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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
08 October 2020 (08.10.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/201997 A1

(51) International Patent Classification:
H04L 9/06 (2006.01) *H04L 9/30* (2006.01)
H04L 9/08 (2006.01)

(21) International Application Number:
PCT/IB2020/053032

(22) International Filing Date:
30 March 2020 (30.03.2020)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
752210 29 March 2019 (29.03.2019) NZ

(71) Applicant: **SOUL MACHINES** [NZ/NZ]; Level 1, 106 Customs St West, Auckland (NZ).

(72) Inventor: **MUNRO, Robert**; 203/c Hobson Street Auckland (NZ), Auckland (NZ).

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

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report (Art. 21(3))*
— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*
— *in black and white; the international application as filed contained color or greyscale and is available for download from PATENTSCOPE*

(54) Title: CRYPTOGRAPHIC SYSTEMS

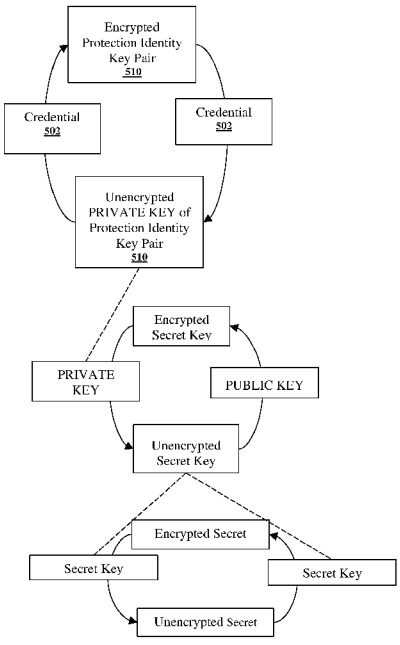

FIGURE 8

(57) Abstract: The cryptographic system allows a cloud service provider to persist customer data on behalf of an intermediary service provider, but provide end-to-end encryption such that cloud service provider is not able to read the data. A Secrets Vault provides a cryptographic ally enforced mechanism by which Secrets are protected and only accessible by authorised users or services, and access to Secrets is cryptographically provable. An entity with a valid Credential has an associated Key Pair. The Credential is used to encrypt/wrap the Key Pair (namely the private key of the Key Pair, as by definition public Keys are designed to be publicly accessible). This allows the Key Pair to be stored online, in an encrypted form, Secrets have corresponding Secret Keys which are used to symmetrically encrypt the Secrets. The Secret Keys are then asymmetrically encrypted using public- private key-pairs.

WO 2020/201997 A1

CRYPTOGRAPHIC SYSTEMS

TECHNICAL FIELD

[0001] Embodiments of the invention relate to cryptographic systems.

BACKGROUND ART

[0002] Key management refers to management of cryptographic keys in a cryptographic system, including the generation, exchange, storage, use, destruction and replacement of keys. Traditionally, key management services provide & protect cryptographic keys, such that in the long term, unencrypted keys are stored in a different location from the storage of encrypted data, and users must authenticate themselves with the key management service provider to gain access to the unencrypted key in order to decrypt the data. Key management systems are software applications deployed on servers. Rights to data are generated by the system and enforced through Boolean mechanisms, and access to a key store is unlocked with a username or password which authenticates a person as a user of the system. However, such key management services depend on software-based control (logic / rule-based control), meaning that a flaw in the software may allow unauthorised users to access the keys.

[0003] US20180287785 describes a cryptographic system providing a data storage service that uses a key provider to acquire wrap keys (key-encryption key or key-wrapping key that is used by the key-wrap algorithm to encrypt another key) and form a wrap key pool.

Objects of the Invention

[0004] It is an object of the present invention to improve cryptographic systems or to at least provide the public or industry with a useful choice.

SUMMARY OF THE INVENTION

A cryptographic system allows a cloud service provider to persist customer data on behalf of an intermediary service provider, but provide end-to-end encryption such that cloud service provider is not able to read the data. A Secrets Vault provides a cryptographically enforced mechanism by which Secrets are protected and only accessible by authorised users or services, and access to Secrets is cryptographically provable. An entity with a valid Credential has an associated Key Pair. The Credential is used to encrypt/wrap the Key Pair (namely the private key of the Key Pair, as by definition public Keys are designed to be publicly accessible). This allows the Key Pair to be stored online, in an encrypted form. Secrets have corresponding Secret Keys which are used to symmetrically encrypt the Secrets. The Secret Keys are then asymmetrically encrypted using public-private key-pairs.

According to one embodiment: A method of cryptographically securing a Secret for access by an entity, the entity having a Credential and an associated Key Pair, the key pair including a private key and a public key, wherein the private key of the Key Pair is encrypted using the Credential, the method including the steps of: generating a Secret Key; symmetrically encrypting the Secret using the Secret Key; and asymmetrically encrypting the Secret Key using the public key of the Key Pair. Optionally: The private key of the Key Pair is symmetrically encrypted using the Credential. The Secret is encrypted using AES192 or AES256. The Secret Key is encrypted using the RSA algorithm or the Elliptic Curve Digital Signature algorithm. The Secret is a data stream.

According to one embodiment: A method of cryptographically securing at least one Secret for access by an entity, the entity having a Credential and an associated public key, the method including the steps of: for each Secret, generating or receiving a Secret Key Pair including the associated public key and a new private key unique to the Secret; encrypting the private key of the Secret Key Pair using the Credential; and asymmetrically encrypting each Secret using the associated public key of the Secret Key Pair.

According to one embodiment: A method of cryptographically securing at least one Secret for access by an entity, the entity having a Credential and an associated Identity Key Pair, the Identity Key Pair including a public key and a private key, including the steps of: generating or receiving a Secret Key for encrypting the at least one Secret; encrypting the at least one Secret using the Secret Key; asymmetrically encrypting the Secret Key using the Identity Key Pair; and symmetrically encrypting the private key of the Identity Key Pair with the Credential.

According to one embodiment: a system for cryptographically securing Secrets collected by a Data Collector for access by an entity, such that the Secrets are unreadable by the Data Collector, the system including: the entity, wherein the entity is associated with a Credential and an associated public key; a Secret Key Pair generator configured to generate Secret Key Pairs for each Secret, wherein the Secret Key Pairs include the associated public key and a new private key unique to the Secret; and the Data Collector configured to: capture or receive the each Secret, encrypt the private key of each Secret Key Pair using the Credential; and asymmetrically encrypt each Secret using the associated public key of the Secret Key Pair.

A system for cryptographically securing Secrets collected by a Data Collector for access by an entity, such that the Secrets are unreadable by the Data Collector, the system including: the entity, wherein the entity is associated with a Credential and an associated Identity Key Pair; a Secret Key generator configured to generate Secret Keys for each Secret, and the Data Collector configured to: capture or

receive each Secret, asymmetrically encrypt each Secret using the Identity Key Pair; and symmetrically encrypt the private key of the Identity Key Pair with the Credential.

BRIEF DESCRIPTION OF DRAWINGS

Figure 1 shows a base deployment and usage architecture;

Figure 2 shows a flow diagram of data obtained from user interaction;

Figure 3 shows one example of classification of each part of a data stream that is targeted for capture;

Figure 4 shows a schematic diagram of the components of the Secrets Vault according to one embodiment;

Figure 5 shows storage of secrets according to one embodiment;

Figure 6 shows a cryptographic system according to one embodiment;

Figure 7 shows a cryptographic system according to another embodiment;

Figure 8 shows a schematic diagram of an encrypted Secret according to one embodiment; and

Figure 9 shows a method of storing secrets.

DISCLOSURE OF INVENTION

Technical Problem

[0005] A technical problem to be solved is allowing protection of user data with a derived key, such that a service provider can hold the data for the user but cannot access it (end-to-end encryption), such that the data can be shared with other users at the behest of the user, while maintaining that end-to-end encryption. Preferably the system would not require any re-encryption of the file data itself, avoiding an expensive operation.

[0006] Another challenge is in providing authentication mechanisms which are cryptographically enforced (as opposed to programmatically enforced), decreasing the likelihood of a coding error to result in a third party gaining unauthorized access, and reducing the risk of escalation of privilege attacks. Preferably, a cryptographically enforced system allows new users or service providers to be authorized and de-authorized without affecting other aspects of the system or exposing secrets to third parties.

Technical Solution

[0007] A Secrets Vault provides a cryptographically enforced mechanism by which Secrets are protected and only accessible by authorised users or services. Figure 8 shows an encryption

architecture whereby Secrets are dually-encrypted such that they are only accessible by a user with a valid Credential. An entity with a valid Credential has an associated Key Pair 510. The Credential is used to encrypt/wrap the Key Pair (namely the private key of the Key Pair, as public keys are public). This allows the Key Pair to be stored online, in an encrypted form. Secrets have corresponding Secret Keys which symmetrically encrypt/decrypt the Secrets. The Secret Keys are in turn asymmetrically encrypted using the Key Pair.

[0008] In a "multiple secret keys" embodiment, Secrets are encrypted with Secret Keys which are symmetric keys themselves encrypted with Identity Key Pairs associated with authorized user Credentials. In "multiple secret keys" embodiments, Credentials are 1:1 with Identity Key Pairs. Secret Keys are generated regularly, on demand, and double-encrypt all sensitive information, such as sessions or configuration information. Providing valid Credentials allows decryption of Identity Key Pairs, which in turn allows decryption of all Secret Keys stored under those Identity Key Pairs which can be used to decrypt their respective Secrets. The Secret Key is asymmetrically encrypted using the unencrypted private key of the Identity Key Pair. Figure 6 shows a schematic diagram of Secrets stored according to a "multiple secret keys" embodiment. Four entities, which are Service Providers A, B, C and D each have respective Credentials, and respective Identity Key Pairs. Service Provider B has access to Secret X via a copy of Secret Key X which is encrypted by the Service Provider B's public Key B. Service Provider A has access to two Secrets, Secret X and Secret Y. Service Provider C has access only to Secret Y, and Service Provider D has access to Secret X and Secret Y. For each Secret, multiple Secret Keys are stored; one for each entity with access. Three copies of Secret Key X exist, because Service Provider A, B, and D each have an associated Secret Key. Similarly, three copies of Secret Key Y exist, because Service providers A, C and D each have a copy.

[0009] In a "multiple key-pair" embodiment, a Credential is associated with a public key of an authorised entity. For each new Secret to be accessible by the entity, a new Secret Key Pair is created using the public key and a private key unique to the Secret and the entity. The Secret is asymmetrically encrypted using the private key, and the private key of the Identity Key Pair is encrypted using the Credential. So, instead of using a new Secret Key for each authorized entity, a single Secret Key is used to create multiple Secret Key Pairs associated with the Credential. Figure 7 shows a schematic diagram of Secrets stored according to a "multiple key-pair" embodiment. Each Secret has one corresponding Secret Key, which may be asymmetrically encrypted multiple times to create a new Secret Key Pair for each entity with access to the secret. Service provider B has access to one secret: Secret X. Service Provider A has access to three Secrets: X, Y and Z.

Secret Y is accessible by three entities: Service Provider A, Service provider C and Service Provider D.

Elements of the Secrets Vault

[0010] A Secrets Vault 17 includes Identity Key Pairs or Secret Key Pairs, Credentials, Secret Keys and a key and identity database (Secrets Key DB).

Secret

The system may be used to protect any Secrets. A Secret is any data to which access is controlled and may include Sensitive Configuration and/or Credential Data. In one embodiment, Secrets are sensitive data streams. Configuration and/or credential data such as external service credentials or other application settings that might be considered sensitive can similarly be encrypted as Secrets and stored in the Secrets Vault to keep sensitive data secure and in an authenticated state. Sensitive configuration and credential data may be stored as a binary or text chunk in the Secrets Vault along with a hmac (e.g. HMAC256) which is used to verify the integrity of the data to the data owner (the authorised assessor / creator of the secret data)

Credential

[0011] The Credential is a key, or a key generated via a password using a standard password-based protection mechanism. For example, it may be generated using PBKDF2. The Credential is used to encrypt and protect the Identity Key Pair private key.

Identity Key Pair

[0012] The Identity Key Pair includes a public key certificate unique to the entity / user with an associated Credential. For example, it may be implemented using the RSA, or the Elliptic Curve Digital Signature Algorithm. The public key portion of the Identity Key Pair is used for Secret encryption and validation purposes. In cryptographical systems as described herein, Identity Key Pairs are 1:1 with Credentials. A private key portion of an Identity Key Pair is encrypted using the Credential and is used to decrypt protected Secret Keys.

Secret Key Pair

[0013] Similarly, the Secret Key Pair also includes a public key certificate unique to the entity / user with an associated Credential. For example, it may be implemented using the RSA, or the Elliptic Curve Digital Signature Algorithm. The public key portion of the Secret Key Pair is used for Secret encryption and validation purposes. In cryptographical systems as described herein, a plurality of Secret Key Pairs are created for each Secret to be secured under a particular Credential. The private

key portions of each Secret Key Pair are protected using the Credential and are used to decrypt protected Secret Keys or Secrets.

Secret Key

[0014] The Secret Key (which may be a session protection key) may be any suitable symmetric encryption key. For example, a standard AES192 or AES256 encryption key may be used. These Secret Keys are unique to each session and Service Provider. Secret Keys are never stored or transmitted unencrypted. When stored in the Secrets Key DB, the Identity Key Pair private key is used to encrypt the Secret Key. Each block of sensitive data in a cloud service may be encrypted with an Secret Key.

[0015] A Secret Key may be used to protect data itself protected or encrypted with a user credential e.g. in the form of the Identity Key Pair private key. User credential could be private/public or user's password. In other words, the Secret Key is wrapped in a manner which requires a user credential. The wrapper is cryptographically enforced.

Secrets Key DB

[0016] Secret Keys are always stored in an encrypted form. The Secrets Key DB is a key and identity database. In one embodiment, this may be a network isolated (clustered) database that stores the contents of Service Provider encrypted session keys (Secret Keys). This database is only accessible via approved and access-controlled mechanisms.

Secrets Vault

[0017] Figure 5 shows the basic mechanism of the Secrets Vault according to one embodiment. A stores three credentials, including a Master Credential 502, a Service Provider Credential 504 and a service Credential 506.

[0018] The Master Credential 502 is used to encrypt Identity Key Pair 510, creating an encrypted Private Key 552. The Master Credential 502 is also used to encrypt Protection Identity Key Pair 512, creating an encrypted Private Key 555. The Service Provider Credential 504 is also used to encrypt Identity Key Pair 510, and a second encrypted Private Key 553 of the Identity Key Pair 512 is created and stored with the Identity Key Pair 512, associated with the Service Provider Credential 504. The service Credential 506 is used only to encrypt Identity Key Pair 514, creating a third copy of a private key for Identity Key Pair 514 (in addition to private keys 558 and 559 created by a Master Credential 502 and a Service Provider Credential 504 respectively).

Authentication Code:

[0019] A hash-based message authentication code may be generated when a Secret (e.g. session data) is stored, to allow for cryptographically secure data integrity checks.

Access Control & Auditing

[0020] An access control mechanism may be used to control any access to Event Data Stream data, regardless of the categorization of data, as well as to Secret Keys. This mechanism ensures that a basic identity check has been performed on the user requesting access to the data and that only data the user is authorized to access is available. Access is tied to a user's role and identity. Users have a set of permissions which governs which part of the Event Data Stream can be accessed and what can be done with the data. Additionally, use of Identity Key Pair ensures that a Service Provider's data can only be accessed by users with permission to access the Service Provider's Identity Key Pair.

[0021] All requests of access, regardless of whether they are successful, may result in an audit event being created in a temper resistant audit log. The log allows all data request to be audited and verified, providing a history trace of data streams from creation through to eradication.

Process

Encrypting a Secret

[0022] Figure 9 shows a method of encrypting a Secret, and Figure 5 shows a schematic diagram of a plurality of Secrets stored as per the method shown in Figure 9.

[0023] At step 901, a Credential is generated and added to the Secrets Vault. The Credential may be a private public key pair, a password, or any other suitable credential.

[0024] At step 902, an associated PPKP (private public key pair) is generated. The PPKP (e.g. Identity Key Pair) may be generated at the same time as the Credential is added to the Secrets Vault. The Identity Key Pair may be a given a name if not supplied.

[0025] At step 903, the PPKP is encrypted by the Credential. If the credential was a password then a symmetric key is derived from the password (via an algorithm such as PBKDF2). If the Credential is a public / private key (e.g. in a PEM file) the public key of the Credential is used to encrypt/wrap the private key of the Identity Key Pair.

[0026] For each new Credential authorised to access the new PPKP , an encryption process as described in step 903 is repeated using the new Credential for encryption, and the encrypted a copy of the private key of the PPKP is stored.

At step 904, a Secret to be added to the Secrets Vault is encrypted using a Secret Key. The Secret Key is created using a specified PPKP (which may be specified using the Credential and the name of the PPKP). The PPKP is used (i.e. the public key portion of the PPKP) to protect the Secret which then requires access to the private key portion of the PPKP to decrypt the Secret.

[0027] At step 905, the encrypted Secret is added to the Secrets Vault.

Accessing a Secret

[0028] To access a Secret a valid Credential is needed to access the Identity Key Pair or the Secret Key Pair for the Secret.

[0029] In the "multiple secret keys" embodiment, if the Credential is valid the Identity Key Pair can be unwrapped/decrypted and the private key of the Identity Key Pair may be accessed and used to decrypt the Secret Key. When a Secret being stored is a chunk of data (rather than a session key), a new symmetric key is generated when the data is added, and used to decrypt the secret data.

[0030] In the "multiple keypair" embodiment, if the Credential is valid the Secret Key Pair can be unwrapped/decrypted and the private key of the Identity Key Pair may be accessed and used to decrypt the Secret.

[0031] Key pairs are used as the secret encryption key source as this allows for read and write authorisation to be separately controlled. Allowing one credential write only access (via the public key) and another credential read and write access (using both keys).

Access Control

[0032] The creator of a Secret is the initial authorizer, and authorizes read access to the Secret. The creator wraps the Secret for each entity authorized to access the Secret.

Adding a new user

[0033] A new Credential is authorized access to secrets kept by a Identity Key Pair by using a pre-authorized Credential to decrypt its copy of the private key of the Identity Key Pair, providing it to the new Credential, which re-encrypts its own copy of the private key of the Identity Key Pair using its own Credential.

Revoking user access

[0034] In the "multiple secret keys" embodiment, access to a Secret can be revoked by removing the applicable Secret Key from their keychain. In the "multiple keypair" embodiment, access to a Secret can be revoked by removing the applicable Secret Key Pair. Only access to the entity's keychain is required. A revocation list may be kept, for example, when a user logs into the cryptographic system their keychain may be grabbed and the "key" to access the secret may be removed from their keychain.

Read and Write Access

[0035] As Identity Key Pairs are used as the source of Secret Keys, this allows read and write authorization to be separately controlled, by allowing one Credential write only access (via the public key), and another Credential read-write access (using both keys).

[0036] User A, using User B's credential can load a public key from User B's Identity Key Pair to encrypt a Secret using a Secret Key created using User B's Public Key of User B's Identity Key Pair. A record is created for the Secret. Once the Secret is encrypted, User A can no longer access the data, unless User A has a Secret Key duplicated and wrapped with User A's own public key from User A's Identity Key Pair. To read the data, User B decrypts their private key, so can unwrap the Secret Key using their Private key, and use the unwrapped Secret Key to unencrypt the Secret. Thus any user with a Credential is able to insert data and write Secrets. However, read access is cryptographically enforced.

Storing New Secrets

[0037] Services or entities which collect data can store data that they are not allowed to access. For example, a Data Collector can encrypt data that an analytics service can read and use for analytics purposes. Authorization of the Analytics modules is cryptographically enforced, and the Data Collector does not have access to the data once it has been stored.

Overwriting Secrets

[0038] The overwriting of Secrets may be protected against using hashes or signatures.

Credential Use and Management

[0039] Any access to a Secrets Vault requires a valid Credential. If no valid Credential is presented then no operations on the Secrets Vault are permissible. The Credential specified also controls which secrets within the vault are accessible by nature of the key wrapping mechanism.

[0040] Credentials can be either a password, from which a symmetric key is derived or a PEM file which contains both a public and private key. As PEM files are usually protected with a password (or should be) this would also be required when a PEM file is used as a Credential.

[0041] Credentials can have one of two authorisations writeonly or readwrite. Write only authorised Credential may only add secrets and not read them. When added, a new Credential has created with validation mechanism in the form of a digital signature, this allows the detection of Credentials that have been illicitly replaced.

[0042] In one embodiment, at least two Credentials are automatically added when a new Secrets Vault is created, one of which is required to add new Credentials.

- Master Credential: This may be a PEM only credential with readwrite access to every key-pair in the vault. This may ensure a forgotten or revoked password does not result in secrets becoming inaccessible.
- Service Provider Credential: This is the customer protection identifier (CPI) this Credential likewise has mechanism full access to every PPKP but it can be either a PEM based Credential or a password based Credential.

[0043] Validation requires access to the master Credential private key or the CPI private key. In the case of a Master and Service Provider Credential they act as each other alternate validation mechanism, as the Service Provider Credential validates the master. Key validation is a performance and data validation mechanism only, not a security one. Replacing a Credential by directly overwriting the data will not help as the PPKP wrapping function will fail when used with the replacement Credential.

[0044] Embodiments described herein allow entities to change their passwords in a simple manner. For example, a user has secrets stored under using a Credential which is a password "password1". An expensive hash function such as pkb22 may process the password to create a hashed password. The hashed password is in turn used as an encryption key to store the Service Provider's encryption key which can be used to decrypt the real key to the user's keychain (their Identity Key Pair). If the user wishes to change their password to "password2", the Service Provider only needs to re-encrypt their own key to the user's keychain (their Identity Key Pair).

Implementation according to one embodiment

[0045] In one embodiment, the cryptographic system described herein is used to protect data captured during a session of an end-user interacting with a computer system. The user may be

interacting with an Agent, which may be an embodied Agent interacting with the user in real time through verbal and non-verbal communication via a computer microphone and camera. An Agent provider (Master) may provide the Agent on behalf of a Service Provider. In a further embodiment, end users may also have a Credential to the cryptographic system and Secrets corresponding to their personal data may be only readable by the users and selectively shareable with the Service Provider/s and/or the Agent provider.

[0046] Figure 1 shows a base deployment and usage architecture. A Data Collector 18 collects and processes raw session event data. A Data Stream Processor 16, processes session data, providing normalised, time series data streams for the analytics tasks. The Secrets Vault 17 collects and protects various session encryption keys created during a data collection session.

[0047] Figure 2 outlines the basic flow of data from user interaction. All data that flows through the Agent Server 27 may use secure, encrypted network connections, such as TLS 1.2 or better (HTTPS). The data flows are represented by coded arrows that represent the nature of the data and the directionality of the data flow.

Data Collector

[0048] The Data Collector 18 accepts an incoming socket connection from the Video Host 23 (Video Host process) and reads all session event messages as they flow from the server. The Data Collector 18 has a Credential and is thus able to write and store Secrets. However the Data Collector 18 does not create any copies of Secret Keys with its own credential; and has write-only access to Secrets Vaults and this property may be cryptographically provable by the absence of any encrypted Secret Keys associated with the Data Collector's 18 credential.

[0049] The Data Collector 18 classifies data that is collected. Pseudonymous or Private/Sensitive elements are encrypted with a symmetric session encryption key (a Secret Key) which is generated at session initialization. The Secret Key is then encrypted using the public key of the Identity Key Pair of any services requiring read access to the data. Thus the Data Collector 18 authorises services or users to access the Secret but does not authorise itself to read the Secret. The Secret Key is stored in encrypted form in the Secrets Vault. As data elements are encrypted the raw data is replaced in the event stream with a reference to the encrypted data which is stored in an alternate data stream. The Secret Key is never stored in disk in unencrypted form. The Secret Key is created in memory, transferred only over an encrypted communication channel (such as HTTPS), and stored in the Secrets Vault in an encrypted form. The classified and protected data elements are then written out to a data vault instance (such as a Cassandra cluster node) into a raw session table. An internet

facing REST interface may be built onto the Data Collector to manage session data collection for suitable scenarios, for example, Agent interaction is on a mobile device or non-a cloud hosted kiosk.

Data Stream Processor

[0050] The Data Stream Processor 16 provides the Analytics Module 19 with a normalised, time series view of the raw session data to allow for easier and more consistent processing. The Data Stream Processor 16 may be run on demand to provide access to session data.

[0051] The service will read the raw session data provided by the Data Collector 18 and provide a Spark dataset that can be used during processing. The provided data set will include both anonymous and decrypted (sensitive) data in a normalised view. The dataset can persist in memory (or generated on demand) for various spark task to work off. The data is not preserved beyond the lifetime of the processing tasks as the sensitive data is never to be cached or written out to any form of long-term storage. The Data Stream Processor 16 may be implemented Java to facilitate integration with JVM based services such as Cassandra and Spark.

Data Vault

[0052] A Data Vault provides the storage and retrieval mechanisms for long term structured data or times-series based data. The Data Vault supports safe and resilient storage of both sensitive and generalised or anonymous data. Any suitable server and corresponding architecture may be used to implement the Data Vault. One example is the Cassandra NoSQL server. The Data Vault may be deployed in a clustered server model with a plurality of clusters being initially deployed – for example, one cluster may be in the EU to isolate EU data for GDPR compliance, and another more general cluster may support storage needs of other geographic locations.

Provisioning

[0053] When a new Service Provider is provisioned on an Agent Cloud an initial 'provisioning' step may generate a Identity Key Pair and an associated Credential before any session data streams are protectable. The Credentials and Identity Key Pairs which are generated are then stored securely in a Secrets Key DB.

[0054] Session data streams are protected via the generation, use and storage of Secret Keys. A new Secret Key is generated on demand for each session and is securely stored in the Secrets Vault immediately after generation. Sensitive data streams can optionally use an authenticated encryption mode (like AES GCM) or use an 'authentication code' stored with the secured data stream as a mechanism to ensure the stored data is unchanged from when it was originally collected. A session data stream may comprise different classifications of data. Each part of a data stream that is targeted

for capture may be classified accordingly. One example of possible classifications of data include Private/Sensitive data, Pseudonymous data and Anonymized data.

[0055] Figure 3 shows one example of classification of each part of a data stream that is targeted for capture. An Event Data Stream 31 man include a stream of data. Each part of the Event Data Stream 31 is categorized as Private/Sensitive data 3, Pseudonymous data 4 or Anonymized data 5. Different classifications of data are protected independently, using independent Secret Keys. For example, parts of the Event Data Stream 31 which constitute Private/Sensitive data 3 are encrypted using Secret Key 113, and parts of the Event Data Stream 31 which constitute Pseudonymous data 4 are encrypted using Secret Key 114. Metadata 6 corresponding to the Event Data Stream 31 may also be stored along with the data itself, and may also be encrypted. Optionally, a Video Stream 32 and/or Audio Stream 33 may also be stored under the Secret Key 113. Anonymized data 5 may be unencrypted.

Secrets Vault Service

[0056] In one embodiment, the Secrets Vault service may run as native Linux process, and provide a REST interface and an additional cli based command interface for interacting with the database. The CLI interface simply presents a command line alternative to the same features provided by the REST interface. All access, either CLI or REST based (HTTPS only) is authenticated using the built in credentials mechanism (see Credential Use and Management). The following set of operations may be provided:

- Add / Remove Credential - requires the use of either the Master Credential or the Service Provider Credential to complete
- Authorise Credential- Adds the Credential to an authorise list for a Secret. Requires the use of either the master Credential or the customer Credential to complete
- View Credential - requires any Credential valid in the Secrets Vault.
- View / Remove Identity Key Pairs- View / Remove required an authorised Credential
- Add Identity Key Pairs - requires any Credential valid in the Secrets Vault.
- View / Add / Remove Secret - requires the correct Credential for the Secret

Service API

[0057] In one embodiment, a service REST API may provide endpoints for the primary functions of the Secrets Vault. All REST API methods may be secured using a JWT token that the client will need to generate, and the server will verify. An API such as the following may be implemented:

```
/vault/{vault name}/{entity type}/{action}
```

{vault name}	name of a vault db file
{entity type}	Credential, Identity Key Pair, or Secret
{action}	add, remove, get

[0058] With corresponding REST URL examples as follows:

```
/vault/soulmachines1/credential/add
/vault/rbs-qa/secret/add
/vault/autodesk-prod/secret/get
```

[0059] Figure 4 shows a schematic diagram of the components of the Secrets Vault 17.

Database Management & Schema Design

[0060] Any suitable backend storage mechanism may be used for the Secrets Vault, e.g. sqlite3 or Cassandra. The Secrets Vault design may allow for a single vault db file and multiple secrets db files. This provision facilitates the large volume of session keys that may be generated by allowing a new secrets db to be created at specific periods of time (e.g. monthly) preserving measurable end predictable performance (for I/O) and easier data management (backup). A distributed file system mechanism may provide db distribution, replication and snapshotting. Individual db files may be stored in a hierarchical directory structure to facilitate ease of management.

```
CredentialsDB Schema
CREATE TABLE IF NOT EXISTS Credentials (
ID INTEGER PRIMARY KEY,
       Owner TEXT NOT NULL,
       Name TEXT NOT NULL,
       Type TEXT,
       ValidationData BLOB,
       Signature BLOB,
       CreationDate DATETIME DEFAULT(DATETIME('now'))
);
CREATE INDEX IF NOT EXISTS Credentials_Owner ON Credentials (Owner, Name);

CREATE TABLE IF NOT EXISTS KeyData (
       ID INTEGER PRIMARY KEY,
       CredentialID INTEGER NOT NULL,
       Creator TEXT,
       Name TEXT,
       Algorithm TEXT,
       AuthType INTEGER,
       PublicKey BLOB,
       WrappedPrivateKey BLOB,
       KeyWrapper BLOB,
       ValidationData BLOB,
       CreationDate DATETIME DEFAULT(DATETIME('now'))
);
CREATE INDEX IF NOT EXISTS KeyData_Credential ON KeyData (CredentialID, Name);
CREATE INDEX IF NOT EXISTS KeyData_Name ON KeyData (Name);


SecretDB Schema

CREATE TABLE IF NOT EXISTS Secrets (
       ID INTEGER PRIMARY KEY,
       KeyID INTEGER NOT NULL,
       Name TEXT,
```

```
        Creator TEXT,
        DataType TEXT,
        SecretData BLOB,
        ValidationData BLOB,
        CreationDate NUMERIC DEFAULT(DATETIME('now'))
);

CREATE INDEX IF NOT EXISTS Secret_KeyData ON Secrets (KeyID, ID);
CREATE INDEX IF NOT EXISTS Secret_Name ON Secrets (Name);

CREATE TABLE IF NOT EXISTS SecretsKeyData (
        ID INTEGER PRIMARY KEY,
        KeyID INTEGER NOT NULL,
        SecretID INTEGER NOT NULL,
        KeyType TEXT,
        KeyData BLOB,
        CreationDate NUMERIC DEFAULT(DATETIME('now'))
);
CREATE INDEX IF NOT EXISTS Secret_Id ON SecretsKeyData (SecretID);
```

Threat Models

- Security considerations for the design of the secrets vault and securing the data contained within the various tables data repositories include mitigations for possible threats as follows:
- Threat: An attacker gains access to the backend storage mechanism using the remote REST interface. → Mitigation: Secrets servers are not accessible to the public internet. Secrets servers are deployed behind private VPC's. → Mitigation: Access to the REST interface requires authentication. Authentication is implemented using the credentials model specified, with all REST users requiring a valid credential for perform any action.
- Threat: Attacker adds a new credential to a backend storage mechanism → Mitigation: New credentials can only be added by the master credential or customer CPI
- Threat: Attacker overrides master or CPI credential → Mitigation: During creation the master and CPI credentials are created and added together, during that process each credential has a signature generated to confirm the integrity of the credential. → Mitigation: Even if the Master and CPI keys are able to be overridden, all existing data is still secure are the existing PPKP keys will not be accessible due to differences in the new credentials.
- Threat: Attacker attempts to brute force a session key to gain access to the raw data. → Mitigation: Only secure encryption algorithms are used (AES256 or better)
- Threat: Attacker compromises a credential gaining access to a set of the secured key data. → Mitigation: This is difficult to detect (other than audit). However if detected the credential can be replaced using the Master credential, and also remove the old wrapped PPKP.

Advantageous Effects

- Each Service Provider (which may be a customer of Master) is assigned a unique protection identity (Identity Key Pair).

- Each Service Provider has the ability to assign a protection mechanism/key(Credential) to their unique protection identity (Identity Key Pair), making it accessible only to them (optionally, even to the exclusion of the Master).
- Each authorised user has its own set of access rights and Credentials to Secrets. In other words, new encrypted private keys of Identity Key Pair are stored for each user with Credentials to access the Identity Key Pair.
- The Identity Key Pair is used to secure all keys used for protecting individual session data streams.
- Each session data stream can be assigned a unique Secret Key for the protection of sensitive data.
- Different sources or classifications of data may be classified and protected with independent Secret Keys.
- Secret Keys used in protecting sensitive session data are not accessible to non-authorised individuals.
- The protection process protects an individual (identified) user's data independently from other data streams of a customer.
- All session and customer identities and keys may be protected and stored in a secure, centralised database (Secrets Vault) that may be protected by access control mechanism as an additional protection step.
- Each authorized user has their own / independent set of credentials and access rights to material
- Integrity checks can be performed on sensitive data to ensure that no tampering has occurred.
- Encryption is cryptographically enforced, rather than depending on software logic roles / algorithms/ rules-based control.
- As only the private keys of Identity Key Pairs are wrapped, the insertion and encryption of new Secrets is possible without Credentials, however the subsequent decryption of the Secrets is only possible with someone with Credentials associated with corresponding Identity Key Pairs.
- The creation and revocation of access is facilitated by the notion of unique Credentials, without requiring any other aspects of the system to be changed. Once a user's Identity Key Pair is removed the user can no longer access any Secrets.
- The separation of Credentials and Identity Key Pair security protects against unauthorized access because access is granted cryptographically, prohibiting backdoor access.

INTERPRETATION

[0061] The methods and systems described may be utilised on any suitable electronic computing system. According to the embodiments described below, an electronic computing system utilises the methodology of the invention using various modules and engines.

[0062] The electronic computing system may include at least one processor, one or more memory devices or an interface for connection to one or more memory devices, input and output interfaces for connection to external devices in order to enable the system to receive and operate upon instructions from one or more users or external systems, a data bus for internal and external communications between the various components, and a suitable power supply. Further, the electronic computing system may include one or more communication devices (wired or wireless) for communicating with external and internal devices, and one or more input/output devices, such as a display, pointing device, keyboard or printing device.

[0063] The processor is arranged to perform the steps of a program stored as program instructions within the memory device. The program instructions enable the various methods of performing the invention as described herein to be performed. The program instructions, may be developed or implemented using any suitable software programming language and toolkit, such as, for example, a C-based language and compiler. Further, the program instructions may be stored in any suitable manner such that they can be transferred to the memory device or read by the processor, such as, for example, being stored on a computer readable medium. The computer readable medium may be any suitable medium for tangibly storing the program instructions, such as, for example, solid state memory, magnetic tape, a compact disc (CD-ROM or CD-R/W), memory card, flash memory, optical disc, magnetic disc or any other suitable computer readable medium.

[0064] The electronic computing system is arranged to be in communication with data storage systems or devices (for example, external data storage systems or devices) in order to retrieve the relevant data.

[0065] It will be understood that the system herein described includes one or more elements that are arranged to perform the various functions and methods as described herein. The embodiments herein described are aimed at providing the reader with examples of how various modules and/or engines that make up the elements of the system may be interconnected to enable the functions to be implemented. Further, the embodiments of the description explain, in system related detail, how the steps of the herein described method may be performed. The conceptual diagrams are provided to indicate to the reader how the various data elements are processed at different stages by the various different modules and/or engines.

[0066] It will be understood that the arrangement and construction of the modules or engines may be adapted accordingly depending on system and user requirements so that various functions may be performed by different modules or engines to those described herein, and that certain modules or engines may be combined into single modules or engines.

[0067] It will be understood that the modules and/or engines described may be implemented and provided with instructions using any suitable form of technology. For example, the modules or engines may be implemented or created using any suitable software code written in any suitable language, where the code is then compiled to produce an executable program that may be run on any suitable computing system. Alternatively, or in conjunction with the executable program, the modules or engines may be implemented using, any suitable mixture of hardware, firmware and software. For example, portions of the modules may be implemented using an application specific integrated circuit (ASIC), a system-on-a-chip (SoC), field programmable gate arrays (FPGA) or any other suitable adaptable or programmable processing device.

[0068] The methods described herein may be implemented using a general-purpose computing system specifically programmed to perform the described steps. Alternatively, the methods described herein may be implemented using a specific electronic computer system such as a data sorting and visualisation computer, a database query computer, a graphical analysis computer, a data analysis computer, a manufacturing data analysis computer, a business intelligence computer, an artificial intelligence computer system etc., where the computer has been specifically adapted to perform the described steps on specific data captured from an environment associated with a particular field.

REFERENCE SIGNS LIST

1 Agent
2 Secret
3 Private/Sensitive
4 Pseudonymous
5 Anonymized
6 Metadata
7 PPKP
8 Identity Key Pair
9 Secret Key Pair
10 Credential
11 Secret Key
12 Secrets Key DB
13 Data Vault
14 Cluster Management
15 Cluster
16 Data Stream Processor
17 Secrets Vault
18 Data Collector
19 Analytics Module
20 Analytics Task Engine
21 Agent Server
22 Node Server
23 Video Host
24 Storage Zone
25 Distributed File Storage
26 Reporting Module
27 Agent Cloud
28 Client Interface
29 Service Provider
30 Master
31 Event Data Stream
32 Video Stream
33 Audio Stream
34 Long Term Storage
35 Key Manager
36 Credentials Manager
37 Credentials Store
38 KeyPair Store
39 Secrets Store
40 Service API
41 Credentials Controller
42 PEM File
43 Service Key

CLAIMS

1. A method of cryptographically securing a Secret for access by an entity, the entity having a Credential and an associated Key Pair, the key pair including a private key and a public key, wherein the private key of the Key Pair is encrypted using the Credential, the method including the steps of:

   generating a Secret Key;
   symmetrically encrypting the Secret using the Secret Key; and
   asymmetrically encrypting the Secret Key using the public key of the Key Pair.

2. The method of claim 1 wherein the private key of the Key Pair is symmetrically encrypted using the Credential.

3. The method of claim 2 wherein the Secret is encrypted using AES192 or AES256.

4. The method of claim 3 wherein the Secret Key is encrypted using the RSA algorithm or the Elliptic Curve Digital Signature algorithm.

5. The method of claim 4 wherein the Secret is a data stream.

6. A method of cryptographically securing at least one Secret for access by an entity, the entity having a Credential and an associated public key, the method including the steps of:

   for each Secret, generating or receiving a Secret Key Pair including the associated public key and a new private key unique to the Secret;
   encrypting the private key of the Secret Key Pair using the Credential; and
   asymmetrically encrypting each Secret using the associated public key of the Secret Key Pair.

7. A method of cryptographically securing at least one Secret for access by an entity, the entity having a Credential and an associated Identity Key Pair, the Identity Key Pair including a public key and a private key, including the steps of:

   generating or receiving a Secret Key for encrypting the at least one Secret;
   encrypting the at least one Secret using the Secret Key;
   asymmetrically encrypting the Secret Key using the Identity Key Pair; and
   symmetrically encrypting the private key of the Identity Key Pair with the Credential.

8. Decrypting a Secret encrypted by any of the preceding claims.

9. A system for cryptographically securing Secrets collected by a Data Collector for access by an entity, such that the Secrets are unreadable by the Data Collector, the system including:

the entity, wherein the entity is associated with a Credential and an associated public key;

a Secret Key Pair generator configured to generate Secret Key Pairs for each Secret, wherein the Secret Key Pairs include the associated public key and a new private key unique to the Secret; and

the Data Collector configured to:

capture or receive each Secret,

encrypt the private key of each Secret Key Pair using the Credential; and

asymmetrically encrypt each Secret using the associated public key of the Secret Key Pair.

10. A system for cryptographically securing Secrets collected by a Data Collector for access by an entity, such that the Secrets are unreadable by the Data Collector, the system including:

the entity, wherein the entity is associated with a Credential and an associated Identity Key Pair;

a Secret Key generator configured to generate Secret Keys for each Secret, and

the Data Collector configured to:

capture or receive each Secret,

asymmetrically encrypt each Secret using the Identity Key Pair; and

symmetrically encrypt the private key of the Identity Key Pair with the Credential.

DRAWINGS

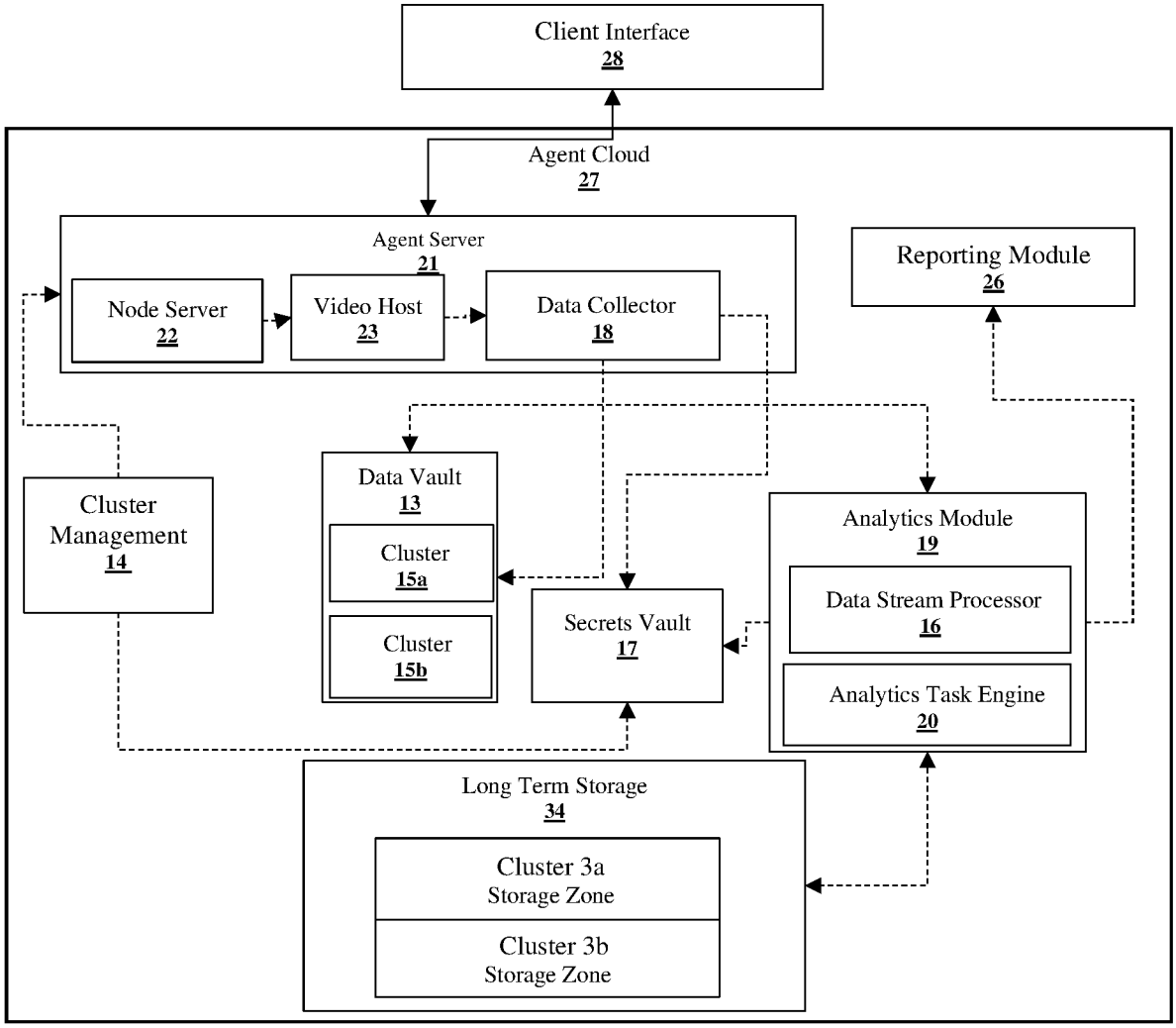

FIGURE 1

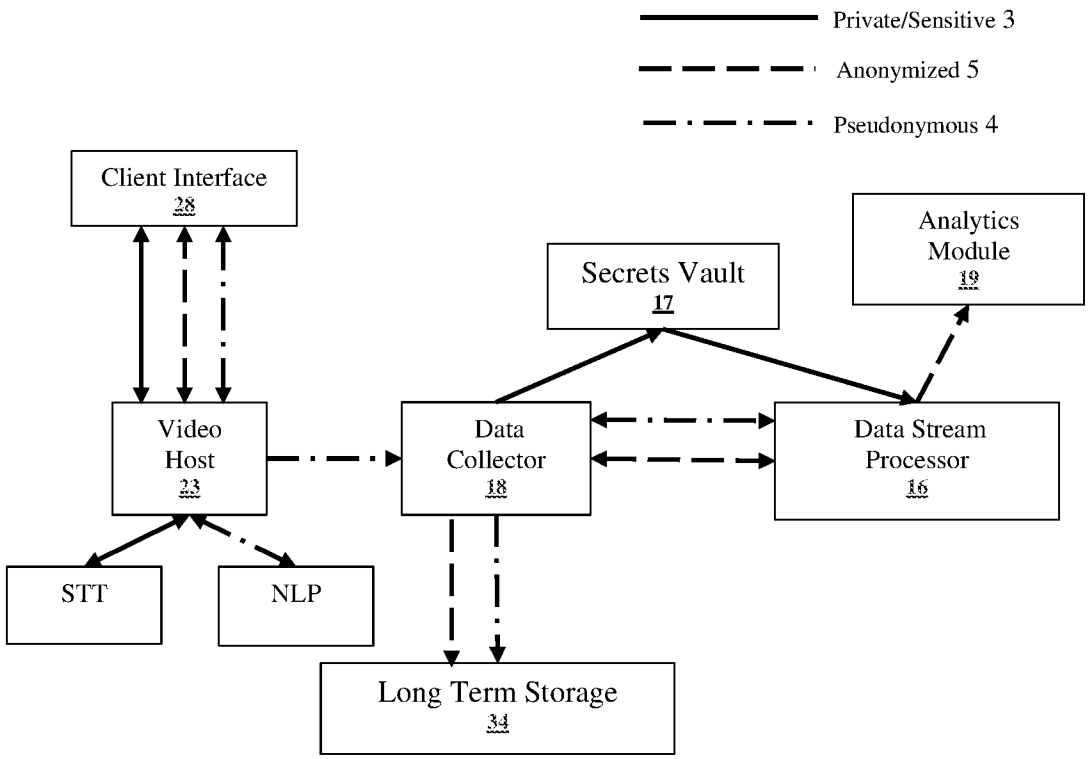
Private/Sensitive 3
Anonymized 5
Pseudonymous 4
Client Interface
28
Secrets Vault
17
Analytics
Module
19
Video
Host
23
Data
Collector
18
Data Stream
Processor
16
STT
NLP
Long Term Storage
34

FIGURE 2

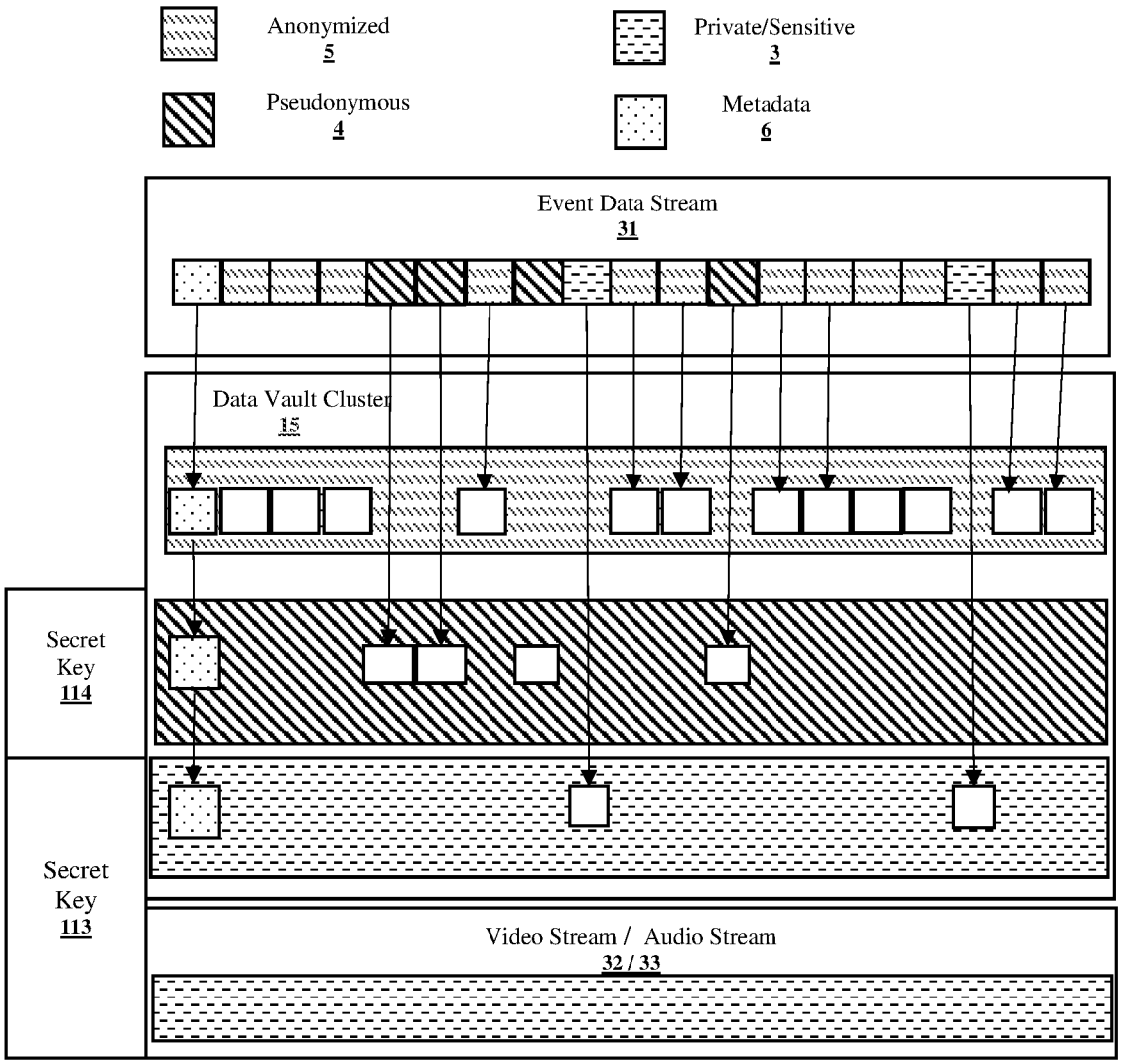
Anonymized
5
Private/Sensitive
3
Pseudonymous
4
Metadata
6
Event Data Stream
31
Data Vault Cluster
15
Secret
Key
114
Secret
Key
113
Video Stream / Audio Stream
32 / 33

FIGURE 3

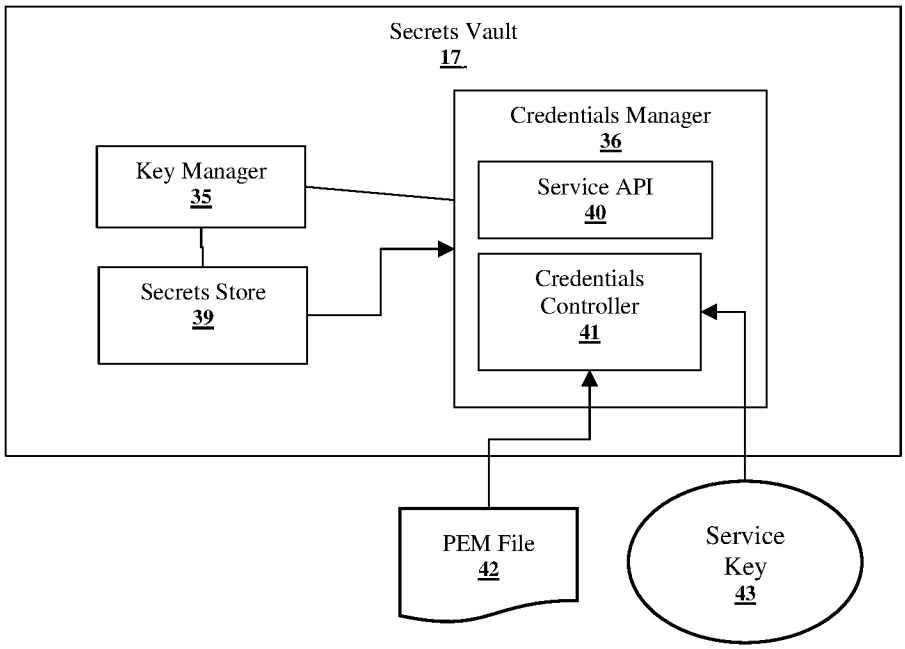

FIGURE 4

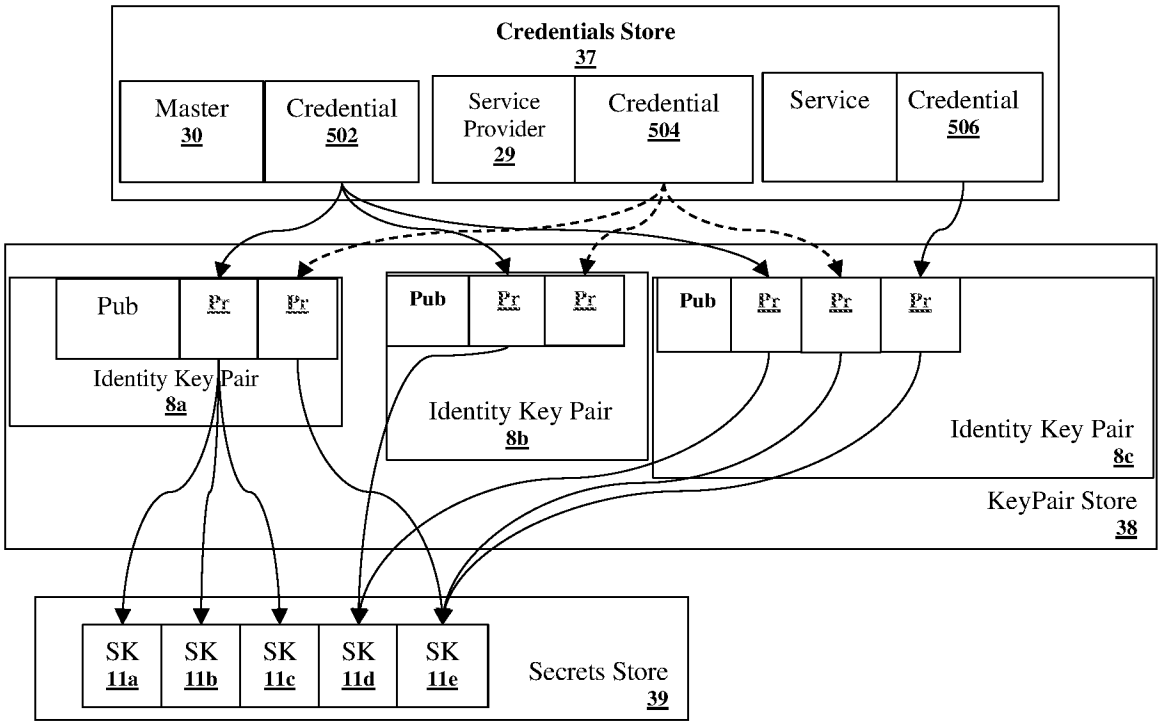

FIGURE 5

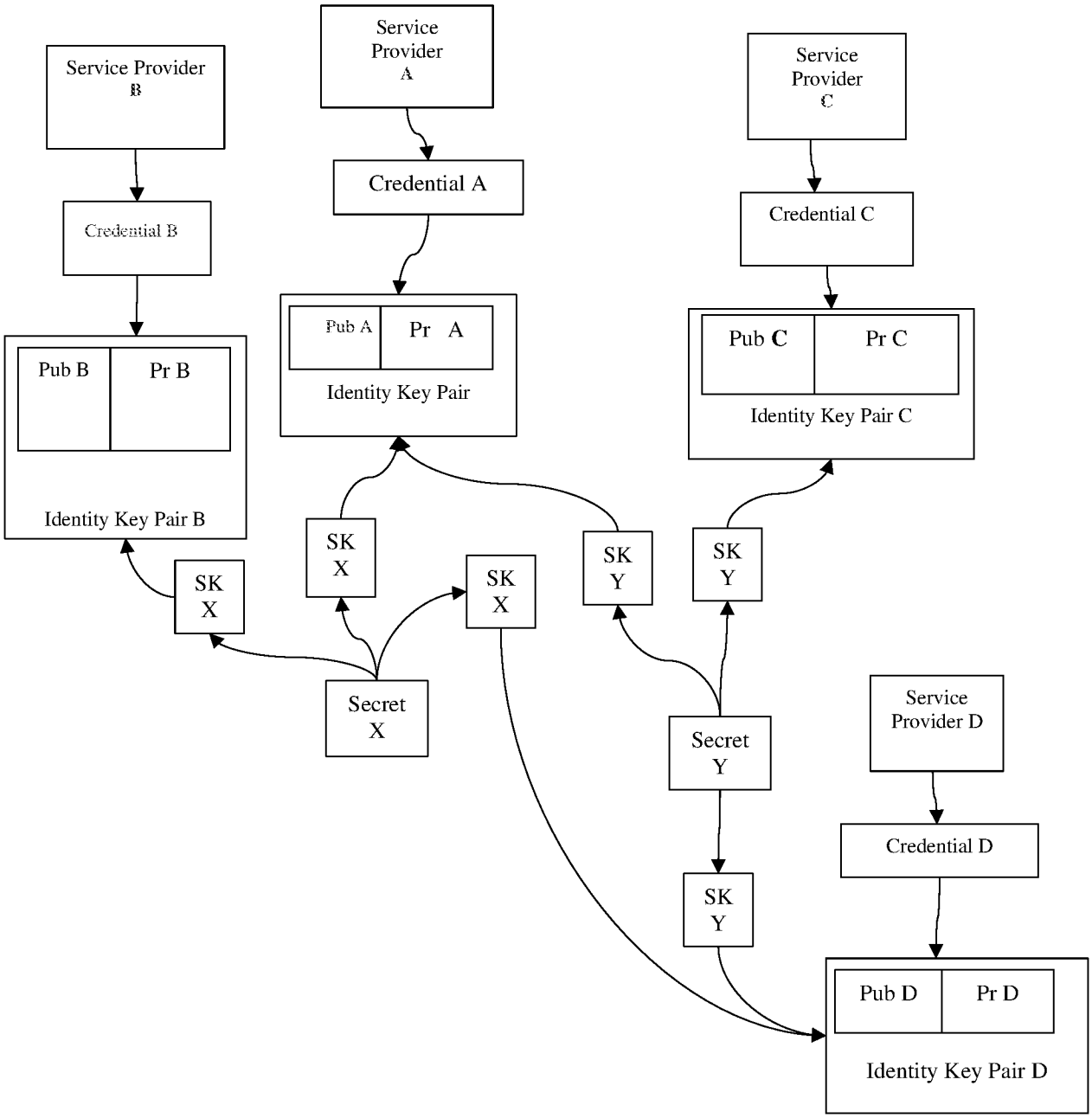
Service Provider B
Credential B
Pub B
Pr B
Identity Key Pair B
SK X
Service Provider A
Credential A
Pub A
Pr A
Identity Key Pair
SK X
Secret X
SK X
SK Y
Service Provider C
Credential C
Pub C
Pr C
Identity Key Pair C
SK Y
Secret Y
SK Y
Service Provider D
Credential D
Pub D
Pr D
Identity Key Pair D

FIGURE 6

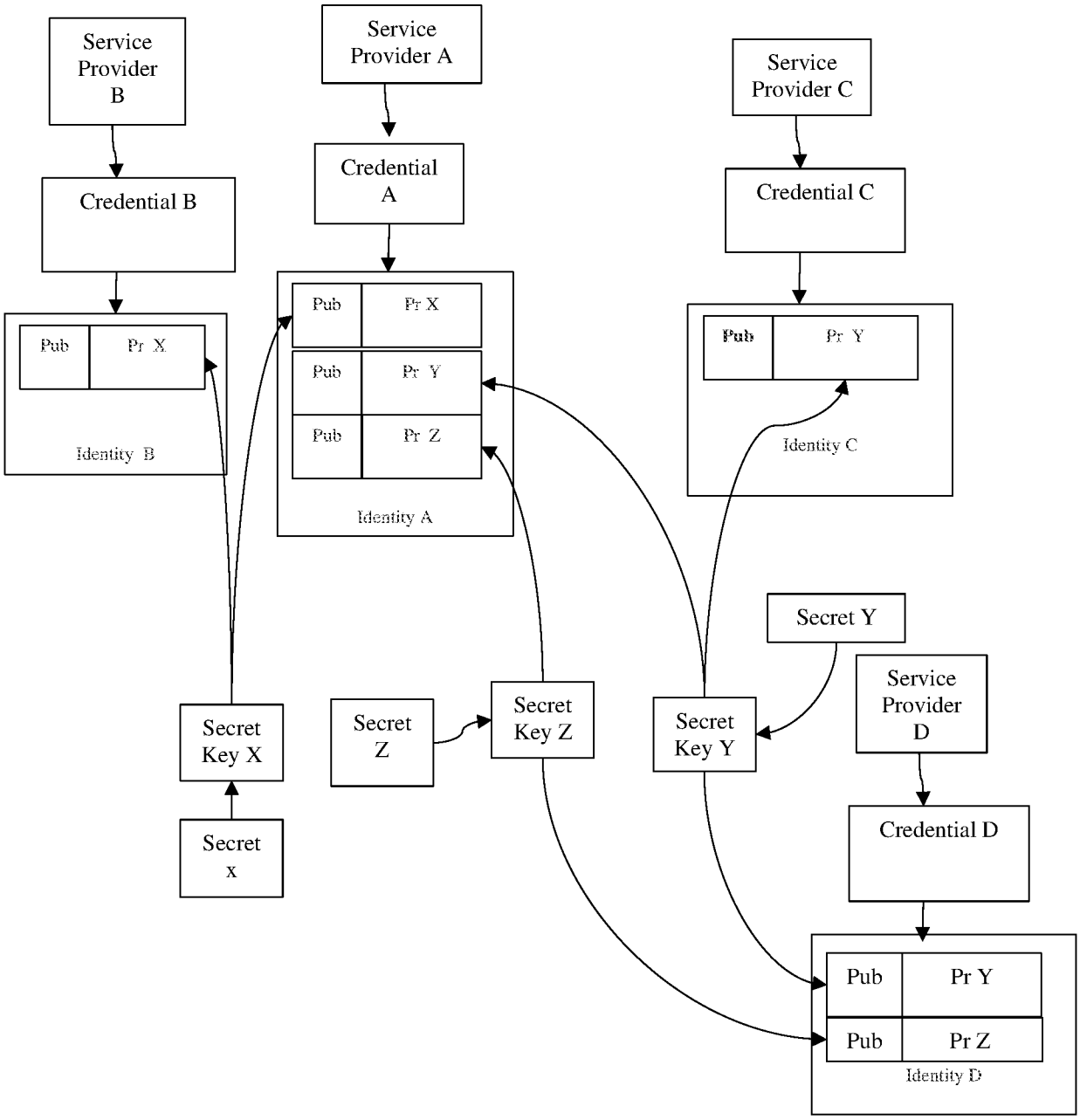
Service Provider B
Credential B
Pub
Pr X
Identity B
Service Provider A
Credential A
Pub
Pr X
Pub
Pr Y
Pub
Pr Z
Identity A
Service Provider C
Credential C
Pub
Pr Y
Identity C
Secret Key X
Secret x
Secret Z
Secret Key Z
Secret Key Y
Secret Y
Service Provider D
Credential D
Pub
Pr Y
Pub
Pr Z
Identity D

FIGURE 7

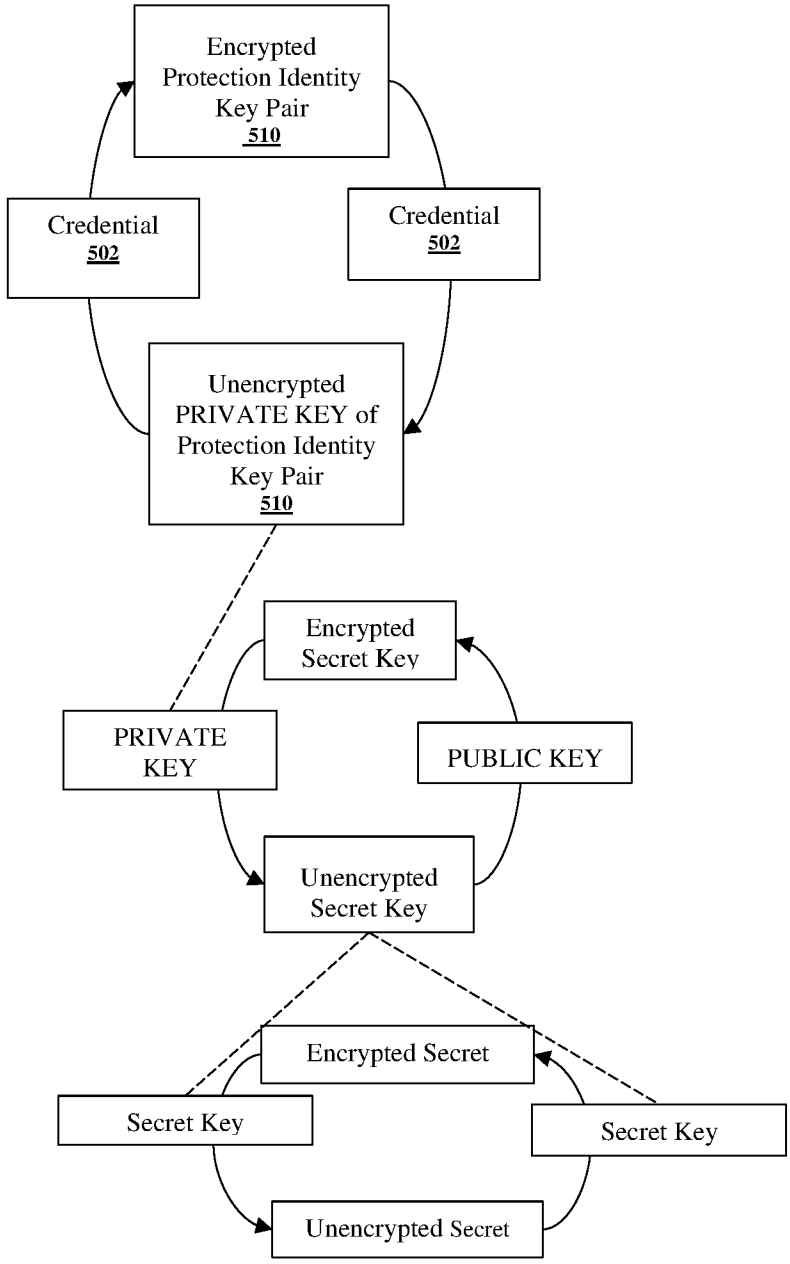
Encrypted Protection Identity Key Pair 510
Credential 502
Credential 502
Unencrypted PRIVATE KEY of Protection Identity Key Pair 510
Encrypted Secret Key
PRIVATE KEY
PUBLIC KEY
Unencrypted Secret Key
Encrypted Secret
Secret Key
Secret Key
Unencrypted Secret

FIGURE 8

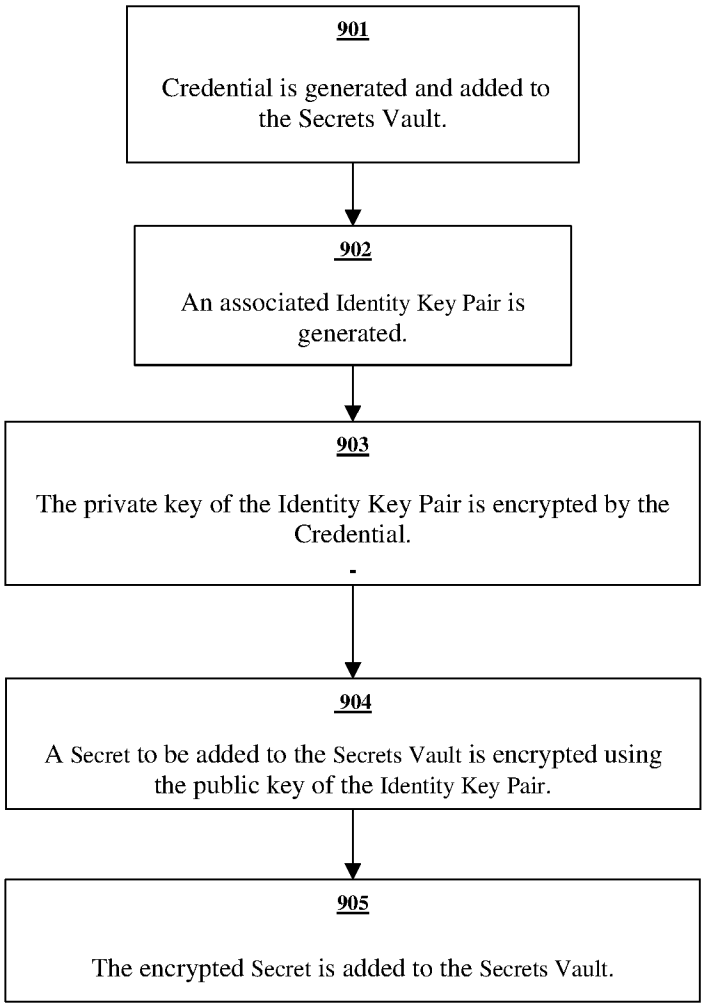
901
Credential is generated and added to the Secrets Vault.
902
An associated Identity Key Pair is generated.
903
The private key of the Identity Key Pair is encrypted by the Credential.
904
A Secret to be added to the Secrets Vault is encrypted using the public key of the Identity Key Pair.
905
The encrypted Secret is added to the Secrets Vault.

FIGURE 9

INTERNATIONAL SEARCH REPORT

International application No.
PCT/IB2020/053032

A. CLASSIFICATION OF SUBJECT MATTER

H04L 9/06(2006.01)i, H04L 9/08(2006.01)i, H04L 9/30(2006.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04L 9/06; G06F 12/14; G06Q 20/32; G06Q 20/38; H04L 9/08; H04L 9/30; H04L 9/32; H04W 12/04; H04W 72/04

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & keywords: cryptographically securing, public key, private key, encrypting, symmetric, asymmetric, credential

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2006-0101287 A1 (GLENN A. MORTEN) 11 May 2006 paragraphs [0024], [0029], [0049], [0061], [0070]; and claim 1	1,8
Y		2-7,9-10
Y	US 2014-0143548 A1 (DONGLIN WANG) 22 May 2014 claim 1	2-7,9-10
A	US 10169587 B1 (JOHN A. NIX) 01 January 2019 column 42 line 54 - column 65 line 11	1-10
A	US 2015-0372811 A1 (ERIC LE SAINT et al.) 24 December 2015 paragraphs [0081]-[0129]; and figures 4-5	1-10
A	US 2015-0213433 A1 (APPLE INC.) 30 July 2015 paragraphs [0032]-[0109]; and figure 5	1-10

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"D" document cited by the applicant in the international application
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents,such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
16 September 2020 (16.09.2020)	**17 September 2020 (17.09.2020)**

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea
Facsimile No. +82-42-481-8578

Authorized officer
YANG JEONG ROK
Telephone No. +82-42-481-5709

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/IB2020/053032

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2006-0101287 A1	11/05/2006	CA 2559323 A1	30/09/2004
		CN 1761948 A	19/04/2006
		EP 1606715 A2	21/12/2005
		JP 2006-524007 A	19/10/2006
		KR 10-0749947 B1	21/08/2007
		KR 10-2005-0119122 A	20/12/2005
		US 2004-0184616 A1	23/09/2004
		US 2006-0143481 A1	29/06/2006
		US 7007170 B2	28/02/2006
		US 7356143 B2	08/04/2008
		US 7640435 B2	29/12/2009
		WO 2004-084035 A2	30/09/2004
		WO 2004-084035 A3	10/03/2005
		WO 2007-100975 A2	07/09/2007
		WO 2007-100975 A3	29/11/2007
US 2014-0143548 A1	22/05/2014	CN 102629940 A	08/08/2012
		CN 103237040 A	07/08/2013
		CN 103237040 B	13/04/2016
		CN 103812927 A	21/05/2014
		EP 2830282 A1	28/01/2015
		EP 2830282 B1	15/03/2017
		US 2013-0246790 A1	19/09/2013
		US 2013-0246811 A1	19/09/2013
		US 2014-0075193 A1	13/03/2014
		US 9164926 B2	20/10/2015
		WO 2013-139079 A1	26/09/2013
US 10169587 B1	01/01/2019	US 10621352 B2	14/04/2020
		US 2019-0332774 A1	31/10/2019
		US 2020-0226258 A1	16/07/2020
		WO 2019-209598 A1	31/10/2019
US 2015-0372811 A1	24/12/2015	AU 2015-277000 A1	23/12/2015
		AU 2015-277000 B2	16/05/2019
		AU 2019-216646 A1	05/09/2019
		CN 106664206 A	10/05/2017
		CN 106664206 B	12/05/2020
		CN 111355749 A	30/06/2020
		EP 3158680 A1	26/04/2017
		US 10574633 B2	25/02/2020
		US 2020-0092269 A1	19/03/2020
		WO 2015-195978 A1	23/12/2015
US 2015-0213433 A1	30/07/2015	None	