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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
8 December 2016 (08.12.2016)

WIPO | PCT

(10) International Publication Number
WO 2016/193962 A1

(51) International Patent Classification:
G06F 21/62 (2013.01) *H04L 9/08* (2006.01)

(21) International Application Number:
PCT/IL2015/050564

(22) International Filing Date:
2 June 2015 (02.06.2015)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **K2VIEW LTD** [IL/IL]; 52 Bar-Yehuda St., P.O box 230, Nesher (IL).

(72) Inventors: **ITAMAR, Einav**; 3 Sold St., Naharia (IL). **ROTEM, Achi**; 52 Bar-Yehuda St., P.O box 230, Nesher (IL).

(74) Agent: **DRORI, Yonatan**; Drori - Werzansky - Orland, Law Firm, 23 Menachem Begin, 66183 Tel Aviv (IL).

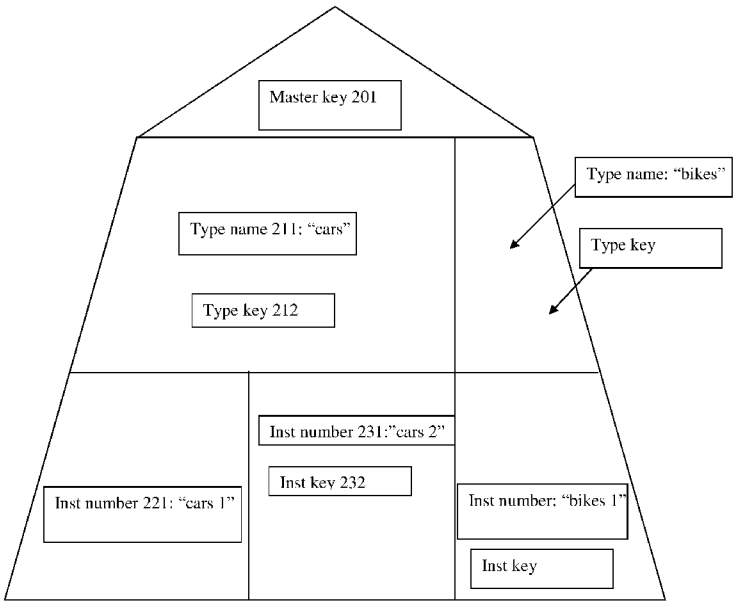
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP,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

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

(54) Title: ENCRYPTION DIRECTED DATABASE MANAGEMENT SYSTEM AND METHOD



(57) Abstract: A granular encryption database management system characterized by hierarchical internal key management using asymmetric encryption of the user's keys, which are stored within a (local) encryption key table. Data is encrypted, decrypted, stored, and accessed from rows within the database (hereinafter 'instances'). Instances are ordered in categories called Types. Instances are encrypted with Instance keys. All instances of the same type have their Instance keys generated by means of a hash function of their common Type's key, while Type keys are generated from the key of the higher category which is often the Master key of the whole system.

Fig. 2

WO 2016/193962 A1

Encryption directed database management system and method

FIELD OF THE INVENTION

The present invention relates to the field of database management systems and more particularly database management systems and methods directed towards content encryption.

BACKGROUND OF THE INVENTION

Today, many organizations store their data in organization-wide databases. Often, the data stored is very sensitive and may contain commercial secrets, costumers' personal data, or otherwise sensitive data which may cause serious damage to the organization if leaked.

The danger of data leakage grows where the organization has geographically separate offices that connect to the organization's database with the use of regular internet infrastructure.

To mitigate the risk for compromising sensitive data, amongst various data security provisions, many organizations encrypt their databases using encryption-directed database management systems. Such systems enable the organization to encrypt and decrypt the data with the use of various known algorithms.

In most encryption methods, secret keys are used to decrypt and access the encrypted data. The database management systems enable the creation, storage, retrieval and security of the keys which serve for encryption of all data within the encrypted database.

Internal vs. External encryption key management systems

Different database management systems manage their keys in different ways. Key management using only the server of the encrypted database itself is called internal key management, while key management using a dedicated external key management server is called external key management.

Various internal key management modules handle the keys in different ways. The keys are usually stored in a separate table within the administrative database within the server of the encrypted database. Some methods and systems allow users to simply access their keys, others encrypt the key table with a single master key, while others perform key operations through a proxy.

Internal key management is simple and doesn't require an external server, driving down operation costs. Nevertheless, there are drawbacks to internal key management: (a) database administrators have access to the whole key table thus the whole system is vulnerable at the level of the individual system administrator; (b) upon database breach, the trespasser may easily gain access to the whole key table by hacking the master key of the encrypted key table, or otherwise by sniffing the various keys being used outside the encrypted key table.

External key management is often considered safer than internal key management, since such systems use separate dedicated servers and prevent any association of the keys with the encrypted database, thus also preventing access to the keys by system administrators. The requirement of a second dedicated server however may prove problematic; for example, outsourcing the key server responsibilities to third parties suffers from frequently unknown reliability levels.

Granularity

Another shortcoming in known systems involves the granularity of the table encryption. Certain methods (e.g., transparent encryption in MSSQL) allow the encryption of entire tables within the database while other methods (e.g., granular encryption in MSSQL) allow the encryption of specific fields or columns thereof. One must trade off security for simplicity; it is simpler when entire tables are encrypted and a minimal number of keys are created and managed, but more secure when every field is encrypted (generally requiring a dedicated key management system for effective management of the large number of keys).

As should be clear from the brief review above there is no system that combines granular encryption of the database with simple and straight forward key management. Such a system would ideally obviate the need for dedicated key management systems and further allow for internal management of a granularly encrypted database and its keys.

Permissions to access data

Another shortcoming in known systems involves permissions to access data in the database. In known systems, permissions are granted using a different process from the key granting process and from the decryption process. Permissions and encryption keys for accessing allowed data are accomplished separately.

Although the use of permissions has evolved naturally independent of issues of encryption, permissions still persist where encryption is required. For the sake of simplification there is a drive to provide a system that through encryption management, obviates the need for a separate permission system.

Background synopsis

Therefore, there is a pressing need to provide an encryption directed database management system that allows:

(a) internal management of keys within the server of the encrypted database itself preventing costs and risks related to external key management;

(b) high security standards for the key table that is located within the database server, preventing server administrators from accessing the keys in the key table, and further having higher security than systems using a single master key for accessing a common key table;

(c) granular encryption of the database involving different keys for the columns and rows within; as mentioned above in this case if a certain encrypted field or row is compromised, data stored in fields and rows having a different key remain secure; and,

(d) obviating the need for a separate permission system through encryption management.

SUMMARY OF THE INVENTION

The present invention provides a granular encryption database management system characterized by hierarchical internal key management using asymmetric encryption of the user's keys, which are stored within a (local) encryption key table.

Data is encrypted, decrypted, stored, and accessed from rows within the database (hereinafter 'instances'). Instances are ordered in categories called Types.

Instances are encrypted with Instance keys. All instances of the same type have their Instance keys generated by means of a hash function of their common Type's key and their instance identifier, while Type keys are generated from the key of the higher category and their type identifier. The higher category key is often the Master key of the whole system.

The master key is preferably a random string of at least 256 bit length, which is generated during the system installation, based on the credentials of the system's super user.

The term 'master user' hereunder refers to a database user that has received access to a master key either by the system upon system installation, or otherwise by another master user using access granting method (see below).

The Type key is preferably a hash function of the Master key and the type's identifier, this being a necessarily unique identifier for the type and preferably comprising the string name of the type. For example: Type key= Hash[Master key, str(type_name)].

Similarly, the Instance key is a hash function of the Type key and the Instance's identifier. The Instance identifier is a unique identifier for an instance and preferably comprises the serial number of the instance. For example: Instance key= Hash[Type key, ID(instance)].

The system provides a method for hierarchical and granular database encryption that comprises the following steps:

(a) Upon system installation, the system generates a master key. This step is done concurrently with the creation of the first system user, since each key is encrypted based on its owner's password (see below).

(b) Upon creation of a type within the database for administration of individual instances, a Type key is generated from the Master key and the Type identifier by means of (for example) a hash function of these two values.

(c) Upon creation of an instance of a type for data storage, an Instance key is generated (e.g. hashed) from the Type key and the Instance identifier.

The encryption key table lists users in rows while having a list of columns for every user that is comprised of: a plurality of encryption key columns ('wallet'), private key column and public key column. It is to be mentioned that users' passwords may be managed by any means such as, external services, or encrypted and stored in a column in the encryption key table.

The present invention further provides a method for new user registration with the system:

(a) Upon user registration, the user chooses a password.

(b) Password is stored.

(c) The system generates a public/private key pair for said user.

(d) The public/private key pair is stored within the encryption key table.

(e) The private key is encrypted using the user's password as a passphrase (key) for the encryption.

(f) If user is a 'first user' (or 'super user'), then system generates the Master key and stores it in said user's wallet and encrypts it with said user's public key.

The present invention further provides a method for a master user to grant access to a type for grantee system users:

(a) Master user decrypts his Private key with His password.

(b) Master user decrypts his Master key with His Private key.

(c) Master user hashes the Type key for the Type with the Master key and the Type identifier (e.g., string of the type name).

(d) Master user encrypts the type key with the grantee user's public key.

(e) Master user stores the encrypted Type key in the grantee's wallet.

The present invention further provides a method for granting permission access to an instance through key granting for grantee system users:

(a) Grantor decrypts his Private key with his password.

(b) Grantor decrypts his key with his Private key.

(c) if key is a Master key, grantor hashes a Type key for the instance with the Master key and the Type identifier

(d) Grantor hashes the Instance key for the Instance with the Type key and the Instance identifier (e.g., instance's serial number).

(e) Grantor stores the Instance key in the grantee's wallet.

(f) Grantor encrypts the stored Instance key with the grantee's public key.

It is to be mentioned that for the method of granting access to an instance, the grantor must either be granted access to a type key, or to have access to the master key (as a master user) and then generate the type key from the master key.

As was described above, the present invention holds many advantages over known systems and methods, and amongst others, provides an encryption directed database management system that allows:

(a) internal management of keys within the server of the encrypted database itself preventing costs and risks related to external key management;

(b) highly secure key table that is located within the database server, preventing server administrators from accessing the keys in the key table, and further having higher security standards than having a single master key for accessing the key table;

(c) granular encryption of the database involving multitude of different keys for the cells within, whereby if a certain encrypted cell is compromised, the sensitive data stored within the fields and rows having a different key, remains secure; and

(d) through encryption management, obviating the need for a separate permission system.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments and features of the present invention are described herein in conjunction with the following drawings:

Fig. 1 is a simplified depiction of a database managed by a system consistent with current practice;

Fig. 2 is an illustration of the granular hierarchical character of a system consistent with the invention;

Fig. 3 is a simplified depiction an encryption key table consistent with the invention;

Fig. 4 is a block diagram depicting a method for hierarchical and granular database encryption consistent with one embodiment of the invention.

Fig 5 is a block diagram depicting a method for new user registration with a system consistent with one embodiment of the invention.

Fig 6 is a block diagram depicting a method for granting access to a type for other users in a system consistent with one embodiment of the invention.

Fig 7 is a block diagram depicting a method for granting access to an instance for other users in a system consistent with one embodiment of the invention.

It should be understood that the drawings are not necessarily drawn to scale.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

Fig 1 illustrates a simple mortgage table within a bank database, and an encryption key table. The mortgage column contains sensitive data and therefore is stored in encrypted form within the table. For the purposes of this example consider the bank uses a simple internal key management system, whereby the keys are stored and managed in a table on the database server. The keys are further encrypted by a master key stored so as to be accessible to authorized bank personnel.

As an example transaction of how many current systems work, we take the case that Admin 1, a bank functionary, inputs a query involving the mortgage of Mr. John Doe.

To access the data requested, admin 1 inputs the key for decrypting his encryption key while also unintentionally decrypting admin 2 keys. The system decrypts the requested data but also allows Admin 1 to decrypt information of admin 2, such as Ms. Roe's data.

Although the above is an overly simplified illustration, it shows how many existing systems, even such of much higher complexity, suffer from crucial security flaws whereby a user gains the means to decrypt an entire table (or even the entire database) when he actually

needed only to access one piece of information. In this case, the database permission system allows admin 1 to access only allowed data. If a trespasser gains access to the encryption key, he will not be affected by the permission system and will gain access to the whole database.

Referring to Fig 2. The master key **201** is a random string of at least 256 bit length. It holds all access to the database since all other keys are hashed from the Master key.

The Type key **212** is a hash function output of the Master key **201** and the type's identifier. The type's identifier is a unique identifier of the type and preferably the string name of the type **211**. For example: Type key= Hash[Master key **201**, str("cars" **211**)].

The Instance key **222** is a hash function output of the Type key **212** and the Instance's identifier. The Instance identifier is a unique identifier of the instance and preferably the serial number of the instance. For example: Instance key= Hash[Type key **212**, Num("1" **221**)].

Referring to Fig 3. The encryption key table lists users in rows while having a list of columns for every user that is comprised of: a plurality of encryption key columns ('wallet'), private key column, public key column.

Referring to Fig 4. The system provides a method for hierarchical and granular database encryption that is comprised of the following steps:

(a) Upon initial system installation, the system generates a master key.

(b) Upon creation of a type within the database for administration of individual instances, a Type key is generated (hashed) from the Master key and the Type identifier.

(c) Upon creation of an instance within a type for data storage, an Instance key is generated (hashed) from the Type key and the Instance identifier.

Referring to Fig 5. The present invention further provides a method for new user registration with the system:

(a) Upon user registration, the user chooses a password.

(b) Password is stored.

(c) The system generates a public/private key pair for said user.

(d) The public/private key pair is stored within the encryption key table.

(e) The private key is encrypted using the user's password as a passphrase (key) for the encryption.

(f) If user is first user, system generates the Master key and stores it in said user's wallet and encrypts it with said user's public key.

Referring to Fig 6. The present invention further provides a method for a master user to grant access to a type for grantee system users:

(a) Master user decrypts his Private key with His password.

(b) Master user decrypts his Master key with His Private key.

(c) Master user hashes the Type key for the Type with the Master key and the Type identifier (e.g., string of the type name).

(d) Master user stores the Type key in the grantee's wallet.

(e) Master user encrypts the stored type key with the grantee user's public key.

Referring to Fig 7. The present invention further provides a method for granting permission access to an instance through key granting for grantee system users:

(a) Grantor decrypts his Private key with His password.

(b) Grantor decrypts his key with His Private key.

(c) If grantor's key is a Master key, grantor hashes a Type key for the instance with the Master key and the Type identifier.

(d) Grantor hashes the Instance key for the Instance with the Type key and the Instance identifier (e.g., instance's serial number).

(e) Grantor stores the Instance key in the grantee's wallet.

(f) Grantor encrypts the stored Instance key with the grantee's public key.

The present invention will be understood from the following detailed description of preferred embodiments, which are meant to be descriptive and not limiting. For the sake of brevity, some well-known features, methods, systems, procedures, components, circuits, and so on, are not described in detail.

The foregoing embodiments of the invention have been described and illustrated in conjunction with systems and methods thereof, which are meant to be merely illustrative, and not limiting. Furthermore just as every particular reference may embody particular methods/systems, yet not require such, ultimately such teaching is meant for all expressions notwithstanding the use of particular embodiments.

Any term that has been defined above and used in the claims, should be interpreted according to this definition.

The reference numbers in the claims are not a part of the claims, but rather used for facilitating the reading thereof. These reference numbers should not be interpreted as limiting the claims in any form.

CLAIMS

1. A granular encryption database management system having hierarchical internal key management using asymmetric encryption of the keys, wherein, data is encrypted, decrypted, stored, and accessed from Instances which are ordered in Types, and further wherein instances are encrypted with Instance keys that are generated by hashing their Type key and their identifier, while Type keys are generated by hashing the Master key and their identifier, wherein said keys are stored in an encryption key table.

2. The system of claim 1 wherein said encryption key table lists users in rows while having a list of columns that is comprised of: a plurality of encryption key columns ('wallet'), a private key column, a public key column.

3. A method for hierarchical, granular database encryption by the system of claim 1 comprising the following steps:

    a. upon system installation, the system generates a master key;

    b. upon creation of a type within the database for administration of individual instances, a Type key is generated by hashing the Master key and the Type identifier;

    c. upon creation of an instance within a type for data storage, an Instance key is generated by hashing the Type key and the Instance identifier.

4. A method for new user registration in an encrypted database system by the system of claim 1 comprising steps of:

    a. upon user registration, the user chooses a password;

    b. password is stored;

c. the system generates a public/private key pair for said user;

d. the public/private key pair is stored within the encryption key table;

e. The private key is encrypted using the user's password as a passphrase (key) for the encryption;

f. If user is first user, system generates the Master key and stores it in super user's wallet and encrypts it with super user's public key.

5. A method for Master user to grant access to fields of a granularly encrypted database by the system of claim 1 comprising steps of:

a. Master user decrypts his Private key with his password;

b. Master user decrypts his Master key with his private key;

c. Master user hashes the Type key for the Type with the Master key and the Type identifier;

d. Master user stores the Type key in the grantee's wallet;

e. Master user encrypts the stored type key with the grantee user's public key.

6. A method for granting permission access to a granularly encrypted database by the system of claim 1 using a key granting system for grantee users comprising steps of:

a. grantor decrypts his Private key with his password;

b. grantor decrypts his key with his Private key;

c. if key is a Master key, grantor hashes a Type key for the instance with the Master key and the Type identifier;

d. grantor hashes the Instance key for the Instance with said Type key and the Instance identifier;

e. Grantor stores the Instance key in the grantee's wallet;

f. Grantor encrypts the stored Instance key with the grantee's public key;

325

Mortgage TABLE/DB

CID	*Last Name*	*First Name*	*Mortgage Due…*
1	Doe	John	$#dfjh423
2	Roe	Jane	$#dopf4gr
…			

Encryption key table

Database admin	*Encrypted keys*
Admin 1	q7e345632
Admin 2	kerq14879

Fig. 1

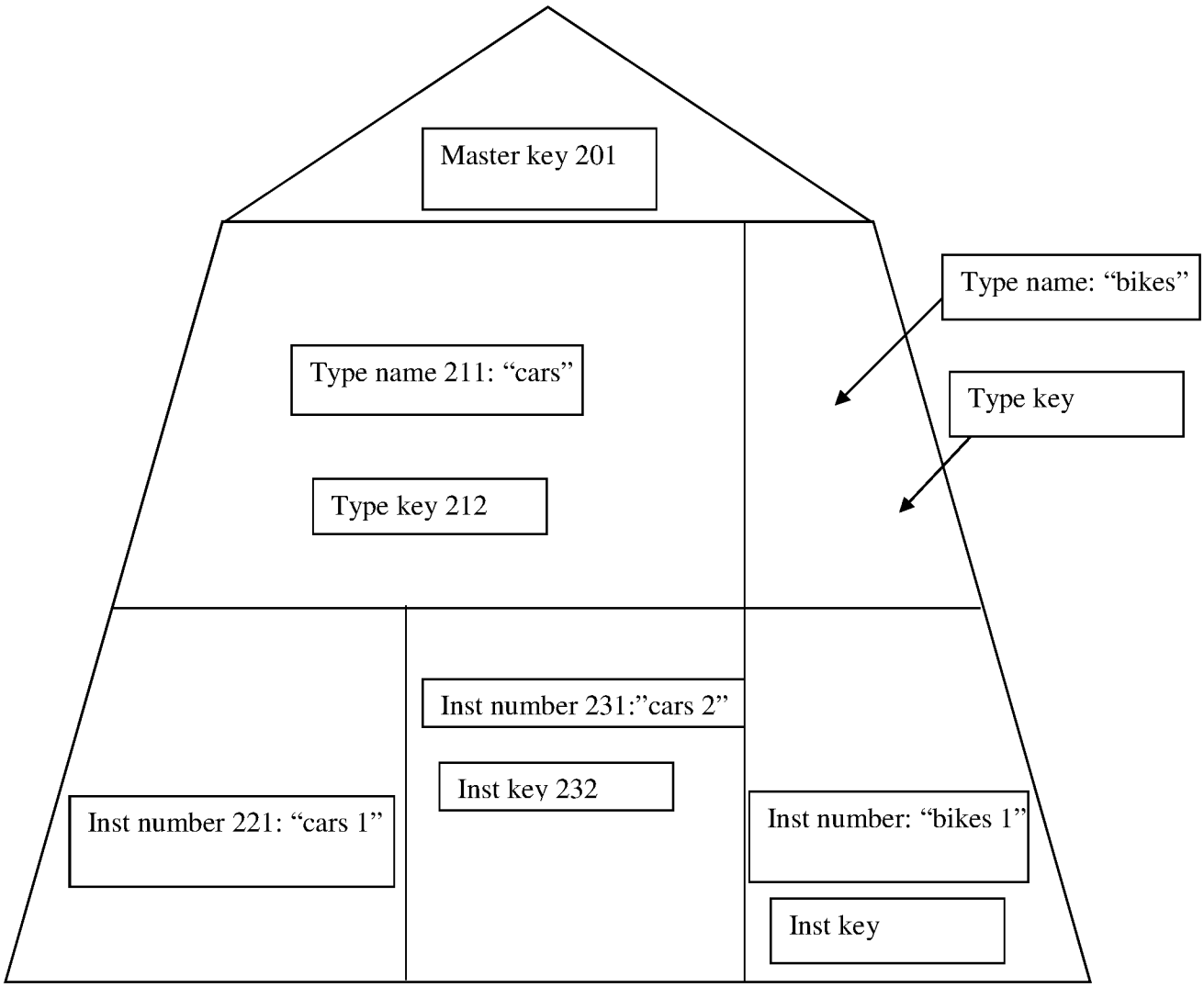

Master key 201
Type name: “bikes”
Type name 211: “cars”
Type key
Type key 212
Inst number 231:”cars 2”
Inst key 232
Inst number 221: “cars 1”
Inst number: “bikes 1”
Inst key

Fig. 2

User name	Private key	Public key	Encryption keys
John Doe	#hjcf2515	123547dvd	Smgio4w90;
Jane Doe	%ld5489	13297bvb	Asbjh37456; Aigfhjdf55fr;

Fig. 3

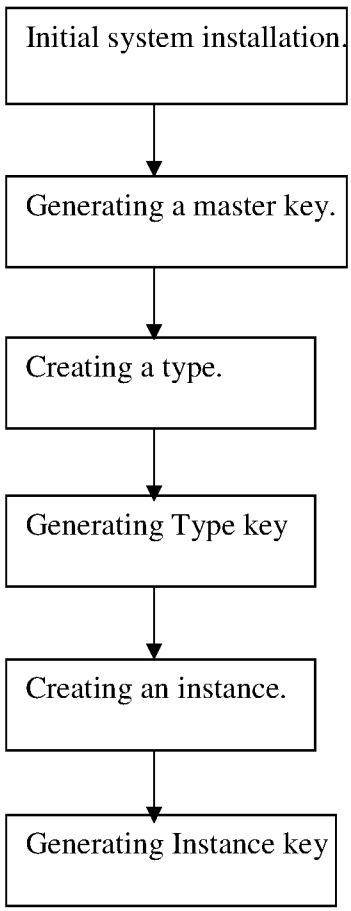

Initial system installation.
Generating a master key.
Creating a type.
Generating Type key
Creating an instance.
Generating Instance key

Fig. 4

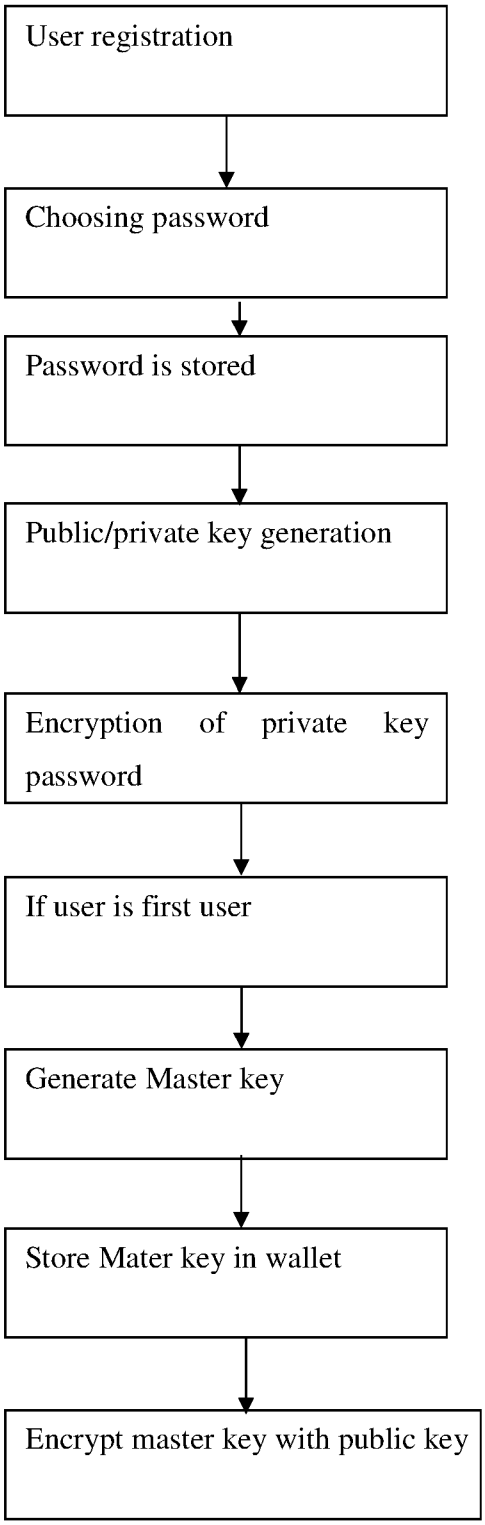

User registration
Choosing password
Password is stored
Public/private key generation
Encryption of private key password
If user is first user
Generate Master key
Store Mater key in wallet
Encrypt master key with public key

Fig. 5

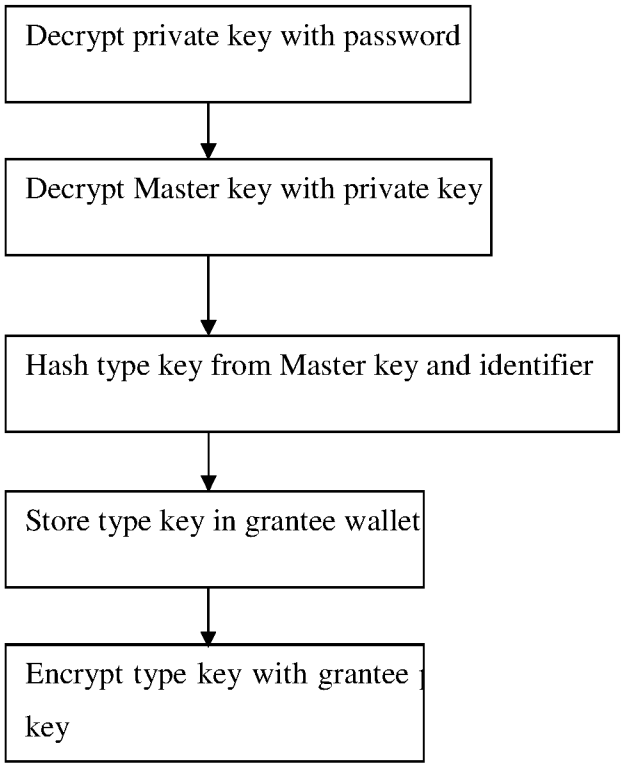

Decrypt private key with password
Decrypt Master key with private key
Hash type key from Master key and identifier
Store type key in grantee wallet
Encrypt type key with grantee
key

Fig. 6

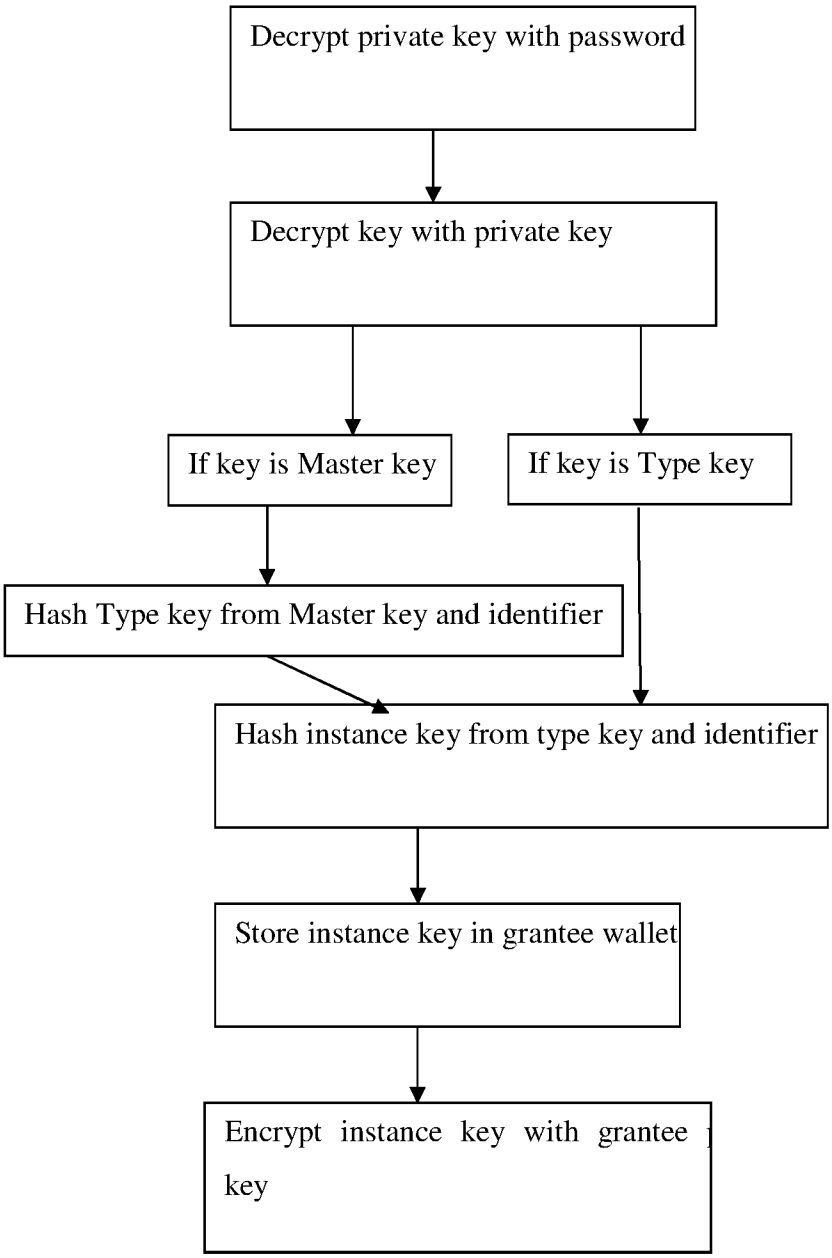

Decrypt private key with password
Decrypt key with private key
If key is Master key
If key is Type key
Hash Type key from Master key and identifier
Hash instance key from type key and identifier
Store instance key in grantee wallet
Encrypt instance key with grantee key

Fig. 7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/IL2015/050564

A. CLASSIFICATION OF SUBJECT MATTER
IPC (2015.01) G06F 21/62, H04L 9/08

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC (2015.01) G06F 21/62, H04L 9/00

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

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
Databases consulted: Esp@cenet, Google Patents, FamPat database
Search terms used: encrypt, granular, database, row, column, field, cell, hash, key management,

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2008077806 A1 CUI et al. 27 Mar 2008 (2008/03/27) ¶ 0023, [0031, 0033,0034	1-6
A	CN 102891876 A ZTE CORP. 23 Jan 2013 (2013/01/23) Whole Document	1-6
A	US 5787169 A ELDRIDGE et al 28 Jul 1998 (1998/07/28) Whole document	1-6
A	US 2006288232 A1 HO et al. 21 Dec 2006 (2006/12/21) Whole Document	1-6
A	White Paper: SafeNet DataSecure vs. Native SQL Server Encryption. Retrieved from the internet on 06/10/2015 at< http://www.acapacific.com.sg/aca_promo/edm/security/2010/06/DataSecure_SQLServer_Native_Encryption.pdf> SAFENET 31 Dec 2009 (2009/12/31) Whole Document	1-6

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

☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
06 Oct 2015	08 Oct 2015
Name and mailing address of the ISA: Israel Patent Office Technology Park, Bldg.5, Malcha, Jerusalem, 9695101, Israel Facsimile No. 972-2-5651616	Authorized officer COPPENHAGEN Uri Telephone No. 972-2-5657811

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/IL2015/050564

Patent document cited search report	Publication date	Patent family member(s)	Publication Date
US 2008077806 A1	27 Mar 2008	US 2008077806 A1	27 Mar 2008
		US 7904732 B2	08 Mar 2011
		EP 2076865 A1	08 Jul 2009
		WO 2008037605 A1	03 Apr 2008
US 5787169 A	28 Jul 1998	US 5787169 A	28 Jul 1998
		AT 298436 T	15 Jul 2005
		CA 2241745 A1	10 Jul 1997
		CA 2241745 C	29 Apr 2003
		DE 69634880 D1	28 Jul 2005
		DE 69634880 T2	11 May 2006
		EP 0976049 A1	02 Feb 2000
		EP 0976049 B1	22 Jun 2005
		JP 2000502827 A	07 Mar 2000
		JP 3807747 B2	09 Aug 2006
		US 6178508 B1	23 Jan 2001
		WO 9724675 A1	10 Jul 1997
US 2006288232 A1	21 Dec 2006	US 2006288232 A1	21 Dec 2006
		US 7639819 B2	29 Dec 2009
CN 102891876 A	23 Jan 2013	CN 102891876 A	23 Jan 2013