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 19?1; 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.

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 keys 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: $(x_{i_1}, y_{i_1}) \cdots (x_{i_m}, y_{i_m})$:

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

The group key for G, k_G is given by:

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

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

* 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.

sends the group key k_G to any member of G, using some cryptographic system for security.

2.2. The problem

The security of the above scheme depends on the 2N secret values, $x_1, y_1, ..., x_N, y_N$. Unfortunately, if $N \geqslant 4$ a conspiracy of two (or more) users can obtain enough information to compute keys for groups to which they do not belong. This is done as follows. The two users are individually or together members of a total of $3(2^{N-2}-1)+1$ groups. Thus, they can acquire $3(2^{N-2}-1)+1$ group keys and form 2^N-N-1 equations of the form (1) in the unknowns $x_1, y_1, ..., x_N, y_N$ and the remaining $2^{N-2}-(N-2)-1$ keys. These equations are linearly dependent for $N \geqslant 4$. This dependency can be used to compute some of the unknown keys. For example, users 1 and 2 can collaborate and acquire the following keys for the subgroups of $\{1, 2, 3, 4\}$: $k_{12}, k_{13}, k_{14}, k_{23}, k_{24}, k_{123}, k_{124}$. They can then compute the key k_{34} private to group $\{3, 4\}$ by

$$k_{34} = \frac{k_{14}(k_{123}-k_{23})(k_{12}-k_{24}) - k_{13}(k_{124}-k_{24})(k_{12}-k_{23})}{(k_{123}-k_{12})(k_{124}-k_{24}) - (k_{124}-k_{12})(k_{123}-k_{23})} .$$

(This equation does not hold in the (very unlikely) case that the denominator is 0.) Thus, two users can collaborate to derive the group key for almost any subgroup of size two. It is very likely that the same technique would also allow group keys for larger subgroups to be compromised.

2.3. The solution

Let E be a one-way function. Clearly, such a function must be non-linear. Thus, by distributing $k_G = E(f_G(0))$ (in some suitably encrypted form) as the group key for G, the equations are no longer linearly dependent. Hence, compromise by this method is no longer possible.

3. The general scheme

A scheme for generating group keys that supports master keys can be constructed as follows. First, a technique is devised to generate an exponential number of *key-seed values* using only a linear number of *user values*. Moreover, there should be no way to synthesize the key-seed value for a group G without possession of the user values for all users in G. The technique discussed above generates key-seed values by using interpolating polynomials; the second scheme described in [1] generates key-seed values by computing a product of user values. Next, a one-way function is devised to convert these key-seed values in group keys. This destroys any linear dependency among the keys that might arise by virtue of the generating procedure. The second scheme described in [1] was not vulnerable to our method of compromise because the key-seed values were passed through a one-way function (exponentiation over a finite field).

Acknowledgements

We are grateful to Selim G. Akl for his encouragement and useful suggestions and to John Hopcroft for helpful discussions.

References

[1] D.E. Denning and F.B. Schneider, Master keys for group sharing, Information Processing Lett. 12 (1) (1981) 23–25.