computing device (170);
    - Comparing (740) the computed first hash value with a second hash value associated with the portion of the digital content in the computing device (170), wherein the second hash value is obtained by the computing device (170) from a distributed ledger system (130); and
    - Identifying (750) the evidence of tampering of the digital content when the computed first hash value is not equal to the second hash value.
12. The method of claim 11, further comprising providing (760) an alert message to a user or an output medium upon detecting tampering of the digital content.
13. The method according to claim 11 or 12, wherein detecting tampering of the digital content-further comprises :
    - Receiving one or more additional portions of the digital content at the computing device (170);
    - Encrypting one of the additional portions of the digital content by the computing device (170) when that additional portion of the digital content contains proof-of-verification;
    - Computing a first hash value for the encrypted additional portion of the digital content by the computing device (170);
    - Comparing the computed first hash value with a second hash value associated with the additional portion of the digital content in the computing device (170), wherein the second hash value is obtained by the computing device from the distributed ledger system (130); and
    - Identifying the evidence of tampering of the digital content when the computed first hash value is not equal to the second hash value.

14. The method according to claim 11-13, wherein the computing device (170) is a media decoding device.
15. The method according to any of the preceding claims, wherein the digital content comprises media data wherein the media data comprises media frames; and **each said portion of the digital content comprises at least one media frame.**
16. The method according to any of the preceding claims, wherein the digital content comprises MPEG encoded data; and each said portion of the digital content comprises at least one MPEG I-frame.
17. The method according to any of the preceding claims, further comprising determining if the portion of the digital content contains proof-of-verification by:
    - Retrieving a metadata information associated with the portion of the digital content; and
    - Verifying the retrieved metadata to identify whether the portion of the digital content contains proof-of-verification.
18. The method according to any of the preceding claims, wherein the second hash value associated with the portion of the digital content is obtained from the distributed ledger system (130) using the metadata information associated with the portion of the digital content.
19. The method according to any of the preceding claims, wherein encrypting is done using either a symmetric cryptography or an asymmetric cryptography.
20. The method according to any of the preceding claims, wherein the distributed ledger system (130) is a block chain or a database.
21. A computing device (110) for creating a tamper-evident digital content, the computing device (110) comprising:
    - one or more processing circuits (830);
    - at least one communication interface (810) for communicating with a distributed ledger system (130) and a storage device (150);
    - an encoding unit (820)configured to:
        - Receive a portion of the digital content at the computing device (110);

- Encrypt the portion of the digital content by the computing device (110) when the portion of the digital content is selected for proof-of-verification;
- Send the encrypted portion of the digital content from the computing device (110) to the distributed ledger system (130);
- Retrieve hash identification data associated with the encrypted portion of the digital content from the distributed ledger system (130); and
- Create an updated portion of the digital content using the retrieved hash identification data.

22. The computing device (110) of claim 21, wherein the encoding unit (820) is further configured to store the updated portion of the digital content in the storage device (150).

23. The computing device (110) according to claim 20 or 21, wherein the encoding unit (820) is further configured to:

- Receive one or more additional portions of the digital content at the computing device (110);
- Encrypt one of the additional portions of the digital content by the computing device (110) when that additional portion of the digital content is selected for proof-of-verification;
- Send the encrypted additional portion of the digital content from the computing device (110) to the distributed ledger system (130);
- Retrieve hash identification data associated with the encrypted additional portion of the digital content from the distributed ledger system (130); and
- Create an updated additional portion of the digital content using the retrieved hash identification data.

24. The computing device (110) according to claim 20-23, wherein the computing device (110) is a media encoding device.

25. The computing device (110) according to any of the preceding claims, wherein the digital content comprises media data wherein the media data comprises of

media frames; and each said portion of the digital content comprises at least one media frame.

26. The computing device (110) according to any of the preceding claims, wherein the digital content comprises MPEG encoded data; and each said portion of the digital content comprises at least one MPEG I-frame.

27. The computing device (110) according to claim 21, wherein the encoding unit (820) is further configured to perform the method of any one of claims 7-10.

28. A computing device (170) for detecting tampering of a digital content, the computing device (170) comprising:

- one or more processing circuits (930);
- at least one communication interface (910) for communicating with a distributed ledger system (130) , an output medium or an user;
- a decoding unit (920) configured to:
  - Receive a portion of the digital content at the computing device (170);
  - Encrypt the portion of the digital content by the computing device (170) when the portion of the digital content contains proof-of-verification;
  - Compute a first hash value for the encrypted portion of the digital content by the computing device (170);
  - Compare the computed first hash value with a second hash value associated with the portion of the digital content in the computing device (170), wherein the second hash value is obtained by the computing device (170) from the distributed ledger system (130); and
  - Identify the evidence of tampering of the digital content when the computed first hash value is not equal to the second hash value.

29. The computing device (170) of claim 28, wherein the decoding unit (920) is further configured to provide an alert message to an user or an output medium upon detecting tampering of the digital content.

30. The computing device (170) according to claim 28 or 29, the decoding unit (920) is further configured to:
    - Receive one or more additional portions of the digital content at the computing device (170);
    - Encrypt one of the additional portions of the digital content by the computing device (170) when that additional portion of the digital content contains proof-of-verification;
    - Compute a first hash value for the encrypted additional portion of the digital content by the computing device (170);
    - Compare the computed first hash value with a second hash value associated with the additional portion of the digital content in the computing device (170), wherein the second hash value is obtained by the computing device (170) from the distributed ledger system (130); and
    - Identify the evidence of tampering of the digital content when the computed first hash value is not equal to the second hash value.
31. The computing device (170) according to claim 28-30, wherein the computing device (170) is a media decoding device.
32. The computing device (170) according to any of the preceding claims, wherein the digital content comprises media data wherein the media data comprises of media frames; and each said portion of the digital content comprises at least one media frame.
33. The computing device (170) according to any of the preceding claims, wherein the digital content comprises MPEG encoded data; and each said portion of the digital content comprises at least one MPEG I-frame.
34. The computing device (170) according to claim 28, wherein the decoding unit (920) is further configured to perform the method of any one of claims 17-20.
35. A computer program (1220) comprising computer-executable instruction for causing a computing device (110) to perform the method according to any one of claims 1 to 10, when the computer-executable instructions are executed on a processor comprised in the computing device (110).

36. A computer program product (1210) comprising a computer program (1220) according to claim 35 and a computer readable means (1230) on which the computer program (1220) is stored.
37. A computer program (1320) comprising computer-executable instruction for causing a computing device (170) to perform the method according to any one of claims 11 to 20, when the computer-executable instructions are executed on a processor comprised in the computing device (170).
38. A computer program product (1310) comprising a computer program (1320) according to claim 37 and a computer readable means (1330) on which the computer program (1320) is stored.

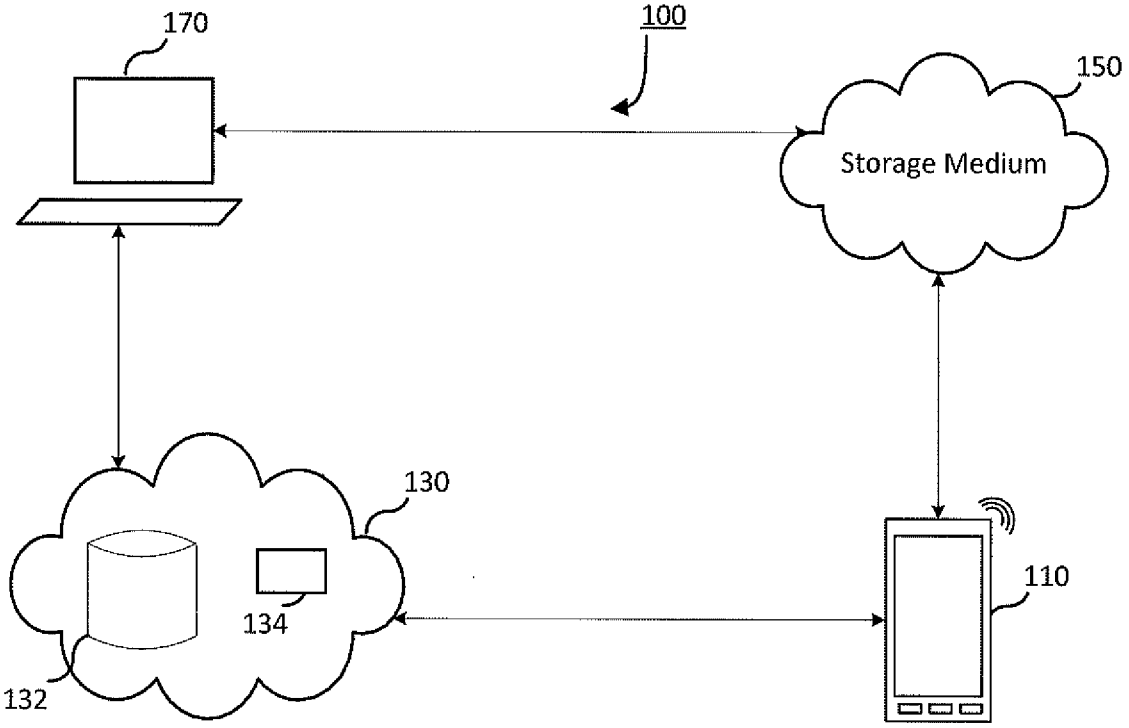
170
100
150
Storage Medium
130
134
132
110

FIG. 1a

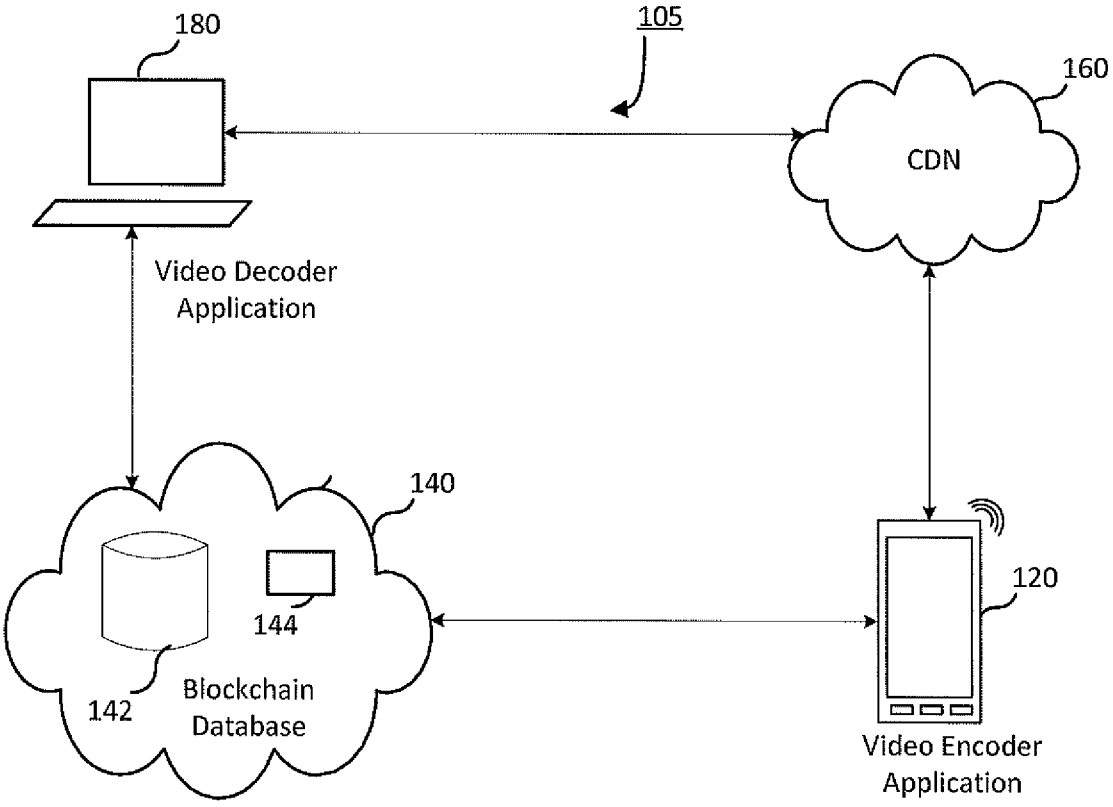
180
105
160
CDN
Video Decoder
Application
140
144
142
Blockchain
Database
120
Video Encoder
Application

FIG. 1b

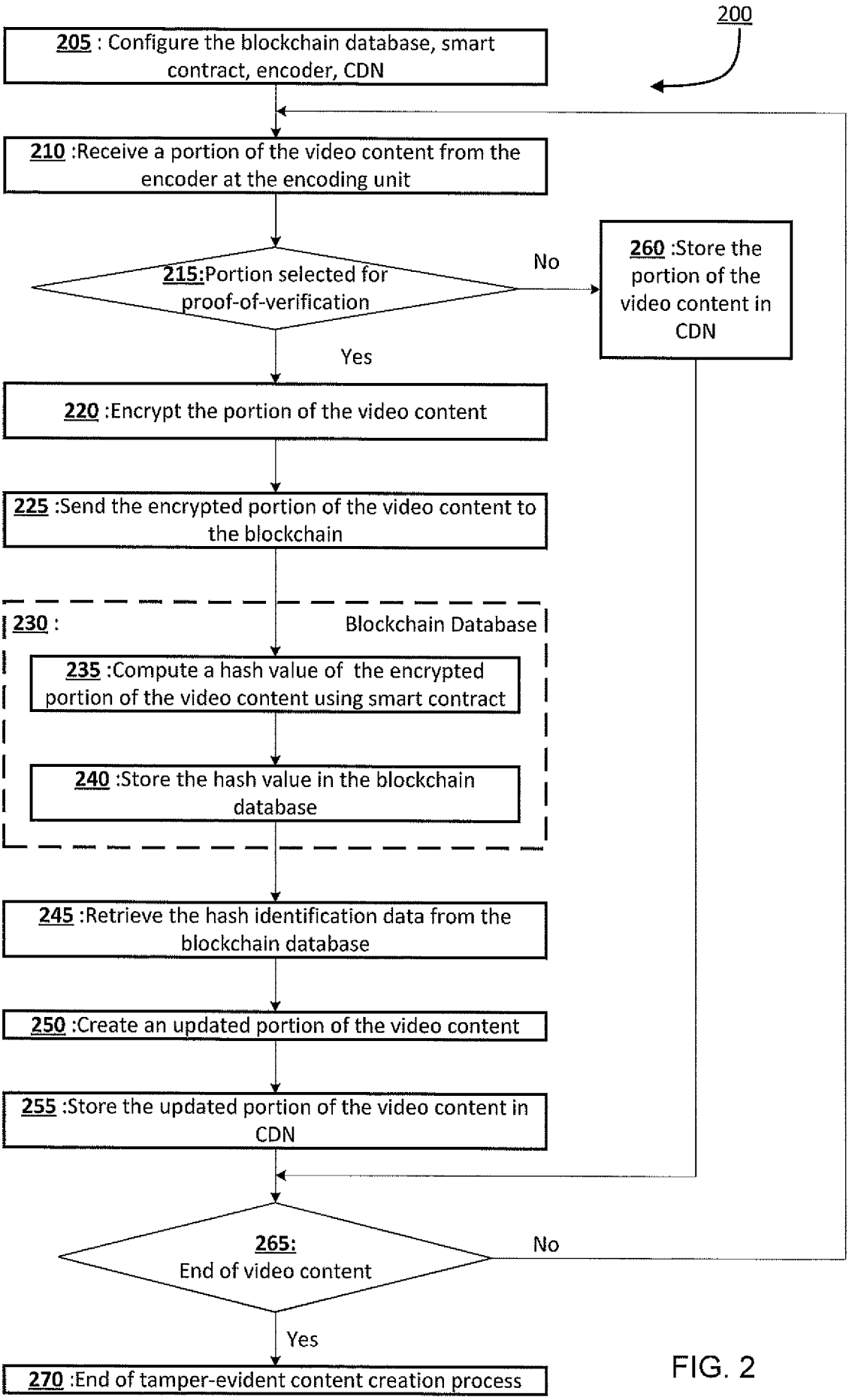
200
205 : Configure the blockchain database, smart contract, encoder, CDN
210 :Receive a portion of the video content from the encoder at the encoding unit
215:Portion selected for proof-of-verification
No
260 :Store the portion of the video content in CDN
Yes
220 :Encrypt the portion of the video content
225 :Send the encrypted portion of the video content to the blockchain
230 :
Blockchain Database
235 :Compute a hash value of the encrypted portion of the video content using smart contract
240 :Store the hash value in the blockchain database
245 :Retrieve the hash identification data from the blockchain database
250 :Create an updated portion of the video content
255 :Store the updated portion of the video content in CDN
265:
End of video content
No
Yes
270 :End of tamper-evident content creation process

FIG. 2

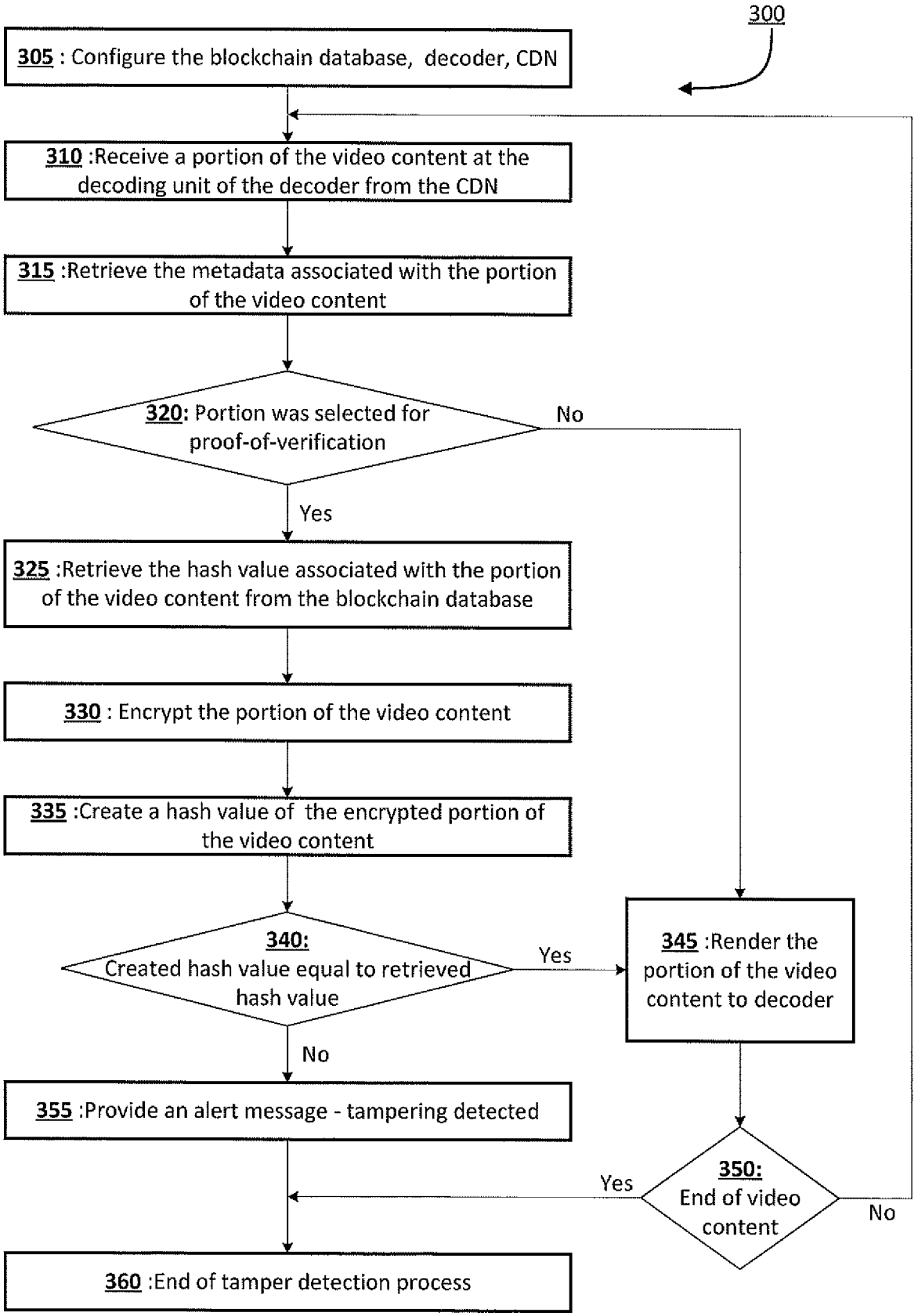
300
305 : Configure the blockchain database, decoder, CDN
310 :Receive a portion of the video content at the decoding unit of the decoder from the CDN
315 :Retrieve the metadata associated with the portion of the video content
320: Portion was selected for proof-of-verification
No
Yes
325 :Retrieve the hash value associated with the portion of the video content from the blockchain database
330 : Encrypt the portion of the video content
335 :Create a hash value of the encrypted portion of the video content
340: Created hash value equal to retrieved hash value
Yes
345 :Render the portion of the video content to decoder
No
355 :Provide an alert message - tampering detected
350: End of video content
Yes
No
360 :End of tamper detection process

FIG. 3

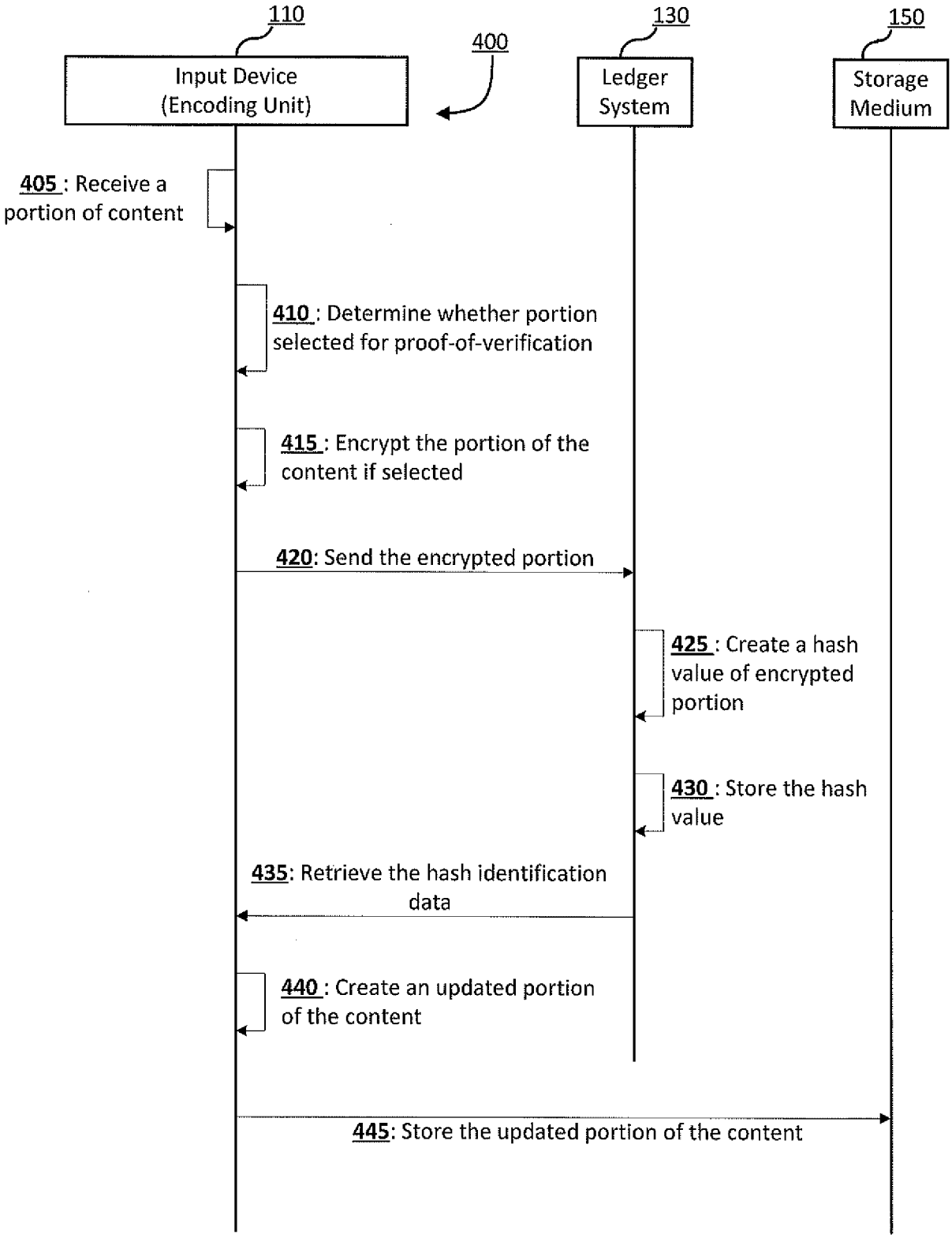
110
400
130
150
Input Device (Encoding Unit)
Ledger System
Storage Medium
405 : Receive a portion of content
410 : Determine whether portion selected for proof-of-verification
415 : Encrypt the portion of the content if selected
420: Send the encrypted portion
425 : Create a hash value of encrypted portion
430 : Store the hash value
435: Retrieve the hash identification data
440 : Create an updated portion of the content
445: Store the updated portion of the content

FIG. 4

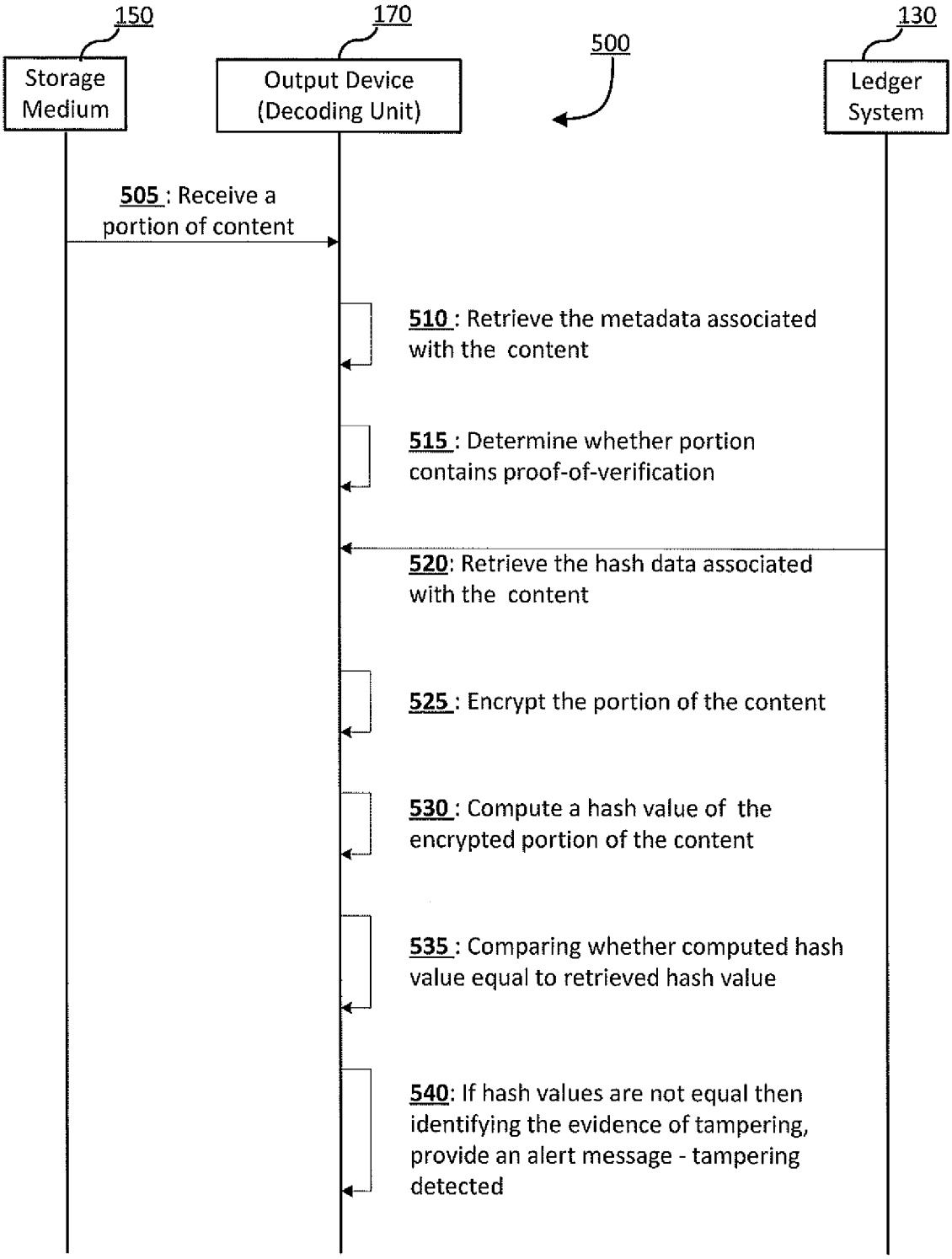
150
Storage Medium
170
Output Device (Decoding Unit)
500
130
Ledger System
505 : Receive a portion of content
510 : Retrieve the metadata associated with the content
515 : Determine whether portion contains proof-of-verification
520: Retrieve the hash data associated with the content
525 : Encrypt the portion of the content
530 : Compute a hash value of the encrypted portion of the content
535 : Comparing whether computed hash value equal to retrieved hash value
540: If hash values are not equal then identifying the evidence of tampering, provide an alert message - tampering detected

FIG. 5

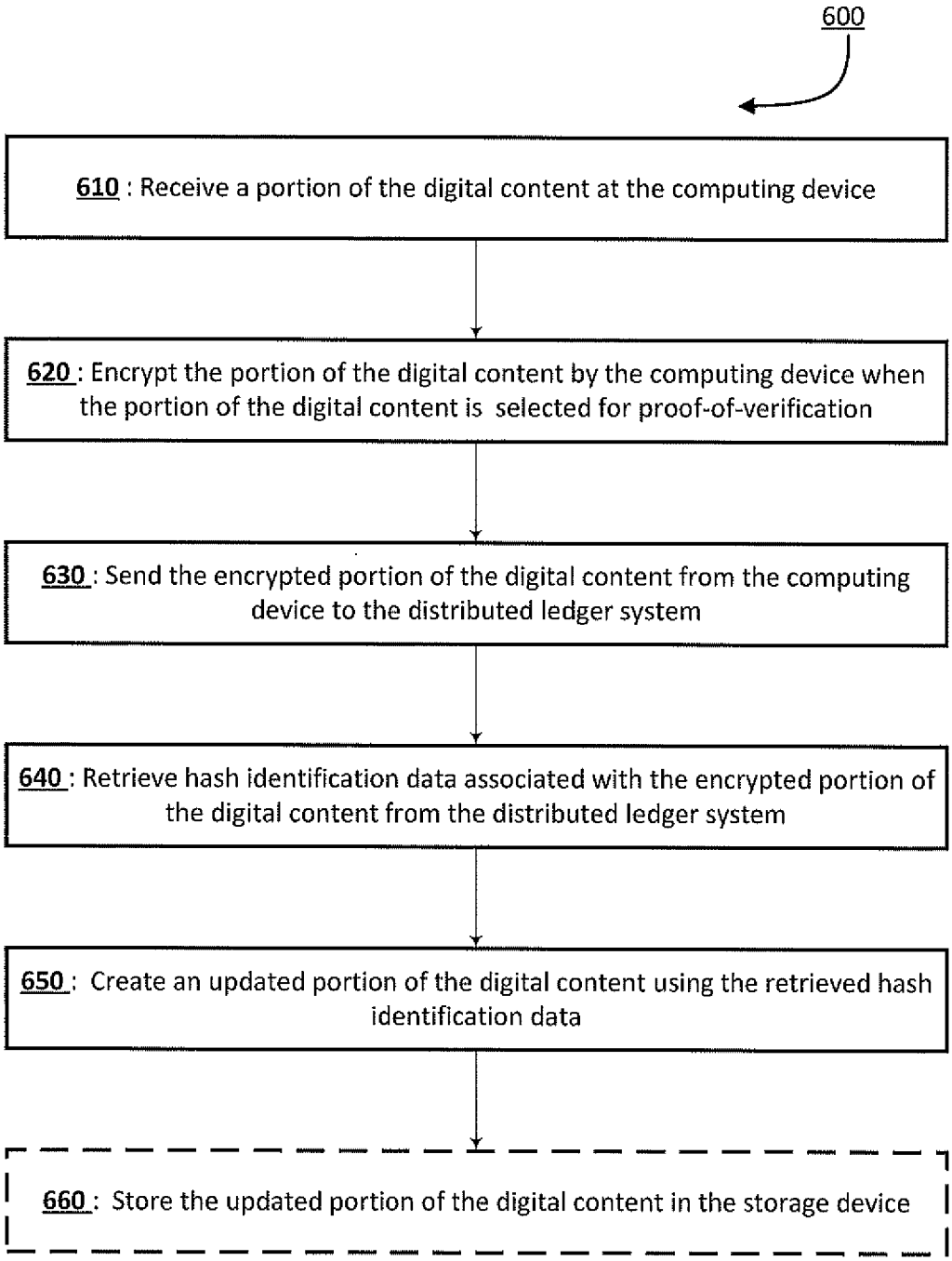
600
610 : Receive a portion of the digital content at the computing device
620 : Encrypt the portion of the digital content by the computing device when the portion of the digital content is selected for proof-of-verification
630 : Send the encrypted portion of the digital content from the computing device to the distributed ledger system
640 : Retrieve hash identification data associated with the encrypted portion of the digital content from the distributed ledger system
650 : Create an updated portion of the digital content using the retrieved hash identification data
660 : Store the updated portion of the digital content in the storage device

FIG. 6

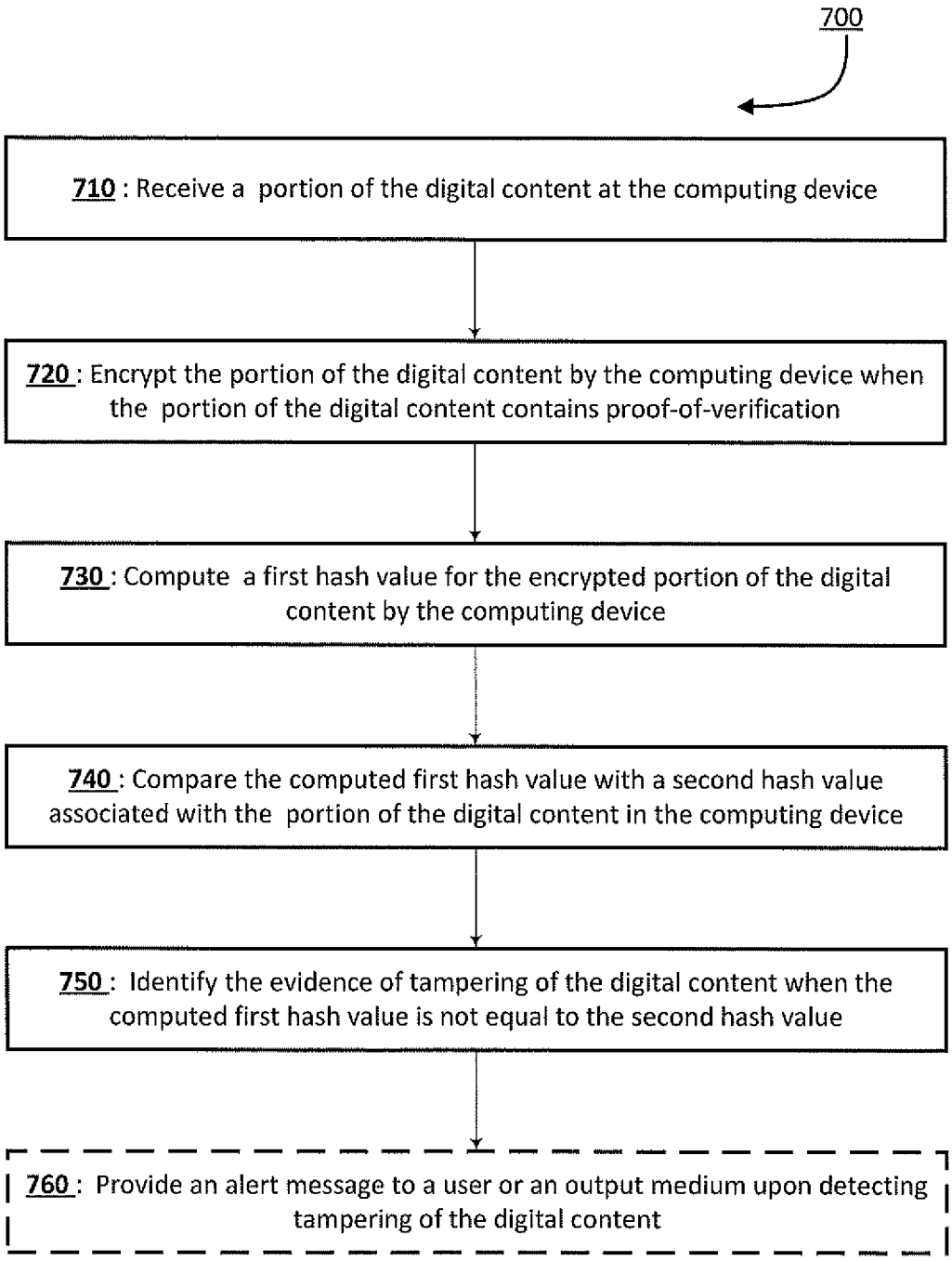
700
710 : Receive a portion of the digital content at the computing device
720 : Encrypt the portion of the digital content by the computing device when the portion of the digital content contains proof-of-verification
730 : Compute a first hash value for the encrypted portion of the digital content by the computing device
740 : Compare the computed first hash value with a second hash value associated with the portion of the digital content in the computing device
750 : Identify the evidence of tampering of the digital content when the computed first hash value is not equal to the second hash value
760 : Provide an alert message to a user or an output medium upon detecting tampering of the digital content

FIG. 7

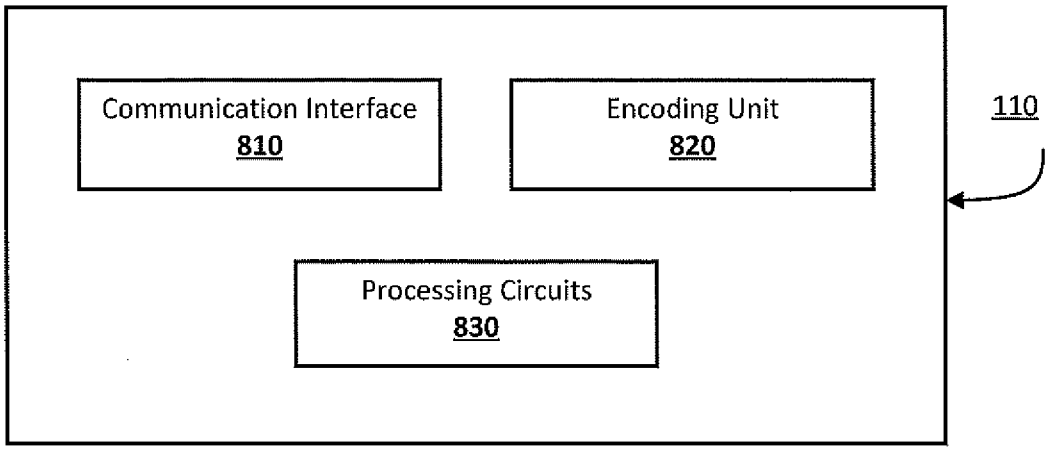
Communication Interface
810
Encoding Unit
820
Processing Circuits
830
110

FIG. 8

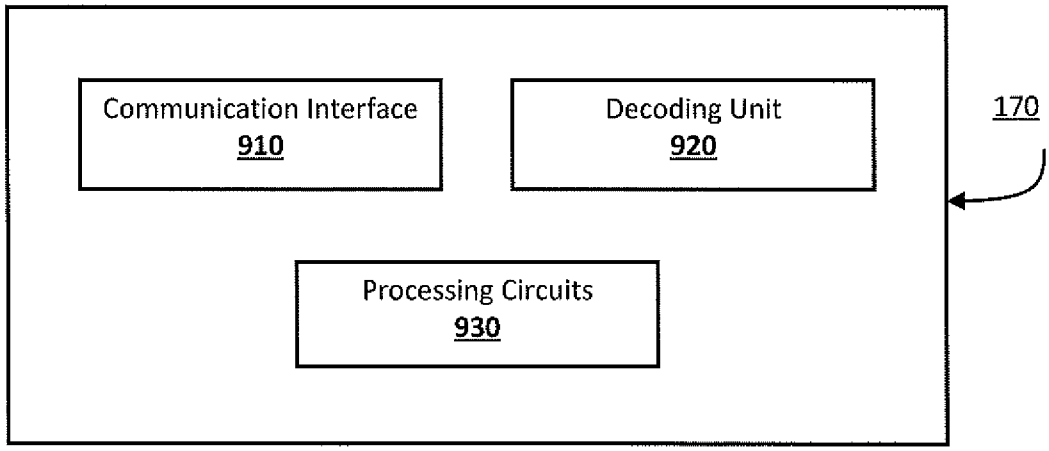
Communication Interface
910
Decoding Unit
920
Processing Circuits
930
170

FIG. 9

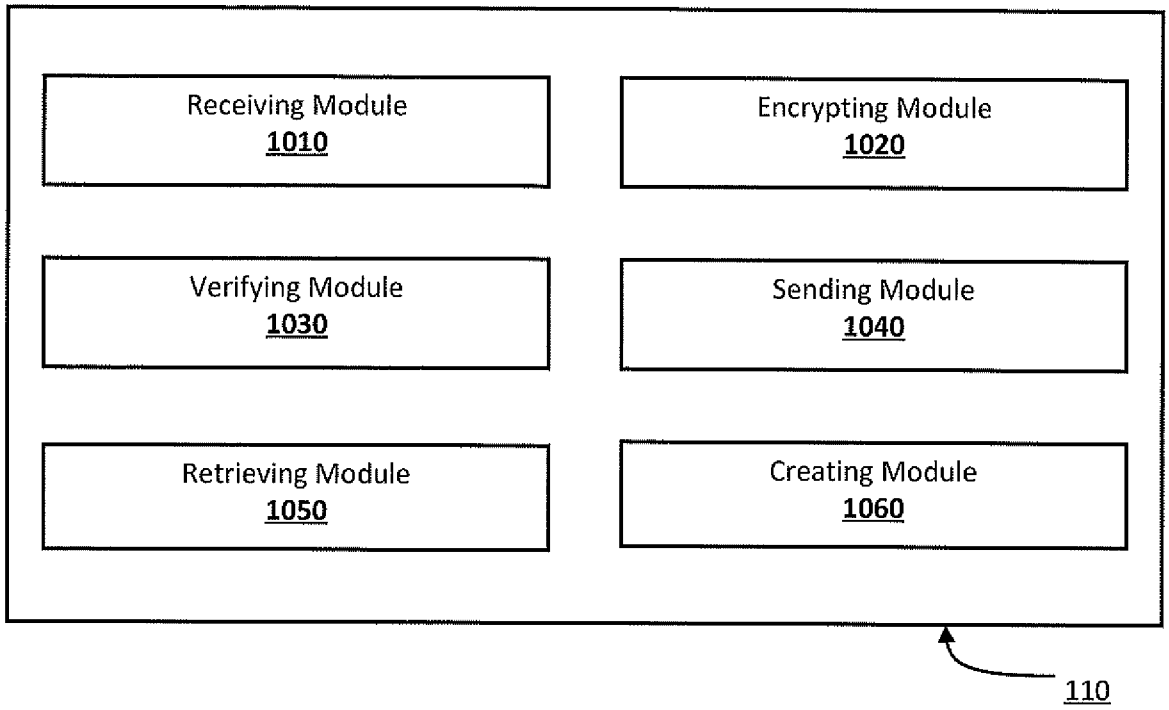
Receiving Module
1010
Encrypting Module
1020
Verifying Module
1030
Sending Module
1040
Retrieving Module
1050
Creating Module
1060
110

FIG. 10

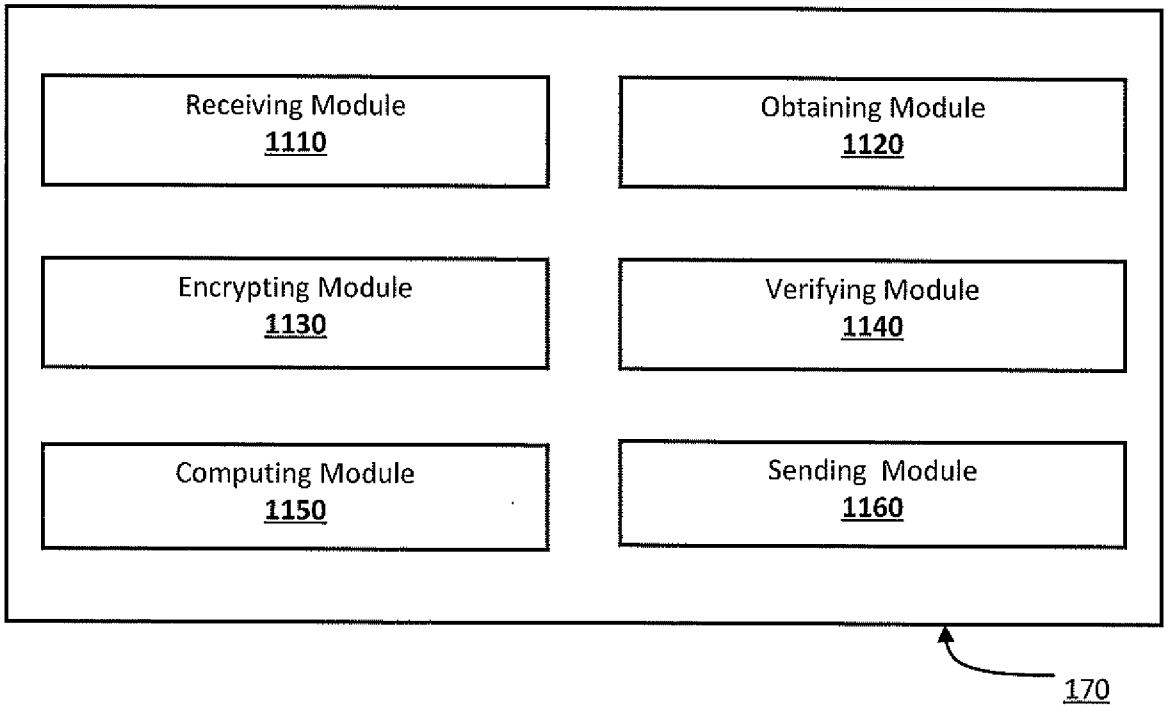
Receiving Module
1110
Obtaining Module
1120
Encrypting Module
1130
Verifying Module
1140
Computing Module
1150
Sending Module
1160
170

FIG. 11

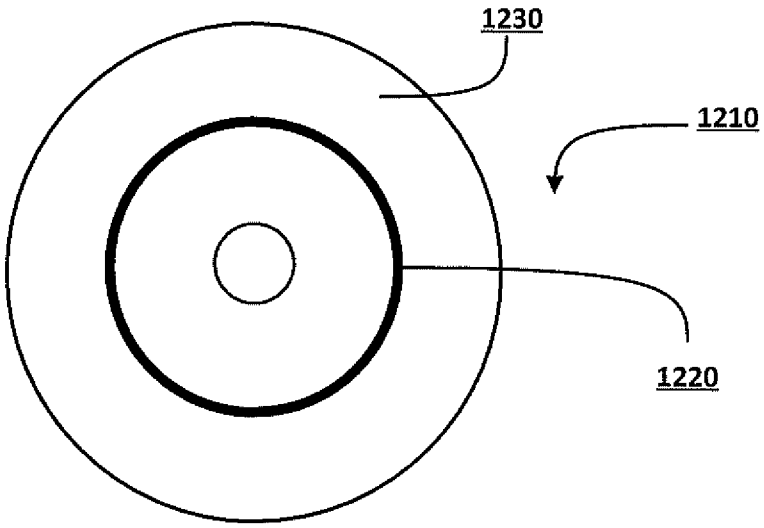


FIG. 12

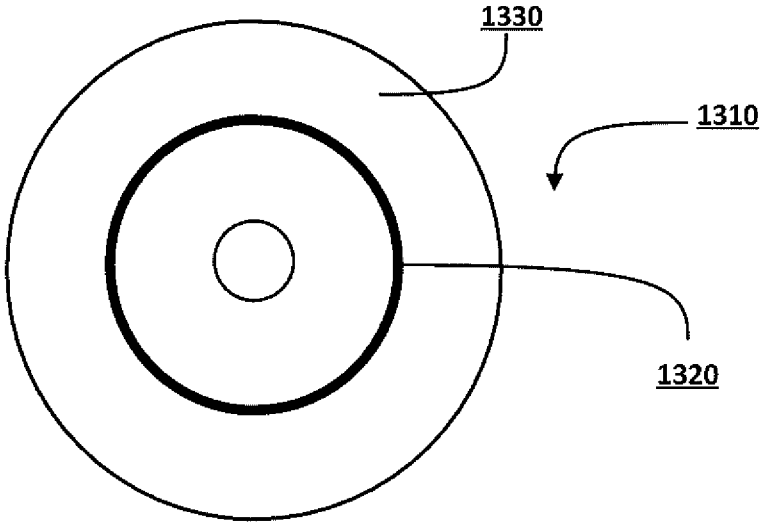


FIG. 13

**INTERNATIONAL SEARCH REPORT**

International application No.
PCT/SE201 7/051 145

A. CLASSIFICATION OF SUBJECT MATTER

IPC: see extra sheet

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC: G06F, H04N, H04W

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE, DK, FI, NO classes as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, PAJ, WPI data, COMPENDEX, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| Y | KR 201 601 27878 A (GALAXIA COMMUNICATIONS CO LTD), 7 November 2016 (2016-1 1-07); abstract; paragraphs [0001 ]-[001 3], [001 8]-[001 9], [0033]-[0034], [0039] -- | 1-38 |
| Y | CN 107104977 A (BEIJING TIANDE TECH CO LTD), 29 August 201 7 (201 7-08-29); abstract; paragraphs [0027]-[0031]; figure 1 -- | 1-38 |
| A | EP 2988238 A 1 (APPLE INC), 24 February 201 6 (201 6-02-24); abstract; paragraphs [001 1]-[0035] -- | 1-38 |
| A | US 201 50288704 A 1 (HUANG VINCENT ET AL), 8 October 2015 (201 5-1 0-08); abstract; paragraphs [001 2]-[0031] -- | 1-38 |

☒ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

| Date of the actual completion of the international search | Date of mailing of the international search report |
|---|---|
| 27-09-2018 | 27-09-2018 |
| Name and mailing address of the ISA/SE<br>Patent- och registreringsverket<br>Box 5055<br>S-102 42 STOCKHOLM<br>Facsimile No. + 46 8 666 02 86 | Authorized officer<br>Vincent Huang<br>Telephone No. + 46 8 782 28 00 |

**Continuation of:** second sheet

**International Patent Classification (IPC)**

***G06F 21/64*** *{*201 3.01 )
***G06F 3/06*** (2006.01 )
***H04N 21/854*** (201 1.01 )
***H04W 12/10*** (2009.01 )

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| KR | 201601 27878 | A | 07/1 1/201 6 | KR | 10171 2726 | B 1 | 14/03/201 7 |
| CN | 1071 04977 | A | 29/08/201 7 | NONE | | | |
| EP | 2988238 | A 1 | 24/02/201 6 | CN | 101443774 | B | 25/03/201 5 |
| | | | | DE | 10200701 2490 | A 1 | 20/09/2007 |
| | | | | EP | 1835432 | A 1 | 19/09/2007 |
| | | | | EP | 2284755 | B 1 | 22/04/201 5 |
| | | | | EP | 2284754 | A 1 | 16/02/201 1 |
| | | | | FR | 291 141 9 | A 1 | 18/07/2008 |
| | | | | SE | 534208 | C2 | 31/05/201 1 |
| | | | | SE | 0700601 | L | 16/09/2007 |
| | | | | SE | 535797 | C2 | 27/12/201 2 |
| | | | | SE | 1150067 | A 1 | 01/02/201 1 |
| | | | | US | 8364965 | B2 | 29/01/201 3 |
| | | | | US | 20070220261 | A 1 | 20/09/2007 |
| | | | | US | 8886947 | B2 | 11/1 1/2014 |
| | | | | US | 201 301451 67 | A 1 | 06/06/201 3 |
| | | | | WO | 20071 06831 | A 1 | 20/09/2007 |
| US | 20150288704 | A 1 | 08/1 0/201 5 | EP | 2918057 | A4 | 18/1 1/201 5 |
| | | | | US | 9432384 | B2 | 30/08/201 6 |
| | | | | WO | 2014074041 | A 1 | 15/05/2014 |