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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
20 November 2008 (20.11.2008)

PCT

(10) International Publication Number
WO 2008/140798 A1

(51) International Patent Classification:
H04L 9/00 (2006.01)

(21) International Application Number:
PCT/US2008/006027

(22) International Filing Date: 12 May 2008 (12.05.2008)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/928,716 11 May 2007 (11.05.2007) US

(71) Applicant *(for all designated States except US)*: **INDIANA UNIVERSITY RESEARCH & TECHNOLOGY CORPORATION** [US/US]; 351 West 10th Street, Indianapolis, IN 46202 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **XUKAI, Zou** [CH/US]; 5724 Fairbourne Court, Carmel, IN 46033 (US). **DAI, Yuanshun** [CH/US]; 409 East Stadium Hall,, Knoxville, TN, 37996 (US).

(74) Agent: **HOUDEK, Jason, A.**; Krieg Devault LLP, One Indiana Square, Suite 2800, Indianapolis, IN 46204-2079 (US).

(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, 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, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, 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:
— *with international search report*

(54) **Title:** FLEXIBLE MANAGEMENT OF SECURITY FOR MULTI-USER ENVIRONMENTS

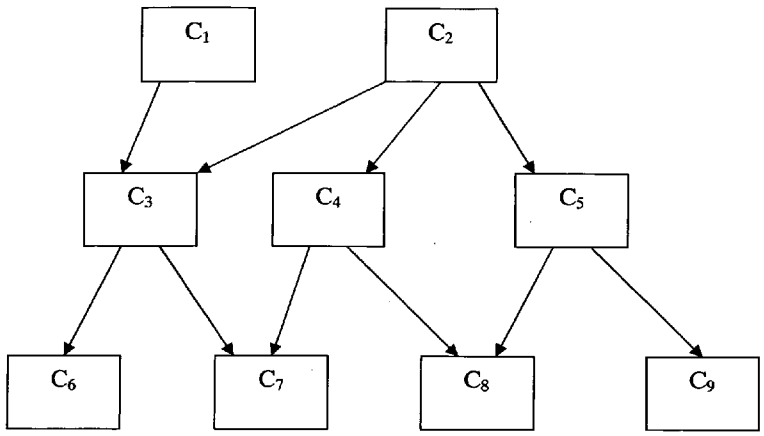

Fig. 2

(57) **Abstract:** One embodiment is a method including computing or storing in a computer accessible medium a first polynomial which is a function of a set of numbers each associated with a member of a group to be provided a cryptographic information, determining a second polynomial which is a function of the first polynomial and an information to be privately shared with the group, and using at least one of the first polynomial and the second polynomial in providing communication between or among two or more members of the group. Further embodiments include systems and computer readable media including an access control polynomial.

WO 2008/140798 A1

FLEXIBLE MANAGEMENT OF SECURITY FOR MULTI-USER ENVIRONMENTS

BACKGROUND

Multiuser environments, such as trusted collaborative computing ("TCC") environments, present a number of unmet challenges including those relating to secure group communication (SGC), secure dynamic conferencing (SDC), differential access control (DIF-AC), hierarchical access control (HAC), and other functionalities. Cryptography and key management have been investigated in various attempts to secure information; however, until now there has been no mechanism which is able to address the requirements for trusted or secure information transmission and data access in TCC or other multiuser environments.

SUMMARY

One embodiment is a method including computing or storing an access control polynomial. Further embodiments include systems and computer readable media including an access control polynomial. Further embodiments, forms, objects, features, advantages, aspects, and benefits shall become apparent from the following description and drawings.

BRIEF DESCRIPTION OF THE FIGURES

Fig. 1 is an exemplary CC environment.

Fig. 2 is an exemplary access control hierarchy.

DETAILED DESCRIPTION

For the purposes of promoting an understanding of the principles of the invention, reference will now be made to the embodiments illustrated in the drawings and specific language will be used to describe the same. It will nevertheless be understood that no limitation of the scope of the invention is thereby intended, and that all alterations and further modifications of the following embodiments and such further applications of the principles of the invention as would occur to one skilled in the art to which the invention relates are contemplated.

With reference to Fig. 1, there is illustrated an exemplary collaborative computing ("CC") environment 100. Exemplary CC applications include, but are not limited to, multi-party military actions, tele-conferencing, video conferencing, tele-medicine, video medicine, interactive and collaborative decision making or conferencing, grid-computing, information distribution, and pay per view services. Further examples include enterprise management software and related applications, electronic mail systems and archives, key management systems, and others. Trust and/or security in such environment can eventually determine its success and popularity due to the desire for confidentiality, privacy and integrity of personal and/or shared information. Existing communication infrastructure such as the internet does not provide high assurance security for data transmission. Security patches and other computing/storage resources available to hackers result in more security vulnerabilities. Compared to two-party interaction models (such as the client-server service model), multiuser and CC environments may present additional challenges owing to the environments being group-oriented, involving a large number of entities and shared resources, being complex, dynamic, distributed, and heterogeneous and even possibly including hostile elements. Systems

experience failures due to intrusions and attacks from hostile entities. In addition, there is the problem of insider threats, by which attacks are from malicious parties inside the organizations or members of CC groups. Consequently, establishing and maintaining trusted collaborative computing (TCC) environments is very difficult.

As illustrated in Fig. 1, exemplary CC environment 100 is complex and includes a diverse, heterogeneous group of users, resources, systems, communication links, hierarchies, access authorities, and may include internal and external threats. A central server 111 may distribute information to and receive information from a plurality of group members such as group members 101, 102, 103, 103a, 103b ... 103n, 109, 112, and 116 via one or more communication links. Group members can also form sub-groups, such as the sub-group including members 103, 103a, 103b ... 103n. Sub groups could also include greater or fewer numbers of members. The membership of groups and subgroups is dynamic and can increase or decrease. The functionalities of server 111 may also be distributed, for example, a second server 110 may also distribute information to and receive information from a plurality of group members such as group members 113, 114, 112, and 116 via one or more communication links. The nature of the distribution may be physical, virtual, or a combination thereof. In a exemplary embodiment, the central server is a server cluster, such as a blade or rack server system, with physical and software interconnections among cluster servers.

A variety of communication links are also illustrated in environment 100. Communication links may be electrical, magnetic, optical or combinations thereof. Communication links can also be wireless, such as the point-to-point wireless link interconnecting server 111 and group member 102, or point to multipoint wireless link between group members 106 and 107 and point to multipoint transceiver 105. One example of a point to

point wireless link is a microwave transmission link. One example of a point to multipoint wireless link is a cell phone network such as one utilizing CDMA, TDMA, FDMA and other types of transmission protocols and systems. Another example is a WIFI network. A further example is a satellite network such as a direct broadcast satellite network. An additional example is a WIMAX network. There are a plurality of user types that may utilize such networks including cell phone, computer, PDA, video conferencing, audio conferencing, and other types of users. The communication links may include routers such as router 108, repeaters such as repeater 104, and other communication link features such as feature 115. Communication links may follow a variety of protocols such as IP, TCP, UDP, VOIP, SSL, and others, and may facilitate communication of a variety of types of information, such as packets, data, voice, picture, video and/or audio information. A variety of system and user resources, such as resource 110a of server 110, and resource 116a of user 116 may also be present in environment 110.

A exemplary embodiment may establish a trusted collaborative computing (TCC) environment to facilitate user collaboration in which entities work together and share resources and/or information. One security issue for such environments is that multiple participating entities should be able to communicate securely among one another via one or more communication channels. Techniques such as conventional IP multicast permit transmission of messages to a group of users; however, the open nature of conventional IP multicast makes it unable to provide strong confidentiality. Another security issue is related to resource sharing and data exchange. Access to shared resources/data may need to be precisely and accurately controlled; otherwise attackers and malicious users can access resources to which they are not entitled to access, abuse, tamper, and/or damage. Selective data sharing, at different granularity

levels, along with access control is another security issue. It may be desirable for these classes of functions to be sufficiently flexible as to support various possible forms of interactive access relations between the parties and the resources in the system. Thus, security issues relevant for multi-user environments and TCC include hierarchical access control (HAC), secure group communication (SGC), secure dynamic conferencing (SDC), and differential access control (DIF-AC). Cryptography is a powerful tool to support all these and other security functions. Key management is a difficult issue in such context, and the generation, distribution, updating, and revocation of keys, such public keys, private keys, security tokens, seeds, or identifiers, in such environments, which may be large and dynamic, is a significant challenge.

Exemplary embodiments include an Access Control Polynomial ("ACP"). Some embodiments include an ACP through which secret information can be distributed so that only the intended recipients (i.e., their IDs are included as a term $(x - f(ID))$ in the polynomial) can derive that secret information. Some embodiments utilize an ACP to support security in highly dynamic environments where, for example, users join/leave and there are addition/deletion of resources/data/messages, addition and removal of user/resource relations, random user/data structures/formats according to fine-tuned granularity (e.g., in the levels of users, user groups, data sets, data records, record fields), and/or anonymity (i.e., group membership and size can be hidden from both outsiders and insiders). Some embodiments utilize an ACP to support a plurality of different security functions and provide integration of various application systems. Some embodiments provide resistance or immunity to various attacks, including external hackers and internal malicious members, and even collusion between internal and external attackers.

An ACP can be described with mathematical rigor. For this discussion the following notation will be used (though different notation might also be applied in other contexts):

$A(x)$	The access control polynomial in the form of $A(x) = \prod_{i \in \psi} (x - f(SID_i, z))$
F_q:	The finite field
f :	A public cryptographic hash function. It is used in the form of $f(x,y)$, i.e. $f(x \| y)$
GID_i :	Secret Group Identification, a positive integer
$P(x)$	The public polynomial sent to users for key distribution, $P(x) = A(x) + K$
q:	A large prime, as a predefined system parameter
SID_i:	Personal Permanent Portable Secret, a positive integer
U_i	A group member in a certain group
v_j	A certain vertex in the hierarchy
z	A random integer which is changed and made public every time.
%	Mod operation

Let us consider exemplary environments having the following characteristics: (1) q is a large prime from which a finite field F_q is formed, preferably a large prime number, such as 512 bits, 1024 bits, or an even greater number of bits, (2) $f : \{0,1\}^* \rightarrow \{0,1\}^q$ is a cryptographic hash function, and (3) there is a trusted system component, resource, or computer, such as, for example, a server. Every valid user, say U_i, in the system is assigned a Personal Permanent Portable Secret, called P3-Secret and denoted as SID_i (a random positive integer less than q). This secret is only known to the user and the central server. Since users are generally required to register to the system, the assignment of an SID to a user can be performed during the registration procedure, for example, by using a two-party security mechanism.

An exemplary ACP is a polynomial over a finite field F_q [x] and defined as follows.

$$A(x) = \prod_{i \in \psi} (x - f(SID_i, z)) \qquad \text{Eq. (1)}$$

where ψ denotes the user group under consideration, SID_i are group members' P3-Secrets assigned to the members in the group ψ, and z is a random integer from F_p and is made public. In addition, z is changed every time $A(x)$ is computed. $A(x)$ is equated to 0 when x is substituted with $f(SID_i, z)$ by a valid user with SID_i in the group ψ; otherwise, $A(x)$ is a random value if other numbers or invalid users' P3-Secrets are used in the substitution.

In order to broadcast a secret value such as K to the users in group ψ, the following polynomial can be computed (for example, by a trusted server):

$$P(x) = A(x) + K \qquad \text{Eq. (2)}$$

Then, $(z, P(x))$ is distributed or publicized (for example, broadcast) and K is hidden, mixed with $A(x)$. From this public information, any group member U_i with SID_i can obtain the secret value, K, by:

$$K = P(f(SID_i, z)) \qquad \text{Eq. (3)}$$

Utilizing an ACP, key management for a large range of security functions and applications can be accomplished. For example, ACP key management can be accomplished for SGC, SDC, DIF-AC and/or HAC.

SGC refers to a setting in which a group of members can communicate (or share the information) among themselves, in a way that outsiders are unable to understand the communication (or the information) even when they are able to intercept the communication (or the information). The confidentiality of the SGC communication is provided by encrypting the communication with a group key which is distributed to only the group members.

In one SGC embodiment a trusted server computes $A(x)$ by Eq. (1) (Step 1) , $P(x)$ by Eq. (2), and then multicasts $(z, P(x))$ (Step 2). Every user in the group can then compute the key via Eq. (3) (Step 3). After all group members obtain the same key, they can conduct group communication securely.

Let us consider group dynamics. Users can join, leave or be revoked from the system. From the construction of $A(x)$, it can be seen that regardless of whether we deal with single join, single leave, multiple joins, multiple leaves, or multiple joins and leaves simultaneously, dynamics can be implemented with great elegance and easily: the above steps 1), 2), and 3) are followed but in the formation of A(x), just the joining users' *SID*s (in fact, $f(SID_i, z)$) are included and the leaving users' *SID*s are excluded. Note that z and K in these steps are new random numbers. Once the key is changed, the encryption with the new key will prevent the leaving (or joining) users from accessing the future (or the past) information.

SDC refers to a scenario where any subset, for example a random subset, of the given user population can form a secure communication (sub)group. As it is evident, SDC is closely related to SGC: as an extension of SGC or equivalently, SGC as a specific case of it. Suppose the size of the universe under consideration is n, there will be 2^n-n-1 possible conferences. Pre-generating all these 2^n-n-1 conferences might not be preferred because many conferences may never need to be activated. In addition, conferences may not occur at the same time.

A preferred ACP embodiment includes an on-the-fly feature, which means that whenever there is a need to distribute a secret to a specific user group, just the above steps 1), 2), and 3) are executed. This feature is useful for supporting SDC. Whenever there will be a conference of any subset of users, the server just performs the three steps where $A(x)$ includes *SID*s of the conference members. If a user participates in multiple conferences at the same time, the user's

SID can be included in multiple corresponding $A(x)$'s and the user then can get the keys for all these conferences. Whenever users want to join or leave a conference, the above three steps are executed with $A(x)$ just including the intended users. Thus, group dynamics can be efficiently processed in SDC.

Access control is used for checking whether a user has the right to access a certain resource or information and for granting or denying access as required. Access control can be a fundamental security issue for many computing systems in which users and resources are involved. In DIF-AC, a user can (and only can) access certain resources and a resource can (and only can) be accessed by certain users (i.e., many-to-many relation, determined by, for example, subscription and payment). Exemplary applications requiring DIF-AC include, but are not limited to, e-newspapers, pay-per-view broadcast TV, multiple streaming services and/or secret or confidential communications.

Like the above SDC scheme, every resource R_k is associated with a dynamic key K_k, and the users who can access R_k are treated as a conference. The server computes $A_k(x)$ and $P_k(x)$, and publicizes $(z, P_k(x))$. Thus, the user, who can access R_k, can derive key K_k and is granted access to resource R_k. If a user can access multiple resources, the user's SID_i will be included in the $A_k(x)$'s of all these resources. Thus, the user can access all these resources. Similarly, dynamics can be implemented by inclusion and exclusion of users' *SID*s in the formation of new $A_k(x)$'s.

HAC occurs when resources (and users) have some hierarchical relation: resources are assigned levels and a user who has the access right to a resource at one level is automatically granted access to the resources which are the resource's children or descendants at lower levels. However the reverse is not allowed. The most generic format of HAC can be represented as a

Directed Acyclic Group (DAG) (as illustrated in Fig. 2). A node in the hierarchy can represent a user, a resource, a set of users, a set of resources, or both users and resources.

For every node/class C_k in the hierarchy, the server selects a unique CID_k and distributes securely CID_k to C_k's users $\{U_1, U_2, \cdots, U_n\}$ using the same scheme as that in SGC, i.e., the server computes $P(x) = (x - f(SID_1, z)) \cdot (x - f(SID_2, z)) \cdots (x - f(SID_n, z)) + CID_k$ and multicasts $(z, P(x))$ to C_k's users. The server also selects a dynamic key K_k for every C_k. Now, the server constructs $A_k(x)$ using this node's CID_k as well as CIDs of all its ancestors:

$$A_k(x) = (x - f(CID_k, z)) \prod_{i \in \psi} (x - f(CID_i, z)) \qquad \text{Eq. (4)}$$

where the first term is C_k itself and the next terms are associated with all the ancestors C_i of C_k (ψ is the set of ancestors of C_k). Then, the server constructs $P_k(x) = A_k(x) + K_k$ and publicizes $(z, P_k(x))$. The node C_k (i.e., the users in C_k) can compute the key K_k as $K_k = P_k(f(CID_k, z))$. Furthermore, any ancestor (i.e., the users in) C_i of C_k can also derive the key K_k as $K_k = P_k(f(CID_i, z))$. However, C_k cannot reversely get C_i's key. Thus, the hierarchical access control is correctly and securely enforced.

In this ACP-based HAC scheme, the key derivation by the node's ancestors is performed in the identical way as the key computation by a node. Moreover, nodes do not need to know the exact hierarchy. The nodes that are ancestors of a node will obtain the correct key of the node when substituting their CID into $P(x)$ but others will not.

There are two level dynamics in HAC: node level and user level. The node level dynamics include adding a node, deleting a node, moving a node from one place to another, adding one link between two nodes, and deleting a link between two nodes. User level dynamics

include addition and deletion of a user from a node group and movement of a user from one node group to another. Based on ACP, both level dynamics can be accomplished efficiently.

Let us consider the operation of deleting a node, since revocation/deletion is generally more difficult to deal with than joining/addition. There are two cases to consider: a leaf node and an internal node. If the deleted node is a leaf node, nothing needs to be done other than discarding the information/values related to this node. If the deleted node is an internal node, a technique should be used to relocate the node's children, for example, a relocation policy or algorithm. However, the particular technique used for such purpose does not matter here. Since the deleted node knew the keys of all its descendants, these keys need to be changed, which is easy. For each of the descendant nodes of the deleted node, the server computes $A(x)$ which includes the *CID*s of all new ancestors of the node but excludes the *CID* of the deleted node and multicasts $(z, P(x)=A(x)+K)$.

Consider the second level dynamics. For example, if one member (with SID_l) leaves group C_k and attends another group C_j, the following two steps complete the update.

1) The new node *CID* in node C_k is updated by the above polynomial excluding the term $(x - f(SID_l, z'))$ (Note: a new z' is used).

2) The new node *CID* of the group C_j is updated with the above polynomial including the term $(x - f(SID_l, z''))$ (Note: a new z'' is used).

An ACP embodiment can address the HAC problem in the same manner and the same efficiency of SGC/SDC. Exemplary applications involving HAC include government or private organization computer systems, digital libraries, medical information systems, systems storing proprietary information, and systems including other confidential or limited access information.

We now analyze the security and performance of the above ACP embodiment. By the security analysis, we show that the proposed ACP mechanism is very robust and secure not only against outside attackers which do not know the shared key but also against the insiders which know the shared key. By the performance analysis, we show that the ACP mechanism is very efficient.

We discuss the security of ACP embodiments in terms of external attackers, internal attackers, and collusion of attackers. First, let us consider the key space and the guessing or brute-force attack. K is randomly and uniformly selected from 0 to q-1. In addition, K can be coincident with any of SID_i and $v_i = f(SID_i, z)$, for $i = 1, \cdots, n$ since it will not affect the correctness of the ACP mechanism. Thus, the introduction of the access polynomial (no matter how high its degree is) will not reduce the size of the key space. As for the brute-force attack, an external attacker can either guess K directly or guess one of v_i and then compute K, or guess one of SID_i and compute v_i and then K. The probability that a random guess hits K is $1/q$ whereas it is n/q to hit any of v_i and another n/q to hit any of SID_i. Thus, the overall probability for a random trial to success is $(2n+1)/q$. This means that the access control polynomial increases the success chance of the brute-force attack by a factor of $2n$. The more users are included in the polynomial, the higher the probability of success by the brute-force attack. However, due to the efficiency of the ACP mechanism (as discussed below), q can be selected to be very large, thus, making the brute-force attack inapplicable. Next, let us consider the attacks in which an external attacker tries to obtain the group key K or group users' SIDs from $P(x)$. The K is hidden in the publicized constant term of $P(x)$, i.e. $c_0 = (K + V)\%q$ where $V = v_1 \cdot v_2 \cdots v_n$ and $v_i = f(SID_i, z)$, for $i = 1, \cdots, n$. Since there are many other pairs of K' and V' such that

$c_0 = K'+V'$, the attacker cannot uniquely determine K from c_0. As for trying to determine all of $K, v_1, v_2, \cdots, v_n$ from (the coefficients of) $P(x)$ at the same time, the attacker will fail because only n equations can be formed for n+1 unknown $K, v_1, v_2, \cdots, v_n$. As for trying to determine SID_i, the only relevant value is $v_i = f(SID_i, z)$ which is difficult to be obtained from $P(x)$ as discussed above. Even if the attacker were able to determine $v_i = f(SID_i, z)$ somehow, the attacker still would not be able to get SID_i since this would require inversion of the cryptographic hash function f. Finally, multiple external attackers may collude to determine K or SID_i, but their collusion provides no more information than the information that would be obtained by a single attacker; collusion is thus useless. ACP embodiments are resistant to external attacks.

We now consider the case of internal malicious users. Obviously, an internal user can obtain K from its own SID_i. Thus the purpose of an internal malicious user is to obtain the *SID*s of some other users so that he can get the secret information, reserved to other users, to which he is not authorized to access. He can obtain the exact polynomial $A(x)$ as $A(x)=P(x)-K$ and then set $A(x)=0$ to determine the roots of $A(x)$. He may find $v_i = f(SID_i, z)$, however, it is computationally infeasible to get SID_i from $v_i = f(SID_i, z)$ due to the one-way feature of the cryptographic hash function f. Getting v_i of the other user does not therefore help the attacker. First, v_i will result in K to be disclosed, but this does not help at all because he had been allowed to get K from his own *SID*. Additionally, this $v_i = f(SID_i, z)$ can be only used for getting this K and cannot help in determining any other keys from other $P(x)$'s because z is updated every time and two v_is in two $P(x)$'s will be different even though SID_i is the same. As a result, the internal malicious user cannot violate the security of the ACP embodiment. Furthermore, it is useless for multiple internal users to collude because their collusion cannot help to make the inverse of the

cryptographic hash function easier, thus, making impossible to get SID_i from v_i. The collusion of internal malicious users and external attackers is also useless in getting other users' *SID* (Note: the collusion here does not include the case of an internal user giving his *SID* or the key to an outsider so that the outsider can access the information. If this case is considered as a collusion, then it is inherent in all cryptosystems and there is no technological solution to it).

The attackers may hope to glean multiple $P(x)$'s and try to get useful information from them; however, this attempt would also be useless due to the changing $P(x')$s. There are different forms of collusions in the hierarchy such as two siblings trying to figure out their parent's key, a node and its nephew trying to figure out its parent key. However, these cases of attacks can be reduced to the collusion of external attackers, or internal malicious members or internal/external users depending on whether (and how many) their *SID*s are included in $P(x)$. As discussed above, a preferred ACP embodiment is able to defend against any such collusion.

The storage complexity (at both user end and server end), computation complexity (at both the user end and server end), and communication complexity can be analyzed. The user-end storage cost is $O(1)$ since a user just needs to store its P3-Secret *SID* (plus its node *CID* if in the HAC hierarchy). The server storage cost is $O(n+m)$ since the server needs to store all n users' *SID*s (plus m nodes *ID*s if in the HAC hierarchy). Suppose there are n terms involved in the generation of $P(x)$. There are two parts to consider. The first part is related to computing $f(SID, z)$. The running time of the cryptographic hash function totally depends on itself but is independent from the number of terms n. Suppose its running time is $O(B)$, then computing n $f(SID, z)$ has a cost in $O(nB)$. The other part is to multiply n terms $(x\text{-}v)$'s. The main operations are multiplication (with modulo) and addition (with modulo). There are in total $O(n^2)$ of such operations. The computation complexity for multiplying n terms $(x\text{-}v)$'s is in

$O(n^2)$. Thus, the total computation complexity for generating $P(x)$ is in $O(nB+n^2)=O(n^2)$. This polynomial computation complexity is efficient for the server. We now consider the computation complexity for computing the key from a polynomial $P(x)$ of degree n when replacing x with the computed value $v=f(SID,z)$. The main operations here are: 1) the computation of $v, v^2\%q, \cdots, v^n\%q$ which requires n multiplications (with modulo); 2) the multiplication of each of these values with its corresponding coefficient, which requires another n multiplications; and 3) the addition of the results, which requires n additions. In total, the complexity of computing the key from $P(x)$ is in $O(n)$. With respect to the communication complexity, broadcasting $P(x)=a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ requires to broadcast the coefficients $a_n, a_{n-1}, \cdots, a_1, a_0$. Thus, the communication complexity is in $O(n)$. These complexities are summarized in Table 1 below. Note: key derivation is similar to key computation.

Table 1. Complexities of the ACP based key management

Terms	Complexity		Terms	Complexity
User end storage	$O(1)$		**Key computation**	$O(n)$
			Key derivation	$O(n)$
Server end storage	$O(n+m)$		***P*(*x*) generation**	$O(n^2)$
			Communication	$O(n)$

From the above complexity analysis, it is clear that all complexities are proportional to n, the number of current users in the group. If n is large but just a single user or few users join or leave the group, $O(n)$ or $O(n^2)$ is not efficient. There are several ways to improve its efficiency. 1) As for join, the server can just generate a new key and encrypt the new key with the old group key and send it to the group. The server also encrypts the new key with the *SID* of

the joining user and sends it to the joining user. 2) In order to improve the efficiency of computing $P(x)$, we can store and save $A(x)$ in advance. If one or a few users $U_1,\cdots,U_k$ leave, we can get the new A(x) by directly dividing $A(x)$ by $(x-f(SID_1,z))\cdots\cdots(x-f(SID_k,z))$, thus, the complexity for $P(x)$ generation will reduce to $O(n)$. 3). For improving the efficiency of key computation and derivation, we can divide the n users into $k=n/l$ separate groups of l users each. The server forms k polynomials of degree l each. Every user can obtain the key by replacing its own *SID* to its corresponding polynomial. Thus, the complexity for key computation/derivation will reduce to $O(l)$. Next, we describe a mechanism which can improve the efficiency greatly: tree based multiple level and hierarchical grouping.

Suppose n is the number of all users and m is the size of a small group which can be managed easily and efficiently, for example, m=16. Then every m users form a first level group, so a total n/m of such groups $G_{1,1},\cdots,G_{1,n/m}$ are formed. Next, every m first level groups form a second level group, thus, a total n/m^2 of such groups $G_{2,1},\cdots,G_{2,n/m^2}$ are formed. By continuing with this strategy, finally a highest level group is formed $G_{\log_m{}^n,1}$. All these groups can be treated as nodes in an m-ary tree of height $\log_m{}^n$. Every group $G_{i,j}$ is associated with a group key $K_{i,j}$ and the $K_{\log_m{}^n,1}$ will be the group key for all users. The group keys are distributed to their members using an ACP embodiment. For example, $K_{1,j}$ is distributed to group $G_{1,j}$ by forming the ACP polynomial using the *SID*s of the users in its group, i.e. $P_{1,j}(x)=\prod(x-f(SID_i,z))+K_{1,j}$ where $U_i\in G_{1,j}$. The second level key $K_{2,j}$ is distributed to all users belonging to group $G_{2,j}$ by forming the ACP polynomial using the group keys of its first level groups, i.e. $P_{2,j}(x)=A_{2,j}(x)+K_{2,j}=\prod(x-f(K_{1,i},z))+K_{2,j}$ where $G_{1,i}\in G_{2,j}$.

Finally, the highest level key will be distributed by forming $P_{\log_m{}^n,1} = \prod(x - f(K_{\log_m{}^n-1,i}, z)) + K_{\log_m{}^n,1}$.

Let us consider the case of a single user leaving his group. The group keys along the path from the leaf group of the leaving user to the root group need to be changed. Total $\log_m{}^n$ polynomials of degree m need to be computed and broadcast. Thus, the total polynomial generation time will be $O(m^2 \log_m{}^n)$, the communication complexity is $O(m \log_m{}^n)$, and the key computation and derivation are also in $O(m \log_m{}^n)$. For example, suppose m=16, $n = 2^{64}$, then the polynomial generation time, key computation time, and communication complexity are in 2048 units of time, 256 units of times, and 256 units of numbers for transmission.

An ACP embodiment can preferably hide the group membership and size from outsiders (and even insiders) preferably without member serialization. Without a preferred ACP embodiment, in a multicast to a group of users, the information identifying users would need to be included in the multicast packet, and the users would need to be ordered according to some strategy (referred to as *serialization*), so that each user knows which portion of the protected key material belongs to him and is thus able to extract the group key from that portion. This would not only result in more computation work (e.g., a user needs to search for his portion) and need synchronization due to the serialization but also unintentionally result in disclosures concerning the group membership information. Keeping group membership information private to outsiders may be important in some applications. Furthermore, it may be desirable or even necessary to hide the group membership from the group users themselves in some applications, for example, a user knows that he is in the group but does not have knowledge about which are the other members of the group. It may also be desirable to hide the size of the group.

A preferred ACP embodiment provides an efficient and elegant solution to address one or more (even all) of the aforementioned features wherein a polynomial hides the group users and does not need to sort the group members. A valid user does not need to know (in fact, he cannot know if the server does not want to tell him) the membership and the order of members but he can get the group key easily by just plugging its *SID* into the polynomial. A preferred ACP embodiment can be easily extended for the purpose of hiding group size by simply including some random pseudo terms in the polynomial such as:

$$A(x) = \prod_{i \in \psi} (x - f(SID_i, z)) \prod_{j=1 \cdots d} (x - VID_j)$$

where $VID_1, \cdots, VID_d$ are random numbers in F_q, called pseudo terms, and d is a random positive integer. As a result, the degree of $P(x)$ does not indicate the number of members involved in the computation. These pseudo terms make $P(x)$ even more randomized.

Adding random terms will increase the degree of $P(x)$, thus, impacting the efficiency of the ACP embodiment. However, using the extended tree-based key distribution mechanism discussed above, the impact on efficiency is reduced. Decisions whether to add or how many random terms to be added is a trade-off between security and efficiency, and are preferably determined based on the requirements of concrete applications.

A preferred ACP embodiment is powerful enough to adapt to random forms of interactive/access relations among users and/or resources. These relations include, but are not limited to, equivalent users/resources, one-to-many, many-to-one, many-to-many, hierarchy, multiple levels, etc. For example, if a node C_i's access permission needs to be transferred to a random other node C_j , regardless of the relation and distance between the two nodes in the hierarchy, just include C_j's CID_j in the construction of $A_i(x)$.

An additional exemplary embodiment includes software stored in a computer accessible medium including an ACP which is: adaptable to different kinds of key management and different kinds of access control relation schemes; able to enforce access control and secure group communication at a plurality of scales and granularities; able to integrate heterogeneous data sources and systems; able to protect against external attacks, internal attacks, and combined external and internal attacks; supports dynamic environments including the adding and/or revocation of members and/or resources; does not require member serialization or synchronization and does not disclose membership; able to hide the identities of members of the group and the group size; and able to implement flexible key management on the fly. A further exemplary embodiment is a system which utilizes such software. Another exemplary embodiment is a method which utilizes the functionalities of such software.

One exemplary embodiment is a method including computing or storing in a computer accessible medium a first polynomial which is a function of a set of numbers each associated with a member of a group to be provided a cryptographic information, determining a second polynomial which is a function of the first polynomial and an information to be privately shared with the group, and using at least one of the first polynomial and the second polynomial in providing communication between or among two or more members of the group. A further exemplary embodiment includes the providing communication between or among two or more members of the group includes providing at least one of a trusted collaborative computing environment, a secure dynamic conferencing environment, a differential access control environment, and a hierarchical access control environment. In a further exemplary embodiment the first polynomial is a function of a public random number. In a further exemplary embodiment the first polynomial includes a term which is zero when evaluated with one of the

set of numbers each associated with a member of a group to be provided a cryptographic key. In a further exemplary embodiment the first polynomial is described by the formula:

$$A(x) = \prod_{i \in \psi} (x - f(SID_i, z))$$

where $A(x)$ denotes the first polynomial, i denotes a member of the group, ψ denotes the group, SID_i denotes the numbers each associated with a member of the group, and z denotes a random number. In a further exemplary embodiment the cryptographic information is a cryptographic key. A further exemplary embodiment includes distributing the second polynomial to the group. A further exemplary embodiment includes at least one member of the group receiving the second polynomial. A further exemplary embodiment includes obtaining the cryptographic information from a distributed polynomial. A further exemplary embodiment the includes obtaining the cryptographic information by calculating

$$K = P(f(SID_i, z))$$

where K denotes the cryptographic information, P denotes second polynomial, i denotes a member of the group, SID_i denotes the numbers each associated with a member of the group, and z denotes a random number. A further exemplary embodiment includes communicating among two or more members of the group and utilizing the cryptographic information to secure the communication. A further exemplary embodiment the includes defining a subset of the group, communicating among the subset, and utilizing the cryptographic information to allow access to the communication only to the subset. A further exemplary embodiment includes conditionally granting access to a resource to one or more members of the group. In a further exemplary embodiment the first polynomial is defined in a finite field which is formed from a prime number. In a further exemplary embodiment the resource is one of a broadcast of information and a stream of digital information. A further exemplary embodiment includes adding a member

to the group. A further exemplary embodiment includes storing computing or storing a third polynomial which is a function of a new group including one or more added members. A further exemplary embodiment includes removing a member from the group. In a further exemplary embodiment the removing a member from the group includes computing a third polynomial which is a function of a new group removing one or more members. In a further exemplary embodiment the providing communication between or among two or more members of the group includes providing secure group communication, secure dynamic conferencing, differential access control, and hierarchical access control. In a further exemplary embodiment the communication between or among two or more members includes communication via at least one of a packet switched communication link, a wireless communication link, a WIFI communication link, a WIMAX communication link, a communication link utilizing a IP, TCP, UDP, VOIP or SSL, and a communication link utilizing CDMA, TDMA, or FDMA. In a further exemplary embodiment the removing a member from the group includes determining a fourth polynomial which is a function of the third polynomial.

One exemplary embodiment is a system including at least one computer accessible memory configured to store an access control polynomial which is a function of a set of integers personal to and secret to members of a group, a processor operable to process the access control polynomial and information to be shared with at least one member of the group to generate a public polynomial, and an interface to a communication link operable to output the information to be shared with at least one member of the group to the communication link. In a further exemplary embodiment the computer accessible memory is configured to store instructions for distributing the public polynomial. In a further exemplary embodiment the information to be shared with at least one member of the group information is a key. In a further exemplary

embodiment the computer accessible memory further includes instructions for distributing the key to the group members. In a further exemplary embodiment the computer accessible memory further includes instructions for providing SGC. In a further exemplary embodiment the computer accessible memory further includes instructions for providing SDC. In a further exemplary embodiment the computer accessible memory further includes instructions for providing DIF-AC. In a further exemplary embodiment the computer accessible memory further includes instructions for providing HAC. In a further exemplary embodiment the computer accessible memory further includes instructions for providing SGC, SDC, DIF-AC, and HAC.

While multiple embodiments, forms, objects, features, advantages, aspects, and benefits have been illustrated and described in detail in the drawings and foregoing description, the same are to be considered as illustrative and not restrictive in character, it being understood that only exemplary embodiments have been shown and described and that all changes and modifications that come within the spirit of the inventions shall be protected. It should be understood that while the use of words such as exemplary, preferable, preferably, preferred or more preferred utilized in the description above indicate that the feature so described may be more desirable, it nonetheless may not be necessary and embodiments lacking the same may be contemplated as within the scope of the invention, the scope being defined by the claims that follow. It is intended that words such as "a," "an," "at least one," or "at least one portion" are not limited to only one item unless specifically stated to the contrary. When the language "at least a portion" and/or "a portion" is used the item can include a portion and/or the entire item unless specifically stated to the contrary.

CLAIMS

1. A method comprising:

computing or storing in a computer accessible medium a first polynomial which is a function of a set of numbers each associated with a member of a group to be provided a cryptographic information;

determining a second polynomial which is a function of the first polynomial and an information to be privately shared with the group; and

using at least one of the first polynomial and the second polynomial in providing communication between or among two or more members of the group.

2. A method according to claim 1 wherein the providing communication between or among two or more members of the group includes providing at least one of a trusted collaborative computing environment, a secure dynamic conferencing environment, a differential access control environment, and a hierarchical access control environment.

3. A method according to claim 1 wherein the first polynomial is a function of a public random number.

4. A method according to claim 1 wherein the first polynomial includes a term which is zero when evaluated with one of the set of numbers each associated with a member of a group to be provided a cryptographic key.

5. A method according to claim 1 wherein the first polynomial is described by the formula:

$$A(x) = \prod_{i \in \psi} (x - f(SID_i, z))$$

where $A(x)$ denotes the first polynomial, i denotes a member of the group, ψ denotes the group, SID_i denotes the numbers each associated with a member of the group, and z denotes a random number.

6. A method according to claim 1 wherein the cryptographic information is a cryptographic key.

7. A method according to any of the preceding claims further comprising distributing the second polynomial to the group.

8. A method according to any of the preceding claims further comprising at least one member of the group receiving the second polynomial.

9. A method according to any of the preceding claims further comprising obtaining the cryptographic information from a distributed polynomial.

10. A method according to claim 9 further comprising obtaining the cryptographic information by calculating

$$K = P(f(SID_i, z))$$

where K denotes the cryptographic information, P denotes second polynomial, i denotes a member of the group, SID_i denotes the numbers each associated with a member of the group, and z denotes a random number.

11. A method according to any of the preceding claims further comprising communicating among two or more members of the group and utilizing the cryptographic information to secure the communication.

12. A method according to any of the preceding further comprising defining a subset of the group, communicating among the subset, and utilizing the cryptographic information to allow access to the communication only to the subset.

13. A method according to any of the preceding further comprising conditionally granting access to a resource to one or more members of the group.

14. A method according to claim 1 wherein the first polynomial is defined in a finite field which is formed from a prime number.

15. A method according to claim 13 wherein the resource is one of a broadcast of information and a stream of digital information.

16. A method according to any of the preceding further comprising adding a member to the group.

17. A method according to claim 16 wherein the adding a member to the group includes storing computing or storing a third polynomial which is a function of a new group including one or more added members.

18. A method according to any of the preceding claims further comprising removing a member from the group.

19. A method according to claim 18 wherein the removing a member from the group includes computing a third polynomial which is a function of a new group removing one or more members.

20. A method according to any of the preceding claims wherein the providing communication between or among two or more members of the group includes providing secure group communication, secure dynamic conferencing, differential access control, and hierarchical access control.

21. A method according to any of the preceding claims wherein the communication between or among two or more members includes communication via at least one of a packet switched communication link, a wireless communication link, a WIFI communication link, a WIMAX communication link, a communication link utilizing a IP, TCP, UDP, VOIP or SSL, and a communication link utilizing CDMA, TDMA, or FDMA.

22. A method according to claim 18 wherein the removing a member from the group includes determining a fourth polynomial which is a function of the third polynomial.

23. A system including at least one computer accessible memory configured to store an access control polynomial which is a function of a set of integers personal to and secret to members of a group, a processor operable to process the access control polynomial and information to be shared with at least one member of the group to generate a public polynomial, and an interface to a communication link operable to output the information to be shared with at least one member of the group to the communication link.

24. A system according to claim 23 wherein the computer accessible memory is configured to store instructions for distributing the public polynomial.

25. A system according to claim 23 wherein the information to be shared with at least one member of the group information is a key.

26. A system according to claim 23 wherein the computer accessible memory further includes instructions for distributing the key to the group members.

27. A system according to claim 23 wherein the computer accessible memory further includes instructions for providing SGC.

28. A system according to claim 23 wherein the computer accessible memory further includes instructions for providing SDC.

29. A system according to claim 23 wherein the computer accessible memory further includes instructions for providing DIF-AC.

30. A system according to claim 23 wherein the computer accessible memory further includes instructions for providing HAC.

31. A system according to claim 23 wherein the computer accessible memory further includes instructions for providing SGC, SDC, DIF-AC, and HAC.

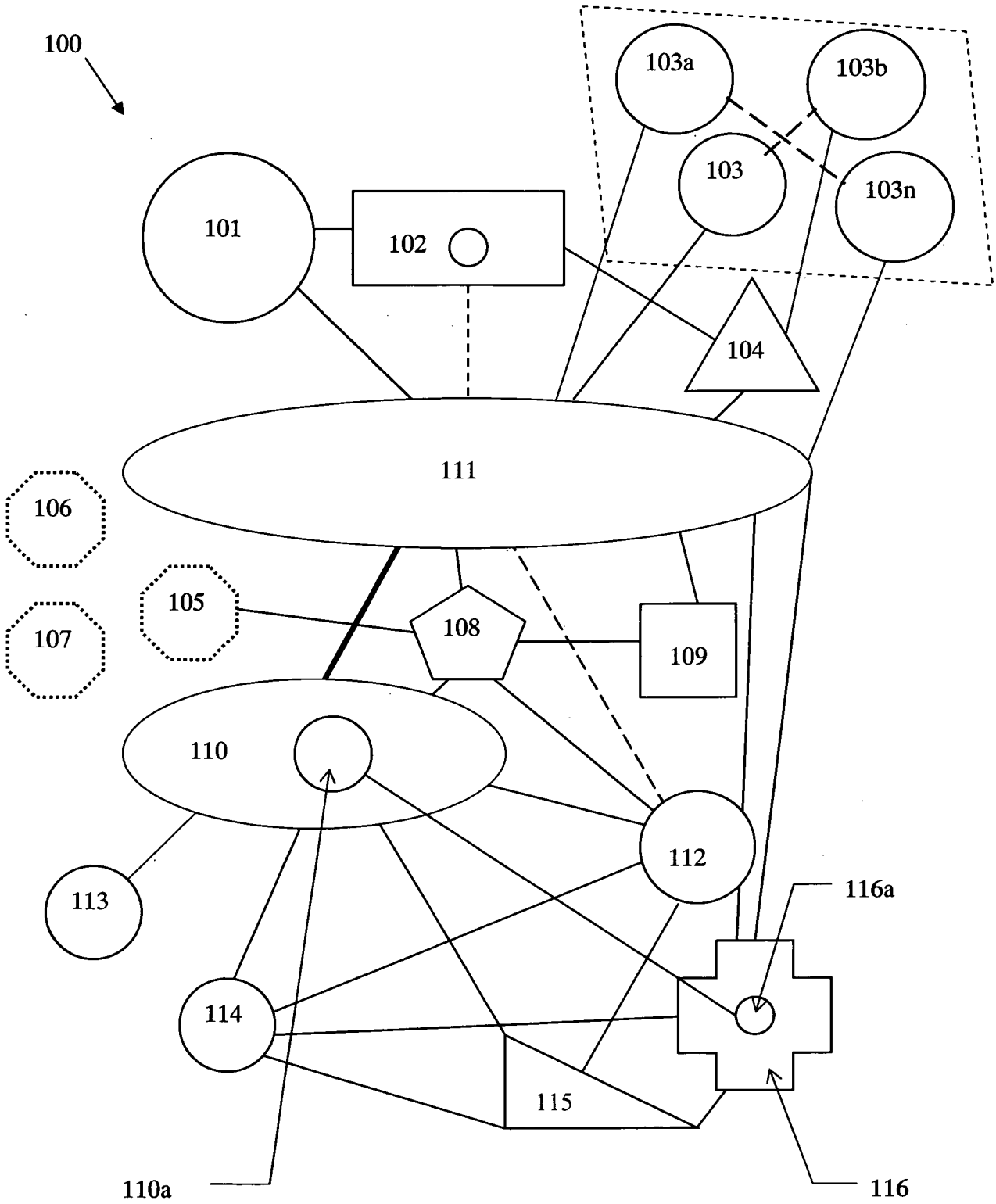
100
103a
103b
103
103n
101
102
104
111
106
105
107
108
109
110
112
113
116a
114
115
110a
116

Fig. 1

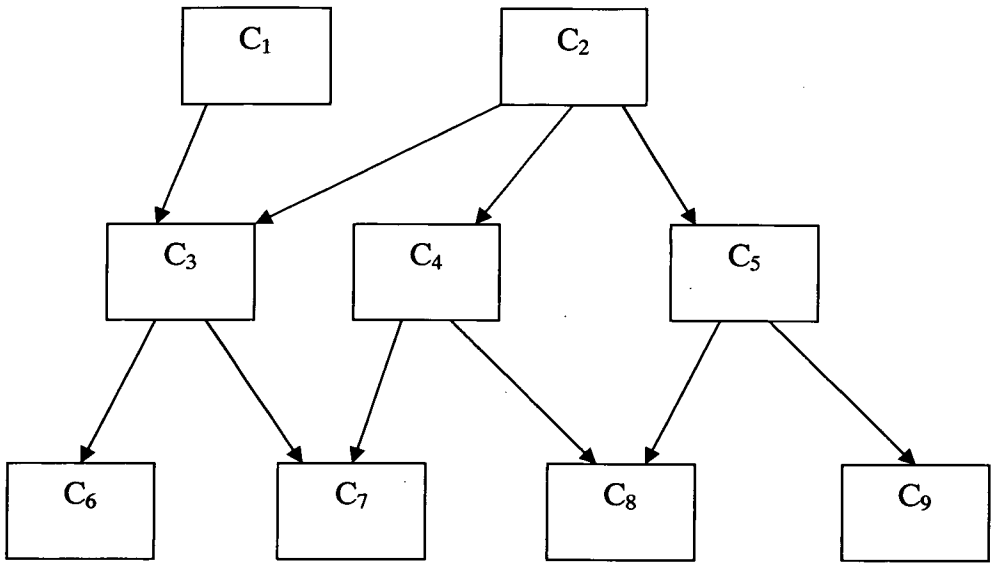

Fig. 2

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 08/06027

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - H04L 9/00 (2008.04)
USPC - 380/264
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)
USPC - 380/264

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
USPC - 380/1, 255, 264; 709/201-203, 217-219, 225 -- text search, see search terms below

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
PubWEST(PGPB,USPT,USOC,EPAB,JPAB); Google Scholar
Search Terms Used: collaborativ, comput, multiuser, multi-user, hierarch, differential, secure, dynamic, conferenc, nod, class, access, control, polynomial, trust, secur, encrypt, cryptograph, key, hierarch, differential, user

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X --- Y	ZOU et al. "Dual-Level Key Management for secure grid communication in dynamic and hierarchical groups." In: Science Direct [online], 11 December 2006, page 1-11 [retrieved on 2008-08-27], Retrieved from the Internet: <URL: http://www.cs.iupui.edu/~xkzou/Papers/FGCS-DGKM.pdf>, especially page 2, right column, ln 48; page 3, left column, ln 10-11, 17-18, 28-40; page 3, right column, ln 55-59, 67, 74-75; page 5, Fig 2; page 5, left column, ln 2-7, 9-11; page 5, right column, ln 78-79; page 7, left col, ln 50-52; page 9, Fig 3; page 9, right column, ln 57-58; and page 10, right column, ln 44-46.	1-7, 14, 23-27, 30 ---------- 28-29, 31
Y	KARANDIKAR et al. "An Effective Key Management Approach to Differential Access Control in Dynamic Environments." In: Journal of Computer Science (2) 6 [online], 2006, page 542-549 [retrieved on 2008-08-29], Retrieved from the Internet: <URL: http://www.scipub.org/fulltext/jcs/jcs26542-549.pdf>, especially page 542, left column, para 1; page 543, left column, para 3; page 543, right column, para 5; and page 544, left column, para 2 -3.	28-29, 31
A	US 2004/0225570 A1 (ALGESHEIMER et al.) 11 November 2004 (11.11.2004), entire document.	1-7, 14, 23-31
A	US 5,202,921 A (HERZBERG et al.) 13 April 1993 (13.04.1993), entire document.	1-7, 14, 23-31
A	US 2004/0196842 A1 (DOBBINS) 07 October 2004 (07.10.2004), entire document.	1-7, 14, 23-31
A	US 2006/0259965 A1 (CHEN) 16 November 2006 (16.11.2006), entire document.	1-7, 14, 23-31

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

* 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
29 August 2008 (29.08.2008)	04 SEP 2008
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201	Authorized officer: Lee W. Young PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

Form PCT/ISA/210 (second sheet) (April 2007)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 08/06027

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☒ Claims Nos.: 8-13 and 15-22
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

☐ No protest accompanied the payment of additional search fees.