Multiuser Message Authentication, Application to Verifiable Secret Sharing and Key Management Schemes

Aparna Ram^{1,*}, B.B. Amberker²

¹Dept. of Computer Science and Engg. Siddaganga Institute of Technology, Tumkur, Karnataka, India

²Dept. of Computer Science and Engg. National Institute of Technology, Warangal, Andhra Pradesh, India

*Corresponding Author: raparna27@gmail.com

Copyright ©2013 Horizon Research Publishing All rights reserved.

Abstract Providing authentication for the messages exchanged among a group of users is an important issue in secure group communication. We develop multiuser authentication schemes with perfect protection against colluding malicious users numbering fewer than k , where *all* the n users are allowed to be senders (simultaneously with being receivers). In our scheme each user is required to store secret information of size $2k \log_2 q$ bits, and tags to authenticate messages are of length $k \log_2 q$. We use this to obtain, in the setup in which the participants are allowed to employ previously distributed private keys, a non-interactive verifiable secret sharing scheme for multiple dealers, in which shares reveal no information about the secret, and dealers cannot deal inconsistent shares. We also provide authentication to the group key management schemes proposed by Blundo et al. and Fiat-Naor without incurring extra storage cost.

Keywords Authentication Code, Multiuser Authentication, Message, Verifiable Secret Sharing, Key Management, Threshold

and one cannot cheat others in the group by claiming as a different entity. Authentication code is generated and sent along with the message by the sender and the same is used by the receivers to verify authenticity of the received message. The system comprises of a set of N users $\{u_1, u_2, \dots, u_N\}$ and a trusted KDC. At the outset, the KDC generates and gives keys to all the users in the system securely. It requires any set of users in the system to be able to broadcast a message to all the other users who will individually verify the authenticity of the received message. Users in the system do not trust each other and may collude to construct fraudulent messages and try to perform impersonation attack. The objective of this paper is to develop a system providing perfect protection against such attacks.

To provide protection for such a system, a set of conventional point-to-point authentication system can be used, in which each pair of users is given with a shared key. To broadcast an authenticated message, user will construct a separate authentication tag for every other user, concatenates them and will append it to the message. This method increases the amount of key storage at each user, produces a very long authentication tag for the message which results in high communication cost.

Desmedt, Frankel and Yung consider the problem with a single transmitter (fixed sender) in [12] and provide a solution: A trusted KDC distributes secret key information to all the users in the system. When a transmitter broadcasts a message to N receivers, they will individually verify the authenticity of the message using their secret key information. There are malicious groups of receivers - the size of the largest such group is less than k - who use their secret keys and all the previous communications in the system to construct fraudulent messages. Bounds and construction for multi-receiver authentication codes are given in [15], [17], [25], [24].

Safavi-Naini and Wang considered extensions of the scheme of Desmedt et al. [12] in [26], [27] and [25]. In [26] and [25], they relaxed the restriction that the sender be fixed beforehand. Instead, they allowed any one of the user to become a sender *after* the initial stage of key distribution by a trusted KDC. They call this as the case of the *dynamic* sender. They showed (Theorem 3.2, [26]) that, in

1 Introduction

In a conventional point-to-point authentication system [29], a sender wants to send a message over a public channel to a receiver. In a telephone conversation, separate channel is available for communication, hence no authentication is required. In applications like Internet based video conferencing, board meeting, scientific discussion etc., several users may need to communicate over a broadcast channel securely. There may be an active attacker who can try to perform *impersonation attack* or *substitution attack*. To provide protection against such attacks, the sender and receiver use an *authentication code*.

In [23], an authentication scheme is developed which enables all the users in the group to verify the intended sender

¹Throughout the paper q is a prime power such that $q \geq \text{size of message space}$. We assume that k and n are such that $\text{size of message space} \geq 2kn$.

any such scheme with perfect protection, the secret key information at each user and the size of the authentication tag sent along with the message, cannot be less than $2k \log_2 q$ bits and $k \log_2 q$ bits respectively; and they presented a scheme meeting these bounds. Here, q is a prime power such that $q \geq \text{size of message space}$.

In [27] authors dropped the restriction of a single sender, and proposed a scheme for the situation with t ($t > 0$) senders. This scheme uses symmetric polynomials in two variables over $GF(q)$ and is developed from Blom's key distribution scheme [6]. Here the size of the secret key information at each user, and of the authentication tag for a message, grew linearly with t .

In this paper we continue to consider the setup in [26], [27]. We drop the restriction on the number of senders. That is, we allow subset of N users or all N users $P_1, P_2, \dots, P_N$ to broadcast messages. A group of malicious parties - who number fewer than k (where k is the threshold) - may collude and try to launch an impersonation attack (by using their secret keys and all previous communications) against a pair, say P_i and P_j , by generating a message such that P_j accepts it as authentic and being sent from P_i . We develop schemes in which perfect protection is guaranteed against such attacks.

In our schemes, the secret storage required at each user, and the size of the authentication tag sent with a message, *do not* depend on the number of senders as they do in [27]. In fact, these sizes meet the lower bound on the sizes fixed by Theorem 3.2 [26] quoted above, which was proved for the case of a *single* (dynamic) sender. We present a scheme in which, for a given k and N , each party is required to store secret information of size $2k \log_2 q$ bits, and the size of an authentication tag is $k \log_2 q$ bits. (Here, we assume that k and N are such that size of message space $\geq 2kN$). In this scheme each party is required to store a further $8(N-1)k \log_2 q$ bits of information, which need not be guarded against exposure. But the security of the scheme is indifferent to exposure of this further information, either to an adversary or to the other participants.

We begin by treating the case $k = 2$ separately. We are unable to generalize the approach leading to this scheme to the case of arbitrary k . Hence for arbitrary k we present an alternate scheme in which the *total* storage at each participant is $4 \log_2 q$ bits (all of which is to be held secret from the adversary and the other participants), and the size of the authentication tag is $2 \log_2 q$ bits.

All our schemes are algebraic in nature. Further, they involve suitably distributing the evaluations and/or coefficients of polynomials of a single variable. It may here be recalled that the algebraic constructions of [12] also involve polynomials of a single variable, while those of [26], [27] involve polynomials in two variables. Our scheme can be used to construct a secure dynamic conferencing by combining with the scheme proposed by Blundo et. al. in [8].

We also present an application of our scheme to Verifiable Secret Sharing [14], [22]. In the setup of [14], [22], it is shown that it is not possible to satisfy the following requirements: the dealers should be able to deal shares of their secrets to the other participants without exposing any information about these secrets; at the same time, every k participants, who have verified the shares they received as correct, should recover the (same) secret from their shares. In our scheme we show that by allowing participants to possess private key information (which may be distributed at the outset

by a trusted KDC) permits the satisfaction of the above mentioned requirements.

We organize the paper as follows: Model used in the scheme is discussed in Section II, we discuss the case $k = 2$ in Section III, in Section IV we present the scheme for arbitrary k . We present an application of our scheme to Verifiable Secret Sharing in Section V. In Section VI we discuss providing authentication to group key management schemes proposed by Blundo et. al. [8] and Fiat-Naor [2]. We conclude the paper in Section VII.

2 Model

The system comprises of a trusted KDC and N users, say, $P_1, P_2, \dots, P_N$, who want to communicate over a broadcast channel. Out of N users in the system any user can broadcast a message to all other users in the system who can individually verify the authenticity of the received message. Hence all N users can act as both senders and receivers. The channel is subject to an active attack i.e., an attacker may try to perform impersonation attack or substitution attack. Also, users in the system may not trust each other, that is some users may collude to construct fraudulent messages. The scheme developed provides a perfect protection against collusion of up to k (threshold) members in which senders and receivers use an authentication code to verify the authenticity of the message received.

The scheme comprises of three phases:

1. **Key Distribution:** The trusted KDC picks and distributes private key information to all the users securely.
2. **Broadcast:** The sender broadcasts a message to all the other users in the system, along with an authentication tag.
3. **Verification:** Each user verifies the authenticity of the message broadcast by the sender.

3 A scheme for the case $k = 2$

Key distribution: A trusted KDC picks at random two polynomials $F_A(x)$ and $F_B(x)$, each of degree 2 over $GF(q)$. It sends $(F_A(2i-1), F_B(2i-1), F_A(2i), F_B(2i))$ to $P_i, i = 1, \dots, N$, securely.

Broadcast: P_i , in order to send a message s , broadcasts

$$(\alpha, \beta, s) = (F_A(2i-1) + sF_B(2i-1), F_A(2i) + sF_B(2i), s).$$

Verification: On receiving (α, β, s) , P_j finds the polynomial of degree 2 with values of α at $x = 2i-1$, β at $x = 2i$, and $F_A(2j-1) + sF_B(2j-1)$ at $x = 2j-1$. P_j accepts the message received from P_i as authentic, and sent by P_i , if this polynomial takes the value $F_A(2j) + sF_B(2j)$ at $x = 2j$.

Size of secret information: Each participant being required to store 4 values from $GF(q)$, the size of the secret information at each participant is $4 \log_2 q$ bits.

Proof of security: Suppose $P_{i'}$ ($i' \neq i, j$) is a malicious participant who, while sending a message s' , wishes to appear as P_i to the receiver P_j . Since P_i is claimed to be the origin of the transmission, the receiver P_j , in order to verify the authenticity of the message, expects the value of the polynomial $F_A(x) + s'F_B(x)$ at $x = 2i-1$ and $x = 2i$. But the secret key

at $P_{i'}$, $(F_A(2i' - 1), F_B(2i' - 1), F_A(2i'), F_B(2i'))$, along with s' , reveals no information about the values $F_A(2i - 1) + s'F_B(2i - 1)$ and $F_A(2i) + s'F_B(2i)$ (recall that $F_A(x)$ and $F_B(x)$ are of degree 2). Nor does $P_{i'}$ gain any information about these values from observing $F_A(2i - 1) + s''F_B(2i - 1)$ and $F_A(2i) + s''F_B(2i)$, corresponding to previous message(s) s'' .

4 A scheme for arbitrary k

Key distribution: The trusted KDC fixes the security parameter k . It picks at random $2Nk$ ($0 \leq k \leq N$) elements $a_{11}, a_{12}, \dots, a_{1k}, a_{21}, a_{22}, \dots, a_{2k}, \dots, a_{N1}, a_{N2}, \dots, a_{Nk}$, and $b_{11}, b_{12}, \dots, b_{1k}, b_{21}, b_{22}, \dots, b_{2k}, \dots, b_{N1}, b_{N2}, \dots, b_{Nk}$, from $GF(q)$. It sends the $2k$ elements $a_{i1}, a_{i2}, \dots, a_{ik}$, and $b_{i1}, b_{i2}, \dots, b_{ik}$ to P_i *securely*. Denote by $A_i(x)$ the polynomial $a_{i1}x^k + a_{i2}x^{k-1} + \dots + a_{ik}x$, and by $B_i(x)$ the polynomial $b_{i1}x^k + b_{i2}x^{k-1} + \dots + b_{ik}x$. For each ordered pair (i, j) ($i \neq j$) it determines an $\alpha_{i,j}$ satisfying the equation

$$\alpha_{i,j}^k A_i(\alpha_{i,j}) + A_j(\alpha_{i,j}) = 1;$$

in other words, such that

$$a_{i1}\alpha_{i,j}^{2k} + a_{i2}\alpha_{i,j}^{2k-1} + \dots + a_{ik}\alpha_{i,j}^{k+1} + a_{j1}\alpha_{i,j}^k + \dots + a_{jk}\alpha_{i,j} = 1.$$

It then determines the value $\beta_{i,j}$ taken by the polynomial $x^k B_i(x) + B_j(x)$ at $x = \alpha_{i,j}$, i.e.,

$$\beta_{i,j} = \alpha_{i,j}^k B_i(\alpha_{i,j}) + B_j(\alpha_{i,j})$$

(Notice that, since $x^k A_i(x) + A_j(x)$ is a polynomial of degree $2k$ over $GF(q)$, $\alpha_{i,j}$ (therefore also $\beta_{i,j}$) may be taken to be elements of $GF(q^{2k})$.) It *publicly* sends to P_i the values $\alpha_{i,j}, \alpha_{j,i}, \beta_{i,j}$ and $\beta_{j,i}$, $j \in \{1, \dots, N\} \setminus \{i\}$. (Thus any participant could find out all the $\alpha_{i,j}$'s and, $\beta_{i,j}$'s.)

Broadcast: P_i , in order to send a message s , broadcasts $(a_{i1} + sb_{i1}, a_{i2} + sb_{i2}, \dots, a_{ik} + sb_{ik}, s)$.

Verification: In effect P_i has broadcast the message s and the polynomial $F_{s,i}(x) = A_i(x) + sB_i(x)$. On receiving this, P_j forms the polynomial $x^k \{F_{s,i}(x)\} + A_j(x) + sB_j(x)$ using its secret information (which, in effect, consists of the polynomials $A_j(x)$ and $B_j(x)$). It verifies that this polynomial evaluates to $1 + s\beta_{i,j}$ at $x = \alpha_{i,j}$, which will indeed be the case, since

$$\begin{aligned} & \alpha_{i,j}^k \{F_{s,i}(\alpha_{i,j})\} + A_j(\alpha_{i,j}) + sB_j(\alpha_{i,j}) \\ &= \alpha_{i,j}^k \{A_i(\alpha_{i,j}) + sB_i(\alpha_{i,j})\} + A_j(\alpha_{i,j}) + sB_j(\alpha_{i,j}) \\ &= \alpha_{i,j}^k A_i(\alpha_{i,j}) + A_j(\alpha_{i,j}) + s\{\alpha_{i,j}^k B_i(\alpha_{i,j}) + B_j(\alpha_{i,j})\} \\ &= 1 + s\beta_{i,j} \end{aligned}$$

Storage required: P_i is required to hold the $2k$ values $a_{i1}, a_{i2}, \dots, a_{ik}, b_{i1}, \dots, b_{ik} \in GF(q)$ in secret. This comes to $2k \log_2 q$ bits of secret information. P_i has to also store the $4(N - 1)$ elements $(\alpha_{i,j}, \alpha_{j,i}, \beta_{i,j}, \beta_{j,i}) \in GF(q^{2k})$, $j \in \{1, \dots, N\} \setminus \{i\}$, but these do not need to be guarded against exposure. This is a further $8(N - 1)k \log_2 q$ bits.

Proof of security: Suppose $P_{i1}, P_{i2}, \dots, P_{ik-1}, i_1, \dots, i_{k-1} \in \{1, \dots, N\} \setminus \{i, j\}$, is a group of malicious receivers who wish to impersonate P_i while sending the message s' to P_j . In order to do so, they need to determine the k coefficients of the polynomial

$F_{s',i}(x) = A_i(x) + s'B_i(x)$. The secret information they possess between them corresponds to the conditions

$$F_{s',i}(\alpha_{i,i_r}) = \alpha_{i,i_r}^{-k} \{1 + s'\beta_{i,i_r} - A_{i_r}(\alpha_{i,i_r}) - s'B_{i_r}(\alpha_{i,i_r})\}$$

for $r = 1, \dots, k - 1$. Since these fix the evaluations of $x^{-1}F_{s',i}(x)$, which is known only to be a polynomial of degree $= k - 1$ and on whose coefficients there are no further restrictions, only at $k - 1$ points, they only determine a set of q polynomials (of degree $= k - 1$) to which $x^{-1}F_{s',i}(x)$ belongs.

5 Application to Verifiable Secret Sharing

Suppose in a group of N users some users (called *dealers*) have secrets, and each dealer wishes to distribute his secret as shares to the remaining users. Since the users do not trust each other completely, they insist that they be able to verify (without talking with the dealer, or the other users) that the shares they receive are consistent. That is, for every secret, if any k users have each verified their corresponding shares as correct, then these shares should reconstruct that secret. At the same time, the dealer would like to create shares each of which reveal no information about the secret.

Schemes to achieve this, called (non-interactive) Verifiable Secret Sharing schemes, were first investigated in [14] and [22]. While in the scheme of [14] an infinitely powerful adversary can learn the secret by listening to broadcast information, in the scheme of [22] an *infinitely powerful dealer* can compute inconsistent shares.

Motivated by this it is natural to seek, as Pedersen points out in [22], a non-interactive verifiable secret sharing scheme satisfying the following requirements:

1. No information about the secret is revealed, and
2. Even an infinitely powerful dealer cannot compute inconsistent shares.

But he shows that it is impossible to construct such a scheme in the setting he considers.

In particular, in that setting the users have no (prior) private information apart, possibly, from the secrets which they wish to share. Here, we change the setting to allow a trusted KDC, at the outset, to distribute private keys to the users. (Thus while distributing these keys, KDC has no knowledge about the secrets which some of the users may later decide to share. The KDC has no further role after this key distribution phase.) We ask if some of the users, who later have secrets they wish to distribute, can take advantage of the private keys they possess, in order to construct the shares satisfying the requirements (1) and (2).

They can do this in a simple way, by making use of the message authentication codes presented in the previous section. The idea is for the dealer to determine the shares of the secret using Shamir's secret sharing scheme [28], and send to each user (*in secret*) his corresponding share along with the authentication tag calculated by treating this share as a message. That (1) is satisfied is obvious. That (2) is satisfied is to say that substitution attacks on the underlying authentication code fail, and this we have already shown in the previous section.

Let us outline our proposal in detail: for each i , $i = 1, \dots, N$, the trusted KDC picks at random the $2k$ elements $a_{i1}, a_{i2}, \dots, a_{ik}, b_{i1}, b_{i2}, \dots, b_{ik}$ and sends these elements to P_i in secret. Denote by $A_i(x)$ the polynomial $a_{i1}x^k + a_{i2}x^{k-1} + \dots + a_{ik}x$, and by $B_i(x)$ the polynomial $b_{i1}x^k + b_{i2}x^{k-1} + \dots + b_{ik}x$. For each ordered pair (i, j) ($i \neq j$) it determines $\alpha_{i,j}$ satisfying the equation

$$\alpha_{i,j}^k A_i(\alpha_{i,j}) + A_j(\alpha_{i,j}) = 1$$

and determines the value $\beta_{i,j}$ taken by the polynomial $x^k B_i(x) + B_j(x)$ at $x = \alpha_{i,j}$. It publicly sends to P_i the values $\alpha_{i,j}$, $\alpha_{j,i}$, $\beta_{i,j}$ and $\beta_{j,i}$, $j \in \{1, \dots, N\} \setminus \{i\}$. After this the KDC has no further role.

Suppose that, at a later point in time, a user P_i has a secret s , which it wishes to share. It determines N shares $s_1, s_2, \dots, s_N$ using Shamir's (N, k) threshold secret sharing scheme [28] and sends to P_j , $j = 1, \dots, N$, $(a_{i1} + s_j b_{i1}, a_{i2} + s_j b_{i2}, \dots, a_{ik} + s_j b_{ik}, s_j)$ (in effect, $(F_{s_j,i}(x), s_j)$) in secret.

On receiving this, P_j forms the polynomial $x^k \{F_{s_j,i}(x)\} + A_j(x) + s_j B_j(x)$ using its secret information, and verifies that this polynomial evaluates to $1 + s_j \beta_{i,j}$ at $x = \alpha_{i,j}$.

6 Secure Authentic Communication

Providing authentication for the messages exchanged between group members in addition to confidentiality is an important issue in SGC. Several group key management techniques have been proposed [8, 11, 9, 30, 19, 3, 20, 2, 18, 4, 1, 21, 13, 10, 7, 5]. All these schemes address computation of group key for confidential communication among the group members. Among the above mentioned schemes, we consider two group key management schemes: one proposed by Blundo et al. [8] and the other proposed by Fiat-Naor [2] and try to provide secure authentic group communication by using the authentication scheme we have developed.

6.1 Blundo et al. Authentic SGC Scheme

In [8], a non-interactive protocol has been developed to derive a common group key for secure communication. It provides only confidentiality of the messages exchanged between the users of the secure group. Here, we apply the authentication scheme developed to provide authentication for the protocol in [8]. To provide authentication using our multi-party authentication scheme, each user is required to store secret information of size $3k \log_2 q$ bits. This incurs an extra storage cost for the users. To avoid this extra storage, our scheme makes use of a part of the available information with the users which is used for group key computation of Blundo et al. conference keying protocol. So, no extra information is generated and communicated by KDC to users in order to provide authenticity for the group communication and no extra storage is required at the users.

Our Secure Authentic Communication scheme comprises of seven phases: *Polynomial Selection*, *Key Distribution*, *Polynomial Construction for Authentication*, *Computation of $\alpha_{i,j}$ and $\beta_{i,j}$* , *Group Key Computation*, *Secure Authentic Communication* and *Verification*.

The Blundo et al. [8] non-interactive group keying protocol is given in Appendix A. In order for the users in the

secure group to communicate with confidentiality and authenticity, we can combine the features of our authentication scheme with that of Blundo et al. [8] group keying protocol. Figure 1 demonstrates the secure authentic group communication for a group with t users.

In the protocol each user u_i , $i = 1, \dots, N$ is required to store in secret $\binom{k+t-2}{t-2}$ values from $GF(q)$, which is same as the storage required by Blundo et al. scheme. That means we provide authentication facility for the Blundo et al. group communication scheme with the existing information, without incurring extra storage overhead at the users. Every user u_i , $i = 1, \dots, N$, after receiving the polynomial $P(i, x_2, \dots, x_t)$, will pick $2k$ elements for authentication from this polynomial itself. In the protocol it is specified that u_i , $i = 1, \dots, N$ picks $2k$ elements sequentially starting from the i^{th} coefficient in cyclic order from the coefficients of the polynomial i.e., u_1 picks $2k$ elements starting from coefficient 1 to coefficient $2k$, u_2 picks $2k$ elements starting from coefficient $2k+1$ and so on. The use of this assumption is two fold. First, no two users can construct the same polynomials $A(x)$ and $B(x)$ i.e., for every two users u_i and u_j ($i \neq j$), $A_i(x) \neq A_j(x)$ and $B_i(x) \neq B_j(x)$. Second, this helps KDC to construct the polynomials $A(x)$ and $B(x)$ for each user and to compute the values of $\alpha_{i,j}$ and $\beta_{i,j}$ for each ordered pair (i, j) ($i \neq j$), which reduces the computation burden with the users.

If $(k-1)$ malicious parties $u_{i_1}, u_{i_2}, \dots, u_{i_{k-1}}, i_1, \dots, i_{k-1} \in \{1, \dots, N\} \setminus \{i, j\}$ collude and try to impersonate u_i while sending a message to u_j , then u_j fails to verify the authenticity of u_i as per the proof of security illustrated in Section IV. Hence the scheme is secure against collusion of fewer than k malicious parties and provides confidential authentic communication between group members which is more appropriate for the applications like scientific discussion, board meeting etc.

6.2 Fiat-Naor Authentic SGC Scheme

The Broadcast Encryption is the problem of sending an encrypted message to a larger user base such that the message can only be decrypted by a dynamically changing privileged subset. A Broadcast encryption scheme is a collection of algorithms that allows a centralized transmitter to send encrypted message to a collection of users such that only a privileged subset of users decrypt them [16]. This is made possible with the computation of a secret key which is known only to members of the privileged set.

As another key management scheme to provide authentication, we consider an information theoretic based key management scheme proposed by Amos Fiat and Moni Naor [2] and we provide authenticity for the messages exchanged. Fiat and Naor [2] were first to introduce broadcast encryption and they suggested methods for securely broadcasting information among privileged user set. The broadcast information can only be decrypted by authorized users and coalition of up to k other users will reveal no information about the secret key.

The basic scheme proposed by Fiat and Naor in [2] allows users to determine a common key for every subset, which is resilient to any set S of size $\leq k$ and is discussed in Appendix B. Here, in Figure 2 we discuss the protocol for providing authentication to this scheme.

- **Polynomial Selection:** KDC picks at random a symmetric polynomial $P(x_1, \dots, x_t)$ of degree k with t variables whose coefficients are from $GF(q)$, $q > N$ and is a prime.
- **Key Distribution:** To each user u_i , $i = 1, \dots, N$, in the system, KDC distributes the polynomial $f_i(x_2, \dots, x_t) = P(i, x_2, \dots, x_t)$, that is the polynomial obtained by evaluating $P(x_1, \dots, x_t)$ at $x_1 = i$.
- **Polynomial Construction for Authentication:** Each user u_i , $i = 1, \dots, N$, picks $2k$ elements sequentially starting from i^{th} coefficient in cyclic order from the coefficients of the symmetric polynomial distributed by KDC in Key Distribution step. Let these coefficients be denoted as $a_{i1}, a_{i2}, \dots, a_{ik}$, and $b_{i1}, b_{i2}, \dots, b_{ik}$, and constructs the polynomials $A_i(x) = a_{i1}x^k + a_{i2}x^{k-1} + \dots + a_{ik}x$, and $B_i(x) = b_{i1}x^k + b_{i2}x^{k-1} + \dots + b_{ik}x$.
- **Computation of $\alpha_{i,j}$ and $\beta_{i,j}$:** KDC also constructs the polynomials $A_i(x)$ and $B_i(x)$ for each user u_i , $i = 1, \dots, N$ as depicted in previous step.

For each ordered pair (i, j) ($i \neq j$) it determines $\alpha_{i,j}$ satisfying the equation

$$\alpha_{i,j}^k A_i(\alpha_{i,j}) + A_j(\alpha_{i,j}) = 1;$$

and determines the value $\beta_{i,j}$ taken by the polynomial $x^k B_i(x) + B_j(x)$ at $x = \alpha_{i,j}$, i.e., $\beta_{i,j} = \alpha_{i,j}^k B_i(\alpha_{i,j}) + B_j(\alpha_{i,j})$.

- **Group Key Computation:** If the users $u_{j_1}, \dots, u_{j_t}$ want to set up a group key then each user u_{j_i} evaluates $f_{j_i}(x_2, \dots, x_t)$ at $(x_2, \dots, x_t) = (j_1, \dots, j_{i-1}, j_{i+1}, \dots, j_t)$. The group key for users $u_{j_1}, \dots, u_{j_t}$ is equal to $BK = P(j_1, \dots, j_t)$.
- **Secure Authentic Communication:** If u_i wants to send a message m securely, it broadcasts $(a_{i1} + mb_{i1}, a_{i2} + mb_{i2}, \dots, a_{ik} + mb_{ik}, E_{BK}(m))$ i.e., it broadcasts the polynomial $F_{m,i}(x) = A_i(x) + mB_i(x)$ along with the encrypted message.
- **Verification:** Upon receiving this information, u_j first decrypts $E_{BK}(m)$ using group key BK to get the message m . Now, u_j constructs the polynomial $x^k \{F_{m,i}(x)\} + A_j(x) + mB_j(x)$ and verifies that this polynomial evaluates to $1 + m\beta_{i,j}$ at $x = \alpha_{i,j}$.
Hence u_j has verified the authenticity of the sender i.e., u_i .

Figure 1: Protocol for Blundo et al. Secure Authentic Group Communication

- **Selection of Keys :** KDC randomly selects $\sum_{i=0}^k \binom{N}{i}$ keys from $GF(q)$, $q > N$ and is a prime.
- **Distribution of Keys :** KDC distributes every key K_B selected for a subset $B \subset U$, to every user $x \in U - B$.
- **Polynomial construction for Authentication:** Each user u_i , $i = 1, \dots, N$ considers first $2k$ number of keys received by KDC as coefficients for constructing the polynomials. Let these coefficients be denoted as $a_{i1}, a_{i2}, \dots, a_{ik}$, and $b_{i1}, b_{i2}, \dots, b_{ik}$, and let the polynomials be $A_i(x) = a_{i1}x^k + a_{i2}x^{k-1} + \dots + a_{ik}x$, and $B_i(x) = b_{i1}x^k + b_{i2}x^{k-1} + \dots + b_{ik}x$.
- **Computation of $\alpha_{i,j}$ and $\beta_{i,j}$:** KDC also constructs the polynomials $A_i(x)$ and $B_i(x)$ for each user u_i , $i = 1, \dots, N$ as depicted in previous step.
For each ordered pair (i, j) ($i \neq j$) it determines an $\alpha_{i,j}$ satisfying the equation

$$\alpha_{i,j}^k A_i(\alpha_{i,j}) + A_j(\alpha_{i,j}) = 1;$$
 and determines the value $\beta_{i,j}$ taken by the polynomial $x^k B_i(x) + B_j(x)$ at $x = \alpha_{i,j}$, i.e., $\beta_{i,j} = \alpha_{i,j}^k B_i(\alpha_{i,j}) + B_j(\alpha_{i,j})$.
- **Key Computation by privileged user set:** Let T denote the privileged user set. The common secret key, SK to the privileged set T is obtained by performing exclusive-or of all keys K_B , $B \in U - T$.
- **Adding Authenticity for Message Broadcast:** If u_i wants to send a message m securely, it broadcasts $(a_{i1} + mb_{i1}, a_{i2} + mb_{i2}, \dots, a_{ik} + mb_{ik}, E_{SK}(m))$ i.e., it broadcasts the polynomial $F_{m,i}(x) = A_i(x) + mB_i(x)$ along with encrypted message.
- **Verification:** Upon receiving this information, u_j first decrypts $E_{SK}(m)$ using secret key SK to get the message m . Now, $User_j$ constructs the polynomial $x^k \{F_{m,i}(x)\} + A_j(x) + mB_j(x)$ and verifies that this polynomial evaluates to $1 + m\beta_{i,j}$ at $x = \alpha_{i,j}$.
Hence u_j has verified the authenticity of the sender i.e., u_i .

Figure 2: Protocol for Authentic Broadcast Encryption

KDC randomly selects $\sum_{i=0}^k \binom{N}{i}$ keys from $GF(q)$.

Keys in the system are of the form $K_0, K_1, \dots, K_N, K_{1,2}, K_{1,3}, \dots, K_{1,N}, K_{2,3}, K_{2,4}, \dots, K_{2,N}, \dots, K_{N-1,N}, \dots, K_{1,2,\dots,k}, \dots, K_{N-k,N-k+1,\dots,N}$. Among these keys in the system, KDC assigns each user $User_i$ with the keys which do not possess i as the subscript.

For example, let $|U|=10$ and $|T|=6$ i.e., out of 10 users in the system 6 are privileged and let threshold be $k=3$. Keys in the system are $K_0, K_1, \dots, K_{10}, K_{1,2}, K_{1,3}, \dots, K_{1,10}, \dots, K_{1,2,3}, \dots, K_{8,9,10}$. $User_1$ is assigned with keys $K_0, K_2, K_3, \dots, K_{10}, K_{2,3}, K_{2,4}, \dots, K_{2,10}, K_{3,4}, \dots, K_{9,10}, K_{2,3,4}, \dots, K_{8,9,10}$ i.e., keys which do not have 1 as subscript. Similarly, keys are distributed to other users in the system.

Users will construct polynomials $A(x)$ and $B(x)$ for authentication using the same keys which are used for key computation. Hence no extra information is required for providing authentication, which does not incur extra storage at the users. KDC also constructs the polynomials $A(x)$ and $B(x)$ for all the users in the system, which allows KDC to compute the values of $\alpha_{i,j}$ and $\beta_{i,j}$ for each ordered pair (i,j) ($i \neq j$), thus reducing the computation burden with the users.

As illustrated in Section IV, this scheme is also secure against collusion of fewer than k malicious parties.

7 Conclusion and Future Work

Providing authenticity for the messages exchanged between users in the system is an important issue in secure dynamic group communication. In this paper we have developed schemes which provide perfect protection against colluding malicious parties. In our scheme, for a given k and N , each party is required to store secret information of size $2k \log_2 q$ bits and $k \log_2 q$ bits of authentication tag. We have applied our authentication scheme to verifiable secret sharing, in which users in the system can verify the correctness of the share they received. We have tried to provide authentication to group key management schemes proposed by Blundo et al. and Fiat-Naor without incurring extra storage cost.

In this paper We have provided authentication for group key management schemes proposed by Blundo et al. and Fiat-Naor. As a future work, the same concept can be extended to provide authentication for other group communication schemes. In all the schemes it may not be possible to provide authentication without incurring extra storage cost. But, with little bit extra storage, authentic group communication can be made possible among group members and same thing may be verified.

Appendix A

Blundo et al. non-interactive k -secure t -group protocol is as follows:

- KDC picks at random a symmetric polynomial $P(x_1, \dots, x_t)$ of degree k with t variables with coefficients over $GF(q)$, $q > N$.
- To each user $User_i$, $i = 1, \dots, N$, in the system, KDC distributes the polynomial $f_i(x_2, \dots, x_t) = P(i, x_2, \dots, x_t)$, that is the polynomial obtained by evaluating $P(x_1, \dots, x_t)$ at $x_1 = i$.
- If the users $User_{j_1}, \dots, User_{j_t}$ want to set up a group

key, then each user $User_{j_i}$ evaluates $f_{j_i}(x_2, \dots, x_t)$ at $(x_2, \dots, x_t) = (j_1, \dots, j_{i-1}, j_{i+1}, \dots, j_t)$.

- The group key for users $User_{j_1}, \dots, User_{j_t}$ is equal to $s_{j_1, \dots, j_t} = P(j_1, \dots, j_t)$.

Appendix B

Fiat and Naor proposed a basic scheme which allows users to determine a common key for every subset, which is resilient to any set S of size $\leq k$. In this scheme, for a given k each user is preassigned with a set of keys such that, once the users are told which of them are in the privilege set and which are not, each user in the privilege set can construct the group key on its own. This scheme considers a set U of n users. For every set $B \subset U, 0 \leq |B| \leq k$, a key K_B is given to every user $x \in U - B$. Let T denote the privilege user set. The group key to the privilege set T is obtained by performing exclusive-or of all keys $K_B, B \in U - T$.

The scheme is resilient to every coalition of up to k users, since even if up to k users collude, they all will be missing the key K_S . Hence they are unable to compute the group key. In this scheme, each user in the system is assigned with $\sum_{i=0}^k \binom{N-1}{i}$ keys and each user is required to perform $\sum_{i=0}^k \binom{N-t}{i}$ exclusive-or operations to obtain the group key.

REFERENCES

- [1] A.Ballardie. Scalable Multicast Key Distribution, 1996. Request For Comments 1949, Internet Engineering Task Force.
- [2] A.Fiat and M.Naor. Broadcast Encryption. pages 480–491, Proceedings of CRYPTO'93, 1993.
- [3] A.Perrig. Efficient Collaborative Key Management Protocol for Secure Autonomous Group Communication. Proceedings of International Workshop CryptEC, 1999.
- [4] A.T.Sherman and D.A.McGrew. Key Establishment in Large Dynamic groups using One-way Function trees. *IEEE Transactions on Software Engg.*, 29, No.5 :444–458, 2003.
- [5] Ayan Roy-Chowdhary, John S Baras. Energy Efficient Source Authentication for Secure Group Communication with Low-Powered Smart Devices in Hybrid Wireless/Satellite Networks. *Eurasip Journal on Wireless Communications and Networking*, 2011.
- [6] Blom.R. An Optimal Class of Symmetric Key Generation Systems. pages 335–338, Proceedings of Advances in Cryptology-Eurocrypt '84, LNCS, 209, 1985.
- [7] Bruhadeshwar B, Kulkarni S S. Balancing Revocation and Storage Trade-offs in Secure Group Communication. *IEEE Transaction on Dependable and Secure Computing*, 8(1):58–73, 2011.
- [8] C.Blundo, A.De Santis, A.Herzberg, S.Kutten, U. Vaccaro, M. Yung. Perfectly Secure Key Distribution for Dynamic Conferences. In *Advances in Cryptology-CRYPTO'92*, pages 471–486, LNCS, 740, 1993.
- [9] C.K.Wong, M. Gouda, and S.S. Lam. Secure Group Communication Using key Graphs. *IEEE/ACM Transactions on Networking*, 8(1):1630, Feb. 2000.

- [10] C.K.Wong., Simon S. Lam. Keystone: A Group Key Management Service. In *International Conference on Telecommunications*, Acapulco, Mexico, May 2000.
- [11] D.Balenson, D.McGrew, and A.Sherman. Key Management for Large Dynamic Groups: One-Way Function Trees and Amortized Initialization, August 2000. Internet Draft, draft-irtf-smug-groupkeymgmt-oft-00.txt.
- [12] Desmedt.Y, Frankel.Y, Yung.M. Multi-receiver/Multi-sender Network Security: Efficient Authenticated Multicast/Feedback. pages 2045–2054, Proceedings of IEEE Infocom '92, 1992.
- [13] D.Wallner, E.Harder and R.Agee. Key Management for Multicast: Issues and Architectures, June 1999. Request For Comments (Informational)2627, Internet Engineering Task Force.
- [14] Feldman.P. A practical scheme for non-interactive verifiable secret sharing. pages 427–437, Proceedings of the 28th IEEE Symposium on the Foundations of Computer Science, 1987.
- [15] Fujii.H, Kachen.W, Kurosawa.K. Combinatorial Bounds and Design of Broadcast Authentication. *IEICE Transactions*, E79-A:502–506, 1996.
- [16] Jeremy Horwitz. A survey of Broadcast Encryption, 13 Jan. 2003. Manuscript.
- [17] Kurosawa.K, Obana.S. Characterisation of (k, n) Multi-receiver Authentication. pages 204–215, Information Security and Privacy, ACISP'97, LNCS, 1270, 1997.
- [18] M.Burmester and Y.Desmedt. A Secure and Efficient Conference Key Distribution System. In *Advances in Cryptology*, -EUROCRYPT'94, 1994.
- [19] M.P.Howarth, S.Iyengar, Z.Sun, and H.Cruickshank. Dynamics of Key Management in Secure Satellite Multicast. *IEEE Journal on selected Areas in Communication (JSAC)*, Feb. 2004.
- [20] M.Steiner, G.Tsudik and M.Waidner. Diffie-Hellman key distribution extended to group communication. pages 31–37, In SIGSAC proceedings of the Third ACM conference on Computer and Communications Security. New Delhi, India, ACM, New York, March 1996.
- [21] M.Waldvogel, G.Caronni, D.Sun, N.Weiler, and B.Plattner. The Versakey Frameworks: Versatile Group Key Management. *IEEE Journal on selected Areas in Communication (JSAC)*, 17(9):1614–1631, Sept. 1999.
- [22] Pedersen.T.P. Non-interactive and Information-Theoretic Secure Verifiable Secret Sharing. *Advances in Cryptology - CRYPTO'91*, Lecture Notes in Computer Science, 576, 1992.
- [23] R.Aparna, B.B.Amberker, Prashant Koulgi. Multi-sender Multi-receiver Message Authentication and an Application to Verifiable Secret Sharing. National conference on Mathematical Foundations of Coding, Complexity, Computation and Cryptography, IISc., Bangalore, June 2006.
- [24] Safavi-Naini.R, Wang.H. Bounds and Constructions for Multireceiver Authentication codes. pages 242–256, *Advances in Cryptology-Asiacrypt '98*, LNCS, 1514, 1998.
- [25] Safavi-Naini.R, Wang.H. New Results on Multi-receiver Authentication codes. pages 527–541, *Advances in Cryptology-Eurocrypt '98*, LNCS, 1438, 1998.
- [26] Safavi-Naini.R, Wang.H. Multi-receiver authentication codes: Models, Bounds, Constructions and Extensions. pages 148–172, *Information and Computation*, 151, 1999.
- [27] Safavi-Naini.R., Wang.H. Broadcast Authentication for Group Communication. pages 1–21, *Theoretical Computer Science*, 269 (1-2), 2001.
- [28] Shamir, A. How to Share a Secret. *Communications of the ACM*, 22:612–613, 1979.
- [29] Simmons.G.J. A Survey of Information Authentication. *Contemporary Cryptology, The Science of Information Integrity*, G.J.Simmons, ed., IEEE Press, pages 379–419, 1992.
- [30] W.Diffie and M.Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, IT 22(6):644–654, Nov. 1976.