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.

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

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 $(X_1, Y_1), \ldots, (X_n, Y_n)$ be the secret values for the

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. Sham for noting that even if the (X, Y) values for polynom derived group keys are known only to their associate 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 cryptograph IEEE Trans. Information Theory 22 (6) (1976) 644–65

[3] R.M. Needham and M.D. Schroeder, Using encryption fo 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.

MORE ON MASTER KEYS FOR GROUP SHARING *

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

Henk MEIJER
Department of Mathematics and Statistics and Department of Computing and Information Science, Queen's University, Kingston, Ontario, Canada

Fred. B. SCHNEIDER
Department of Computer Science, Cornell University, Ithaca, NY 14850, U.S.A.

Received May 1981; revised version received July 1981

Cryptography, cryptographic keys, encryption, master keys

1. Introduction

Two schemes for key distribution in a computer network are presented in [1]. Given a network of N users, these schemes allow the construction of group keys and master keys for each of the $2^N - N - 1$ subgroups of two or more users from only O(N) pieces of secret information. A *group key* allows the members of a group to communicate among themselves; a *master key* allows the holder to compute group keys for all subgroups of the group for which it is the master.

In this paper we show that the first of the two schemes presented in [1] can be compromised. We present a solution to this problem, and then show that our solution leads to a general strategy for constructing schemes that support group keys and master keys.

* Denning is supported in part by NSF Grant MCS 80-15484; Meijer is supported by a scholarship from the Ontario Graduate Scholarship program and Grants A3336 and G0381 from the Natural Sciences and Research Council of Canada; Schneider is supported in part by NSF Grant MCS 76-22360.

2. Polynomial derived group keys

2.1. The original scheme

Associated with each user i is a pair of secret values (x_i, y_i). These values are not known by the user, but are known to holders of master keys for groups of which i is a member. Holders of master key are assumed to be trustworthy.

Given a group, $G = \{i_1, i_2, ..., i_m\}$, of m users, let $f_G(x)$ be the unique (m−1)-st degree Lagrange interpolating polynomial through the points: $(x_{i_1}, y_{i_1}) \cdots (x_{i_m}, y_{i_m})$:

$$f_G(x) = \sum_{h=1}^{m} \left[\prod_{j=1,\, j\neq h}^{m} \frac{x - x_{ij}}{x_{ih} - x_{ij}} \right] y_{ih}.$$

The group key for G, k_G is given by:

$$k_G = f_G(0) = \sum_{h=1}^{m} \left[\prod_{j=1,\, j\neq h}^{m} \frac{-x_{ij}}{x_{ih} - x_{ij}} \right] y_{ih}. \qquad (1)$$

All arithmetic is done in the field GF(p), where p is a fixed prime number known to all users of the network. On request, the holder of a master key for G

0020-0190/81/0000–0000/$02.75