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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
25 August 2016 (25.08.2016)

WIPO | PCT

(10) International Publication Number
WO 2016/131473 A1

(51) International Patent Classification:
G06F 17/30 (2006.01) *H04L 9/00* (2006.01)
G06F 21/30 (2013.01) *H04L 29/06* (2006.01)
G06F 21/60 (2013.01)

(21) International Application Number:
PCT/EP2015/053242

(22) International Filing Date:
16 February 2015 (16.02.2015)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicants: **NEC EUROPE LTD.** [DE/DE]; Kurfürsten-Anlage 36, 69115 Heidelberg (DE). **UNIVERSITÄT MANNHEIM** [DE/DE]; Schloss, 68131 Mannheim (DE).

(72) Inventors: **BOHLI, Jens-Matthias**; Römerstraße 30, 69181 Leimen (DE). **KARAME, Ghassan**; Ringstraße 3, Heidelberg Heidelberg (DE). **ARMKNECHT, Frederik**; Alzeyer Straße 53, 67549 Worms (DE).

(74) Agent: **ULLRICH & NAUMANN**; Schneidmühlstraße 21, 69115 Heidelberg (DE).

(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**: METHOD AND SYSTEM FOR VERIFYING INFORMATION OF A DATA ITEM IN A PLURALITY OF DIFFERENT DATA ITEMS

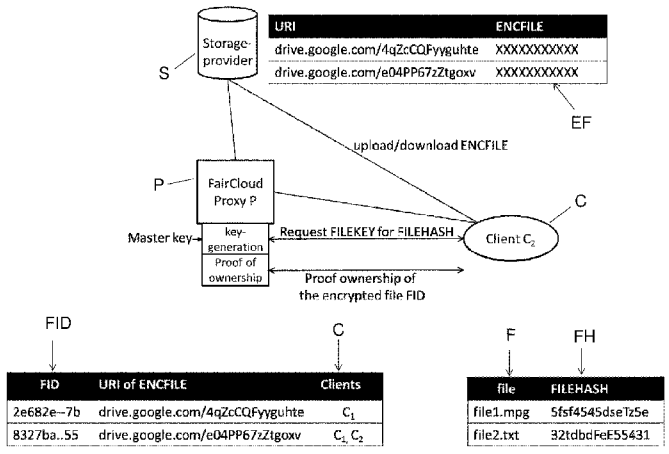

Fig. 1

(57) **Abstract**: The present invention relates to a method for verifying information of a data item (DI) in a plurality of different data items (DI), preferably stored on a server (SP) like a cloud or the like, wherein: a) a hash tree is generated from the plurality of data items (DI), such that the data items (DI) forming the leaves (LO) of the hash tree (HT) and such that the non-leaf nodes (L1, L2, L3,...) are computed by hashing the data items (DI) of their respective child nodes (L0, L1, L2,...) and when computing the root-hash (R) at least the distance between the root node (R) and the leaf-nodes (L0) is included into the hashing; b) an authentication path for said data item (DI) is computed based on a recomputation of the hash tree (HT), wherein an authentication path comprises all siblings of tree nodes from the data item (DI) to the root (R) of the hash tree (HT); c) the root-hash (R) is recomputed based on said data item (DI) and the computed authentication path of said data item (DI) and the recomputed root-hash (R) is compared with the root-hash (R) of the hash-tree (HT) of step a); d) the side element (RME) in the leaves (L0) or a tree level (L1) above of the hash tree (HT) and its authentication path is determined; e) the authentication path of said side element (RME) is verified, wherein based on the result of step e) the number of data items (DI) of said plurality is determined and wherein a membership of said data item (DI) to the plurality of data items is determined based on the result of step c).

WO 2016/131473 A1

METHOD AND SYSTEM FOR VERIFYING INFORMATION OF A DATA ITEM IN A PLURALITY OF DIFFERENT DATA ITEMS

The present invention relates to a method for verifying information of a data item in a plurality of different data items, preferably stored on a server like a cloud or the like.

The present invention further relates to a system for verifying information of a data item in a plurality of different data items, preferably stored on a server like a cloud or the like and preferably for performing with a method according to one of the claims 1-7.

Although applicable to any kind of storage in general, the present invention will be described with regard to cloud storage in particular shared or deduplicated files stored on the cloud.

Although applicable to any kind of information to be verified of a data item, the present invention will be described with regard storage costs of a data item.

Cloud storage is receiving increasing attention and importance recently. Cloud storage offers their users cost-effective, convenient and highly available storage services. Conventional clouds rely on cost-effective techniques such as data compression and data deduplication in order to save storage costs for the cloud. Data deduplication is beneficial as it significantly reduces the costs of storage. However, cloud users do not benefit from this deduplication, since deduplication is usually performed by the cloud itself. In other words users are usually being charged the same price, irrespective whether their data has been deduplicated by the cloud. This is a significant disadvantage for the users since users who are storing popular files which are usually deduplicated by the cloud storage should not be charged the same amount for storing non-deduplicated content.

In the non-patent literature of

- Pasquale Puzio, Refik Molva, Melek Önen and Sergio Loureiro. ClouDedup: Secure Deduplication with Encrypted Data for Cloud Storage, Proceedings of IEEE CloudCom 2013,
- A Secure Data Deduplication Scheme for Cloud Storage, Jan Stanek, Alessandro Sorniotti, Elli Androulaki, and Lukas Kenc, Proceedings of Financial Cryptography and Data Security, 2014,
- Boosting Efficiency and Security in Proof of Ownership for Deduplication, Roberto Di Pietro, Alessandro Sorniotti, Proceedings of ASIACCS 2012, and
- Mihir Bellare and Sriram Keelveedhi, Thomas Ristenpart, DupLESS: Server-Aided Encryption for Deduplicated Storage, Proceedings of Usenix Security 2013,

techniques are disclosed for performing deduplication over encrypted data or for a construction for a proof of ownership to attest that a user indeed possesses a file which is deduplicated by a cloud for example. These conventional techniques are directed to increase the profitability of clouds by allowing the cloud service provider to save on his storage costs.

However one of the disadvantages is, that these techniques are not transparent for the users of a cloud storage provider. A further disadvantage is, that such a cloud storage is costly for the users. An even further disadvantage is that users cannot verify if they are the only user of an uploaded file, or if other users also have uploaded the same file.

It is therefore an objective of the present invention to provide a method and a system for verifying information of a data item in a plurality of different data items enabling a fair allocation of storage costs among users according to the effective storage space that each user is occupying.

It is a further objective of the present invention to provide a method and a system for verifying information of a data item in a plurality of different data items enabling account savings for users achieved by a deduplication of their files.

It is an even further objective of the present invention to provide a method and a system for verifying information of data item in a plurality of different data items enabling the clients to prove a degree of deduplication of their files achieved in the cloud.

It is an even further objective of the present invention to provide a method and a system for verifying information of a data item in a plurality of different data items reducing the total size of data needed to be uploaded for storage by users.

It is an even further objective of the present invention to provide a method and a system for verifying information of da data item in a plurality of different data items which is flexible and easy to implement.

The aforementioned objectives are accomplished by a method of claim 1 and a system of claim 8.

In claim 1 a method for verifying information of data item in a plurality of different data items, preferably stored on a server like a cloud or the like is defined.

According to claim 1 the method is characterized in that

a) a hash tree is generated from the plurality of data items, such that the data items forming the leaves of the hash tree and such that the non-leaf nodes are computed by hashing the data items of their respective child nodes and when computing the root-hash at least the distance between the root node and the leaf-nodes is included into the hashing,

b) an authentication path for said data item is computed based on a recomputation of the hash tree, wherein an authentication path comprises all siblings of tree nodes from the data item to the root of the hash tree,

c) the root-hash is recomputed based on said data item and the computed authentication path of said data item and the recomputed root-hash is compared with the root-hash of the hash-tree of step a),

d) the side element in the leaves or a tree level above of the hash tree and its authentication path is determined,

e) the authentication path of said side element is verified,

wherein based on the result of step e) the number of data items of said plurality is determined and wherein a membership of said data item to the plurality of data items is determined based on the result of step c).

In claim 8 a system for verifying information of a data item in the plurality of different data items, preferably stored on a server like a cloud or the like and preferably for performing with a method according to one of the claims 1-7 is defined.

According to claim 8 the system is characterized by
a hashing entity adapted to generate a hash tree from the plurality of data items, such that the data items forming the leaves of the hash tree and such that the non-leaf nodes are computed by hashing the data items of their respective child nodes and when computing the root-hash at least the distance between the root node and the leaf-nodes is included into the hashing,
authentication entity adapted to compute an authentication path for said data item based on a recomputation of the hash tree, wherein an authentication path comprises all siblings of tree nodes from the data item to the root of the hash tree,
a recomputation entity adapted to recompute the root-hash based on said data item and the computed authentication path of said data item and comparing the recomputed root-hash with the root-hash of the hash-tree provided by said hashing entity,
a determining entity adapted to determine the side element in the leaves or a tree level above of the hash tree and its authentication path,
a verification entity adapted to verify the authentication path of said side element and
a result providing entity adapted to determine based on a provided result of the verification entity the number of data items of said plurality and a membership of said data item to the plurality of data items based on a provided result of said recomputation entity.

The term “side element” is to be understood as the element in the hash tree which is the right-most or the left-most element depending on the side from which the counting of data items is performed: If the position within the level of the leaves is counted from left to right, then the side element is the right-most element whereas when the position within the level is counted from right to left, then the side element is the left-most element when counted from the right.

According to the invention it has been recognized that the information of a data item can be verified with an efficient proof of membership.

According to the invention it has been further recognized that a cardinality of the data item in a plurality of different data items can be proved and provided.

According to the invention it has been further recognized that in particular the total size of data can be reduced needed to be uploaded for storing on a server for example.

According to the invention it has been even further recognized that in particular accounting information assigned to the data item can be verified.

According to the invention it has been even further recognized that storage costs of users can be reduced without compromising the performance.

According to the invention it has been even further recognized that storage costs savings for cloud users can be significantly reduced up to 30% when compared conventional commodity storage services for a number of realistic profiles of users.

According to the invention it has been even further recognized that flexibility is significantly enhanced since the present invention can be applied in various fields: For example in cloud storage in particular shared or deduplicated slides, reputation systems where a score is computed based on a set of users who provided a rating, subscribers to services, for example publish or subscribe

systems, mailing lists, kickstarter projects or the like, inventory of digital items and/or electronic voting.

According to the invention it has been even further recognized that an easy implementation is provided.

According to the invention it has been further recognized that the IDs of the users storing the same file within any time epoch, preferably maintained by a proxy entity, a gateway P or the like can be efficiently accumulated: Each user can check that his ID is correctly accumulated at billing time.

According to the invention it has been even further recognized that the number of accumulated values can be encoded as well such that (i) any client can verify this number while (ii) it is not visible to outsiders.

Further features, advantages and preferred embodiments are described in the following sub claims.

According to a preferred embodiment when the number of leaves of the hash-tree formed by the data items compared with a required number to form a full hash-tree is lower, then further data items are used as leaves for compensation and are filed with a distinct data item. This allows in an easy way to complete any kind of hash-tree for example binary trees or the like. Thus flexibility is enhanced.

According to a further preferred embodiment the data items forming the leaves of the hash-tree are itself hash values of plaintext information. This enables to hide the information of the data items from the computing entity computing the hash-tree.

According to a further preferred embodiment the hash-tree is a binary tree, preferable a Merkle tree. The binary tree enables a fast and efficient hashing since every node comprises only maximum of two child nodes.

According to a further preferred embodiment the hash-tree is a tiger hash-tree with the hashing based on the tiger hash function. A tiger hash-tree is based on the crypto hash-function tiger and is preferably used to check the integrity of large data files during or after a transmission. The tiger tree hash hashes on the leave level preferably data blocks each having 1024 bytes.

According to a further preferred embodiment said plurality of data items is associated to a file with a file identification wherein said file identification is announced together with a root of the hash-tree. This allows in an easy way to verify the information of the data item for every client which wants to store the file with the corresponding file FID.

According to a further preferred embodiment in case of different files each having a different file identification and associated to different pluralities of data items one or more of the files are randomly selected and steps a)-e) are performed for each data item in said corresponding plurality. This enables to keep the amount of data generated and transmitted small for verification since only a partial checking of selected files randomly is performed. For each of the selected file a proof of membership and the cardinality for each client is computed.

There are several ways how to design and further develop the teaching of the present invention in an advantageous way. To this end it is to be referred to the patent claims subordinate to patent claim 1 on the one hand and to the following explanation of preferred embodiments of the invention by way of example, illustrated by the figure on the other hand. In connection with the explanation of the preferred embodiments of the invention by the aid of the figure, generally preferred embodiments and further developments of the teaching will be explained.

In the drawings

Fig. 1 shows a system for verifying information according to a first embodiment of the present invention;

Fig. 2 a hash-tree according to a second embodiment of the present invention and

Fig. 3 a schematic view of a use case of a third embodiment of the present invention.

Fig. 1 shows a system for verifying information according to a first embodiment of the present invention.

In Fig. 1 a proxy P is shown which receives requests from clients C when the client wants to upload a file to a storage provider SP.

The client C and the proxy entity or gateway P starts executing a proxy-aided key generation protocol for an encryption key. The client C then encrypts the file using an encryption algorithm with the encryption key. A root of a hash-tree is computed over the encrypted file and used as a file ID – FID -. The file ID is then associated with the file to be stored and when a plurality of users is associated to a file with a corresponding file ID FID then for example the clients may benefit from lower costs caused by a deduplication of files. Usually at the end of a predefined epoch the proxy P bills the clients C for the files held by them at any time during the epoch.

If a client C deletes a file during the epoch the proxy P still bills the user for this file and epoch and removes the marked client C from the list after computation of a corresponding bill. To provide a fair billing for example each client C is billed with a cost incurred by storing in the file F at the storage provider SP as a margin charged by the proxy P. Once every client C obtains the bill the proxy P starts the proving process to convince clients C of a file with FID that all clients C are billed in the same way. The clients C of a file F with file identification FID are convinced of their billing by creating a set of all clients C of said file and assuring that the size corresponds to the set of clients C of the file F, that all clients C are billed for a file being referenced to the same set of clients C and that a client C addressed is indeed part of that set of clients C. This information can be verified using an embodiment of the present invention: First step a) is performed with the data item being one of the clients, e.g. his Client ID and the plurality of different items being

the set of clients C of a file. When the top or root-hash, preferably computed by the proxy P, is published among all clients C then each client C may check the number of clients C billed for a file F and the size of the set of clients C for a file F. Therefore the verified information, i.e. whether the client C belongs to the plurality of clients C associated to one file F and the number of clients C within said plurality can be verified using the steps a)-e) according to an embodiment of the present invention.

Fig. 2 shows a hash-tree according to a second embodiment of the present invention.

In Fig. 2 a hash-tree according to an embodiment of the present invention is shown. This hash-tree is based on a Merkle tree HT with a cryptographic hash-function H. The Merkle tree HT is a binary tree HT where the data DI is put into the leaves of the tree. The non-leave nodes are computed by computing the hash of their respective to child nodes. A node of the tree is denoted by $a_{i,j}$ where i denotes the level L1, L2, L3 of the node and j the position of the node $a_{i,j}$ in said level L1, L2, L3. The levels L1, L2, L3 are counted as the distance to the level of the leaf nodes, with the leaf nodes thus being $a_{0,j}$. The position within a level L1, L2, L3 is counted from left to right, the leftmost node of a level thus being $a_{i,0}$. A standard Merkle tree HT is formed by $a_{i+1,j} = H(a_{i,2j};a_{i;2j+1})$.

In order to be able to verify the cardinality of the set, the computation of the Merkle tree HT is modified by including the level in the hash operation. Still, the data is put in the leaf nodes. The remaining empty nodes are filled with a distinct symbol 0.

Unlike the standard Merkle tree HT, the inner nodes are computed as

$$a_{i+1,j} = H(i+1, a_{i,2j}, a_{i,2j+1}).$$

The digest d that is output by a procedure Acc accumulating the set is given by the root of the tree $a_{3,0}$. This is visualized in Figure 2.

A proving procedure ProveS(S;x) outputs a proof that the element/data item x is contained in the set S. To do so, the procedure recomputes the Merkle Tree HT as described above. A proof that x is contained as a leave node is given as the authentication path for x, as in standard Merkle Trees HT. An authentication path consists of all siblings of nodes that are on the path from x to the root. The output px of the proving procedure is the element x and the authentication path for said element x.

The verification of the proof is performed with a verification procedure VerifyS(d;x; px) which recomputes the root element using x and the elements given in the authentication path and compares the obtained root element with the original digest d, i.e. the output of the procedure Acc.

To prove the cardinality of the set a procedure ProveC(S) is used. The proof pC comprises the rightmost element RNE $a_{0;|S|-1}$ and the authentication paths of that element as well as the path of the first 0 element. Given this information, the verifier can use a verification procedure VerifyC(d;c; p|S|) to confirm the cardinality c of the set S. The procedure checks that $a_{0,c-1}$ is indeed an nonempty element that is part of the set by verifying the authentication path. It then checks the right part of the hash tree HT which is possible, given the authentication path of the first 0 element, because the empty leaves are known to the verifier to be 0. This is efficiently possible with pre-computed inner nodes.

Fig. 3 shows a schematic view of a use case of a third embodiment of the present invention.

In Fig. 3 a sketch of a billing process between the proxy P and a client is shown: The clients C first obtain bills stating the file with file identification FID associated to them and the cost charged for that file. After the bills are sent out to the clients C the proxy P may preferably obtains a random string for example using a GetRandomness procedure, that was provably unknown at the time of the creation of the bill. A random string of length n representing the number of clients can be used as a mask on the file identification FID to select with a probability the number of files whose file identification FID starts with this string. The proxy P will then for

each selected file compute the proof-of-membership and the cardinality as information to be verified for each owner using steps a)-f) according to an embodiment of the present invention. To proxy P takes the set SF of the clients C associated with the file FID.

Therefore for each FID the set of users is created and the proxy P provides a proof-of-membership for the FID and the proof of the size of the user set for the corresponding FID enabling a client C to check whether he was fairly billed with deduplication of the file.

To summarize the present invention provides preferably a counting accumulator based on a cryptographic accumulator and added with the possibility to give a proof for the number of elements that are accumulated. This counting accumulator is tuple of procedures (Acc; ProveS; ProveC; VerifyS; VerifyC) where Acc(S) = d accumulates the set when this digest d is then committed to a set and publicly commit to set digest for example on a public bulletin board such that each client can verify billing.

The present invention has several applications in particular where accountability of statements about the set or group of users or elements has to be made. Used cases are

- Cloud storage, in particular shared or deduplicated slides as described above
- Reputation systems, where a score is computed based on a set of users who provided a rating
- Subscribers to services (e.g. publish/subscribe systems, mailing lists, Kickstarter projects)
- Inventory of digital items
- Electronic Voting.

In summary the present invention enables to rely on a novel tree-based counting accumulator providing efficient proofs of membership and cardinality. The present invention further enables a combination of the use of the counting accumulator with probabilistic methods based on external randomness in order to reduce the

total size of data needed to be uploaded on the public bulletin board to enable bill verification.

The present preferably provides a method comprising the steps of generating a counting accumulator providing a proof-of-membership and cardinality, publishing on a public bulletin board an association between a chosen file with file identification FID and a digest of the accumulator, preferably the root of a hash-tree of the accumulator and sending proof-of-membership and cardinality information for each client subscribed to the chosen file identification FID. To reduce the work load after generation of the counting accumulator and prior to publish the association committing to a given bill sent to each user/client may be performed and/or using external randomness to sample which file identification FID to prove.

The present invention further enables a probabilistic procedure which selectively reveals details about a number of file accumulators in each epoch. If the gateway/proxy could make the selection on his own, he could easily cheat by creating only correct bills for the selected files while overcharging the clients registered to the remaining files. There, the selection procedure is seeded by a trusted external source of pseudorandomness which is preferably based on a virtual currency like Bitcoin. It ensures that any client can check that the selection has been done correctly. Moreover, as this source is unpredictable, using it does not give any advantage for the gateway/proxy entity P to misbehave.

The present invention has inter alia the following advantages: The present invention provides protection against attempts to overcharge users. The present invention is further more flexible and transparent from the perspective of the users and the storage provider, provides cheaper storage costs than conventional storage services and when combined with encryption without compromising the confidentiality of data. Further the performance is not significantly reduced: The overhead incurred on the proxy in orchestrating for example data deduplication is minimal and a tolerable overhead on the users is enabled when verifying their bills at the end of every epoch.

Many modifications and other embodiments of the invention set forth herein will come to mind to the one skilled in the art to which the invention pertains having the benefit of the teachings presented in the foregoing description and the associated drawings. Therefore, it is to be understood that the invention is not to be limited to the specific embodiments disclosed and that modifications and other embodiments are intended to be included within the scope of the appended claims. Although specific terms are employed herein, they are used in a generic and descriptive sense only and not for purposes of limitation.

Claims

1. A method for verifying information of a data item (DI) in a plurality of different data items (DI), preferably stored on a server (SP) like a cloud or the like, characterized in that

   a) a hash tree is generated from the plurality of data items (DI), such that the data items (DI) forming the leaves (LO) of the hash tree (HT) and such that the non-leaf nodes (L1, L2, L3, ...) are computed by hashing the data items (DI) of their respective child nodes (L0, L1, L2, ...) and when computing the root-hash (R) at least the distance between the root node (L3) and the leaf-nodes (L0) is included into the hashing,

   b) an authentication path for said data item (DI) is computed based on a recomputation of the hash tree (HT), wherein an authentication path comprises all siblings of tree nodes from the data item (DI) to the root (R) of the hash tree (HT),

   c) the root-hash (R) is recomputed based on said data item (DI) and the computed authentication path of said data item (DI) and the recomputed root-hash (R) is compared with the root-hash (R) of the hash-tree (HT) of step a),

   d) the side element (RME) in the leaves (L0) or a tree level (L1) above of the hash tree (HT) and its authentication path is determined,

   e) the authentication path of said side element (RME) is verified,

   wherein based on the result of step e) the number of data items (DI) of said plurality is determined and wherein a membership of said data item (DI) to the plurality of data items is determined based on the result of step c).

2. The method according to claim 1, characterized in that when the number of leaves (L0) of the hash tree (HT) formed by the data items (DI) compared with the required number to form a full hash tree (HT) is lower then further data items are used as leaves (L0) for compensation and are filed with a distinct data item.

3. The method according to one of the claims 1-2, characterized in that the data items (DI) forming the leaves (L0) of the hash tree (HT) are itself hash values of plaintext information.

4. The method according to one of the claims 1-3, characterized in that the hash-tree (HT) is a binary tree, preferably a Merkle tree.

5. The method according to one of the claims 1-4, characterized in that the hash-tree (HT) is a Tiger hash-tree with the hashing based on the tiger hash function.

6. The method according to one of the claims 1-5, characterized in that said plurality of data items (DI) is associated to a file with a file identification (FID), wherein said file identification (FID) is announced together with the root (R) of the hash tree (HT).

7. The method according to claim 6, characterized in that in case of different files each having a different file identification (FID) and associated to different pluralities of data items (DI) one or more of the files are randomly selected and steps a)-e) are performed for each data item (DI) in said corresponding plurality.

8. A system, preferably a proxy entity, connectable to a storage entity and one or more clients, for verifying information of a data item (DI) in a plurality of different data items (DI), preferably stored on a server (SP) like a cloud or the like and preferably for performing with a method according to one of the claims 1-7,
characterized by
a hashing entity adapted to generate a hash tree (HT) from the plurality of data items (DI), such that the data items (DI) forming the leaves (L0) of the hash tree (HT) and such that the non-leaf nodes are computed by hashing the data items (DI) of their respective child nodes (L0, L1, L2, ...) and when computing the root-hash (R) at least the distance between the root node (L3) and the leaf-nodes (L0) is included into the hashing.

authentication entity adapted to compute an authentication path for said data item (DI) based on a recomputation of the hash tree (HT), wherein an authentication path comprises all siblings of tree nodes from the data item (DI) to the root (R) of the hash tree (HT),

a recomputation entity adapted to recompute the root-hash (R) based on said data item (DI) and the computed authentication path of said data item (DI) and comparing the recomputed root-hash (R) with the root-hash (R) of the hash-tree (HT) provided by said hashing entity,

a determining entity adapted to determine the side element (RME) in the leaves (LO) or a tree level above of the hash tree (HT) and its authentication path,

a verification entity adapted to verify the authentication path of said side element and

a result providing entity adapted to determine based on a provided result of the verification entity the number of data items (DI) of said plurality and a membership of said data item (DI) to the plurality of data items (DI) based on a provided result of said recomputation entity.

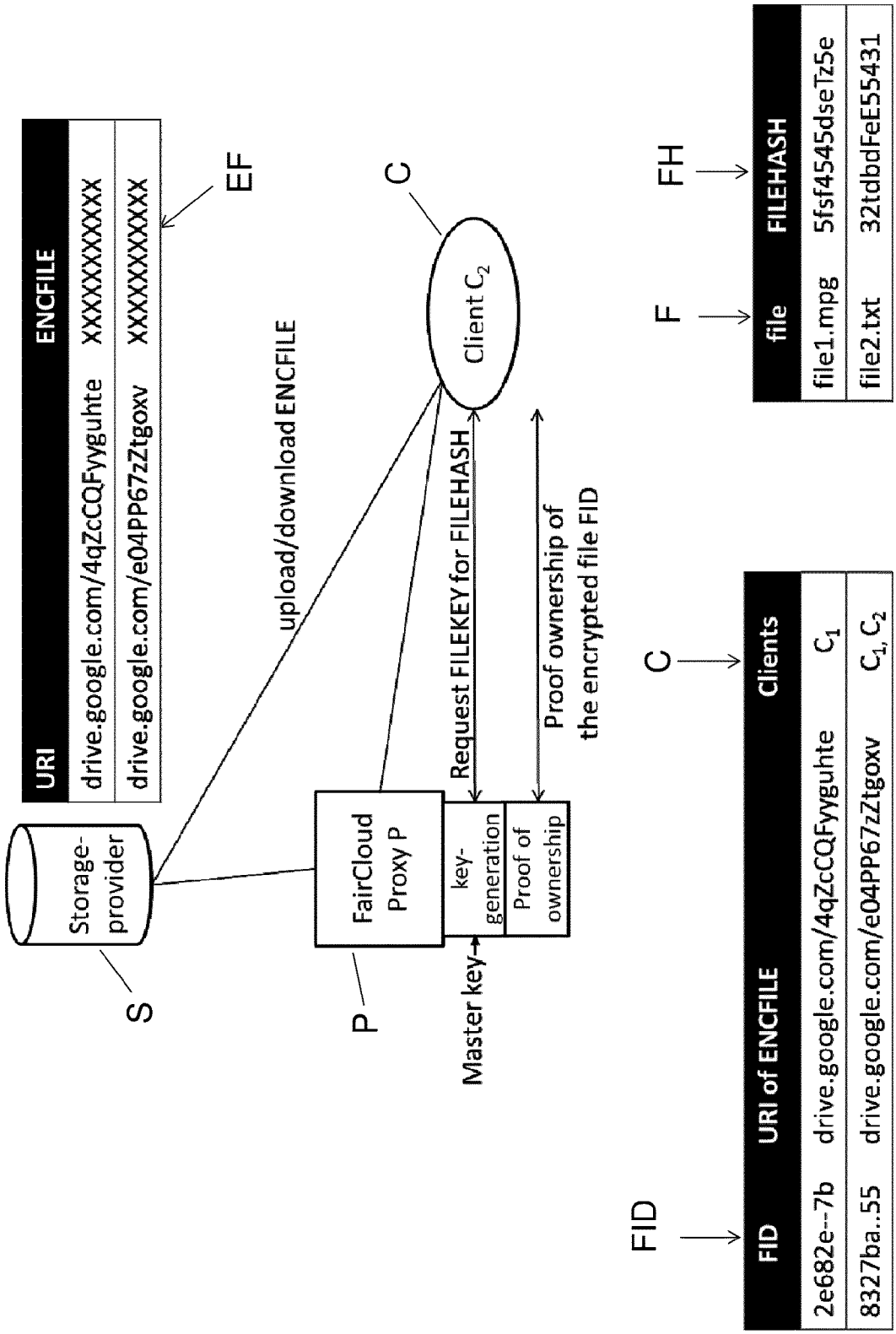
URI
ENCFILE
drive.google.com/4qZcCQFyyguhte XXXXXXXXXX
drive.google.com/e04PP67zZtgoxv XXXXXXXXXX
EF
Storage-provider
S
upload/download ENCFILE
FairCloud Proxy P
P
key-generation
Proof of ownership
Master key
Request FILEKEY for FILEHASH
Proof ownership of the encrypted file FID
C
Client C_2
FH
F
file
FILEHASH
file1.mpg 5fsf4545dseTz5e
file2.txt 32tdbdFeE55431
C
FID
FID
URI of ENCFILE
Clients
2e682e--7b drive.google.com/4qZcCQFyyguhte C_1
8327ba..55 drive.google.com/e04PP67zZtgoxv C_1, C_2

Fig. 1

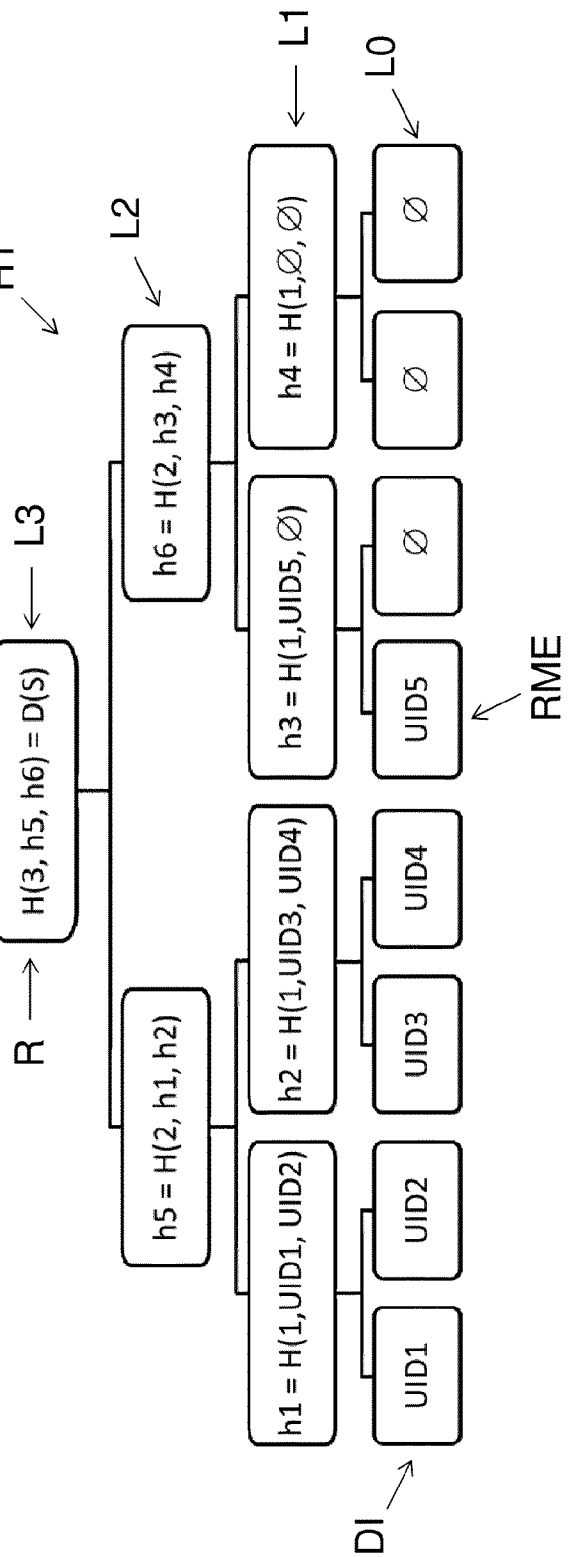
HT
L3
L2
L1
L0
R
H(3, h5, h6) = D(S)
h6 = H(2, h3, h4)
h5 = H(2, h1, h2)
h4 = H(1,∅,∅)
h3 = H(1,UID5,∅)
h2 = H(1,UID3, UID4)
h1 = H(1,UID1, UID2)
∅
∅
∅
UID5
UID4
UID3
UID2
UID1
RME
DI

Fig. 2


Client
C
Fair amount of billing with deduplication
Proof of membership for FID
Proof size of user set for FID
For each FID, create set of users
FairCloud Proxy P
P

Fig. 3

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2015/053242

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F17/30 G06F21/30 G06F21/60 H04L9/00 H04L29/06
ADD.

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)
G06F H04L

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)
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	PREMKUMAR DEVANBU ET AL: "Authentic Data Publication over the Internet", JOURNAL OF COMPUTER SECURITY, vol. 11, no. 3, 1 April 2003 (2003-04-01), XP055219881, abstract page 2, last paragraph - page 3, paragraph 7 page 4, paragraph 1 - paragraph 4 page 5, last paragraph - page 7, paragraph 5 page 8, paragraph 5 - page 9, paragraph 1 ----- -/--	1-8

[X] 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
14 October 2015	26/10/2015

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Herri, Edmond

1

Form PCT/ISA/210 (second sheet) (April 2005)

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	CHARLES MARTEL ET AL: "A General Model for Authenticated Data Structures", ALGORITHMICA., vol. 39, no. 1, 27 January 2004 (2004-01-27), pages 21-41, XP055220357, US ISSN: 0178-4617, DOI: 10.1007/s00453-003-1076-8 page 26, paragraph 1 - paragraph 3	1-8
A	Roberto Tamassia: "Authenticated Data Structures" In: "Advances in Communication Networking : 20th EUNICE/IFIP EG 6.2, 6.6 International Workshop, Rennes, France, September 1-5, 2014, Revised Selected Papers", 16 September 2003 (2003-09-16), SPRINGER VERLAG, DE 032548, XP055220055, ISSN: 0302-9743 ISBN: 978-3-642-36699-4 vol. 2832, pages 2-5, DOI: 10.1007/978-3-540-39658-1_2, page 3 - page 4	1-8
A	Michael Szydlo: "Recent Improvements in the Efficient Use of Merkle Trees: Additional Options for the Long Term", , 10 May 2004 (2004-05-10), XP055218992, Retrieved from the Internet: URL:http://www.emc.com/emc-plus/rsa-labs/historical/recent-improvements-efficient-use-merkle-trees.htm [retrieved on 2015-10-07] page 1, paragraph 5 - paragraph 7	1-8
A	Anonymous: "Tiger (cryptography) - Wikipedia, the free encyclopedia", , 15 December 2014 (2014-12-15), XP055220933, Retrieved from the Internet: URL:https://en.wikipedia.org/w/index.php?title=Tiger_(cryptography)&oldid=638123361 [retrieved on 2015-10-14] page 1	5

1