AN OPTIMAL CLASS OF SYMMETRIC KEY GENERATION SYSTEMS

Rolf Blom
Ericsson Radio Systems AB
S-163 80 Stockholm, Sweden

Abstract.

It is sometimes required that user pairs in a network share secret information to be used for mutual identification or as a key in a cipher system. If the network is large it becomes impractical or even impossible to store all keys securely at the users. A natural solution then is to supply each user with a relatively small amount of secret data from which he can derive all his keys. A scheme for this purpose will be presented and we call such a scheme a symmetric key generation system (SKGS). However, as all keys will be generated from a small amount of data, dependencies between keys will exist. Therefore by cooperation, users in the system might be able to decrease their uncertainty about keys they should not have access to.

The objective of this paper is to present a class of SKGS for which the amount of secret information needed by each user to generate his keys is the least possible while at the same time a certain minimum number of users have to cooperate to resolve the uncertainty of unknown keys.

T. Beth, N. Cot, and I. Ingemarsson (Eds.): Advances in Cryptology - EUROCRYPT '84, LNCS 209, pp. 335-338, 1985.

Introduction

We picture an application in which messages in a network are protected by a symmetric cipher. Each user pair should have a unique key which enables them to encipher messages to be exchanged and thereby get protection against information disclosure to other users and possible wiretappers. The keys shared between users are distributed at start up time by what we will call a key generation authority. The keys could be seen as master keys used to generate session keys.

A network with n users implies that each user must have access to n-1 keys. Now, if n is large it becomes impractical or even impossible to store all keys securely. A natural solution would then be to supply each user with a relatively small amount of secret data from which he can derive all his keys. A scheme for this purpose will be called a symmetric key generation system (SKGS). However, if all keys are generated from a small amount of data attention must be payed to the fact that dependencies between keys will exist. These dependencies will be such that a group of cooperating users might be able to decrease their uncertainty about keys they should not know about.

In this paper we will present a class of SKGS for which the amount of secret data, needed by each user, is as small as possible while at the same time a certain minimum number k of users have to cooperate to determine keys which are used by other user pairs. The presentation will be made without proofs. For a more detailed analysis and proofs the interested reader is refered to (1).

Preliminaries.

Let G denote the generator matrix of a (n,k) linear code over GF(q). Here n denotes the length of the codewords and k is the dimension of the code, i.e. G is a kxn matrix with elements in FG(q), q a prime power. The number of codewords is q^k and the set of codewords consists of all linear combinations of the rows of G. If $d \in GF(q)^k$ denotes a vector of k information symbols, they will be encoded into c=dG.

A MDS code is usually defined by the condition that the minimum distance of the code is n-k+1. This condition can be shown to be equivalent with the condition that every k columns in G are linearly independent. For a general introduction to MDS codes see (2). From the property that every set of k columns in the generator matrix of a MDS code is independent it follows that a codeword is uniquely determined by any k elements in the codeword. It also follows that knowledge of less than k elements of a codeword reveals no information about another element.

MDS codes exist when $n<q+2$. In the application we consider, q should be much larger than n, so suitable codes will always exist.

A class of SKGS based on MDS codes.

Let the users in the system be numbered consequtivly from 1 to n. Also assume that at least k users shall have to cooperate to get any information about a key they should not have access to. We also assume that the keys should be in GF(q).

The construction of the SKGS starts with selecting a (n,k) MDS code over GF(q) with generator matrix G. This G will be known by all users in the network. Then the key generating authority draws a random symmetric matrix D, also with elements in GF(q). The keys to be used by the user pairs are then given by

$$K = (DG)^T G.$$

User pair (i, j) will use $(K)_{i,j}$ i.e. the element in row i and column j in K. Obviously K is symmetric and hence $(K)_{i,j} = (K)_{j,i}$. Then if user i knows row i of K and user j knows row j of K, they have a common key.

The i:th row in K is given by the i:th row in $(DG)^T$ and G. But G is assumed publicly known so the only data that the key generation authority has to distribute to user i is the i:th row of $(DG)^T$.

$(DG)^T$ is a nxk matrix which means that each row consists of k elements in GF(q). Thus the required secret store at each user is kxlb(q) bits. This is really the least possible value for a SKGS with the assumed parameters (see (1)).

At last we will give a simple explanation of why at least k users have to cooperate to get any information about keys they do not have. Assume $m < k$ users cooperate. Then they know m rows of K. But K is symmetric and then they also know m columns. This means that they know m elements in each row of K for all other users. They do not know any other elements in these rows. Now observe that each row in K is a codeword in the code generated by G. Then from what was stated in the Preliminaries, knowledge of less than k elements in a codeword does not reveal any information about any other element in the codeword. So if less than k users cooperate they get no information about an unknown key. However, if k or more users cooperate they know all of K, because knowledge of k elements in a codeword uniquely determines the codeword.

Implementation aspects.

From a theoretical point of view it is a straightforward task to implement a SKGS based on MDS codes. All that is needed is a generator matrix and computational capability for matrix multiplication. However, rather much storage space is required to store a general generator matrix. To decrease this amount of storage space one could use a punctured Reed-Solomon code because the elements in the generator matrix are given by a simple expresseion, viz.

$$(G)_{i,j}=\alpha^{(i-1)(j-1)}$$

where α is a primitive element in GF(q). Then if c = dG the j:th element in c will be given by

$$c_j = \sum_{i=1}^{k} d_i \, \alpha^{(i-1)(j-1)}$$

This technique is easily applied on the class of SKGS described in the previous section and it shows that a simple and practical implementation exists.

References

(1) R. Blom, "An optimal class of symmetric key generation systems", Report LiTH-ISY-I-0641, Linköping University, 1984.

(2) F.J. MacWilliams and N.J.A. Sloane, The Theory of error correcting codes, North-Holland, New York, 1977.