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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
6 March 2008 (06.03.2008)

PCT

(10) International Publication Number
WO 2008/026184 A2

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

(21) International Application Number:
PCT/IB2007/053498

(22) International Filing Date: 30 August 2007 (30.08.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
06119878.4 31 August 2006 (31.08.2006) EP

(71) Applicant *(for all designated States except US)*: KONINKLIJKE PHILIPS ELECTRONICS N.V. [NL/NL]; Groenewoudseweg 1, NL-5621 BA Eindhoven (NL).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **ZYCH, Anna, K.** [PL/NL]; c/o High Tech Campus Building 44, NL-5656 AE Eindhoven (NL). **DOUMEN, Jeroen, M.** [NL/NL]; c/o High Tech Campus Building 44, NL-5656 AE Eindhoven (NL). **JONKER, Willem** [NL/NL]; c/o High Tech Campus Building 44, NL-5656 AE Eindhoven (NL). **HARTEL, Pieter, H.** [NL/NL]; University of Twente, PO Box 217, NL-7500 AE Enschede (NL). **PETKOVIC, Milan** [RS/NL]; c/o High Tech Campus Building 44, NL-5656 AE Eindhoven (NL).

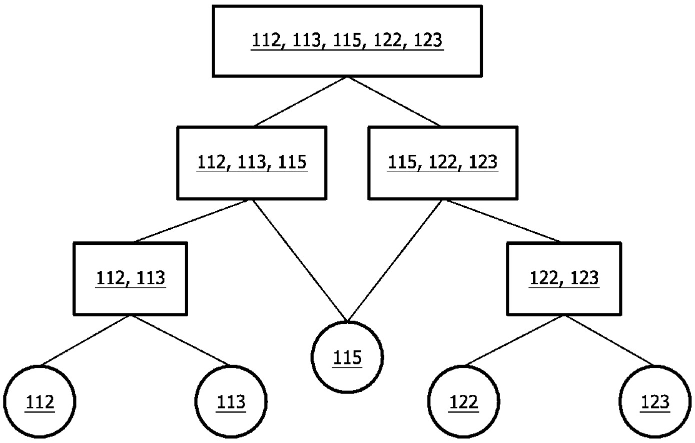
(74) Agents: **GROENENDAAL, Antonius, W., M.** et al.; High Tech Campus Building 44, NL-5656 AE Eindhoven (NL).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, 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, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, 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, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

(54) Title: METHOD OF KEY MANAGEMENT



(57) Abstract: A method of key management for group-based controlled access to encrypted data, in which a decryption key for the encrypted data can be obtained by a party if the party is a member of at least one group which is authorized to access the data, the groups being organized in a hierarchical tree in which each non-leaf node represents a group and each leaf node represents a member of all groups represented by nodes hierarchically superior to the leaf node in question, characterized in that the leaf nodes are each assigned a respective arbitrarily chosen private key and corresponding public key, in that the private key associated with a particular non-leaf node is obtained by executing a key agreement protocol using a private key associated with a first child of the particular non- leaf node and a public key associated with a second child of the particular non-leaf node, and in that the private key for a group associated with a particular node is obtained by recursively obtaining the group private keys of the nodes on a path from the leaf node corresponding to the party in question and the node corresponding to the authorized group.

METHOD OF KEY MANAGEMENT

FIELD OF THE INVENTION

The invention relates to a method of key management for group-based controlled access to encrypted data, in which a decryption key for the encrypted data can be obtained by a party if the party is a member of at least one group which is authorized to access the data, the groups being organized in a hierarchical tree in which each non-leaf node represents a group and each leaf node represents a member of all groups represented by nodes hierarchically superior to the leaf node in question.

BACKGROUND OF THE INVENTION

In many data security situations, it is desirable to enforce hierarchical access control to certain data. The persons, devices or other parties for whom access control is to be regulated are made members of groups. The groups are hierarchically organized. The topmost or root node represents all parties, and the lowermost or leaf nodes represent the individual parties. The nodes between root node and leaf nodes represent subsets of the parties.

To ensure that only authorized parties can access the data, the data is encrypted using a key associated with a node corresponding to an authorized group of parties. This means that only the parties that are member of this group can obtain the decryption key for this data.

Several methods have been proposed that are based on the hierarchical organization of the parties. Two examples are Akl and Taylor, “Cryptographic solution to a problem of access control in a hierarchy”, *ACM Transactions on Computer Systems* 1(3):239-248, 1983 and Harn and Lin, “A cryptographic key generation scheme for multilevel data security”, *Computer Security,* 9(6):539-546, 1990. These schemes are based on RSA public key cryptography to construct node private keys from the root node’s private key and certain carefully chosen parameters. They require each node to only store one private key, since the others can be derived from that one key. However for *n* parties the amount of public information is on the order of $n^2 log(n)$, which grows unacceptably large for non-trivially large groups.

Another approach is found in Kuo et al., “Cryptographic key assignment scheme for dynamic access control in a user hierarchy”, *Computers and Digital Techniques*, IEE Proceedings, vol.146, no.5, pp.235-240, Sep 1999. This approach is based on the Chinese Remainder Theorem. Following this approach requires storing more than *n* private keys for *n* parties, which is disadvantageous especially when dealing with devices with small (secure) storage capabilities.

A combination of the above is proposed in Ray et al., “A cryptographic solution to implement access control in a hierarchy and more”, *SACMAT ’02: Proceedings of the seventh ACM symposium on access control models and technologies*, pages 65-73, ACM Press, 2002. This proposal also suffers from the disadvantage of having to store more than *n* private keys for *n* parties.

Many other approaches have been proposed. See e.g. Lin, "Dynamic key management schemes for access control in a hierarchy" Computer Communications, 20(15):1381–1385(5), 1997; Lin, “Hierarchical key assignment without public key cryptography”, *Computers & Security,* 20(7):612-619, 2001; Tzeng, "A time-bound cryptographic key assignment scheme for access control in a hierarchy", IEEE Transactions on Knowledge and Data Engineering, 2001; Chang et al., “Cryptographic key assignment scheme for access control in a hierarchy”, *Information Systems,* 17(3):243-247, 1992; or Shen and Chen, “A novel key management scheme based on discrete logarithms and polynomial interpolations”, *Computers & Security,* 21(2):164-171, 2002. In most approaches, each party needs to store one piece of private information. The amount of public information is on the order of n^2 at least. Moreover many of these approaches suffer from security flaws.

SUMMARY OF THE INVENTION

It is an object of the invention to provide a method according to the opening paragraph, in which each party stores at most one private key and the computational overhead needed to access the data by a user does not exceed O(log n).

This object is achieved according to the invention in a method which is characterized in that the leaf nodes are each assigned a respective arbitrarily chosen private key and corresponding public key, and in that the private key for a group associated with a particular node is obtained by executing a key agreement protocol using a private key associated with a first child of the particular node and a public key associated with a second child of the particular node. Preferably the key agreement protocol comprises the Diffie-Hellman key agreement protocol, although other key agreement protocols are also possible.

In this method, each party holds one private key that was assigned to him. To get group-based access to encrypted data, a party needs to reconstruct the group key for an authorized group. This is done by executing a key agreement protocol between two child nodes of the node corresponding of the authorized group. A key agreement protocol such as Diffie-Hellman allows two parties to establish a common key. Each party computes the common key (group key) by combining the private key held by one child node of the group node in question with the public key of the other child node of the group node in question.

A child node can either be a leaf node, i.e. the node corresponding to the party in question, or a non-leaf node. If the child node is a leaf node, the private key is simply the private key assigned to the party in question.

The private key associated with a particular non-leaf node is obtained by executing the key agreement protocol using a private key associated with a first child of the particular non-leaf node and a public key associated with a second child of the particular non-leaf node. This way for each node the group private key can be obtaining by recursively obtaining the group private keys of the nodes on a path from the leaf node corresponding to the party in question and the node corresponding to the authorized group.

The invention has two main embodiments: dynamically managed hierarchies and statically determined hierarchies. In the dynamic case, where access policies are given in advance, the amount of public information that needs to be available is less than O(n^2) when n parties are involved. In the static case, where access policies are not known in a setup phase (referred in the literature as the stateless receivers case) the amount of public information is O(n^2), and the data encryption key may need to be encrypted with the number of keys at most equal to the number of users not authorized to access that data.

For each node a single public key needs to be available. Each party further only needs to hold one item of private information, namely its private key. Consequently, in the dynamic case, the required amount of public information is less than n^2 for n parties and it is expected to drop down to linear size O(n). This is a significant improvement over the above-mentioned methods in which the amount of public information is more than n^2 for n parties.

In an embodiment the method further comprises constructing the hierarchical tree from a given a two-dimensional access control table that represents for each of a number of parties and each of a number of data items whether the particular party is allowed to access the particular data item. In many situations the access control information is initially available only as a table. By transforming this table into the hierarchical tree the method of the invention can be used for such access control situations. This embodiment satisfies properties required for applying the Diffie-Helman-based key generation scheme described below,

In an alternative embodiment it is assumed the access control information is not available in advance. Then the method comprises constructing the hierarchy as a partially ordered set of intervals of users. The intervals are constracted from a number of consecutive parties. Each interval can be uniquely split into two intervals of the size which is half of the size of the original interval. The original interval is put as direct predecessor of the two intervals into which it is split. This property is used to order the intervals.

In an embodiment the method further comprises encrypting the data with an arbitrarily chosen data key and creating an encrypted data key by encrypting the data key with a public key for a group which is authorized to access the data. This is much more efficient than encrypting the data itself with the public key, as public key encryption of large amounts of data is very time-consuming.

In a further embodiment the method further comprises encrypting the data with an arbitrarily chosen data key and creating an encrypted data key by encrypting the data key with a key encryption key that has been derived from the private key for a group which is authorized to access the data. Preferably the key encryption key is obtained by applying a cryptographic hash function to the private key in question. This embodiment obviates the need for public key encryption.

In a further embodiment the method comprises creating multiple encrypted data keys each of which is created by encrypting the data key using a respective public key for a respective group which is authorized to access the data, thereby allowing members of each of the respective groups to obtain the data key. This advantageously achieves that the data only needs to be encrypted once even when multiple groups need to be authorized to access the data. Encrypting the data key multiple times is much more efficient than encrypting the data itself multiple times.

The invention further provides a device for group-based controlled access to encrypted data. The device is configured for obtaining the private key for a group associated with a particular node by recursively obtaining the group private keys of the nodes on a path from the leaf node corresponding to the party in question and the node corresponding to the authorized group.

BRIEF DESCRIPTION O F THE DRAWINGS

These and other aspects of the invention will be apparent from and elucidated with reference to the illustrative embodiments shown in the drawings, in which:

Fig. 1 schematically illustrates an arrangement for group-based controlled access to encrypted data;

Fig. 2 schematically shows a hierarchical tree comprising nodes for the dynamic case;

Figs. 3(a) and (b) illustrate a way to construct the hierarchical tree of Fig. 2;

Fig. 4 schematically shows a hierarchical tree comprising nodes for the static case.

Throughout the figures, same reference numerals indicate similar or corresponding features. Some of the features indicated in the drawings are typically implemented in software, and as such represent software entities, such as software modules or objects.

DETAILED DESCRIPTION

Fig. 1 schematically illustrates an arrangement for group-based controlled access to encrypted data, in which the invention is used to manage the keys for the access control. Data 210, which preferably comprises content such as music, songs, movies, animations, speeches, video clips for music, TV programs, pictures, games, ringtones, spoken books and the like, but which could also be electronic health records, e-mail, computer files and so on, is processed at a server 200 before delivery to devices 112, 113, 115, 122, 123.

The data 210 is first encrypted by data encryption module 202 before it is delivered to the devices 112, 113, 115, 122, 123 by delivery module 203. Delivery may involve transmission over radio, television or telephone networks, or computer networks such as the Internet. Another means of delivery is the recording of the encrypted data 210 on a carrier such as a DVD disc which is subsequently made available to the devices 112, 113, 115, 122, 123.

To encrypt the data 210, the data encryption module 202 receives one or more encryption keys from authorization module 201. The authorization module 201 is provided with information, such as an access control table, to determine which of the devices 112, 113, 115, 122, 123 should be able to decrypt the encrypted data 210.

In accordance with the invention, each node has a corresponding public key and private key, preferably suitable for the Diffie-Hellman key agreement protocol. How these keys are obtained is discussed below.

In a first embodiment the hierarchy is managed in a dynamic manner. Fig. 2 schematically shows a hierarchical tree comprising nodes 100, 110-113, 115, 120-123. In the tree, the node 100 is the root node, nodes 112, 113, 115, 122, 123 are leaf nodes, and nodes 100, 110, 111, 120, 121 are non-leaf nodes. Note that each node has two child nodes, although as node 115 shows a node can be connected to multiple parent nodes. Note that two nodes must not have the same set of children.

The nodes 112, 113, 115, 122, 123 correspond to the devices 112, 113, 115, 122, 123 in Fig. 2, hence the same numerals are used. The devices 112, 113, 115, 122, 123 are divided into authorized groups, which groups correspond to the non-leaf nodes in the hierarchical tree of Fig. 2. That is, devices 112 and 113 are comprised in group 111, and together with device 115 they are also comprised in group 110. Similarly, devices 122 and 123 are comprised in group 121, and together with device 115 they are also comprised in group 120. All devices 112, 113, 115, 122, 123 finally can be referred to as group 100.

To enable devices 112, 113 to decrypt the encrypted data 210, the authorization module 201 provides a group encryption key for group 111 to the data encryption module 202. If also device 115 should be able to decrypt the encrypted data 210, the authorization module 201 would instead supply the group encryption key for group 110. The data encryption module 202 then encrypts the data 210, preferably by encrypting the data 210 with an arbitrarily chosen data key and creating an encrypted data key by encrypting the data key with a public key for a group which is authorized to access the data.

To enable devices 112, 113, 122, 123 to decrypt the encrypted data 210, the authorization module 201 provides the group encryption key for group 111 as well as the group encryption key for group 121. The data encryption module 202 then encrypts the data twice, once for each key. Preferably this is done by encrypting the data 210 with an arbitrarily chosen data key and creating multiple encrypted data keys each of which is created by encrypting the data key using a respective public key for a respective group which is authorized to access the data, thereby allowing members of each of the respective groups to obtain the data key.

As stated above, the authorization module 201 is provided with information to determine which of the devices 112, 113, 115, 122, 123 should be able to decrypt the encrypted data 210. This information will often be available in the form of a two-dimensional access control table that represents for each of a number of parties and each of a number of data items whether the particular party is allowed to access the particular data item. An example of such a table is shown below:

	C_1	C_2	C_3	C_4	C_5	C_6	C_7
Device 112	+	+	-	-	-	+	+
Device 113	+	+	-	-	-	+	+
Device 115	+	+	+	+	+	+	+
Device 122	-	-	+	+	+	+	+
Device 123	-	-	+	+	+	+	+

This access control table indicates that the devices 112, 113, 115, 122, 123 may access a data item C_1, C_2, C_3, C_4, C_5, C_6, C_7 if the corresponding cell indicates a '+'. Several methods exist to construct the hierarchical tree of Fig. 1 from this table. One particularly advantageous method will now be discussed.

First, leaf nodes are created for the individual devices. Next, for each unique column a node is created, under which the leaf nodes and/or other nodes that are contained by this node are positioned that correspond to the devices or groups of devices that are authorized according to that column. In the above example, the columns for C_1, C_3 and C_7 are the unique columns. The result of these steps is shown in Fig. 3(a).

Subsequently the non-leaf nodes are decomposed by finding two non-leaf nodes that have maximal intersections, and adding a non-leaf node representing that intersection. For any non-leaf nodes having more than two child nodes, intermediary nodes are created to ensure all non-leaf nodes have two child nodes. The result of these steps is shown in Fig. 3(b). It is evident that the tree obtained in Fig. 3(b) is equivalent to the tree shown in Fig. 1.

In a second embodiment the hierarchy is managed in a static manner. Now the tree is constructed in a different manner. There is no access table, so the groups of users are static and determine at the beginning of systems lifetime. The hierarchy for this embodiment is illustrated in Fig. 4. Again the devices 112,113,115,122 and 123 are the leafs of the hierarchy. But the non-leaf nodes in this embodiment are intervals on the sets of consecutive receivers. Thus there are the following non-leaf nodes:

- Intervals of size 2: {112,113}, {113,115}, {115,122}, {122,123}, {123,112}

- Intervals of size 3: {112,113,115}, {113,115,122}, {115,122,123}, {122,123,112}, {123,112,113}
- Intervals of size 4: {112,113,115,122}, {113,115,122,123}, {115,122,123,112}, {122,123,112,113}, {123,112,113,115}
- One interval of size 5: {112,113,115,122,123}

The edges in the tree are assigned by splitting these intervals in two. Thus, each interval I={r_1,...,r_k} can be split into I_{left} and I_{right} as follows: I_{left}={r_1,...,$r_{\lceil k/2 \rceil}$} and I_{right}={$r_{\lceil k/2 \rceil}$,..., r_k}.

I_{left} and I_{right} are the children of I in the constructed tree. The tree for the example of Fig. 1 is shown in Fig. 4.

In accordance with the invention, the group encryption and decryption keys are obtained as follows. The leaf nodes 112, 113, 115, 122, 123 are each assigned a respective arbitrarily chosen private key and corresponding public key. This assignment may be done by the server 200 or by another entity.

The private key for a group associated with a particular node is obtained by recursively obtaining the group private keys of the nodes on a path from the leaf node corresponding to the party in question and the node corresponding to the authorized group. The private key associated with a particular non-leaf node is obtained by executing a key agreement protocol using a private key associated with a first child of the particular non-leaf node and a public key associated with a second child of the particular non-leaf node.

This process will now be discussed using the preferred Diffie-Hellman key agreement protocol. First some notation: the abbreviation S_x indicates the private (secret) key of party x, and P_x indicates the corresponding public key of that party.

The Diffie-Hellman key agreement protocol uses a generator g which is an integer less than a given prime number p, such that for every number n between 1 and p-1, there is a power k of g such that $n = g^k \bmod p$. For details on a secure implementation, see “Security Issues in the Diffie-Hellman Key Agreement Protocol” by Jean-François Raymond and Anton Stiglic, *IEEE Transactions on Information Theory,* pages 1–17, 1998.

As noted above, the leaf nodes 112, 113, 115, 122, 123, i.e. the individual devices, are assigned respective arbitrarily chosen private keys S_{112}, S_{113}, S_{115}, S_{122}, S_{123}. The corresponding public keys P_{112}, P_{113}, P_{115}, P_{122}, P_{123} are obtained in accordance with the Diffie-Hellman protocol:

$$P_x = g^{S_x}$$

All operations are of course performed in the finite group, but this has been omitted for the sake of brevity.

The private key of a non-leaf node is obtained using a private key associated with a first child of the particular non-leaf node and a public key associated with a second child of the particular non-leaf node. It does not matter which node's public key and which node's private key is used. For example, the private key S_{111} of node 111 can be obtained using the private key S_{112} of node 112 and the public key P_{113} of node 113, or using the public key P_{112} of node 112 and the private key S_{113} of node 113. The private key S_{111} is obtained as follows:

$$S_{111} = (P_{112})^{S_{113}} = \left(g^{S_{112}}\right)^{S_{113}} = g^{S_{112}S_{113}} = g^{S_{113}S_{112}} = \left(g^{S_{113}}\right)^{S_{112}} = (P_{113})^{S_{112}}$$

Note that both node 112 and node 113 are able to obtain this private key because of the symmetry in the above equation. All other nodes lack knowledge of either S_{112} or S_{113} and so cannot obtain S_{111}. The corresponding public key P_{111} is obtained using the same equation as above:

$$P_{111} = g^{S_{111}}$$

Nodes 112 and 113 are also able to obtain the private key S_{110} for group node 110 because they can obtain a private key associated with a first child of node 110, i.e. node 111, and a public key associated with a second child of the node 110, i.e. the public key of node 115. Node 115 can also obtain the private key S_{110} for group node 110 because it can obtain a private key associated with a first child of node 110, i.e. node 115 itself, and a public key associated with a second child of the node 110, i.e. the public key of node 111. Private key S_{110} is obtained as follows:

$$S_{110} = (P_{115})^{S_{111}} = \left(g^{S_{115}}\right)^{S_{111}} = g^{S_{115}S_{111}} = g^{S_{111}S_{115}} = \left(g^{S_{111}}\right)^{S_{115}} = (P_{111})^{S_{115}}$$

All public keys should be made available to all devices to enable them to obtain all group private keys for the groups of which they are members. With a device's assigned private key and access to all public keys, a device is able to obtain any private key for a group of which it is a member using the above method.

It should be noted that the above-mentioned embodiments illustrate rather than limit the invention, and that those skilled in the art will be able to design many alternative embodiments without departing from the scope of the appended claims. For instance, although in the above the keys are assigned to devices, it is equally possible to assign keys to persons, households or other entities to enable decryption by groups of these entities at any device rather than at one specific device.

In the claims, any reference signs placed between parentheses shall not be construed as limiting the claim. The word "comprising" does not exclude the presence of elements or steps other than those listed in a claim. The word "a" or "an" preceding an element does not exclude the presence of a plurality of such elements. The invention can be implemented by means of hardware comprising several distinct elements, and by means of a suitably programmed computer.

In a device claim enumerating several means, several of these means can be embodied by one and the same item of hardware. The mere fact that certain measures are recited in mutually different dependent claims does not indicate that a combination of these measures cannot be used to advantage.

CLAIMS:

1. A method of key management for group-based controlled access to encrypted data, in which a decryption key for the encrypted data can be obtained by a party if the party is a member of at least one group which is authorized to access the data,
the groups being organized in a hierarchical tree in which each non-leaf node represents a group and each leaf node represents a member of all groups represented by nodes hierarchically superior to the leaf node in question,
characterized in that the leaf nodes are each assigned a respective arbitrarily chosen private key and corresponding public key,
in that the private key associated with a particular non-leaf node is obtained by executing a key agreement protocol using a private key associated with a first child of the particular non-leaf node and a public key associated with a second child of the particular non-leaf node,
and in that the private key for a group associated with a particular node is obtained by recursively obtaining the group private keys of the nodes on a path from the leaf node corresponding to the party in question and the node corresponding to the authorized group.

2. The method of claim 1, in which the key agreement protocol comprises the Diffie-Hellman key agreement protocol.

3. The method of claim 1, further comprising constructing the hierarchical tree from a given two-dimensional access control table that represents for each of a number of parties and each of a number of data items whether the particular party is allowed to access the particular data item.

4. The method of claim 1, further comprising constructing the hierarchical tree by grouping the users into intervals and ordering the intervals.

5. The method of claim 1, further comprising encrypting the data with an arbitrarily chosen data key and creating an encrypted data key by encrypting the data key with a public key for a group which is authorized to access the data.

6. The method of claim 5, further comprising creating multiple encrypted data keys each of which is created by encrypting the data key using a respective public key for a respective group which is authorized to access the data, thereby allowing members of each of the respective groups to obtain the data key.

7. The method of claim 1, further comprising encrypting the data with an arbitrarily chosen data key and creating an encrypted data key by encrypting the data key with a key encryption key that has been derived from the private key for a group which is authorized to access the data. Preferably the key encryption key is obtained by applying a cryptographic hash function to the private key in question. This embodiment obviates the need for public key encryption.

8. A device for group-based controlled access to encrypted data, configured for obtaining a decryption key for the encrypted data if the device is a member of at least one group which is authorized to access the data, the groups being organized in a hierarchical tree in which each non-leaf node represents a group and each leaf node represents a member of all groups represented by nodes hierarchically superior to the leaf node in question, characterized in that the device has available an arbitrarily chosen private key and corresponding public key, and in that the device is configured for obtaining the private key associated with a particular non-leaf node by executing a key agreement protocol using a private key associated with a first child of the particular non-leaf node and a public key associated with a second child of the particular non-leaf node,

and for obtaining the private key for a group associated with a particular node by recursively obtaining the group private keys of the nodes on a path from the leaf node corresponding to the party in question and the node corresponding to the authorized group.

9. A computer program product comprising instructions for causing a processor to execute the method of claim 1.

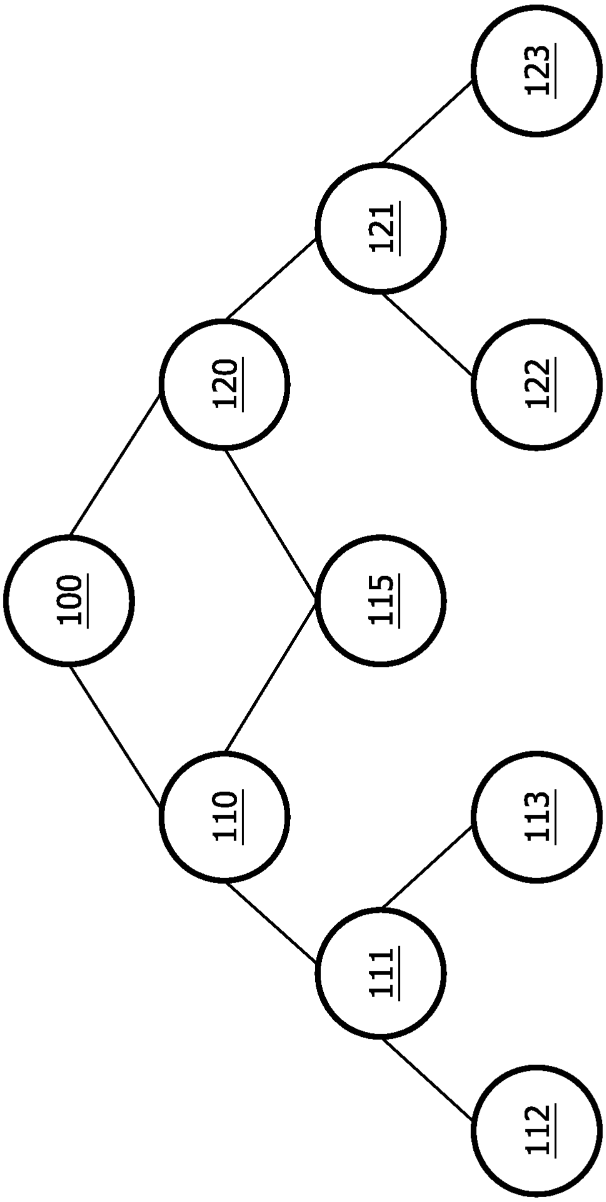


FIG. 1

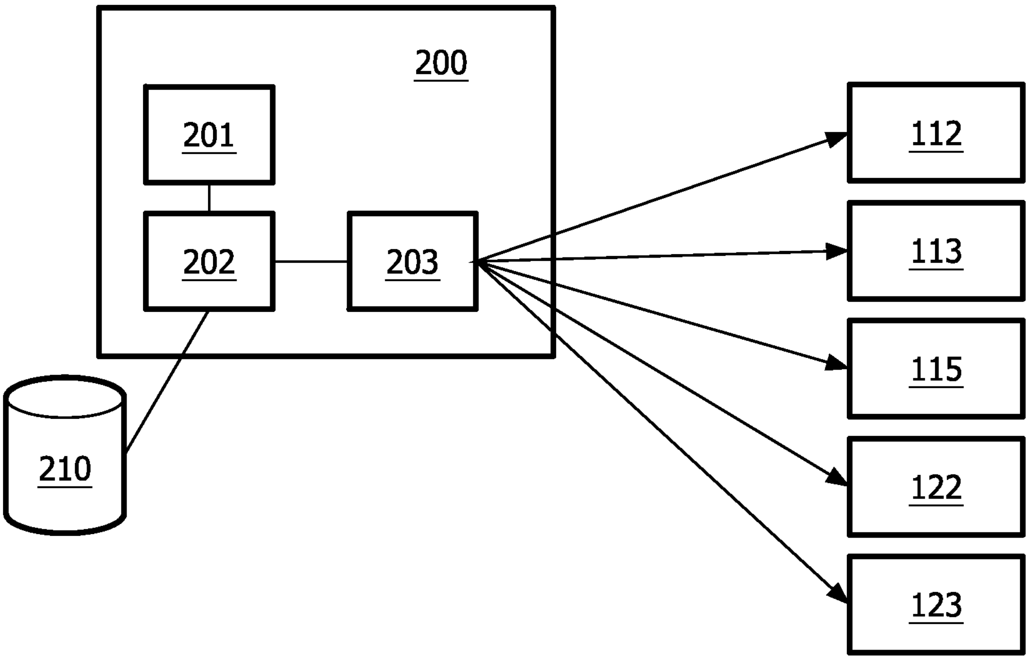


FIG. 2

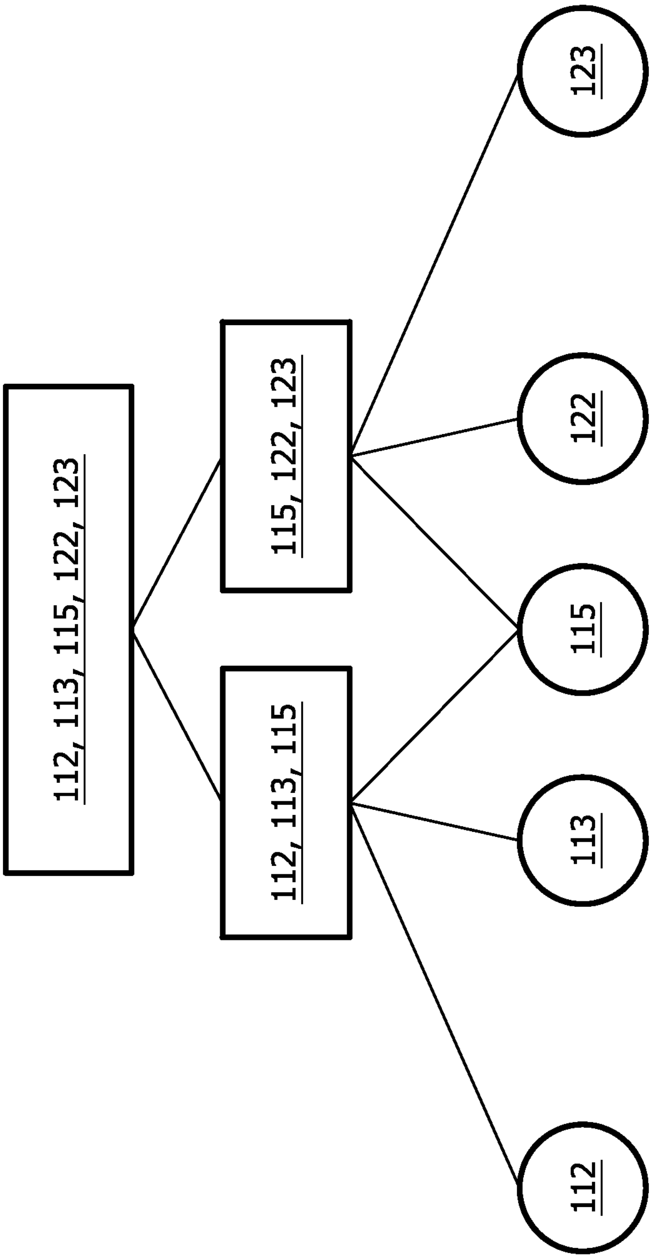


FIG. 3A

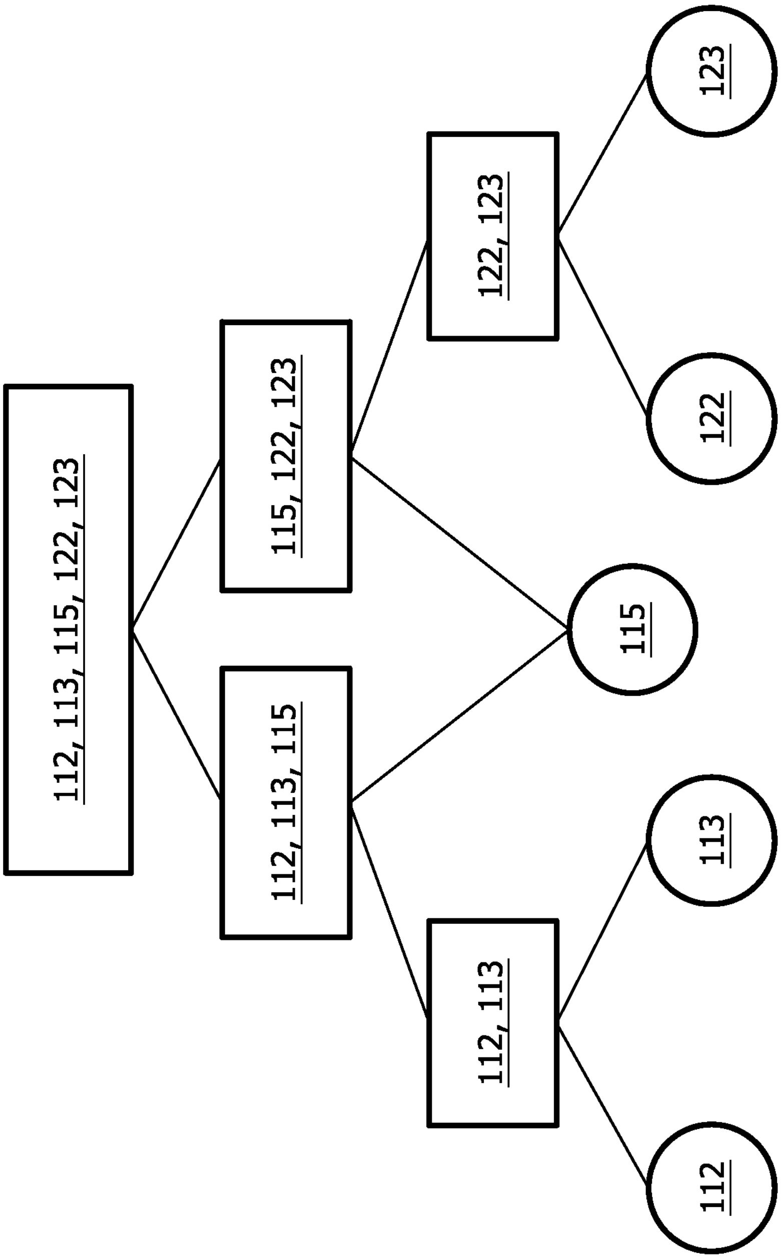

112, 113, 115, 122, 123
115, 122, 123
112, 113, 115
122, 123
112, 113
123
122
115
113
112

FIG. 3B