IDENTIFYING COPYRIGHTED MATERIAL USING EMBEDDED COPYRIGHT INFORMATION

TECHNICAL FIELD

[0001] This specification generally relates to copyright protection based on blockchain technology.

BACKGROUND

[0002] Distributed ledger systems (DLSs), which can also be referred to as consensus networks, and/or blockchain networks, enable participating entities to securely, and immutably store data. DLSs are commonly referred to as blockchain networks without referencing any particular user case. DLS technology creates a transparent chain-of-records by allowing all parties to a transaction to verify what will be entered onto a ledger, and prevent any party from changing the ledger after the transaction is entered. Each transaction or block of transactions is transmitted to all the participants in the blockchain network and is verified by each participant node before adding to the blockchain.

[0003] With the advancement of digital technologies, online digital content can often be reproduced, edited, recorded, and distributed by Internet users. In some cases, such uses can violate copyright of the creator or owner of the content. New technologies can also make management and protection of copyrighted content difficult. For example, such technologies can enable copies of copyrighted content to be rapidly produced and spread to large numbers of users, which can significantly affect the economic interests of the copyright holder.

[0004] DLS technology creates a secure, time-stamped, and immutable chain-of-records, making it well-suited to protect against copyright infringement. For example, a distributed ledger containing the copyright holder's information and update history relating to the copyrighted content can be used to determine ownership or infringement of the content.

[0005] However, when only a portion of copyrighted content is used by unauthorized users, it can be difficult to identify that the used portion is part of a copyrighted work. Therefore, it would be desirable to embed copyright information in the background, foreground, or the content itself, such that copyrighted material can be more easily identified.

SUMMARY

[0006] This specification describes technologies for embedding copyright information in textual content. These technologies generally involve generating a unique identifier (ID) based on copyright information associated with text; adding one or more function words to the text without altering a meaning of the text; and embedding the unique ID in the function words to provide an information-embedded text configured to enable retrieval of the copyright information from the blockchain based on the unique ID.

[0007] This specification also provides one or more non-transitory computer-readable storage media coupled to one or more processors and having instructions stored thereon which, when executed by the one or more processors, cause the one or more processors to perform operations in accordance with embodiments of the methods provided herein.

[0008] This specification further provides a system for implementing the methods provided herein. The system includes one or more processors, and a computer-readable storage medium coupled to the one or more processors having instructions stored thereon which, when executed by the one or more processors, cause the one or more processors to perform operations in accordance with embodiments of the methods provided herein.

[0009] It is appreciated that methods in accordance with this specification may include any combination of the aspects and features described herein. That is, methods in accordance with this specification are not limited to the combinations of aspects and features specifically described herein, but also include any combination of the aspects and features provided.

[0010] The details of one or more embodiments of this specification are set forth in the accompanying drawings and the description below. Other features and advantages of this specification will be apparent from the description and drawings, and from the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] FIG. 1 is a diagram illustrating an example of an environment that can be used to execute embodiments of this specification.

[0012] FIG. 2 is a diagram illustrating an example of an architecture in accordance with embodiments of this specification.

[0013] FIG. 3 is a flowchart illustrating an example of a process that can be executed in accordance with embodiments of this specification.

[0014] FIG. 4 depicts an example of a content processing process in accordance with embodiments of this specification.

[0015] FIG. 5 depicts an example of a blockchain transaction record in accordance with embodiments of this specification.

[0016] FIG. 6 depicts an example of a copyright infringement determination process in accordance with embodiments of this specification.

[0017] FIG. 7 depicts an example of a process that can be executed in accordance with embodiments of this specification.

[0018] FIG. 8 depicts examples of modules of an apparatus in accordance with embodiments of this specification.

[0019] Like reference numbers and designations in the various drawings indicate like elements.

DETAILED DESCRIPTION

[0020] This specification describes technologies for embedding copyright information in textual content. These technologies generally involve generating a unique identifier (ID) based on copyright information associated with text; adding one or more function words to the text without altering a meaning of the text; and embedding the unique ID in the function words to provide an information-embedded text configured to enable retrieval of the copyright information from the blockchain based on the unique ID.

[0021] To provide further context for embodiments of this specification, and as introduced above, distributed ledger systems (DLSs), which can also be referred to as consensus networks (e.g., made up of peer-to-peer nodes), and blockchain networks, enable participating entities to securely, and immutably conduct transactions, and store data. Although the term blockchain is generally associated with particular networks, and/or use cases, blockchain is used herein to generally refer to a DLS without reference to any particular use case.

[0022] A blockchain is a data structure that stores transactions in a way that the transactions are immutable. Thus, transactions recorded on a blockchain are reliable and trustworthy. A blockchain includes one or more blocks. Each block in the chain is linked to a previous block immediately before it in the chain by including a cryptographic hash of the previous block. Each block also includes a timestamp, its own cryptographic hash, and one or

more transactions. The transactions, which have already been verified by the nodes of the blockchain network, are hashed and encoded into a Merkle tree. A Merkle tree is a data structure in which data at the leaf nodes of the tree is hashed, and all hashes in each branch of the tree are concatenated at the root of the branch. This process continues up the tree to the root of the entire tree, which stores a hash that is representative of all data in the tree. A hash purporting to be of a transaction stored in the tree can be quickly verified by determining whether it is consistent with the structure of the tree.

[0023] Whereas a blockchain is a decentralized or at least partially decentralized data structure for storing transactions, a blockchain network is a network of computing nodes that manage, update, and maintain one or more blockchains by broadcasting, verifying and validating transactions, etc. As introduced above, a blockchain network can be provided as a public blockchain network, a private blockchain network, or a consortium blockchain network. Embodiments of this specification are described in further detail herein with reference to a consortium blockchain network. It is contemplated, however, that embodiments of this specification can be realized in any appropriate type of blockchain network.

[0024] In general, a consortium blockchain network is private among the participating entities. In a consortium blockchain network, the consensus process is controlled by an authorized set of nodes, which can be referred to as consensus nodes, one or more consensus nodes being operated by a respective entity (e.g., a financial institution, insurance company). For example, a consortium of ten (10) entities (e.g., financial institutions, insurance companies) can operate a consortium blockchain network, each of which operates at least one node in the consortium blockchain network.

[0025] In some examples, within a consortium blockchain network, a global blockchain is provided as a blockchain that is replicated across all nodes. That is, all consensus nodes are in perfect state consensus with respect to the global blockchain. To achieve consensus (e.g., agreement to the addition of a block to a blockchain), a consensus protocol is implemented within the consortium blockchain network. For example, the consortium blockchain network can implement a practical Byzantine fault tolerance (PBFT) consensus, described in further detail below.

[0026] FIG. 1 is a diagram illustrating an example of an environment 100 that can be used to execute embodiments of this specification. In some examples, the environment 100 enables entities to participate in a consortium blockchain network 102. The environment 100 includes computing devices 106, 108, and a network 110. In some examples, the network 110 includes a local area network (LAN), wide area network (WAN), the Internet, or a combination thereof, and connects web sites, user devices (e.g., computing devices), and back-end systems. In some examples, the network 110 can be accessed over a wired and/or a wireless communications link. In some examples, the network 110 enables communication with, and within the consortium blockchain network 102. In general, the network 110 represents one or more communication networks. In some cases, the computing devices 106, 108 can be nodes of a cloud computing system (not shown), or each computing device 106, 108 can be a separate cloud computing system including a number of computers interconnected by a network and functioning as a distributed processing system.

[0027] In the depicted example, the computing systems 106, 108 can each include any appropriate computing system that enables participation as a node in the consortium blockchain network 102. Examples of computing devices include, without limitation, a server, a desktop computer, a laptop computer, a tablet computing device, and a smartphone. In some examples, the computing systems 106, 108 hosts one or more computer-implemented services for interacting with the consortium blockchain network 102. For example, the computing system 106 can host computer-implemented services of a first entity (e.g., user A), such as a transaction management system that the first entity uses to manage its transactions with one or more other entities (e.g., other users). The computing system 108 can host computer-implemented services of a second entity (e.g., user B), such as a transaction management system that the second entity uses to manage its transactions with one or more other entities (e.g., other users). In the example of FIG. 1, the consortium blockchain network 102 is represented as a peer-to-peer network of nodes, and the computing systems 106, 108 provide nodes of the first entity, and second entity respectively, which participate in the consortium blockchain network 102.

[0028] FIG. 2 depicts an example of an architecture 200 in accordance with embodiments of this specification. The architecture 200 includes an entity layer 202, a hosted services layer 204, and a blockchain network layer 206. In the depicted example, the entity layer 202

includes three participants, Participant A, Participant B, and Participant C, each participant having a respective transaction management system 208.

[0029] In the depicted example, the hosted services layer 204 includes interfaces 210 for each transaction management system 210. In some examples, a respective transaction management system 208 communicates with a respective interface 210 over a network (e.g., the network 110 of FIG. 1) using a protocol (e.g., hypertext transfer protocol secure (HTTPS)). In some examples, each interface 210 provides communication connection between a respective transaction management system 208, and the blockchain network layer 206. More particularly, the interface 210 communicate with a blockchain network 212 of the blockchain network layer 206. In some examples, communication between an interface 210, and the blockchain network layer 206 is conducted using remote procedure calls (RPCs). In some examples, the interfaces 210 "host" blockchain network nodes for the respective transaction management systems 208. For example, the interfaces 210 provide the application programming interface (API) for access to blockchain network 212.

[0030] As described herein, the blockchain network 212 is provided as a peer-to-peer network including a plurality of nodes 214 that immutably record information in a blockchain 216. Although a single blockchain 216 is schematically depicted, multiple copies of the blockchain 216 are provided, and are maintained across the blockchain network 212. For example, each node 214 stores a copy of the blockchain. In some embodiments, the blockchain 216 stores information associated with transactions that are performed between two or more entities participating in the consortium blockchain network.

[0031] A blockchain (e.g., the blockchain 216 of FIG. 2) is made up of a chain of blocks, each block storing data. Examples of data include transaction data representative of a transaction between two or more participants. While transactions are used herein by way of non-limiting example, it is contemplated that any appropriate data can be stored in a blockchain (e.g., documents, images, videos, audio). Examples of a transaction can include, without limitation, exchanges of something of value (e.g., assets, products, services, currency). The transaction data is immutably stored within the blockchain. That is, the transaction data cannot be changed.

[0032] Before storing in a block, the transaction data is hashed. Hashing is a process of transforming the transaction data (provided as string data) into a fixed-length hash value

(also provided as string data). It is not possible to un-hash the hash value to obtain the transaction data. Hashing ensures that even a slight change in the transaction data results in a completely different hash value. Further, and as noted above, the hash value is of fixed length. That is, no matter the size of the transaction data the length of the hash value is fixed. Hashing includes processing the transaction data through a hash function to generate the hash value. An example of a hash function includes, without limitation, the secure hash algorithm (SHA)-256, which outputs 256-bit hash values.

[0033] Transaction data of multiple transactions are hashed and stored in a block. For example, hash values of two transactions are provided, and are themselves hashed to provide another hash. This process is repeated until, for all transactions to be stored in a block, a single hash value is provided. This hash value is referred to as a Merkle root hash, and is stored in a header of the block. A change in any of the transactions will result in change in its hash value, and ultimately, a change in the Merkle root hash.

[0034] Blocks are added to the blockchain through a consensus protocol. Multiple nodes within the blockchain network participate in the consensus protocol, and perform work to have a block added to the blockchain. Such nodes are referred to as consensus nodes. PBFT, introduced above, is used as a non-limiting example of a consensus protocol. The consensus nodes execute the consensus protocol to add transactions to the blockchain, and update the overall state of the blockchain network.

[0035] In further detail, the consensus node generates a block header, hashes all of the transactions in the block, and combines the hash value in pairs to generate further hash values until a single hash value is provided for all transactions in the block (the Merkle root hash). This hash is added to the block header. The consensus node also determines the hash value of the most recent block in the blockchain (i.e., the last block added to the blockchain). The consensus node also adds a nonce value, and a timestamp to the block header.

[0036] In general, PBFT provides a practical Byzantine state machine replication that tolerates Byzantine faults (e.g., malfunctioning nodes, malicious nodes). This is achieved in PBFT by assuming that faults will occur (e.g., assuming the existence of independent node failures, and/or manipulated messages sent by consensus nodes). In PBFT, the consensus nodes are provided in a sequence that includes a primary consensus node, and backup consensus nodes. The primary consensus node is periodically changed, Transactions are

added to the blockchain by all consensus nodes within the blockchain network reaching an agreement as to the world state of the blockchain network. In this process, messages are transmitted between consensus nodes, and each consensus nodes proves that a message is received from a specified peer node, and verifies that the message was not modified during transmission.

[0037] In PBFT, the consensus protocol is provided in multiple phases with all consensus nodes beginning in the same state. To begin, a client sends a request to the primary consensus node to invoke a service operation (e.g., execute a transaction within the blockchain network). In response to receiving the request, the primary consensus node multicasts the request to the backup consensus nodes. The backup consensus nodes execute the request, and each sends a reply to the client. The client waits until a threshold number of replies are received. In some examples, the client waits for f+1 replies to be received, where f is the maximum number of faulty consensus nodes that can be tolerated within the blockchain network. The final result is that a sufficient number of consensus nodes come to an agreement on the order of the record that is to be added to the blockchain, and the record is either accepted, or rejected.

[0038] In some blockchain networks, cryptography is implemented to maintain privacy of transactions. For example, if two nodes want to keep a transaction private, such that other nodes in the blockchain network cannot discern details of the transaction, the nodes can encrypt the transaction data. An example of cryptography includes, without limitation, symmetric encryption, and asymmetric encryption. Symmetric encryption refers to an encryption process that uses a single key for both encryption (generating ciphertext from plaintext), and decryption (generating plaintext from ciphertext). In symmetric encryption, the same key is available to multiple nodes, so each node can en-/de-crypt transaction data.

[0039] Asymmetric encryption uses key pairs that each include a private key and a public key, the private key being known only to a respective node, and the public key being known to any or all other nodes in the blockchain network. A node can use the public key of another node to encrypt data, and the encrypted data can be decrypted using other node's private key. For example, and referring again to FIG. 2, Participant A can use Participant B's public key to encrypt data, and send the encrypted data to Participant B. Participant B can use its private key to decrypt the encrypted data (ciphertext) and extract the original data (plaintext).

Messages encrypted with a node's public key can only be decrypted using the node's private key.

[0040] Asymmetric encryption is used to provide digital signatures, which enables participants in a transaction to confirm other participants in the transaction, as well as the validity of the transaction. For example, a node can digitally sign a message, and another node can confirm that the message was sent by the node based on the digital signature of Participant A. Digital signatures can also be used to ensure that messages are not tampered with in transit. For example, and again referencing FIG. 2, Participant A is to send a message to Participant B. Participant A generates a hash of the message, and then, using its private key, encrypts the hash to provide a digital signature as the encrypted hash. Participant A appends the digital signature to the message, and sends the message with digital signature to Participant B. Participant B decrypts the digital signature using the public key of Participant A, and extracts the hash. Participant B hashes the message and compares the hashes. If the hashes are same, Participant B can confirm that the message was indeed from Participant A, and was not tampered with.

[0041] As described above, blockchain can be used to generate an immutable and transparent chain-of-record that can assist in identification and authentication of creators or right holders of original copyrighted works. For example, a transaction related to copyrighted content (e.g., a content update or newly entered identity information associated with creator of the content) recorded on the blockchain can be encoded as a unique ID. The unique ID can be invisibly embedded in the copyrighted content such that it does not affect the visual presentation of the content and is undiscernible or not readily apparent to an inspecting user. In some embodiments, the unique ID can be embedded in other data carriers such as the foreground or background of the copyrighted content.

[0042] To determine the copyright holder and identify potential copyright infringement, the embedded unique ID can be extracted from the content to decode the transaction. The transaction can then be used to retrieve a chain-of-record associated with the copyrighted content. In some embodiments, the chain-of-record can be used to identify a copyright holder, and seek permission from the copyright holder to copy, modify, distribute, or publicly display copyrighted works. In some embodiments, the chain-of-record can also be used to

resolve copyright dispute or determine whether the copyrighted content has been tampered with.

[0043] FIG. 3 is a flowchart illustrating an example of a process 300 that can be executed in accordance with embodiments of this specification. The process 300 can be performed by one or more blockchain nodes or a computing device communicably coupled to the one or more blockchain nodes. For clarity of presentation, the description that follows generally describes process 300 in the context of the other figures in this description. However, it will be understood that process 300 can be performed, for example, by any system, environment, software, and hardware, or a combination of systems, environments, software, and hardware, as appropriate. In some embodiments, various steps of process 300 can be run in parallel, in combination, in loops, or in any order.

[0044] At 302, a digital content is identified. The digital content can be content of a copyrighted work (i.e., copyrighted digital content, or simply, copyrighted content) associated with a copyright holder. Copyrighted works can be any original or creative works, such as literary works, motion pictures, choreography, musical compositions, sound recordings, paintings, drawings, sculptures, photographs, computer software, radio and television broadcasts, or industrial designs. Copyrighted content can exist in the form of digital data, which can be digitally transmitted, streamed, contained in computer files, or stored as a transaction record in a blockchain.

[0045] At 304, blockchain-based copyright authentication is performed. In some embodiments, copyright authentication can include authenticating a right holder of copyrighted content. As discussed earlier, blockchain provides an immutable chain-of-record of transactions related to the copyrighted content. For example, transactions related to the copyrighted content can include the original digital content, timestamps or locations of content creation and distributions, content update or modification history, creatorship, ownership, licensing information, or other information. After a transaction is approved, a corresponding blockchain transaction ID can be recorded to the corresponding block and distributed to the distributed ledger of the blockchain network. In some embodiments, the transactions can also be embedded with smart contract information associated with an Ethereum blockchain.

[0046] The right holder of copyrighted content can be authenticated based on traversing transaction records of the copyrighted content on the blockchain. For example, a blockchain transaction ID associated with a record of the copyrighted content on the blockchain can be determined. A chain-of-record associated with the blockchain transaction ID can then be traversed until a record on ownership or creatorship associated with the copyrighted content is identified. The identified owner or creator of the content recorded on the blockchain can be determined as the authentic right holder. In some embodiments, the ownership or creatorship information may not be recorded on the blockchain. In such cases, the right holder may be authenticated if other information (e.g., detailed update and modification history) can be provided to match records on the blockchain.

[0047] At 306, a unique ID associated with the copyright holder can be generated. The unique ID can be used to uniquely identify the copyright holder. In some embodiments, the unique ID can be a public key certificate issued by a certificate authority (CA). For example, the CA can be one or more trusted entities of a consortium blockchain. The certificate can include the copyrighted content, information about the identity of the copyrighted holder, and the digital signature of the CA. In some embodiments, the unique ID can include the copyright holder's identity information digitally signed by its private key. The copyright holder can then be identified by decrypting the digital signature with the public key corresponding to the copyright holder and verified by comparing the decrypted information to the copyright holder's information in the blockchain.

[0048] The copyright holder can be an individual or a business entity. Example identity information of individual copyright holder can include one or more of the right holder's residence ID, passport number, driver's license number, phone number, or name. Example identity information of business copyright holder can include one or more of business registration number, tax ID, registered name, phone number, address, or trademark.

[0049] In some embodiments, the unique ID can be generated based on transactions associated with the copyrighted content recorded on the blockchain. As discussed earlier, those transactions can include the copyrighted content, timestamps or locations of content creation and distributions, content update or modification history, creatorship, ownership, licensing information, etc. The unique ID can be generated based on one or more transactions. In some cases, at least a portion of the copyrighted content can be used to generate the unique

ID, such as by generating a hash value based on the portion of the content. In some examples, the portion of the copyrighted content preserves the originality of the copyrighted, and can be used to identify the copyrighted content it is extracted from. To identify the copyright holder, the portion of the copyrighted content can be used to identify a blockchain transaction ID corresponding to the recordation of the copyrighted content on the blockchain. The blockchain transaction ID can then be used to retrieve an entire chain-of-record associated with the copyrighted content to identify the copyright holder. In some embodiments, the unique ID can also be generated based on the one or more transactions and one or more of the corresponding timestamps and locations.

[0050] In some embodiments, the unique ID can be invisibly embedded in data carriers as a digital watermark, such that it can be extracted to obtain associated copyright information. In some embodiments, the unique ID can be encrypted or hashed to enhance the robustness and security of the digital watermark. For example, the encryption can be performed based on public key encryption using the copyright holder's public key. In some embodiments, the encryption can also be performed based on encryption schemes such as the Arnold scrambling encryption to adapt to the embedding technique used. By performing encryption, even if an attacker extracts the digital watermark, the associated identity information cannot be decrypted if the encryption scheme or encryption key is unknown to the attacker.

[0051] In some embodiments, timestamps and addresses associated with the transactions of the copyrighted content can be added to generate the unique ID. The addition of timestamp and address information can further ensure uniqueness of the ID. The timestamp and address information can also be used to identify and authenticate the right holder for copyright infringement detection and enforcement. In some embodiments, the unique ID can also be a digital signature generated based on the right holder's private key.

[0052] At 308, the unique ID is invisibly embedded. In some examples, the embedded unique ID can be considered invisible if its addition is not noticeable or visible to naked eyes. In other words, the unique ID can be considered invisible if the visual difference between the copyrighted contents before embedding the unique ID and after embedding the unique ID is not apparent to an unaided human eye. The unique ID can be embedded in copyrighted content, foreground of the copyrighted content, or background of the copyrighted content.

The foreground of the copyrighted content can be a film or screen attached or integrated to a platform where the copyrighted content is displayed. The foreground of the copyrighted content can also be a layer including the unique ID that is configured not to be rendered by a viewing application. The background of the copyrighted content can be a physical object (e.g., paper) the copyrighted content is printed on, or a background of a visual representation of an electronic file containing the copyrighted content that is rendered on a display.

[0053] The invisible embedding of copyrighted content can be performed based on digital watermarking. Digital watermarking is a technique of using data carriers to embed copyright information. Data carriers can include text, images, physical or electronic media. The unique ID can be embedded in the data carriers as an invisible digital watermark. When a copyright dispute occurs, the unique ID can be extracted from the data carriers to verify the ownership of the copyright.

[0054] Digital watermarking techniques can include background brightness masking, illuminance masking, texture masking, spatial masking and frequency masking. Different techniques can be based on different digital watermarking algorithms. In some embodiments, digital watermark embedding can be based on human visual system (HVS) characteristics to achieve visual imperceptibility or invisibility. For example, the human eyes are more sensitive to the changes in the low intensity pixels than the higher intensity ones. In illuminance masking, pixels of digital copyrighted content can be scanned to identify those with high intensity levels. The high intensity pixels can be used to embed the digital watermark. Modifications made to the high intensity pixels can be difficult to perceive by human eyes, such that the digital watermark can be considered invisibly embedded. Using illuminance masking, the digital watermark can be embedded in the entire copyrighted content or a portion of the content. For example, if the copyrighted content is textual content, the digital watermark can be embedded in a letter, a word, a sentence, or all the way up to the entire textual content.

[0055] As another example, in texture masking, the more complex the texture is, the more difficult the human eyes can discern the change of the texture. Therefore, the digital watermark can be embedded to a portion of a copyrighted image or background of copyrighted content with more complex texture to be less perceptible by human eyes.

[0056] As yet another example, digital watermarking can be performed based on frequency masking by transforming the copyrighted content to frequency domain coefficients based on mathematical transformations such as discrete cosine transform (DCT), discrete wavelet transform (DWT), or Arnold transform. In frequency masking, intermediary or low frequency coefficients are normally modified according to the watermark information. Those frequency ranges are less impactful to the appearance of the original content. The digital watermark (i.e., the unique ID) can be in the form of a character string. The character string can be converted to a single binary string. Each bit can be embedded to adjust lower frequency coefficients depending on the transformation technique used. In frequency masking, the digital watermark can be dispersed to the whole copyrighted content, which can increase robustness and invisibility. Other example mathematical transformations for performing digital watermarking can include DWT and singular value decomposition (DWT-SVD), least significant bit (LSB), or highly undetectable steganography (HUGO).

[0057] In some embodiments, more than one digital watermarking technique can be used to embed multiple layers of digital watermarks to the data carrier. In such cases, an index of the layers of digital watermarking can also be generated and embedded in the data carrier. The index can be used to more easily extract digital watermarks from the data carriers when performing copyright authentication or infringement identification. In some embodiments, the unique ID can be randomized or mixed with a random signal before embedding to the data carrier.

[0058] In some examples, the copyrighted content can be textual content. The embedded unique ID can be considered invisible if it does not materially change the content. The content can be considered not materially changed if the embedded information does not alter the meaning of the text. For example, the original text is “he becomes a monster after seven years,” and the information embedded text is “he has become a ‘monster’ after seven years.” The information embedded in the added text is not considered as materially changed from the original text, since the meaning of the text stays the same. In this example, the unique ID can be embedded in the quotation marks, or the auxiliary word “has,” according to the techniques described herein. In some embodiments, the unique ID can be embedded in spaces or function words such as articles, prepositions, conjunctions, and auxiliaries, which do not normally alter the material meaning of textual content.

[0059] In some embodiments, a plurality of unique IDs can be used to track the updates of the textual content. For example, one or more unique IDs can be generated based on identity information of the copyright holder and timestamps corresponding to one or more content updates recorded on the blockchain. When a content update is performed and recorded on the blockchain, a unique ID including the corresponding timestamp can be embedded in at least a portion of the updated content. To authenticate the right holder, the timestamps in the unique IDs extracted from the copyrighted content can form a timeline of creation and update history of the content. The timeline can be used as evidence of the content development process, and compared to the update history recorded on the blockchain to authenticate authorship of the textual content.

[0060] In some embodiments, the unique ID can be embedded in characteristics information of textual or image content. Example characteristics information can include color, brightness, and transparency of text or image. For example, text or image color can be encoded based on a red, green, and blue (RGB) color model. The unique ID can be embedded to slightly change the RGB value combination not noticeable by naked eyes. As another example, the unique ID can be embedded in YCbCr formatted text or image, where Y represents the luminance signal, Cb represents chrominance blue signal, and Cr represents chrominance red signal. The luminance signal is relatively insensitive to information embedding, which can be preferably used to invisibly embed the unique ID.

[0061] In some embodiments, the unique ID can be embedded in punctuations of textual content. For example, in the text "after seven years, he becomes a 'monster'," the unique ID can be embedded in the encoding of the commas and quotation marks.

[0062] In some embodiments, the unique ID can be embedded in fonts used by textual content. Fonts can be encoded using Unicode such as UTF-8. The UTF-8 is based on variable width character encoding, which uses one to four 8-bit bytes. Embedding the unique ID in some of the bits may slightly change pixel positions, size, angle, shape of font or punctuations invisible to the naked eye. For example, pixels of text fonts and punctuations can be encoded with redundancy. Redundancy is normally added to pixels for error checking in data communications. Redundancy such as check bits can be used to check data integrity at the receiver end. If the check bits are not consistent with the rest of the information bearing pixels when they arrive at the receiver, the receiver can ask the sender to retransmit the

content. The unique ID can be randomly embedded in redundant pixels or embedded in redundant pixels in fixed positions. It is to be understood that the unique ID can also be invisibly embedded in other hidden characteristics of fonts or punctuations.

[0063] In some embodiments, the unique ID can be embedded in forms or tables within or containing the copyrighted content. Those embodiments are especially suitable for right authentication of text or images inserted in online forms or tables. For example, an online form can be a form defined in the Hypertext Markup Language (HTML) definition of a webpage (such as by using the "<form>" tag) and rendered by a browser program displaying the webpage. Similarly, an online table can be a table defined in the Hypertext Markup Language (HTML) definition of a webpage (such as by using the "<table>" tag) and rendered by a browser program displaying the webpage. The unique ID can be invisibly embedded in values associated with color, shade, cell size, or other properties of the forms or tables. The unique ID can be extracted for copyright infringement identification when the forms or tables containing the copyrighted content are copied by potential copyright infringers without permission.

[0064] In some embodiments, the unique ID can be embedded in background texture of physical or digital data carriers. In some cases, the unique ID can be embedded to physical media that copyrighted content can be printed or written on. For example, the unique ID can be converted into a unique mesh pattern, which can serve as a digital watermark. A document image in which copyright information is embedded by superimposing the mesh pattern as a document background can then be printed on physical papers. A scan or picture image of the watermarked printed papers can be filtered to extract the digital watermark in copyright authentication. As another example, the unique ID can be embedded as digital watermark into hue component of the physical paper, such that it is more tolerant to digital-to-analog conversion and can be more easily extracted.

[0065] In some cases, the unique ID can be embedded in electronic media that can be used to display or publish copyright content. For example, the unique ID can be embedded in background textures of webpages or electronic files such as MICROSOFT WORD, POWERPOINT, or ADOBE PDF. In some cases, the unique ID can be embedded in a film or screen attached or integrated to an electronic display. In those cases, the embedding of the unique ID can take advantage of the texture of the physical paper, the background of the

electronic files, or the firm or screen. Those embodiments are especially suitable for scenarios that potential copyright infringers use scanned copies or pictures taken of the copyrighted content without permission from the right holder, the unique ID embedded in the physical media or electronic display that carry the copyrighted content can be extracted to identify potential copyright infringement.

[0066] FIG. 4 depicts an example of a content processing process 400 in accordance with embodiments of this specification. At a high-level, the process 400 can be performed by a computing device 401, a blockchain database 414, and a blockchain network 416. The computing device 401 can be communicably coupled to the blockchain network 416 through wired or wireless communications. Alternatively or additionally, the computing device 401 can be a consensus node of the blockchain network 416.

[0067] The computing device 401 can generate or update a digital content 402. The digital content can be 402 a copyrighted digital content. The computing device 401 can also perform blockchain-based copyright authentication based on blockchain transaction record 404 retrieved from a distributed ledger maintained by the blockchain network 416. In some embodiments, the blockchain transaction record 404 can be retrieved from the blockchain database 414 that maintains a complete ledger of the blockchain. Blockchain-based copyright authentication can be performed similarly to the description of step 304 of FIG. 3 to provide copyright authenticated content 406.

[0068] In some embodiments, the blockchain network 416 can be a consortium blockchain network. In such cases, copyrighted blockchain data can be managed by one or more primary nodes of the blockchain network 416. In some embodiments, the one or more primary nodes can issue a copyright certificate to the computing device 401 to authenticate the digital content 402 based on blockchain data stored in the blockchain database 412 or the consortium blockchain.

[0069] After copyright authentication, the digital content 402 becomes copyright authenticated content 406. The computing device 401 can then generate a unique ID 408 associated with copyright information of the digital content 402. The unique ID 408 can be invisibly embedded in the copyright authenticated content 406. Generation of unique ID 408 can be similarly performed as discussed in the description of step 306 of FIG. 3. Invisibly

embedding the unique ID 408 to copyright authenticated content 406 can be similarly performed as discussed in the description of 308 of FIG. 3.

[0070] After invisibly embedding copyright information, the computing device 401 can perform information embedded content distribution 410 to one or more nodes of the blockchain network 416. In some embodiments, blockchain nodes can generate blockchain data corresponding to the information embedded content. The blockchain data can include block data (block header) and block transactions (block body). Blockchain transactions corresponding to information embedded content distribution 410 or other content updates can be recorded to update the distributed ledger of the blockchain network 416.

[0071] FIG. 5 depicts an example of a blockchain transaction record 500 in accordance with embodiments of this specification. The blockchain transaction record 500 can be identified and retrieved based on a blockchain transaction ID 502 associated with the transaction. The blockchain transaction record 500 can include histories of copyrighted content such as time of the transaction 504, ownership information 506, sender's address 508, receiver's address 510, fees and prices 512, and block address corresponding to the transaction 514.

[0072] Referring back to FIG. 4, after information embedded content distribution 410 is performed, a blockchain database 414 communicably coupled to the blockchain network 416 can be updated 412. The blockchain database 414 can store updates of the information embedded content and newly performed transactions associated with the content. The updated blockchain transaction record 404 can include a longer chain-of-records associated with the digital content 402. As such, it can be more reliably used to authenticate the digital content 402 in future transactions.

[0073] FIG. 6 depicts an example of a copyright infringement determination process 600 in accordance with embodiments of this specification. At a high-level, the process 600 can be performed by a computing device 601, a blockchain network 612, and a blockchain database 610. The process 600 can be performed to detect potential copyright infringement based on copyright information embedded content 602. As discussed in the description of FIG. 3, a unique ID can be embedded in copyrighted content to produce the information embedded content 602. The unique ID can be invisibly embedded in pixels, angles, colors, fonts, texture, brightness, spatial or frequency coefficients, or other characteristics of data carriers based on

digital watermarking. Suitable data carriers can include text, punctuations, images, videos, forms, tables, physical media, electronic media, films or screens.

[0074] To police unauthorized use of copyrighted content, the computing device 601 can perform unique ID extraction 604 based on at least a portion of the information embedded content 602 found online or from a digital file. The extracted unique ID 606 can be used to identify transaction records associated with the copyrighted content from the blockchain network 612 or the blockchain database 610. If the transaction records are successfully identified and the right holder associated with the records does not match the user of the information embedded content 602, it can be determined that potential copyright infringement may have occurred. In some embodiments, the unique ID 606 can also be used to retrieve the latest copyrighted content stored in the blockchain database 610. The retrieved copyrighted content can be used as an authentic copy to determine whether another copy of the copyrighted content is tampered with.

[0075] FIG. 7 depicts an example of a process 700 that can be executed in accordance with embodiments of this specification. For clarity of presentation, the description that follows generally describes the example process 700 in the context of the other figures in this description. However, it will be understood that the example process 700 can be performed, for example, by any system, environment, software, and hardware, or a combination of systems, environments, software, and hardware, as appropriate. In some implementations, various steps of the example process 700 can be run in parallel, in combination, in loops, or in any order.

[0076] For convenience, the process 700 will be described as being performed by a one or more computing devices, located in one or more locations, and programmed appropriately in accordance with this specification. For example, a computing device in a computing system, e.g., the computing system 106, 108 of FIG. 1, appropriately programmed, can perform the process 700.

[0077] At 702, the computing device generates a unique ID based on copyright information associated with textual content, wherein the textual content and the copyright information are recorded on a blockchain of a blockchain network.

[0078] At 704, the computing device addes one or more function words to the textual content without altering a meaning of the textual content.

[0079] At 706, the computing device embeds the unique ID in the function words to produce an information-embedded textual content that enables retrieval of the copyright information from the blockchain based on the unique ID.

[0080] In some cases, a visual difference between the textual content and the information-embedded textual content is not apparent to an unaided human eye.

[0081] In some cases, before generating the unique ID, the computing device authenticates that a copyright holder holds a copyright of the textual content based on a blockchain transaction ID associated with a transaction that recorded the textual content in the blockchain; and after embedding the unique ID, the computing device distributes the information-embedded textual content to the blockchain network.

[0082] In some cases, the copyright information associated with the textual content includes one or more of at least a portion of the textual content or identity information of a copyright holder of the textual content.

[0083] In some cases, the textual content is recorded on the blockchain as a first transaction associated with a first blockchain transaction ID and the identity information is associated with a second transaction associated with a second blockchain transaction ID, and wherein the first blockchain transaction ID or the second blockchain transaction ID is associated with one or more transaction records associated with the textual content.

[0084] In some cases, the one or more transaction records includes one or more of creatorship, copyright ownership, update history, a timestamp, a location, or a license associated with the textual content.

[0085] In some cases, the one or more function words include one or more spaces.

[0086] In some cases, a computing device digitally signs the unique ID using a private key assigned to the copyright holder.

[0087] In some cases, the unique ID is generated based on one of hashing the copyright information using a hash function or encrypting the copyright information using a public key assigned to the copyright holder.

[0088] In some cases, the unique ID is further generated based on one or more of a timestamp or an address associated with the copyright information.

[0089] In some cases, the embedding the unique ID is performed based on digital watermarking using one or more of a discrete wavelet transform, a discrete cosine transform, a singular value decomposition, a least significant bit, or undetectable steganography.

[0090] In some cases, the unique ID is embedded in redundant pixels of the one or more function words.

[0091] In some cases, the one or more function words include one or more of articles, prepositions, conjunctions, or auxiliaries.

[0092] FIG. 8 is a diagram of an example of modules of an apparatus 800 in accordance with embodiments of this specification. The apparatus 800 can be an example of an embodiment of a client computing device configured to perform copyright information embedding for copyright protection. The apparatus 800 can correspond to the embodiments described above, and the apparatus 800 includes the following: a generating module 802 for generating a unique ID based on copyright information associated with textual content, wherein the textual content and the copyright information are recorded on a blockchain of a blockchain network; an adding module 804 for adding one or more function words to the textual content without altering a meaning of the textual content; an embedding module 806 for embedding the unique ID in the function words to provide an information-embedded textual content configured to enable retrieval of the copyright information from the blockchain based on the unique ID.

[0093] In an optional embodiment, a visual difference between the textual content and the information-embedded textual content is not apparent to an unaided human eye.

[0094] In an optional embodiment, the apparatus 800 further includes the following: an authentication sub-module to authenticate that a copyright holder holds a copyright of the textual content based on a blockchain transaction ID associated with a transaction that recorded the textual content in the blockchain before generating the unique ID; and a distribution sub-module to distribute the information-embedded textual content to the blockchain network after embedding the unique ID.

[0095] In an optional embodiment, the copyright information associated with the textual content includes one or more of at least a portion of the textual content or identity information of a copyright holder of the textual content.

[0096] In an optional embodiment, the textual content is recorded on the blockchain as a first transaction associated with a first blockchain transaction ID and the identity information is associated with a second transaction associated with a second blockchain transaction ID, and wherein the first blockchain transaction ID or the second blockchain transaction ID is associated with one or more transaction records associated with the textual content.

[0097] In an optional embodiment, the one or more transaction records include one or more of creatorship, copyright ownership, update history, a timestamp, a location, or a license associated with the textual content.

[0098] In an optional embodiment, the one or more function words include one or more spaces.

[0099] In an optional embodiment, the apparatus 800 further includes a digital signing sub-module to digitally sign the unique ID using a private key assigned to the copyright holder

[0100] In an optional embodiment, the unique ID is generated based on one of hashing the copyright information using a hash function or encrypting the copyright information using a public key assigned to the copyright holder.

[0101] In an optional embodiment, the unique ID is further generated based on one or more of a timestamp or an address associated with the copyright information.

[0102] In an optional embodiment, the embedding the unique ID is performed based on digital watermarking using one or more of a discrete wavelet transform, a discrete cosine transform, a singular value decomposition, a least significant bit, or undetectable steganography.

[0103] In an optional embodiment, the unique ID is embedded in redundant pixels of the one or more function words.

[0104] In an optional embodiment, the one or more function words include one or more of articles, prepositions, conjunctions, or auxiliaries.

[0105] The system, apparatus, module, or unit illustrated in the previous embodiments can be implemented by using a computer chip or an entity, or can be implemented by using a product having a certain function. A typical embodiment device is a computer, and the computer can be a personal computer, a laptop computer, a cellular phone, a camera phone, a smartphone, a personal digital assistant, a media player, a navigation device, an email

receiving and sending device, a game console, a tablet computer, a wearable device, or any combination of these devices.

[0106] For an embodiment process of functions and roles of each module in the apparatus, references can be made to an embodiment process of corresponding steps in the previous method. Details are omitted here for simplicity.

[0107] Because an apparatus embodiment basically corresponds to a method embodiment, for related parts, references can be made to related descriptions in the method embodiment. The previously described apparatus embodiment is merely an example. The modules described as separate parts may or may not be physically separate, and parts displayed as modules may or may not be physical modules, may be located in one position, or may be distributed on a number of network modules. Some or all of the modules can be selected based on actual demands to achieve the objectives of the solutions of the specification. A person of ordinary skill in the art can understand and implement the embodiments of the present application without creative efforts.

[0108] The techniques described in this specification produce several technical effects. For example, embodiments of the subject matter permit identification and authentication of creators or right holders of copyrighted content based on transaction records stored on a blockchain. The transaction records can be used as evidence to resolve copyright dispute between alleged right holders. The transaction records can also be used for seeking permissions from the authenticated right holders to copy, modify, license, distribute, or publicly display the copyrighted content. Moreover, records on blockchain can be used by right holders to identify and police copyright infringement.

[0109] In some embodiments, a unique ID can be invisibly embedded in copyrighted content as a digital watermark to identify potential copyright infringement or illegal changes of copyrighted content. Because the digital watermark is invisibly embedded, it does not visibly affect the presentation of the copyrighted content. The invisible digital watermark can also be difficult to discern and remove by potential infringers to counter infringement detection.

[0110] To police unauthorized use of copyrighted content, the unique ID can be extracted to retrieve transaction records of the copyrighted content from the blockchain. If the transaction records are identified and the user of the copyrighted content is not the right

holder or a licensee of the copyrighted content, it can be determined that potential copyright infringement may have occurred. In some embodiments, the unique ID can also be used to retrieve the latest copyrighted content stored on the blockchain to determine whether the copyrighted content is tampered with.

[0111] The unique ID can also be embedded in physical media, or a screen attached or integrated to an electronic display. When a potential infringer uses scanned copy or pictures taken of the copyrighted content without permission from the right holder, the unique ID embedded in the physical media or electronic display that carry the copyrighted content can be extracted to identify potential copyright infringement.

[0112] Described embodiments of the subject matter can include one or more features, alone or in combination.

[0113] For example, in a first embodiment, a computer-implemented method for embedding copyright information to content comprises: generating, by a computing device, a unique ID based on copyright information associated with the textual content, wherein the textual content and the copyright information are recorded on a blockchain of a blockchain network; adding, by the computing device, one or more function words to the textual content without altering a meaning of the textual content; and embedding, by the computing device, the unique ID in the function words to produce an information-embedded textual content that enables retrieval of the copyright information from the blockchain based on the unique ID

[0114] The foregoing and other described embodiments can each, optionally, include one or more of the following features:

[0115] A first feature, combinable with any of the following features, further comprising: before generating the unique ID, authenticating that a copyright holder holds a copyright of the textual content based on a blockchain transaction ID associated with a transaction that recorded the textual content in the blockchain; and after embedding the unique ID, distributing the information-embedded textual content to the blockchain network.

[0116] A second feature, combinable with any of the previous or following features, specifies that the copyright information associated with the textual content includes one or more of at least a portion of the textual content or identity information of a copyright holder of the textual content.

[0117] A third feature, combinable with any of the previous or following features, specifies that the textual content is recorded on the blockchain as a first transaction associated with a first blockchain transaction ID and the identity information is associated with a second transaction associated with a second blockchain transaction ID, and wherein the first blockchain transaction ID or the second blockchain transaction ID is associated with one or more transaction records associated with the textual content.

[0118] A fourth feature, combinable with any of the previous or following features, specifies that the one or more transaction records includes one or more of creatorship, copyright ownership, update history, a timestamp, a location, or a license associated with the textual content.

[0119] A fifth feature, combinable with any of the previous or following features, specifies that the one or more function words include one or more spaces.

[0120] A sixth feature, combinable with any of the previous or following features, further comprising digitally signing the unique ID using a private key assigned to the copyright holder.

[0121] A seventh feature, combinable with any of the previous or following features, specifies that the unique ID is generated based on one of hashing the copyright information using a hash function or encrypting the copyright information using a public key assigned to the copyright holder.

[0122] An eighth feature, combinable with any of the previous or following features, specifies that the unique ID is further generated based on one or more of a timestamp or an address associated with the copyright information.

[0123] A ninth feature, combinable with any of the previous or following features, specifies that the embedding the unique ID is performed based on digital watermarking using one or more of a discrete wavelet transform, a discrete cosine transform, a singular value decomposition, a least significant bit, or undetectable steganography.

[0124] A tenth feature, combinable with any of the previous or following features, specifies that the unique ID is embedded in redundant pixels of the one or more function words.

[0125] An eleventh feature, combinable with any of the previous or following features, specifies that the one or more function words include one or more of articles, prepositions, conjunctions, or auxiliaries.

[0126] An twelfth feature, combinable with any of the previous or following features, specifies that a visual difference between the textual content and the information-embedded textual content is not apparent to an unaided human eye.

[0127] Embodiments of the subject matter and the actions and operations described in this specification can be implemented in digital electronic circuitry, in tangibly-embodied computer software or firmware, in computer hardware, including the structures disclosed in this specification and their structural equivalents, or in combinations of one or more of them. Embodiments of the subject matter described in this specification can be implemented as one or more computer programs, e.g., one or more modules of computer program instructions, encoded on a computer program carrier, for execution by, or to control the operation of, data processing apparatus. For example, a computer program carrier can include one or more computer-readable storage media that have instructions encoded or stored thereon. The carrier may be a tangible non-transitory computer-readable medium, such as a magnetic, magneto optical, or optical disk, a solid state drive, a random access memory (RAM), a read-only memory (ROM), or other types of media. Alternatively, or in addition, the carrier may be an artificially generated propagated signal, e.g., a machine-generated electrical, optical, or electromagnetic signal that is generated to encode information for transmission to suitable receiver apparatus for execution by a data processing apparatus. The computer storage medium can be or be part of a machine-readable storage device, a machine-readable storage substrate, a random or serial access memory device, or a combination of one or more of them. A computer storage medium is not a propagated signal.

[0128] A computer program, which may also be referred to or described as a program, software, a software application, an app, a module, a software module, an engine, a script, or code, can be written in any form of programming language, including compiled or interpreted languages, or declarative or procedural languages; and it can be deployed in any form, including as a stand-alone program or as a module, component, engine, subroutine, or other unit suitable for executing in a computing environment, which environment may

include one or more computers interconnected by a data communication network in one or more locations.

[0129] A computer program may, but need not, correspond to a file in a file system. A computer program can be stored in a portion of a file that holds other programs or data, e.g., one or more scripts stored in a markup language document, in a single file dedicated to the program in question, or in multiple coordinated files, e.g., files that store one or more modules, sub programs, or portions of code.

[0130] Processors for execution of a computer program include, by way of example, both general- and special-purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor will receive the instructions of the computer program for execution as well as data from a non-transitory computer-readable medium coupled to the processor.

[0131] The term "data processing apparatus" encompasses all kinds of apparatuses, devices, and machines for processing data, including by way of example a programmable processor, a computer, or multiple processors or computers. Data processing apparatus can include special-purpose logic circuitry, e.g., an FPGA (field programmable gate array), an ASIC (application specific integrated circuit), or a GPU (graphics processing unit). The apparatus can also include, in addition to hardware, code that creates an execution environment for computer programs, e.g., code that constitutes processor firmware, a protocol stack, a database management system, an operating system, or a combination of one or more of them.

[0132] The processes and logic flows described in this specification can be performed by one or more computers or processors executing one or more computer programs to perform operations by operating on input data and generating output. The processes and logic flows can also be performed by special-purpose logic circuitry, e.g., an FPGA, an ASIC, or a GPU, or by a combination of special-purpose logic circuitry and one or more programmed computers.

[0133] Computers suitable for the execution of a computer program can be based on general or special-purpose microprocessors or both, or any other kind of central processing unit. Generally, a central processing unit will receive instructions and data from a read only memory or a random access memory or both. Elements of a computer can include a central

processing unit for executing instructions and one or more memory devices for storing instructions and data. The central processing unit and the memory can be supplemented by, or incorporated in, special-purpose logic circuitry.

[0134] Generally, a computer will also include, or be operatively coupled to receive data from or transfer data to one or more storage devices. The storage devices can be, for example, magnetic, magneto optical, or optical disks, solid state drives, or any other type of non-transitory, computer-readable media. However, a computer need not have such devices. Thus, a computer may be coupled to one or more storage devices, such as, one or more memories, that are local and/or remote. For example, a computer can include one or more local memories that are integral components of the computer, or the computer can be coupled to one or more remote memories that are in a cloud network. Moreover, a computer can be embedded in another device, e.g., a mobile telephone, a personal digital assistant (PDA), a mobile audio or video player, a game console, a Global Positioning System (GPS) receiver, or a portable storage device, e.g., a universal serial bus (USB) flash drive, to name just a few.

[0135] Components can be "coupled to" each other by being commutatively such as electrically or optically connected to one another, either directly or via one or more intermediate components. Components can also be "coupled to" each other if one of the components is integrated into the other. For example, a storage component that is integrated into a processor (e.g., an L2 cache component) is "coupled to" the processor.

[0136] To provide for interaction with a user, embodiments of the subject matter described in this specification can be implemented on, or configured to communicate with, a computer having a display device, e.g., a LCD (liquid crystal display) monitor, for displaying information to the user, and an input device by which the user can provide input to the computer, e.g., a keyboard and a pointing device, e.g., a mouse, a trackball or touchpad. Other kinds of devices can be used to provide for interaction with a user as well; for example, feedback provided to the user can be any form of sensory feedback, e.g., visual feedback, auditory feedback, or tactile feedback; and input from the user can be received in any form, including acoustic, speech, or tactile input. In addition, a computer can interact with a user by sending documents to and receiving documents from a device that is used by the user; for example, by sending web pages to a web browser on a user's device in response to requests received from the web browser, or by interacting with an app running on a user device, e.g., a

smartphone or electronic tablet. Also, a computer can interact with a user by sending text messages or other forms of message to a personal device, e.g., a smartphone that is running a messaging application, and receiving responsive messages from the user in return.

[0137] This specification uses the term "configured to" in connection with systems, apparatus, and computer program components. For a system of one or more computers to be configured to perform particular operations or actions means that the system has installed on it software, firmware, hardware, or a combination of them that in operation cause the system to perform the operations or actions. For one or more computer programs to be configured to perform particular operations or actions means that the one or more programs include instructions that, when executed by data processing apparatus, cause the apparatus to perform the operations or actions. For special-purpose logic circuitry to be configured to perform particular operations or actions means that the circuitry has electronic logic that performs the operations or actions.

[0138] While this specification contains many specific embodiment details, these should not be construed as limitations on the scope of what is being claimed, which is defined by the claims themselves, but rather as descriptions of features that may be specific to particular embodiments. Certain features that are described in this specification in the context of separate embodiments can also be realized in combination in a single embodiment. Conversely, various features that are described in the context of a single embodiments can also be realized in multiple embodiments separately or in any suitable subcombination. Moreover, although features may be described above as acting in certain combinations and even initially be claimed as such, one or more features from a claimed combination can in some cases be excised from the combination, and the claim may be directed to a subcombination or variation of a subcombination.

[0139] Similarly, while operations are depicted in the drawings and recited in the claims in a particular order, this should not be understood as requiring that such operations be performed in the particular order shown or in sequential order, or that all illustrated operations be performed, to achieve desirable results. In certain circumstances, multitasking and parallel processing may be advantageous. Moreover, the separation of various system modules and components in the embodiments described above should not be understood as requiring such separation in all embodiments, and it should be understood that the described

program components and systems can generally be integrated together in a single software product or packaged into multiple software products.

[0140] Particular embodiments of the subject matter have been described. Other embodiments are within the scope of the following claims. For example, the actions recited in the claims can be performed in a different order and still achieve desirable results. As one example, the processes depicted in the accompanying figures do not necessarily require the particular order shown, or sequential order, to achieve desirable results. In some cases, multitasking and parallel processing may be advantageous.

What is claimed is:

CLAIMS

1. A computer-implemented method for embedding copyright information in textual content, the method comprising:

generating, by a computing device, a unique identifier (ID) based on copyright information associated with the textual content, wherein the textual content and the copyright information are recorded on a blockchain of a blockchain network;

adding, by the computing device, one or more function words to the textual content without altering a meaning of the textual content; and

embedding, by the computing device, the unique ID in hidden characteristics of the added one or more function words to produce an information-embedded textual content that enables retrieval of the copyright information from the blockchain based on the unique ID, wherein the added one or more function words include one or more of articles, prepositions, conjunctions, or auxiliaries.

2. The computing-implemented method of claim 1, wherein a visual difference between the textual content and the information-embedded textual content is not apparent to an unaided human eye.

3. The computer-implemented method of any one of the preceding claims, further comprising:

before generating the unique ID, authenticating, by the computing device, that a copyright holder holds a copyright of the textual content based on a blockchain transaction ID associated with a transaction that recorded the textual content in the blockchain.

4. The computer-implemented method of any one of the preceding claims, further comprising:

after embedding the unique ID, distributing, by the computing device, the information-embedded textual content to the blockchain network.

5. The computer-implemented method of claim 1, wherein the copyright information associated with the textual content includes one or more of at least a portion of the textual content or identity information of a copyright holder of the textual content.

6. The computer-implemented method of claim 5, wherein the textual content is recorded on the blockchain as a first transaction associated with a first blockchain transaction ID and the identity information is associated with a second transaction associated with a second blockchain transaction ID, and wherein the first blockchain transaction ID or the second blockchain transaction ID is associated with one or more transaction records associated with the textual content.

7. The computer-implemented method of claim 6, wherein the one or more transaction records include one or more of creatorship, copyright ownership, update history, a timestamp, a location, or a license associated with the textual content.

8. The computer-implemented method of any one of the preceding claims, further comprising:

digitally signing, by the computing device, the unique ID using a private key assigned to the copyright holder.

9. The computer-implemented method of any one of the preceding claims, wherein the unique ID is generated based on one of hashing the copyright information using a hash function or encrypting the copyright information using a public key assigned to the copyright holder.

10. The computer-implemented method of any one of the preceding claims, wherein the unique ID is further generated based on one or more of a timestamp or an address associated with the copyright information.

11. The computer-implemented method of any one of the preceding claims, wherein the embedding the unique ID is performed based on digital watermarking using one or more of a

discrete wavelet transform, a discrete cosine transform, a singular value decomposition, a least significant bit, or undetectable steganography.

12. A system for embedding copyright information to content, comprising:

one or more processors; and

one or more computer-readable memories coupled to the one or more processors and having instructions stored thereon which are executable by the one or more processors to perform the method of any one of claims 1-11.

13. An apparatus for embedding copyright information to content, the apparatus comprising a plurality of modules for performing the method of any one of claims 1-11.