MASTER KEYS FOR GROUP SHARING *

Dorothy E. DENNING
Computer Sciences Department, Purdue University, W. Lafayette, IN 47907, U.S.A.

Fred B. SCHNEIDER
Computer Sciences Department, Cornell University, Ithaca, NY 14853, U.S.A.

Received 31 March 1980; revised version received 14 October 1980

Cryptography, cryptographic keys, encryption, master keys

1. Introduction

Consider a conventional (single-key) cryptosystem with enciphering function E and deciphering function D. Let S be a set of keys, where the key length is b bits. A *master key* for S is a key MK such that:

(1) E(MK, P) = E(K, P) for any plaintext P and K in S;
(2) D(MK, C) = D(K, C) for any ciphertext C and K in S;
(3) $|MK| << |S|b$, where $|MK|$ is the length of MK in bits, and $|S|$ is the number of keys in S.

The first two requirements state that messages enciphered (deciphered) with any key in the set S must be decipherable (encipherable) with the master key MK. The third requirement states that the space requirements for MK must be substantially less than that of all keys in S; otherwise, MK could be implemented simply as a list of the keys in S. MK, therefore, provides a compact representation of S.

Consider a network of N users. A *group* G is any nonempty subset of the N users. Members of G share a secret *group key* K_G, which allows them to broadcast and receive messages from other members of G, and to access and update files private to G. Users not in G are not allowed access to K_G.

There can be at most $2^N - 1$ nonempty groups in the system. We shall present two methods for deriving group keys and a master key MK for the entire set of $2^N - 1$ group keys such that the space requirements for MK are linear in N. The first method is based on Shamir's threshold scheme, the second on Diffie and Hellman's public-key distribution scheme. We shall also show how both methods can be used to provide master keys for sets of groups that are hierarchically structured.

We assume that each user A has a personal key K_A registered with an Authentication Server (AS) [3]. The AS derives all group keys and transmits them to the users enciphered under their personal keys.

2. Polynomial derived group keys

In this scheme, we assume that for each user A, the AS stores A's personal key K_A and two secret values, X_A and Y_A. However, unlike the personal key, the secret values are known only to the AS and not to A (the reason for this will be explained later). We shall show how all group keys can be derived from the secret values X and Y of the users. Thus, the $2^N - 1$ group keys are generable from a table of only 2N elements. This table represents the master key.

The method is based on Shamir's threshold scheme for constructing a key from a set of components [4]. Let $(X_1, Y_1), ..., (X_n, Y_n)$ be the secret values for the

* This research was supported in part by NSF Grant MCS77-04835 at Purdue University and MCS 76-22360 at Cornell University.

users of some group G of size n. Construct the unique polynomial P_G of degree $n - 1$ through the n points in the 2-dimensional plane: $(X_1, Y_1), ..., (X_n, Y_n)$. The group key K_G is the value of the polynomial at 0; that is,

$$K_G = P_G(0).$$

For a group G consisting of a single user A (i.e. G = {A}), n = 1 and the polynomial $P\{A\}$ is a constant function independent of the (X, Y)-coordinates. In this case, we shall assume that $P\{A\}(0) = K_A$; that is, the group key for a single user is the user's personal key. Arithmetic is done modulo a prime number p, where $\log_2(p)$ is not greater than the key length b. The X-coordinates for all users are distinct but randomly drawn from the range $[1, ..., p - 1]$. Thus, each group has a different polynomial, and it is not possible for one group to guess either the polynomial or the key for another group.

In Shamir's application, it is unnecessary for the X-coordinates to be secret, because the individual users are not given the polynomial derived key. Since in our application the users are given the key, both the X- and Y-coordinates must be secret. Furthermore, the pair (X_A, Y_A) associated with user A must not be known even to A. If each user had access to his (X, Y)-coordinates, it would be possible for any $n - 1$ of the members of a group G of size n to reconstruct the polynomial P_G (since the key K_G gives then an n^{th} point). Users could then collaborate and determine the secret (X, Y)-coordinates of other users. For example, suppose users A and B wish to determine the secret values (X_C, Y_C) for user C. If user A requests the key for the group $G\{A,C\}$, he could determine the coefficients a_1 and b_1 of the group polynomial:

$$P\{A,C\} = a_1X + b_1 .$$

Similarly, if user B requests the key for the group $G\{B,C\}$, he could determine the coefficients a_2 and b_2 of the group polynomial:

$$P\{B,C\} = a_2X + b_2 .$$

Since (X_C, Y_C) is a solution to both $P\{A,C\}$ and $P\{B,C\}$, A and B could determine (X_C, Y_C) by solving the system:

$$Y_C - a_1X_C = b_1, \quad Y_C - a_2X_C = b_2 .$$

Similarly, A and B could determine the values (X_D, Y_D) for a user D, and then listen in on a conversation between C and D!

A user A requests a group key K_G from AS by supplying a list of the members of the group:

$$A \rightarrow AS: \text{`}G = \{U_1, U_2, ..., U_n\}\text{'}. \qquad (1)$$

If A belongs to the group (i.e., $A = U_i$ for some i, $1 \leq i \leq n$), AS constructs K_G and returns it to A, enciphered under A's personal key K_A:

$$AS \rightarrow A: E(K_G, K_A) . \qquad (2)$$

3. Exponentially derived group keys

The second method is similar to Diffie and Hellman's public-key distribution scheme [2]. However, it is not a public-key distribution method, because the AS must have access to users' personal secret keys.

Let $K_1, ..., K_n$ be the personal keys of the members of a group G of size n. The group key is:

$$K_G = 2^{K_1K_2 \cdots K_n} \bmod p ,$$

where p is a prime number fixed by AS such that $\log_2(p) \leq b$. When a member A of G requests K_G from the AS, the AS returns K_G, enciphered under the personal key K_A. The master key is represented by the list of personal keys.

Another member of G may be able to determine $2^{K_A} \bmod p$, but he cannot compute K_A without computing a discrete logarithm. Now, if p is only 200 bits long, K_A can be computed in about 2.6 days on a 1 μs per instruction machine [1]. However, if p is 400 bits long (i.e., $b \geq 400$), K_A cannot be practically computed by the fastest known algorithms.

4. Application to hierarchical group structures

Consider a tree structure in which nodes correspond to subsystems or processes. Let the root of the tree correspond to the entire system, and the descendents of a node to its components. These components might cooperate by sharing information, either by accessing a common database or by exchanging messages. Such communication can be made secure by defining a group G that includes only these component subsystems, and enciphering all communications and data files using the group key K_G. In systems of this type, it is often useful to designate some process M_G as the manager of all communication among and within the components of G. Such a process can oversee resource utilization and monitor other aspects of system operation. We desire to permit M_G access to all subgroup keys for subgroups formed from subsets of G, and no others.

Both methods of derived group keys provide attractive methods for providing group managers with master keys. With polynomial derived group keys, each manager M_G for a group G of size n needs only store a list of the n pairs (X_i, Y_i) for each user i in G. With exponentially derived group keys, each manager needs only store a list of n personal keys. Either list represents a master key, from which any of the $2^n - 1$ subgroup keys for G can be generated.

Acknowledgment

We wish to thank P.J. Denning, G.R. Andrews, and an anonymous referee for critically reading this paper and providing numerous comments and suggestions, and T. Berger, G. Birkhoff, and J. Hopcroft for helpful discussions. We are especially grateful to A. Shamir for noting that even if the (X, Y) values for polynomial derived group keys are known only to their associated users, two users can form a coalition to determine other users' values.

References

[1] L. Adleman, A subexponential algorithm for the discrete logarithm problem with applications to cryptography, in: Proc. 20th Symposium on Foundations of Computer Science (1979) 55–60.

[2] W. Diffie and M. Hellman, New directions in cryptography, IEEE Trans. Information Theory 22 (6) (1976) 644–654.

[3] R.M. Needham and M.D. Schroeder, Using encryption for authentication in large networks of computers, Comm. ACM 21 (12) (1978) 993–999.

[4] A. Shamir, How to share a secret, Comm. ACM 22 (11) (1979) 612–613.