Mobile Netw Appl
DOI 10.1007/s11036-016-0776-7

Centralized Group Key Establishment Protocol without a Mutually Trusted Third Party

Lein Harn[1] · **Ching-Fang Hsu**[2] · **Bohan Li**[3]

Abstract The type of centralized group key establishment protocols is the most commonly used one due to its efficiency in computation and communication. A key generation center (KGC) in this type of protocols acts as a server to register users initially. Since the KGC selects a group key for group communication, all users must trust the KGC. Needing a mutually trusted KGC can cause problem in some applications. For example, users in a social network cannot trust the network server to select a group key for a secure group communication. In this paper, we remove the need of a mutually trusted KGC by assuming that each user only trusts himself. During registration, each user acts as a KGC to register other users and issue sub-shares to other users. From the secret sharing homomorphism, all sub-shares of each user can be combined into a master share. The master share enables a pairwise shared key between any pair of users. A verification of master shares enables all users to verify their master shares are generated consistently without revealing the master shares. In a group communication, the initiator can become the server to select a group key and distribute it to each other user over a pairwise shared channel. Our design is unique since the storage of each user is minimal, the verification of master shares is efficient and the group key distribution is centralized. There are public-key based group key establishment protocols without a trusted third party. However, these protocols can only establish a single group key. Our protocol is a non-public-key solution and can establish multiple group keys which is computationally efficient.

Keywords Group key establishment · Centralized server · Key generation center · Mutually trusted server · Secret sharing homomorphism · Bivariate polynomial

Lein Harn and ChingFang Hsu contributed equally to this work.

✉ Ching-Fang Hsu
cherryjingfang@gmail.com

Lein Harn
harnl@umkc.edu

[1] Department of Computer Science Electrical Engineering, University of Missouri-Kansas City, Kansas City 64110, MO, USA

[2] Computer School, Central China Normal University, 430079 Wuhan, China

[3] School of Optical and Electronic Information, Huazhong University of Science and Technology, 430074 Wuhan, China

1 Introduction

Communication has been developed from traditional one-to-one communication into many-to-many communication, also called the *group communication*. In a secure group communication, a secret group key needs to be established and shared among all group members. Employing this group key on exchange messages in a group communication can provide security services such as message confidentiality and message authentication. There are two types of group key establishment protocols.

- Centralized group key establishment protocols in which a server is needed to select a group key and distribute the group key to all group members.
- Distributed group key establishment protocols in which there is no server and the group key is determined by all group members.

Most distributed group key establishment protocols are based on the generalization of the Diffie-Hellman (DH) public-key distribution scheme [1]. For example, the group

Published online: 21 November 2016

key establishment protocols [2–5] extended the DH scheme in a ring topology. Later, protocol [6] has been proposed with authentication services and has proved to be secure. In 2006, Bohli [7] developed a framework for robust group key agreement that provides security against malicious insiders and active adversaries in an unauthenticated point-to-point network. Harn et al. [8] proposed a group DH protocol based on the secret sharing scheme. The main disadvantage of the group DH key exchange is due to its computational and communication complexity because the group key is determined by all group users so each member needs to compute DH keys and exchange information to other users. Recently, Wu et al. [9] proposed a new approach which is a hybrid of group key agreement and public-key broadcast encryption. Their scheme is built from public-key based bilinear groups.

The centralized group key establishment protocols are the most widely used protocols. For example, the IEEE 802.11i standard [10] employs an online server to select a group key and transport the group key to each group user. The first centralized group key establishment protocol [11] using a (t, n) secret sharing scheme is proposed in 1989. Later, there are several papers [12–14] following the same concept to propose ways to distribute group messages to multiple users. In 2010, Harn et al. [15] proposed a secret sharing with RSA modulus to transmit a group key to all users.

The primary advantage of employing this type of protocols is its efficiency in both transmission and computation. However, these protocols need a mutually trusted server to register all users initially. During the registration, a pairwise secret key is shared between the server and each user. Later, in the event of requesting a secure group communication, the server acts as a key generation center (KGC) to select a group key first. Then, the KGC encrypts the group key using each pairwise shared key and transmits the encrypted group key to each group user separately. Since the group key is selected by the KGC, the group communication is not kept secrecy from the KGC. In other words, users in a secure group communication need to trust the KGC. Needing a mutually trusted KGC can cause problem in some applications. For example, users in an Internet chat-room want to hold a secure conference. Although the Internet chat-room provides a convenient communication platform to exchange information, but users cannot trust the service provider. Thus, a centralized group key establishment protocol without a mutually trusted KGC is very desirable.

To overcome the problem of a single trusted KGC, one straightforward alternative approach is to enable each user to act as a KGC to register other users. If we assume that each user only trusts himself, each user has to generate and store pairwise shared keys with other users. When the user wants to initiate a conference, the user can act as a KGC to select a group key and encrypt the group key for other users using pairwise shared keys. In addition to store pairwise share keys generated by himself, each user also needs to store pairwise shared keys issued by other users. Overall, each user needs to store $2(n-1)$ pairwise shared keys, where n is the number of users in the application. Key storage and key management become the main problems of this approach.

There are many public-key based group key establishment protocols without a trusted third party. The most commonly used public-key agreement protocol is Diffie-Hellman (DH) key exchange protocol [1]. In DH key exchange, the session key is determined by exchanging public keys of two communication entities. Since the public key itself does not provide any authentication, a digital signature of the public key needs to be attached to provide authentication. However, DH key exchange can only provide session key for two entities; not for a group more than two members. Most distributed group key management protocols, for example, Ingemarsson et al. [2] protocol, took natural generalization of the DH key agreement. In 2006, Bohli [16] developed a framework for robust group key agreement that provides security against malicious insiders and active adversaries in an unauthenticated point-to-point network. Also, in 2007, Katz and Yung [17] proposed the first constant-round and fully scalable group DH protocol which is provably secure in the standard model (i.e., without assuming the existence of "random oracles"). There are two major concerns of these public-key based group DH key management protocols, that are, (a) public-key computation takes more computational time, and (b) these protocols can only establish a single group key. Computational cost is one of the most important factors in Wireless and Ad Hoc Networks since most mobile devices have limited battery and computational power. To develop an efficient group key establishment protocol becomes an important issue. In this paper, we propose a novel group key establishment protocol which takes advantage of centralized group key distribution but without a mutually trusted KGC. In our proposed design, we assume that there has no mutually trusted KGC. Each user can act as a KGC to register other users and to issue pairwise shared keys. One unique feature of our design is that each user needs to store only one private share which is the t coefficients of a univariate polynomial, where $t-1$ is the degree of the polynomial. The private share enables each user to share a pairwise key with every other user. Thus, any user can be an initiator of a secure group communication to act as a KGC. In addition, this private share is only known to each user. The storage of the share is independent of the number of users in the application. Moreover, the master shares generated during the initial phase can be used to establish multiple group keys. Since our protocol is polynomial based scheme, it is more efficient than all public-key based protocols.

There is one additional problem before using these private shares to establish secure group communications. These private shares may be generated inconsistently by users or contain transmission errors. A verification of private shares is

needed before using them to establish secure group communications. In 1985, Chor et al. [18] presented the notion of verifiable secret sharing (VSS). In VSS, shareholders are able to verify that their shares are consistent without revealing their shares or the secret. There are vast research papers on the VSS in the literature. Based to security assumptions, we can classify VSSs into two different types, schemes that are computationally secure and unconditionally secure. For example, Feldman [19] and Pedersen [20] developed non-interactive VSSs based on cryptographic commitment schemes. The security of Feldman's VSS is based on the hardness of solving discrete logarithm, while the privacy of Pedersen's VSS is unconditionally secure and the correctness of the shares depends on a computational assumption. Benaloh [21] proposed an interactive VSS scheme and it is unconditionally secure. In 1996, Stadler [22] proposed the first publicly verifiable secret sharing (PVSS) scheme. A PVSS scheme allows each shareholder to verify the validity of all shares, including both shares of his/her own and other shareholders. However, in most non-interactive VSSs [2, 20], shareholders can only verify the validity of his/her own share; but not other shareholders' shares. In some PVSSs [22, 23], the verification algorithm involves interactive proofs of knowledge. These proofs are made non-interactive by means of the Fiat-Shamir technique [24]. Peng and Wang's PVSS [25] is based on linear code, Ruiz and Villar's PVSS [26] is based on Pailler's cryptosystem [27]. There are non-interactive PVSSs based on bilinear pairing [28, 29].

In this paper, we propose a verification scheme to enable users to verify that their private shares are generated consistently. The uniqueness in our proposed verification is that all users work together to verify their private shares. The outcome of the verification can be either (a) all private shares are generated consistently or (b) there are inconsistent private shares so new generation is needed. Since private shares are verified all at once, our verification is very efficient in terms of computation and communication.

We summarize our contributions below.

- A novel centralized group key establishment protocol without a mutually trusted KGC is proposed.
- The generation of master shares of users is based on bivariate polynomials. The secret sharing homomorphism is used to minimize the storage size of each master share.
- The verification of master shares is to verify master shares all at once so it is very efficient in communication and computation.

The organization of this paper is as follows. In Section 2, we review of bivariate polynomial and the secret sharing homomorphism, In Section 3, we present the model of our group key establishment protocol including protocol description, types of adversaries and security of verification. In Section 4, we present our group key establishment protocol. In Section 5, we present the analysis of our proposed protocol including functionalities, security analysis and comparison to other related protocols. We conclude in Section 6.

2 Preliminaries

2.1 Review of bivariate polynomial

A bivariate polynomial having degree $t-1$ can be represented as $F(x,y) = a_{0,0} + a_{1,0}x + a_{0,1}y + a_{1,1}xy + a_{2,0}x^2 + a_{0,2}y^2 + a_{1,2}xy^2 + a_{2,1}x^2y + a_{2,2}x^2y^2 + \ldots + a_{t-1,t-1}y^{t-1}y^{t-1} nod p$, where $a^{i,j} \in GF(p)$, $\forall i, j \in [0, t-1]$, and p is a prime modulus. A bivariate polynomial has been used in the design of a verifiable secret sharing scheme, we denote it as a BVSS. We can classify BVSSs into two types, the symmetric BVSSs, denote them as SBVSSs [30–33] and the asymmetric BVSSs, denote them as ABVSSs, [30, 31]. If the coefficients of the polynomial satisfy $a_{i,j} = a_{j,i}$, $\forall i$, $j \in [0, t-1]$, it is a symmetric bivariate polynomial. Shares generated by a bivariate polynomial can be used to establish pairwise keys between any pair of shareholders. In all (t, n) SBVSSs, a dealer selects a bivariate polynomial, $F(x, y)$, having degree $t-1$ and $F(0, 0) = s$, where s is the secret. The dealer generates shares, $F(x_i, y) \bmod p$, $i = 1, 2, \ldots, n$, for shareholders, where p is a prime with $p > s$, and x_i is the public information associated with each shareholder, U_i. Each share, $F(x_i, y)$, is a univariate polynomial having degree$t-1$. Since shares generated by a symmetric bivariate polynomial satisfy $F(x_i, x_j) = F(x_j, x_i)$, $\forall i, j \in [0, t-1]$, pairwise keys, such as $F(x_i, x_j) = F(x_j, x_i)$, can be established between any pair of shareholders, U_i and U_j..

2.2 Secret sharing homomorphism

Benaloh [21] introduced the property of the secret sharing homomorphism. Let S be the domain of the secret and T be the domain of shares corresponding to the secret. The function $F_I: T \rightarrow S$ is an induced function of the (t, n) SS. This function defines the secret s based on any subset containing t shares, $\{s_{i_1}, s_{i_2}, \ldots, s_{i_t}\}$, such that $s = F_I(s_{i_1}, s_{i_2}, \ldots, s_{i_t})$, where $I = \{s_{i_1}, s_{i_2}, \ldots, s_{i_t}\}$..

Definition 1: Homomorphism of the Secret Sharing **[21]** *Let $\oplus$ and $\otimes$ be two functions on elements in sets S and T, respectively. We say that a (t, n) SS has the $(\oplus, \otimes)$-homomorphic property if for any subset I and*

$s = F_I(s_{i_1}, s_{i_2}, ..., s_{i_t})$, $s' = F_I\left(s'_{i_1}, s'_{i_2}, ..., s'_{i_t}\right)$, *then* $s \oplus s' = F_I\left(s_{i_1} \otimes s'_{i_1}, s_{i_2} \otimes s'_{i_2}, ..., s_{i_t} \otimes s'_{i_t}\right)$.

We note that shares generated by Shamir's (t, n) SS scheme satisfy $(+,+)$-homomorphism property. In other words, the sum of shares of two polynomials, $f(x)$ and $g(x)$, is the share of additive polynomial, $f(x) + g(x)$.

3 Model of our group key establishment protocol

In this section, we describe the model of our group key establishment protocol including the system requirements, the adversary and security objectives.

3.1 Protocol description

Our proposed protocol consists four phases: (i) generation of master shares of users (GMS), (ii) verification of master shares (VMS), (iii) transmitting a group key (TGK) and (iv) recovering of the group key (RGK).

(i) GMS In this paper, we assume that each user does not trust other users but only trust himself. Therefore, each user needs to register other users. The same approach can be easily converted to applications where there are multiple KGCs to register users. Using multiple KGCs to register users is an alternative approach to overcome the problem of needing a mutually trusted KGC.

In this phase, every user acts as a registration center to register other users. During registration, the user selects a symmetric bivariate polynomial having degree $t-1$ and uses the polynomial to generate sub-shares for other users. Each sub-share is a univariate polynomial having degree $t-1$ and is sent to each other user, U_l, secretly. After receiving all sub-shares from other users, each user computes the master share based on the sub-secrets. The master share is a univariate polynomial having degree $t-1$.

(ii) VMS Master shares need to be verified by all users before using them to establish a secure group communication. The objective of the verification is to allow all users to verify that their master shares obtained are generated consistently by a symmetric bivariate polynomial having degree $t-1$ without revealing the secrecy of the polynomial and master shares. This objective is similar to most VSS schemes. However, we propose a different approach that all users work together to verify master shares all at once. Most existing VSSs verify shares one at a time. Thus, our verification is very efficient.

(iii) TGK In this phase, any user who initiates a conference, call the *initiator*, can act as a KGC to select a group key and use a pairwise shared key with each other user to encrypt the group key. Then transmit the ciphertext of the group key to each other user separately. The pairwise shared key is computed from the master share of the initiator.

(iv) RGK In this phase, any user in the group can use his/her master share to compute a pairwise shared key with the initiator, Then, use the pairwise shared key as the decryption key to decipher the ciphertext and obtain the group key.

3.2 Type of attacks

We consider two types of adversaries: insider and outsider.

Inside Attack The inside attacker is a legitimate user who own a master key determined by all users. But inside attacker may try to recover other user's master share. After obtaining other user's master share, the inside attacker is able to reveal other group keys that he/she is unauthorized to know or the attacker is able to impersonate other user in a secure group communication.

Outside Attack On the other hand, an outside attacker may try to recover the group key that he is unauthorized to know. This attack is related to the secrecy of group keys.

In the protocol analysis, we will show that none of these attacks can work properly against our protocol.

3.3 Security of vms

Since user does not trust other users and each sub-share may contain generation/transmission errors, master shares need to be verified by all users before using them to establish a secure group communication. In verification, there are two following security objectives.

(a) *Secrecy of the master shares*: The objective of verification is to ensure that all master shares are generated consistently form a symmetric bivariate polynomial. However, each master share should not be revealed to others in the process.

(b) *Secrecy of the polynomial used to generate master shares*: In a similar manner, the polynomial used to generate all master shares should not be revealed in the process.

4 Proposed protocol

We consider a set of n users, $U = \{U_i | U_i \in U, i = 1, 2, \ldots, n\}$ in a group communication.

4.1 GMS

Each user needs to register other users. During registration, each user does the following steps.

Step 1. Each user, U_i, selects a symmetric bivariate sub-polynomial having degree $t-1$ as $f_i(x,y) = a^i_{0,0} + a^i_{1,0}x + a^i_{0,1}y + a^i_{1,1}xy + a^i_{2.0}x^2 + a^i_{0,2}y^2 + a^i_{1,2}xy^2 + a^i_{2,1}x^2y + a^i_{2,2}x^2y^2 + \ldots + a^i_{t-1.t-1}y^{t-1}y^{t-1} nod\ p$, where $a^i_{j.k} \in GF(p)$ and $a^i_{j.k} = a^i_{k,j}, \forall j,k \in [0, t-1]$, and p is a prime.

Step 2. U_i, generates sub-shares, $f_i(x_l, y)$, $l = 1, 2, \ldots, n$, for all users, where x_l is the public information associated with user, U_l and $x_l \notin [1, n]$. Each sub-share, $f_i(x_l, y)$, is a univariate polynomial having degree $t-1$ and is sent to user, U_l, secretly.

Step 3. After receiving all sub-shares, $f_l(x_i, y)$, $l = 1, 2, \ldots, n$, from other users, U_i, computes the master share, $F_i(y)$, as $F_i(y) = \sum_{l=1}^{n} f_l(x_i, y) \bmod p$.

Theorem 1 *The master share, $F_i(y)$, obtained in Step 3 is a share of the polynomial, $F(x,y) = \sum_{l=1}^{n} f_l(x,y) \bmod p$.*

Proof In Step3, the master share, $F_i(y) = \sum_{l=1}^{n} f_l(x_i, y) \bmod p$, is the sum of sub-shares of polynomials, $f_l(x,y)$, $l = 1, 2, \ldots, n$. From secret sharing homomorphism as defined in ***Definition 1***, sub-shares generated by polynomials satisfy $(+,+)$-homomorphism property. In other words, the master share, $F_i(y)$, is a share of the polynomial, $F(x,y) = \sum_{l=1}^{n} f_l(x,y) \bmod p$. ∎

Theorem 2 *With master shares computed from Step 3, pairwise shared keys, $F_i(x_j) = F_j(x_i)$, can be established between any pair of users, U_i and U_j, $\forall i, j \in [1, n]$.*

Proof From Step 1, since each bivariate sub-polynomial, $f_i(x,y) = a^i_{0,0} + a^i_{1,0}x + a^i_{0,1}y + a^i_{1,1}xy + a^i_{2.0}x^2 + a^i_{0,2}y^2 + a^i_{1,2}xy^2 + a^i_{2,1}x^2y + a^i_{2,2}x^2y^2 + \ldots + a^i_{t-1.t-1}y^{t-1}y^{t-1} nod\ p$, is symmetric, the additive polynomial, $F(x,y) = \sum_{l=1}^{n} f_l(x,y) \bmod p$, is also a symmetric bivariate polynomial. It satisfies $f_i(x_j, x_k) = f_i(x_k, x_j)$. Thus, in Step 3, we can obtain $F_i(x_j) = \sum_{l=1}^{n} f_l(x_i, x_j)^j \bmod p = \sum_{l=1}^{n} f_l(x_j, x_i)^i \bmod p = F_j(x_i)$. ∎

Note that from ***Theorem 1*** , each master share is a share of the polynomial, $F(x,y) = \sum_{l=1}^{n} f_l(x,y) \bmod p$, which is determined by all users. If all users act honestly to generate and distribute sub-shares for other users, each master share is known only to an individual user. The following theorem proves that the proposed scheme can resist up to certain colluded users to recover the secret polynomial, $F(x,y)$.

Theorem 3 *The proposed GMS can resist up to $\lfloor \frac{t-1}{2} \rfloor$ colluded users to recover the secret polynomial $F(x,y)$.*

Proof $F(x,y)$ is a symmetric polynomial determined by all users which contains $\frac{t(t+1)}{2}$ different coefficients. In the proposed generation of master shares, each master share, $F_i(y)$, is a univariate polynomial with degree $t-1$. In other words, each user can use his master share to establish t linearly independent equations in terms of the coefficients of the polynomial $F(x,y)$. There are ht linearly independent equations if there are h colluded users with knowing h master shares. If the proposed scheme can resist up to h colluded users from recovering the secret polynomial, $F(x,y)$ it needs $\frac{t(t+1)}{2} > ht$ $(\Rightarrow t + 1 > 2h)$. ∎

4.2 VMS

The objective of our proposed VMS is to allow all users to verify that their master shares obtained are generated by a symmetric bivariate polynomial having degree $t-1$ without revealing the secrecy of both the polynomial and master shares.

Step 1. All users need to agree a set of random integers, $\{r_l | l = 1, 2, \ldots, n\}, \forall r_l \in GF(p)\}$.

Step 2. Each user, U_i, uses his sub-shares, $f_l(x_i, y)$, $l = 1, 2, \ldots, n$, to compute $v_i = \sum_{l=1}^{n} r_l f_l(x_i, x_i) \bmod p$. v_i is made available to all other users.

Step 3. After receiving all v_l, $l = 1, 2, \ldots, n$, from other users, each user, U_i, following the Lagrange interpolation formula to compute $\sum_{l=1}^{n} v_l \prod_{j=1, j\neq l}^{n} \frac{x - x_j}{x_l - x_j} \prod_{j=1, j\neq l}^{n} \frac{y - x_j}{x_i - x_j} \bmod p = G(x,y)$.

Step 4. Each user, U_i checks whether $G(x,y)$ is a symmetric bivariate polynomial having degree $t-1$. If it is, then the verification is passed; otherwise, master shares need to be re-generated.

Theorem 4 *If VMS is passed, master shares are generated by a symmetric bivariate polynomial having degree $t-1$.*

Proof From Step 2, we have $v_i = \sum_{l=1}^{n} r_l f_l(x_i, x_i) \bmod p$. According to the secret sharing homomorphism as defined in ***Definition 1***, v_i is a share of polynomial, $\sum_{l=1}^{n} r_l f_l(x, y) \bmod p$. Then, in Step 3, the Lagrange interpolating polynomial $G(x, y)$ is $\sum_{l=1}^{n} r_l f_l(x, y) \bmod p$. If the verification is passed in Step 4, $G(x, y)$ is a symmetric bivariate polynomial having degree $t-1$. From ***Theorem 1*** , the master share, $F_i(y)$, is a share of the polynomial, $F(x, y) = \sum_{l=1}^{n} f_l(x, y) \bmod p$. Since $\{r_l | l = 1, 2, \ldots, n\}$ is a set of random integers, it has very high probability that master shares are generated by a symmetric bivariate polynomial having degree $t-1$. ■

Theorem 5 *The verification does not reveal the secrecy of both the polynomial and master shares*.

Proof Each master share is $F_i(y) = \sum_{l=1}^{n} f_l(x_i, y) mod\ p$. In Step 2, each released public value is $v_i = \sum_{l=1}^{n} r_l f_l(x_i, x_i) mod\ p$. It is obvious that the master share, $F_i(y)$, is unconditionally protected from the public value, v_i. Furthermore, the secret polynomial of master shares is $F(x, y) = \sum_{l=1}^{n} f_l(x, y) mod\ p$ and the recovered Lagrange interpolating polynomial is $G(x, y) = \sum_{l=1}^{n} r_l f_l(x, y) mod\ p$. It is obvious that the polynomial, $F(x, y)$, is unconditionally protected from the recovered polynomial, $G(x, y)$. ■

4.3 TGK

We assume that an initiator, U_i, in the set wants to transmit a secret group key, $k \in GF(p)$, to a subset of l users, $\{U_{r_j} | U_{r_j} \in U, j = 1, 2, \ldots, l\}$, where $1 \le l \le n$, in the set, U.

Step 1. The initiator, U_i, with his master share, $F_i(y)$, computes pairwise shard secrets. $s_{i,j} = F_i(x_{r_j})$, between any pair of users, U_i and $U_{r_j}, j = 1, 2, \ldots, l$.

Step 2. The initiator computes, $c_j = E_{s_{i,j}}(k), j = 1, 2, \ldots, l$, where $E_{s_{i,j}}(k)$ represents the encryption of the group key, k, using the key, $s_{i,j}$. U_i sends c_j to each user, U_{r_j}, separately.

4.4 RGK

Step 1. Each user, U_{r_j}, in the subset uses his master share, $F_{r_j}(y)$, to compute the pairwise shared secret, $s_{i,j} = F_{r_j}(x_i)$, between users, U_i and U_{r_j}.

Step 2. U_{r_j} can decrypt the ciphertext, c_j, using the pairwise shared secret, $s_{i,j}$, to obtain the group key as $k = D_{s_{i,j}}(c_j), j = 1, 2, \ldots, l$, where $D_{s_{i,j}}(c_j)$ represents the decryption of the ciphertext, c_j, using the key, $s_{i,j}$.

5 Analysis

5.1 Functionalities

The functionalities of our proposed protocol are summarized below.

(a) The master shares of users are generated by all users together and each master share is only known to each individual user.
(b) The storage of each master share is the coefficients of a univariate polynomial. Each master share of user enables to establish pairwise shared keys between the user and any other users.
(c) After generating master shares, the verification can help users to verify that their master shares are generated consistently without revealing the secrecy of master shares.
(d) Each user can act as a KGC to initiate a secure group communication. The KGC selects a group key and encrypts the group key using each pairwise shared key separately. The ciphertext of group key is sent to each user. To decrypt the ciphertext of group key, each user works individually to recover the group key. The group key transmission is very efficient in terms of computation and communication.

5.2 Security

We discuss in detail how our proposed protocol can resist attacks described in Section 3.2.

Inside Attack: From ***Theorem 3***, we know that the proposed GMS can resist up to $\lfloor \frac{t-1}{2} \rfloor$ colluded users to recover the secret polynomial $F(x, y)$. Thus, any legitimate user who owns a master share cannot recover other user's master share and therefore is not able to reveal other group keys that he/she

is unauthorized to know nor the attacker is able to impersonate other user in a secure group communication.

Outside Attack: In our proposed TGK, each group key is sent to users using pairwise shared keys. Since any outside attacker does not own any valid master share and therefore cannot recover any group key that he/she is unauthorized to know.

5.3 Comparison

In this section, we compare our proposed group key establishment protocol with three other related protocols in terms of trust feature, computational and communication complexities, and storage requirement. We summarize the comparison result in Table 1.

In the comparison, we will only focus on transmitting the group key (TGK only) to users. Furthermore, we will only make some high-level comparisons among protocols since different technologies have been used in these protocols. For more low-level evaluations in terms of execution time is currently undertaken. General speaking, there are three different technologies have been used in these protocols, conventional encryption, polynomial evaluation and public-key evaluation. Conventional encryption such as AES [34] has the fastest processing speed among these technologies. On the other hand, public-key evaluation has the slowest processing speed since most public-key evaluations involve modulo exponentiations with a very large modulus.

In our protocol, there is no trust KGC needed. Both GMS and VMS are needed only initially. The initiator of a group communication needs to do l polynomial evaluations and l encryptions respectively to transmit the group key to l users in a secure group communication. At the same time, each user needs to do one polynomial evaluation and one decryption to recover the group key. Harn et al. [15] protocol was proposed in 2010 which uses a secret sharing with RSA modulus to transmit a group key to all users. This protocol uses a trusted KGC and KGC needs to use polynomial evaluations to transmit the group key to all users. Each user needs to use polynomial evaluations to recover the group key. We want to point out that there is a major difference in polynomial evaluation between our proposed protocol and Harn et al. protocol [15]. In our proposed protocol, the modulus, say 100 bit long, used in polynomial evaluation, is much smaller than the modulus, say 1024 bit long, used in [15] protocol. Thus, our proposed protocol is much faster than Harn et al. [15] protocol. In 2014, Harn et al. [8] proposed a group DH protocol based on the secret sharing. Their protocol does not need a trust KGC to transmit the group key. The group key is determined by all users in a group communication using both secret sharing and DH schemes. The main problem of their protocol is due to its computational and communication complexity because the group key is determined by all group users so each member needs to compute DH keys and exchange information to other users. In 2015, Wu et al. [9] proposed a group key agreement based on the public-key broadcast encryption without needing a trusted KGC. Their sprotocol is built from public-key based bilinear groups.

6 Conclusion

We propose a novel design of a centralized group key establishment protocol without a mutually trusted party. Our design is unique since all existing centralized group key establishment protocols need a mutually trusted KGC to register users and transmit a randomly selected group key to users in a group communication. Our design removes the need of a mutually trusted KGC such that each user can act as a KGC to register other users and transmit a group key to other users. A symmetric bivariate polynomial is used in our design such that pairwise shared keys can be established between any pair of users. We believe that our design create a new research direction in the area of group key establishment protocols especially suitable for applications such as users in a social network who want to hold a secure group communication.

Table 1 Comparisons among our protocol and related protocols

	Trust Feature	Computational Complexity	Communication Complexity	Storage requirement
Proposed Protocol	no trust need	conventional encryption and polynomial evaluation with a small modulus	one round transmission	coefficients of a univariate polynomial
Harn et al. [15]	a trusted KGC needed	polynomial evaluation with a large modulus	one round transmission	one coordinate point
Harn et al. [8]	no trust needed	public-key and polynomial evaluations	three round transmissions	a pair of private and public keys
Wu et al. [9]	no trust needed	Public-key evaluations	One round transmission	a pair of private and public keys

References

1. Diffie W, Hellman ME (1976) New directions in cryptography. IEEE Trans Inf Theory 22(6):644–654
2. Ingemarsson I, Tang DT, Wong CK (1982) A conference key distribution system. IEEE Trans Inf Theory 28(5):714–720
3. Steer DG, Strawczynski L, Diffie W, Wiener MJ (1988) A secure audio teleconference system. Proc. of Crypto '88, LNCS, vol. 403, pp 520–528
4. Burmester M, Desmedt Y (1995) A secure and efficient conference key distribution system. Proc. of Eurocrypt '94, LNCS, vol. 950, pp 275–286
5. Steiner M, Tsudik G, Waidner M (1996) Diffie-Hellman key distribution extended to group communication. Proc. Third ACM Conf. Computer and Comm. Security (CCS '96), pp 31–37
6. Bresson E, Chevassut O, Pointcheval D, Quisquater J-J (2001) Provably authenticated group Diffie-Hellman key exchange. Proc. of ACM Conf. Computer and Comm. Security (CCS '01), pp 255–264
7. Bohli JM (2006) A framework for robust group key agreement. Proc. of Int'l Conf. Computational Science and Applications (ICCSA '06), LNCS, vol. 3982, pp 355–364
8. Harn L, Lin C (2014) Efficient group Diffie-Hellman key agreement protocols. Comput Electr Eng 40:1972–1980
9. Wu Q, Qin B, Zhang L, Domingo-Ferrer J, Manjón JA (2013) Fast transmission to remote cooperative groups: a new key management paradigm. IEEE/ACM Trans Networking 21(2):621–633
10. IEEE CS (2004) 802.1X, IEEE standard for local and metropolitan area networks, port-based network access control. The Inst. of Electrical and Electronics Engineers, Inc
11. Laih C, Lee J, Harn L (1989) A new threshold scheme and its application in designing the conference key distribution cryptosystem. Inf Process Lett 32:95–99
12. Berkovits S (1991) How to broadcast a secret. Proc. of Eurocrypt '91, LCNS, vol. 547, pp 536–541
13. Li CH, Pieprzyk J (1999) Conference key agreement from secret sharing. Proc. of Fourth Australasian Conf. Information Security and Privacy (ACISP '99), LNCS, vol. 1587, pp 64–76
14. Saze G (2003) Generation of key predistribution schemes using secret sharing schemes. Discret Appl Math 128:239–249
15. Harn L, Lin C (2010) Authenticated group key transfer protocol based on secret sharing. IEEE Trans Comput 59(6):842–846
16. Bohli JM (2006) A framework for robust group key agreement. Proc. Int'l Conf. Computational Science and Applications (ICCSA '06), pp 355–364
17. Katz J, Yung M (2007) Scalable protocols for authenticated group key exchange. J Cryptol 20:85–113
18. Chor B, Goldwasser S, Micali S, Awerbuch B (1985) Verifiable secret sharing and achieving simultaneity in the presence of faults. In Proceedings of the 26th IEEE Symposium on the Foundations of Computer Science. IEEE Press, pp 383–395
19. Feldman P (1987) A practical scheme for non-interactive verifiable secret sharing. In Proceedings of the 28th IEEE Symposium on Foundations of Computer Science, 27–29 October, Los Angeles, IEEE Computer Society, pp 427–437
20. Pedersen TP (1992) Non-interactive and information-theoretic secure verifiable secret sharing. In Advances in Cryptology - CRYPTO '91, LNCS, vol. 576, Springer-Verlag, pp129–140
21. Benaloh JC (1987) Secret sharing homomorphisms: keeping shares of a secret secret. In Advances in Cryptology - CRYPTO '86, Lecture Notes in Computer Science, vol. 263, Springer-Verlag, pp 251–260
22. Stadler M (1996) Publicly verifiable secret sharing. In Advances in Cryptology - EUROCRYPT '96, LNCS, vol. 1070, Springer-Verlag, pp 190–199
23. Fujisaki E, Okamoto T (1998) A practical and provably secure scheme for publicly verifiable secret sharing and its applications. In Advances in Cryptology - EUROCRYPT '98, LNCS, vol. 1403, Springer-Verlag, pp 32–46
24. Fiat A, Shamir A (1987) How to prove yourself: Practical solutions to identification and signature problems. In Advances in Cryptology - CRYPTO 1986, LNCS, vol. 263, Springer-Verlag, pp186–194
25. Peng A, Wang L (2010) One publicly verifiable secret sharing scheme based on linear cod. In Proceeding of 2nd Conference on Environmental Science and Information Application Technology, pp 260–262
26. Ruiz A, Villar JL (2005) Publicly verifiable secret sharing from Paillier's cryptosystem. In Proceedings of WEWoRC '05, LNI P-74, pp 98–108
27. Paillier P (1999) Public-key cryptosystems based on composite degree residuosity classes. In Advances in Cryptology - EUROCRYPT '99, LNCS, vol. 1592, Springer-Verlag, pp 223–238
28. Tian Y, Peng C, Ma J (2012) Publicly verifiable secret sharing schemes using bilinear pairings. Int J Netw Secur 14(3):142–148
29. Wu T, Tsenga Y (2011) A pairing-based publicly verifiable secret sharing scheme. J Syst Sci Complex 24(1):186–194
30. Gennaro R, Ishai Y, Kushilevitz E, Rabin T (2001) The round complexity of verifiable secret sharing and secure multicast. STOC, pp 580–589
31. Katz J, Koo C, Kumaresan R (2008) Improved the round complexity of VSS in point-to-point networks. Proceedings of ICALP '08, Part II, in: LNCS, vol. 5126, Springer, pp 499–510
32. Kumaresan R, Patra A, Rangan CP (2010) The round complexity of verifiable secret sharing: the statistical case. Advances in Cryptology - ASIACRYPT 2010, LNCS, vol. 6477, Springer, pp 431–447
33. Nikov V, Nikova S (2005) On proactive secret sharing schemes. LNCS, vol. 3357, Springer, pp 308–325
34. Standard N F. Announcing the advanced encryption standard (AES)[J]. Federal Information Processing Standards Publication, 2001, 197: 1-51.

Lein Harn received the B.S. degree in electrical engineering from the National Taiwan University in 1977, the M.S. degree in electrical engineering from the State University of New York-Stony Brook in 1980, and the Ph.D. degree in electrical engineering from the University of Minnesota in 1984. He is currently a Professor at the Department of Electrical and Computer Engineering, University of Missouri, Kansas City (UMKC). He is currently investigating new ways of using secret sharing in various applications.

Ching-Fang Hsu received the M. Eng. and the Ph.D. degrees in information security from the Huazhong University of Science and Technology, Wuhan, China, in 2006 and 2010 respectively. From Sep. 2010 to Mar. 2013, she was a Research Fellow at the Huazhong University of Science and Technology. She is currently an Assistant Professor at Central China Normal University, Wuhan, China. Her research interests are in cryptography \and network security, especially in secret sharing and its applications.

Bohan Li is currently attending the undergraduate program at Huazhong University of Science and Technology, Wuhan, China, major in optoelectronic information. His study interests are in optoelectronic communication and network security.