

Final report on consultancy for the DCCM project: Efficient variants of the Fiat-Naor key predistribution scheme

Yvo Desmedt
Consultant
3780 East Millers Bridge Road
Tallahassee FL 32312

December, 1999

Abstract

Several conference key distribution schemes have been presented. We focus on a non-interactive solution. The Fiat-Naor key predistribution scheme, allows any subset of n users to compute a conference key without any interaction. The scheme is unconditionally secure against a conspiracy of at most w users pooling their secrets together and the size of the secret key is optimal. However, to compute a conference key a user may need to add exponentially many secrets, making the scheme not too practical. In the variants we present there is often no need for any computation to find the conference key. These keys are truly predistributed and our schemes do not require more memory nor more randomness than the Fiat-Naor scheme.

1 Introduction

The problem of key distribution is one that has received a lot of attention by the cryptographic and computer security communities. The Diffie-Hellman scheme allows 2 users to exchange a secret key [7]. The Needham-Schroeder scheme [15] and Kerberos [8] are other such key distribution schemes among many others. Generalizing this problem to more than 2 users was first addressed in 1982 by Ingemarsson-Tang-Wong [11]. Another example is the Burmester-Desmedt scheme [5]. Such schemes are often called conference key distribution schemes. All schemes mentioned so far require interaction.

In certain applications one wants to avoid interaction. Examples of such scenarios are:

when transmission delay is large, an interactive conference key scheme will slow down the start of the communication, which may be urgent.

when the number of parties in the conference is very large, the resources needed to set up the communication can be too large to be practical,

broadcast encryption, in which one party uses a broadcast station to send messages to a set of people. In some circumstances the receiving parties may only have a receiver to receive the broadcasted information, but not be able to transmit themselves, so that an interactive scheme is impossible.

Some of these scenarios have military as well as civilian applications. Indeed, a civilian example of broadcast encryption is pay-per-view TV and a military example is the radio transmission to troops that must keep a radio silence.

The problem to *avoid interaction* in key distribution was first studied by Blom at Eurocrypt 84 [2]. In this scheme each of the n users receives a secret key. From his/her secret key user i can compute a common key with any other user j . In other words any 2 users can compute a common key without interaction. The scheme is secure against a conspiracy of at most w (other) users. The more general problem has been discussed in numerous papers. At Crypto '92 Blundo *et al.* generalized Blom's scheme to allow any t users (out of the n users) to compute a common key without interaction [4]. They also showed that the size of the secret key of each user is optimal. At Crypto '93 Fiat and Naor presented a scheme that allows *any subset* of n users to compute a conference key which is secure against a conspiracy of at most w users [10]. At Eurocrypt '94 Blundo-Cresti proved the optimality of the space complexity of the Fiat-Naor scheme, *i.e.*, that the size of the secret key of each user is optimal in the Fiat-Naor scheme. All schemes discussed in this paragraph have unconditional security.

A problem that has not been studied is the time complexity of conference key distribution with unconditional security. When we survey the Fiat-Naor scheme (see Sections 3.2–3.3) we will see that the computation involved in computing a conference key can be rather extensive. We will see that in some cases the computation of a conference key requires a time exponential in the number of participants. In many applications we cannot afford waiting a long time to compute a conference key. Indeed emergency cyber meetings should not take too long to set up. Another example is the computation of a common key in pay television. Viewers often decide just before the start of a program that they want to watch a pay TV program [9].

From now on we will call non-interactive conference key distribution schemes: conference key predistribution schemes, a terminology first introduced by Matsumoto-Imai [14]. We should observe that Fiat and Naor and Blundo-Cresti called their schemes broadcast encryption schemes, but it is straightforward to see that those actually correspond to conference key predistribution schemes (for a deeper discussion see [13]).

In this report we proceed as follows. In Section 2 we first discuss the notation we use in this text and survey the definition of conference key predistribution scheme. In Section 3 we survey the Fiat-Naor key predistribution scheme. In Section 4 we discuss a variant of the Fiat-Naor scheme which requires no computation, *i.e.*, is just a table look-up method, when the size of the conference is large. In Section 5 we discuss the case that the conference is small. We end with discussing practical remarks and how to use the scheme in a dynamic setting in Section 6.

2 Notation

2.1 Notation and informal definition

When we choose an element $a \in A$ uniformly random and independent (of other events) we just write: $a \in_R A$. When we write $A \subset B$ we mean that A is a subset of B and do not exclude that A could be the empty set or that $A = B$.

We assume that there is a Trusted Initial Key Distributor and that the conference will consist of a subset of n parties denoted by $\mathcal{U} = \{1, 2, \dots, n\}$. (Note that it is easy to combine the action of several parties so that no single Trusted Initial Key Distributor is needed, by using secure multiparty computation (see *e.g.*, [1, 6])). The Trusted Initial Key Distributor generates secrets and privately gives the secret u_i to user i . This initial distribution is done in a preliminary phase, *i.e.*, well before a conference key is needed. This secret information will enable parties to set up a conference key.

Informally we say that a scheme is a $(\leq n, \leq w)$ -Key Predistribution Scheme if:

1. in any conference $P \subset \mathcal{U}$ any user i in P can compute from its secret u_i the common conference key k_P .
2. any set of conspirators F such that $P \cap F = \emptyset$ and $|F \cap \mathcal{U}| \leq w$ has no knowledge of the conference key k_P .

Observe that the condition $P \cap F = \emptyset$ makes sense. Indeed, if a party $j \in P \cap F$, it can compute k_P and leak it to the other parties in F . We call the set of outsiders who are conspirators: $O = F \setminus \mathcal{U}$.

2.2 Formal definition

We now give a formal definition of a $(\leq n, \leq w)$ -Key Predistribution Scheme. This can be skipped in a first reading.

For $1 \leq i \leq n$, let U_i denote the set of all possible secret values u_i . For any set of parties X , let U_X denote the Cartesian product $U_{i_1} \times U_{i_2} \times \dots \times U_{i_j}$, where $X \cap \mathcal{U} = \{i_1, i_2, \dots, i_j\}$ and $1 \leq i_1 < i_2 < \dots < i_j \leq n$. We assume that there is a probability distribution on $U_{\mathcal{U}}$, and that the Trusted Initial Key Distributor chooses $u_{\mathcal{U}} \in U_{\mathcal{U}}$ according to this probability distribution.

Let K_P denote the set of all possible conference keys associated with P . We assume that $K_P = K$ for each $P \in \mathcal{P}$.

Definition 1 We say that the scheme is a $(\leq n, \leq w)$ -Key Predistribution Scheme if the following conditions are satisfied:

1. (Completeness) $\forall P \subset \mathcal{U}, \forall i \in P, \forall u_i \in U_i, \exists k_P \in K_P$:

$$\Pr(K_P = k_P \mid U_i = u_i) = 1.$$

2. (Security) $\forall P \subset \mathcal{U}, \forall k_P \in K_P, \forall F (|F \cap \mathcal{U}| \leq w \text{ and } P \cap F = \emptyset), \forall u_F \in U_F (\Pr(U_F = u_F) > 0)$:

$$\Pr(K_P = k_P \mid U_F = u_F) = \Pr(K_P = k_P).$$

3 The Fiat-Naor scheme

3.1 The scheme

Fiat and Naor presented the following $(\leq n, \leq w)$ -Key Predistribution Scheme [10].

Set-up phase

Let q be a (large enough) positive integer. For every subset $F \subset \mathcal{U}$ of cardinality at most w the Trusted Initial Key Distributor chooses $s_F \in_R Z_q$ and gives s_F to every member of $\bar{F} = (\mathcal{U} \setminus F)$ as the secret information.

Computing a conference key

When members of a conference $P \subset U$ want to compute a common conference key k_P each member computes:

$$k_P = \sum_{\substack{F \\ |F| \leq w \\ F \subset \bar{P}}} s_F \pmod{q}, \quad (1)$$

where $\bar{P} = \mathcal{U} \setminus P$. It is rather obvious to see that the scheme satisfies the definition of Key Predistribution Scheme. Indeed, from the set-up phase we know that each party in P knows the values s_F when $F \subset \bar{P}$ and $|F| \leq w$. So, each member can calculate k_P . The security follows from the fact that a set of conspirators F' (where $|F' \cap \mathcal{U}| \leq w$ and $F' \cap P = \emptyset$) does not know $s_{(F' \cap \mathcal{U})} \in_R Z_q$ using the properties of the one-time pad.

Note that the Fiat-Naor scheme works over any finite (Abelian) group. So, an implementation in $GF(q)$ makes the scheme more efficient.

3.2 An example

We now give an example for illustration.

Take $n = 5$ and $w = 1$ then the Trusted Initial Key Distributor chooses $s_\emptyset, s_{\{1\}}, s_{\{2\}}, s_{\{3\}}, s_{\{4\}}, s_{\{5\}} \in_R Z_q$. The secret information of the users is:

$$\begin{aligned} u_1 &= (s_\emptyset, s_{\{2\}}, s_{\{3\}}, s_{\{4\}}, s_{\{5\}}) \\ u_2 &= (s_\emptyset, s_{\{1\}}, s_{\{3\}}, s_{\{4\}}, s_{\{5\}}) \\ u_3 &= (s_\emptyset, s_{\{1\}}, s_{\{2\}}, s_{\{4\}}, s_{\{5\}}) \\ u_4 &= (s_\emptyset, s_{\{1\}}, s_{\{2\}}, s_{\{3\}}, s_{\{5\}}) \\ u_5 &= (s_\emptyset, s_{\{1\}}, s_{\{2\}}, s_{\{3\}}, s_{\{4\}}) \end{aligned}$$

If $P = \{1, 2\}$ then $k_P = s_\emptyset + s_{\{3\}} + s_{\{4\}} + s_{\{5\}} \pmod{q}$.

3.3 Required amount of computation

We now briefly discuss the number of keys s_F that need to be added up when computing a conference key k_P in the Fiat-Naor scheme. It is obvious that this corresponds to:

$$|\{F \mid F \subset \bar{P} \text{ and } |F| \leq w\}| = \sum_{i=0}^{\min(w, n-|P|)} \binom{n-|P|}{i}. \quad (2)$$

This is maximal when $|P|$ is minimal, *i.e.*, $|P| = 2$. So, using Fiat-Naor as a two party scheme is very inefficient from a computational viewpoint.

Observe that the number of keys s_F that need to be stored by each party is $\sum_{i=0}^w \binom{n-1}{i}$ and that the Trusted Initial Key Distributor chooses $\sum_{i=0}^w \binom{n}{i}$ independent keys. Both values were proven to be optimal by Blundo-Cresti [3]. Since the computation time of the Fiat-Naor scheme is in the worst case proportional to the storage requirement (for a fixed key size) it seems that the computational complexity of the scheme is no problem. However, if a key is 128 bits long, and one stores up to 4 Gbytes of keys on the disk, one can roughly store 40 million keys. However, seeking those from the disk will be very slow! So, in practical situations it is important to reduce the number of keys s_F to seek when computing k_P . We now focus on how to address this problem.

4 A variant of Fiat-Naor optimal for large conferences

4.1 The scheme

It is obvious that a scheme is optimal from a computational viewpoint when no computation is required, *i.e.*, when only one table look-up is required. When $P = \mathcal{U}$ the Fiat-Naor scheme only requires one table look-up. Indeed, $k_{\mathcal{U}} = s_{\emptyset}$ as can easily be seen from (1). *Observe that no other conference P satisfies this property in the Fiat-Naor scheme, as follows easily from (2).*

We now present a variant of the Fiat-Naor scheme. **Set-up phase**

Let q be a (large enough) positive integer. The Trusted Initial Key Distributor:

Step 1 chooses $k_P \in_R Z_q$ for each P of cardinality at least $n - w$.

Step 2 privately gives to user i the conference key k_P for each P in which $i \in P$ and $|P| \geq n - w$.

Computing a conference key

When $|P| \geq n - w$, each party in P uses k_P from memory, else each party computes:

$$k_P = \sum_{\substack{F \\ P \subset F \subset \mathcal{U} \\ |F|=n-w}} k_F \pmod{q}. \quad (3)$$

We now prove that the new scheme is a $(\leq n, \leq w)$ -Key Predistribution Scheme.

Theorem 1 *The scheme in this section is a $(\leq n, \leq w)$ -Key Predistribution Scheme.*

Proof. We first prove completeness. When $|P| \geq n - w$ it is obvious that the parties in P can compute k_P . We now focus on the case that $|P| < n - w$. In this case k_P is the sum of k_F , where $P \subset F$ and $|F| = n - w$. So, when $i \in P$, we have that $i \in F$ and since $|F| = n - w$, party i knows k_F .

We now prove the security of the scheme. Let F' be a set of collaborators, where $|F' \cap \mathcal{U}| \leq w$ and $F' \cap P = \emptyset$. When $|P| \geq n - w$, then k_P is only known to the parties in P , so the security follows trivially.

We now consider the security for the case that $|P| < n - w$. Since $|F' \cap \mathcal{U}| \leq w$ and $|P| < n - w$, we have that there exists a set F such that $|F| = n - w$ and $P \subset F \subset \mathcal{U}$ and $F' \cap F = \emptyset$. Indeed take F a subset of $\mathcal{U} \setminus F'$ where $|\mathcal{U} \setminus F'| \geq n - w$ containing P . Now k_F is not known to any user in F' (and any party outside \mathcal{U}) and is known to all parties in P . The formal proof of security follows from the independent and uniformly random choices by the Trusted Initial Key Distributor and the property of the one-time pad. \square

It is obvious to see that the scheme generalizes over any finite (Abelian) group.

4.2 An example

Take $n = 5$ and $w = 1$. The Trusted Initial Key Distributor chooses $k_{\{1,2,3,4,5\}}, k_{\{2,3,4,5\}}, k_{\{1,3,4,5\}}, k_{\{1,2,4,5\}}, k_{\{1,2,3,5\}}, k_{\{1,2,3,4\}} \in_R Z_q$. The secret information of the users is:

$$\begin{aligned} u_1 &= (k_{\{1,2,3,4,5\}}, k_{\{1,3,4,5\}}, k_{\{1,2,4,5\}}, k_{\{1,2,3,5\}}, k_{\{1,2,3,4\}}) \\ u_2 &= (k_{\{1,2,3,4,5\}}, k_{\{2,3,4,5\}}, k_{\{1,2,4,5\}}, k_{\{1,2,3,5\}}, k_{\{1,2,3,4\}}) \\ u_3 &= (k_{\{1,2,3,4,5\}}, k_{\{2,3,4,5\}}, k_{\{1,3,4,5\}}, k_{\{1,2,3,5\}}, k_{\{1,2,3,4\}}) \\ u_4 &= (k_{\{1,2,3,4,5\}}, k_{\{2,3,4,5\}}, k_{\{1,3,4,5\}}, k_{\{1,2,4,5\}}, k_{\{1,2,3,4\}}) \\ u_5 &= (k_{\{1,2,3,4,5\}}, k_{\{2,3,4,5\}}, k_{\{1,3,4,5\}}, k_{\{1,2,4,5\}}, k_{\{1,2,3,5\}}) \end{aligned}$$

If $P = \{1, 2\}$ then $k_P = k_{\{1,2,4,5\}} + k_{\{1,2,3,5\}} + k_{\{1,2,3,4\}} \pmod q$.

4.3 Required amount of computation

From a computation complexity viewpoint, it is trivial to see that this scheme is optimal for conferences P where $|P| \geq n - w$. When $|P| < n - w$ the number of keys that must be added is:

$$|\{F \mid P \subset F \subset \mathcal{U} \text{ and } |F| = n - w\}| = \binom{n - |P|}{n - w - |P|} \quad (4)$$

Comparing (2) and (4) we immediately see that the new scheme always requires less additions, provided $w > 0$. Indeed, since $|P| < n - w$ we have that $w < n - |P|$, so $w = \min(w, n - |P|)$. So,

$$\binom{n - |P|}{n - w - |P|} = \binom{n - |P|}{w} < \sum_{i=0}^{\min(w, n - |P|)} \binom{n - |P|}{i}.$$

The larger w the larger the saving.

It seems that when one requires that the computational complexity corresponds to a table look-up for all P for which $|P| \geq n - w$ that the computational complexity of our scheme is then optimal for all P . Since other schemes may exist that are based on simpler operations than addition, such a claim seems hard to prove formally. We leave this as an open problem. We now address the complementary question.

5 A variant of Fiat-Naor optimal for small conferences

5.1 A generalization

Let us discuss a generalization of Section 4. In the scheme of Section 4 we only needed a table look-up to find the key of P in $\mathcal{P} = \{P \mid P \subset \mathcal{U} \text{ and } |P| \geq n - w\}$. One can wonder whether there exists other choices for \mathcal{P} that satisfy this property without the need for the Trusted Initial Key Distributor to use more randomness than required or for the parties to store more keys than required (see [3]).

We affirm that such other \mathcal{P} exists. We discuss the case that $w = 1$ and $\mathcal{P} = \{P \mid |P| = 2 \text{ or } |P| = n\}$.

5.2 A scheme for $w = 1$

We first argue that when $n > 3$ the keys that will be distributed by the Trusted Initial Key Distributor must be dependent. Indeed, when $n > 3$ we require a table look-up for more sets than the Trusted Initial Key Distributor will choose independent keys, as follows from the fact that $|\{P \mid |P| = 2 \text{ or } |P| = n\}| = 1 + \binom{n}{2} > 1 + n$ when $n > 3$. We now explain the scheme.

Set-up phase

Let q be a (large enough) positive integer such that $\gcd(q, n - 2) = 1$. The Trusted Initial Key Distributor:

Step 1 chooses $s_F \in_R Z_q$ for each subset $F \subset \mathcal{U}$ of cardinality at most 1.

Step 2 computes k_P for each $P \subset \mathcal{U}$ of cardinality 2 using the Fiat-Naor scheme.

Step 3 privately gives to party i the keys $k_{\mathcal{U}} = s_{\emptyset}$ and k_P for each $P \subset \mathcal{U}$ of cardinality 2 for which $i \in P$, and does this for each party $i \in \mathcal{U}$.

Computing a conference key

When $|P| = 2$ or $|P| = n$, each party i in P uses k_P from memory, else each party $i \in P$ computes $s_F \in_R Z_q$ for each subset $F \subset \mathcal{U}$ of cardinality 1, except when $F = \{i\}$, and uses the Fiat-Naor scheme to compute the key k_P using $s_{\emptyset} = k_{\mathcal{U}}$.

Theorem 2 *The scheme in this section is a $(\leq n, \leq 1)$ -Key Predistribution Scheme.*

Proof. It is sufficient to prove that there is a bijection (one-to-one correspondence) between the keys party i receives and the keys s_F it would have received in the Fiat-Naor scheme.

As will be obvious soon, it is sufficient to consider the case $i = 1$. The following relationship between the keys k_F party 1 received from the Trusted Initial Key Distributor and the keys $s_{\bar{j}}$ ($j \neq 1$) it would have received in the Fiat-Naor scheme follows from (1),

giving:

$$\begin{pmatrix} k_{\mathcal{U}} \\ k_{\{1,2\}} \\ k_{\{1,3\}} \\ k_{\{1,4\}} \\ \vdots \\ k_{\{1,n-2\}} \\ k_{\{1,n-1\}} \\ k_{\{1,n\}} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & \dots & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & \dots & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & \dots & 1 & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 1 & 1 & 1 & 1 & \dots & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & \dots & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & \dots & 1 & 1 & 0 \end{pmatrix} * \begin{pmatrix} s_{\emptyset} \\ s_{\{2\}} \\ s_{\{3\}} \\ s_{\{4\}} \\ \vdots \\ s_{\{n-2\}} \\ s_{\{n-1\}} \\ s_{\{n\}} \end{pmatrix}. \quad (5)$$

The square matrix in (5) defines a bijection if and only if its determinant is invertible over Z_q [12]. To check this, it is obviously sufficient to consider the determinant of the square matrix stripped of its first row and first column, giving:

$$A = \begin{vmatrix} 0 & 1 & 1 & \dots & 1 & 1 & 1 \\ 1 & 0 & 1 & \dots & 1 & 1 & 1 \\ 1 & 1 & 0 & \dots & 1 & 1 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 1 & 1 & 1 & \dots & 0 & 1 & 1 \\ 1 & 1 & 1 & \dots & 1 & 0 & 1 \\ 1 & 1 & 1 & \dots & 1 & 1 & 0 \end{vmatrix} = \pm \begin{vmatrix} 1 & 1 & 1 & \dots & 1 & 1 & 0 \\ 1 & 1 & 1 & \dots & 1 & 0 & 1 \\ 1 & 1 & 1 & \dots & 0 & 1 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 1 & 1 & 0 & \dots & 1 & 1 & 1 \\ 1 & 0 & 1 & \dots & 1 & 1 & 1 \\ 0 & 1 & 1 & \dots & 1 & 1 & 1 \end{vmatrix} = \begin{vmatrix} 1 & 1 & 1 & \dots & 1 & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & -1 & 1 \\ 0 & 0 & 0 & \dots & -1 & 0 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & -1 & \dots & 0 & 0 & 1 \\ 0 & -1 & 0 & \dots & 0 & 0 & 1 \\ 0 & 1 & 1 & \dots & 1 & 1 & 1 \end{vmatrix}$$

where the last determinant is obtained from the second-to-last determinant by subtracting the first row from all other rows, except from the last one, (the second determinant is obtained from the first by interchanging the rows).

We continue to proceed by first rearranging the rows. In the obtained determinant one then adds row 2 to row 3, then the new row 3 to row 4, etc. This gives:

$$A = \pm \begin{vmatrix} 1 & 1 & 1 & \dots & 1 & 1 & 0 \\ 0 & 1 & 1 & \dots & 1 & 1 & 1 \\ 0 & -1 & 0 & \dots & 0 & 0 & 1 \\ 0 & 0 & -1 & \dots & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -1 & 0 & 1 \\ 0 & 0 & 0 & \dots & 0 & -1 & 1 \end{vmatrix} = \begin{vmatrix} 1 & 1 & 1 & \dots & 1 & 1 & 0 \\ 0 & 1 & 1 & \dots & 1 & 1 & 1 \\ 0 & 0 & 1 & \dots & 1 & 1 & 2 \\ 0 & 0 & 0 & \dots & 1 & 1 & 3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 1 & n-3 \\ 0 & 0 & 0 & \dots & 0 & 0 & n-2 \end{vmatrix}$$

So $A = \pm(n-2)$, which has an inverse in Z_q when $\gcd(q, n-2) = 1$.

The case $i \neq 1$ is identical when rearranging the columns and rows of the matrix in (5) appropriately, as can easily be seen. \square

Note that the scheme extends to $GF(q)$ when $\gcd(q, n-2) = 1$.

5.3 Required amount of computation when $w = 1$

It is obvious that the required amount of computation is minimum when $|P| = 2$ or $|P| = n$. In the other cases the scheme requires more computation than the Fiat-Naor scheme since the parties need to compute s_F .

The advantage of the scheme is clear in circumstances where it will primarily be used for two-party conferences and exceptionally for larger conferences. *Each party then stores almost the same number of keys it would store in the trivial two-party only scheme.* Indeed the number of keys a party must store in our scheme is n , while the trivial two-party only scheme requires each party to store a key to communicate with each other party, *i.e.*, $n - 1$ keys. Moreover, in certain applications $w = 1$ is a sufficiently high security. In the military this corresponds to the security given in two-man control.

5.4 Generalization to $w > 1$

We conjecture that for an appropriate choice of q the scheme generalizes to the case that $\mathcal{P} = \{P \mid 2 \leq |P| \leq w + 1 \text{ or } |P| = n\}$.

6 Practical remarks

We conclude this report with some practical comments.

Although this scheme (as well as the Fiat-Naor scheme) only guarantees unconditional security when it is used at most once, it is easy to make a scheme that can be reused. Indeed, for a party P each member i computes the key k_P as described above and uses k_P as the seed of a pseudo-random generator. The output of the generator is used as the conference key. When freshness is required, the members of the party remember for which parties they have used the generator and how often. When they need to compute a new key for a conference P they will start from k_P and run the generator one more time.

We now discuss dynamic aspects for the following scenarios:

one party is taken over by the enemy. In this case no new keys need to be distributed.

The members of \mathcal{U} avoid setting up new conferences involving this member.

multi parties are taken over by the enemy. As long as the number of parties taken over by the enemy is less or equal than w , there is no need for a new distribution.

If that number is higher than w , the Trusted Initial Key Distributor must give new keys. A proactive [16] solution is trivial to design.

a member leaves an ongoing conference. The members compute without interaction the new key.

a known member is added to the ongoing conference. The members compute without interaction the new key.

a new party joins the set \mathcal{U} of known members, as happens in dynamic coalitions. If this was not foreseen in advance extra keys need to be distributed.

References

- [1] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proceedings of the twentieth annual ACM Symp. Theory of Computing, STOC*, pp. 1–10, May 2–4, 1988.

- [2] R. Blom. An optimal class of symmetric key generation systems. In T. Beth, N. Cot, and I. Ingemarsson, editors, *Advances in Cryptology. Proc. of Eurocrypt 84 (Lecture Notes in Computer Science 209)*, pp. 335–338. Springer-Verlag, Berlin, 1985. Paris, France, April 9–11, 1984.
- [3] C. Blundo and A. Cresti. Space requirements for broadcast encryption. In A. De Santis, editor, *Advances in Cryptology — Eurocrypt '94, Proceedings (Lecture Notes in Computer Science 950)*, pp. 471–486. Springer-Verlag, 1995. Perugia, Italy, May 9–12.
- [4] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung. Perfectly-secure key distribution for dynamic conferences. In E. F. Brickell, editor, *Advances in Cryptology — Crypto '92, Proceedings (Lecture Notes in Computer Science 740)*, pp. 471–486. Springer-Verlag, 1993. Santa Barbara, California, U.S.A., August 16–20.
- [5] M. Burmester and Y. Desmedt. A secure and efficient conference key distribution system. In A. De Santis, editor, *Advances in Cryptology — Eurocrypt '94, Proceedings (Lecture Notes in Computer Science 950)*, pp. 275–286. Springer-Verlag, 1995. Perugia, Italy, May 9–12.
- [6] D. Chaum, C. Crépeau, and I. Damgård. Multiparty unconditionally secure protocols. In *Proceedings of the twentieth annual ACM Symp. Theory of Computing, STOC*, pp. 11–19, May 2–4, 1988.
- [7] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Trans. Inform. Theory*, IT-22(6), pp. 644–654, November 1976.
- [8] J. G. Steiner et al. Kerberos: an authentication server for open network system. In *Proceedings of Usenix Conf. (Winter 88)*, 1988.
- [9] A. Fiat. New work on watermarking, traitor tracing, and related issues. Presented at the Second International Workshop on Practice and Theory in Public Key Cryptography, PKC'99, Kamakura, Japan, March, 1-3, 1999., 1999.
- [10] A. Fiat and M. Naor. Broadcast encryption. In D. R. Stinson, editor, *Advances in Cryptology — Crypto '93, Proceedings (Lecture Notes in Computer Science 773)*, pp. 480–491. Springer-Verlag, 1994. Santa Barbara, California, U.S.A., August 22–26.
- [11] I. Ingemarsson, D. T. Tang, and C. K. Wong. A conference key distribution system. *IEEE Trans. Inform. Theory*, 28(5), pp. 714–720, September 1982.
- [12] N. Jacobson. *Basic Algebra I*. W. H. Freeman and Company, New York, 1985.
- [13] K. Kurosawa, T. Yoshida, Y. Desmedt, and M. Burmester. Some bounds and a construction for secure broadcast encryption. In K. Ohta and D. Pei, editors, *Advances in Cryptology — Asiacrypt '98, Proceedings (Lecture Notes in Computer Science 1514)*, pp. 420–433. Springer-Verlag, October, 18–22 1998. Beijing, China.

- [14] T. Matsumoto and H. Imai. On the key predistribution systems. In C. Pomerance, editor, *Advances in Cryptology, Proc. of Crypto '87 (Lecture Notes in Computer Science 293)*, pp. 185–193. Springer-Verlag, 1988. Santa Barbara, California, U.S.A., August 16–20.
- [15] R. M. Needham and M. D. Schroeder. Using encryption for authentication in large networks of computers. *Commun. ACM*, 21(12), pp. 993–999, December 1978.
- [16] R. Ostrovsky and M. Yung. How to withstand mobile virus attacks. In *Proceedings of the 10-th Annual ACM Symp. on Principles of Distributed Computing*, pp. 51–60, August 19–21, 1991. Montreal, Quebec, Canada.