Cryptographic master-key-generation scheme and its application to public key distribution

T. Kiesler
L. Harn

Indexing terms: Cryptography, Public key distribution, AKl-Taylor-MacKinnon-Meijer scheme, Master-key problems

Abstract: We present a cryptographic master-key-generation scheme based on a recently developed cryptographic multi-level key generation scheme. That multilevel key generation scheme is a logical complement to the well known Akl-Taylor-MacKinnon-Meijer scheme, and is particularly efficient for hierarchical structures of limited depth, which is the case for the master-key application at hand. The master-key problem is defined in this paper and our solution is presented, and illuminated, by way of comparison with a solution presented recently by Chick and Tavares, based on a variation on the Akl-Taylor-MacKinnon-Meijer scheme. Once this master key scheme is obtained and understood, it becomes a natural step to apply it to solve the public-key-distribution problem. This is achieved in a manner, which, unlike the earlier Diffie/Hellman solution, is extendible to groups of more than two users.

1 Introduction

In this paper is presented a cryptographic master key generation scheme on top of a cryptographic multilevel key-generation scheme recently developed by Harn and Lin [1]. That multilevel key-generation scheme is a logical complement to the well known Akl-Taylor-MacKinnon-Meijer (ATMM) scheme, and is particularly efficient for hierarchical structures of limited depth. In this paper, after presenting a brief review of the multilevel key-generation schemes of Akl *et al.* [2, 3, 4] and of Harn and Lin, we build our master-key-generation scheme on top of that of Harn and Lin and apply this master-key-generation scheme to develop a new public-key-distribution scheme.

2 Cryptographic master-key problem description

The following general definition of master key is used:

A master key for keys $K_1, K_2, \ldots, K_n (n \geqslant 2)$ is a single key K which

(i) accesses all information accessible by any subset of $\{K_1, K_2, \ldots, K_n\}$;

(ii) accesses no information not accessible by some subset of $\{K_1, K_2, \ldots, K_n\}$;

(iii) cannot be determined by any proper subset of $\{K_1, K_2, \ldots, K_n\}$; and

(iv) has length satisfying the property length $(K) \ll \Sigma$ length (K_i), $(1 \leqslant i \leqslant n)$

(ideally K is a single key of the same size as $K_1, \ldots, K_n$).

Clearly, in the ideal case, if one is not to contradict basic theorems of information theory, there must be a functional relationship between K and $\{K_1, K_2, \ldots, K_n\}$, i.e. K must contain within its structure room for the structures of $K_1, \ldots, K_n$, and each of $K_1, \ldots, K_n$ must bear only a portion of the structure in K.

We assume a set of nodes in a directed, acyclic graph G, ordered by partial order $\geqslant$, and associated with each node i of the graph G a key K_i, such that the set K of all keys itself constitutes a directed, acyclic graph, ordered by relation $\supseteq$, where $K_i \supseteq K_j$ means that key K_i accesses all information accessible to key K_j. Then $(G, \geqslant)$ is isomorphic to $(K, \supseteq)$ under the bijection $f: i \rightarrow K_i$. If we refer to nodes i and j of graph G as the classes of users that possess keys K_i and K_j, respectively, then $K_i \supseteq K_j$ means that class i is supervisor to class j, or class j is subordinate to class i. The isomorphism determines an acyclic digraph structure on the key space K. Henceforth we replace the expression 'class of users' by the simpler term 'user'.

We wish to be able to create a master key K_{MS} for any subset S of keys in K (i.e. add a new key, a virtual key, to graph K, and a new node, a virtual node, to graph G). Without loss of generality, we instead create master keys K_{MS} for only each minimal subset S^* of keys in K, where minimal subsets are those having the property that if K_i and K_j belong to set S^*, then K_i is unrelated to K_j (i.e. $K_i \not\supseteq K_j$ and $K_j \not\supseteq K_i$). This means that each element of S^* accesses unique information which no other element of S^* can access. Based on the isomorphism between $(G, \geqslant)$ and $(K, \supseteq)$, the above definition of minimal subsets applies to nodes of G as well as to keys of K.

Example: Let G consist of the graph of seven nodes with keys $K_1, K_2, \ldots, K_7$ shown in Fig. 1. The master key for

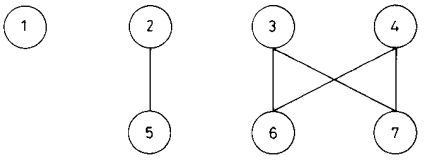

Fig. 1 *Graph of seven nodes with keys $K_1, K_2, \ldots, K_7$*

$S_1^* = \{K_1, K_2\}$ is the same as that for $\{K_1, K_2, K_5\}$, but different from that for $\{K_1, K_5\}$. The master key for $S_2^* = \{K_2, K_3\}$ is the same as that for $\{K_2, K_3, K_5, K_6,$

Paper 8694E (C1, C3), first received 8th April and in revised form 28th October 1991

The authors are with the Computer Science Telecommunication Program, University of Missouri-Kansas City, Kansas City, M0, 64110, USA

Authorized licensed use limited to: University of Missouri System. Downloaded on March 23, 2009 at 15:20 from IEEE Xplore. Restrictions apply.

$K_7\}$, and any set 'in between'. Hence only $\{K_2, K_3\}$ need be considered. There are 34 distinct minimal subsets of G and hence 34 distinct master keys for this key space $\{K_1, K_2, \ldots, K_7\}$.

3 Master-key-generation scheme

We now present a multilevel hierarchical scheme for generating keys for all nodes in a graph G, i.e. key graph K such that $(G, \geqslant)$ is isomorphic to (K, $\supseteq$), and then extend that scheme to generate also master keys for all minimal subsets S^* of K such that the size of the master key is ideal.

Akl *et al.* [2, 3, 4] developed the first such multilevel hierarchical scheme. Their scheme (in its original form, sufficient for current purposes) utilises a key centre which assigns to each node of graph G a prime (say, with node i, the ith prime p_i) and assigns to node i key $K_i = \alpha^{t_i} \bmod N$, where $N = p * q$ for two large primes p and q, and $\alpha \in [2, N-1]$, α relatively prime to N, and $t_i = \Pi p_j$, where the product is over all users $j \in G$ such that $i \not\geqslant j$. The value N and the values t_i (for all users i) are made public, while p, q and α are kept secret.

Recently, Chick and Tavares [5] adapted this scheme of Akl *et al.* to produce a master-key scheme. The primary difference from the original Akl scheme is that, whereas Akl rules out the possibility of all the immediate descendants of a node together calculating the key for that node, Chick and Tavares allow for that possibility, since that possibility is implied in the concept of master key. The authors made a similar adaptation in another context [6].

The Harn and Lin Scheme [1] is as follows. A key centre selects safe primes p and q, calculates $N = p * q$, selects $\alpha \in [2, N-1]$ such that α is relatively prime to N, assigns to node i prime $e_i = i$th odd prime, calculates $d_i = e_i^{-1} \bmod \Phi(N)$, where Φ is the Euler totient function, and assigns to node i key

$$K_i = \alpha^{\Pi d_j} \bmod N$$

where the product is over all j, $i \geqslant j$. The centre keeps p, q and the values d_i (for all users i) secret and makes N and the values e_i (for all users i) public. Note that this definition is complementary to that of Akl *et al.* whereas the Akl *et al.* scheme uses public information in the exponent, and that information relates to all nodes not subordinate to the given node i, this scheme uses secret information in the exponent, and that information relates to all nodes that are subordinate to the given node i (including node i).

Corresponding to each exponent is the corresponding value $t_i = \Pi e_j$, where this product is also over all nodes j such that $i \geqslant j$. In this scheme, it is necessary to publish, for each user i, the value t_i. Access to such public keys is all that is needed for the holder of any key to derive the key of any of its subordinates. Specifically, if $i \geqslant j$, then $K_j = K_i^{\Pi e_k} \bmod N$, where the product is over all k such that $i \geqslant k$ and $j \not\geqslant k$, or equivalently as $K_j = K_i^{t_i/t_j} \bmod N$.

Harn and Lin [1] show that their scheme allows for expansion of the user graph G, requiring changes to keys only at nodes of which the new node is subordinate, unlike the Akl scheme where new nodes cannot be added without requiring the regeneration of a completely new system of keys.

To produce master keys as an extension of this scheme, we follow the following procedure. Corresponding to minimal subset $S^* = \{K_{i_1}, K_{i_2}, \ldots, K_{i_k}\}$ master key K_{MS*} would be calculated by a key centre as

$$K_{MS^*} = \alpha^{\Pi d_j} \bmod N$$

where the product is over all values j such that $t \geqslant j$ for some $t \in \{i_1, i_2, \ldots, i_k\}$. Then any key K_{i_j} can be calculated by

$$K_{i_j} = (K_{MS^*})^{\Pi e k} \bmod N$$

where the product is over all k such that $i_j \not\geqslant k$ but $i_s \geqslant k$ for some $i_s \in \{i_1, i_2, \ldots, i_k\}$, $i_s \neq i_j$. If we express the $\Pi\, e_j$ over all t, $i_j \geqslant t$, as E_{i_j}, and $\Pi\, e_j$ over all t, $i_s \geqslant t$ for some $t \in \{i_1, i_2, \ldots, i_k\}$ as E_{S^*} then key K_{i_j} can be derived from the master key K_{MS^*} by the simple computational formula

$$K_{i_j} = (K_{MS^*})^{E_{S^*}/E_{i_j}} \bmod N$$

The significance of minimal subsets becomes evident. Using minimal subsets allows us to associate each master key to a virtual node of a 2-level virtual graph, which node has as its subordinates in this virtual graph only the set of nodes $\{i_1, i_2, \ldots, i_k\}$, and where the master key $K_{MS^*} = \alpha^{\Pi d_j} \bmod N$, has exponent product over the union of the sets of secret keys found in the exponents of the keys $K_{i_1}, K_{i_2}, \ldots, K_{i_k}$.

We can now replace the service graph used in the Chick and Tavares scheme by the entire user graph of the Akl *et al.* scheme, and, in the terminology of the present paper, present master keys associated with virtual nodes superimposed on the Akl user graph as follows. For minimal subset $S^* = \{K_{i_1}, K_{i_2}, \ldots, K_{i_k}\}$ the corresponding master key is given by

$$K_{MS^*} = \alpha^{gcd(t_{i,1}, t_{i,2}, \ldots, t_{i,k})} \bmod N$$

Defining $t_{S^*} = gcd(t_{i,1}, t_{i,2}, \ldots, t_{i,k})$, any subordinate key then can be calculated as

$$K_{i_j} = (K_{MS^*})^{t_{ij}/t_{S^*}} \bmod N$$

3.1 Security

The security of our master-key scheme is based on the security of the underlying Harn and Lin multilevel key-generation scheme, specifically the fact that the keys p, q and d_i (for all users i) are secret. It is therefore based ultimately on the fact that users are unable to factor N into p and q or, equivalently, to find d_i from e_i for any i.

For any owner of a master key to calculate any other master key or user key to which it should not have access would require that that owner obtain access to at least one secret value d_j. Owners of master keys, like users, can remove secret keys from exponents, using the publicly known values e_i (for all users i), but cannot add secret keys to exponents. What was just stated about an owner of a master key holds just as well for any group of owners of masters keys, who would attempt to act in collusion to calculate a key (master key or user key) to which they should not have access.

4 Application to a group-key-distribution scheme

We now apply our master-key scheme to develop a group-key-distribution scheme. Let us suppose that there will be at most n users registered into the system, and that once the size reaches n no more users can be added. Let us call the users (real and potential) $u_1, u_2, \ldots, u_n$.

We first show how keys for any groups of size two can be generated. This will be comparable to the original public-key-distribution scheme (PKDS) of Diffie and

Authorized licensed use limited to: University of Missouri System. Downloaded on March 23, 2009 at 15:20 from IEEE Xplore. Restrictions apply.

Hellman [7]. Then we will generalise the scheme to cover any groups of size k ($2 \leqslant k \leqslant max$) for some fixed maximum size, max, much smaller than n. This the Diffie-and-Hellman scheme could not do.

4.1 Groups of size 2

For our scheme a key centre is required, and the key centre, rather than individual users, generates all secret and public keys for all users (real or potential) of the system. To understand the method of generation of these keys, we need to conceive of a two-level graph G. We start with level 2.

There are ${}_nC_2$ potential groups of size 2. We represent each group as (q, r) with $q < r$ and $1 \leqslant q, r \leqslant n$, and order these groups by the natural relation $(q, r) < (s, t)$ if and only if $q < s$ or ($q = s$ and $r < t$), i.e. lexicographic order. We let these ordered pairs (q, r) constitute the nodes on level 2 of a graph G, with (1, 2) being node 1 and $(n-1, n)$ being node ${}_nC_2$, and in general (q, r) being node h, where h can be calculated from

$$h = (q-1)(2n-q)/2 + (r-q)$$

The users (real or potential) $u_1, u_2, \ldots, u_n$ of the system will be constituted as level 1, with user $u_i (1 \leqslant i \leqslant n)$ being supervisor to every group (q, r) on level 2, and only those groups, for which $q = i$ or $r = i$.

Given this virtual graph G, the key centre can now generate keys for the system. The key centre selects safe primes p and q, calculates $N = p * q$, and then selects $\alpha \in [2, N-1]$ such that α and N are relatively prime. The centre assigns odd primes e_h, to the nodes h on level 2 of graph G, where, as above e_h is the hth odd prime. It also calculates their inverses d_h modulo $\Phi(N)$. The centre implicitly associates to node h on level 2 key $K_h = \alpha^{d_h} \bmod N$. Then, for user u_i, the corresponding key K_i is defined as $K_i = \alpha^{(\Pi d_h)} \bmod N$, where the product is over all secret keys d_h associated with nodes h on level 2 to which i belongs.

Since each user is supervisor to every group to which he/she potentially can belong, then that user can calculate all needed group keys. For example, user u_i can calculate the key for communication with user u_j as follows: $K_{i,j} = K_i^{E'} \bmod N$, where E' is the product of all public keys corresponding to groups to which i belongs, except group (i, j) or (j, i). Note that, in theory, each user does not need to have a table of all public keys, since anyone can calculate any public key as follows: given group (q, r), calculate the corresponding position h of that group on level 2, using the equation given above, and then calculate the hth odd prime. For simplicity, user u_i could calculate all of these values for the groups (i, k) or (k, i) for all k with whom this user is likely to want to communicate, and store them locally (or of couse the system could calculate them and given them to user u_i). Or user u_i (or the system) could calculate all $(n-1)$ such public keys as well as the product of all of them (call it E), and the user could store these n values (only one of which, E, is long). Then the above calculation of key $K_{i,j}$ could be made much more efficient, since the user would need to fetch from the table the value E and the key corresponding to group (i, j) or (j, i), call it e, and calculate $K_{i,j} = K_i^{E/e'} \bmod N$.

4.2 Groups of size k, $2 \leqslant k \leqslant max$

The scheme just presented generalises very simply to the case where we wish to consider group keys for groups of any size k between 2 and some reasonably sized bound, max. Two possible approaches follow.

4.2.1 Approach 1:

First, generalise the case just presented to any specific group size k, $2 \leqslant k \leqslant max$. Now level 2 of our virtual graph contains ${}_nC_k$ nodes, corresponding to all groups of size k, with these groups ordered by 'lexicographic order', with the group in position h being assigned public key $e_h = h$th odd prime, using as base α_k, and with the rest of the scheme precisely as for the special case $k = 2$ presented above. To find the position h of group $(i_1, i_2, \ldots, i_k)$, we need to use a more complicated algorithm as follows:

Input $\{n, \text{group } (i_1, i_2, \ldots, i_k)\}$
If necessary, sort $i_1, i_2, \ldots, i_k$ such that $i_1 < i_2 < \cdots < i_k$
output $= \sum_{s=1}^{k-1} S\{s, (i_{s-1})', (i_s)'\} + (i'_k - i'_{k-1})$
where $(i_0)' = 0$, and

$$S\{s, (i_{s-1})', (i_s)'\} = \sum_{r=(i_{s-1})'+1}^{(i_s)'-1} {}_{(n-r)}C_{(k-s)}$$

Let the master key given to user i in this scheme be referred to as $K_i^{(k)}$.

The key centre executes this procedure ($max - 1$) times and distributes to each user ($max - 1$) keys: to user i keys $(K_i^{(2)}, K_i^{(3)}, \ldots, K_i^{(max)})$. Then when user i wants to communicate with a group of size k, he/she uses as master key for that group the key $K_i^{(k)}$.

4.2.2 Approach 2:

For each user to use exactly one master key for communication within any group of any size k, $2 \leqslant k \leqslant max$, the number of nodes on level 2 must be $\Sigma\ {}_nC_k$, where the summation is over all k, $2 \leqslant k \leqslant max$. We use the same lexicographic order as we used in approach 1 for all groups of any specific size k, and we then order all groups of size k_1 in front of all groups of size k_2 if $k_1 < k_2$. The rest of the procedure is basically the same, with the one change that the position h of group $(i_1, i_2, \ldots, i_k)$ is obtained by adding to the value obtained from the algorithm above the sum $\Sigma\ {}_nC_j$, where the summation is over all j, $2 \leqslant j \leqslant k-1$.

4.3 Comparison between Diffie and Hellman PKDS and our scheme

There are several significant similarities and differences between our public-key-distribution scheme just presented and that of Diffie and Hellman [7]. They are:

(i) In the Diffie and Hellman scheme, if one thinks of their secret key for each user as a master key which that user can use as needed to calculate any needed secret session key for communication with any other user (using also that other user's public key), then their scheme becomes very similar to our scheme just presented.

(ii) In the Diffie and Hellman scheme, the secret key (master key) is chosen by the user and can be very short. However, the public key, also determined by the user, is very long. In our scheme, public keys and secret keys are determined by a key centre. The public key for each group to which a user belongs is very short while the individual user's secret key (master key) is very long. Hence, in our scheme, each user needs only maintain a table of small public keys associated with groups to which the user is likely to belong, as well as a single long key. This is in contrast to a table of long public keys required for the Diffie and Hellman scheme.

(iii) The Diffie and Hellman scheme is based on the complexity of the discrete-logarithm problem. Our scheme is based on the complexity of the factorisation problem.

(iv) While the Diffie and Hellman scheme works only for the case of generating and distributing a key for secret

Authorized licensed use limited to: University of Missouri System. Downloaded on March 23, 2009 at 15:20 from IEEE Xplore. Restrictions apply.

communication between two users, our scheme can be used for a group of any size within a system bound.

5 Summary

In this paper we have presented a cryptographic master-key-generation scheme based on a cryptographic-multilevel-key-generation scheme recently developed by Harn and Lin which is particularly efficient for hierarchical structures of limited depth, which is the case for the master-key and public-key distribution applications.

In the Harn and Lin key-generation scheme the nodes stand for real entities, and each entity has a single cryptographic key which controls access to all information associated with that entity, and also can derive the cryptographic keys of all entities that are its descendants in the graph hierarchy. The scheme prevents collusion of any number of descendants to obtain the key of a common ancestor in the graph hierarchy. In this regard, the Harn and Lin scheme is equivalent to the Akl *et al.* scheme.

We then superimposed on this real graph virtual nodes (entities) which possess master keys which can derive the keys of all of their immediate descendants in the real graph. These virtual nodes which possess master keys cannot be thought of as real nodes in an extension of the original graph, since, unlike the real nodes in that graph, no primes were assigned to these virtual nodes, and hence only primes associated with nodes that are descendants of these virtual nodes are included in the definition of their master keys. In other words, the master keys contain exactly the same amount of information as does the set of keys of the minimal set of nodes that the virtual node associated with the master key immediately dominates. On the contrary, a key for a real node in the graph contains its own unique information as well as that of all of the keys associated with its descendants.

We then treated the public-key-distribution problem as an application of this master-key problem, and hence solved it in the same manner. Unlike the earlier Diffie and Hellman solution, this solution is extendible to groups of more than two users up to a fixed (agreed-upon) maximum number of users. In this application, the nodes on the 'upper level' are the real nodes and their immediate descendants on the 'lower level' are the virtual nodes, corresponding to groups of users. Practically speaking, this scheme works best for small values of the size of the graph and of the agreed-upon maximum group size.

6 References

1 HARN, L., and LIN, H.Y.: 'A new cryptographic key generation scheme for multilevel data security', *Computers and Security*, 1990, **9**, (6), pp. 539–546

2 AKL, S.G., and TAYLOR, P.D.: 'Cryptographic solution to a multilevel security problem', *in* 'Advances in cryptology — Proceedings of Crypto '82' (Springer-Verlag, New York, 1982), pp. 237–249

3 MACKINNON, S., and AKL, S.G.: 'New key generation algorithms for multilevel security'. Proceedings of IEEE, symposium on security and privacy, 1983, pp. 72–78

4 MACKINNON, S.J., TAYLOR, P.D., MEIJER, H., and AKL, S.G.: 'An optimal algorithm for assigning cryptographic keys to control access in a hierarchy', *IEEE Trans.*, 1985, **C-34**, pp. 797–802

5 CHICK, G.C., and TAVARES, S.E.: 'Flexible access control with master keys', *in* 'Advances in Cryptology — Proceedings of Crypto '89' (Springer-Verlag, New York, 1989), pp. 316–322

6 HARN, L., and KIESLER, T.: 'Authenticated group key distribution scheme for a large distributed network'. Proceedings of 1989 IEEE Computer Society Symposium on Security and Privacy, 1989, pp. 300–309

7 DIFFIE, W., and HELLMAN, M.: 'New directions in cryptography', *IEEE Trans.*, 1976, **IT-22**, (6), pp. 644–654

Authorized licensed use limited to: University of Missouri System. Downloaded on March 23, 2009 at 15:20 from IEEE Xplore. Restrictions apply.