Conference Key Establishment Using Polynomials

Lein Harn and Guang Gong*

Department of Networking
University of Missouri at Kansas City
Email: lharn@umkc.edu
*Department of Electrical and Computer Engineering
University of Waterloo, Waterloo, Ontario, Canada
Email: ggong@uwaterloo.ca

Abstract. In 1992, Blundo *et al.* have proposed a non-interactive k-secure m-conference protocol based on an m-variate polynomial over a finite field $\mathbb{F}_p$. The key distribution center (KDC) is responsible to pick a symmetric m-variate polynomial of degree k and generate shares for users. Each share is a symmetric polynomial involving $m-1$ variables of degree k, and needs to store the coefficients of such a symmetric polynomial. The storage space of each user is exponentially proportional to both the size of the conference and the security level. This makes their protocol impractical except for the case of the pairwise key distribution of $m = 2$. In this paper, we propose a non-interactive k-secure m-conference protocol based on a special type of multivariate polynomials of degree k. The advantage of using this type of multivariate polynomials can limit the storage space of each user to be $k + 1$, the number of the coefficients of a univariate polynomial of degree k, which is independent of the conference size. Furthermore, the shares generated for a k-secure m-conference protocol can support any conference with size t where $t \leq m$. We show that the proposed protocol is information-theoretic secure in terms of a concept of homomorphic random source and can resist known shares and known conference keys attacks.

Keywords. Conference key distribution, non-interactive conference key protocol, multi-variate symmetric polynomial, homomorphic random source, secret share, Lagrange interpolation.

1 Introduction

In a secure communication involving n members ($n \geq 2$), a conference key is needed to be shared among group users and uses it to encrypt and authenticate messages. A conference key establishment protocol is a method to enable multiple users to share a secret conference key. Most well-known conference key establishment protocols can be classified into two categories:

(a) Centralized conference key establishment protocols: a key distribution center (KDC) is engaged in managing the entire group.
(b) Distributed conference key establishment protocols: there is no explicit KDC, and each group member can contribute to the key generation and distribution.

The class of centralized conference key management protocols is the most widely used protocols due to its efficiency in implementation. For example, the IEEE 802.11i standard [1] employs an online server to select a conference key and transport it to each group member. The server in the IEEE 802.11i encrypts the group temporal key (GTK) using the key encryption key (KEK) obtained from the authentication server and the server transmits the encrypted message to each mobile client (group member) separately. In IEEE 802.11i, the confidentiality of conference key is

protected by conventional encryption algorithm which is conditionally secure. Recently, Harn and Lin [10] have proposed an authenticated conference key transfer protocol based on secret sharing. Their protocol uses a RSA modulus to resist the insider attack. One major requirement of the centralized conference key management protocols is that a pre-shared key is needed between each group member and the KDC.

In 1992, Blundo *et al.* [3] have proposed a non-interactive k-secure m-conference protocol based on a multivariate polynomial. Their protocol can establish a conference key of m participants. A system is said to be k-secure m-conference key scheme if any k users, pooling together their shares, have no information on keys that they should not know and m users in the qualified set can compute their conference key using their shares. The KDC is responsible to pick a symmetric m-variate polynomial of degree k, $F(x_1, \cdots, x_m)$, and generates shares, $F_i(x_2, \cdots, x_m)$ for users $i = 1, \cdots, l$. Then, later, each user can use his share to establish a conference involving m members. Since each share is a symmetric polynomial involving $m-1$ variables of degree k, each user needs to store all its coefficients in a finite field with p elements where p is a prime, denoted as $\mathbb{F}_p$. The storage space of each user is exponentially proportional to both the size m of the conference and the security level k. This makes their protocol impractical when $m > 2$. In [3], it has shown that the k-secure 2-conference protocol is a special case of Bloms scheme [2] and it is based on a symmetric bivariate polynomial. The storage space of each user needs only to store $k+1$ coefficients from $\mathbb{F}_p$. This special case can significantly reduce the size of stored information for each user.

Since then, key distribution using symmetric bivariate polynomial has been widely adopted in communication applications, such as in the sensor network [15][16][4][19][14][17]. The general design approach of these schemes is that a server picks a symmetric bivariate polynomial and generates shares for users. Each share is sent to user secretly. Due to the property of symmetry of a bivariate polynomial, secret communication keys can be derived from these shares when secure peer-to-peer communication is needed. There are some key establishment schemes that use polynomials other than a bivariate polynomial. For example, the key establishment scheme proposed by Zhou *et al.* [20] is based on a trivariate polynomial and the scheme proposed by Fanian *et al.* [6] is based on an m-variate polynomial. Although both schemes used a multivariate polynomial, they can only establish a secret key for two users.

In this paper, we proposed a non-interactive k-secure m-conference protocol based on a special type of multivariate polynomials of degree k. The advantage of using this type of multivariate polynomials in the design of any non-interactive k-secure m-conference protocol can limit the storage space of each user to be the number of the coefficients of a univariate polynomial of degree k. In addition, we show that shares generated for a k-secure m-conference protocol can support any conference with size t, where $t \leq m$. The contributions of this paper are below.

- A non-interactive k-secure m-conference protocol with the storage space of each user to be the coefficients of a special univariate polynomial over $\mathbb{F}_p$ of degree k is proposed. Shares generated for a k-secure m-conference protocol can support any conference with size t, where $t \leq m$.
- The scheme is information-theoretic secure in terms of a concept of homomorphic random source and can resist known shares and known conference keys attacks.

The rest of this paper is organized as follows. In the next section, we review of key distribution protocol proposed by Blundo *et al.* [3]. In Section 3, we introduce our non-interactive k-secure m-conference protocol. We show that the proposed m-conference key protocol is k-secure with

information-theoretic security in Section 4 and security analysis for resisting attacks from known shares and known conference keys in Section 5. Section 6 concludes the paper.

Note that one related problem of key distribution schemes using polynomials is that faulty shares of users may be caused by the shares generation or share distribution. We will discuss those issues in a separate paper.

2 Preliminaries and Review of Key Distribution Protocol Proposed by Blundo *et al.*

Blundo *et al.* [3] have proposed a non-interactive key distribution protocol based on a multivariate polynomial. In this section, we introduce some basic concepts on symmetric polynomials and the protocol. The following notations will be used throughout of the paper: n is a positive integer, $\mathbb{Z}_n = \{0, 1, \cdots, n-1\}$, and $S_n = \{1, 2, \cdots, n\}$; p is a prime number, $\mathbb{F}_p = \mathbb{Z}_p = \{0, 1, \cdots, p-1\}$ is a prime finite field with p elements, and $\mathbb{F}_p^*$ is the multiplicative group of $\mathbb{F}_p$; $\mathbb{Z}_r^s = \{(x_0, x_1, \cdots, x_{s-1}) \,|\, x_i \in \mathbb{Z}_r\}$, when $r = p$ is a prime, $\mathbb{Z}_p^s$ is an s-dimensional linear space over $\mathbb{F}_p$; $\mathbb{F}_p[x]$ is the set consisting of all polynomials with coefficients in $\mathbb{F}_p$; and $P\{A = t\}$ denotes the probability of an event that a discrete random variable A takes some value t in a sample space.

2.1 Symmetric Polynomials

A symmetric m-variate polynomial of degree k respect to each variable is defined as

$$F(x_1, \cdots, x_m) = \sum_{(j_1, \cdots, j_m) \in \mathbb{Z}_{k+1}^m} a_{j_1 \cdots j_m} x_1^{j_1} \cdots x_m^{j_m}, \; a_{j_1 \cdots j_m} \in \mathbb{F}_p \tag{1}$$

where $F(x_1, \cdots, x_m) = F(x_{\sigma(1)}, \cdots, x_{\sigma(m)})$ for any permutation $\sigma(x)$ of $S_m = \{1, 2, \cdots, m\}$. We can rewrite (1) as

$$F(x_1, x_2, \cdots, x_m) = \textstyle\sum_{\mathbf{j} \in \mathcal{S}} a_{\mathbf{j}} \sigma_{\mathbf{j}} \tag{2}$$

where $\mathbf{j} = (j_1, \cdots, j_m)$ with $0 \le j_1 \le j_2 \le \cdots \le j_m, j_i \in Z_{k+1}$, $perm(\mathbf{j})$ is the set consisting of all permutations on $\mathbf{j}$, $\mathcal{S} = \{\mathbf{j} = (j_1, \cdots, j_m) \mid 0 \le j_1 \le \cdots \le j_m \le k\}$, $\mathbf{a_j} = a_{j_1 \cdots j_m}$ and

$$\sigma_{\mathbf{j}} = \sum_{(j_1', \cdots, j_m') \in perm(\mathbf{j})} x_1^{j_1'} \cdots x_m^{j_m'}. \tag{3}$$

The cardinality of S, denoted as $T(k, m)$, is equal to the number of choosing m objects from a set with $k + 1$ distinct elements with replacements, i.e.,

$$T(k, m) = |\mathcal{S}| = \binom{m + k}{k}. \tag{4}$$

This is equal to the number of coefficients of a symmetric polynomial in m variables with degree k. Thus any $T(k, m)$ random values from $\mathbb{F}_p$ determines a symmetric m-variate polynomial of degree k.

Property 1. Applying the Lagrange interpolation to $F(i, x_2, \cdots, x_m)$ for $i = 1, 2, \cdots, k+1$, the m-variate symmetric polynomial defined by (1) can be recovered by

$$F(x_1, x_2, \ldots, x_m) = \sum_{i=1}^{k+1} F(i, x_2, \cdots, x_m) \prod_{j=1, j\neq i}^{k+1} \frac{x_1 - j}{i - j}.$$

2.2 Non-interactive k-secure m-conference Key Schemes

We need some basic terms from information theory. Let X be a discrete-time discrete-valued random process with a sample space S. The *entropy* of X is defined as

$$H(X) = E[-\log_2 p_X(X)] = -\sum_{x \in S} p_X(x) \log_2 p_X(x) \tag{5}$$

where E is the expectation operator and $p_X(x)$ is the probability density function of X.

Let $\mathcal{P} = \{P_1, \cdots, P_n\}$ be a set of entities in the system, which are treated as random variables. Let I be an m-subset of S_n, denoted as $I = \{i_1, \cdots, i_m\} \subset S_n$. We use the notation $P_I = \{P_i \,|\, i \in I\}$ to represent an m-subset of entities in $\mathcal{P}$, and K_I, the common key shared by the m entities P_I. A system is said to be k-secure if any k users, pooling together their shares, have no information on keys that they should not know. Formally, we give this definition below.

Definition 1. *([3]) Let k, n, m be positive integers with $m \leq n$ and $k \leq n - m$. A non-interactive* k-secure m-conference key distribution scheme *for $\mathcal{P}$ is a scheme which satisfies the following properties.*

1. *For each m users can non-interactively compute the common key, i.e., for any m-subset $I = \{i_1, \cdots, i_m\}$ of S_n, $H(K_I \,|\, P_{i_1}) = \cdots = H(K_I \,|\, P_{i_t}) = 0$.*
2. *Any group of k entities have no information on a key that they are not entitled to know, i.e., for any k-subset $J = \{j_1, \cdots, j_k\}$ of S_n with $J \cap I = \emptyset$, $H(K_I \,|\, P_J\} = H(K_I)$.*

A non-interactive k-secure m-conference protocol, designed by Blundo *et al.* [3], is based on a symmetric m-variate polynomial of degree k, as shown in Scheme 1.

Scheme 1. m-conference key scheme constructed from m-variate symmetric polynomials of degree k [3]. The KDC randomly selects a symmetric m-variate polynomial of degree k, say $F(x_1, \cdots, x_m)$ and computes shares for user i: $F_i(x_2, \cdots, x_m) = F(i, x_2, \cdots, x_m)$, $i = 1, \cdots, n$. A conference key involving m entities $\{P_1, \cdots, P_m\}$ is computed as follows. User i uses his share to compute $K_i = F_i(1, \cdots, i-1, i+1, \cdots, m)$. Then $K_i = F(1, 2, \cdots, m), i = 1, \cdots, m$, which is the conference key shared by those m users.

Fact 1 *([3]) Scheme 1 is a k-secure m-conference key distribution scheme if all coefficients of the symmetric polynomial $F(x_1, \cdots, x_m)$ in m variables of degree k are uniformly selected in $\mathbb{F}_p$.*

However, there is one main problem to implement this k-secure m-conference protocol for $m > 2$. Since each share is a symmetric $(m-1)$-variate polynomial of degree k, according to (4), each user

needs to store $T(k, m-1)$ elements in $\mathbb{F}_p$. Thus the storage space is in the exponential complexity to both the size of the conference and the security level of k.

In the rest of the paper, first, we propose a special type of symmetric m-variate polynomial to design a k-secure m-conference protocol. The storage space of each user is used to store $k+1$ elements in $\mathbb{F}_p$, the number of the coefficients of an univariate polynomial of degree k which is independent of the size of the conference. Then we show that our scheme satisfies the condition of Fact 1 where $\mathbb{F}_p$ is replaced by $\mathbb{F}_p^*$, the multiplicative group of $\mathbb{F}_p$ and it resists to attacks from known shares and known conference keys.

3 Conference Key Establishment Protocol Using Univariate Polynomials

We keep the notations introduced in the last section. We assume that there is a key distribution center (KDC) at the initial stage, and there is a secure channel between the KDC and each user in the initial stage for distributing shares. KDC is not involved in the running phase of the key establishment among a group of entities.

3.1 *m*-conference Key Establishment

Domain public parameters: KDC selects a prime p, m and n with $m \leq n$ such that either $(m-1)^{-1}$ or m^{-1} modulo $p-1$ exists. For an m-subset of $\mathcal{P}$, for simplicity, we can assume that this group of m entities in $\mathcal{P}$ is indexed as $S_m = \{1, 2, \cdots, m\} \subset S_n$. Let

$$h = (m - 1 + \delta)^{-1} \text{ where } \delta = \begin{cases} 0 & \text{for } m \text{ even} \\ 1 & \text{for } m \text{ odd.} \end{cases} \tag{6}$$

Protocol 1. Non-interactive m-conference Key Establishment

Secret master key of KDC: KDC randomly selects a random number $b \in \mathbb{F}_p$ and a polynomial with degree k in $\mathbb{F}_p[x]$, $f(x) = a_k x^k + \cdots + a_1 x + a_0, a_i \in \mathbb{F}_p$.

Shares distribution: KDC computes a share for entity i as follows: $f_i(x) = bf(i)^h f(x) \in \mathbb{F}_p[x], i = 1, 2, \cdots, m$. Equivalently, the share for entity i is a $(k+1)$-dimensional vector in $\mathbb{F}_p^{k+1}$: $\mathbf{v}_i = (by_i a_0, by_i a_1, \cdots, by_i a_k)$, where $y_i = f(i)^h$. Then, the KDC sends share $f_i(x)$, i.e., $\mathbf{v}_i$ to P_i through a secure channel.

Conference key establishment: Let $I = \{i_1, \cdots, i_t\} \subset S_m = \{1, \cdots, m\}$ be a t-subset of S_m. For the group $\{P_{i_1}, \cdots, P_{i_t}\}$, each entity computes the conference key non-interactively using its share non-interactively, i.e., entity $P_i, i \in I$ computes:

$$K_i = \prod_{j \in I \setminus \{i\}} f_i(j) f_i(0)^{m-t+\delta}$$

where $I \setminus \{i\}$ is the set consisting of all the elements in I but i. The conference key is established in a non-interactive way, i.e., the key is generated by the individual entity.

Proposition 1. *The shared conference key is given by $K_i = K$ for all $i \in I$ where*

$$K = \begin{cases} b^{m-1} f(0)^{\delta} \prod_{j=1}^{m} f(j) & \text{for } t = m \\ \\ b^{m-1} f(0)^{m-t+\delta} \prod_{j \in I} f(j) & \text{for } t < m. \end{cases}$$

3.2 Performance Evalution

From Protocol 1, each entity only needs to store the $k+1$ coefficients of a polynomial with degree k. There is no overhead to exchange information in order to establish a conference key. The computational effort to establish a conference key with group size $t \leq m$ is to evaluate its secret polynomial $t-1$ times. For $m > 2$, the storage of each share is reduced from $T(k, m-1)$ elements in $\mathbb{F}_p$, which is the exponential complexity to both the conference size m and the security level k, to $k+1$ elements in $\mathbb{F}_p$ which is independent of the conference size m. The computation for a conference key is reduced from evaluating an $(m-1)$-variate symmetric polynomial of degree k over $\mathbb{F}_p$ to evaluating an univariate polynomial of degree k over $\mathbb{F}_p$ m times. For the detailed evaluation, see Appendix A. The trade-off among the sizes for the parameters of p, m, and k can be determined according to the security level and conference sizes in different applications.

4 The *k*-secure Property

Since the analysis for both m even and m odd are similar, from now on, we only give the analysis for m even case unless otherwise specified. We first present general properties of multivariate polynomials related to conference keys. We then show that Protocol 1 is a k-secure m-conference key distribution scheme. We keep the notations about a symmetric polynomials in m variables introduced in Section 2.

From the definition of Protocol 1, we have the following result.

Proposition 2. *Any conference key of t entities is computed from the following t-variate polynomial where $2 \leq t \leq m$*

$$F_t(x_1, x_2, \ldots, x_t) = b^{m-1} f(0)^{m-t} f(x_1) f(x_2) \cdots f(x_t). \tag{7}$$

In particular, if $t = m$, the conference key of all m entities $\{P_1, \cdots, P_m\}$ is computed from the following m-variate polynomial

$$F_m(x_1, x_2, \ldots, x_m) = b^{m-1} f(x_1) f(x_2) \cdots f(x_m). \tag{8}$$

We denote F_m by F for simplicity if the context is clear. From Fact 1 in Section 2, we have the following theorem immediately.

Theorem 1. *Protocol 1 is a k-secure m-conference key distribution scheme if all coefficients of the symmetric polynomial $F(x_1, \cdots, x_m)$, given by (8), in m variables of degree k are uniformly distributed in $\mathbb{F}_p^*$.*

From Theorem 1, Protocol 1 is a k-secure m-conference key distribution scheme if any coefficient $a_{\mathbf{j}}$ of $F(x_1, \cdots, x_m)$, given in (8), treated as a random variable, satisfies

$$P\{a_{\mathbf{j}} = c\} = \frac{1}{p-1} \text{ for any } c \in \mathbb{F}_p^* \text{ and } \mathbf{j} \in \mathcal{S}. \tag{9}$$

In the following, we shall prove that the coefficients in $F(x_1, \cdots, x_m)$ satisfy (9) if all variables in the set $\{b, a_0, \cdots, a_k\}$ are uniformly distributed in $\mathbb{F}_p^*$. In order to do so, we need the following two lemmas.

Lemma 1. *Let x and y be two random variables with uniform distribution in $\mathbb{F}_p^*$. Then the following random variables are uniformly distributed in $\mathbb{F}_p^*$.*

1. *$s = x^{m-1}$ when m is even and $m-1$ is relatively coprime with $p-1$.*
2. *$u = x^m y$ where m is odd and relatively coprime with $p-1$.*
3. *$v = xy^r$ where $\gcd(r, p-1) > 1$.*

Proof. For Case 1, since $m-1$ is relatively coprime with $p-1$, s, as a function of x, is a permutation in $\mathbb{F}_p^*$. Thus, s is uniformly distributed in $\mathbb{F}_p^*$ when x is uniformly distributed in $\mathbb{F}_p^*$. For Case 2, using the result of Case 1, $r = x^m$ is a random variable with uniform distribution in $\mathbb{F}_p^*$. Thus, when $(r, y) \in \mathbb{F}_p^* \times \mathbb{F}_p^*$, $u = ry$ takes $p-1$ times each value in $\mathbb{F}_p^*$. Consequently,

$$P\{u = c\} = \frac{p-1}{(p-1)^2} = \frac{1}{p-1}, \ \forall c \in \mathbb{F}_p^*.$$

The assertion follows. For Case 3, let N be the order of r in $\mathbb{F}_p$, then y^r only takes N elements in $\mathbb{F}_p^*$ with repetition $\frac{p-1}{N}$. Note that $\{xc \,|\, x \in \mathbb{F}_p^*\} = \mathbb{F}_p^*$ for each $c \in \mathbb{F}_p^*$. Thus the assertion follows. □

Lemma 2. *If all random variables in $\{b, a_0, a_1, \cdots, a_k\}$ are uniformly distributed in $\mathbb{F}_p^*$, then any coefficient of $F(x_1, \cdots, x_m)$ is a random variable with uniform distribution in $\mathbb{F}_p^*$.*

Proof. By expanding (8) (including m odd) in Proposition 2 and using (2) in Section 2, we have

$$F(x_1, x_2, \cdots, x_m) = b^{m-1+\delta} a_0^{\delta} \sum_{\mathbf{j} \in \mathcal{S}} a_{\mathbf{j}} \sigma_{\mathbf{j}} \tag{10}$$

where $a_{\mathbf{j}}$ is a function of $\{a_0, a_1, \cdots, a_k\}$. We define

$$g_{\mathbf{j}}(b, a_0, \cdots, a_k) = b^{m-1+\delta} a_0^{\delta} a_{j_1} \cdots a_{j_m}, 0 \le j_1 \le j_2 \le \cdots \le j_m \le k. \tag{11}$$

Then we have

$$g_{\mathbf{j}}(b, a_0, \cdots, a_k) = b^{m-1+\delta} a_0^{\delta+t_0} a_1^{t_1} \cdots a_k^{t_k} \tag{12}$$

where $0 \le t_j \le m$, determined by $\mathbf{j}$ as follows

$$t_s = |\{i \mid j_i = s, 0 \le i \le m\}|, s = 0, 1, \cdots, k. \tag{13}$$

Thus, $g_{\mathbf{j}}(b, a_0, \cdots, a_k)$ is a function of $\mathbb{F}_p^*$ in $k+2$ random variables in $\mathbb{F}_p^*$. Repeatedly using Lemma 1, we obtain that $g_{\mathbf{j}}(b, a_0, \cdots, a_k)$ is a random variable which is uniformly distributed in $\mathbb{F}_p^*$ when all random variables in $\{b, a_0, \cdots, a_k\}$ are uniformly distributed in $\mathbb{F}_p^*$. □

Theorem 2. *Protocol 1 is a k-secure m-conference key distribution scheme if b and all the coefficients of $f(x)$, i.e., $\{b, a_0, \cdots, a_k\}$, are uniformly distributed in $\mathbb{F}_p^*$.*

Proof. From Lemma 2, any coefficients of $F(x_1, \cdots, x_n)$, given by (10) are uniformly distributed in $\mathbb{F}_p^*$. According to Theorem 1, the assertion is true. □

The condition for our m-conference key distribution schemes being k-secure is somehow like a random source analogue to affine source [7]. We conclude this section by introducing a concept of a homomorphic source as follows. Let $m < p-2$ and $\mathcal{G} \subset Z_{m+1}^r$.

Definition 2. *A distribution Y is called an $(r, \mathcal{G})$-*homomorphic source *in $\mathbb{F}_p^*$ if for any r independently sampled values $y_1, \cdots, y_r \in \mathbb{F}_p^*$ of Y with uniform distribution in $\mathbb{F}_p^*$, a monomial term $y_1^{t_1} \cdots y_r^{t_r}$ is an uniform random variable in $\mathbb{F}_p^*$ for any $(t_1, \cdots, t_r) \in \mathcal{G}$.*

Under this definition, the following result follows immediately.

Theorem 3. *Protocol 1 is a k-secure m-conference key distribution scheme if $\{b, a_0, \cdots, a_k\}$ are sampled values of a $(k+2, \mathcal{G})$-homomorphic source in $\mathbb{F}_p^*$ where $\mathcal{G}$ consists of all vectors $(m-1+\delta, t_0, \cdots, t_k) \in Z_{m+1}^{k+2}$ where $(t_0, \cdots, t_k)$ is determined by (13).*

Note that $\mathcal{G}$ has the same cardinality as that of $\mathcal{S}$.

5 Security Analysis

In the previous section, we have shown that our scheme satisfies the k-secure m-conference as defined in Definition 1. In this section, we consider attacks from known shares and known conference keys and we use a generic term adversary $\mathcal{A}$ to represent the following two scenarios:

1. The adversary possesses a single share or multiple shares (insiders' attacks): (a) Entity i itself is malicious or entity i is captured by an attacker. This implies that $\mathcal{A}$ holds the share $f_i(x)$. (b) Either t entities collude together or the attacker captures t entities. This implies that the adversary has the shares of t entities.
2. The adversary possesses some conference keys, but no shares (outsiders' attacks).

5.1 Attacks on the Master Secret Key from A Single Share

Before we present the analysis of the attacks, we single out the following result about the shares, which is a direct consequence of the definition of Protocol 1.

Proposition 3. *Entity i can obtain the following polynomial from its share $f_i(x) = bf(i)^h f(x)$ (recall $h = (m-1)^{-1}$ for m even):*

$$F_t(i, x_2, \cdots, x_t) = b^{m-1} f(0)^{m-t} f(i) f(x_2) f(x_3) \cdots f(x_t), t = 2, \cdots, m \tag{14}$$

A. Effect of Random Value b. The random value b can be considered as a random masking for protecting the master secret key $f(x)$. Since entity i's share is $f_i(x) = bf(i)^{(m-1)^{-1}} f(x)$, from Property 3, the adversary can compute $f_i(i)^{m-1} = b^{m-1} f(i)^m$. However, since b is a random number, the adversary cannot separate $f(i)^m$ from $b^{m-1} f(i)^m$. The effect of random value b in the case that the adversary has multiple shares is the same, which will be discussed later.

B. Masking Random Vectors. Recall $y_i = f(i)^h$, $i = 1, 2, \cdots, m$ (here $f(i) \neq 0$) in Protocol 1. From the secret share $f_i(x)$, the adversary can form a system of $(k+1)$ cubic equations in $k+2$ unknowns $b, y_i, a_j, 0 \leq j \leq k$ as follows.

$$v_{i,j} = by_i a_j \text{ in } \mathbb{F}_p,\ 0 \leq j \leq k \Longrightarrow \mathbf{v}_i = c_i \mathbf{a} \tag{15}$$

where b, y_i and $a_j, 0 \leq j \leq k$ are secret values of KDC, $c_i = by_i$ and $\mathbf{a} = (a_0, \cdots, a_k)$. (In general, it is not solvable in a polynomial time (see [8]).) The adversary knows the above relation, $\mathbf{v}_i$ and p and the adversary tries to recover c_i and $\mathbf{a}$ using Shamirs attack on Knapsack public-key crypto scheme [18] or the algorithm for solving the vector with the shortest length in a lattice [13] [12][5]. However, $\{a_0, \cdots, a_k\}$ is an identical independent distributed random sequence with uniform distribution in $\mathbb{F}_p^*$, which may not satisfy the super-increasing condition. Thus those methods are not applicable to this case.

C. Master Key's Entropy for Individual Entity. From (15), the adversary can obtain the ratios $r_j = \frac{v_{i,j}}{v_{i,j-1}} = \frac{a_j}{a_{j-1}},\ j = 1, 2, \cdots, k$. The ratios $\{r_1, \cdots, r_k\}$ are known to adversary. Thus, recovering $Y = (b, a_0, \cdots, a_k)$ from $\{r_i\}_{i=1}^k$, the adversary only needs to guess a_0 with successful probability $1/(p-1)$ when a_0 is a random variable uniformly distributed in $\mathbb{F}_p^*$. Thus we have established the following results.

Theorem 4. *Adversary can compute the master secret key $Y = (b, a_0, \cdots, a_k)$ using the following iterative relation*

$$a_j = r_j a_{j-1},\ j = 1, \cdots, k, \tag{16}$$

where $r_j = \frac{v_{i,j}}{v_{i,j-1}}$ and $a_0 \in \mathbb{F}_p^*$ is a guessed initial value with successful probability $1/(p-1)$ and $v_{i,0} = bf(i)^h a_0$ for deriving b. Furthermore, the entropy of the master key for each entity is approximately equal to $\log p$ for moderate p. Precisely,

$$H(Y|\mathbf{v}_i) = H(a_0) \approx \log p, i = 1, \cdots, m.$$

Consequently, the entropy of the master key for each entity only has $\log(p-1)$ bits, although it needs to store $(k+1)\log(p-1)$ bits.

D. A System of Homogenous Linear Equations from Evaluation of Univariate Polynomials. Using entity i's share, the adversary computes $t_j = f_i(j)/f_i(j-1)$. Thus $t_j = x_j/x_{j-1}$ where $x_j = f(j)$ which yields the following linear equations on unknowns $(a_0, \cdots, a_k)$

$$x_j = t_j x_{j-1},\ \text{where } x_j = a_0 + a_1 j + \cdots + a_k j^i = k, j = 0, 1, \cdots. \tag{17}$$

We denote M as the coefficient matrix of the homogeneous linear system of equations (17). Since $(a_0, \cdots, a_k) \neq (0, \cdots, 0)$ is a solution to (17), then the rank of the coefficient matrix M, denoted as $rank(M)$, is less than $k+1$. Thus the probability of the valid solution to the unknowns $\{a_i\}$ is at most $1/p$. We summarize the above discussion into the following proposition.

Proposition 4. *With the above notations, the adversary can recover the master secret key $(b, a_0, \cdots, a_k)$ of KDC by solving the system of the homogeneous equations given by (17) with a successful probability $1/p^r \leq 1/p$ where $r = k + 1 - rank(M)$ and $rank(M) < k + 1$.*

5.2 Attacks from Multiple Shares

In this subsection, we discuss the case that the adversary has multiple shares, i.e., either t entities collude together or the attacker captures the shares of t entities. Thus, the adversary holds multiple shares, say t shares, $f_i(x), i \in I = \{i_1, \cdots, i_t\} \subset S_m$. We classify the attacks into two cases. One is that the adversary attempts to recover unknown conference keys and the unknown shares, i.e., the shares $f_i(x), i \notin I$ from the known shares that it possesses. The other aims to recover the master key of the KDC.

A. Attacks on Conference Keys and Other Shares from Known Multiple Shares. If either $m > k$ and $t \leq k$ or $m \leq k$, so that $t \leq k$, then no conference keys of involving entities in $S_m \setminus I$ and no secret shares of the entities in $S_m \setminus I$ can be recovered from known t secret shares of the entities. For the remaining case $k < m$ and $t > k$, we have the following result.

Proposition 5. *For $k \leq m$, if a set of shares $\{f_j(x) \mid j \in I\}$ where $I \subset S_m$ with $|I| > k$ is known to the adversary, then $F(x_1, \cdots, x_m)$ can be reconstructed, so that the shares of all entities $P_i, i \in S_m$ can be reconstructed.*

Proof. Applying Proposition 3 in Section 4, from the share of entity $i \in I$, we can obtain

$$F_{m-1}(i, x_2, \cdots, x_m) = b^{m-1} f(i) f(x_2) \cdots f(x_m), i \in I.$$

Using Property 1 in Section 2, $F(x_1, \cdots, x_m) = b^{m-1} \prod_{j=1}^{m} f(x_j)$ can be constructed. Inversely, from $F(x_1, \cdots, x_m)$, we have

$$F(i, \cdots, i, j) = b^{m-1} f(i)^{m-1} f(j) \Longrightarrow b f(j)^h f(i) = F(i, \cdots, i, j)^h.$$

Again, applying the Lagrange interpolation to $b f(j)^h f(i), i = 1, \cdots, k$, we obtain $f_j(x) = b f(j)^h f(x)$ for all $j \in S_m$. □

B. Attacks on the Master Secret Key from Known Multiple Shares by Linearization

Theorem 5. *Assume that $\{b, f(x)\}$ are sample values of a $(k+2, \mathcal{G})$-homomorphic source in $\mathbb{F}_p^*$. Then by knowing $F(x_1, \cdots, x_m)$, the successful probability of recovering the master key $\{b, f(x)\}$ is $\frac{1}{p-1}$ which is negligible for a moderate p.*

Proof. According to Proposition 5, if the adversary obtains t shares with $t > k$, then he can reconstruct the m-variate polynomial F, and all shares $f_j(x) = bf(j)^h f(x)$. The adversary could try to solve for $Y = (b, a_0, \cdots, a_k)$ using linearization to obtain the coefficients of $F(x_1, \cdots, x_m)$. In details, using the proof in Lemma 2 in Section 4, we have

$$F(x_1, x_2, \cdots, x_m) = b^{m-1} \sum_{\mathbf{j} \in \mathcal{S}} a_{\mathbf{j}} \sigma_{\mathbf{j}} \tag{18}$$

where

$$a_{\mathbf{j}} = a_{j_1} a_{j_2} \cdots a_{j_m} = a_0^{t_0} a_1^{t_1} \cdots a_k^{t_k}, \text{ for some } 0 \leq t_j \leq m.$$

By linearization of (18), i.e., treat each coefficient $b^{m-1} a_{\mathbf{j}}$ as a new variable, one can solve a system of linear equations with the following unknowns

$$g_{\mathbf{j}} = b^{m-1} z_{\mathbf{j}} \ \ \forall \mathbf{j} = (j_1, j_2, \cdots, j_m) \in \mathcal{S}. \tag{19}$$

where

$$z_{\mathbf{j}} = a_{j_1} a_{j_2} \cdots a_{j_m}$$

is a new variable. Thus, attacker can obtain $\{g_{\mathbf{j}}\}_{\forall \mathbf{j} \in \mathcal{S}}$. However, it cannot remove b^{m-1}. Applying Theorem 3 to this case where the dimension $k+1$ is replaced by $|\mathcal{S}|$, the initial value for recovering the sequence $\{a_{\mathbf{j}}\}$ and b is a_0. Thus, the successful probability is $\frac{1}{p-1}$. □

This result is remarkable. If a set of shares $\{f_j(x) \,|\, j \in I\}$ where $I \subset S_m$ with $|I| > k$ is known to the adversary, then the m-conference key polynomial $F(x_1, \cdots, x_m)$ is known to the adversary. However, the successful probability of recovering master secret key $\{b, f(x)\}$ is negligible when p is moderate large. This is different from Blundo *et al.*'s scheme in which $F(x_1, \cdots, x_m)$ is the master secret key. For Blundo *et al.*'s scheme, if $F(x_1, \cdots, x_m)$ is known to the adversary, then all the shares and the master secret key can be recovered. Another impact of Theorem 5 is that the entropy of the master secret key is independent of the number of the shares compromised. Thus the adversary cannot generate shares for the entities P_i where $i \notin S_m$, even the adversary knows more than k shares. However, in this case, the adversary has all shares of the entities in $P_i, i \in S_m$ and all conference keys of those entities.

5.3 Attacks from Known Multiple Conference Keys

We now assume that the adversary has no success to compromise any entities. In other words, the adversary does not possess any shares. However, he may capture some conference keys. Using those captured conference keys and Lagrange interpolation, he may recover the rest of conference keys. The following two theorems provide the answers to those concerns and their proofs are given in Appendix B.

Theorem 6. *If $m \leq k$, then no additional conference keys can be recovered from known conference keys. If $m > k$, then some additional keys can be recovered from a known conference key set for which the Lagrange interpolation is applicable.*

Theorem 7. *No secret shares can be reconstructed from known conference keys.*

6 Conclusions

We proposed a non-interactive k-secure m-conference key distribution scheme constructed from an univariate polynomial over $\mathbb{F}_p$ with degree k, say $f(x)$. The KDC holds the master key $\{b, f(x)\}$ where $b \in \mathbb{F}_p$ and $f(x) \in \mathbb{F}_p[x]$. Each entity has a secret share $f_i(x) = bf(i)^h f(x)$ for each $i \in S_m = \{1, \cdots, m\}$. Any t-subset of m entities with index $I \subset S_m$ can share a t-conference key for which entity i in this group computes the m-conference key non-interactively as $K = \prod_{j \in I \setminus i} f_i(j) f_i(0)^{m-t+\delta}$. Compared with Blundo *et al.*'s scheme, the storage of each share in our scheme is reduced from $T(k, m-1)$ elements in $\mathbb{F}_p$, which is the exponential complexity in terms of both the conference size m and the security level k, to the complexity of $k+1$ elements in $\mathbb{F}_p$ which is independent of the conference size m. The computation for a conference key is reduced from evaluating an m-variate symmetric polynomial of degree k over $\mathbb{F}_p$ to evaluating an univariate polynomial of degree k over $\mathbb{F}_p$ m times.

We have showed that if b and the coefficients of $f(x)$ are uniformly selected from $\mathbb{F}_p^*$, then all the coefficients of an m-variate symmetric polynomial of degree k, given by $F(x_1, \cdots, x_m) = b^{m-1+\delta} f(0)^\delta f(x_1) \cdots f(x_m)$ are uniformly distributed in $\mathbb{F}_p^*$. This result yields the k-secure property of the proposed m-conference key distribution scheme with storage $(k+1)\log p$ bits instead of $T(k, m)\log p$ bits. Each share functions like a homomorphic source for generating uniformly distributed random coefficients in $\mathbb{F}_p^*$ of the m-variate symmetric polynomial F.

The scheme is resistant to the known shares and known conference keys attacks. If the adversary holds at least one share, then the successful probability of recovering the master key $\{b, f(x)\}$ is at most $1/p$ which is negligible for a moderate size p. This result is independent of the number of the shares known to the adversary, which is the property that Blundo *et al.*'s scheme does not have. In other words, if the adversary holds more than k shares, then the adversary can recover all the conference keys and all the unknown shares, but not the master secret key $\{b, f(x)\}$. No secret shares can be reconstructed from known conference keys. If $m > k$, then some additional conference keys can be recovered from a known conference key set for which the Lagrange interpolation is applicable.

References

1. IEEE 802.11i 2004. Amendment 6: medium access control (mac) security enhancements, 2004.
2. R. Blom. An optimal class of symmetric key generation systems. In *Advances in Cryptology Eurocrypt 84, LNCS 209, Springer-Verlag*, pages 335–338, 1993.
3. C. Blundo, A. De Santis, A. Herzberg, U. Vaccaro S. Kutten, and M. Yung. Perfectly-secure key distribution for dynamic conferences. In *Advances in Cryptology Crypto 92, LNCS 740, Springer-Verlag*, pages 471 – 486, 1993.
4. Y. Cheng and Y. Agrawal. A improved key distribution mechanism for large-scale hierarchical wireless sensor networks. *Journal of Ad Hoc Networks*, 5(1):35–48, 2007.
5. M. J. Coster, B. A. LaMacchia, A. M. Odlyzko, and C. P. Schnorr. An improved low-density subset sum algorithm. In *Advances in Cryptology EUROCRYPT '91, LNCS, Springer-Verlag*, pages 54–67, 1992.
6. A. Fanian, M. Berenjkou, H. Saidi, and T. A. Gulliver. An efficient symmetric polynomial-based key establishment protocol for wireless sensor networks. . *The ISC Int'l Journal of Information Security*, 2(2):89–105, 2010.
7. A. Gabizon and R. Raz. Deterministic extractors for affine sources over large fields. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '05, pages 407–418. IEEE Computer Society, 2005.

8. M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, 1979.
9. D. Gorden. A survey of fast exponentiation methods. *Journal of Algorithms*, 27(1):129–146, 1998.
10. L. Harn and C. Lin. Authenticated group key transfer protocol based on secret sharing. *IEEE Trans. on Computers*, 59(6):842–846, 2010.
11. D.E. Knuth. *The Art of Computer Programming, Semi-numerical Algorithms, vol. II*. AddisonWesley, Reading Massuchussetts, 1981.
12. J. C. Lagarias and A. M. Odlyzko. Solving low-density subset sum problems. *Journal of the ACM*, 32(1):229–246, 1985.
13. A. K. Lenstra, H. W. Lenstra, and L Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.
14. H. Liang and C. Wang. An energy efficient dynamic key management scheme based on polynomial and cluster in wireless sensor netwoks. *Journal of Convergence Information Technology*, 6(5):321–328, 2011.
15. D.G. Liu and P. Ning. Establishing pairwise keys in distributed sensor networks. In *ACM CCS*, pages 52–61, 2003.
16. D.G. Liu, P. Ning, and R.F. Li. Establishing pairwise keys in distributed sensor networks. *ACM Transactions on Information and System Security*, 8(1):41–77, 2005.
17. N. Saxena, G. Tsudik, and J. H. Yi. Efficient node admission and certificateless secure communication in short-lived manets. *IEEE Trans. on Parallel and Distributed Systems*, 20(2):158–170, 2009.
18. A. Shamir. A polynomial-time algorithm for breaking the basic merkle-hellman cryptosystem. In *In Proceedings of the 23rd IEEE Symposium on Foundations of Computer Science*, pages 145 – 152. IEEE, 1982.
19. G. Song and Q. Zhuhong. A compromise-resilient group rekeying scheme for hierarchical wireless sensor networks. In *the Proceedings of Wireless Communications and Networking Conference (WCNC)*, pages 1–6, 2010.
20. Y. Zhou and Y. Fang. A two-layer key establishment scheme for wireless sensor networks. *IEEE Trans. on Mobile Computing*, 6(9):1009–1020, 2007.

Appendix A. Performance Evaluation

Each entity needs to store the $k+1$ coefficients of a polynomial with degree k. There is no overhead to exchange information in order to establish a conference key. The computational effort to establish a conference key with group size $t \leq m$ is to evaluate its secret polynomial $t-1$ times. Recall that entity i's secret share is given by

$$f_i(x) = bf(i)^h f(x) = \sum_{j=0}^{k} v_j x^j \quad \text{where } h = \begin{cases} (m-1)^{-1} & m \text{ even} \\ m^{-1} & m \text{ odd} \end{cases} \tag{20}$$

where we omit the subscript i in $v_{i,j}$ and shortened as v_j since the context is clear in this section. There are many algorithms proposed in the literature for evaluating a polynomial over $\mathbb{F}_p$. Here we present a standard one which is shown in Algorithm 1 (using Horner's rule [11]) as an example to show the cost for computing groups keys. The computation of conference key is presented in Algorithm 2 for m even. The case for m odd is similar. Thus all the following analytic results are only given to the case of m even.

Note that each entity has its secret share $f_i(x)$ given in (20). Thus, entity i needs to store the vector

$$(v_0, v_1, \cdots, v_k), v_j \in \mathbb{F}_p.$$

Thus the memory cost is $(k+1)\log p$ bits where $\log(x)$ is the base 2 throughout this paper. From Algorithm 1, evaluating a polynomial of degree k needs k multiplications and $k+1$ additions in $\mathbb{F}_p$.

Algorithm 1 *Procedure*$(f_i(x), u)$ for evaluating a polynomial $f_i(x)$ at $u \in \mathbb{F}_p$

Input: $f_i(x) = v_0 + v_1 x + \cdots + v_k x^k, v_j \in \mathbb{F}_p$, and $u \in \mathbb{F}_p$
Output: $f_i(u)$
1: $t_{-1} = 0$
2: **for** $j = 0$ to $k-1$ **do**
3: $\quad t_j = (t_{j-1} + v_{k-j})u$
4: $t = t_{k-1} + v_0$
5: **return** t

Algorithm 2 Procedure for entity i to compute conference key with size t

Input: $f_i(x) = v_0 + v_1 x + \cdots + v_k x^k, v_j \in \mathbb{F}_p$, and $I = \{i_1, \cdots, i_t\} \subset S_m$ and $i \in I$
Output: $K = \prod_{j \in I \setminus \{i\}} f_i(j) \cdot v_0^{m-t}$
1: $K = 1$
2: **for** $j \in I \setminus \{i\}$ **do**
3: $\quad K \leftarrow K \cdot Procedure(f_i(x), j)$
4: $K \leftarrow K v_0^{m-t}$ (it needs another loop to compute v_0^{m-t} which we omit here)
5: **return** K

From Algorithm 2, the computation cost to establish a conference key with group size t consists of the cost in Steps 3 and 4. For Step 3, there are $t-1$ evaluations of the polynomial $f_i(x)$ which needs $(t-1)k$ multiplications and $(t-1)(k+1)$ additions in $\mathbb{F}_p$. For Step 4, evaluating an element in $\mathbb{F}_p$ with power $m-t$ needs $\log(m-t)$ multiplications in $\mathbb{F}_p$ $(t \leq m)$ (see [9]). Thus, Step 4 needs $1 + \log_2(m-t)$ multiplications in $\mathbb{F}_p$. We summarize those results in the following property.

Property 2. For each user,

(a) the storage, denoted as C_{mem} in bits is $C_{mem} = (k+1)\log p$ bits, and
(b) the computation cost to establish a conference key with group size $t \leq m$ is $(t-1)k + 1 + \log_2(m-t)$ multiplications and $(t-1)(k+1)$ additions in $\mathbb{F}_p$.

Appendix B. Attacks from Known Multiple Conference Keys

In order to prove Theorem 6, we need the following treatments. Using Property 2 in Section 5, there are conference keys with size 2, 3, and up to m. The number of all those conference keys is given by

$$\sum_{t=2}^{m} \binom{m}{t} = 2^m - m - 1$$

which are given by

$$G_t = \{F_t(j_1, \cdots, j_t) \,|\, \{j_1, \cdots, j_t\} \subset S_m\},\ t = 2, \cdots, m,$$

i.e., G_t consists of all conference keys with group size t.

Definition 3. *For a fixed $2 \leq t \leq m$, let $H \subset G_t$ and $I \subset S_m$. Then H is said to be a* $(t,1)$-principle conference key set *if for a fixed vector $(j_1, \cdots, j_{t-1}) \in \mathbb{F}_p^{t-1}$,*

$$\{b^{m-1} f(0)^{m-t} f(j_1) \cdots f(j_{t-1}) f(i) \,|\, i \in I\} \subset H$$

with $|I| > k$.

Proposition 6. *If H is a $(t,1)$-principle conference key set, then the adversary can recover the polynomial*

$$b^{m-1} f(0)^{m-t} f(j_1) \cdots f(j_{t-1}) f(x). \tag{21}$$

Proof. According to Property 3, if H is a $(t,1)$-principle conference key set, then using the Lagrange interpolation to the points

$$(i, F_t(j_1, \cdots, j_{t-1}, i)), i \in I, |I| > k,$$

the polynomial given by (21) can be recovered.

□

Now we extend Definition 3 to a general case.

Definition 4. *For fixed t and $r : 1 \leq r \leq t$, $I_j \in S_m, j = 1, \cdots, r$ and $(j_1, \cdots, j_{t-r}) \in \mathbb{F}_p^{t-r}$, then H is said to be a* (t,r)-principle conference key set *if*

$$\{b^{m-1} f(0)^{m-t} f(j_1) \cdots f(j_{t-r}) f(j_{t-r+1}) \cdots f(j_t) \,|\, j_{t-r+1} \in I_1, \cdots, j_t \in I_r\} \subset H.$$

with $|I_j| > k$ for all j.

Proposition 7. *If H is a (t,r)-principle conference key set, then the adversary can recover the polynomial*

$$b^{m-1} f(0)^{m-t} f(j_1) \cdots f(j_{t-r}) f(x_1) \cdots f(x_r). \tag{22}$$

In particular, if $r = t$, then the adversary can recover the polynomial

$$F_t(x_1, \cdots, x_t) = b^{m-1} f(0)^{m-t} f(x_1) \cdots f(x_t). \tag{23}$$

Proof. If H is a (t,r)-principle conference key set, then H is a $(t,1)$-principle conference key set. Thus, the polynomial in (22) can be recovered step by step.

□

Proof of Theorem 6. From Proposition 7, if $m \leq k$, then no additional conference keys can be recovered from any known conference keys. If $m > k$, then some additional keys can be recovered from a principle conference key set as follows.

1. If the adversary captures a $(t,1)$-principle conference key set, then it can recover all the following conference keys with group size t

   $$b^{m-1} f(0)^{m-t} f(j_1) \cdots f(j_{t-1}) f(i), \ \ \forall i \in S_m.$$

   for all subsets $\{j_1, \cdots, j_{t-1}\} \subset S_m$ which have keys in H.

2. If the adversary captures a (t, r)-principle conference key set, then it can recover all conference keys in G_t in which all subsets $\{j_1, \cdots, j_{t-r}\} \subset S_m$ have keys in H.
3. If $r = t$, then the adversary can recover G_t, the set of all conference keys with group size t.

Thus the assertion of Theorem 6 is established. □

Theorem 7 claims that it is not possible to recover any secret shares of entities from known conference keys. We first need the following result.

Proposition 8. *If $m \leq k$, then from any conference key set, polynomials $F_t(i, x_1, \cdots, x_t), t = 2, \cdots, m$, $i \in S_m$ cannot be recovered. If $m > k$, then from a (t, t)-principle conference key set, only the following polynomials*

$$b^{m-1} f(0)^{m-t} f(x_1) \cdots f(x_t), t = 2, \cdots, m - k$$

can be constructed.

Proof of Theorem 7. According to Proposition 8, $F(x_1, \cdots, x_m)$ cannot be constructed from any conference key set. Applying Proposition 5 in Section 5, the share of each entity cannot be reconstructed from a conference key set. Thus, no secret shares can be reconstructed from any known conference keys. □