ANNEXURE A

1.Relevant Citations

Patent Citation 1: US20250342232A1

Title	Protecting tokenized structures using a protection architecture
Priority Date	01 MAY 2024
Filing Date	01 MAY 2024
Publication Date	06 NOV 2025
Inventors	Shepherd Matthew Mullin; Eutsler Nathaniel C.; Natarajan Sivakumar; Varley Piers
Assignees	Wells Fargo Bank NA
IPC Classes	G06F21/10; G06Q20/36
CPC Classes	G06F21/1014; G06F21/1085; G06Q20/36
US Classes	None
Family Members	None

Abstract:

Systems, methods, and computer-readable storage media to protect tokens using a protection architecture. One method includes identifying asset tokens including links to a plurality of asset metadata objects. Further, the method includes generating a container metadata object including metadata of the asset tokens. Further, the method includes generating a container token including a link with the container metadata object. Further, the method includes encapsulating the container token and the asset tokens within a container including a container control structure restricting outputs of the container metadata object and the plurality of asset metadata objects. Further, the method includes generating an allocation token compatible with a segmented allocation control structure restricting outputs by the container of a first segmented allocation of the asset tokens based on metadata of a subset of the plurality of asset metadata objects. Further, the method includes providing, using the segmented allocation control structure, the allocation token.

Key Features	Relevant Excerpts
1. A system for programmable video assembly, comprising:	**Para [0030]** The technical solution described herein can including **smart contract control structures and a secure container that encapsulates one or more tokens.** The smart contract control structures can allow output of various metadata linked to the tokens upon detection of NFTs, semi-fungible tokens, or fungible tokens compatible with the smart contract control structures or particular requests (e.g., distribution, exchange, withdrawal, deposit, exchange instrument, on-us exchange). For example, **the smart contract control structures can be restricted to execution at a particular computing environment by a container token restricted to within the particular computing environment.** The smart contract control structures, and the tokens within the smart contract control structures, can be rendered unusable outside the particular computing environment. This technical solution can include multiple layers of secure access control to tokens, including authorization control at a smart contract layer by one or more tokens, and authorization control at a container layer by a private key. The private key can be based on one or more tokens, and can be fully contained within a single tokens or partially contained within multiple tokens. **This technical solution can include generation of smart contract**

	control structures and modification of blockchain architecture to restrict particular tokens. A smart contract control structure can, for example, generate or modify a smart contract to contain one or more particular tokens. The smart contract control structures can search a blockchain to identify tokens satisfying particular attributes, parameters, or metrics. The parameters can be transmitted to the generated smart contract by a token. The generated smart contract control structures can generate a token that can include an NFT, a semi-fungible toke, or a fungible token, and can distribute that token while retaining locally the smart contract and its restricted tokens. **Para [0037]** …The metadata I/O processor 110 can identify one or more characteristics of a metadata object. For example, **the metadata I/O processor 110 can obtain and identify metadata objects of a segmented allocation including video,** audio, text, any media, executable programs, or any combination thereof. The metadata I/O processor 110 can transmit one or more of metadata objects or references or links with one or more metadata objects to the token generator 112. **Note:** *The mapped citation describes a system for programmable video assembly in which smart contract control structures are tightly bound to a specific computing environment. This restriction is enforced through a container token that ensures execution only within the designated environment. In this framework, the metadata I/O processor plays a crucial role by obtaining and identifying metadata objects from segmented allocations, including video segments, thereby enabling controlled and secure assembly of programmable video content.*
1.1 a virtualization engine, wherein the engine is configured to convert a rendered video file into a reference container by extracting metadata and removing media sample data;	**Para [0037]** …The metadata I/O processor 110 can identify one or more characteristics of a metadata object. For example, **the metadata I/O processor 110 can obtain and identify metadata objects of a segmented allocation including video,** audio, text, any media, executable programs, or any combination thereof. The metadata I/O processor 110 can transmit one or more of metadata objects or references or links with one or more metadata objects to the token generator 112. **Para [0003]** Some implementations relate to a system, including a data processing system including memory and one or more processing circuits configured to identify a plurality of asset tokens including links to a plurality of asset metadata objects. **The one or more processing circuits are further configured to generate a container metadata object including metadata of the plurality of asset tokens. The one or more processing circuits are further configured to generate a container token including a link with the container metadata object. The one or more processing circuits are further configured to encapsulate the container token and the plurality of asset tokens within a container including a container control structure restricting outputs of the container metadata object and the plurality of asset metadata objects**. The one or more processing circuits are further configured to generate an allocation token compatible with a segmented allocation control structure restricting outputs by the container of a first segmented allocation of the plurality of asset tokens based on metadata of a subset of the plurality of asset metadata objects. The one or more processing circuits are further configured to provide, by the segmented allocation control structure to a client system, the allocation token. **Para [0072]** Although shown in the arrangements of FIG. 2 as singular, stand-alone devices, one of ordinary skill in the art will appreciate that, **in some arrangements, the computing system 200 may include virtualized systems and/or system resources.** For example, in some arrangements, the computing system 200 may be a virtual switch, virtual router, virtual host, virtual server. In various arrangements, computing system 200 may share physical storage, hardware, and other resources with other virtual machines.

	[Para 0038] The token generator 112 it at least one processor structured or configured to generate and modify one or more smart contracts. The token generator 112 can execute instructions to generate or modify a cryptographic container, to add or **remove objects from a cryptographic container**, and to execute various processors linked with or embedded with a smart contract. **Note:** *The mapped citation describes a system for the computing environment which may include virtualized systems and resources, and the processing circuits generate a container metadata object from multiple asset tokens. These circuits then create a container token linking to the metadata object and encapsulate both the container token and asset tokens inside a container. The container incorporates a control structure that restricts outputs of the container metadata object and the asset metadata objects, ensuring secure, organized, and controlled assembly of video-related metadata.*
1.1.1 wherein the virtual video container retains ISO Base Media File Format box structures while being devoid of media sample data.	**Para [0004]** In some implementations**, the one or more processing circuits are further configured to generate, by the segmented allocation control structure, a plurality of segmented allocations within the container based on an output of the container control structure corresponding to access to the plurality of asset metadata objects.** **Para [0037]** …The metadata I/O processor 110 can identify one or more characteristics of a metadata object. For example, **the metadata I/O processor 110 can obtain and identify metadata objects of a segmented allocation including video,** audio, text, any media, executable programs, or any combination thereof. The metadata I/O processor 110 can transmit one or more of metadata objects or references or links with one or more metadata objects to the token generator 112. **Para [0126]** At block 520, the processors (e.g., data processing system 102) can generate a container metadata object including metadata of the plurality of asset tokens-controllable electronic records. **For example, the container metadata object can include records of each asset token's financial status, ownership, and transaction history. In some implementations, generating can include compiling data from various sources into a standardized format.** Furthermore, generating the container metadata object can include a link to metadata objects of the plurality of asset tokens… **Note:** *The mapped citation discloses the metadata I/O processor can obtain and identify metadata objects of segmented allocations, including video. The container metadata object may include records such as each asset token's financial status, ownership, and transaction history. In some implementations, the generation process compiles data from various sources into a standardized format, which means that the segmented allocations within the container and container metadata objects retains a standardized format. However, ISO Base Media File Format box structures are not as such disclosed.*
1.1.2 wherein the step of resolving referenced media data involves issuing MDAT range requests to the remote media data sources.	**Para [0037]** …The metadata I/O processor 110 can identify one or more characteristics of a metadata object. For example, **the metadata I/O processor 110 can obtain and identify metadata objects of a segmented allocation including video,** audio, text, any media, executable programs, or any combination thereof. The metadata I/O processor 110 can transmit one or more of metadata objects or references or links with one or more metadata objects to the token generator 112. **Para [0088]** Each asset token 330 can include a link 332 to an asset metadata object 334, which stores detailed information about the underlying asset, such as its financial attributes, ownership history, and performance data. The asset token 330 can be a controllable electronic record. **Link 332 can include a reference, pointer, or the like, to or between an asset token 330 and an asset metadata object 334. Link**

	332 can be used to ensure that every asset token 330 has accessible, up-to-date information on its characteristics. For example, potential investors can examine the metadata to assess risk and potential returns before acquiring an asset token 330. **Asset tokens 330 can each include a particular fungible or non-fungible token and can correspond to particular asset metadata objects 334. An asset token 330 can be associated with a particular asset metadata object, and can be required to transmit output of the metadata object, transfer the metadata object to another storage location, or any combination thereof, for example.** Each of the asset token 330 can indicate control of a particular metadata object of the asset metadata objects 334 by a corresponding metadata link of the metadata links 332. The asset metadata objects 224 can each include a particular data or instructions… **Note:** *The mapped citation discloses the metadata I/O processor can obtain and identify metadata objects of segmented allocations for video. The link provides a reference or pointer between an asset token and its associated asset metadata object (i.e., part of media data), ensuring that each asset token has accessible and up-to-date information on its characteristics. Just as MDAT range requests retrieve specific portions of remote media data, asset tokens may be required to transmit or transfer outputs of their metadata objects to other storage locations (say remote sources), thereby maintaining controlled, precise, and verifiable access to distributed resources. However, MDAT range specifically not disclosed.*
1.1.3 further comprising an artificial intelligence orchestration layer configured to modify the sequencing instructions of the reference container without accessing raw media samples.	**Para [0042]** The allocation processor 130 is at least one processor in the data processing system 102 that is structured or configured to execute operations related to the generation and management of segmented allocation smart contract control structures. **In some implementations, the allocation processor 130 can process tasks and actions such as classifying asset tokens within container tokens based on parameters set in the smart contracts. The allocation processor 130 can interact with blockchain 158 to deploy the contracts, dynamically adjust asset segmentation in response to changes in metadata, and manage the issuance and authentication of allocation tokens.** Additionally, the allocation processor 130 can interface with the blockchain 158 to record and maintain logs of all distributions and modifications in token allocations, aligned with the predefined rules and conditions of the smart contracts. **Para [0152]** In some implementations, **artificial intelligence (AI) and/or generative AI (GAI) can be used by the processing circuits to perform method 500.** Generally, one or more AI models, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), decision trees, support vector machines (SVM) for classification, gradient boosting machines (GBM) for predictive analytics, and decision trees for rule-based decision making can be used by the processing circuits to perform method 500. For example, **at block 510 the processing circuits can employ an AI model (e.g., SVM) to classify asset tokens based on their attributes and historical data. In another example, at block 520 the processing circuits can use an AI model (e.g., GBM) to forecast the potential future values of assets based on trends derived from historical transaction data.** In another example, at block 530 the processing circuits can utilize decision trees to determine the criteria for minting container tokens, incorporating logic that addresses various contingencies in token attributes. In yet another example, at block 540 the processing circuits can implement cryptographic secure hash algorithms like SHA-256 to encrypt token data. In yet another example, at block 550 the processing circuits can apply logistic regression models to determine the probability of asset performance meeting investor expectations. **In yet another example, at block 560 the processing circuits can use rule-based AI systems to automatically check and enforce compliance with smart contract conditions when issuing allocation tokens**… **Note:** *The mapped citation describes the allocation processor process tasks and*

	actions such as classifying asset tokens within container tokens based on parameters set in the smart contracts. And artificial intelligence (AI) is being used by the processing circuits to perform the method where the AI model (e.g., SVM) classify asset tokens based on their attributes and historical data and automatically check and enforce compliance with smart contract conditions when issuing allocation tokens, which means that the AI model is performs actions on asset tokens within container tokens without accessing the raw data.
1.2 a token issuance service, wherein the service configured to generate a cryptographicvideo token binding transaction, consent, and licensing information;	**Para [0037]** …The metadata I/O processor 110 can identify one or more characteristics of a metadata object. For example, **the metadata I/O processor 110 can obtain and identify metadata objects of a segmented allocation including video,** audio, text, any media, executable programs, or any combination thereof. The metadata I/O processor 110 can transmit one or more of metadata objects or references or links with one or more metadata objects to the token generator 112. **Para [0038]** The token generator 112 it at least one processor structured or configured to generate and modify one or more smart contracts. **The token generator 112 can execute instructions to generate or modify a cryptographic container, to add or remove objects from a cryptographic container, and to execute various processors linked with or embedded with a smart contract.** For example, the token generator 112 can execute various processors of a smart contract in response to an indication from the metadata I/O processor 110 that a performance metric satisfies a particular threshold for distribution… **Para [0042]** …**The allocation processor 130 can interact with blockchain 158 to deploy the contracts, dynamically adjust asset segmentation in response to changes in metadata, and manage the issuance and authentication of allocation tokens. Additionally, the allocation processor 130 can interface with the blockchain 158 to record and maintain logs of all distributions and modifications in token allocations, aligned with the predefined rules and conditions of the smart contracts.** **Para [0045]** The NFT storage 152 can store one or more NFTs and corresponding addresses for particular NFTs that indicate links with the corresponding NFT. The NFT storage 152 can include NFTs associated with the data processing system 102 or any component thereof, the client system 103 or any component thereof, any metadata object, or any combination thereof. **The key dataset 159 can store cryptographic keys associated with the data processing system 102 or any component thereof, the client system 103 or any component thereof, any metadata object, or any combination thereof.** For example, the key dataset 159 can include public-private key pairs or private keys corresponding to particular accounts, NFTs, smart contracts, devices, users, systems, or any combination thereof. **Para [0054]** In some implementations, the cryptographic key processor 120 can sign the NFT using a private key and verify the NFT using a public key. Thus, in some implementations, **verifying can include decrypting the NFT using the public key to verify the digital signature came from the particular private key (e.g., particular digital wallet of a user), and signing can include encrypting the NFT using the private key to create a digital signature**… **Note:** *The mapped citation describes a token generator (i.e., token issuance service) can execute instructions to generate or modify a cryptographic container and the cryptographic keys are being stored in the key data sets. The blockchain deploy manages the asset segmentation in response to changes in metadata, and manage the issuance and authentication of allocation tokens, also token allocations are aligned with the predefined rules and conditions of the smart contracts, which means that the video segmentation allocations are used in the blockchain process and used to generate a cryptographic video token binding with some set of regulation under*

	smart contracts which includes dynamic allocations of segmentations (i.e., transactions), digital signatures (i.e., consent).
1.2.1 wherein the token issuance service includes a unique session nonce in each generated token to prevent unauthorized resolution reuse.	**Para [0037]** …The metadata I/O processor 110 can identify one or more characteristics of a metadata object. For example, **the metadata I/O processor 110 can obtain and identify metadata objects of a segmented allocation including video, audio, text, any media,** executable programs, or any combination thereof… **Para [0038]** The token generator 112 it at least one processor structured or configured to generate and modify one or more smart contracts. **The token generator 112 can execute instructions to generate or modify a cryptographic container, to add or remove objects from a cryptographic container, and to execute various processors linked with or embedded with a smart contract.** For example, the token generator 112 can execute various processors of a smart contract in response to an indication from the metadata I/O processor 110 that a performance metric satisfies a particular threshold for distribution… **Para [0054]** In some implementations, **the cryptographic key processor 120 can sign the NFT using a private key and verify the NFT using a public key. Thus, in some implementations, verifying can include decrypting the NFT using the public key to verify the digital signature came from the particular private key (e.g., particular digital wallet of a user), and signing can include encrypting the NFT using the private key to create a digital signature**… **Para [0055]** Still referring to FIG. 1 , in some implementations, the interface controller 120 can establish a data channel between a source address and a destination address, such that receivals or transmissions of a token or token distributions occurs between the addresses on a ledger (e.g., blockchain storage 158) and/or a digital wallet (e.g., wallet system 105). **An address can be generated based on executing, by the cryptographic key processor 114, a math-based function (e.g., hash, symmetric encryption, asymmetric encryption) on a public key of a public and private key pair (or a verification key of a verification and signing key pair)**. For example, if an interface controller 120 receives a token from any system or device described herein, the token or other data received may include metadata associated with a source address, and the interface controller 120 may determine a destination address (e.g., may be provided to the system sending the NFT in advance) to store the token or provide a distribution in the blockchain storage 158. **In various implementations, the addresses may be a unique sequence of randomized (or pseudo-randomized) numerical digits, characters, punctuation, whitespace, code (e.g., QR), or symbols.** **Para [0056]** …In some implementations, **the data processing system 102 can maintain (e.g., store and access keys) the key dataset 159 such that each token may be locked-unlocked and associated with a public key or public-private key pair stored on the key dataset 159. In various implementations, public-private key pairs can be shared amongst a plurality of tokens or can be unique to each token on the blockchain storage 158.** **Para [0128]** At block 540, the processors (e.g., data processing system 102) can encapsulate the container token and the plurality of asset tokens within a container including a container control structure restricting outputs of the container metadata object and the plurality of asset metadata objects. Generally, **encapsulating can include securing the tokens and their metadata. For example, the encapsulation process could include applying encryption techniques to the data. That is, encapsulating can ensure that sensitive information remains accessible only to authorized parties. In some implementations, the container can be a digitally secured environment and encapsulating the container where the container control structure restricts outputs includes applying access control rules and monitoring**

	interactions with the token data. For example, restricting outputs can be related to controlling who can access metadata of the asset tokens or container tokens. In some implementations, the container control structure can be a set of smart contracts that manage access rights and data integrity. **Note:** *The mapped citation describes token generator (i.e., the token issuance service) manages a cryptographic container linked with the smart contract. The cryptographic keys for each token to be locked-unlocked and the token are associated with an address that is as per unique sequence of randomized (or pseudo-randomized) numerical digits. Also there is encapsulation for securing the tokens which ensures that sensitive information remains accessible only to authorized parties (i.e., prevent unauthorized use), which is same in case of resolution use of a video.*
1.2.2 wherein the transaction metadata includes instructions for micro payments triggered by individual segment resolution events	**Para [0037]** …The metadata I/O processor 110 can identify one or more characteristics of a metadata object. For example, **the metadata I/O processor 110 can obtain and identify metadata objects of a segmented allocation including video, audio, text, any media,** executable programs, or any combination thereof… **Para [0043] The allocation processor 130 in the data processing system 102 can be also configured to manage the computational tasks for determining and updating the segmentation of asset tokens within container tokens based on defined parameters.** These parameters may include financial metrics such as loan type, maturity date, interest rate, credit risk, payment history, and principal amount remaining. The allocation processor 130 can generate and use the parameters to classify and reclassify asset tokens into appropriate segmented allocations, such as tranches or risk pools, in accordance with the smart contract specifications. **Additionally, the allocation processor 130 can initiate distributions based on performance metrics. The allocation processor 130 can be programmed to automatically calculate and trigger financial distributions or other outputs once the underlying asset tokens meet specific performance thresholds, which can be detailed in the smart contracts (e.g., in smart contract storage 156).** This can include evaluating compliance with payment schedules and adjusting distributions as asset conditions change, providing that token holders (e.g., client systems 103 and third-party devices 106) receive returns that reflect the current performance of their investment holdings. The allocation processor 130 can interact with blockchain 158 to execute these functions. **Para [0066]** The data processing system 102 (in particular, interface controller 120) can be configured to process exchanges of tokens (e.g., withdrawal, deposit, update) and may be configured to perform various actions and/or access various types of data or metadata, some of which may be provide over network 101. **In particular, the interface controller 120 can be configured to process token exchanges based on received public keys (or public and private key pairs, or private key), environmental data, off-chain data, and metadata of one or more tokens (e.g., fungible or non-fungible) stored by and on data processing system 102 (e.g., in 152, 154, and 158) from the systems and devices described herein.** In some implementations, exchanges of tokens on-chain or off-chain include utilizing a control structure specific to the token or group of tokens (e.g., within a container). Although the FIGS. and specification generally discuss **utilizing control structures on token exchanges and distributions (e.g., withdrawals, deposits, updates),** the systems, methods, and apparatuses disclosed herein can also be used for a plurality of tokens such as, but not limited to, utility tokens, security tokens, payment tokens, exchange tokens, decentralized finance (DeFi) tokens, stablecoins, asset-backed tokens, privacy tokens, and so on. Additional details and examples relating to exchanging and restricting tokens or distributing underlying asset incomes or

	expenses of the tokens are described in detail with reference to FIGS. 3-4 . **Note:** *The mapped citation describes the system can be programmed to automatically calculate and trigger financial distributions (e.g., withdrawals, deposits, updates) or other outputs once the underlying asset tokens meet specific performance thresholds, same in case of resolution use, the payment distribution is utilized as the mechanism is same for the video allocation segmentation.*
1.2.3 wherein the cryptographic video token may function as an integrated authorization gatekeeper that operates independently of the media payload by excluding media sample data and direct file paths, enforces the right to resolve by providing the necessary cryptographic permissions and binds governance logic to the resolution event.	**Para [0037]** …The metadata I/O processor 110 can identify one or more characteristics of a metadata object. For example, **the metadata I/O processor 110 can obtain and identify metadata objects of a segmented allocation including video, audio, text, any media,** executable programs, or any combination thereof… **Para [0055]** Still referring to FIG. 1 , in some implementations, the interface controller 120 can establish a data channel between a source address and a destination address, such that receivals or transmissions of a token or token distributions occurs between the addresses on a ledger (e.g., blockchain storage 158) and/or a digital wallet (e.g., wallet system 105). **An address can be generated based on executing, by the cryptographic key processor 114, a math-based function (e.g., hash, symmetric encryption, asymmetric encryption) on a public key of a public and private key pair (or a verification key of a verification and signing key pair)**. For example, if an interface controller 120 receives a token from any system or device described herein, the token or other data received may include metadata associated with a source address, and the interface controller 120 may determine a destination address (e.g., may be provided to the system sending the NFT in advance) to store the token or provide a distribution in the blockchain storage 158. **In various implementations, the addresses may be a unique sequence of randomized (or pseudo-randomized) numerical digits, characters, punctuation, whitespace, code (e.g., QR), or symbols.** **Para [0093]** Sill referring to FIG. 3 , the container smart contract control structure 310 can execute and manage the functions related to both the container tokens 320 and the asset tokens 330. The container smart contract control structure 310 can enforce the conditions under which data from the container metadata object 326 and the asset metadata objects 334 can be accessed and modified. **The container smart contract control structure 310 can restricts outputs of the container metadata object 326 and asset metadata objects 334 by coded rules within the smart contract code of container smart contract control structure 310 that govern the release and modification of metadata-controlling the visibility and update permissions.** **Para [0102] The segmented allocation smart contract control structure 350 can utilize the allocation tokens 382 to manage and authorize specific actions. For example, an output of financial distributions or sensitive data from the asset tokens 330 may be contingent upon the presence of an allocation token 382, verified by the segmented allocation smart contract control structure 350. That is, the presence of the allocation token 382 can include validating the allocation token's 382 existence and its associated rights within a transaction or access request. When a request or distribution is made, the segmented allocation smart contract control structure 350 can check if the requester or user receiving a distribution holds a valid allocation token 382, verifying its authenticity and permissions encoded within the token**. **Note:** *The mapped citation describes the metadata objects of a segmented video allocation, and the segmented allocation smart contract control structure which manage and authorize specific actions and verify authenticity and permissions encoded within the token, which means that the token and cryptographic key function as an integrated authorization gatekeeper which manages govern the*

	release and modification of metadata-controlling the visibility and update permissions.
1.3 a secure resolution service, wherein the service dereferences remote media data upon successful validation of the cryptographic video token; further comprising logging each individual media dereference event to an auditable transaction layer.	**Para [0037]** …The metadata I/O processor 110 can identify one or more characteristics of a metadata object. For example, **the metadata I/O processor 110 can obtain and identify metadata objects of a segmented allocation including video, audio, text, any media,** executable programs, or any combination thereof… **Para [0055]** Still referring to FIG. 1 , in some implementations, the interface controller 120 can establish a data channel between a source address and a destination address, such that receivals or transmissions of a token or token distributions occurs between the addresses on a ledger (e.g., blockchain storage 158) and/or a digital wallet (e.g., wallet system 105). **An address can be generated based on executing, by the cryptographic key processor 114, a math-based function (e.g., hash, symmetric encryption, asymmetric encryption) on a public key of a public and private key pair (or a verification key of a verification and signing key pair)**… **Para [0093]** Sill referring to FIG. 3 , the container smart contract control structure 310 can execute and manage the functions related to both the container tokens 320 and the asset tokens 330. The container smart contract control structure 310 can enforce the conditions under which data from the container metadata object 326 and the asset metadata objects 334 can be accessed and modified. **The container smart contract control structure 310 can restricts outputs of the container metadata object 326 and asset metadata objects 334 by coded rules within the smart contract code of container smart contract control structure 310 that govern the release and modification of metadata-controlling the visibility and update permissions.** **Para [0100]** For example, when the segmented allocation smart contract control structure 350 needs to re-segment asset tokens 330 based on updated risk profiles, the segmented allocation smart contract control structure 350 can initiate a request to access updated credit metadata from the asset metadata objects 334, which are managed by the container smart contract control structure 310**. The request could be structured as a function call, "getAssetMetadata(assetId)", embedded within the segmented allocation smart contract control structure's 350 code. Upon receiving the request, the container smart contract control structure 310 can analyze it through a function "validateAccess(requestorId, assetId)". This function can check if the requesting contract (identified by requestorId) has the permissions to access the metadata of the specified asset token (identified by assetId).** If the validation passes, the requested metadata is provided, allowing the segmented allocation smart contract control structure 350 to proceed with the re-segmentation of the asset token 330 based on its current credit profile, thereby dynamically adjusting the allocation to reflect changes in the underlying asset conditions. **Para [0145]** Furthermore, at or after block 560, the processing circuits can in response to determining the allocation distribution, either (1) automatically initiate an on-chain exchange of the allocation distribution from an instrument owner's wallet address to a mobile wallet address of the client system, wherein the instrument owner's wallet address corresponds to an instrument owner with a security interest in one or more underlying physical assets of the first segmented allocation of the plurality of asset tokens or (2) automatically process an off-chain exchange from the instrument owner's wallet address to an account of a user operating the client system. In some implementations, automatically initiating can include generating a transaction on the blockchain. **That is, the blockchain transaction would transfer the designated amounts to the respective wallets. For example, this could include broadcasting the transaction to network nodes for verification and completion. In another example, details of the transaction such as**

	time-stamped logs could be appended for auditability… **Para [0150]** Additionally, the plurality of asset tokens can correspond to a controllable electronic record representing an ownership instrument of at least one security interest in at least one of the plurality of tokens. **That is, the electronic record can be updated to reflect ownership changes or rights adjustments. For example, tokenized shares of a company could be updated to reflect a transfer of ownership. Furthermore, these updates can be made in real-time as transactions occur. Moreover, each update could be logged to ensure a traceable history of ownership changes.** In some implementations, providing, by the segmented allocation control structure to the client system, the allocation token can be responsive to receiving a first allocation request including a first amount in exchange for a portion of an allocation distribution of the first segmented allocation… **Note:** *The mapped citation describes the cryptographic keys and segmented allocation smart contract control structure for the successful validation of the access. Also each tokenization updates are made and could be logged to ensure a traceable history of ownership which makes the process detailed about the transaction such as time-stamped logs could be appended for auditability.*
1.3.1 wherein the secure resolution service is restricted to operating within a hardware-attested trusted execution environment.	**Para [0050] The client system 103 (sometimes referred to herein as a "computing system") may be a mobile computing device, desktop computer, smartphone, tablet, smart watch, smart sensor, or any other device configured to facilitate receiving, displaying, and interacting with content (e.g., web pages, mobile applications, such as decentralized application (dApp), etc.).** Client system 103 may also include an interface controller 104 for communicating data over network 101 to data processing system 102 and third-party devices 106. In some implementations, each client system 103 can have a digital wallet address or exchanging (e.g., receiving or sending) fungible or non-fungible values (e.g., cryptocurrency, digital currency, stocks, bonds, loan, deed, etc.). **Para [0111]** In some implementations, the control structure processor 420 can generate a container token 320 including a link with the container metadata object 326. **The link can be established via a digital signature or cryptographic hash that securely associates the container token 320 with its metadata. The container metadata object 326 can be provided to a metadata interface 370 such that a blockchain can verify and store the metadata securely on the chain.** Additionally, the control structure processor 420 can encapsulate a container token 320 and a plurality of asset tokens 330 within the container smart contract control structure 310**. Encapsulating can include encrypting the data and setting permissions for data access. That is, the encapsulation can restrict outputs of the container metadata object and the plurality of asset metadata objects**… **Note:** *The mapped citation describes the system being secured and is operating on the client device such as mobile computing device, desktop computer, smartphone, etc. The mapped citation mentions security mechanisms (encryption, signatures, and restricted access). But it does not explicitly mention hardware attestation.*
1.3.2 wherein the logged dereference events trigger a commercial action selected from a group consisting of billing, royalty allocation, and advertising insertion.	**N/A**
1.4 a playback environment, wherein the playback	**Para [0032]** Moreover, aspects of the present disclosure address problems in the speed and resource **requirement/allocation associated with verifying and**

environment stores multiple tokens and dynamically assembling video streams in real-time from authorized references.	**processing token exchanges, allocation segmentations, and distributions.** Additionally, aspects of the present disclosure address problems in the **issuance of dynamic exchange and distribution instruments that includes internal states that dynamically change in real-time or near real-time.** In some implementations, the **systems and methods described herein can issue and dynamically update segmented allocations and tokens.** That is, since an asset's value or other financial parameters (represented and/or linked to a token in metadata) or use can fluctuate often**, the systems and methods described herein provide improvements over current token exchange and distribution instruments by providing a physical and digital distribution and exchange instrument that can monitor and provide (e.g., present on the physical or digital exchange instrument) real-time information about the internal states of the tokens and segmented allocations can be update in real-time or near real-time to protect the issuer of the token…** **Para [0037]** …The metadata I/O processor 110 can identify one or more characteristics of a metadata object. For example, **the metadata I/O processor 110 can obtain and identify metadata objects of a segmented allocation including video, audio, text, any media,** executable programs, or any combination thereof… **Para [0088] Each asset token 330 can include a link 332 to an asset metadata object 334, which stores detailed information about the underlying asset, such as its financial attributes, ownership history, and performance data.** The asset token 330 can be a controllable electronic record. **Link 332 can include a reference, pointer, or the like, to or between an asset token 330 and an asset metadata object 334.** Link 332 can be used to ensure that every asset token 330 has accessible, up-to-date information on its characteristics. For example, potential investors can examine the metadata to assess risk and potential returns before acquiring an asset token 330**. Asset tokens 330 can each include a particular fungible or non-fungible token and can correspond to particular asset metadata objects 334. An asset token 330 can be associated with a particular asset metadata object, and can be required to transmit output of the metadata object, transfer the metadata object to another storage location, or any combination thereof,** for example. Each of the asset token 330 can indicate control of a particular metadata object of the asset metadata objects 334 by a corresponding metadata link of the metadata links 332. The asset metadata objects 224 can each include a particular data or instructions. Metadata objects can correspond to a collections of executable instructions or data that can be finite. **For example, a metadata object can include a video file corresponding to a limited number of instances of video metadata…** **Para [0048] The client system 103 can include a computing system located remotely from the data processing system 102. The client system 103 can include a wallet system 105.** The wallet system 105 can include an interface to execute instructions corresponding to a particular wallet account, and to modify the structure or contents of a particular smart contract corresponding to a wallet account. **For example, the mobile wallet system 105 can include a user interface to receive allocation tokens and input that indicates selections of various tokens, transactions, accounts, devices, users, or systems. For example, the user interface can include a graphical user interface (GUI) that can be presented at a display device. The display device can display at least one or more user interface presentations, and can include an electronic display**… **Note:** *The mapped citation describes client system includes mobile wallet system which is a user interface to receive allocation tokens and input that indicates selections of various tokens, transactions, accounts, devices, users, or systems (i.e., playback device where the dynamic output is reflected) from the authorized references.*

1.4.1 wherein the playback environment is configured to manage independent tokens from multiple content owners within a single session	**N/A**
1.4.2 wherein the playback environment may function as a content wallet capable of managing multiple tokens from independent content owners.	**Para [0009]** In some implementations, the one or more processing circuits is further configured to in response to determining the allocation distribution, either automatically initiate an on-chain exchange of the allocation distribution from an instrument owner's wallet address to a mobile wallet address of the client system, **wherein the instrument owner's wallet address corresponds to an instrument owner with a security interest in one or more underlying physical assets of the first segmented allocation of the plurality of asset tokens, or automatically process an off-chain exchange from the instrument owner's wallet address to an account of a user operating the client system**. **Para [0048] The client system 103 can include a computing system located remotely from the data processing system 102. The client system 103 can include a wallet system 105.** The wallet system 105 can include an interface to execute instructions corresponding to a particular wallet account, and to modify the structure or contents of a particular smart contract corresponding to a wallet account. **For example, the mobile wallet system 105 can include a user interface to receive allocation tokens and input that indicates selections of various tokens, transactions, accounts, devices, users, or systems. For example, the user interface can include a graphical user interface (GUI) that can be presented at a display device. The display device can display at least one or more user interface presentations, and can include an electronic display**… **[Para 0077]** Additionally, data protection system 102 can also include mechanisms for **issuing allocation tokens 382 through a token interface 380 to a client system 103, which can be stored in a wallet system 105**. **Note:** *The mapped citation describes which is capable of managing various tokens and input from individual content owners.*
1.4.3 wherein the system is configured to terminate the video stream instantly upon the revocation of user consent parameters associated with the token.	**[Para 0060]** In some implementations, **when the token (or tokens) are transferred between computing systems or device**, **the sender's ledger or wallet (e.g., in wallet system 105) may be voided since the public-private key pair would be invalid** (**e.g., cannot be used to sign or verify an exchange**). **[Para 0139]** The processing circuits executing the segmented **allocation control structure can execute smart contract functions such as validateTokenOwnership( ) which can check and confirm that the request for dividends comes from a valid token holder**.

Patent Citation 2: US20170195697A1

Title	Media distribution with sample variants for normalized encryption
Priority Date	15 DEC 2015
Filing Date	14 DEC 2016
Publication Date	06 JUL 2017
Inventors	Raj Nair; Prabhudev Navali; Mikhail Mikhailov; David Alexander; Pablo Argon
Assignees	Telefonaktiebolaget LM Ericsson AB
IPC Classes	G06F21/60; H04L29/06; H04N21/2347
CPC Classes	G06F21/602; H04L63/0428; H04L65/65; H04L67/02; H04L67/568; H04N21/23439; H04N21/2347; H04N21/23476; H04N21/2351; H04N21/2362; H04N21/2381
US Classes	None
Family Members	US10306308B2 | US10237589B2 | US10158894B2 | WO2017103856A1 | CN108702527A | EP3391653B1 | WO2017103854A1 | US10771843B2

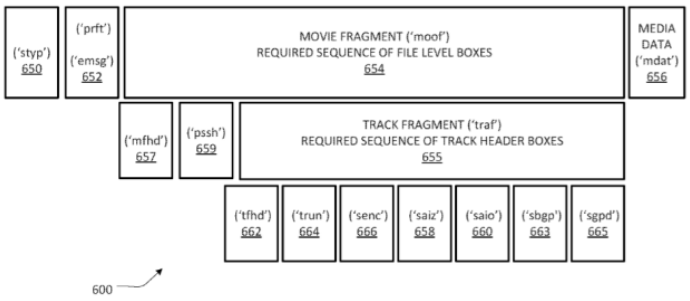
Abstract:

A media distribution system and method with sample variants for normalized encryption involves encrypting a main track of a media content asset using a first encryption scheme and encrypting a sample variant track of the media content asset using a second encryption scheme, and performing at least one of: storing the encrypted main track and encrypted sample variant track of the media content asset packaged in a storage format, and transmitting the encrypted main track and the encrypted sample variant track in a distribution container format to an edge media router (EMR) device configured to repackage the media content asset into a delivery container format without reencrypting the media content asset.

Key Features	Relevant Excerpts
1. A system for programmable video assembly, comprising:	**[Abstract]** A media distribution system and method with sample variants for normalized **encryption involves encrypting a main track of a media content** asset using a first encryption scheme and encrypting a sample variant track of the media content asset using a second encryption scheme, and performing at least one of: storing the encrypted main track and encrypted sample variant track of the media content asset packaged in a storage format, and **transmitting the encrypted main track and the encrypted sample variant track in a distribution container** format to an edge media router (EMR) device configured to repackage the media content asset into a delivery container format without reencrypting the media content asset. **Para [0046]** Additionally, there may be an implementation where **a video pipe delivering services to a premises is operative to deliver content to one or more progressive download clients of the premises** that are designed to receive the video in bursts in a file-based mechanism. In one embodiments, UE devices (like STB, IP-STB, for example) that are consuming unicast or multicast streams may perform adaptive bitrate streaming. **[Para 0012]** In a still further aspect, embodiments of **a system and method for processing main tracks and sample variant tracks of media content assets are disclosed. Encrypted media content having an encrypted main track and one or more sample variant tracks may be received**, for example in at least one of **an ISO Base Media File Format (ISOBMFF) container format,** an ISOBMFF carried in a

	Packetized Elementary Stream (PES) payload of an MPEG-TS elementary stream, an MPEG-TS elementary stream container format, and an MPEG-TS elementary stream with media content asset sample variants track data and track metadata in the PES payload, having one or more encryption schemes. **[Para 0012]** In a still further arrangement, the decrypted variant media data sample track may be provided to **a media player associated with the UE device for rendering the media content asset**. In a still further arrangement, **a sample variant track extractor can be used to construct the sample variant media track from the main track and sample variant track metadata and media data.** **[Para 0132] The content publisher (or its authorized agent) may interface with one or more suitable DRM entities to manage the release of KIDs/keys such that the playback path (for the actual sample data) is controlled and the player can only decrypt and render the data that it has been authorized to render.** **Note:** *The mapped citation describes a media distribution system that encrypts both main and sample variant tracks using different schemes, packages them in storage or distribution formats. It further explains delivery mechanisms such as progressive download, adaptive bitrate streaming, and handling of multiple container formats (ISOBMFF, MPEG-TS). A sample variant track extractor can dynamically construct variant tracks from metadata and media data, and DRM entities manage release of keys/KIDs to ensure authorized playback.*
1.1 a virtualization engine, wherein the engine is configured to convert a rendered video file into a reference container by extracting metadata and removing media sample data;	**[Para 0005]** The present patent disclosure is broadly directed to systems, methods, apparatuses, as well as network nodes and associated non-transitory computer-readable media for facilitating distribution of media content in a network architecture (e.g., involving managed and/or unmanaged networks) using a common intermediary **mezzanine distribution format (CMZF),** wherein the content is protected all the way from an encoding entity to the end user device (i.e., end-to-end content protection). In one example embodiment, a media content asset may be processed for packaging, at a headend facility, in a **CMZF container structure, also referred to as CMZF stream carrying one or more CMZF stream scheme elementary streams, that is configured to carry each bitrate representation of the media content asset encrypted in one or more encryption schemes.** The CMZF-formatted media content may be provided to an origin server for file-based distribution over an unmanaged/managed network and/or to a streaming network node for stream-based distribution over an unmanaged/managed network. **[Para 0006]** In another aspect, an embodiment of a system or apparatus configured as a network element is provided for **facilitating CMZF containerization of media content**. **[Para 0006]** Preferably, one or more extra ES definitions may be configured to define additional PES streams, each having a separate Packet Identifier (PID), for carriage of at least one of the ISOBMFF track and **track metadata data objects in the PES payload** and/or the sample variants track data and track metadata objects in the PES payload. **[Para 0008]** In a further variation, **the program instructions of an EMR component may further comprise instructions for repackaging or transformatting the media content asset into an output format for facilitating local storage of the media content asset while retaining the encryption scheme(s) performed at the headend node**. Example delivery formats output by an EMR component may include, but not limited to: HTTP Live Streaming (HLS), HTTP Dynamic Streaming (HDS), Dynamic Adaptive Streaming over HTTP (DASH), HTTP Smooth Streaming (HSS), Common Media File Format (CMAF), ISOBMFF, as well as MPEG-TS, Real-time Transport Protocol (RTP)-encapsulated MPEG-TS, RTP/MPEG-TS over ISOBMFF, and RTP/MPEG-

	TS with Encoder Boundary Point (EBP) or **virtual segmentation information, e.g., as referenced** in [http://www.ibc.org/_media/PDF/alex_giladi_passing_the_tuning_test_providing_cableequivalent_adsupported_linear_progra1.pdf], incorporated by reference herein. **[Para 0054]** To facilitate the foregoing media distribution architecture, **a CMZF container** format is advantageously provided, which in an example embodiment **comprises a packaging structure based on MPEG-TS container format extended according to the teachings** herein, where carriage of both TS and additional streams having new stream definitions for purposes of the present invention may be effectuated. In one implementation, **MPEG-TS extensions may be configured to carry ISOBMFF track data and track metadata data objects (referred to as "boxes" in the ISO/IEC 14496-12 standard) with new stream IDs, stream types, descriptors, etc**. **[Para 0070]** Because CMZF elementary streams are based on the ATS-compliant MPEG2-TS elementary streams, they may be configured to carry additional signaling **metadata for providing information about segment boundaries such as EBP and virtual segmentation metadata** as noted above. The signaling of a stream that is comprised of EBP or virtual segmentation may also be indicated within the PMT structure. **[Para 0055]** As set forth in ISO 13818-1, incorporated by reference herein, Transport Streams may be logically constructed from PES packets, which may comprise one or more programs, each described according to a Program Map Table (PMT) that may be provided as part of a metadata structure, **Program-Specific Information (PSI) table**, contained in TS payload. **PSI is typically carried in the form of a table structure and provides metadata about a program (i.e., a media content channel).** Each PSI table structure may be segmented into sections and can span multiple TS packets. Adaption Fields (AF) may also be provided in TS packets carrying PSI data. In general, the PSI data is not scrambled so that a receiving decoder can easily identify the properties of a stream for processing. **[Para 0056] The PSI data defined by ISO 13818-1 includes four tables:** Program Association Table (PAT); Conditional Access Table (CAT); Network Information Table (NIT), as well as the PMT structure noted above. PAT lists all programs carried in a TS, each of the listed programs having a program number. Each listed program has **a unique identifier (Program Identifier or PID)** in a corresponding PMT that contains information about the program. **There may be multiple PMT sections in a stream; each section is given a unique user-defined PID and maps a program number to the metadata describing that program and the streams within it.** The streams themselves may be contained in PES packets with PIDs specified in the PMT. Each program element descriptor in a transport stream table may be defined by an 8-bit descriptor tag. **[Para 0134]** Several aspects related to using suitable Variant Constructors, Variant Samples and Variant Byte Ranges in an example SVNE implementation are set forth in additional detail in the following sections. In the context of the present patent application, **a Variant Constructor defines which bytes are used to assemble a Sample Variant. According to an example SVNE use case implementation, there may be only one Variant Constructor defined for a given ISOBMFF sample**. **[Para 0139]** With respect to ISO storage, **Sample Variant data may be stored in one or more ISOBMFF metadata tracks (variant tracks**) according an example **SVNE implementation. An ISOBMFF video media track (media track) may be associated with a variant track** as defined further below. When an association is established between a media track and a variant track, **Sample Variant processing may be executed whenever a decoder does not have access to the KID/key defined for a**

	sample in the media track. **Note:** *The mapped citation describes CMZF container format and SVNE processing which act as functional equivalents of a virtualization layer. CMZF container format extends MPEG-TS to carry ISOBMFF track data and metadata. PSI tables (PAT, PMT, CAT, NIT) define metadata structures. Variant track processing occurs when decoder lacks access to the key for a sample, meaning playback relies on metadata/variants rather than original media samples.*
1.1.1 wherein the virtual video container retains ISO Base Media File Format box structures while being devoid of media sample data.	**[Para 0011]** In one variation, **the distribution container format may comprise at least one of an ISO Base Media File Format (ISOBMFF) container format**, an ISOBMFF carried in a Packetized Elementary Stream (PES) payload of an MPEG-TS elementary stream, an MPEG-TS elementary stream container format, and an MPEG-TS elementary stream with media content asset sample variants track data and track metadata in the PES payload. In yet another variation, **the distribution container format may be adapted to carry the media content asset encrypted in one or more encryption schemes** using a valid CMZF stream scheme, e.g., in a CMZF container format. **[Para 0023] FIG. 6 depicts an example media object compliant with ISO Base Media File Format (ISOBMFF)** or Common Media Application Format (CMAF) for carriage in a CMZF container structure in accordance with an embodiment of the present invention; **[Para 0058]** In order to **identify the specifications to which a file based on ISOBMFF complies, brands are used as identifiers in the file format**. They are set in a box named **File Type Box ("ftyp"),** which must be placed in the beginning of the file. A file that supports streaming includes information about the data units to stream (e.g., how to serve the elementary stream data in the file over streaming protocols). **This information is placed in additional tracks of the file called "hint" tracks. Separate "hint" tracks for different protocols may be included within the same file. Additional boxes relating to streaming include "moov" box, "mdat" box, "moof" box, etc., which will be further described in reference to example CMZF stream types below**.  FIG. 6
1.1.2 wherein the step of resolving referenced media data involves issuing MDAT range requests to the remote media data sources.	**[Para 0058]** A file that supports streaming includes information about the data units to stream (e.g., how to serve the elementary stream data in the file over streaming protocols). This information is placed in additional tracks of the file called "hint" tracks. Separate "hint" tracks for different protocols may be included within the same file. **Additional boxes relating to streaming include** "moov" box, "**mdat" box,** "moof" box, etc., which will be further described in reference to example CMZF stream types below. **[Para 0094]** Tracks may begin with a File Header and the samples are stored in Segments that each contain a single Track Fragment referencing a complete sample

	sequence stored in **a Media Data Box ("mdata"),** which immediately follows each Movie Fragment Box in delivery/storage order. **[Para 0135]** Further, **each Variant Constructor may be configured to define a sequence of one or more Variant Byte Ranges.** Each Variant Byte Range defines the location of a sequence of bytes that might constitute bytes in a Sample Variant. In an example SVNE use case implementation, **Variant Byte Ranges can contain only data used as part of the sample. I**n SVNE use case, the sequence of Variant Byte Ranges defined in a Variant Constructor may be grouped into only one Variant Byte Range group.
1.1.3 further comprising an artificial intelligence orchestration layer configured to modify the sequencing instructions of the reference container without accessing raw media samples.	**N/A**
1.2 a token issuance service, wherein the service configured to generate a cryptographicvideo token binding transaction, consent, and licensing information;	**[Para 0056]** The PSI data defined by ISO 13818-1 includes four tables: Program Association Table (PAT); Conditional Access Table (CAT); Network Information Table (NIT), as well as the PMT structure noted above. PAT lists all programs carried in a TS, each of the listed programs having a program number. Each listed program has **a unique identifier (Program Identifier or PID)** in a corresponding PMT that contains information about the program. **There may be multiple PMT sections in a stream; each section is given a unique user-defined PID and maps a program number to the metadata describing that program and the streams within it.** The streams themselves may be contained in PES packets with PIDs specified in the PMT. Each program element descriptor in a transport stream table may be defined by an 8-bit descriptor tag. **[Para 0132]** Consistent with the [SMPLVAR] specification, control/selection of encryption schemes may be located with a content publisher in the foregoing SVNE processing framework. Accordingly, **a content publisher (or an authorized headend entity or agent) may be configured to encode, encrypt (using multi-encryption), and compress media sample variants into the ISOBMFF file and provide that each set of Sample Variant data for a given sample time may be encrypted with same/different key and signaled with a same/different KID, wherein the Sample Variant data for different variants of sample block is encrypted with a different encryption scheme. The content publisher (or its authorized agent) may interface with one or more suitable DRM entities to manage the release of KIDs/keys such that the playback path (for the actual sample data) is controlled and the player can only decrypt and render the data that it has been authorized to render.** **[Para 0077]** With respect to carriage of **encrypted CMAF/ISOBMFF fragments over MPEG2-TS streams,** an example embodiment may include encryption of CMZF CMAF/ISOBMFF fragments using any of the **encryption schemes defined in the *CENC+ specification, e.g., four example available schemes being "cenc", "cbc1", "cbcs" and "cens"**. The bitstreams may be encrypted either with full sample or subsample pattern based (partial) encryption schemes, wherein the encryption signaling may be according to the [CENC] specification. The [CENC]-specific boxes may be carried in the same elementary stream along with the other MOOF related boxes. **The PSSH (Protection System Specific Header) data, if present, may be presented in the CMAF Header box (in the MOOV related boxes).** In one arrangement, carriage of CMAF/ISOBMFF initialization segment data may be carried in a separate elementary stream, the presence of the CMAF/ISBMFF initialization

	segment elementary stream being signaled in a modified PMT with a predefined stream type. **[Para 0087]** in the case of ADTS frames, raw_data_bytes may be encrypted, while adts_fixed_header, adts_variable_header, adts_error_check, adts_header_error_check, and adts_raw_data_block_error_check are not encrypted. Video ECM Elementary Streams may be set to CMZF-TS-ECM-ES type, which much be included if a Video main elementary stream is present, wherein **CETS ECM messages provide encryption signaling parameters.** Audio ECM Elementary Streams may also be set to CMZF-TS-ECM-ES type, which must be included if an Audio elementary stream is present, **wherein CETS ECM messages provide encryption signaling parameters with respect to the audio streams**. **Note:** *The mapped citation describes handling of encrypted media streams where content publishers or DRM entities manage the release of keys and KIDs/Tokens, ensuring playback only occurs when authorized. It describes encryption schemes (cenc, cbcs, cens, etc.), signaling via PSSH headers, and ECM streams that provide encryption parameters for video and audio.However, information regarding transaction binding, consent, and licensing metadata is not disclosed.*
1.2.1 wherein the token issuance service includes a unique session nonce in each generated token to prevent unauthorized resolution reuse.	**[Para 0132]** Consistent with the [SMPLVAR] specification, control/selection of **encryption schemes may be located with a content publisher in the foregoing SVNE processing framewor**k. Accordingly, a content publisher (or an authorized headend entity or agent) may be configured to encode, encrypt (using multi-encryption), and compress media sample variants into the ISOBMFF file and provide that **each set of Sample Variant data for a given sample time may be encrypted with same/different key and signaled with a same/different KID**, wherein **the Sample Variant data for different variants of sample block is encrypted with a different encryption scheme**. The content publisher (or its authorized agent) may interface with one or more suitable DRM entities to manage the release of KIDs/keys such that the playback path (for the actual sample data) is controlled and the player can only decrypt and render the data that it has been authorized to render. **[Para 0105] The CENS CMAF/ISOBMFF track will have all the corresponding track and encryption metadata boxes in it**. The CENS MOOF fragments may be encapsulated in the TS PES packets, which carry the same corresponding presentation and decoding **timestamp values (PTS/DTS values) from the input stream.** The main video samples and the sample variants samples are provided as time parallel samples. **The generated output stream will be CMAF over TS with CENS scheme**, which can be unicast and/or multicast and may be consumed by the downstream STB/Reach device/EMR. **[Para 0129]** In one example SVNE use case, two tracks of a media content asset may be provided, with one main track that carries samples with one (i.e., first) encryption scheme and another sample variants track that carries samples in a different (i.e., second) encryption scheme. In an illustrative arrangement, **for every sample in the main track, there may be provided an associated sample variant in the sample variant track, wherein the sample variants may have the same KID(s) as the main track samples. In another arrangement, the sample variants may be provided with KID(s) different than the main track's KID.** **[Para 0130]** As illustrated with respect to media sample block 1204-1, **main sample 1206-1 is encrypted using "cbcs" scheme with KID(1) while corresponding variant sample 1206-2 is encrypted using "cens" scheme with KID(2)**, although the same KIDs may also be used in some arrangements. **Note:** *The mapped citation describes publishers manage encryption schemes, KIDs, and DRM integration for CMAF/ISOBMFF streams, ensuring controlled playback*

	authorization. There are different encryption schemes for samples data. Sample Variant data for given sample time may be encrypted with same/different key (inferred as unique sessions keys).
1.2.2 wherein the transaction metadata includes instructions for micro payments triggered by individual segment resolution events	**N/A**
1.2.3 wherein the cryptographic video token may function as an integrated authorization gatekeeper that operates independently of the media payload by excluding media sample data and direct file paths, enforces the right to resolve by providing the necessary cryptographic permissions and binds governance logic to the resolution event.	**[Para 0012]** An example method may further comprise, inter alia, **determining that a decoder does not have access to the main track of a media content asset received at the decoder,** e.g., **responsive to a content request generated from a UE device. Responsive to the determination, an unencrypted Variant Constructor is obtained, which is signaled in a distribution container format as sample variant metadata that defines how to assemble an individual sample variant with respect to the media content asset**. Each Variant Byte Range in a sequence of Variant Byte Ranges defined in the unencrypted Variant Constructor is processed to assemble a variant media data sample track. **The assembled variant media data sample track is then decrypted using a media key defined in a metadata structure associated with the unencrypted Variant Constructor**. In one arrangement, the main track of the media content asset and the variant media data sample track are encrypted at a headend node using, respectively, a **first encryption scheme and a second encryption scheme** having respective **Key Identifiers (KIDs).** **[Para 0123] storing the encrypted main track and encrypted sample variant track of the media content asset packaged in a storage format**, e.g., any of the container formats set forth in the present patent application, and/or transmitting the encrypted main track and the encrypted sample variant track in a **distribution container format to a downstream node (e.g., an EMR device)** configured to repackage the media content asset into a delivery container format without reencrypting the media content asset, the delivery container format comprising a format compatible for processing by at least one of a premises gateway node, a set-top-box (STB), and a user equipment (UE) device. **[Para 0014]** As will be seen in further detail below, multiple encyption schemes may be carried in such a way that **only subsample partially encrypted data is transported in sample variant streams**, thereby providing the advantage of multi-encryption carriage with little or negligible overhead. **[Para 0132]** Consistent with the [SMPLVAR] specification, control/selection of encryption schemes may be located with a content publisher in the foregoing SVNE processing framework. Accordingly, **a content publisher (or an authorized headend entity or agent) may be configured to encode, encrypt (using multi-encryption), and compress media sample variants into the ISOBMFF file and provide that each set of Sample Variant data for a given sample time may be encrypted with same/different key and signaled with a same/different KID, wherein the Sample Variant data for different variants of sample block is encrypted with a different encryption scheme. The content publisher (or its authorized agent) may interface with one or more suitable DRM entities to manage the release of KIDs/keys such that the playback path (for the actual sample data) is controlled and the player can only decrypt and render the data that it has been authorized to render.** **[Para 0135]** Further, each Variant Constructor may be configured to define a **sequence of one or more Variant Byte Ranges. Each Variant Byte Range defines the location of a sequence of bytes that might constitute bytes in a Sample Variant**. In

	an example SVNE use case implementation, Variant Byte Ranges can contain only data used as part of the sample **Note:** *The mapped citation describes that unencrypted Variant Constructor and KID metadata acts as integrated authorization gatekeeper. The unencrypted constructor contains no media sample data (i.e., operates independently of the media payload). Also described Variant Byte Ranges (internal references inside the container/stream). KIDs/media keys needed to decrypt the assembled variant. The encryption schemes and track references bind governance logic to the resolution event.*
1.3 a secure resolution service, wherein the service dereferences remote media data upon successful validation of the cryptographic video token; further comprising logging each individual media dereference event to an auditable transaction layer.	**[Para 0012] The assembled variant media data sample track is then decrypted using a media key defined in a metadata structure associated with the unencrypted Variant Constructor.** In one arrangement, the main track of the media content asset and the variant media data sample track are encrypted at a headend node using, respectively, a first encryption scheme and a second encryption scheme having respective Key Identifiers (KIDs). In a still further arrangement, the decrypted variant media data sample track may be provided to **a media player associated with the UE device for rendering the media content asset.** In a still further arrangement, a sample variant track extractor can be used to construct the sample variant media track from the main track and sample variant track metadata and media data. **[Para 0131] Processed media content samples obtained at a receiver decoder (e.g., downstream EMR and/or UE device) may comprise samples output** 1208 of a sample variant processing module (not specifically shown in this FIG.). Depending on the encryption scheme, the output may comprise either CENS samples 1212 or CBCS samples 1214. As illustrated **with respect to the input media sample block 1204-1, such output may therefore comprise CENS sample 1206-2 or CBCS sample 1206-1, with respective KID values**. **[Para 0132]** The content publisher (or its authorized agent) may interface with one or more suitable DRM entities to manage the release of KIDs/keys such that the playback path (for the actual sample data) is controlled and **the player can only decrypt and render the data that it has been authorized to render**. **[Para 0133] The selected encrypted output sample is provided to a decryptor module 1322 and associated decoder 1324**, which together may comprise a standard CENC decoder in an example implementation involving “cens” and “cbcs” schemes. **The decrypt/decode of the selected encrypted output sample is facilitated by appropriate key(s) received from DRM 1310, whereupon the decrypted/decoded sample may be presented to either a native and/or connected renderer/player** (not shown). **Note:** *The mapped citation describes that the playback environment securely assembles and decrypts variant media tracks using keys and Key Identifiers (KIDs), with DRM entities controlling the release of those keys so only authorized playback occurs. This demonstrates a secure resolution service where remote media data is dereferenced only after successful validation of cryptographic tokens/KIDs.*
1.3.1 wherein the secure resolution service is restricted to operating within a hardware-attested trusted execution environment.	**[Para 0040]** Accordingly, **such client devices may include legacy set-top boxes (STBs), Next Generation IP-based STBs**, **networked TVs, personal/digital video recorders (PVR/DVRs)**, networked media projectors, **portable laptops**, **netbooks**, **palm tops, tablets, smartphones, multimedia/video phones, mobile/wireless user equipment, portable media players, portable gaming systems or consoles (such as the Wii®, Play Station 3®, etc.) and the like**, **which may access or consume content/services provided via an end-to-end encrypted media distribution network** using a common intermediary distribution container format in accordance with to one or more embodiments set forth herein.

1.3.2 wherein the logged dereference events trigger a commercial action selected from a group consisting of billing, royalty allocation, and advertising insertion.	**N/A**
1.4 a playback environment, wherein the playback environment stores multiple tokens and dynamically assembling video streams in real-time from authorized references.	**[Para 0012] The assembled variant media data sample track is then decrypted using a media key defined in a metadata structure associated with the unencrypted Variant Constructor**. In one arrangement, the main track of the media content asset and the variant media data sample track are encrypted at a headend node using, respectively, a first encryption scheme and a second encryption scheme having respective Key Identifiers (KIDs). In a still further arrangement, **the decrypted variant media data sample track may be provided to a media player associated with the UE device for rendering the media content asset. In a still further arrangement, a sample variant track extractor can be used to construct the sample variant media track from the main track and sample variant track metadata and media data**. **[Para 0198] One or more media players 2014 may be provided for operating in conjunction with the other subsystems of the client device 2000 for facilitating user control over media playback, including channel change requests.** Example media players may be configured to operate with one or more A/V **coder/decoder (codec) functionalities** based on known or hereto unknown standards or specifications including but not limited to, e.g., Moving Pictures Expert Group (MPEG) codecs (MPEG, MPEG-2, MPEG-4, etc.), H.264 codec, High Efficiency Video Coding or HEVC (H.265) codec, and the like. **[Claim 18]** A method for effectuating **playback of encrypted media content** having an encrypted main track and one or more sample variant tracks received in at least one of an ISO Base Media File Format (ISOBMFF) container format, an ISOBMFF carried in a Packetized Elementary Stream (PES) payload of an MPEG-TS elementary stream, an MPEG-TS elementary stream container format, and **an MPEG-TS elementary stream with media content asset sample variants track data and track metadata in the PES payload, having one or more encryption schemes**, the method comprising:… **decrypting the assembled variant media data sample track using a media key defined in a metadata structure associated with the unencrypted Variant Constructor**; and **providing the decrypted variant media data sample track to a media player for rendering the media content asset**, wherein the main track of the media content asset and the variant media data sample track are encrypted at a headend node using, respectively, **the first encryption and second encryption schemes having respective Key Identifiers (KIDs).** **[Para 0014]** The apparatus as recited in claim 10, wherein **the media content asset comprises** at least one of live TV content, IPTV content, time-shifted (TS) TV content, place-shifted (PS) TV content, gaming content, and **Video on Demand (VOD) content,** ABR content, Virtual Reality (VR) content, and user equipment (UE) device metadata content. **[Para 0006]** In another aspect, an embodiment of a system or apparatus configured as a network element is provided for facilitating **CMZF containerization of media content.** The claimed embodiment comprises, inter alia, one or more processors and a plurality of network interfaces configured to receive media content assets from one or more content sources. An encoder is provided for generating a plurality of bitrate representations **for each media content asset, which may be segmented by a segmenter module**. **An encryptor is configured to encrypt bitrate representations**

	of a media content asset using one or more encryption schemes. **[Para 0098]** For example, EMR as part of a suitable workflow may be configured to **transformat the CMZF segments into ABR delivery format segments and stores them in its local cache 712 and/or at a remote cache** (not specifically shown), or may have a more involved workflow to upload to an origin server for end devices to access the segments for ABR playout. **Note:** *The mapped citation describes a playback environment that receives encrypted main and variant tracks protected by different encryption schemes and Key Identifiers (KIDs). These KIDs and media keys function like tokens, allowing the system to store and manage multiple access controls. A Variant Constructor dynamically assembles media sample tracks from byte ranges in real time, then decrypts them using authorized KID metadata before delivering them to the media player.*
1.4.1 wherein the playback environment is configured to manage independent tokens from multiple content owners within a single session	**[Para 0056] The PSI data defined by ISO 13818-1 includes four tables**: **Program Association Table (PAT)**; Conditional Access Table (CAT); Network Information Table (NIT), as well as **the PMT structure** noted above. PAT lists all programs carried in a TS, **each of the listed programs having a program number**. **Each listed program has a unique identifier (Program Identifier or PID) in a corresponding PMT that contains information about the program**. **There may be multiple PMT sections in a stream; each section is given a unique user-defined PID and maps a program number to the metadata describing that program and the streams within it.** The streams themselves may be contained in PES packets with PIDs specified in the PMT. Each program element descriptor in a transport stream table may be defined by an 8-bit descriptor tag. **[Para 0132] The content publisher (or its authorized agent) may interface with one or more suitable DRM entities to manage the release of KIDs/keys such that the playback path (for the actual sample data) is controlled and the player can only decrypt and render the data that it has been authorized to render** **Note:** *The mapped citation describes that Program Specific Information (PSI) tables, including the Program Map Table (PMT), assign each program in a transport stream a unique Program Identifier (PID). Multiple PMT sections can exist within one stream, each linked to its own program number and metadata. These PIDs and descriptors act as independent control elements, effectively functioning like tokens by uniquely identifying and regulating access to different content streams within a single session.*
1.4.2 wherein the playback environment may function as a content wallet capable of managing multiple tokens from independent content owners.	**[Para 0012] The assembled variant media data sample track is then decrypted using a media key defined in a metadata structure associated with the unencrypted Variant Constructor**. In one arrangement, the main track of the media content asset and the variant media data sample track are encrypted at a headend node using, respectively, a first encryption scheme and a second encryption scheme having respective Key Identifiers (KIDs). In a still further arrangement, **the decrypted variant media data sample track may be provided to a media player associated with the UE device for rendering the media content asset.** In a still further arrangement, a sample variant track extractor can be used to construct the sample variant media track from the main track and sample variant track metadata and media data. **[Para 0198] One or more media players 2014 may be provided for operating in conjunction with the other subsystems of the client device 2000 for facilitating user control over media playback, including channel change requests.** Example media players may be configured to operate with one or more A/V **coder/decoder (codec) functionalities** based on known or hereto unknown standards or specifications including but not limited to, e.g., Moving Pictures Expert Group (MPEG) codecs (MPEG, MPEG-2, MPEG-4, etc.), H.264 codec, High Efficiency Video Coding or HEVC

	(H.265) codec, and the like. **[Claim 18]** wherein the main track of the media content asset and the variant media data sample track are encrypted at a headend node using, respectively, **the first encryption and second encryption schemes having respective Key Identifiers (KIDs).** **[Para 0006]** In another aspect, an embodiment of a system or apparatus configured as a network element is provided for facilitating **CMZF containerization of media content.** The claimed embodiment comprises, inter alia, one or more processors and a plurality of network interfaces configured to receive media content assets from one or more content sources. An encoder is provided for generating a plurality of bitrate representations **for each media content asset, which may be segmented by a segmenter module**. **An encryptor is configured to encrypt bitrate representations of a media content asset using one or more encryption schemes.** One or more persistent memory modules are provided with program instructions stored thereon,… **[Para 0098]** For example, EMR as part of a suitable workflow may be configured to **transformat the CMZF segments into ABR delivery format segments and stores them in its local cache 712 and/or at a remote cache** (not specifically shown), or may have a more involved workflow to upload to an origin server for end devices to access the segments for ABR playout. **[Para 0056] The PSI data defined by ISO 13818-1 includes four tables**: **Program Association Table (PAT**); Conditional Access Table (CAT); Network Information Table (NIT), as well as **the PMT structure** noted above. PAT lists all programs carried in a TS, **each of the listed programs having a program number**. **Each listed program has a unique identifier (Program Identifier or PID) in a corresponding PMT that contains information about the program**. **There may be multiple PMT sections in a stream; each section is given a unique user-defined PID and maps a program number to the metadata describing that program and the streams within it.** **[Para 0132] The content publisher (or its authorized agent) may interface with one or more suitable DRM entities to manage the release of KIDs/keys such that the playback path (for the actual sample data) is controlled and the player can only decrypt and render the data that it has been authorized to render** **Note:** *The mapped citation describes that the playback environment can act as a content wallet. The environment manages multiple encryption keys and Key Identifiers (KIDs), which function as tokens tied to different tracks and encryption schemes. The environment handles multiple Program Identifiers (PIDs) and Program Map Table (PMT) sections within a single transport stream, which can be interpreted as independent tokens associated with different content owners.*
1.4.3 wherein the system is configured to terminate the video stream instantly upon the revocation of user consent parameters associated with the token.	**[Para 0132] The content publisher (or its authorized agent) may interface with one or more suitable DRM entities to manage the release of KIDs/keys such that the playback path (for the actual sample data) is controlled** and the player can only decrypt and **render the data that it has been authorized to render**. **Note:** *The mapped citation describes that the system authorizes the user based on the credentials. If the credentials are not matched the user wont have access. However, the playback is also stopped if the authentication is revoked is not explicitly disclosed in the prior art.*

Patent Citation 3: WO2025245645A1

Title	Object-level access control for multimedia content
Priority Date	31 MAY 2024
Filing Date	30 MAY 2025
Publication Date	04 DEC 2025
Inventors	Alexey Iskrov; Alexander Kurtynin; Alisa Gagina; Sav Sidorov; Igor Kolosiuk
Assignees	Breeze Labs Inc
IPC Classes	G06T5/60; H04L9/32; H04N21/2347
CPC Classes	G06T5/60; H04L9/0618; H04L9/0643; H04L9/0825; H04L9/0891; H04L9/50; H04N21/23418; H04N21/2541; H04N21/835; H04N21/8456; H04L2209/60
US Classes	None
Family Members	None

Abstract:

Systems and methods for managing access to media content involve generating a media container for object-based media access by: generating data streams on a basis of objects of interest in a multimedia file such that each data stream is a segment of the multimedia file that corresponds to a representation of a respective object of interest; generating an encryption key of each data stream, each encryption key corresponding to a respective data stream; encrypting the data streams with the corresponding encryption key; generating an identifier for each encryption key; and generating a media container file by packaging the data streams and the encryption key identifiers.

Key Features	Relevant Excerpts
1. A system for programmable video assembly, comprising:	**[Abstract] Systems and methods for managing access to media content** involve generating a media container for object-based media access by: generating data streams on a basis of objects of interest in a **multimedia file such that each data stream is a segment of the multimedia file** that corresponds to a representation of a respective object of interest; **generating an encryption key of each data stream,** each encryption key corresponding to a respective data stream **[Claim 38]** A method of managing access to **multimedia content,** the method comprising: **requesting access, at an access control server, of one or more of a plurality of encrypted data streams each corresponding to a segment of a multimedia file** and **a representation of a respective object of interest;** receiving **authorization to the one or more encrypted data streams from the access control server;** retrieving one or more encryption key identifiers from the one or more encrypted data streams. **Note:** *The mapped citation describes that system is a programmable video assembly in which media container for object-based media access data streams on a basis of objects of interest in a multimedia file such that each data stream is a segment of the multimedia file*
1.1 a virtualization engine, wherein the engine is configured to convert a rendered video file into a	**[Para 00102]** depicts generation of a **multimedia container file** for providing segmented access and verification of media content. A user 202 may upload a multimedia file 204 for processing by the systems and methods of the present disclosure. A **plurality of data streams may be generated from the multimedia file**

reference container by extracting metadata and removing media sample data;	**204, which may be viewed and managed during playback once the generation of the multimedia container file is complete... each data stream can be considered to be an object or data layer, which can be encoded and packaged into a media container or rendered for streaming**. **[Para 0271] The media engine 808** can also perform additional processes for data stream generation. For example, **the media engine 808 may synchronize the generated data streams**. **The media engine 808 may also include data with regard to each of the data streams as metadata**. The generated data streams can be encoded using a suitable method based on the type of data stream by the media engine 808 and forwarded to a file engine 810. **[Para 00161]** In accordance with the present disclosure, a **media container file 232 (e.g. a multimedia container file) is generated**. In particular, the encrypted data streams corresponding to the objects of interest, the encrypted **encryption keys,** the digital signatures, and the hash values 228 may be packaged into the **media container file** 232. The **media container file 232 may be useful in improving the portability, organization,** and secure distribution of the multimedia content while maintaining compatibility and data integrity. [**[Para 0017]** In some aspects, the method further comprises: embedding, in a header of the **media container, metadata** for synchronizing the at least one data stream, the **metadata comprising: frame number, decode timestamp, presentation timestamp, object identifier, spatial information, object timestamp, or combinations thereof; and** synchronizing the at least one data stream. **Note:** *The mapped citation describes the creation of a media container file from an uploaded multimedia file shows segmentation of the multimedia file into data streams packaged into a container that encrypts streams, keys, signatures, and hash values that are packaged into the container for secure distribution.*
1.1.1 wherein the virtual video container retains ISO Base Media File Format box structures while being devoid of media sample data.	**[Claim 25] 25.** The method of claim 1 , further comprising: encoding the at least one data stream, **wherein the at least one data stream is video-based, audio-based**, text-based, or combinations thereof; **wherein the video-based at least one data stream is encoded using H.264 or HEVC codec**; **wherein the audio-based at least one data stream is encoded using AAC or MP3 codec**; and wherein the at least one data stream comprises one or more mask images for separating the respective object of interest from background, and the one or more mask images are encoded using Run-Length Encoding (RLE), bit-plane encoding, PNG compression, GIF compression, H.264 codec, HEVC codec, or combinations thereof. **[Para 00105] Source separation models may be used to identify distinct sounds (e.g. from the object of interest) and output estimates of the individual source sounds**. **Speaker diarization models can segment the audio or the audio stream 212 into speaker-homogeneous regions and label each segment with a speaker** (e.g. as an object of interest) identity. A data stream is generated for each of the selected objects of interest. **Note:** *The mapped citation describes separation of media file into separate video and audio format. Further the source separation module helps in identifying the distinct sound from individual source sounds. Therefore, we assume that once the data is being separated into audio and video file, the source separation module would delink it from the rendered data.*
1.1.2 wherein the step of resolving referenced media data involves issuing MDAT range requests to the	**[Para 0027]** In some aspects, the method further comprises: **in response to a request to access the encrypted time-based segment, decrypting the time-based segment of the one of the at least one data stream with a corresponding one of the at least one time-specific encryption key; and outputting the decrypted time-**

remote media data sources.	**based segment to a video display or audio speaker**. **Note:** *The mapped citation describes the data being requested in response to the encrypted data that is synchronized with the reference media. However, the MDAT range being assigned to the referenced data is not explicitly disclosed.*
1.1.3 further comprising an artificial intelligence orchestration layer configured to modify the sequencing instructions of the reference container without accessing raw media samples.	**[Para 0041]** In some aspects, **the de-identified representation is generated by modifying the respective object of interest using at least one artificial intelligence model to generate a representation of the respective object of interest lacking features permitting identification of the respective object of interest.** **[Para 0042] In some aspects, the at least one artificial intelligence model is configured to modify features in the respective according to a predetermined policy or threshold by performing feature identification and feature alteration.** **[Para 0079]** Correspondingly, the present disclosure relates **to novel systems and methods for granular access control, content authentication, and secure sharing of multimedia content**, as to help address the limitations of existing file-level protection mechanisms. **By employing deep learning for object segmentation and creating separate encrypted streams**, the present disclosure can enable selective access and privacy protection at the object level. **Note:** *The mapped citation describes the use of deep learning and AI model for modifying feature or object of interest.*
1.2 a token issuance service, wherein the service configured to generate a cryptographic video token binding transaction, consent, and licensing information;	**[Para 00211]** Referring now to FIG. 4B, a user may access media content, for example included in the media container file 412 through the use of an access control server 430. The access control sever 430 is coupled to the secure repository 406, implemented as a key management service (KMS). The access control sever 430 may have stored on it the media container file 412. The user can access or retrieve the media container file 412 from the access control server 430. **The access control sever 430 can also provide the user 402 with access credentials, for example authentication tokens, for accessing content (e.g., data streams) in the media container file 412. Each authentication credentials may correspond to a particular data stream. The access control sever 430 can grant the credentials based on default permissions/policies or those set by the owner of the media container file 412. The user 402 can also request access to content in the media container file 412 from the owner through the access control sever 430.** **[Para 00121] Note that each data stream, including each distinct data representation/ data stream for a particular object of interest can be independently manageable and associated with the primary object of interest. Each object of interest can be identified by an object identifier or ID.** As described herein, **each distinct representation of a particular object of interest (e.g., each of a plurality of data streams for an object of interest) can be encrypted with its own unique cryptographic key for granular, context- aware access control.** The generation of these multiple representations can occur concurrently with, or subsequent to, initial object identification and tracking. Specific representations generated for an object of interest may be determined by system configuration, content-owner policies, the object's nature, or real-time analysis of its context and sensitivity. **Note:** *The mapped citation describes that the access control server provide the authentication tokens for accessing content in media container file and each data stream is managed by an ID where each distinct representation is encrypted by a unique cryptographic key which corresponds to the token issuance service configured to generate a cryptographic token. However, a cryptographic token binding the transaction, consent and licensing information has not been disclosed.*

1.2.1 wherein the token issuance service includes a unique session nonce in each generated token to prevent unauthorized resolution reuse.	**[Para 00121] Note that each data stream, including each distinct data representation/ data stream for a particular object of interest can be independently manageable and associated with the primary object of interest. Each object of interest can be identified by an object identifier or ID.** As described herein, **each distinct representation of a particular object of interest (e.g., each of a plurality of data streams for an object of interest) can be encrypted with its own unique cryptographic key for granular, context- aware access control.** The generation of these multiple representations can occur concurrently with, or subsequent to, initial object identification and tracking. Specific representations generated for an object of interest may be determined by system configuration, content-owner policies, the object's nature, or real-time analysis of its context and sensitivity. **[Para 00154]** The encryption keys 230 may be stored in a secure repository 226, as depicted in FIG. 2A. The secure repository 226 may be managed by **the system of the present disclosure and can provide a protected and isolated environment for key management to reduce the risk of unauthorized access or key leakage.** An access control module or similar process may be implemented to interact with the secure repository 226 to manage and control the retrieval and distribution of the decryption keys 230 to authorized users based on user instructions and/or pre-set policies. This can ensure that only authorized individuals can access the content of the individual data streams. **Note:** *The mapped citation describes that each data stream (i.e. segment) is independently managed by an object identifier or an ID where each distinct representation can be encrypted with its own unique cryptographic key for access control which ultimately provide a protected environment for key management to prevent unauthorized access.*
1.2.2 wherein the transaction metadata includes instructions for micro payments triggered by individual segment resolution events	**N/A**
1.2.3 wherein the cryptographic video token may function as an integrated authorization gatekeeper that operates independently of the media payload by excluding media sample data and direct file paths, enforces the right to resolve by providing the necessary cryptographic permissions and binds governance logic to the resolution event.	**[Para 00121] Note that each data stream, including each distinct data representation/ data stream for a particular object of interest can be independently manageable and associated with the primary object of interest. Each object of interest can be identified by an object identifier or ID.** As described herein, **each distinct representation of a particular object of interest (e.g., each of a plurality of data streams for an object of interest) can be encrypted with its own unique cryptographic key for granular, context- aware access control.** The generation of these multiple representations can occur concurrently with, or subsequent to, initial object identification and tracking. Specific representations generated for an object of interest may be determined by system configuration, content-owner policies, the object's nature, or real-time analysis of its context and sensitivity. **Claim 34** A method for managing access to multimedia content, comprising: **receiving a request for access of multimedia content, the multimedia content comprising a plurality of data streams, each data stream of the plurality of data streams is a segment of the multimedia content** that comprises and corresponds to a respective representation of a respective object of interest in the multimedia content; authenticating the request for access, the request for accessing identifying one or more first data streams for access from the at least one data stream;

	retrieving one or more encryption key identifiers, each corresponding to a respective one of the first data streams; **retrieving one or more encryption keys corresponding to the one or more first data streams from a key management service in response to authentication by the key management service based on the authenticating of the request for access and the one or more encryption key identifiers; and decrypting the one or more first data streams using the one or more encryption keys in response to the request for access.** **Note:** *The mapped citation describes that the each data stream is managed by a distinct representation which is encrypted by its own cryptographic key and on retrieving the encryption key corresponding to data streams authentication from key management service takes place and upon that authentication or validation the content corresponding to the requested data stream is decrypted which can be inferred to the point that token may function as authorization gatekeeper.*
1.3 a secure resolution service, wherein the service dereferences remote media data upon successful validation of the cryptographic video token; further comprising logging each individual media dereference event to an auditable transaction layer.	**[Para 0068]** In accordance with another aspect of the present disclosure, a method of managing access to multimedia content is disclosed, the method comprising: requesting access, at an access control server, of one or more of **a plurality of encrypted data streams each corresponding to a segment of a multimedia file and a representation of a respective object of interest; receiving authorization to the one or more encrypted data streams from the access control server; retrieving one or more encryption key identifiers from the one or more encrypted data streams, each encryption key identifier corresponding to a respective encrypted data stream; requesting one or more encryption keys corresponding to the one or more encryption key identifiers from a key management server based on the authorization and using the one or more encryption keys; receiving the one or more encryption keys; and decrypting the one or more encrypted data streams using the one or more encryption keys.** **Note:** *The mapped citation describes that on receiving the authorization from content server to retrieve the encrypted keys and further receiving an authorization from key management server the decryption of requested encrypted data stream takes place. However, logging each individual media dereference event to an auditable transaction layer has not been disclosed.*
1.3.1 wherein the secure resolution service is restricted to operating within a hardware-attested trusted execution environment.	**[Para 00154]** The encryption keys 230 may be stored in a secure repository 226, as depicted in FIG. 2A. **The secure repository 226 may be managed by the system of the present disclosure and can provide a protected and isolated environment for key management to reduce the risk of unauthorized access or key leakage. An access control module or similar process may be implemented to interact with the secure repository 226 to manage and control the retrieval and distribution of the decryption keys 230 to authorized users based on user instructions and/or pre-set policies.** This can ensure that only authorized individuals can access the content of the individual data streams. In some embodiments, the secure repository 226 is implemented as a Key Management Service (KMS), which manages access to the decryption keys 230. In some aspects, the secure repository 226 is implemented as a secure key-value databases. These databases can be implemented with various access control mechanisms and encryption features and can offer a software-based option for secure storage. **In some aspects, the secure repository 226 is implemented as hardware security modules (HSMs). The HSMs can provide protection against physical and logical attacks and can be suitable for high-security environments.** **[00222] Note that where intermittent connectivity is required, decrypted encryption keys can be cached by the user 402 inside a Trusted Execution Environment (TEE) or similar secure storage. A cache lifetime for such keys can be set by the owner or KMS, At_max (e.g., default of 2 hours and configurable by**

	policy/license), which can be enforced by the TEE's secure clock or by the delivered license terms. After expiry, the user 402 is required to re-authenticate with the KMS and re-acquire or re-validate keys, thereby enforcing intervening revocation events. **Note:** *The mapped citation describes that the system provide a protected and isolated environment to reduce the risk of unauthorized access and provides the decryption keys to the authorized user only where the secure repository is implemented as hardware security modules which are suitable for high-security environments. The decryption keys are stored inside a trusted execution environment for providing secure storage and prevent the risk of unauthorized reuse.*
1.3.2 wherein the logged dereference events trigger a commercial action selected from a group consisting of billing, royalty allocation, and advertising insertion.	**N/A**
1.4 a playback environment, wherein the playback environment stores multiple tokens and dynamically assembling video streams in real-time from authorized references.	**[Para 0067] In some aspects, the plurality of encryption keys are stored in a key management service secure repository; and wherein each of the plurality of encryption keys is retrievable for decrypting the respective data stream from the key management service using a respective encryption key identifier and corresponding authentication data.** **[Para 00121] Note that each data stream, including each distinct data representation/ data stream for a particular object of interest can be independently manageable and associated with the primary object of interest. Each object of interest can be identified by an object identifier or ID.** As described herein, **each distinct representation of a particular object of interest (e.g., each of a plurality of data streams for an object of interest) can be encrypted with its own unique cryptographic key for granular, context- aware access control.** The generation of these multiple representations can occur concurrently with, or subsequent to, initial object identification and tracking. Specific representations generated for an object of interest may be determined by system configuration, content-owner policies, the object's nature, or real-time analysis of its context and sensitivity. **Note:** *The mapped citation describes that decrypting the requested data stream is on the basis of encryption keys which are stored in a key management service secure repository where the each data stream is managed by a distinct representation which are encrypted by cryptographic keys which simulates to the storing of multiple tokens.*
1.4.1 wherein the playback environment is configured to manage independent tokens from multiple content owners within a single session.	**[Para 00213] The key management service 406 can verify the request from the key representations and the credentials.** Policy checks may also be applied (e.g., role, geo-fence, time window, revocation list). **If access for the data streams is granted** (e.g., the credentials are approved), **the requested encryption keys stored in the key management service 406 and corresponding to the key representations and desired data streams is returned to the user 402**. **In some embodiments, the key management service 406 wraps the encryption keys (e.g., with AES Key-Wrap per RFC 3394) using a per-session secret derived from the secure channel handshake (e.g., via a TLS exporter).** **Note:** *The mapped citation describes that the key management system uses the encryption keys per session which means that the encrypted keys are expired after a session.*

1.4.2 wherein the playback environment may function as a content wallet capable of managing multiple tokens from independent content owners.	**[Para 00240]** At 507, **the user authenticates to a key management service (e.g., via the access control server) using their credentials** (e.g., issued by the access control server). **The key management service can verify the user request**, **which comprises the encryption key representations for the desired encryption keys, and verify the user based on authorization token, policies**, and encryption ID revocation status. Once authenticated, the key management service, **based on the user's identity, roles, permissions, and request context, can provide the user with the secure set of encryption keys for accessing the data streams** (or tokens/licenses allowing CEK derivation) at 508. ***Para 0065+ …encrypting each of the plurality of data streams with a corresponding encryption key of the plurality of encryption keys**, each encrypted data stream configured to be decrypted by the corresponding encryption key; **generating a plurality of encryption key identifiers**, **each being a unique identifier corresponding to a respective encryption key**; and generating a media container by packaging **the plurality of data streams and the plurality of encryption key identifiers**. **[Para 0097]** In accordance with the present disclosure, **a second user 126 may wish to access the media content of the multimedia file 122**. Accordingly, **the second user 126 may request access of media content** (e.g. access of one or more encrypted data streams 124a) from the first user 102 directly or through the servers 108. If the first user 102 would like to grant access of the media content to the second user 126, for example, by allowing the second user to access one or more encrypted data streams 124a, **the servers 108 can authenticate the first user 102 to retrieve the encryption keys (if previously stored) and decrypt the encryption keys (if previously encrypted) with their private key**. **Note:** *The mapped citation describes that the system can check for multiple users at a time which are requesting for playback request. The system in return creates the plurality of encrypted keys which can verify plurality of users. Therefore, we can deduce that the playback system acts a content wallet capable of monitoring multiple users content at same time.*
1.4.3 wherein the system is configured to terminate the video stream instantly upon the revocation of user consent parameters associated with the token.	**[Para 00220]** Encryption Policy: An object of interest's ciphertext can be encrypted under a CP-ABE policy defined by the content owner/system, for example setting the key to expire after a certain time (e.g., policy := object_uuid_X AND (user_consent_status_for_X = true) AND (current_time < key_expiry_timestamp)). **[Para 00220] Attribute Issuance: The KMS can act as an attribute authority, issuing cryptographic attributes to users, allowing the key to expire after a certain time (e.g., user_consent_status_for_X = true, key_expiry_timestamp = [timestamp]). The user 402 can use these attributes to derive the corresponding decryption key.** **[Para 00220] Revocation: The KMS can revoke or expire user attributes. For example, if access to data streams for an object of interest is withdrawn, the KMS can change the user's consent or access attribute to the object of interest to false. If the policy then evaluates to false based on the user's current attributes, the playback client cannot derive the decryption key, even with a local ciphertext copy.** **Note:** *The mapped citation describes that the key management system can expire or revoke the attributes. The key management system can change the user consent and makes access attribute to be false where the playback client cannot derive the decryption key which corresponds to the termination of the video stream upon revocation.*

Patent Citation 4: US11582206B2

Title	Device independent encrypted content access system
Priority Date	04 JAN 2017
Filing Date	12 FEB 2021
Publication Date	14 FEB 2023
Inventors	Jad S. Boutros; Jiayuan Ma; Filipe Jorge Marques de Almeida; Marcel M. Yung
Assignees	Snap Inc
IPC Classes	G06F21/62; H04W12/06; H04L9/40; H04M1/724; H04M1/72463; H04L9/08; H04L9/14; H04L9/32; H04W12/04;
CPC Classes	G06F21/602; G06F21/6218; H04L63/0428; H04L63/06; H04L63/08; H04L63/083; H04L9/0822; H04L9/0836; H04L9/0861; H04M1/67; H04W12/06; H04L2209/80; H04L9/0822; H04L9/0861; H04L9/14; H04L9/3226; H04L9/3247; H04L9/3271; H04M1/724; H04M1/72463; H04W12/04
US Classes	**None**
Family Members	US10341304B1 | US10944730B1

Abstract:

Systems, devices, media, and methods are presented for retrieving authentication credentials and decryption keys to access remotely stored user-generated content. The systems and methods receive a first authentication credential and access a second authentication credential based on receiving the first authentication credential. The system and methods generate an authentication token and an encryption token. Based on the authentication token, the system and methods access a set of encrypted content and an encrypted content key. The systems and methods decrypt the encrypted content key using the encryption token and decrypt the set of encrypted content using the decrypted content key. At least a portion of the content is presented at the user device.

Key Features	Relevant Excerpts
1. A system for programmable video assembly, comprising:	**[Column 3; Lines 34-54] The systems and methods described in the present disclosure enable protection of user-generated content (e.g., data) stored remotely from a user computing device used to generate or capture the content.** These systems and methods may employ a plurality of servers or network resources on which portions of content (e.g., encrypted user-generated content) and portions of authentication credentials are distributed. **In some embodiments, the systems and methods use derived authentication and encryption credentials. These authentication and encryption credentials (e.g., authentication tokens and encryption tokens) are derived or generated using credential elements distributed across one or more of the servers and computing devices (e.g., a user device) accessing or comprising the system. In some instances, at least one authentication element (e.g., a first authentication credential or a first authentication element) is held or remembered by the user.** The first authentication element may be used to retrieve other authentication elements and derive authentication and encryption tokens used to access encrypted user content and encrypted or secured encryption keys used to access the encrypted content. **[Column 3; Lines 55-67] As described by embodiments of the present disclosure, a user may capture a video, image, or other information with an application on a**

	smartphone. Portions of the systems and methods of the present disclosure, such as the application, may encrypt the video or other information. The encrypted video or other information prevents other users without an encryption key from accessing and viewing the video. The encryption key, used to encrypt and decrypt the video, may be generated by the system using information provided by the user and information obtained from a key server. The system may also generate one or more token to encrypt the encryption key using information provided by one or more of the servers. **Note:** *The mapped citation describes a system that employs token-based authentication and encryption mechanisms to control access to video content where the system encrypt the video or other information and prevent the other users to access the video by generating the encryption key which is used to encrypt or decrypt the video.*
1.1 a virtualization engine, wherein the engine is configured to convert a rendered video file into a reference container by extracting metadata and removing media sample data;	**[Column 2,3; Lines 58-67] Some systems have stored user-generated content in encrypted data structures. Some of these encryption-enabled systems employ user-generated passphrases (e.g., passwords) to access or decrypt the user-generated content.** Passphrase systems are often limited in security provided and encryption strength. The limited encryption, used for encrypting content or keys to access content, has a strength based on the length or complexity of user-generated passphrases which the user must remember. These systems may also be vulnerable to attack on the passphrase, such as by dictionary attacks. **Some remote storage systems employ stronger encryption with a strong key. These keys are often stored on the user's computing device.** Storing or tethering the key to the computing device prevents device mobility, limiting a user to accessing the user-generated content with the device containing the strong key. **[Column 3; Lines 55-67]** As described by embodiments of the present disclosure, **a user may capture a video, image, or other information with an application on a smartphone. Portions of the systems and methods of the present disclosure, such as the application, may encrypt the video or other information.** The encrypted video or other information prevents other users without an encryption key from accessing and viewing the video. The encryption key, used to encrypt and decrypt the video, may be generated by the system using information provided by the user and information obtained from a key server. The system may also generate one or more token to encrypt the encryption key using information provided by one or more of the servers. **[Column 4; Lines 01-06] The smartphone may then transmit the encrypted video and the encrypted key to a content server for secure storage, and transmit tokens to the key server.** When the user subsequently logs in to the application, using the same smartphone or another device, the identity of the user is verified and authenticated by the application. **[Column 19; Lines 59-63]** In some embodiments, **decrypting the one or more encrypted content elements with the one or more content keys generates one or more content elements by converting the one or more content elements from encrypted cyphertext to readable or renderable data**. **Note:** *The mapped citation describes that the system converts the user generated content (i.e. rendered video file) into encrypted data structure (i.e. encrypted video) which are stored remotely on user's computing device to a content server for secure storage where the encrypted data structure (i.e. encrypted video) is associated with the encryption keys.*
1.1.1 wherein the virtual video container retains ISO	**[Column 3; Lines 34-41]** The systems and methods described in the present disclosure **enable protection of user-generated content** (e.g., data) **stored**

Base Media File Format box structures while being devoid of media sample data.	**remotely from a user computing device used to generate or capture the content**. These systems and methods may employ **a plurality of servers or network resources on which portions of content (e.g., encrypted user-generated content) and portions of authentication credentials are distributed**. **[Column 10; Lines 9-15]** During interaction with the application on the client device 110, **the user initiates access with a content server** (e.g., database server 132) containing encrypted user-generated content. **The user-generated content is data (e.g., images, video, audio, document, or messages) generated by the user by interaction with a computing device (e.g., the client device 110)**. **[Column 20; Lines 6-13]** Although encrypted data is described with respect to cyphertext, **it should be understood that the term "cyphertext" is applicable to any encrypted data type**, without limitation. As such, although in some instances "cyphertext" refers to encrypted textual content, **it may also refer to encrypted video, audio, or any other suitable encrypted data, data type, or format**. **[Column 28; Lines 34-50]** In addition, **the libraries 806 can include API libraries 832 such as media libraries** (e.g., libraries to support presentation and manipulation of various media formats such as **Moving Picture Experts Group-4 (MPEG4)**, **Advanced Video Coding (H.264 or AVC)**, **Moving Picture Experts Group Layer-3 (MP3), Advanced Audio Coding (AAC), Adaptive Multi-Rate (AMR) audio codec, Joint Photographic Experts Group (JPEG or JPG), or Portable Network Graphics (PNG))**, graphics libraries (e.g., an OpenGL framework used to render in two dimensions (2D) and three dimensions (3D) in a graphic content on a display), database libraries (e.g., SQLite to provide various relational database functions), web libraries (e.g., WebKit to provide web browsing functionality), and the like. **The libraries 806 can also include a wide variety of other libraries 834 to provide many other APIs to the applications 810**. **Note:** *The mapped citation describes that the system stores the encrypted content of the user on the remote server. This stored data is separated into audio and video files format for further processing and these audio and video files can be of any format. However, delinking these separated files from the main rendered file is not explicitly disclosed.*
1.1.2 wherein the step of resolving referenced media data involves issuing MDAT range requests to the remote media data sources.	**[Abstract] Systems, devices, media, and methods are presented for retrieving authentication credentials and decryption keys to access remotely stored user-generated content.** The systems and methods receive a first authentication credential and access a second authentication credential based on receiving the first authentication credential. The system and methods generate an authentication token and an encryption token. Based on the authentication token, the system and methods access a set of encrypted content and an encrypted content key. **The systems and methods decrypt the encrypted content key using the encryption token and decrypt the set of encrypted content using the decrypted content key. At least a portion of the content is presented at the user device.** **Note:** *The mapped citation describes the system for decrypting the encrypted content using the encryption token where the encrypted content is decrypted using the decrypted content keys which are retrieved to access the remotely stored content which can inferred in such a way that media data is issued by accessing or requesting the remotely stored content with the help of the encryption keys. However, specific MDAT range request has not been disclosed.*
1.1.3 further comprising an artificial intelligence orchestration layer	**[Column 13; Lines 17-29]** The encryption token may encrypt the master key using one or more cryptographic functions. For example, the cyphertext of the master key (CIPH) **may be generated using an algorithm represented as CIPH=AES-ENC**

configured to modify the sequencing instructions of the reference container without accessing raw media samples.	(ET, MK), where AES-ENC is an application of AES encryption, using the encryption token (ET), to the master key (MK). Although described **with respect to encryption and decryption of specified encryption keys**, **it should be understood that the encryption token may be used for any suitable encryption, decryption, or key management function** related to the encrypted user-generated content stored on the content server. **[Column 15; Lines 40-50] The presentation component 260 may modify an existing user interface to incorporate presentation of the portion of the set of content**. For example, the presentation component 260 and the interface component 250 **may generate and insert one or more frames into a user interface presented at a display device of the client device 110**. The frames may be configured to receive and serve or display one or more content elements of the set of content. In some instances, **the user interface or frames present a representation of a content element, such as a thumbnail**.
1.2 a token issuance service, wherein the service configured to generate a cryptographicvideo token binding transaction, consent, and licensing information;	**[Column 12; Lines 60-67] In operation 330, the token component 230 generates an authentication token and an encryption token.** In some embodiments, the token component 230 generates the authentication token and the encryption token using at least one of the second authentication credential, the first authentication credential, and the session credential. **In some embodiments, the authentication token and the encryption token are generated as cryptographic keys**. **[Column 17; Lines 39-44]** In some embodiments, **after receiving the first authentication credential**, in operation 310, the access component 210 contacts the content server for a digital signature associated **with a unique identification component validating a set of transactions** (**e.g., attempted access of the encrypted content**). **[Column 13; Lines 31-47] The token component 230 may perform one or more cryptographic functions, configured to generate cryptographic tokens or keys, on the value (e.g., the user-specific random string) of the second authentication credential, to generate the authentication token and the encryption token.** For example, where the second authentication credential comprises two random strings, the token component 230 may perform a first cryptographic function on a first user-specific random string to generate the authentication token, and **may perform a second cryptographic function on a second user-specific random string to generate the encryption token.** Although described with respect to a plurality of functions, operations, or cryptographic functions, it should be understood that the **token component 230 may generate the authentication token and the encryption token using the same function, by using the function on differing values of user-specific random strings.** **Note:** *The mapped citation describes that the token component generates the authentication token and encryption token where the authentication and encryption tokens are generated as cryptographic keys. The authentication token refers to a secure user specific string that verifies user's identity corresponding to the consent information.*
1.2.1 wherein the token issuance service includes a unique session nonce in each generated token to prevent unauthorized resolution reuse.	**[Column 4,5; Lines 66-67, 1-5]** In some embodiments**, portions of content requested and transmitted to the user are encrypted, while other portions of content are unencrypted.** Unencrypted content may be provided by the system, the user, other users, third parties, combinations thereof, or any suitable source. **In some instances, the user determines the content to be encrypted and the content which remains unencrypted.** **[Column 5; Lines 28-36]** In some embodiments, the systems and methods

	described herein manage keys and authentication credentials or portions thereof used to access and encrypt/decrypt information used by or provided by the system. In some instances, each piece, portion, or file of user-generated content is encrypted with a distinct encryption key (e.g., a content key). In such instances, a set of content keys comprise the distinct encryption keys encrypting the content of a user. **[Column 3; Lines 55-67]** As described by embodiments of the present disclosure, a user may capture a video, image, or other information with an application on a smartphone. Portions of the systems and methods of the present disclosure, such as the application, may encrypt the video or other information. **The encrypted video or other information prevents other users without an encryption key from accessing and viewing the video.** The encryption key, used to encrypt and decrypt the video, may be generated by the system using information provided by the user and information obtained from a key server. The system may also generate one or more token to encrypt the encryption key using information provided by one or more of the servers. **Note:** *The mapped citation describes that only a requested content portion is encrypted while the other portion of the data remains unencrypted and in this system the keys are used to access and encrypt or decrypt the selected portion where each portion or file of content is encrypted with distinct encryption keys for encrypting the content of a user where the user without the encryption key is prevented to access or view the video which corresponds to the prevention of unauthorized use.*
1.2.2 wherein the transaction metadata includes instructions for micro payments triggered by individual segment resolution events	**N/A**
1.2.3 wherein the cryptographic video token may function as an integrated authorization gatekeeper that operates independently of the media payload by excluding media sample data and direct file paths, enforces the right to resolve by providing the necessary cryptographic permissions and binds governance logic to the resolution event.	**[Column 12; Lines 60-67] In operation 330, the token component 230 generates an authentication token and an encryption token.** In some embodiments, **the token component 230 generates the authentication token and the encryption token using at least one of the second authentication credential, the first authentication credential**, and the session credential. **In some embodiments, the authentication token and the encryption token are generated as cryptographic keys.** **[Column 3; Lines 55-67]** As described by embodiments of the present disclosure, a user may capture a video, image, or other information with an application on a smartphone. Portions of the systems and methods of the present disclosure, **such as the application, may encrypt the video or other information**. **The encrypted video or other information prevents other users without an encryption key from accessing and viewing the video. The encryption key, used to encrypt and decrypt the video, may be generated by the system using information provided by the user and information obtained from a key server**. **The system may also generate one or more token to encrypt the encryption key using information provided by one or more of the servers**. **[Column 18; Lines 50-63]** In operation 510, **the access component 210 accesses a set of encrypted content, an encrypted master key, and a plurality of encrypted content keys**. In some embodiments, the access component 210 accesses the set of encrypted content, **the encrypted master key, and the plurality of encrypted content keys** in a manner similar to or the same as described above with respect to

	methods 300 or 400. **The access component 210 may access the above-referenced data by receiving the set of encrypted content, the encrypted master key, and the plurality of encrypted content keys from the content server**. **The content server may transmit the data as a result of validating and authenticating a request for the data and an identity of the user requesting the data as being associated with the requested data.** **Note:** *The mapped citation describes that the token component generate authentication token and encryption token which are generated as cryptographic keys and the user without the encryption key cannot access or view the encrypted video where the data is transmitted only as a result of successful validation and authentication which means that the cryptographic video token may function as an integrated authorization gatekeeper.*
1.3 a secure resolution service, wherein the service dereferences remote media data upon successful validation of the cryptographic video token; further comprising logging each individual media dereference event to an auditable transaction layer.	**[Column 1; Lines 15-21] Embodiments of the present disclosure relate generally to secure storage and retrieval of user-generated content.** More particularly, but not by way of limitation, **the present disclosure addresses systems and methods for generating and distributing encryption and authentication tokens enabling access to secure user-generated content independent of a user device used to retrieve the content.** **[Column 4; Lines 01-13] The smartphone may then transmit the encrypted video and the encrypted key to a content server for secure storage, and transmit tokens to the key server. When the user subsequently logs in to the application, using the same smartphone or another device, the identity of the user is verified and authenticated by the application. The smartphone or other device, using the application, generates tokens to decrypt the encryption key.** In some embodiments, the smartphone generates the tokens with the aid of the servers. The device retrieves the encrypted key and the encrypted video from the content server and decrypts the encryption key and then the video. The smartphone or other device then displays the video for user. **[Column 18; Lines 50-63]** In operation 510, the access component 210 accesses a set of encrypted content, an encrypted master key, and a plurality of encrypted content keys. In some embodiments, the access component 210 accesses the set of encrypted content, the encrypted master key, and the plurality of encrypted content keys in a manner similar to or the same as described above with respect to methods 300 or 400. The access component 210 may access the above-referenced data by receiving the set of encrypted content, the encrypted master key, and the plurality of encrypted content keys from the content server. **The content server may transmit the data as a result of validating and authenticating a request for the data and an identity of the user requesting the data as being associated with the requested data.** **Note:** *The mapped citation describes that the system provides a secure retrieval of user generated content where the user generated content is accessed by the encryption and authentication tokens (i.e. cryptographic tokens) which are transmitted to the server to retrieve the content. When the user logs in to the application the identity of user is verified and authenticated. The content server generates or provides the data by validating and authenticating the request for the data and by validating and authenticating the identity of user too.*
1.3.1 wherein the secure resolution service is restricted to operating within a hardware-attested trusted execution environment.	**[Column 3; Lines 55-67] As described by embodiments of the present disclosure, a user may capture a video, image, or other information with an application on a smartphone.** Portions of the systems and methods of the present disclosure, such as the application, **may encrypt the video or other information.** The encrypted video or other information prevents other users without an encryption key from

	accessing and viewing the video. The encryption key, used to encrypt and decrypt the video, may be generated by the system using information provided by the user and information obtained from a key server. The system may also generate one or more token to encrypt the encryption key using information provided by one or more of the servers. **[Column 4; Lines 01-13] The smartphone may then transmit the encrypted video and the encrypted key to a content server for secure storage, and transmit tokens to the key server. When the user subsequently logs in to the application, using the same smartphone or another device, the identity of the user is verified and authenticated by the application. The smartphone or other device, using the application, generates tokens to decrypt the encryption key. In some embodiments, the smartphone generates the tokens with the aid of the servers. The device retrieves the encrypted key and the encrypted video from the content server and decrypts the encryption key and then the video. The smartphone or other device then displays the video for user.** **Note:** *The mapped citation describes that the smartphone captures the video and later the content is encrypted and transmit the encrypted video and encrypted key to content server where the content is accessed or viewed on secure validation and authentication corresponding to the secure resolution service. The smartphone is a hardware device comprising a processor which can be inferred as a hardware-attested trusted execution environment where secure resolution is taking place.*
1.3.2 wherein the logged dereference events trigger a commercial action selected from a group consisting of billing, royalty allocation, and advertising insertion.	**N/A**
1.4 a playback environment, wherein the playback environment stores multiple tokens and dynamically assembling video streams in real-time from authorized references.	**[Column 4; Lines 01-13]** The smartphone may then transmit the encrypted video and the encrypted key to a content server for secure storage, and transmit tokens to the key server. **When the user subsequently logs in to the application, using the same smartphone or another device, the identity of the user is verified and authenticated by the application.** The smartphone or other device, using the application, generates tokens to decrypt the encryption key. In some embodiments, the smartphone generates the tokens with the aid of the servers. The device retrieves the encrypted key and the encrypted video from the content server and decrypts the encryption key and then the video. **The smartphone or other device then displays the video for user.** **[Column 4; Lines 14-31]** As explained in more detail below, when first using the systems described herein, a user chooses or generates login information (e.g., a personal identification number, a password, or a passphrase). The system authenticates the user to a key server and a content server using the login information and associates the user with the login information on the servers. The user generates, provides, or is assigned a master key (e.g., a content key) for encrypting content generated by the user. **The system generates an authentication token and an encryption token. The encryption token is used to encrypt the master key, which is then deposited or stored on the content server and the key server in the encrypted form.** In some embodiments, prior to depositing the encrypted master key, content generated by the user on a device is encrypted. **The encrypted content is deposited or stored on the content server and the encrypted master key is deposited or stored along with the encrypted content, such that the two sets of information are associated.**

	[Column 5; Lines 28-36] In some embodiments, the systems and methods **described herein manage keys and authentication credentials or portions thereof used to access and encrypt/decrypt information used by or provided by the system. In some instances, each piece, portion, or file of user-generated content is encrypted with a distinct encryption key (e.g., a content key). In such instances, a set of content keys comprise the distinct encryption keys encrypting the content of a user.** **Note:** *The mapped citation describes that the smartphone displays the video to a user upon verification and authentication by generating tokens and keys which are then deposited or stored on the content server and key server where token encrypt the keys and upon the validation of keys the data is decrypted and each potion of user generated content is encrypted by distinct keys which can be inferred as multiple tokens.*
1.4.1 wherein the playback environment is configured to manage independent tokens from multiple content owners within a single session	**[Column 9; Lines 10-19] The content access system 160 is shown to include an access component 210**, an authentication component 220, a token component 230, **an encryption component 240**, an interface component 250, and a presentation component 260. All, or some, **of the components 210-260, communicate with each other, for example**, via a network coupling, shared memory, and the like. **Each component of components 210-260 can be implemented as a single component, combined into other components, or further subdivided into multiple components**. **[Column 16; Lines 30-37]** In some embodiments, **a plurality of content servers and a plurality of key servers are available to the user and the user may elect to replace one or more of the servers** used to store the user-generated content and portions of the keys unlocking the content. In such embodiments, **the content access system 160 performs one or more of the methods described in the present disclosure (e.g., method 300) to retrieve the content and keys associated with the user**. **[Column 19; Lines 22-31]** In some embodiments, **the encryption component 240 accesses the plurality of encrypted content keys as a set or packet**. In such instances, **where the plurality of encrypted content keys are provided together, the encryption component 240 may decrypt the plurality of content keys simultaneously or in a single instance of using the one or more decryption operations.** In some embodiments, **the encryption component 240 accesses the plurality of encrypted content keys as distinct files, portions of a file, data, or other separate operable elements**. In such embodiments, the encryption component 240 may decrypt one or more of the plurality of encrypted content keys separately. **[Column 17,18; Lines 64-67, 1-4]** In some embodiments, **the permission is received based on the response matching the expected response**. In some instances, **the permission is transmitted to the access component 210 so that a subsequent request for access to the encrypted content is authorized**. In such instances, **the authorization may be limited to a specified period of time, a specified number of access attempts, or a current session**.
1.4.2 wherein the playback environment may function as a content wallet capable of managing multiple tokens from independent content owners.	**[Column 9; Lines 10-19] The content access system 160 is shown to include an access component 210**, an authentication component 220, a token component 230, **an encryption component 240**, an interface component 250, and a presentation component 260. All, or some, **of the components 210-260, communicate with each other, for example**, via a network coupling, shared memory, and the like. **Each component of components 210-260 can be implemented as a single component, combined into other components, or further**

	subdivided into multiple components. **[Column 16; Lines 30-37]** In some embodiments, **a plurality of content servers and a plurality of key servers are available to the user and the user may elect to replace one or more of the servers** used to store the user-generated content and portions of the keys unlocking the content. In such embodiments, **the content access system 160 performs one or more of the methods described in the present disclosure (e.g., method 300) to retrieve the content and keys associated with the user**. **[Column 19; Lines 22-31]** In some embodiments, **the encryption component 240 accesses the plurality of encrypted content keys as a set or packet**. In such instances, **where the plurality of encrypted content keys are provided together, the encryption component 240 may decrypt the plurality of content keys simultaneously or in a single instance of using the one or more decryption operations.** In some embodiments, **the encryption component 240 accesses the plurality of encrypted content keys as distinct files, portions of a file, data, or other separate operable elements**. In such embodiments, the encryption component 240 may decrypt one or more of the plurality of encrypted content keys separately. **[Column 17,18; Lines 64-67, 1-4]** In some embodiments, **the permission is received based on the response matching the expected response**. In some instances, **the permission is transmitted to the access component 210 so that a subsequent request for access to the encrypted content is authorized**. In such instances, **the authorization may be limited to a specified period of time, a specified number of access attempts, or a current session**.
1.4.3 wherein the system is configured to terminate the video stream instantly upon the revocation of user consent parameters associated with the token.	**The authentication component 220 verifies the login credentials** with the application and opens at least a portion of the application such that the interface component 250 and **the presentation component 260 generate and cause presentation of a subsequent user interface screen comprising a subsequent set of user interface elements**. The access component 210 **transmits the authentication token to the content server**. In some instances, **the access component 210 transmits the authentication token along with the session credential used by the key server to verify that the request for the second authentication credential was valid**. In such embodiments, **the key server may compare the session credential to an expected session credential**. **A match between the session credential and the expected session credential validates that the user is authenticated with the content server, and the request for the value is valid**. In some instances, **a content removal request is accompanied by an indication of authentication**, **generated upon the user validating an identity with the authentication credentials and the authentication token. The content removal request may be subject to a delay to ensure the user time to rescind the request**. **Note:** *The mapped citation describes that the authenticating system checks for the credentials of the user which are input to view the content. In case the credentials are not matching with the stored credentials, the system takes it as content removal request from the server. Therefore, we can infer that the video playback will be terminated if the credentials are not matching with the stored authentic credentials.*

Patent Citation 5: US20110173653A1

Title	Virtual video on demand using multiple encrypted video segments
Priority Date	26 JAN 2000
Filing Date	21 MAR 2011
Publication Date	14 JUL 2011
Inventors	Robert G. Arsenault; Leon J. Stanger
Assignees	AT&T MVPD Group LLC
IPC Classes	H04N5/445; H04N7/167
CPC Classes	H04N21/26216; H04N21/4331; H04N21/44016; H04N21/47202; H04N7/17336
US Classes	725/31; 725/39
Family Members	US6701528B1 | US7926078B2 | US8584183B2

Abstract:

A method and apparatus for providing a virtual video on demand services is disclosed. The method and apparatus disclose the storing of a segment of the video program in advance for VOD viewing at a later time. When the subscriber selects VOD service, a pre-stored video segment is retrieved for presentation to the subscriber. Remaining video program segments simultaneously broadcast on a plurality of channels are recorded in parallel while the pre-stored video program segment is retrieved and presented to the user.

Key Features	Relevant Excerpts
1. A system for programmable video assembly, comprising:	**[Abstract] A method and apparatus for providing a virtual video on demand services is disclosed. The method and apparatus disclose the storing of a segment of the video program in advance for VOD viewing at a later time. When the subscriber selects VOD service, a pre-stored video segment is retrieved for presentation to the subscriber. Remaining video program segments simultaneously broadcast on a plurality of channels are recorded in parallel while the pre-stored video program segment is retrieved and presented to the user.** **Note:** *The mapped citation describes a method and apparatus for providing video on demand where the segments of video are stored, retrieved and presented to the subscriber*
1.1 a virtualization engine, wherein the engine is configured to convert a rendered video file into a reference container by extracting metadata and removing media sample data;	**[Para 0069] The storage of encrypted data packets can be accomplished in one of two ways.** First, the encrypted data packets can be decrypted by the decryption module 506 as described above, and passed through the system RAM 228 to the video storage device 232. This method is appropriate, for example, when the subscriber 110 is entitled to view all encrypted program material (e.g. a monthly subscription). Alternatively, **the receiver 200 can store the data packets in encrypted form and decrypt them for later viewing after a purchase choice is made by the customer. In one embodiment data router 514 directs encrypted data segments directly to the video storage device 232. In another embodiment, the encrypted data is sent to the video storage device 232 via the system RAM 228**. **[Para 0094]** Each **sub-segment can be assembled by sorting by a channel identifier (such as the SCID), and the sorting the assembled sub-segments in accordance**

	with a time code. The SMPTE time code, which is usually defined in terms of hours, minutes, and seconds of the program as HH.MM.SS, can be used for this purpose. **Alternatively or in combination with the foregoing, a recirculating program time stamp (PTS) value described above can be used for the time stamp**. **Note:** *The mapped citation describes that the data packets containing the data are stored in the video storage device and the data is also containing the time stamps of the sub-segments of that specific video.*
1.1.1 wherein the virtual video container retains ISO Base Media File Format box structures while being devoid of media sample data.	**[Para 0041] The transport module 208 performs many of the data processing functions performed by the receiver 200**. The transport module 208 **processes data received from the FEC decoder module 206 and provides the processed data to the video MPEG decoder 214 and the audio MPEG decoder 216**. In one embodiment of the present invention, the transport module, **video MPEG decoder and audio MPEG decoder are all implemented on integrated circuits**. This design promotes both space and power efficiency, and increases the security of the functions performed within the transport module 208. **Note:** *The mapped citation describes that the transport system fetches data from audio and video processor separately. Since, the main stream file is in single format and the processing is done on audio and video file format separately. We can deduce that there must be a system to assign media format.*
1.1.2 wherein the step of resolving referenced media data involves issuing MDAT range requests to the remote media data sources.	**[Para 0043] Video data is processed by the MPEG video decoder 214**. Using the video random access memory (RAM) 236, **the MPEG video decoder 214 decodes the compressed video data and sends it to an encoder or video processor 216, which converts the digital video information received from the video MPEG module** 214 into an output signal usable by a display or other output device. By way of example, **processor 216 may comprise a National TV Standards Committee (NTSC) or Advanced Television Systems Committee (ATSC) encoder. In one embodiment of the invention both S-Video and ordinary video (NTSC or ATSC) signals are provided. Other outputs may also be utilized, and are advantageous if ATSC high definition programming is processed**. **Note:** *The mapped citation describes that the transport system decodes mpeg video data. Since, this video format contains several other media data including NTSC, ATSC; the system assigns a format to the media data. However, the range for the specific media data is not explicitly disclosed.*
1.1.3 further comprising an artificial intelligence orchestration layer configured to modify the sequencing instructions of the reference container without accessing raw media samples.	**N/A**
1.2 a token issuance service, wherein the service configured to generate a cryptographicvideo token binding transaction, consent, and licensing information;	**[Para 0072]** In one embodiment, the decryption process is accomplished as follows. In response to a user demand and while retrieving the stored first segment 804 for presentation to the user, a message is transmitted from the IRD 200 separately identifying each of the subsequent segments 806 of the selected video program and the user to the program source such as the control center 102 or the uplink center 104. **A code or key such as the PIP is then received by the IRD 200. The key is later used to decrypt the encrypted segments so the video program can be viewed by the user.**

	[Para 0073] In another embodiment, a message is transmitted to the IRD 200 separately identifying each of the subsequent segments 806 of the selected video program and the user to the program source such as the control center 102 or the uplink center 104. **In response, a plurality of codes or keys are transmitted and received by the IRD 200, and each of the keys is used to decrypt an associated one of the encrypted subsequent segments 806.** **Note:** *The mapped citation describes the receiving of a code or key such as PIP (i.e. purchase information packet) which is later used to decrypt encrypted segment where each key is associated with one of the encrypted segment where the PIP key which corresponds to the packet consisting the purchase information i.e. transactional information.*
1.2.1 wherein the token issuance service includes a unique session nonce in each generated token to prevent unauthorized resolution reuse.	**[Claim 18] The method of claim 1, wherein each of the segments is encrypted so as to be decryptable by a different key** before being received and stored in the local storage device, and the method further comprises the steps of : in response to the user command, **transmitting a message separately identifying each of the segments of the selected video program to a source of the video program**; and **receiving the different key for each of the segments.** **[Claim 19]** The method of claim 18, further comprising **the step of decrypting the encrypted segments with the different key associated with each segment.** **[Para 0100]** In one embodiment of the present invention, the pre-stored video program segment 804 is transmitted and stored in an unencrypted form and the subsequent video program segments are transmitted and stored in an encrypted form. This allows the subscriber to store and view pre-stored video segment 804 for VOD playback without requiring the PIP, and also allows the subscriber to view at least a portion of the pre-stored video segment before requesting the remainder of the video program. **Each of the subsequent video program segments 806, however require a PIP, and hence, the remainder of the video program cannot be viewed until the receiver obtains the required PIPs for the remaining segments of the video program. These PIPs can be obtained before VOD service begins, or can be obtained after the commencement of VOD service.** **Note:** *The mapped citation describes that each segment (i.e. session) is decrypted by a different key which means that the different key is associated with each segment where the key or the PIP is specifically for preventing the unauthorized use. It allows the viewing the segment of the video for which the PIP is initiated and for the remaining segments the required PIP is must.*
1.2.2 wherein the transaction metadata includes instructions for micro payments triggered by individual segment resolution events	**[Para 0033]** FIG. 1 is a diagram illustrating an overview of a video distribution system 100. The video distribution system 100 comprises a control center 102 in communication with an uplink center 104 via a ground link 114 and with a subscriber 110 via a public switched telephone network (PSTN) or other link 120. **The control center 102 provides** program material to the uplink center 104, **coordinates with the subscribers 110 to offer pay-per-view (PPV) program services, including billing and associated decryption of video programs.** **Note:** *The mapped citation describes that the control server associates with the subscribers to offer pay per view services which includes billing (i.e. micropayment) associated with the decryption of the video where the pay per view refers to the payment for individual segment resolution which means that the subscriber is paid as he/she decrypt the video.*
1.2.3 wherein the	**[Para 0100]** In one embodiment of the present invention, the pre-stored video

cryptographic video token may function as an integrated authorization gatekeeper that operates independently of the media payload by excluding media sample data and direct file paths, enforces the right to resolve by providing the necessary cryptographic permissions and binds governance logic to the resolution event.	program segment 804 is transmitted and stored in an unencrypted form and the subsequent video program segments are transmitted and stored in an encrypted form. This allows the subscriber to store and view pre-stored video segment 804 for VOD playback without requiring the PIP, and also allows the subscriber to view at least a portion of the pre-stored video segment before requesting the remainder of the video program. **Each of the subsequent video program segments 806, however require a PIP, and hence, the remainder of the video program cannot be viewed until the receiver obtains the required PIPs for the remaining segments of the video program. These PIPs can be obtained before VOD service begins, or can be obtained after the commencement of VOD service.** **Note:** *The mapped citation describes that each of the video segment requires a required PIP and the video program cannot be viewed unit the required PIP for the segment is not obtained this corresponds to the function of authorization gatekeeper which allows the viewing of data only when required PIP for the segment is obtained.*
1.3 a secure resolution service, wherein the service dereferences remote media data upon successful validation of the cryptographic video token; further comprising logging each individual media dereference event to an auditable transaction layer.	**[Para 0061]** In the preferred embodiment, all program material is encrypted. **For viewing encrypted programming, the receiver 200 is responsible for verifying that access should be granted, and if so, decrypting the data packets so the program material can be viewed by the subscriber.** For unencrypted programming, the data router 514 directs the data packets directly in the system RAM 228 via DMA 508. **[Para 0072]** In one embodiment, the decryption process is accomplished as follows. In response to a user demand and while retrieving the stored first segment 804 for presentation to the user, a message is transmitted from the IRD 200 separately identifying each of the subsequent segments 806 of the selected video program and the user to the program source such as the control center 102 or the uplink center 104. **A code or key such as the PIP is then received by the IRD 200. The key is later used to decrypt the encrypted segments so the video program can be viewed by the user.** **[Para 0073]** In another embodiment, a message is transmitted to the IRD 200 separately identifying each of the subsequent segments 806 of the selected video program and the user to the program source such as the control center 102 or the uplink center 104. **In response, a plurality of codes or keys are transmitted and received by the IRD 200, and each of the keys is used to decrypt an associated one of the encrypted subsequent segments 806.** **[Para 0014]** To prevent the customer from being billed for multiple viewing of the same program, a modified billing system recognizes that the program segments sent to the customer's IRD were part of a VOD program, and would bill the customer for a single viewing of all of the video segments. In one embodiment, separate PIPs for VOD service are defined, each of which having a value which is an appropriate (e.g. pro-rated according to the time length of the segment) fraction of the total charge for the complete program defined. In another embodiment, **the billing system recognizes the PIPs as associated with program segments which were broadcast simultaneously, and adjusts the bill for a single viewing accordingly.** **Note:** *The mapped citation describes that the receiver is responsible for verifying the access grant if grant is verified only then the data packets are decrypted and viewed by the subscriber when the key or code such as PIP (i.e. token) is used for decrypting the encrypted segments which corresponds to decrypting the encrypted segment upon the verification of key. Also, the PIP for segment or data packet is further recognized by the billing system (i.e. transaction layer) which adjusts the bill for viewing.*

1.3.1 wherein the secure resolution service is restricted to operating within a hardware-attested trusted execution environment.	**[Para 0043] Video data is processed by the MPEG video decoder 214**. Using the video random access memory (RAM) 236, the MPEG video decoder 214 decodes the compressed video data and sends it to an encoder or video processor 216, which converts the digital video information received from the video MPEG module 214 **into an output signal usable by a display or other output device**. **Note:** *The mapped citation describes that the video data is processed by the MPEG decoder which is in a format that can be displayed on the output devices. However, a specific hardware device is not explicitly disclosed.*
1.3.2 wherein the logged dereference events trigger a commercial action selected from a group consisting of billing, royalty allocation, and advertising insertion.	**[Para 0014]** In one embodiment, **the IRD solves this problem by storing conditional access information such as a purchase information parcel (PIP) with each stored program segment.** When the user makes a VOD demand, a message is sent from the IRD identifying each of the subsequent segments of the video program. In response, the IRD receives the PIPs corresponding to the subsequent program segments, and decodes, and splices them together as required. To prevent the customer from being billed for multiple viewing of the same program, a modified billing system recognizes that the program segments sent to the customer's IRD were part of a VOD program, and would bill the customer for a single viewing of all of the video segments. In one embodiment, **separate PIPs for VOD service are defined, each of which having a value which is an appropriate (e.g. pro-rated according to the time length of the segment) fraction of the total charge for the complete program defined.** In another embodiment, **the billing system recognizes the PIPs as associated with program segments which were broadcast simultaneously, and adjusts the bill for a single viewing accordingly.** **Note:** *The mapped citation describes that the PIP (i.e. purchase information packet) is also stored with each segment where the PIP has the value of total charges for the complete program. The billing system recognizes the PIP and adjust the bill for single viewing which means that the PIP consist the charges for the program which is used by the billing system for billing.*
1.4 a playback environment, wherein the playback environment stores multiple tokens and dynamically assembling video streams in real-time from authorized references.	**[Para 0072]** In one embodiment, the decryption process is accomplished as follows. In response to a user demand and while retrieving the stored first segment 804 for presentation to the user, a message is transmitted from the IRD 200 separately identifying each of the subsequent segments 806 of the selected video program and the user to the program source such as the control center 102 or the uplink center 104. **A code or key such as the PIP is then received by the IRD 200. The key is later used to decrypt the encrypted segments so the video program can be viewed by the user.** **[Claim 18] The method of claim 1, wherein each of the segments is encrypted so as to be decryptable by a different key** before being received and stored in the local storage device, and the method further comprises the steps of : in response to the user command, transmitting a message separately identifying each of the segments of the selected video program to a source of the video program; and **receiving the different key for each of the segments.** **[Claim 19]** The method of claim 18, further comprising **the step of decrypting the encrypted segments with the different key associated with each segment.** **[Para 0012]** Once the user demands VOD playback, the pre-stored video segment is played back to the user, while the remaining subsequent segments of the video program are received and recorded in parallel. These subsequent segments are spliced to the pre-stored segment and to each other to give the appearance of VOD playback. In one embodiment, **the IRD acquires and stores a purchase information**

	packet (PIP) for each program segment. **Note:** *The mapped citation describes that the key or code such as PIP (i.e. purchase information packet) is received by IRD (i.e. integrated receiver/decoder) which is later used to decrypt the encrypted segments of video which means that PIP acts as a token/key and there is specific key associated with the each segment which corresponds to the presence of multiple keys and the PIP (i.e. token/key) is stored for each segment by the IRD (i.e. integrated receiver/decoder).*
1.4.1 wherein the playback environment is configured to manage independent tokens from multiple content owners within a single session	**N/A**
1.4.2 wherein the playback environment may function as a content wallet capable of managing multiple tokens from independent content owners.	**N/A**
1.4.3 wherein the system is configured to terminate the video stream instantly upon the revocation of user consent parameters associated with the token.	**[Para 0070]** Two levels of encryption can also be used to implement an **additional layer of access control for PPV programs**. **A first level of access control can be used to limit access to persons who are authorized to purchase a PPV program** (e.g. subscribers) and **the second level of access control can be used to limit access to the PPV program to those who have actually purchased a PPV program**. **[Para 0061]** In the preferred embodiment, **all program material is encrypted**. For viewing encrypted programming, **the receiver 200 is responsible for verifying that access should be granted, and if so, decrypting the data packets so the program material can be viewed by the subscriber**.

Patent Citation 6: US20250181747A1

Title	Media streaming
Priority Date	20 NOV 2015
Filing Date	22 JAN 2025
Publication Date	05 JUN 2025
Inventors	Racz Pierre; Rioux Frederic
Assignees	Genetec Inc
IPC Classes	G06F21/10; G06F21/60; G06F21/62; H04L9/08; H04L9/40; H04N21/234; H04N21/254; H04N21/44; H04N21/441; H04N21/845; H04N21/8547
CPC Classes	G06F16/113; G06F21/10; G06F21/602; G06F21/6218; H04L63/0457; H04L9/0825; H04N21/234; H04N21/23412; H04N21/2351; H04N21/2541; H04N21/27; H04N21/4223; H04N21/44012; H04N21/441; H04N21/835; H04N21/8456; H04N21/85406; H04N21/8547
US Classes	None
Family Members	EP3378235A1 | CA3005479A1 | US10915647B2 | JP6966439B2 | WO2017083985A1 | US11397824B2 | US11853447B2 | US12229300B2

Abstract:

A media playback system for presenting to a user a composition of a plurality of media streams. It has a media selection component configured to receive a scenario dataset, to receive user input for selecting viewing times defining segments of media and composition selections, and to output a list of segments of media from the scenario dataset that are authorized to be viewed by the user. The system has a playback control component configured to retrieve from media storage at least the segments of media from the output list of segments, to decode the segments of media, and to compile composition instructions. The system has a media playback component configured to receive the rendered media and the composition instructions.

Key Features	Relevant Excerpts
1. A system for programmable video assembly, comprising:	**[Abstract] A media playback system for presenting to a user a composition of a plurality of media streams**. It has a media selection component configured to receive a scenario dataset, to receive user input for selecting viewing times defining segments of media and composition selections, and to output a list of segments of media from the scenario dataset that are authorized to be viewed by the user. The system has a playback control component configured to retrieve from media storage at least the segments of media from the output list of segments, to decode the segments of media, and to compile composition instructions. The system has a media playback component configured to receive the rendered media and the composition instructions. **Para [0011]** In this application, the term "**stream" is used in accordance with its standard meaning of to transmit or receive data (especially video and audio material but can also be other types of data**) over the Internet or other data network as a steady, continuous flow, with the understanding that a stream has inherently synchronization or time stamp data and that a stream can include data, such GPS coordinates, an image, image annotation or a text comment, that can be a series of events referenced in time**.**

	Para [0032] Similar to a video stream that contains a sequence of frames providing the images to display or an audio stream containing audio data, **a "fusion stream" can be defined as a versioned data stream made of a composition of elementary streams (or other fusion streams) for a given logical scenario** that allows secure, traceable, and controlled collaboration and sharing. The fusion stream provides a series of "**fusion nodes" containing the information needed to render the logical scenario, which can include video, audio, motion levels, bookmarks, overlays, events, permissions, encryption keys, etc**. **Para [0073]** A fusion stream can be defined in any markup language format that can be read by humans and machines, but for the purpose of describing this invention, **the JSON format is used in this example to define the structure of a fusion stream**. **Para [0101]** The way for an application (ex. player, viewer, editor) to get and manipulate the data a fusion stream provides will be described. **A programming interface may be used by a software application to use the fusion stream as a data structure.** **Note:** *The mapped citation describes a media playback system that programmable assembles multiple video/audio/metadata streams into a synchronized output using a data-driven hierarchical structure. The fusion stream is a versioned, data-driven structure (e.g., JSON) that combines elementary streams with overlays, events, permissions, and encryption, enabling secure, traceable, and flexible collaboration.*
1.1 a virtualization engine, wherein the engine is configured to convert a rendered video file into a reference container by extracting metadata and removing media sample data;	**[Para 0033] There are two principal elements: 1) the fusion stream (made of fusion nodes) and 2) the elementary stream (made of segments).** Both elements evolve in the streaming dimension (i.e. their data varies as a function of time). However, the fusion stream also has a composition versioning (i.e. its composition varies over time). **[Para 0034]** Fusion node: Logical entity that provides, for a specific time T of a scenario, **a set of key-value pairs (attributes) and a list of segments**. The segments belong to different elementary streams stored independently. Since a stream is an unbounded sequence of data, the fusion stream may be represented as a sequence of fusion nodes describing the composition of elementary streams of a scenario at any given time. **A fusion node records the composition of the fusion stream at time T and all its segments that provide data for time T**. **[Para 0045]** Segment: Logical entity that contains a finite subset of the data (for example, **a finite list of frames**) composing an elementary stream. A segment has a start time and an end time. Typically, **a segment corresponds to a file stored on disk** and several segments are used to compose an elementary stream. **[Para 0046]** Therefore, for any point in time covered by the recording of a logical scenario (i.e. the streaming time dimension), **the fusion stream provides:** **The list of segments composing the scenario to be rendered**. **The location of each segment; that is, where it is stored (server, disks, cloud platform, etc.). Information on how to synchronize the segments to combine them appropriately**. **[Para 0150] Fusion streams provide the capability to store any type of stream (video, audio, metadata, encryption keys, etc.)** into a single entity. Because of this, it becomes easier to manipulate streams to add new content to them after they were recorded. **[Para 0052]** All streams can be individually encrypted in the following way: **Data streams are encrypted with a randomly generated symmetric key that changes periodically**. The resulting sequence of symmetric keys (a.k.a. the master

Page **53** of **85**

	key stream) **is then encrypted with the public key** (asymmetric encryption) of the person to whom a viewing permission over that stream should be granted**.** **Note:** *The mapped citation describes that the system accesses rendered digital video files through fusion nodes and segments.* The system *Identifies timing metadata and synchronization information to align stream. Also describe a mechanism to remove or restrict access to media sample data by encrypting streams and issuing client-specific key streams. This combination of hierarchical stream management, metadata synchronization, and selective encryption/removal of data aligns with the concept of a virtualization engine that can manipulate rendered video files at the metadata and sample level.*
1.1.1 wherein the virtual video container retains ISO Base Media File Format box structures while being devoid of media sample data.	**N/A**
1.1.2 wherein the step of resolving referenced media data involves issuing MDAT range requests to the remote media data sources.	**[Para 0122]** As described herein, **the stream database 105 stores information (stream properties) about segment files: file location (URL), file size, segment length, protection indicator, links with other streams**. A database is used to allow the indexing of information to accelerate queries, however information can also be maintained in files. When a database is used, a database management system (e.g. MySQL, Microsoft SQL Server, Oracle, etc.) may be used to structure the information. Similarly, **the configuration database 104 stores the configuration parameters of the archive manager 100: list of sources, recording schedules, streams to record, etc. The configuration parameters may be stored using a database management system** (e.g. MySQL, Microsoft SQL Server, Oracle, etc.) and/or can also be maintained in files. **[Para 0046]** Therefore, for any point in time covered by the recording of a logical scenario (i.e. the streaming time dimension), the fusion stream provides: The list of segments composing the scenario to be rendered. **The location of each segment; that is, where it is stored (server, disks, cloud platform, etc.).** Information on how to synchronize the segments to combine them appropriately. [Para 0121] Each file represents a segment of a stream, **each segment being stored on the resource server 106.** The stream recorder 102 creates as many files as necessary on the **resource server 106** to store the stream segments following the configuration parameters. **The properties of each segment (stream properties) are transmitted to the archive manager 100 for storage in the stream database 105**. **[Para 0144]** now with this additional text **Module 21** provides **a user interface to select one or more of available, permitted segments for viewing, along with any specific time markers (i.e. start time or optionally one or more time ranges)** and composition details such as offsets, windows, text overlays in a video window (for example for building security badge reader data, GPS data overlay, license plate recognition overlays, user annotations or speech to text generated annotations), text displays, still image displays, relative volumes of sound tracks, etc. **[Para 0147]** As already mentioned, the download module 23 identifies from the **selected segment definitions found in the fusion nodes the location of the resources to be retrieved and manages the download requests to and buffering of the data received from the resource servers 12**.

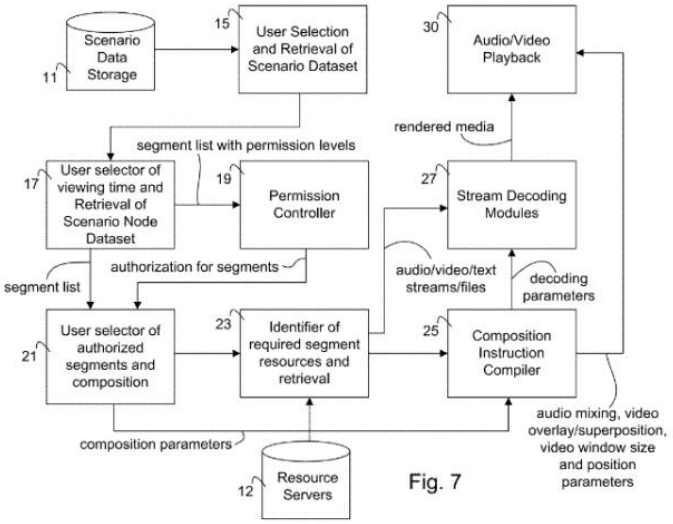
	[Para 0215] This requirement becomes always more important over time with each new installation being deployed. When a video segment **VSeg_1 encoded with the H.264 format and an audio segment ASeg_1 encoded with the G.711 format are encrypted with the certificate (public key) of a user (UCer_1), the fusion node automatically provides an asymmetrically encrypted user-specific key segment (VUKey_1) necessary to decrypt the symmetrically encrypted video segment.** The fusion node also provides a second asymmetrically encrypted user-specific key segment (AUKey_1) necessary to decrypt the symmetrically encrypted audio segment. With this information, the user selector of authorized segments and composition module 21 generates the following composition parameters transmitted to module 25, **where VSeg_1, VUKey_1, ASeg_1, AUKey_1, and UCer_1 are references to the segments and the user certificate**: **Note:** *The mapped citation explains that the fusion stream defines which segments compose a scenario, where they are stored, and how they synchronize; the stream database (105) stores segment metadata, the resource server (106) stores the segment files created by the stream recorder (102), and the download module (23) retrieves and buffers resources based on fusion node definitions. The system also provides a user interface for selecting permitted segments and time markers, and by enabling time-range selection, module 21 fetches only relevant media portions using MDAT range requests (partial byte-range retrieval from the mdat box).*
1.1.3 further comprising an artificial intelligence orchestration layer configured to modify the sequencing instructions of the reference container without accessing raw media samples.	**[Para 0199]** However, it can be appreciated that **each segment referenced by a fusion node can also have a different time span considering that each type of stream (video, audio, GPS, etc.) can have its own algorithm used to create the files**. For example, video files can last for 10 minutes because the quantity of generated video data is more important, **while audio files can last 20 minutes and metadata files can last 1 hour or more**. They can also have the same length, however not being created at the same time, thus containing data for a different period of time.
1.2 a token issuance service, wherein the service configured to generate a cryptographicvideo token binding transaction, consent, and licensing information;	**[Para 0213]** Once permission controller module 19 has completed these checks, the segments for which the user is denied access are known in advance and will not be retrieved uselessly from resource server 12. **Permission controller module 19 then transmits to the user selector of authorized segments and composition module 21 an authorization token for each segment** originally provided by the selector and retrieval module 17. **The token associate the segment GUID with the authorization decision,** which is a Boolean “denied” or “granted”. **[Para 0214]** The user selector of authorized segments and composition **module 21 provides a user interface that shows the segments of the fusion stream that the user is authorized to view as indicated by the authorization tokens received from permission controller module 19.** As shown in FIG. 9 , a timeline is generated and displayed to show the time periods for which content of at least one segment is available within the scenario. The timeline shows authorized content for any given time. It also shows gaps for which no data is available. **[Para 0010]** Applicant has also discovered that **streams can be encrypted using time-varying session keys,** and the encrypted streams can be delivered along with symmetric session key streams to a destination node. **The session key stream can be encrypted using an asymmetric key so that retrieval of the session keys contained in the session key stream requires the destination node to use a complementary key.** The synchronization of the encrypted media stream and the session key stream can be achieved using a root stream that specifies to the destination node how to synchronize the streams, so as to apply correct session keys. **[Para 0211] Permission controller module 19 performs several checks to determine**

	if a user has the permission to view the content provided by a segment of an elementary stream. A first check made by permission controller module 19 is to compare the minimum access level specified for the segment with the user access level configured in the security monitoring system. **[Para 0142]** Module 17 also extracts the permission level or any other suitable permission control data and provides same to module 19. **Module 19 determines whether the user is authorized to view each of the segments**. **[para 0211]** A third check performed by module 19 is to verify for an encrypted segment if the user can decrypt the segment—to avoid, for instance, the inefficient situation where the encrypted segment is obtained from a remote storage server and where it is realized at the rendering time that the user does not have the private key to decrypt the segment. **Permission controller module 19 compares the user's certificate with the collection of certificates associated with the fusion stream.** Using the following method, permission controller module 19 can request from fusion manager 200 through fusion interface 201 the collection of certificates pertaining to the fusion stream to which the segment belongs. **[Para 0212]** Fusion streams are implemented using the ITU-T X.509 standard to manage user certificates. **In cryptography, X.509 is an important standard that is widely accepted internationally for a Public Key Infrastructure (PKI) used to manage digital certificates and public-key encryption**. **Note:** *The mapped citation describes that Permission Controller Module (19) checks user access levels, certificates, and decryption capabilities, and issues authorization tokens for each segment binding the segment GUID to a grant or deny decision. User Selector & Composition Module (21) uses these tokens to display only permitted segments; however, the citation does not explicitly disclose binding transaction, consent, or licensing information within the token.*
1.2.1 wherein the token issuance service includes a unique session nonce in each generated token to prevent unauthorized resolution reuse.	**[Para 0010]** Applicant has also discovered that **streams can be encrypted using time-varying session keys, and the encrypted streams can be delivered along with symmetric session key streams to a destination node**. The session key stream can be encrypted using an asymmetric key so that retrieval of the session keys contained in the session key stream requires the destination node to use a complementary key. The synchronization of the encrypted media stream and the session key stream can be achieved using a root stream that specifies to the destination node how to synchronize the streams, so as to apply correct session keys. **[Claim 27]** The method as defined in any one of claims 20 to 26, wherein one of said streams is a stream of **periodically changing symmetric session keys used to encrypt and decrypt at least one other of said streams.** **[Para 0052] All streams can be individually encrypted** in the following way: **Data streams are encrypted with a randomly generated symmetric key that changes periodically**. The resulting sequence of **symmetric keys (a.k.a. the master key stream) is then encrypted with the public key (asymmetric encryption) of the person to whom a viewing permission over that stream should be granted**. The resulting stream of that encryption process is a client-specific key stream. **There can be as many client-specific key streams as there are persons to whom viewing permission should be granted**. **If a viewing permission must be limited to a given sequence (portion of a stream), the length of the client-specific key stream can be limited by only encrypting a subset of the master key stream. This gives a very granular control on who can see**

	what content. **[Para 0180-0183]** Consider two types of streams: **Data streams: video, audio, metadata.** **Key streams: encryption keys**. When encryption is enabled, **the system encrypts each data stream individually using a symmetric key that changes periodically**, which corresponds to a master key stream. Symmetric encryption can be used because it is faster than asymmetric encryption and requires less processing resources. **[Para 0213]** Once permission controller module 19 has completed these checks, the segments for which the user is denied access are known in advance and will not be retrieved uselessly from resource server 12. **Permission controller module 19 then transmits to the user selector of authorized segments and composition module 21 an authorization token for each segment** originally provided by the selector and retrieval module 17. **The token associate the segment GUID with the authorization decision,** which is a Boolean "denied" or "granted". **[Para 0215]** This requirement becomes always more important over time with each new installation being deployed. When a video segment **VSeg_1 encoded with the H.264 format and an audio segment ASeg_1 encoded with the G.711 format are encrypted with the certificate (public key) of a user (UCer_1), the fusion node automatically provides an asymmetrically encrypted user-specific key segment (VUKey_1) necessary to decrypt the symmetrically encrypted video segment.** The fusion node also provides a second asymmetrically encrypted user-specific key segment (AUKey_1) necessary to decrypt the symmetrically encrypted audio segment. With this information, the user selector of authorized segments and composition module 21 generates the following composition parameters transmitted to module 25, **where VSeg_1, VUKey_1, ASeg_1, AUKey_1, and UCer_1 are references to the segments and the user certificate**: **Note:** *The mapped citation describes that Permission Controller Module checks user access levels, certificates, and decryption capabilities, and issues authorization tokens for each segment binding the segment GUID to a grant or denies decision. User Selector & Composition Module (21) uses these tokens to display only permitted segments; however, the citation does not explicitly disclose binding transaction, consent, or licensing information within the token.*
1.2.2 wherein the transaction metadata includes instructions for micro payments triggered by individual segment resolution events	**N/A**
1.2.3 wherein the cryptographic video token may function as an integrated authorization gatekeeper that operates independently of the media payload by excluding media sample data and direct file paths, enforces the right to resolve by providing the necessary cryptographic	**[Para 0213]** Once permission controller module 19 has completed these checks, the segments for which the user is denied access are known in advance and will not be retrieved uselessly from resource server 12. **Permission controller module 19 then transmits to the user selector of authorized segments and composition module 21 an authorization token for each segment originally provided by the selector and retrieval module 17. The token associate the segment GUID with the authorization decision, which is a Boolean "denied" or "granted**". **[Para 0214]** The user selector of authorized segments and composition module 21 provides a user interface that shows the segments of the fusion stream that **the user is authorized to view as indicated by the authorization tokens received from**

permissions and binds governance logic to the resolution event.	**permission controller module 19**. **[Para 0185]** Using asymmetric encryption, **the system then encrypts the master key stream using the certificate (public key) of each user who is granted access to it.** Performing asymmetric encryption is more resource intensive than symmetric encryption, but because the master key stream represents a small amount of data, it becomes more acceptable. **A separately encrypted client-specific key stream is generated for each combination of user certificate and data stream,** as shown in this example. Camera with one video stream and one audio stream. **Each stream is encrypted with two user certificates (A and B).** **User with certificate B is watching live video (no audio).** **[Para 0142] Module 17 also extracts the permission level or any other suitable permission control data and provides same to module 19. Module 19 determines whether the user is authorized to view each of the segments.** **[Para 0032]** Similar to a video stream that contains a sequence of frames providing the images to display or an audio stream containing audio data, a "fusion stream" can be defined as a versioned data stream made of a composition of elementary streams (or other fusion streams) for a given logical scenario that allows secure, traceable, and controlled collaboration and sharing. **The fusion stream provides a series of "fusion nodes" containing the information needed to render the logical scenario, which can include video, audio, motion levels, bookmarks, overlays, events, permissions, encryption keys, etc**. **Note:** *The mapped citation describes that the permission controller generates tokens that bind each segment's GUID to a Boolean decision ("granted" or "denied"), ensuring access control is enforced without embedding media samples. Asymmetric encryption is used to secure the master key stream with each user's certificate. This ensures that only authorized users receive client-specific key streams, embedding cryptographic permissions into the resolution process.*
1.3 a secure resolution service, wherein the service dereferences remote media data upon successful validation of the cryptographic video token; further comprising logging each individual media dereference event to an auditable transaction layer.	**[Para 0142] Module 17 also extracts the permission level or any other suitable permission control data and provides same to module 19. Module 19 determines whether the user is authorized to view each of the segments.** **[Para 0148]** Module 25 is provided the types of the streams retrieved and determines which decoders 27 should be used to decode the downloaded streams or files (segments). **The decoding parameters can be, for example, provided by software module call parameters sent by module 25. The audio mixing, video overlay/superposition, video window size and position parameters can be provided by calls to a rendering engine, device drivers and/or the operating system.** **[Para 0213]** Once permission controller module 19 has completed these checks, the segments for which the user is denied access are known in advance and will not be retrieved uselessly from resource server 12. **Permission controller module 19 then transmits to the user selector of authorized segments and composition module 21 an authorization token for each segment originally provided by the selector and retrieval module 17. The token associate the segment GUID with the authorization decision, which is a Boolean "denied" or "granted**". **[Para 0217]** The composition instruction **compiler module 25 provides an interface that is used by the user selector of authorized segments** and composition module 21 to pass the composition parameters under the form of commands: **Decrypt** (srcSegment, keySegment, userCertificate) **Decrypt the key segment with the private key corresponding to the user certificate**,

	then decrypt the source segment with the decrypted key segment. **Decode (srcSegment, segmentType)** **Decode the source segment using the specified decoding module**. **[Para 0121] Each file represents a segment of a stream, each segment being stored on the resource server 106.** **Note:** *The mapped citation describes a secure process in which Modules 17 and 19 extract permission data, make authorization decisions, and bind each segment to a grant-or-deny token to exclude unauthorized content before retrieval; Module 25 then issues cryptographic commands to validate certificates and enforce secure access, so remote media data is resolved only after token validation. However, it does not disclose logging each individual media dereference event to an auditable transaction layer.*
1.3.1 wherein the secure resolution service is restricted to operating within a hardware-attested trusted execution environment.	**[Para 0030]** As shown in FIG. 1 **, a playback workstation 10 is connected to servers 11 and 12 via a network 14. The workstation 10 can be a desktop computer, a tablet computer, smartphone, smart TV or the like**. The server 11 is shown as hosting scenario data that guides the computer 10 in processing media streams from server or servers 12. **The scenario data on server 11 can be co-hosted with the resource data on server 12**. **[Para 0166] Composition attribute added to the segments recorded with the smartphone in the current example**. It can represent the delay in milliseconds required to have those streams synchronized with the streams recorded by the device installed in Bus #1. When reading this attribute, the player knows to add a delay.
1.3.2 wherein the logged dereference events trigger a commercial action selected from a group consisting of billing, royalty allocation, and advertising insertion.	**N/A**
1.4 a playback environment, wherein the playback environment stores multiple tokens and dynamically assembling video streams in real-time from authorized references.	**[Para 0149]** The composition instruction compiler 25, **stream decoding modules 27 and playback devices 30 then collaborate to render and display or generate the audio and/or video streams** (step S9 in FIG. 8) and/or any other elementary stream present in the composition of the scenario being played. **[Para 0175] When the scenario is selected for playback, the player retrieves the fusion stream and looks for its composition at the selected time.** If the playback is started at the time the incident occurs (23:00), the player seeks to this time within the fusion stream and retrieves the fusion node that contains the 7 segments shown previously. The player then plays each stream simultaneously at the same time TX to render the complete scenario. **[Para 0210]** This **method returns the list of segments for the specified time range and filtered by elementary streams or all elementary streams if no filter is specified**. **The selector and retrieval module 17 can also extract the segments associated with a coming time range of the playback so that the permissions can be checked and the segments can be retrieved** in advance from the Resource Servers 12 to avoid lags in the playback. **The list of segments retrieved from the fusion node with the QuerySequencesAsync method is transmitted to the permission controller 19.** The selector and retrieval module 17 also transmits the same list to the user selector of authorized segments and composition module 21. **[Para 0213]** Once permission controller module 19 has completed these checks, the

	segments for which the user is denied access are known in advance and will not be retrieved uselessly from resource server 12. **Permission controller module 19 then transmits to the user selector of authorized segments and composition module 21 an authorization token for each segment originally provided by the selector and retrieval module 17. The token associate the segment GUID with the authorization decision, which is a Boolean "denied" or "granted**".  Fig. 7 **Note:** *The mapped citation describes a playback environment in which fusion nodes containing multiple segments are retrieved and assembled in real time. The selector and retrieval module gather the relevant segments, while the permission controller checks access rights before playback. Authorization tokens are issued for each segment, linking them to a "granted" or "denied" decision, and only permitted segments are retrieved from resource servers. The player then dynamically assembles and renders the authorized audio, video, or other streams simultaneously, ensuring smooth playback composed solely of authorized references.*
1.4.1 wherein the playback environment is configured to manage independent tokens from multiple content owners within a single session	**[Para 0128]** A client application may use fusion engine 209 to connect to any fusion archiver 210 available on the network. **The owner parameter allows for fusion stream grouping under a unique owner identifier**. This means that **multiple fusion stream repositories 206 can be stored on the same server**. **For example, fusion streams created under Owner_A will only be visible when retrieving the repository of Owner_A. Fusion streams that belong to other owners will not be returned**. **Once connected to the repository, the client application can obtain the list of fusion streams stored on fusion stream repositories 206**. In this example, a new fusion stream is created in the repository using the following method provided by fusion interface 201. **[Para 0010]** Applicant has also discovered that streams can be encrypted using time-varying session keys, and the **encrypted streams can be delivered along with symmetric session key streams to a destination node**. The session key stream can be encrypted using an asymmetric key so that **retrieval of the session keys contained in the session key stream** requires the destination node to use a complementary key. The synchronization of the encrypted media stream and the session key stream can be achieved using a root stream that specifies to the destination node how to synchronize the streams, so as to apply correct session keys. **[Para 0213]** Once permission controller module 19 has completed these checks, the segments for which the user is denied access are known in advance and will not be retrieved uselessly from resource server 12. **Permission controller module 19 then transmits to the user selector of authorized segments and composition module 21**

	an authorization token for each segment originally provided by the selector and retrieval module 17. The token associate the segment GUID with the authorization decision, which is a Boolean "denied" or "granted". **[Para 0069] A fusion stream purposely regroups independent elementary streams under a single logical scenario,** whether is it a case under investigation, data associated with a unique officer, data from fixed cameras following a moving vehicle throughout the city, etc. An elementary stream, or some of its segments, can be referenced by more than one fusion stream concurrently**.** **Note:** *The mapped citation describes that the playback environment connects to fusion repositories grouped under unique owner identifiers, allowing multiple content owners' streams to coexist on the same server while remaining independently managed and visible only within their respective repositories. It further describes that independent streams are logically fused, permitting shared references across scenarios. However, it does not disclose multiple content owners within a single session.*
1.4.2 wherein the playback environment may function as a content wallet capable of managing multiple tokens from independent content owners.	**[Para 0069] A fusion stream purposely regroups independent elementary streams under a single logical scenario,** whether is it a case under investigation, data associated with a unique officer, data from fixed cameras following a moving vehicle throughout the city, etc. An elementary stream, or some of its segments, can be referenced by more than one fusion stream concurrently**.** **[Para 0128]** A client application may use fusion engine 209 to connect to any fusion archiver 210 available on the network. **The owner parameter allows for fusion stream grouping under a unique owner identifier**. This means that **multiple fusion stream repositories 206 can be stored on the same server**. **For example, fusion streams created under Owner_A will only be visible when retrieving the repository of Owner_A. Fusion streams that belong to other owners will not be returned**. **Once connected to the repository, the client application can obtain the list of fusion streams stored on fusion stream repositories 206**. In this example, a new fusion stream is created in the repository using the following method provided by fusion interface 201: **[Para 0213]** Once permission controller module 19 has completed these checks, the segments for which the user is denied access are known in advance and will not be retrieved uselessly from resource server 12. **Permission controller module 19 then transmits to the user selector of authorized segments and composition module 21 an authorization token for each segment originally provided by the selector and retrieval module 17. The token associate the segment GUID with the authorization decision, which is a Boolean "denied" or "granted**". **[Para 0214] The user selector of authorized segments and composition module 21 provides a user interface that shows the segments of the fusion stream that the user is authorized to view as indicated by the authorization tokens received from permission controller module 19**. As shown in FIG. 9 , a timeline is generated and displayed to show the time periods for which content of at least one segment is available within the scenario. **[Para 0148] The decoding parameters can be, for example, provided by software module call parameters sent by module 25. The audio mixing, video overlay/superposition, video window size and position parameters can be provided by calls to a rendering engine, device drivers and/or the operating system.** **Note:** *The mapped citation describes that independent streams are logically fused, allowing shared references across scenarios, while the playback environment (Modules 17, 19, 21) pre-checks segment permissions and issues tokens to play only authorized content. It also describes connecting to fusion repositories under unique*

	owner identifiers, so multiple content owners' streams can coexist on the same server, with each owner's streams visible only when accessed via authorization tokens.
1.4.3 wherein the system is configured to terminate the video stream instantly upon the revocation of user consent parameters associated with the token.	**[Para 0143] Permissions can also be granted by linking the user's certificate to a segment**, **and more generally to a stream source**. Additionally, if the segments are encrypted, **module 19 can determine the segments for which the user has the proper private key**. **Once the permissions are determined**, **module 19 then provides the authorization information for the segments to module 21**. **[Para 0272]** bool RemoveElementaryStream(Guid id); void AddCertificate(X509Certificate2 certificate); **bool RemoveCertificate(X509Certificate2 certificate, bool revokecertificate)**; Task<IEnumerable<ISequence>> QuerySequencesAsync(IDateTimeRange boundaries, IEnumerable<Guid> elementaryStreamFilter = null);

Patent Citation 7: US20210120278A1

Title	Digital Encryption of Tokens Within Videos
Priority Date	31 MAY 2017
Filing Date	30 DEC 2020
Publication Date	22 APR 2021
Inventors	Ericson Braden Christopher; Pourcyrous Sam; Jou Fun-Chen; An Kaili
Assignees	Paypal Inc.
IPC Classes	H04L9/08; H04L9/32; H04N21/2347; H04N21/235; H04N21/266; H04N21/4405; H04N21/478; H04N21/835; H04N21/84
CPC Classes	H04L9/0825; H04L9/3213; H04L9/3226; H04N21/2347; H04N21/2351; H04N21/26606; H04N21/26613; H04N21/44055; H04N21/47815; H04N21/835; H04N21/84; H04L2209/56
US Classes	None
Family Members	US10893306B2 | US11665382B2

Abstract:

Embedding of digital tokens within a digital video can occur cryptographically using a public key in some embodiments. The digital video may be altered in a variety of ways so that the video itself contains an integrated token that can represent various quantities. Audiovisual data can be altered to contain both a token and a perceptible user auditory or visual cue as to the presence of the encrypted digital token. A video with an embedded digital token may be sent to users on the Internet. A video recipient may be able to view the video and also take additional action or gain additional functionality from the digital token embedded in the video. Tokens can be embedded by altering video metadata so that the perceptible video content itself is not changed in some embodiments.

Key Features	Relevant Excerpts
1. A system for programmable video assembly, comprising:	**Para [0014] The present specification allows for the embedding of digital tokens within a digital video.** These tokens may be cryptographically obscured using an encryption key, such as a public key of a public/private key pair. **An encrypted digital token may therefore be embedded within a digital video,** in various embodiments, and the token may include a variety of encrypted information. **Para [0015] By embedding a digital token within a video, the video may be sent to one or more users on the Internet along with the token.** A recipient of the video may not only be able to view the video, but can take additional action or gain additional functionality from the digital token that is embedded within the video. **Para [0016] Tokens can be embedded by altering video metadata in some embodiments. Metadata may be altered so that the video itself is not changed, but associated data with the video is changed to reflect a created token. In other embodiments, video and/or audio data of the video itself can be modified to reflect a created token.** This may be advantageous in some cases where a video may be shared on a different platform that may strip some or all metadata from the video—by embedding the token within the video/audio data itself, the token cannot (or cannot easily) be stripped. **In yet further embodiments, a token can be embedded in a video such that the resulting video is noticeably visually altered. For example, a filter could be put on all or a portion of the video (black/white,**

	sharpening, softening, color-altering, or any of a number of visual image filters). A watermark (e.g. a particular logo) can also be put on all or a portion of a video. **Para [0040]** The encrypting performed in operation 420 may be done using a public-private key pair. **In one embodiment, for example, operation 420 may include image processing system 120 encrypting a digital token with the public key in a public-private key pair (preventing the encrypted digital token from being decrypted by someone who does not have the corresponding private key, which may be kept secret by image processing system 120).** **Note:** *The mapped citation describes embedding of digital tokens within a digital video (i.e., programmable video assembly) and the system encrypts the token with a public key, ensuring that only the holder of the corresponding private key can decrypt it, thereby preventing unauthorized access.*
1.1 a virtualization engine, wherein the engine is configured to convert a rendered video file into a reference container by extracting metadata and removing media sample data;	**Para [0090] Video file 710 may include video picture data 720, audio data 730, and metadata 740. Video picture data 720 can include all information needed to render and play a series of video frames on a viewing device.** Audio data 730 can likewise include all information needed to play audio accompanying a video file. **Para [0091] Metadata 740 can include various video related data that is not necessarily needed to be able to render the video and/or play its audio. Metadata 740 may contain GPS coordinates or other location information corresponding to a place associated with a video (e.g. a real world location where a video was shot). Metadata 740 may include a date and/or time that a video was taken.** Metadata 740 can also include other data in various embodiments—for example, identities of people in the video, as may correspond to social media websites such as Facebook™. **Para [0088] Further, note that digital tokens are described in various locations as being encrypted, and/or as being embedded in an image. A digital token does not have to be encrypted to be embedded in an image, but encrypting the digital token prior to embedding may enhance security.** Additionally, an "encrypted" digital token may refer to a digital token that is wholly or partially encrypted, unless otherwise indicated. **Further, as noted above, a digital token can be embedded in an image, in some embodiments, by storing image data (e.g. a hash or other uniquely identifying data) in a central repository (e.g. a database associated with image processing system 120) without any need to actually modify data in the image (such as pixel data or metadata).** **[Para 0105] Processing system 820 may maintain information for a large number of different videos with embedded digital tokens,** each of which may be redeemable for an amount of digital currency. In order to be able to track **the digital tokens (and to know how much money is associated with a particular token),** processing system 820 therefore can use unique IDs for each token to access information associated with that token (e.g., monetary amount, creator of token, etc.). **[Para 0108]** Altering data in a video to include an encrypted digital token (e.g. representing an amount of digital currency) therefore can be performed in different manners. In some cases, **altering video metadata 740 may result in a completely unchanged digital video**. **Note:** *The mapped citation describes the system to render and play a series of video frames on a viewing device and also extracts the metadata associated with a video. And digital tokens are described in various locations as being encrypted, which means that there is a rendering of video file resulting in the extraction of the metadata associated with it. The system selectively operates on metadata while leaving the media samples untouched.*
1.1.1 wherein the virtual	**N/A**

video container retains ISO Base Media File Format box structures while being devoid of media sample data.	
1.1.2 wherein the step of resolving referenced media data involves issuing MDAT range requests to the remote media data sources.	**N/A**
1.1.3 further comprising an artificial intelligence orchestration layer configured to modify the sequencing instructions of the reference container without accessing raw media samples.	**NA**
1.2 a token issuance service, wherein the service configured to generate a cryptographicvideo token binding transaction, consent, and licensing information;	**Para [0014] The present specification allows for the embedding of digital tokens within a digital video.** These tokens may be cryptographically obscured using an encryption key, such as a public key of a public/private key pair. **An encrypted digital token may therefore be embedded within a digital video,** in various embodiments, and the token may include a variety of encrypted information. **Para [0015] By embedding a digital token within a video, the video may be sent to one or more users on the Internet along with the token.** A recipient of the video may not only be able to view the video, but can take additional action or gain additional functionality from the digital token that is embedded within the video. **Para [0016] Tokens can be embedded by altering video metadata in some embodiments. Metadata may be altered so that the video itself is not changed, but associated data with the video is changed to reflect a created token. In other embodiments, video and/or audio data of the video itself can be modified to reflect a created token.** This may be advantageous in some cases where a video may be shared on a different platform that may strip some or all metadata from the video—by embedding the token within the video/audio data itself, the token cannot (or cannot easily) be stripped. In yet further embodiments, a token can be embedded in a video such that the resulting video is noticeably visually altered. For example, a filter could be put on all or a portion of the video (black/white, sharpening, softening, color-altering, or any of a number of visual image filters). A watermark (e.g. a particular logo) can also be put on all or a portion of a video. **[Para 0098] The embedded digital token is a payment token**, in various embodiments, corresponding to a desire of a user to have a video itself be able to serve as a representation of money. A user can use an application on a smartphone, such as the PayPal™ app or the Venmo™ app (or another app) to make the request to generate a video with an embedded digital token (e.g., a payment token). **[Para 0093] Processing system 820 may be under the control of an electronic transaction service provider (who may also control transaction system 160), and can take various operations related to embedded digital tokens in video files**. **Note:** *The mapped citation describes how digital tokens can be embedded within a video, either by altering metadata or modifying audiovisual data, and cryptographically obscured using an encryption key such as a public key in a*

	public/private key pair. This allows the video to be distributed to users along with the embedded token, enabling controlled access and potential redemption. Such an issuance process aligns with the concept of a token issuance service that generates cryptographic video tokens. However, consent, and licensing information within the generated token is not explicitly disclosed.
1.2.1 wherein the token issuance service includes a unique session nonce in each generated token to prevent unauthorized resolution reuse.	**Para [0039] The information identifying the amount of digital currency, in some embodiments, can include a unique identifier created by image processing system 120. Image processing system 120 may maintain information for a large number of different images with embedded digital tokens, each of which may be redeemable for an amount of digital currency. In order to be able to track the digital tokens (and to know how much money is associated with a particular token), image processing system 120 therefore can use unique IDs for each token to access information associated with that token (e.g., monetary amount, creator of token, etc.)**. Note that various attribute information/metadata may be associated with a digital token that is embedded in an image. This information can be included at the time the digital token is created, or can be modified at a later time in some embodiments. **Para [0040]** The encrypting performed in operation 420 may be done using a public-private key pair. **In one embodiment, for example, operation 420 may include image processing system 120 encrypting a digital token with the public key in a public-private key pair (preventing the encrypted digital token from being decrypted by someone who does not have the corresponding private key, which may be kept secret by image processing system 120).** In another embodiment, operation 420 may also include using the private key of a public-private key pair to add signature information into an image. For example, a private key for Venmo™ could be used to encrypt information saying "This image includes $5.00 in digital currency redeemable by Venmo™!" (or similar). Anyone with Venmo's public key could decrypt the signature information and would then know that Venmo legitimately authorized placement of a currency-bearing digital token within an image, in this example. **Note:** *The mapped citation describes how the image processing system generates unique identifiers for each digital token, enabling tracking of the associated currency amount and token creator. In one embodiment, the system encrypts the token with a public key, ensuring that only the holder of the corresponding private key can decrypt it, thereby preventing unauthorized access. This aligns with the concept of incorporating unique session nonce in each generated token to prevent unauthorized.*
1.2.2 wherein the transaction metadata includes instructions for micro payments triggered by individual segment resolution events	**Para [0098] The embedded digital token is a payment token, in various embodiments, corresponding to a desire of a user to have a video itself be able to serve as a representation of money. A user can use an application on a smartphone, such as the PayPal™ app or the Venmo™ app (or another app) to make the request to generate a video with an embedded digital token (e.g., a payment token).** Such requests can of course originate from another computing device, such as a desktop or laptop computer, a wearable computing device, etc. A user may therefore take one or more actions on a computing device to cause the request to be received at processing system 820. **Para [0109]** Operation 930 may include **embedding a digital token within a single frame of a video, or within multiple frames of the video.** In some instances, "frames" may not be explicitly delineated by the video encoding algorithm **and the token may simply be embedded within one or more various portions audiovisual data 735. The token can even be included (or made to appear) in multiple locations—for example, the audiovisual data could be altered to show a visual indicator of the token at the 5 minute mark, 10 minute mark, and 25 minute mark.**

	Thus, altering the video for the embedded digital token can cause a perceptible visual shift in an appearance of the video. Audio content may also be altered to reflect the presence of the digital token (e.g. a noise or speech could be included at the 30 second and 60 second mark of a video). In some cases, the altered content itself may be used to encode the actual digital token (e.g. the modified bits of the audiovisual content that give rise to the perceptible change in the video actually contain all or a portion the digital token itself). In other cases, the alteration to the content may not contain the digital token (e.g. it could be stored in metadata), but the alteration to the media content can help alert a user to the presence of the digital token. **Note:** *The mapped citation describes how the embedded digital token can serve as a payment token, enabling a video to act as a representation of money. The token may be embedded within one or multiple frames of the video, allows the video to carry redeemable value at different segments, conceptually aligning with the notion of transaction metadata tied to resolution events. However, micro payments triggered by individual segment resolution events are explicitly not disclosed, nor does it specify how such metadata would govern or automate payment execution at the segment level*
1.2.3 wherein the cryptographic video token may function as an integrated authorization gatekeeper that operates independently of the media payload by excluding media sample data and direct file paths, enforces the right to resolve by providing the necessary cryptographic permissions and binds governance logic to the resolution event.	**Para [0014]** The present specification allows for the embedding of digital tokens within a digital video. **These tokens may be cryptographically obscured using an encryption key, such as a public key of a public/private key pair.** An encrypted digital token may therefore be embedded within a digital video, in various embodiments, and the token may include a variety of encrypted information. **Para [0074]** Images with embedded tokens (e.g. images that have been altered to include data representing a digital token or images that have had data sampled and stored of a hash of the image itself, or another way to uniquely identify the image in association with a digital token) can be shared on various social media platforms in different embodiments. A user requesting creation of a digital token n for an image can receive an image that allows the requesting user to post the image on a platform such as Instagram™, FaceBook™, or any other social media service. Another user(s) on these services can redeem a posted image for currency or another reward. Other users on these services can also transmit a posted image to yet further users, particularly in some embodiments where an image can be redeemed multiple times**. For example, a digital token may be multi-use, such that a first recipient could redeem it once (or possibly more than once, up to a limit). A user of a social media platform for example might be able to both redeem an image for an amount of currency, and then re-post a multi-use digital token embedded image so that other users (such as friends of friends) could also use the digital token for currency or another reward.** **Para [0128] With external verification factor (a condition such as sharing the video to other users on a social media platform), an outside operator may have to verify that the external condition has been met. So, processing system 820 might receive a digital token redemption request and then have to see if the external condition is met.** If a video is shared on a social media platform, for example, the platform operator could provide a cryptographically signed verification that the user has met the condition (e.g. Facebook™ could provide information stating that the user had met the condition for sharing the video). Of course, many different types of external redemption conditions are possible, and may vary by embodiment. **Note:** *The mapped citation describes the digital token can be multi-use, allowing redemption by multiple users across social platforms without exposing the underlying media itself. This mechanism enforces the right to resolve by requiring cryptographic permissions for each redemption, while simultaneously binding governance logic to*

	the resolution event and ensuring that authorization and control remain tied to the token rather than the media distribution channel. However, the citation does not disclose the explicit exclusion of media sample data and direct file paths, nor does it detail the binding of governance logic to the resolution event.
1.3 a secure resolution service, wherein the service dereferences remote media data upon successful validation of the cryptographic video token; further comprising logging each individual media dereference event to an auditable transaction layer.	**Para [0014]** The present specification allows for the embedding of digital tokens within a digital video. **These tokens may be cryptographically obscured using an encryption key, such as a public key of a public/private key pair.** An encrypted digital token may therefore be embedded within a digital video, in various embodiments, and the token may include a variety of encrypted information. **Para [0101]** In some embodiments, the device accessing the digital token in operation 920 did not create the token itself, however—for example, user device 105 or server 805 may request and receive the digital token from processing system 820. **Thus, a computer system executing method 900 may transmit a token request to a remote system (e.g. from server 805 to processing system 820) and receive, from the remote system responsive to the token request, the encrypted digital token.** **Para [0118] In operation 960, processing system 820 decrypts an encrypted digital token from an altered video using a second encryption key, according to some embodiments. As noted above, the altered video can be transmitted to processing system 820 for decryption by any user in possession of the altered video (or a URL or some other form of location pointer),** in various embodiments. **Operation 960 therefore can include using a second encryption key (such as a private encryption key in a public-private key pair) corresponding to a first encryption key (e.g. the public key) in order to perform decryption. After decryption, the embedded digital token can then be checked for validity and/or redeemed for currency**, for example. **Note:** *The mapped citation describes a secure resolution service, wherein the system transmits a token request to a remote processing system and receives an encrypted digital token in response. In particular, operation highlights that the processing system decrypts the encrypted digital token from an altered video using a second encryption key (such as a private key in a public–private key pair). Once decrypted, the embedded digital token is validated, enabling dereferencing of remote media data only upon successful cryptographic verification. However, auditable transaction layer is not disclosed.*
1.3.1 wherein the secure resolution service is restricted to operating within a hardware-attested trusted execution environment.	**Claim 6** The computer system of claim 1, wherein **the computer system comprises a mobile phone device, a desktop computer system, or a laptop computer system.** **Para [0099] In some embodiments, a user can select the video in which he or she wishes to embed a digital token. Thus, a user of a smartphone may be able to browse through a library of captured videos on the phone to select one. In other embodiments, an app on a smartphone (or other device) may actually prompt the user to take a new video by causing a camera application on the smartphone to be opened up.** Accordingly, in one embodiment, the request in operation 810 is received from an application running on a mobile phone device. A video uploaded by the user may also corresponds to a video originally created using a camera of a mobile phone device (or may be a video otherwise acquired by the user, on any device, in various embodiments). In some cases, server 805 may make the request to have a video generated with an embedded digital token as part of an advertisement or social media action. E.g., a business, promoter, etc., may request that the video be created. **Note:** *The mapped citation describes the video for embedding token is being performed on a smartphone (or other device), which means that the process is being restricted to be operated in the hardware-attested trusted execution environment as*

	smartphones are widely considered hardware-attested Trusted Execution Environments (TEEs), particularly in the context of tokenized content delivery (DRM) and secure.
1.3.2 wherein the logged dereference events trigger a commercial action selected from a group consisting of billing, royalty allocation, and advertising insertion.	**Para [0099]** In some embodiments, a user can select the video in which he or she wishes to embed a digital token. Thus, a user of a smartphone may be able to browse through a library of captured videos on the phone to select one. In other embodiments, an app on a smartphone (or other device) may actually prompt the user to take a new video by causing a camera application on the smartphone to be opened up. Accordingly, in one embodiment, the request in operation 810 is received from an application running on a mobile phone device. A video uploaded by the user may also corresponds to a video originally created using a camera of a mobile phone device (or may be a video otherwise acquired by the user, on any device, in various embodiments). **In some cases, server 805 may make the request to have a video generated with an embedded digital token as part of an advertisement or social media action. E.g., a business, promoter, etc., may request that the video be created.** **Para [0114]** Note that transmission, in operation 940, does not necessarily including transmitting an entire video file in some embodiments. **The transmission can explicitly include sending a URL or other selectable download link to allow the receiving user to download/view the altered video containing the digital token.** **Para [0115]** In operation 950, processing system 820 accesses an encrypted digital token from an altered video, according to some embodiments. **This may include processing system 820 receiving the altered video including an encrypted digital token. The altered video may be received from a second user (e.g. a user to which the video was previously transmitted) or any other user, in various embodiments. Thus, video with an embedded digital token representing an amount of money can be sent to one or more various users, and then received again (back) at processing system 820.** After this later receipt, processing system 820 can determine if there is a valid token can should be redeemed for currency, as further discussed below. **Note:** *The mapped citation describes the altered video with embedded token can be a part of advertisement or social media action or may represent an amount of money to be exchanged. However, logged dereference events triggering billing or royalty allocation is not as such disclosed.*
1.4 a playback environment, wherein the playback environment stores multiple tokens and dynamically assembling video streams in real-time from authorized references.	**Para [0117]** In some cases, **various user software (e.g. on device 105) may recognize the presence of a digital payment token in some embodiments, and allow a user to take action to redeem it. For example, if a user is viewing a video with an embedded digital token, the viewing application could generate a graphical control interface (such as a pop-up button) allowing the user to redeem a digital token for currency (or another reward).** Thus, a media viewer within a web browser or other application could alert the user when a digital token-bearing video is being viewed. **The alert may be synchronized to one or more particular moments during playback of the video content (e.g. for one or more particular frames or times). These times may be specified in a user request for creating an altered video with an embedded digital token.** **Para [0118]** In operation 960, processing system 820 decrypts an encrypted digital token from an altered video using a second encryption key, according to some embodiments. **As noted above, the altered video can be transmitted to processing system 820 for decryption by any user in possession of the altered video (or a URL or some other form of location pointer),** in various embodiments. **Note:** *The mapped citation describes the playback environment where the tokens are being embedded in an altered video stream, which means that the multiple token are*

	being stored in the video stream when the user is viewing the video (i.e., in real time) and altered video is transmitted to processing system for decryption by any user in possession of the altered video or any URL (i.e., authorized references).
1.4.1 wherein the playback environment is configured to manage independent tokens from multiple content owners within a single session	**Para [0074] Images with embedded tokens (e.g. images that have been altered to include data representing a digital token or images that have had data sampled and stored of a hash of the image itself, or another way to uniquely identify the image in association with a digital token) can be shared on various social media platforms in different embodiments.** A user requesting creation of a digital token n for an image can receive an image that allows the requesting user to post the image on a platform such as Instagram™, FaceBook™, or any other social media service. **Another user(s) on these services can redeem a posted image for currency or another reward.** Other users on these services can also transmit a posted image to yet further users, particularly in some embodiments where an image can be redeemed multiple times. For example, a digital token may be multi-use, such that a first recipient could redeem it once (or possibly more than once, up to a limit). **A user of a social media platform for example might be able to both redeem an image for an amount of currency, and then re-post a multi-use digital token embedded image so that other users (such as friends of friends) could also use the digital token for currency or another reward.** **Para [0075]** Thus, in various embodiments, a digital token can be created by image processing system 120 (or another system) that allows for multiple redemption. **A token can be created such that it can redeemed 4×$20, for example, with only a unique user allowed to redeem it once (values can be adjusted, of course). When a token is created, the creating user can specify other users and/or groups of users who are allowed to redeem—thus, a given token, which may be embedded in a digital image, can be redeemed multiple times but not necessarily by the same user.** **[Para 0117]** Thus, a media viewer within a web browser or other application could alert the user when a digital token-bearing video is being viewed. The alert may be synchronized to one or more particular moments during **playback of the video content (e.g. for one or more particular frames or times**). These times may be specified in a user request for creating an altered video with an embedded digital token. **Note:** *The mapped citation describes the tokens within the video that has been sampled and stored to be shared. When the token is created it can be redeemed by a unique user for once, or the creating user can specify other users who are allowed to redeem the token, which means that the individual tokens (multiple) are being managed by other specified users. However, independent token from multiple content owners within a single session is not disclosed.*
1.4.2 wherein the playback environment may function as a content wallet capable of managing multiple tokens from independent content owners.	**Para [0117]** In some cases, **various user software (e.g. on device 105) may recognize the presence of a digital payment token in some embodiments, and allow a user to take action to redeem it.** For example, if a user is viewing a video with an embedded digital token, the viewing application could generate a graphical control interface (such as a pop-up button) allowing the user to redeem a digital token for currency (or another reward). Thus, a media viewer within a web browser or other application could alert the user when a digital token-bearing video is being viewed. **The alert may be synchronized to one or more particular moments during playback of the video content (e.g. for one or more particular frames or times). These times may be specified in a user request for creating an altered video with an embedded digital token.** **Para [0075]** Thus, in various embodiments, a digital token can be created by image

	processing system 120 (or another system) that allows for multiple redemption. **A token can be created such that it can redeemed 4×$20, for example, with only a unique user allowed to redeem it once (values can be adjusted, of course). When a token is created, the creating user can specify other users and/or groups of users who are allowed to redeem—thus, a given token, which may be embedded in a digital image, can be redeemed multiple times but not necessarily by the same user.** **Note:** *The mapped citation describes software (say wallet) to manage digital payment token. When the token is created it can be redeemed by a unique user for once, or the creating user can specify other users who are allowed to redeem the token, which means that the individual tokens (multiple) are being managed by other specified users.*
1.4.3 wherein the system is configured to terminate the video stream instantly upon the revocation of user consent parameters associated with the token.	**[Para 0049]** In operation 470, in one embodiment, **image processing system 120 verifies the validity of a decrypted digital token (e.g. from operation 460)**. **Verifying validity can include extracting a unique identifier from the decrypted digital token, and then querying a database with the unique identifier**. The database may include a list of all digital tokens created by image processing system 120, **and a record for the unique identifier can be referenced in the database. This record can contain an indicator of whether or not a given token is valid or invalid, for example**.

2.Additional Citations

S No.	Publication no.	Title	Filing Date	Publication Date	Assignee(s)
1	**US9344606B2**	System and method for compiling and playing a multi-channel video	23 JAN 2013	17 MAY 2016	Radical Urban LLC
2	**US10567489B2**	System and method for seamless switching between data streams	15 MAR 2013	18 FEB 2020	Time Warner Cable Enterprises LLC
3	**US20240213345A1**	Realtime wireless synchronization of live event audio stream with a video recording	12 JAN 2024	27 JUN 2024	Bygge Technologies Inc
4	**US9532005B2**	Methods and apparatus for persistent control and protection of content	13 AUG 2013	27 DEC 2016	Intertrust Technologies Corp
5	**US10878076B2**	Receiving apparatus, transmitting apparatus, and data processing method	03 AUG 2016	29 DEC 2020	Saturn Licensing LLC
6	**US20170055046A1**	Broadcast signal transmitting/receiving method and device	21 MAY 2015	23 FEB 2017	LG Electronics Inc

3.Non-patent Citations

S No.	Title	Authors	Publication Date	Affiliations	Source Link
1	Token-Based URLs & JWT based Authentication Example for Video Protection | What & How	Vishal Sharma	28 AUG 2025	VdoCipher	Link
2	VITA: Video Instance Segmentation via Object Token Association	Miran Heo, Sukjun Hwang, Seoung Wug Oh, Joon-Young Lee, Seon Joo Kim	20 OCT 2022	Yonsei University	Link
3	Video Detection and Segmentation with Language Modeling	Anna Kazanets, Tatiana Shorstova, Khalid Hilmi, Maud Marques, Michael Witcher	04 DEC 2025	University of Alberta	Link

4. Patent Search Strings

S No.	Patseer Search Strings
1	TACD:((((VIDEO OR MP4 OR 4K) W5 (SEGMENTS OR SEGMENTED OR PIECES)) WS (TOKEN* OR PASSWORD OR PINCODE OR PASSCODE)) WS (ACCESS* OR PLAYBACK OR PLAY OR VIEW*)) AND TACD:((TOKEN* OR PASSWORD OR PINCODE OR PASSCODE) WS (ONE WD2 TIME))
2	AC:(H04N21/4341 AND H04N21/4307) AND TACD:((TOKEN OR PASSWORD OR (PIN W2 CODE) OR PASSCODE) WS (ACCESS* OR VIEW* OR DISPLAY* OR PLAYING))
3	AC:(H04N21/43072) AND TACD:(((VIDEO OR MP4 OR 4K) WS (SEGMENTS OR SEGMENTED OR PIECES)) AND ((SEGMENTS OR SEGMENTED OR PIECES) WS (TOKEN* OR PASSWORD OR PINCODE OR PASSCODE OR CRYPT*)))
4	TAC:(((TOKENIZATION OR SEGMENTATION OR PARSING OR ENCODING OR DISCRETIZATION) W3 (SAMPLES OR TIMESTAMP)) WS (TOKEN* OR PASSWORD OR PINCODE OR PASSCODE)) AND TACD: (ACCESS* OR PLAYBACK OR PLAY OR VIEW*) AND TACD:((TOKEN* OR PASSWORD OR PINCODE OR PASSCODE) WS (ONE W2 TIME))
5	TACD:(TOKEN OR KEY OR PIN OR PASSCODE OR PASSWORD) AND TACD:(METADATA W2 (EXTRACT* OR STRUCTUR*)) AND TACD:(PLAYBACK OR PLAY OR VIEW) AND AC:(H04L9/0891 OR H04L9/3226 OR G06V20/49 OR G06F21/1085)
6	TACD:((RENDER* OR PROCESSING) W5 (GRAPHICS OR VIDEO)) AND TACD:(VIDEO W5 (TOKEN* OR SEGMENT* OR PIECES OR SAMPLES)) AND TACD:(METADATA W2 (EXTRACT* OR STRUCTUR*)) AND TACD:(TOKEN OR KEY OR PIN OR PASSCODE OR PASSWORD) AND AC:(H04L9/0891 OR H04L9/3226 OR G06V20/49 OR G06F21/1085)
7	TAC:((VIDEO OR CONTENT OR MEDIA OR SONG* OR MOVIE*) W5 (DELIVER* OR STREAM* OR TRANSMIT* OR BROADCAST*)) AND TACD:(((VIRTUA* W5 (ENGINE* OR PROCESSOR*)) OR (VIRTUALIZER) OR (REFERENCE W3 GENERATOR*) OR (LINK* W3 GENERATOR)) AND ((TOKEN* OR KEY* OR PASSCODE* OR TICKET*) WS (ACCESS* OR DISTRIBUT* OR VALID* OR VERIF*)) AND (PLAYBACK OR (MEDIA W2 (PLAYER OR APPLICATION))) AND ((VIDEO OR MEDIA OR SONG* OR MOVIE*) WS (SEGMENT* OR CHUNK*)))
8	TAC:((MULTIMEDIA OR AUDIO OR VIDEO OR CONTENT OR MEDIA OR SONG* OR MOVIE*) W5 (DELIVER* OR STREAM* OR TRANSMIT* OR BROADCAST* OR TRANSFER OR PLAY*)) AND TACD:(((VIDEO OR SONG OR MOVIE OR MEDIA) W3 ((LINK* OR REFERENCE* OR METADATA OR INDEX* OR BLUEPRINT* OR SUMMAR* OR REFERENCE* OR POINTER*) WS (GENERAT* OR EXTRACT* OR DERIV* OR DESCRIB* OR PARS* OR DEFINE))) AND (SAMPLE* OR SEGMENT* OR CLIP* OR PORTION*) AND (TOKEN* OR TICKET* OR CODES OR PASSCODE* OR KEYS) AND (LICENSE OR CERTIFICAT* OR IDENTITY)) AND AC:(H04N21/27)
9	AC:((H04N21/8456 OR H04N21/23406 OR H04L65/60) AND (H04N21/2541)) AND TACD:(((VIDEO OR SONG OR MOVIE OR MEDIA) W3 ((LINK* OR REFERENCE* OR METADATA OR INDEX* OR BLUEPRINT* OR SUMMAR* OR REFERENCE* OR POINTER*) WS (GENERAT* OR EXTRACT* OR DERIV*))) AND (TOKEN* OR TICKET* OR CODES OR PASSCODE* OR KEYS) AND (PLAYBACK OR DECODER*) AND (SESSION))
10	TAC:(((MEDIA OR VIDEO OR MULTIMEDIA OR AUDIOVISUAL OR AUDIO OR MOVIE) WS (DELIVERY OR PLAY* OR STREAM* OR PRESENT* OR BROADCAST))) AND TACD:(((REMOV* OR DELET* OR DISCARD) WS ((ORIGINAL OR ACTUAL) W3 (FILE OR DATA OR SAMPLE))) AND ((LINK* OR REFERENCE* OR METADATA OR INDEX* OR BLUEPRINT* OR SUMMAR* OR REFERENCE* OR POINTER*) WS (GENERAT* OR EXTRACT* OR DERIV* OR SEPARAT*)) AND (TOKEN* OR TICKET* OR CODES OR PASSCODE* OR KEYS) AND (PLAYBACK OR DECODER*) AND (TRANSACTION* OR

	PAYMENT OR SUBSCRIPTION)) AND AC:(H04N21/8547 OR H04N21/8456)
11	TAC:(((VIDEO_ON_DEMAND) OR VOD OR (LIVE W2 VIDEO) OR (LIVE W2 STREAM*))) AND TACD:(((REMOV* OR DELET* OR DISCARD) WS ((ORIGINAL OR ACTUAL) W3 (FILE OR DATA OR SAMPLE))) AND ((LINK* OR REFERENCE* OR METADATA OR INDEX* OR BLUEPRINT* OR SUMMAR* OR REFERENCE* OR POINTER*) WS (GENERAT* OR EXTRACT* OR DERIV* OR SEPARAT*)) AND (TOKEN* OR TICKET* OR CODES OR PASSCODE* OR KEYS) AND (PLAYBACK OR DECODER*) AND (TRANSACTION* OR PAYMENT OR SUBSCRIPTION) AND ((MEDIA OR VIDEO OR MULTIMEDIA OR AUDIOVISUAL OR AUDIO OR MOVIE) WS (DELIVERY OR PLAY* OR STREAM* OR PRESENT* OR BROADCAST)))
12	AC:(H04N21/8586 AND (H04N21/835 OR H04N21/2541)) AND TACD:(((VIDEO OR SONG OR MOVIE OR MEDIA) W3 ((LINK* OR REFERENCE* OR METADATA OR INDEX* OR BLUEPRINT* OR SUMMAR* OR REFERENCE* OR POINTER*) WS (GENERAT* OR EXTRACT* OR DERIV*))) AND (TOKEN* OR TICKET* OR CODES OR PASSCODE* OR KEYS) AND (PLAYBACK OR DECODER*) AND (SESSION))
13	TAC:(((VIDEO_ON_DEMAND) OR VOD OR (LIVE W2 VIDEO) OR (LIVE W2 STREAM*))) AND TACD:(((LINK* OR REFERENCE* OR METADATA OR INDEX* OR BLUEPRINT* OR SUMMAR* OR REFERENCE* OR POINTER*) WS (GENERAT* OR EXTRACT* OR DERIV* OR SEPARAT*)) AND ((MULTIPLE OR PLURALITY OR NUMBER OR DIFFERENT OR SEPARATE) W5 (TOKEN* OR TICKET* OR CODES OR PASSCODE* OR KEYS)) AND (PLAYBACK OR DECODER*) AND (TRANSACTION* OR PAYMENT OR SUBSCRIPTION OR LICENSE))
14	INV:((ANTHONY SHARICK) OR (BRADEN CHRISTOPHER ERICSON) OR (BRETT RONALD WELCH) OR (CATHERINE LIN-HENDEL) OR (CHRISTOPHER ALBERT HARTLEY) OR (FINBAR O'HANLON) OR (FUN-CHEN JOU) OR (GARCIA BARRIO LAURA) OR (GREGORY SIMON) OR (HIDEKI FUKUDA) OR (HIDESHI ISHIHARA) OR (HIROMU KITAURA) OR (ISMAIL R. HARITAOGLU) OR (JEFFREY A. AITKEN) OR (JIEYI LONG) OR (JOHN D. RINALDO) OR (JONATHAN YANTIS) OR (KAILI AN) OR (KEVIN J. MA) OR (LUKASZ JAKUB SLIWKA) OR (LYONS GRANT) OR (MAHMOUD ELBARKY) OR (MARK A. MALAMUD) OR (MITCHELL C. LIU) OR (MITSUAKI OSHIMA) OR (MORRIS MARC ANTHONY) OR (MOTZEL THOMAS) OR (NEIL C. MARCK) OR (OZTAN HARMANCI) OR (PAUL TWEEDALE) OR (PERKES RONALD) OR (ROBERT E. SEASTROM) OR (ROBERT HICKEY) OR (ROBERT W. LORD) OR (ROYCE A. LEVIEN) OR (RYAN STEELBERG) OR (SAM POURCYROUS) OR (SEAN DENNIS) OR (SHANNON CODE) OR (TAKAOKA TOMOHISA) OR (VINCENT E. COLELLA) OR (WESLEY E. GEORGE) OR (WILLIAM EDWARD QUIGLEY)) AND AC:(H04N21/8456)
15	INV:((ANTHONY SHARICK) OR (BRADEN CHRISTOPHER ERICSON) OR (BRETT RONALD WELCH) OR (CATHERINE LIN-HENDEL) OR (CHRISTOPHER ALBERT HARTLEY) OR (FINBAR O'HANLON) OR (FUN-CHEN JOU) OR (GARCIA BARRIO LAURA) OR (GREGORY SIMON) OR (HIDEKI FUKUDA) OR (HIDESHI ISHIHARA) OR (HIROMU KITAURA) OR (ISMAIL R. HARITAOGLU) OR (JEFFREY A. AITKEN) OR (JIEYI LONG) OR (JOHN D. RINALDO) OR (JONATHAN YANTIS) OR (KAILI AN) OR (KEVIN J. MA) OR (LUKASZ JAKUB SLIWKA) OR (LYONS GRANT) OR (MAHMOUD ELBARKY) OR (MARK A. MALAMUD) OR (MITCHELL C. LIU) OR (MITSUAKI OSHIMA) OR (MORRIS MARC ANTHONY) OR (MOTZEL THOMAS) OR (NEIL C. MARCK) OR (OZTAN HARMANCI) OR (PAUL TWEEDALE) OR (PERKES RONALD) OR (ROBERT E. SEASTROM) OR (ROBERT HICKEY) OR (ROBERT W. LORD) OR (ROYCE A. LEVIEN) OR (RYAN STEELBERG) OR (SAM POURCYROUS) OR (SEAN DENNIS) OR (SHANNON CODE) OR (TAKAOKA TOMOHISA) OR (VINCENT E. COLELLA) OR (WESLEY E. GEORGE) OR (WILLIAM EDWARD QUIGLEY)) AND AC:(H04N21/2368)
16	AASN:((ACTV) OR (ALPHABET) OR (AMBROSIA SOFTWARE) OR (APPLE) OR (BROADCOM) OR (BYGGE TECHNOLOGY) OR (CHARTER COMMUNICATION) OR (DISH NETWORK) OR (ECHOSTAR) OR (ERICSSON) OR (FRISKIT) OR (GOOGLE) OR (INTERDIGITAL) OR (INTERTRUST TECH) OR (INTERTRUST TECHNOLOGIES) OR (INVENTION SCIENCE FUND) OR (ION VIDEO) OR (LINIUS) OR (LOYYAL HOLDINGS) OR (NTT DOCOMO) OR (OPENTV) OR (PANASONIC) OR (PAYPAL) OR (QWEST COMMUNICATIONS INTERNATIONAL) OR (RADICAL URBAN) OR (REZZONATION) OR (SEARETE) OR

	(SNAP) OR (SONY) OR (SRSLY) OR (THE INVENTION SCIENCE FUND) OR (THETA LABS) OR (TIME WARNER CABLE ENTERPRISES) OR (TUNESPOTTER) OR (VANTIVA) OR (VERONA) OR (VIVCOM) OR (WEBTV NETWORKS)) AND AC:(H04N21/2368 AND H04N21/6334)
17	AASN:((ACTV) OR (ALPHABET) OR (AMBROSIA SOFTWARE) OR (APPLE) OR (BROADCOM) OR (BYGGE TECHNOLOGY) OR (CHARTER COMMUNICATION) OR (DISH NETWORK) OR (ECHOSTAR) OR (ERICSSON) OR (FRISKIT) OR (GOOGLE) OR (INTERDIGITAL) OR (INTERTRUST TECH) OR (INTERTRUST TECHNOLOGIES) OR (INVENTION SCIENCE FUND) OR (ION VIDEO) OR (LINIUS) OR (LOYYAL HOLDINGS) OR (NTT DOCOMO) OR (OPENTV) OR (PANASONIC) OR (PAYPAL) OR (QWEST COMMUNICATIONS INTERNATIONAL) OR (RADICAL URBAN) OR (REZZONATION) OR (SEARETE) OR (SNAP) OR (SONY) OR (SRSLY) OR (THE INVENTION SCIENCE FUND) OR (THETA LABS) OR (TIME WARNER CABLE ENTERPRISES) OR (TUNESPOTTER) OR (VANTIVA) OR (VERONA) OR (VIVCOM) OR (WEBTV NETWORKS)) AND AC:(H04N21/6334)
S No.	**USPTO Search Strings**
1	((VIDEO) ADJ3 (PLAYBACK STREAM OR RENDER)) AND (TOKEN OR "ENCRYPTION KEY") AND (METADATA OR ARRANGEMENT OR STRUCTURE) AND (SECURE OR PROTECTED OR AUTHORISED) AND (LINK) AND ("REMOTE STORAGE" OR CACHE OR STORAGE OR "RAM") AND (CRYPTOGRAPHIC) AND (SAMPLE OR SEGMENT) AND H04L9/0891.CPC.
2	(VIDEO ADJ/3 (STREAMING OR PLAYBACK OR RENDER)) AND (TOKEN OR "ENCRYPTION KEY") AND ("REMOTE STORAGE" OR CACHE OR "RAM") AND (TOKEN NEAR3 (VALIDATION OR AUTHENTICATION)) AND (METADATA) AND ("ARTIFICIAL INTELLIGENCE" OR "AI" OR "LLM") AND ((UNIQUE OR SPECIFIC) ADJ3 (TOKEN))
S No.	**J-PlatPat Search Strings**
1	[VIDEO,3C,DELIVERY/TX]+[MEDIA,3C,DELIVERY/TX]*[TOKENIZATION/TX]*[5C164UD63/FT]+[5C164UD64/FT]
2	[VIDEO,3C,DELIVERY/AB]*[TOKENIZATION/AB]*[VIDEO,3C,SEGMENTATION/TX]*[SAMPLES/TX]+[TOKENS/TX]+[FRAGMENTS/TX]*[5C164UD63/FT]
S No.	**KIPRIS Search Strings**
1	(VIDEO^3(TOKENIZATION+SEGMENTATION+DISTRIBUTION))*(TOKEN+KEY+PIN+PASSCODE+PASSWORD)
2	(VIDEO^3(TOKENIZATION+SEGMENTATION+DISTRIBUTION))*(TOKEN+KEY)*(ENCRYPT+DECRYPT)
S No.	**FPO Search Strings**
1	SPEC/(("VIDEO SAMPLES"~3) OR ("VIDEO SEGMENTS"~3) OR ("VIDEO CHUNKS"~2) OR ("MEDIA SAMPLES"~3) OR ("MEDIA SEGMENTS"~3) OR ("MEDIA CHUNKS"~2)) AND (TOKEN*) AND (("REFERENCE DATA"~3) OR ("REFERENCE CONTAINER"~3) OR (METADATA)) AND (PLAYBACK)
2	SPEC/(("CONTENT DELIVERY"~3) OR ("VIDEO DELIVERY"~3) OR ("MEDIA STREAMS"~3)) AND (("VIDEO SAMPLES"~3) OR ("VIDEO SEGMENTS"~3) OR ("VIDEO CHUNKS"~2) OR ("MEDIA SAMPLES"~3) OR ("MEDIA SEGMENTS"~3) OR ("MEDIA CHUNKS"~2)) AND (TOKEN*) AND (("REFERENCE DATA"~3) OR ("REFERENCE CONTAINER"~3) OR (METADATA)) AND (PLAYBACK)
3	ABST/(("CONTENT DELIVERY"~3) OR ("VIDEO DELIVERY"~3) OR ("MEDIA STREAMS"~3)) AND SPEC/(("VIDEO SAMPLES"~3) OR ("VIDEO SEGMENTS"~3) OR ("VIDEO CHUNKS"~2) OR ("MEDIA SAMPLES"~3) OR ("MEDIA SEGMENTS"~3) OR ("MEDIA CHUNKS"~2)) AND (TOKEN*) AND (("REFERENCE DATA"~3) OR ("REFERENCE CONTAINER"~3) OR (METADATA)) AND (PLAYBACK) AND

	(AUTH* OR VALIDATION OR VERIFICATION OR CERTIFICATES)
5	ABST/(("VIDEO PLAYBACK"~3) OR ("VIDEO STREAMING"~2) OR ("CONTENT DELIVERY"~3) OR ("VIDEO DELIVERY"~3) OR ("MEDIA STREAMS"~3)) AND SPEC/((CONSENT) OR (LICENSING) OR (OWNERSHIP) OR ("RIGHTS PARAMETERS"~3)) AND (("VIDEO SAMPLES"~3) OR ("VIDEO SEGMENTS"~3) OR ("VIDEO CHUNKS"~2) OR ("MEDIA SAMPLES"~3) OR ("MEDIA SEGMENTS"~3) OR ("MEDIA CHUNKS"~2)) AND (TOKEN*) AND (("REFERENCE DATA"~3) OR ("REFERENCE CONTAINER"~3) OR (METADATA)) AND (PLAYBACK) AND (AUTH* OR VALIDATION OR VERIFICATION OR CERTIFICATES)
S No.	**WIPO Search Strings**
1	EN_ALLTXT:((TOKEN*) AND ((SEGMENT OR PIECES OR SAMPLES) NEAR5 VIDEO) AND (METADATA OR “STRUCTURED METADATA”) AND ((PIN OR KEY OR PASSCODE OR PASSWORD) NEAR5 (ENCRYPT* OR DECRYPT*))) AND CLASSIF:(H04L9/0891 OR H04L9/3226 OR G06V20/49)
2	EN_ALLTXT:(((RENDER* OR PROCESSING) NEAR5 (GRAPHICS OR VIDEO)) AND (TOKEN*) AND ((SEGMENT OR PIECES OR SAMPLES) NEAR5 VIDEO) AND (METADATA NEAR2 (EXTRACT* OR STRUCTUR*)) AND ((PIN OR KEY OR PASSCODE OR PASSWORD) NEAR5 (ENCRYPT* OR DECRYPT*))) AND CLASSIF:(H04L9/0891 OR H04L9/3226 OR G06V20/49 OR G06F21/1085)
3	“EN_AB:((“”VIDEO PLAYBACK””~3) OR (“”VIDEO STREAMING””~2) OR (“”CONTENT DELIVERY””~3) OR (“”VIDEO DELIVERY””~3) OR (“”MEDIA STREAMS””~3)) AND EN_ALLTXT:((CONSENT) OR (LICENSING) OR (OWNERSHIP) OR (“”RIGHTS PARAMETERS””~3)) AND ((“”VIDEO SAMPLES””~3) OR (“”VIDEO SEGMENTS””~3) OR (“”VIDEO CHUNKS””~2) OR (“”MEDIA SAMPLES””~3) OR (“”MEDIA SEGMENTS””~3) OR (“”MEDIA CHUNKS””~2)) AND (TOKEN*) AND ((“”REFERENCE DATA””~3) OR (“”REFERENCE CONTAINER””~3) OR (METADATA)) AND (PLAYBACK) AND (AUTH* OR VALIDATION OR VERIFICATION OR CERTIFICATES)
4	CLASSIF:(H04N21/8456) AND EN_ALLTXT:(TOKEN*) AND (AUTHENT* OR VALIDAT* OR VERIFICATION) AND (PLAYBACK OR (“MEDIA PLAYER”~2)) AND EN_ALLTXT☹METADATA OR INDEXES OR LIST OR REFERENCE* OR LINKS OR HYPERLINK* OR URLS)
5	CLASSIF:(H04N21/8456 OR H04N21/23406) AND EN_ALLTXT:(TOKEN*) AND (CERTIFICATE* OR LICENC* OR (“TRANSACTION PARAMETERS”~2)) AND (PLAYBACK OR (“MEDIA PLAYER”~2)) AND EN_ALLTXT:(METADATA OR HYPERLINK* OR URLS)
6	EN_ALLTXT: ((VIDEO OR MEDIA OR CONTENT) NEAR3 (RENDER* OR RETRIEVAL)) AND (TOKEN OR KEYS OR "ENCRYPTION KEYS") AND (VALIDAT* OR PROTECTED OR AUTHORISED OR AUTHENTICAT*) AND CLASSIF:(G06F16/7837)
7	EN_ALLTXT:(PROGRAMMABLE OR SYSTEM OR ENVIRONMENT OR APPARATUS) AND (PLAYBACK OR PLAYBACK_MEDIA OR PLAYBACK_SESSION) AND ((TOKEN OR KEY OR PASSWORD OR AUTHORIZATION) NEAR3 (SEGMENTS OR SAMPLES OR FRAMES)) AND CLASSIF:(H04N21/2347)
8	EN_ALLTXT:(PLAYBACK OR PLAYBACK_MEDIA OR PLAYBACK_SESSION) AND ((TOKEN OR KEY OR PASSWORD) NEAR3 (SEGMENTS OR SAMPLES OR FRAMES)) AND ((MEMORY) NEAR3 (NON_VOLATILE OR CACHE OR TEMPORARY)) AND CLASSIF:(H04N21/2347)
9	EN_ALLTXT:(VIDEO OR MULTIMEDIA OR MEDIA_STREAM) AND (METADATA OR REFERENCE OR STRUCTURE) AND ((TOKEN OR KEY OR PASSWORD) NEAR3 (SEGMENTS OR SAMPLES OR FRAMES)) AND ((MEMORY) NEAR3 (NON_VOLATILE OR CACHE OR TEMPORARY)) AND CLASSIF:(H04N21/2347)
S No.	**Espacenet Search Strings**

1	(CTXT ALL "TOKENIZATION" OR CTXT ALL "SEGMENTATION" OR CTXT ALL "PARSING" OR CTXT ALL "ENCODING" OR CTXT ALL "DISCRETIZATION") AND ((FTXT=("VIDEO" PROX/DISTANCE<3 "SEGMENTS") OR FTXT=("VIDEO" PROX/DISTANCE<3 "PIECES") OR FTXT=("VIDEO" PROX/DISTANCE<3 "SAMPLES")) AND FTXT=("STRUCTURE*" PROX/DISTANCE<3 "METADATA") AND FTXT = "VIDEO DELIVERY")
2	(CTXT ALL "TOKENIZATION" OR CTXT ALL "SEGMENTATION" OR CTXT ALL "PARSING" OR CTXT ALL "ENCODING" OR CTXT ALL "DISCRETIZATION") AND ((FTXT=("VIDEO" PROX/DISTANCE<3 "SEGMENTS") OR FTXT=("VIDEO" PROX/DISTANCE<3 "PIECES") OR FTXT=("VIDEO" PROX/DISTANCE<3 "SAMPLES")) AND (FTXT = "SINGLE KEY" OR FTXT = "PER-USER KEY" OR FTXT = "ONE TIME") AND (FTXT ALL "DECENTRALIZED" OR FTXT = "PEER-TO-PEER" OR FTXT ALL "P2P"))
3	(NFTXT ALL "VIDEO" OR NFTXT ALL "IMAGE" OR NFTXT ALL "MEDIA") AND (NFTXT=("GENERATE" PROX/DISTANCE<3 "TOKEN") OR NFTXT ANY "VIRTUALISE") AND (NFTXT ANY "TOKEN" OR NFTXT ANY "VALIDAT*" OR NFTXT=("TOKEN" PROX/DISTANCE<3 "AUTHENTICAT*")) AND (NFTXT ANY "ENCRYPT" OR NFTXT ANY "RENDER" OR NFTXT ANY "RETRIEVE") AND CL ANY "H04L9/3226"
4	(NFTXT ALL "VIDEO" OR NFTXT ALL "IMAGE" OR NFTXT ALL "MEDIA") AND (NFTXT ANY "TOKEN" OR NFTXT ANY "VALIDAT*" OR NFTXT=("TOKEN" PROX/DISTANCE<3 "AUTHENTICAT*")) AND (NFTXT ANY "ENCRYPT" OR NFTXT ANY "RENDER" OR NFTXT ANY "RETRIEVE") AND CL ANY "H04L9/0891" AND (NFTXT=("SINGLE " PROX/DISTANCE<3 "SESSION") OR NFTXT=("ONE" PROX/DISTANCE<3 "PLAY") OR NFTXT ANY "PLAYBACK")
5	(NFTXT ALL "VIDEO" OR NFTXT ALL "MEDIA" OR NFTXT ALL "CONTENT") AND (NFTXT ANY "EXTRACT" OR NFTXT ANY "RETRIEVE") AND (NFTXT ANY "METADATA" OR NFTXT = "STRUCTURAL DATA") AND CL ANY "H04N21/8456" AND (NFTXT ANY "TOKEN" OR NFTXT=("ACCESS" PROX/DISTANCE<3 "TOKEN")) AND (NFTXT=("TOKEN" PROX/DISTANCE<3 "VALIDAT*") OR NFTXT=("TOKEN " PROX/DISTANCE<3 "AUTHENTICAT*"))
6	(nftxt=("VIDEO" prox/distance<3 "DELIVERY") OR nftxt=("VIDEO" prox/distance<3 "RETRIEVAL")) AND (nftxt=("TOKEN" prox/distance<3 "VALIDAT*") OR nftxt=("TOKEN" prox/distance<3 "AUTHENTICAT*")) AND (nftxt=("SPECIFIC" prox/distance<3 "TOKEN") OR nftxt=("DISTINCT" prox/distance<3 "KEY*") OR nftxt=("SEPARATE" prox/distance<3 "KEY*"))
S No	**Google Patent Search Strings**
1	TAC=((PROGRAMMABLE OR CONFIGURABLE OR CUSTOMIZABLE OR ADAPTABLE OR SCRIPTABLE) NEAR3 VIDEO) TAC=((TOKENIZATION OR SEGMENTATION OR PARSING OR ENCODING OR DISCRETIZATION) NEAR3 (SAMPLES OR TIMESTAMP)) (TOKEN NEAR3 (VALIDATION OR ISSUANCE OR GENERATION OR CREATION OR DISTRIBUTION)) AB=(CRYPTOGRAPHIC NEAR2 TOKEN) (MEDIA OR VIDEO)
2	TAC=((MEDIA OR CONTENT OR VIDEO) NEAR3 (DELIVERY OR SHARING)) AB=(TOKENIZATION OR ((SECURE OR PROTECTED OR ENCRYPTED) NEAR2 TOKEN)) (TOKEN NEAR3 (VALIDATION OR ISSUANCE OR GENERATION OR CREATION OR DISTRIBUTION)) ("ONE SESSION" OR "ONE-TIME" OR "SINGLE PLAYBACK") (DECENTRALIZED NEAR3 (SYSTEM OR PLATFORM))
3	AB=(VIDEO NEAR3 (DELIVERY OR SHARING)) AB=(TOKENIZATION OR ((SECURE OR PROTECTED OR ENCRYPTED) NEAR2 TOKEN)) (TOKEN NEAR3 (VALIDATION OR ISSUANCE OR GENERATION OR CREATION OR DISTRIBUTION)) (DECENTRALIZED NEAR3 (SYSTEM OR PLATFORM))
4	AB= ((PROGRAMMABLE OR CONFIGURABLE OR CUSTOMIZABLE OR ADAPTABLE OR SCRIPTABLE) NEAR2 VIDEO) TAC=(TOKENIZATION OR ((SECURE OR PROTECTED OR ENCRYPTED) NEAR2 TOKEN)) (TOKEN NEAR3 (VALIDATION OR ISSUANCE OR GENERATION OR CREATION OR DISTRIBUTION)) (DECENTRALIZED NEAR3 (SYSTEM OR PLATFORM))

5	CPC=(H04L9/3226 OR G06F21/1085 OR H04N21/00 OR G06V20/49) ((SINGLE OR PER-USER OR ONE-TIME) NEAR3 KEY) TAC=(TOKENIZATION OR ((SECURE OR PROTECTED OR ENCRYPTED) NEAR2 TOKEN)) (DECENTRALIZED OR PEER-TO-PEER OR P2P) (TOKEN NEAR3 (VALIDATION OR ISSUANCE OR GENERATION OR CREATION OR DISTRIBUTION)) (MEDIA OR VIDEO OR AUDIO)
6	(VIDEO) AND ((REFERENCE NEAR/2 LINK*) OR METADATA) AND (VIDEO* NEAR/3 (SEGMENT* OR CHUNK* OR INTERVAL*)) AND (TOKEN* OR KEY* OR PASSWORD* OR PASSCODE*)
7	AB=(VIDEO OR MEDIA OR MULTIMEDIA OR MOVIE) AND ((ENCRYPT* OR CRYPTOGRAPHIC*) NEAR/4 (LINK* OR URLS OR REFERENCE OR LIST)) AND DS= (((REFERENCE OR METADATA) NEAR/3 (GENERAT* OR CONTAINER)) SAME (VIDEO*)) AND ((TRANSACTION OR PAYMENT* OR SUBSCRIPTION* OR LICENSE) SAME (VERIF* OR AUTHENT* OR VALID*))
8	((H04N21/2541) AND (H04N21/8547 OR H04N21/8456)) AND ((VIDEO OR MEDIA) NEAR/3 (DELIVERY OR BROADCAST* OR TRANSMIT*)) AND (TOKEN OR TICKET* OR PASSCODE OR PASSWORDS)
9	AB=((VIDEO OR MEDIA) NEAR/3 (DELIVERY OR BROADCAST* OR TRANSMIT*)) AND ((GENERAT* OR CREAT* OR PARS* OR MANIFEST*) NEAR/3 (REFERENCE* OR LINK* OR METADATA OR HYPERLINK*)) AND ((MULTIPLE OR PLURAL* OR NUBER) NEAR/4 (TOKEN* OR CODE* OR KEYS OR PASSCODE*))
10	AB=((VIDEO OR MEDIA OR SONG* OR MULTIMEDIA OR AUDIO) NEAR/3 (DELIVERY OR TRANSMIT* OR BROADCAST* OR PLAY*)) AND DS= (REMOVE NEAR/5 ((ORIGINAL OR ACTUAL) NEAR/3 (DATA OR VIDEO OR SONG OR INFO))) AND ((GENERAT* OR CREAT* OR PARS*) NEAR/3 (REFERENCE* OR LINK* OR METADATA OR HYPERLINK*)) AND ((MULTIPLE OR PLURAL* OR NUMBER) NEAR/4 (TOKEN* OR CODE* OR KEYS OR PASSCODE*)) AND (H04N21/2541 OR H04N21/835)
11	((PROGRAMMABLE OR TOKENISED) ADJ/3 (VIDEO)) AND ((TOKENS OR ((ACCESS OR CRYPTOGRAPHIC) ADJ/3 (CREDENTIALS OR INFORMATION))) WITH (AUTHENTICATE OR AUTHORISE OR PROTECT)) AND (VIDEO OR IMAGE OR MEDIA) AND ((SINGLE OR "ONE TIME") NEAR/3 (SESSION OR PLAYBACK)) AND ((PLAY) SAME ("VOLATILE MEMORY" OR ("RAM"))) AND CPC=(G06V20/49)
12	(VIDEO OR IMAGE) AND ((VIDEO OR MEDIA OR IMAGE) ADJ/3 (VIRTUALISE)) AND (METADATA OR ARRANGEMENT OR CONFIGURATION OR STRUCTURE) AND (SAMPLES OR SEGMENT OR PARTS) AND (("CRYPTOGRAPHIC TOKENS") NEAR/3 (VALIDAT* OR AUTHENICAT* OR AUTHORIZ*))
13	((PROGRAMMABLE OR TOKENISED OR PERSONALISED) ADJ/3 (VIDEO OR IMAGE OR MEDIA)) (METADATA OR "STRUCTURAL DATA" OR ARRANGEMENT) (VIDEO OR IMAGE) ADJ/3 (TOKEN*) ("ONE TIME" OR SINGLE) ADJ/3 (SESSION) (ENCRYPT OR PROTECT OR SECURE) ADJ/3 (LINK OR "URI" OR "URL") ("TOKEN VALIDATION") (RETRIEVE OR PLAYBACK OR RENDER OR STREAM)
14	(VIDEO OR IMAGE) AND ((VIRTUAL* OR REFERENCE) NEAR/3 (ENGINE OR CONVERTER OR CONTAINER OR GENERATOR OR EXTRACTOR OR FILE OR COMPILER)) AND ((CRYPTOGHAPHIC OR SECURITY) NEAR/3 (TOKEN)) AND ((TOKEN) NEAR/3 (VERIF* OR AUTHENTICATION OR CONFIRMATION OR CHECK*)) AND (METADATA OR "META DATA") AND ((SINGLE OR "ONE TIME") ADJ/3 (VIEW* OR PLAYBACK OR STREAM)) AND ("RAM" OR ((VOLATILE OR TEMPORARY OR TRANSIENT) ADJ/3 (MEMORY))) AND (G06F21/125)
15	TAC=(SYSTEM OR ENGINE OR PLATFORM OR PROGRAMABLE) TAC=((SAMPLES OR SEGMENTS OR FRAMES) NEAR3 (TOKEN OR USER_TOKEN OR TOKEN_BASE OR DIGITAL_TOKEN)) ((PLAYBACK) NEAR3 (SESSION OR STREAM*)) TAC=(TOKEN NEAR3 (INVALIDATE OR ENCRYPT)) TAC=(VIDEO OR MULTIMEDIA OR FOOTAGE)

5. Non-patent Search Strings

S No.	Strings (Google /Google Scholar/Science Direct)
1	PROGRAMMABLE AND TOKENIZED VIDEO DELIVERY
2	TOKEN GENERATION/ASSOCIATION WITH VIDEO SEGMENTS
3	ENCRYPTION OF VIDEO OR EMBEDDED TOKEN DATA
4	VIDEO REFERENCE CONTAINER TOKEN" OR "VIRTUALIZED VIDEO RESOLUTION TOKEN".
5	REFERENCE REMOTE VIDEO SEGMENTS WITHOUT EMBEDDING SAMPLES
6	("VIDEO STREAM" OR "VIDEO PLAYBACK") AND (TOKEN OR "ENCRYPTION KEYS" OR KEYS) AND ((VIDEO OR MEDIA) AROUND(3) (SAMPLE OR SEGMENT OR PORTION OR PART)) AND (VIDEO_METADATA)
7	(VIDEO_PLAYBACK OR VIDEO_STREAM OR VIDEO_RENDER) AND (TOKEN OR KEYS OR "ENCRYPTION KEYS") AND (VIDEO_SAMPLE OR VIDEO_SEGMENTS OR VIDEO_FRAME) AND (VALIDAT* OR AUTHENTICAT*) AND (SECURE OR PROTECTED OR ENCRYPTED) AND (METADATA)
8	((VIDEO) AROUND(3) (PLAYBACK OR STREAM OR RENDER)) AND ((VIDEO OR MEDIA OR CONTENT) AROUND(3) (SEGMENTS OR SAMPLES OR FRAMES OR PORTIONS)) AND ((TOKEN OR "ENCRYPTION KEYS") AROUND(3) (VALID* OR AUTHENTICAT*)) AND ("RAM") AND ("REMOTE SOURCES" OR "CLOUD STORAGE")
9	(PLAYBACK OR STREAM OR RENDER) AND ((VIDEO_METADATA OR VIDEO_STRUCTURE OR VIDEO _ARRANGEMENT) AROUND(3) (TOKEN ON ENCRYPTION_KEY)) AND ((TOKEN) AROUND(3) (VALID* OR AUTHENITICAT*)) AND (ONE_TIME_ACCESS OR SINGLE_SESSION)
10	("VIDEO VISUALIZATION") AND ((TOKEN OR ENCRYPTION_KEY) AROUND(3) (VALID* OR AUTHENTICAT* OR AUTHORIZ*)) AND (SEGMENT OR SAMPE OR FRAME) AND (PLAYBACK OR STREAM OR RENDER) AND ((VIDEO OR MEDIA OR CONTENT) AROUND(3) (RETRIEVE OR EXTRACT))
11	(VIDEO) AND ("VIDEO PLAYBACK" OR "VIDEO STREAMING") AND (SEGMENTS OR FRAMES) AND (TOKEN) AND (VALIDATION OR AUTHENTICATION) AND (METADATA)
12	(SECURE OR PROTECTED OR ENCRYPTED) AND ("VIDEO PLAYBACK") AND (TOKEN OR "ENCRYPTION KEY") AND (SEGMENT OR FRAMES) AND (VALIDATION)
13	(SYSTEM OR ENVIRONMENT OR APPARATUS) AND (PLAYBACK OR PLAYBACK_MEDIA OR PLAYBACK_SESSION) AND ((TOKEN OR KEY OR PASSWORD OR AUTHORIZATION) NEAR3 (SEGMENTS OR SAMPLES OR FRAMES))
14	(VIDEO_STREAM OR MEDIA_STREAM OR PLAYBACK_STREAM) AND (USER CONSENT OR USER_AUTHORIZATION OR USER_PERMISSION) AND ((TOKEN OR KEY OR PASSWORD OR AUTHORIZATION) NEAR3 (SEGMENTS OR SAMPLES OR FRAMES))
15	(VIDEO_STREAM OR MEDIA_STREAM OR PLAYBACK STREAM) AND (TOKEN OR KEY OR CRYPTOGRAPHIC_TOKEN OR AUTHORIZATION_KEY) AND ((VIDEO_MEDIA OR MULTIMEDIA) NEAR3 (SEGMENTS OR SAMPLES OR FRAMES))
16	(VIDEO OR MULTIMEDIA OR FOOTAGE) AND ((SAMPLES OR SEGMENTS OR FRAMES) NEAR3 (TOKEN OR USER_TOKEN OR TOKEN_BASE OR DIGITAL_TOKEN)) AND ((PLAYBACK) NEAR3 (SESSION OR STREAM*))

6.Relevant Classifications

a. IPC/CPC Classification

IPC/CPC	Definition
G06F16/00	Information Retrieval; Database Structures Therefor; File System Structures Therefor
G06F16/7837	Using Objects Detected Or Recognised In The Video Content
G06V20/00	Scenes; Scene-Specific Elements
G06V20/49	Segmenting Video Sequences, I.E. Computational Techniques Such As Parsing Or Cutting The Sequence, Low-Level Clustering Or Determining Units Such As Shots Or Scenes
H04L67/00	Network Arrangements Or Protocols For Supporting Network Services Or Applications
H04L67/02	Based On Web Technology, E.G. Hypertext Transfer Protocol [Http]
H04L9/00	Cryptographic Mechanisms Or Cryptographic Arrangements For Secret Or Secure Communications; Network Security Protocols
H04L9/0891	Revocation Or Update Of Secret Information, E.G. Encryption Key Update Or Rekeying
H04L9/3226	Using A Predetermined Code, E.G. Password, Passphrase Or Pin
H04N21/00	Selective Content Distribution, E.G. Interactive Television Or Video On Demand [VOD]
H04N21/2368	Multiplexing Of Audio And Video Streams
H04N21/4307	Synchronising The Rendering Of Multiple Content Streams Or Additional Data On Devices, E.G. Synchronisation Of Audio On A Mobile Phone With The Video Output On The TV Screen
H04N21/43072	Of Multiple Content Streams On The Same Device
H04N21/4341	Demultiplexing Of Audio And Video Streams
H04N21/4622	Retrieving Content Or Additional Data From Different Sources, E.G. From A Broadcast Channel And The Internet
H04N21/4753	For User Identification, E.G. By Entering A Pin Or Password
H04N21/6125	Involving Transmission Via Internet
H04N21/64322	IP
H04N21/8456	By Decomposing The Content In The Time Domain, E.G. In Time Segments
H04N21/8543	Using A Description Language, E.G. Multimedia And Hypermedia Information Coding Expert Group [Mheg], Extensible Markup Language [Xml]

b. F-Terms

F-Term	Definition
5C164UD63	Peer-to-peer (P2P) communications or communications between terminals
5C164UD64	of bonus points or coupons
5L096EA35	Segmentation

7. Key Inventors:

Anthony Sharick	Jieyi Long	Oztan Harmanci
Braden Christopher Ericson	John D. Rinaldo	Paul Tweedale
Brett Ronald Welch	Jonathan Yantis	Perkes Ronald
Catherine Lin-Hendel	Kaili An	Robert E. Seastrom
Christopher Albert Hartley	Kevin J. Ma	Robert Hickey
Finbar O'hanlon	Lukasz Jakub Sliwka	Robert W. Lord
Fun-Chen Jou	Lyons Grant	Royce A. Levien
Garcia Barrio Laura	Mahmoud Elbarky	Ryan Steelberg
Gregory Simon	Mark A. Malamud	Sam Pourcyrous
Hideki Fukuda	Mitchell C. Liu	Sean Dennis
Hideshi Ishihara	Mitsuaki Oshima	Shannon Code
Hiromu Kitaura	Morris Marc Anthony	Takaoka Tomohisa
Ismail R. Haritaoglu	Motzel Thomas	Vincent E. Colella
Jeffrey A. Aitken	Neil C. Marck	Wesley E. George

8. Key Assignee(s)

Actv	Intertrust Tech	Searete
Alphabet	Intertrust Technologies	Snap
Ambrosia Software	Invention Science Fund	Sony
Apple	Ion Video	Srsly
Broadcom	Linius	The Invention Science Fund
Bygge Technology	Loyyal Holdings	Theta Labs
Charter Communication	Ntt Docomo	Time Warner Cable Enterprises
Dish Network	Opentv	Tunespotter
Echostar	Panasonic	Vantiva
Ericsson	Paypal	Verona
Friskit	Qwest Communications International	Vivcom
Google	Radical Urban	Webtv Networks
Interdigital	Rezzonation	

9. Disclaimer

This report is work of analysis and interpretation of publicly available information on various free and paid online sources and should not be construed as a legal opinion. This report is shared with you with mutual understanding and trust that no part of this report shall be publicly distributed or used by you without explicit permission from barcodeIP. If you have purchased this report, please do not resell.