SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks (2012)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.671

RESEARCH ARTICLE

Verifiable symmetric polynomial-based key distribution schemes

Yan-xiao Liu[1]*, Yu-qing Zhang[1,2], Lein Harn[3] and Yu-pu Hu[1]

[1] State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, Shaanxi, 710071, China
[2] National Computer Network Intrusion Protection Center, GUCAS, Beijing, China
[3] CSEE Department, University of Missouri-Kansas City, Kansas City, MO, U.S.A.

ABSTRACT

Symmetric polynomial-based key distribution scheme has been widely adopted in various communication applications. This type of key distribution consists of a server and a set of users, where the server is responsible to distribute shares for each user via a symmetric polynomial. Based on the property of symmetry of this polynomial, each pair of users can compute a common secret key using their shares for establishing a secure communication channel. However, some users may receive faulty shares from the server because of some uncertain factors in the communication environment, such as software failures and transmission errors. As a result, the users who receive faulty shares cannot share common secret keys with other users. To solve this problem, in this paper, we propose two individual verifiable key distribution schemes on the basis of a symmetric polynomial based key distribution. In both our proposed schemes, the server adopts the same approach to distribute shares for users; the users are able to verify the validity of their shares without revealing them before establishing communication channels. If all shares are verified valid, users can ensure that each pair of them possesses a common secret key, they can establish secure communication channels when needed; otherwise, all users can collaborate to identify those users who possess faulty shares and require the server to distribute a set of valid shares for those users. Furthermore, both our proposed schemes are efficient, because the procedures of verification and identification do not involve any complicated cryptographic operation. Copyright © 2012 John Wiley & Sons, Ltd.

KEYWORDS

symmetric polynomial; key distribution; verifiable; identification

***Correspondence**

Yan-xiao Liu, State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, Shaanxi, 710071, China.
E-mail: yanxiao_liu@hotmail.com

1. INTRODUCTION

In a communication system, if two users wish to communicate with symmetric encryption, they must possess a common secret key. A key distribution scheme is a mechanism to distribute initial private pieces of information (which we call shares) among all users, such that each pair of users can compute a common secret key for secure communication. This share is generated and distributed by a server that is active only at the distribution phase (such as a set-up phase of public key systems [1]). Let n denote the number of users in a communication system, a basic and straightforward secure key distribution scheme is that a server generates $\frac{n(n-1)}{2}$ keys, and distributes $n-1$ keys for each user, one for each possible communication. The disadvantage of this approach is that when n is large, it becomes problematic or even impossible to manage all keys. This is known as the n^2 problem.

In 1992, Blundo *et al*. [2] first proposed to adopt a symmetric polynomial in a key distribution scheme (this approach was initiated in [3]). In their approach [2], the server is responsible to pick a secret symmetric bivariate polynomial $F(x,y)$ with both variables x and y of degree k, then generates and sends a univariate polynomial with degree k from $F(x,y)$ as a share for each user. Each pair of users can compute a common secret key using their shares. Compared with the key distribution scheme (i.e., each user needs to store $n-1$ keys) introduced previously, the scheme in [2] can significantly reduce the size of stored information for each user, and the keys are still secure only with respect to an adversary controlling coalitions of a limited size. Since then, the symmetric polynomial-based key distribution has been widely adopted in various communication applications. For instance, in the sensor networks, many key distribution techniques are infeasible because of the resource constraints on sensors. As an alternative, symmetric polynomial-based key distribution requires lower storage for each user, and this type of key distribution scheme has been widely adopted in sensor networks [4–8].

In this paper, we address a realistic problem in a symmetric polynomial-based key distribution scheme, where some users may receive faulty shares from the server because of some uncertain factors, such as software failures and transmission errors. In this case, these users cannot compute common secret keys using their faulty shares with the others to establish secure communication channels. On the basis of the presentation of faulty shares, this problem can be divided into two cases: Case 1, some faulty shares are not k degree polynomials; Case 2, all the faulty shares are also k degree polynomials, but they are not generated from a symmetric polynomial. As described previously, the share in a symmetric polynomial-based key distribution scheme [2] should be a univariate polynomial of degree k; as a result, Case 1 can be easily detected, but in Case 2, users are unable to detect the faulty shares directly. Therefore, in this work, we only consider the problem of faulty shares under the Case 2.

In this paper, we propose two individual verifiable key distribution schemes to solve the problem described previously. Both our schemes are on the basis of symmetric polynomial-based key distribution [2], and after receiving shares from server, all users in our schemes are able to verify the validity of their shares without revealing them. In other words, all users in our proposed schemes are capable of detecting whether faulty shares exist among users before establishing communications channels. When faulty shares exist, all users can also collaborate to identify those users who receive faulty shares from server and require the server to distribute a set of valid shares to them. Both our proposed verifiable key distribution schemes are very efficient because the procedures of verification and identification do not involve any complicated cryptographic operation. Furthermore, the proposed schemes are unconditionally secure only with respect to an adversary controlling coalitions of a limited size. We will discuss the security in Section 4 in detail.

The rest of this paper is organized as follows. In the next section, we introduce the symmetric polynomial-based key distribution in [2]. In Section 3, we describe the problem in symmetric polynomial-based key distribution that we considered in this paper and illustrate the approach to solving this problem. In Section 4, we propose two individual verifiable key distribution schemes and discuss some properties of our proposed schemes. We conclude in Section 5.

2. SYMMETRIC POLYNOMIAL BASED KEY DISTRIBUTION

In this section, we introduce the model of symmetric polynomial-based key distribution [2]. Let S denote the server and $U_i, i = 1, 2, \ldots, n$ denote the n users included in the communication system. The key distribution scheme [2] can be described as follows.

For each pair of users, (U_i,U_j), U_i can compute a secret key $F(i,j)$ by $F(i,j) = f_i(j)$; U_j can compute a secret key $F(j,i)$ by $F(j,i) = f_j(i)$. Because $F(x,y)$ has the property of symmetry, $F(i,j) = F(j,i)$. Therefore, each pair of users possesses a common secret key, $F(i,j)$. In addition, the share of each user is a univariate polynomial $f_i(y)$ with degree k, the user needs to store $k+1$ elements in the field of $GF(p)$. In general setting, the parameter k is much smaller than n. This is why the scheme [2] can significantly reduce the size of stored information for each user. In [2], it also proved that the secret keys are unconditionally secure when the colluded users are no more than k. In other words, any k or less than k users are unable to obtain the secret keys that they should not possess.

3. PROBLEM DESCRIPTION AND SOLUTION

In this section, we give a description of our proposed problem in the symmetric polynomial-based key distribution scheme and illustrate our approach to solving this problem.

3.1. Problem description

As shown in Figure 1, the share of each user is generated from a symmetric polynomial, and each pair of users can compute a common secret key using their shares. The problem we considered is that, because of some uncertain factors such as software failures and transmission errors, some users may receive faulty shares from the server. In this case, during the common secret key phase, users who receive faulty shares cannot compute common secret keys with other users to establish secure communication channels. For instance, let $F(x,y)$ be the symmetric polynomial selected by server. Suppose the user U_i receives the valid share, $f_i(y) = F(i,y)$, but another user U_j receives a faulty share $f'_j(y) \neq F(j,y)$ ($f'_j(y)$ is a kth degree polynomial). Therefore, the user U_j does not possess the common secret key $F(i,j)$ with U_i; because $f'_j(i) \neq f_i(j) = F(i,j)$, they are not able to establish a secure communication channel.

3.2. Solution

First, we introduce a property of "verifiable." The property of "verifiable" was proposed in [9]. In that scheme [9], a

Symmetric polynomial based key distribution

Distribution phase:

1) The server S picks a symmetric bivariate polynomial $F(x, y)$ in the field of GF(p), where both the variables x and y have degree k.
2) The server S computes and distributes the share $f_i(y) = F(i, y)$ to each user $U_i, i = 1, 2, \ldots, n$ secretly.

Common secret key phase:

Any pair of users (U_i, U_j) possess a common secret key $F(i, j)$. i.e., U_i can obtain $F(i, j)$ by evaluating $f_i(y)$ at the point $y = j$; U_j can obtain $F(i, j)$ by evaluating $f_j(y)$ at the point $y = i$.

Figure 1. Symmetric polynomial-based key distribution.

Security Comm. Networks (2012) © 2012 John Wiley & Sons, Ltd.
DOI: 10.1002/sec

dealer adopts Shamir's approach [10] to distribute shares to shareholders, and these shareholders are capable of verifying the validity of their shares without revealing them. If the verification is successful, shareholders ensure that all the shares are valid; otherwise the dealer needs to distribute shares once again. In this paper, adopting the same concept "verifiable" in [9], we propose two individual verifiable key distribution schemes on the basis of the original key distribution scheme (shown in Figure 1). Both our proposed schemes enable users to verify the validity of their shares. (i.e., all shares are generated from a symmetric polynomial). In other words, in our proposed schemes, all users can detect whether there exist faulty shares among users. If all shares are verified valid, then any pair of users ensure that they possess a common secret key via their shares, they can establish a secure communication channel using this key; else we adopt the approach in [11], which enables users to identify those users who receive faulty shares from the server and require the server to distribute a set of valid shares to them.

4. PROPOSED VERIFIABLE KEY DISTRIBUTION SCHEMES

In this section, we propose two verifiable key distribution schemes respectively (Schemes 1 and 2). The properties of verifiability, identifiability and security of our proposed schemes are also analyzed in detail.

4.1. Scheme 1

In this subsection, we describe Scheme 1 as shown in Figure 2. In this scheme, the distribution phase is the same as that in the original key distribution scheme, although users are capable of verifying their shares and identifying faulty shares without any additional information.

In Scheme 1, the verification is based on the property of symmetry of the polynomial $F(x,y)$, and the identification of faulty shares is based on "majority voting," which is inspired by Harn-Lin's scheme [11].

The following theorems are used to prove the properties of Scheme 1. In Theorem 1, we will show that in the verification phase of Scheme 1, users are capable of verifying the validity of their shares successfully. In Theorem 2, we illustrate that when there are at least $k+2$ valid shares among users, then all users can collaborate to identify those users who receive faulty shares. In Theorem 3, we prove that the secret keys in Scheme 1 are unconditionally secure only with respect to an adversary controlling coalitions of a limited size.

Theorem 1. *In verification phase of Scheme 1, all users are able to verify the validity of their shares successfully.*

Proof. In Scheme 1, suppose all the shares $f_i(y), i = 1, 2, \ldots, n$ are generated from a bivariate polynomial $F(x,y)$, which is indeed unknown to each user, the objective of verification phase is to ensure that $F(x,y)$ is a symmetric polynomial with both variables of degree k. Because $f_i(y) = F(i,y)$, we have $f_i(c) = F(i,c), i = 1, 2, \ldots, n$. Therefore, it is easy to know that the interpolated polynomial $h_c(x)$ on the n points $(i, f_i(c)), i = 1, 2, \ldots, n$ satisfies that $h_c(x) = F(x,c)$; by the same way, we have $h_d(x) = F(x,d)$. It is easy to observe that if $h_c(x)$ and $h_d(x)$ are of degree k, then the variable x in $F(x,y)$ is also k. In addition, the share of each shareholder is $f_i(y) = F(i,y)$, where the variable y is of degree k; therefore, shareholders can ensure that both variables in $F(x,y)$ are of degree k. For the property of symmetry, in step 4 of the verification phase, we can deduce the equation $F(d,c) = F(c,d)$ from the presentation $h_c(d) = h_d(c)$. Because the values c and d are randomly picked by users, the probability of $F(d,c) = F(c,d)$ when $F(x,y)$ is an asymmetric polynomial is only $\frac{1}{p}$; when the prime number p is large enough for cryptographic applications, this probability can be just ignored. Therefore, we can conclude that $F(x,y)$ is a symmetric polynomial when $h_c(d) = h_d(c)$. In this case, all the shares are valid, and each pair of users possesses a common secret key using their shares. In addition, if some users receive faulty shares because of uncertain reasons, then the interpolated polynomial $h_c(x)$ and $h_d(x)$, those n points would be of degree more than k with probability almost 1. This fact can be detected in our verification phase. In sum, our Scheme 1 can verify the validity of shares successfully.

Scheme 1

Distribution phase:

Same as the original key distribution scheme (as shown in **Fig. 1**).

Verification phase:

1) All users collaborate to pick two values, $c, d \in GF(p), c, d > n$
2) Each user $U_i, i \in [1, n]$ computes and releases $f_i(c)$ and $f_i(d)$ using his share $f_i(y)$.
3) Computing the interpolated polynomials, $h_c(x)$ and $h_d(x)$ on the points $(i, f_i(c)), i = 1, 2, \ldots, n$ and $(i, f_i(d)), i = 1, 2, \ldots, n$ respectively.
   a) If $h_c(x)$ and $h_d(x)$ are both of degree k and $h_c(d) = h_d(c)$, all shares of each user are valid;
   b) otherwise if $h_c(x)$ and $h_d(x)$ are of degree more than k, there exist faulty shares among users, switch to identification phase.

Identification phase:

1) For each combination of $k+1$ points in the pool of $(i, f_i(c)), i = 1, 2, \ldots, n$, compute an interpolated polynomials $h_{c,l}(x)$ on these $k+1$ points for $l = 1, 2, \ldots, C_n^{k+1}$.
2) Figuring out the majority polynomial $h_c'(x)$ among these C_n^{k+1} polynomials $h_{c,l}(x), l = 1, 2, \ldots, C_n^{k+1}$,
3) Let $A_r, r = 1, 2, \ldots, w$ denote all the combinations of $k+1$ points that deduce the interpolated polynomial $h_c'(x)$. Let U denote the set of all users and $U^r, r = 1, 2, \ldots, w$ denote the set of $k+1$ users included in A_r. The set of users U_f who receive faulty shares can be presented as $U_f = U - (U^1 \cup U^2 \cup, \ldots, \cup U^w)$.
4) The users who are identified to receive faulty shares require the server to distribute valid shares for them.

Figure 2. Our proposed Scheme 1.

Remark. In the verification phase, there exists a special case that both $h_c(x)$ and $h_d(x)$ are of degree k, but $h_c(d) \neq h_d(c)$. This means that all shares are generated from an asymmetric polynomial $F'(x,y)$ with both variables of degree k. When the faulty shares are caused by some uncertain factors, this case cannot happen. The only reason for this case is that a malicious server chose $F'(x,y)$ on purpose. In our works, the server is always assumed honest; thus, this case is not under our consideration.

Theorem 2. *In identification phase of Scheme 1, when there are at least $k+2$ valid shares, all users are able to identify those users who receive faulty shares.*

Proof. Let $F(x,y)$ be the kth degree symmetric polynomial selected by the server and $f_1(y), f_2(y), \ldots, f_n(y)$ be a set of valid shares. It is easy to understand that the interpolated polynomial $h_c(x)$ on the n points $(1, f_1(c)), (2, f_2(c)), , (n, f_n(c))$ equals to $F(x,c)$ and is of degree k. If there exist faulty shares, this interpolated polynomial would be of degree more than k. However, each set of those $k+1$ points can still deduce a kth degree interpolated polynomial and when all these $k+1$ corresponding shares are valid, this interpolated polynomial just equals to $F(x,c)$. Let $h_{c,l}(x), l = 1, 2, \ldots, C_n^{k+1}$ be all the interpolated polynomials that generated from each combination of $k+1$ of those points. The key of our identification is to figure out the legal polynomial $F(x,c)$ among those C_n^{k+1} interpolated polynomials. Considering the case that there are at least $k+2$ valid shares, then there would be at least $C_{k+2}^{k+1} = k+2$ identical interpolated polynomial $F(x,c)$ in the pool of $h_{c,l}(x), l = 1, 2, \ldots, C_n^{k+1}$. On the other hand, any combination of $k+1$ points that contains at least one faulty point would generate a random interpolated polynomial. Therefore, $F(x,c)$ becomes the majority polynomial and is regarded as the legal polynomial $F(x,c)$. Then the users who receive faulty shares can be identified successfully according to our algorithm. As a result, when there are at least $k+2$ valid shares, those users who receive faulty shares can be identified successfully.

As shown in Theorem 2, the precondition of successful identification is that there are at least $k+2$ valid shares. Although we address the problem that users may receive faulty shares because of some uncertain factors, receiving a faulty share is still a small probability event. The assumption that there exist at least $k+2$ valid shares is reasonable. In addition, the approach of "majority voting" in our identification was first adopted in a cheating identification scheme [11] where its practical applicability has been proved. Later, [12] showed that the scheme [11] can be broken by a flexible attack. The basic assumption in [12] is that the attackers are malicious and they collude together to cheating. However, in our assumption, the users in the identification phase are honest and they collaborate to identify faulty shares. Therefore, the identification is effective in our Scheme 1.

Theorem 3. *In Scheme 1, the keys are unconditionally secure when there are no more than $k-2$ colluded users.*

Proof. In [2], it is proved that all keys in the original key distribution scheme are unconditionally secure when the colluded users controlled by an adversary are no more than k. This is because in the original key distribution scheme, the symmetric polynomial $F(x,y)$ picked by server has degree k on both variables x and y, the number of coefficients in $F(x,y)$ is $(k+1)^2$, and on the basis of the property of linearity, any $(k+1)^2$ linear independent points on $F(x,y)$ are capable of reconstructing $F(x,y)$. Because each user U_i, i $[1,n]$ has a share $f_i(y) = F(i,y)$, he or she possesses $k+1$ linear independent points on $F(x,y)$, i.e., $f_i(1) = F(i,1), f_i(2) = F(i,2), , f_i(k+1) = F(i,k+1)$. Any $k+1$ users can gather $(k+1)^2$ linear independent points on $F(x,y)$ to recover it, then they can obtain the secret keys they should not know through $F(x,y)$. On the other hand, any k or less than k users can obtain at most $k(k+1)$ linear independent points on $F(x,y)$, they are unable to recover $F(x,y)$. For our proposed Scheme 1, all users can obtain $h_c(x) = F(x,c)$ and $h_d(x) = F(x,d)$ in the verification phase; therefore, there are already $2(k+1)$ public linear independent points on $F(x,y)$ (i.e., $h_c(1) = F(1, c), h_c(2) = F(2,c), \ldots, h_c(k+1) = F(k+1, c); h_d(1) = F(1,d), h_d(2) = F(2,d), \ldots, h_d(k+1) = F(k+1, d)$). In this case, $k-1$ users are enough to obtain $(k+1)^2$ linear independent points on $F(x,y)$ to recover it, but $k-2$ or less users are still unable to recover $F(x,y)$ to obtain the secret keys they should not know. Therefore, in Scheme 1, the secret keys are unconditionally secure when there are no more than $k-2$ colluded users.

Remark. In Theorem 3, it is assumed that there are some vicious users who want to collaborate to obtain secret keys they should not know. We want to emphasize that these vicious users and the users in our scheme are in the different models. In our model, all users are honest and they will act honestly throughout the scheme. Furthermore, even though we assume that the users in our model can be dishonest, they will still act honestly during the verification phase and identification phase of Scheme 1. Because the objective of the two phases is to detect and identify faulty shares, no information about the secret keys is revealed. If any user acts dishonestly in the verification phase, it only causes the key distribution to be halted, and no pair of users would establish communication channels using unverified keys, which does not benefit vicious users at all.

In Scheme 1, we can observe that the users do not need additional information from server to achieving verification or identification. However, as analyzed in Theorem 3, the minor drawback of Scheme 1 is that the security level is descended a little from the original key distribution scheme, i.e., our Scheme 1 can resist the attack of up to $k-2$ vicious users, whereas the original one could resist such attack of k vicious users.

DOI: 10.1002/sec

4.2. Scheme 2

In this subsection, we propose another verifiable key distribution scheme (Scheme 2) that also enables users to verify the validity of their shares and identify faulty shares. Scheme 2 (as shown in Figure 3) has the same security level as the original key distribution scheme, and as a trade-off, each user in Scheme 2 needs to store more information than Scheme 1. Scheme 2 is outlined as follows.

In the Theorem 4, we show that Scheme 2 enables users to verify the validity of their shares. The identification phase in Scheme 2 is same as the identification in Scheme 1. We do not discuss the property of identification repeatedly. In Theorem 5, we illustrate that Scheme 2 has the same security level as the original key distribution scheme. First, we introduce an auxiliary lemma for Theorem 4.

Lemma 1. *Given two random values $a_1, a_2 \in GF(p)$, if the combination $a_1F(x,y) + a_2G(x,y)$ (both $F(x,y)$ and $G(x,y)$ are generated from $GF(p)$ and have degree k with both variables) is a symmetric polynomial, then both of the polynomials $F(x,y)$ and $G(x,y)$ are symmetric polynomials.*

Proof. We can observe that when both $F(x,y)$ and $G(x,y)$ are symmetric, then $a_1F(x,y) + a_2G(x,y)$ is symmetric. On the other hand, when at least one of $F(x,y)$ and $G(x,y)$ is asymmetric, supposing that $f_{i,j}$ and $g_{i,j}$ $(0 \le i,j \le k)$ are coefficients of x^iy^j in $F(x,y)$ and $G(x,y)$, therefore $a_1F(x,y) + a_2G(x,y)$ is symmetric means that the values a_1, a_2 have to satisfy $a_1f_{i,j} + a_2g_{i,j} = a_1f_{j,i} + a_2g_{j,i}$, $0 \le i,j \le k$. Because the two values a_1, a_2 are randomly selected, it is easy to understand that the probability of the aforementioned equations is much less than $\frac{1}{p}$; when p is large enough for cryptographic applications, this probability can be ignored. Therefore, when at least one of $F(x,y)$ and $G(x,y)$ is asymmetric, the random linear combination of $a_1F(x,y) + a_2G(x,y)$ cannot be symmetric. In sum, we can conclude that when the linear combination $a_1F(x,y) + a_2G(x,y)$ is a symmetric polynomial, then both of $F(x,y)$ and $G(x,y)$ are symmetric.

Scheme 2

Distribution phase:

1) The server S picks two symmetric polynomial $F(x,y)$ and $G(x,y)$, both the two polynomials has degree k on the variables x and y.
2) The server S computes and distributes two share $f_i(y) = F(i,y)$ and $g_i(y) = G(i,y)$ (the share $f_i(y)$ is used for computing common secret keys) to each user $U_i, i = 1, 2, ..., n$ secretly.

Verification phase:

1) All users collaboratively work together to choose two weights, w_1, w_2, and two values, c, d in the field of $GF(p)$.
2) Each user U_i uses his shares, $f_i(y), g_i(y)$, to compute and publish two values, $C_i = w_1f_i(c) + w_2g_i(c)$, and $D_i = w_1f_i(d) + w_2g_i(d)$.
3) All users compute the interpolating polynomials, $r_c(x)$ and $r_d(x)$ on the n points, $(1, C_1), (2, C_2), ..., (n, C_n)$ and $(1, D_1), (2, D_2), ..., (n, D_n)$ respectively.
   a) If $r_c(x)$ and $r_d(x)$ are of degree k, and $r_c(d) = r_d(c)$, $F(x,y)$ is a symmetric polynomial, all the shares are valid;
   b) Otherwise if $r_c(x)$ and $r_d(x)$ are of degree more than k, there exist faulty shares among users. Switch to verification phase.

Identification phase:

All users randomly select a value $e \in GF(p)$. Each user U_i computes and releases $f_i(e), i = 1, 2, ..., n$. Then all users adopt the same approach in **scheme 1** to identify faulty shares.

Figure 3. Our proposed Scheme 2.

Theorem 4. *In Scheme 2, users are able to verify the validity of their shares.*

Proof. Suppose the shares $f_i(y), i = 1, 2, \ldots, n$ are generated from a bivariate polynomial $F(x,y)$. Indeed same as Theorem 1, we will prove that users in Scheme 2 can check whether $F(x,y)$ is a symmetric polynomial. In the verification phase of Scheme 2, the two weights, w_1 and w_2, and two values, c and d, are randomly selected by all users. Because $C_i = w_1f_i(c) + w_2g_i(c) = w_1F(i,c) + w_2G(i,c), i = 1, 2, \ldots, n$ and $r_c(x)$ is the interpolated polynomial on the points, (i,C_i), $i = 1, 2, \ldots, n$, it is easy to observe that, $r_c(x) = w_1F(x,c) + w_2G(x,c)$, by the same way, $r_d(x) = w_1F(x,d) + w_2G(x,d)$. Let $Q(x,y) = w_1F(x,y) + w_2G(x,y)$, then $r_c(x) = Q(x,c), r_d(x) = Q(x,d)$. If $r_c(d) = r_d(c)$, it means that $Q(c,d) = Q(d,c)$. Because the values c and d are randomly selected, it can conclude that $Q(x,y)$ is a symmetric polynomial (same analysis as illustrated in Theorem 1). In addition, $Q(x,y) = w_1F(x,y) + w_2G(x,y)$ is a random linear combination of $F(x,y)$ and $G(x,y)$ (i.e., the weights w_1 and w_2 are randomly selected); according to Lemma 1, $F(x,y)$ is asymmetric polynomial, all the shares are valid. On the other hand, if $r_c(d) \neq r_d(c)$, users can obtain that $F(x,y)$ is asymmetric, there exist faulty shares among users. Therefore, in Scheme 2, the users are capable of judging the validity of their shares.

Theorem 5. *In Scheme 2, the secret keys are unconditionally secure when there are no more than k colluded users.*

Proof. In the verification phase of Scheme 2, the public information released by users are $r_c(x) = w_1F(x,c) + w_2G(x,c)$ and $r_d(x) = w_1F(x,d) + w_2G(x,d)$. In contrast with Scheme 1, users cannot obtain any information about $F(x,c)$ and $F(x,d)$ from $r_c(x)$ and $r_d(x)$; as a result, the users cannot obtain any point on $F(x,y)$ from the public information released from the verification phase. Therefore, only $k+1$ or more than $k+1$ users can recover the polynomial $F(x,y)$ using their shares $f_i(y) = F(i,y)$; any k or less users are not enough to recover $F(x,y)$ to obtain the secret keys they should not know

(as analyzed in Theorem 2). In sum, in Scheme 2, the secret keys are unconditionally secure when the colluded users are up to k. In other words, Scheme 2 has the same security level with the original key distribution scheme. Notice, when the users need to identify faulty shares in Scheme 2, each user releases a value of $f_i(e)$. In this case, as illustrated in Theorem 3, Scheme 2 is unconditionally secure when there are no more than $k-1$ colluded users.

Notice that using a hash function is also an efficient way to achieve share verification and identification. However, we do not adopt hash function in our paper, the reason is as follows. Considering the case that adopting a one-way hash function for verification, the server needs to publish the hash values of all shares, and the verification and identification is based on the commitments of these hash values. As illustrated in [13], such schemes are secure only when the computational power of adversary is bounded by a security parameter; otherwise the adversary could derive the share from its hash value. On the contrary, our schemes do not base on any security assumptions, and they are unconditionally secure to the adversaries.

In our proposed verifiable key distribution schemes, users and the server have to compute some additional operations. However, this is unavoidable. In some similar works [14–16] that also achieve share verification, the server (shareholder) and the users (shareholders) also have to compute additional operations. In our schemes, such additional operations are just computing interpolated polynomials. Compared with the exponentiation operations in 1024-bit number field for discrete logarithm used in [15], the calculated amount of computing interpolated polynomials can be negligible. In addition to achieving verification, the users in our scheme only need to publish their necessary information for one time. Compared with those schemes [14,16] that involve several communication rounds between pairs of users, the communication traffic in our schemes is very small. Therefore, the additional operations in our schemes are acceptable.

There are several differences between Schemes 1 and 2. For instance, Scheme 1 can resist attack of up to $k-2$ colluded users, whereas Scheme 2 can resist k colluded users or $k-1$ colluded users when identification is needed. For the space overhead, users in Scheme 1 do not need to store any additional information to achieve verification and identification, whereas in Scheme 2, users need to store two shares, where each of them has the same size as the original share in Scheme 1. The time overhead during verifications and identifications is almost the same, as shown in the schemes, which mainly involve computing several Lagrange interpolating polynomials. On the basis of the different properties, Schemes 1 and 2 can be used in distinct applications. Scheme 1 is fit for some communication systems where the storage of each user is limited, whereas Scheme 2 can be adopted in communication systems of small scale where the security is required precisely.

5. CONCLUSION

In this paper, we consider a practical problem in a symmetric polynomial-based key distribution scheme such that some users may receive faulty shares from the server because of some uncertain factors such as software failures and transmission errors. As a result, these users who receive faulty shares cannot compute common secret key with other users to establish secure communication channels. In order to solve this problem, we propose two individual verifiable key distribution schemes that enable users to verify the validity of their shares and identify the faulty shares after receiving them from the server. Both our proposed schemes are efficient and unconditionally secure with respect to an adversary controlling coalitions of a limited size.

ACKNOWLEDGEMENT

This work is supported in part by National Science Foundation of China under Grant No. 60970140.

REFERENCES

1. Diffie W, Hellman ME. New direction in cryptography. *IEEE Transaction on Information Theory* 1976; **22** (6):644–654.
2. Blundo C, Santis AD, Herzberg A, Kutten S, Vaccaro U, Yung M. Perfectly-secure key distribution for dynamic conferences In. In *Proceeding of CRYPTO 1992*. Lecture Notes in Computer Science Vol. 740. Springer-Verlag: Berlin, 1993; 644–654.
3. Blom R. An optimal class of symmetric key generation systems In. In *Proceeding of EUROCRYPT 1984*. Lecture Notes in Computer Science Vol. 209. Springer-Verlag: Berlin, 1984; 335–338.
4. Liu D, Ning P. Establishing pairwise keys in distributed sensor networks. In *Proceeding of ACM Conference on Computer and Communications Security*, 2003; 52–61.
5. Liu D, Ning P, Li R. Establishing pairwise keys in distributed sensor networks. *ACM Transaction on Information and System Security* 2005; **8**(2):41–77.
6. Cheng Y, Agrawal DP. An improved key distribution mechanism for large-scale hierarchical wireless sensor networks. *Ad Hoc Networks* 2007; **5**(1):35–48.
7. Song G, Zhuhong Q. A compromise-resilient group rekeying scheme for hierarchical wireless sensor networks. In *Proceeding of Wireless Communications and Networking Conference*, 2010; 1–6.
8. Haijun L, Chao W. An energy efficient dynamic key management scheme based on polynomial and cluster in wireless sensor networks. *Journal*

of Convergence Information Technology 2011; **6** (5):321–328.
9. Chor B, Goldwasser S, Micali S, Awerbuch B. Verifiable secret sharing and achieving simultaneity in the presence of faults. In *Proceeding of the 26th IEEE Symposium on Foundations of Computer Science*, 1985; 383–395.
10. Shamir A. How to share a secret. *Communications of the ACM* 1979; **22**(11):612–613.
11. Harn L, Lin C. Detection and identification of cheaters in (*t*,*n*) secret sharing scheme. *Designs Codes and Cryptography* 2009; **52**(1):15–24.
12. Ghodosi H. Comments on Harn-Lin's cheating detection scheme. *Designs Codes and Cryptography* 2011; **60** (1):63–66.
13. Backes M, Kate A, Patra A. Computational verifiable secret sharing revisited In. In *Proceeding of ASIACRYPT 2011*. Lecture Notes in Computer Science Vol. 7073. Springer-Verlag: Berlin, 2011; 590–609.
14. D'Arco P, Stinson DR. On unconditionally secure robust distributed key distribution centers In. In *Proceedings of ASIACRYPT 2002*. Lecture Notes in Computer Science Vol. 2501. Springer-Verlag: Berlin, 2002; 181–189.
15. Feldman P. A practical scheme for non-interactive verifiable secret sharing. In *Proceedings of the 28th IEEE Symposium on Foundations of Computer Science, IEEE Computer Society*, 1987; 427–437.
16. Patra A, Choudhary A, Rabin T, Rangan CP. The round complexity of verifiable secret sharing revisited In. In *Proceedings of CRYPTO 2009*. Lecture Notes in Computer Science Vol. 5677. Springer-Verlag: Berlin, 2009; 487–504.