doi:10.1093/comjnl/bxx003

A Practical Hybrid Group Key Establishment for Secure Group Communications

LEIN HARN[1,3] AND CHING-FANG HSU[2,3*]

[1]*Department of Computer Science Electrical Engineering, University of Missouri-Kansas City, Kansas City, MO 64110, USA*
[2]*Computer School, Central China Normal University, 430079 Wuhan, P.R. China*
[3]*Lein Harn and ChingFang Hsu contributed equally to this work.*
[*]*Corresponding author: cherryjingfang@gmail.com*

A group key establishment enables a group key shared among all group members. In this paper, we proposed a novel group key establishment, which is a hybrid of the Diffie–Hellman (DH) public-key scheme and the secret sharing scheme. Our protocol takes the advantages of the DH scheme, which does not need a mutually trusted key generation center (KGC) and the secret sharing scheme, which reduces the computational time. Employing the DH scheme allows any group member to act as a KGC to distribute a secret key to all group members. The secret sharing scheme is used as the encryption tool to transfer a group key to group members. Since public-key encryption involves modular exponentiations using a larger modulus (say at least 1024 bits) as compared with the secret sharing encryption involves polynomial operations using a smaller modulus (say only 160 bits), our proposed approach is faster than the broadcast encryption in public-key setting. We show that our protocol can provide key secrecy, key authentication and key independence.

Keywords: secret sharing scheme; group key establishment; unconditional secure; group communications.

Received 13 August 2016; revised 11 December 2016; editorial decision 4 January 2017
Handling editor: Mark Josephs

1. INTRODUCTION

Network applications are no longer just one-to-one communication; but involve multiple users (>2). Group communication implies a many-to-many communication and it goes beyond both one-to-one communication (i.e. unicast) and one-to-many communication (i.e. multicast). For example, after earthquake multiple representatives from federal agencies need to form a secure Ad Hoc Network for rescuing people, multiple users from a Social Network need to form a secure communication for exchanging information and employees in a company need to form a secure communication group to hold an Internet conference. Figure 1 illustrates a secure group communication consisting multiple users.

In a secure group communication, a one-time group key needs to be shared among all group members. The shared key will be used by all members to protect their exchange information. In this paper, we propose a novel group key establishment protocol, which is a hybrid of Diffie–Hellman (DH) public-key distribution scheme [1] and the secret sharing scheme. Our protocol has three following properties: (i) it does not need a mutually trusted key generation center (KGC) since every user can act as a KGC to distribute a group key to all group members, (ii) it is non-interactive and (iii) it uses the secret sharing scheme as an encryption to reduce the computational cost.

1.1. Related works

The original DH scheme [1] can establish a session key for only two users. There are papers [2–5], which connect all group users in a ring and use DH scheme to establish a secret group key among group users. In addition, there are some extended DH-based protocols [5–9] with additional features such as providing a framework, achieving authentication,

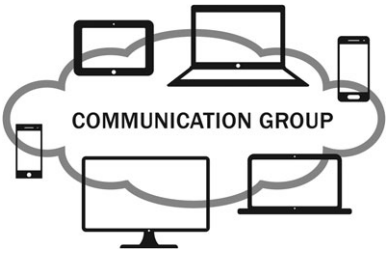

FIGURE 1. Secure group communication.

providing security against malicious insiders and active adversaries, etc. Harn and Lin [10] proposed a group DH protocol using the secret sharing scheme. The main disadvantage of the group DH key exchange is due to its computational and communication complexity. This is because the group key is determined by all group members so each member needs to compute DH keys and exchange information to other members in the process.

Centralized group key establishment protocols are the most widely used group key management protocols due to its efficiency. The centralized group key has a mutually trusted KGC to select a group key and then transport the group key to group members secretly. For example, the IEEE 802.11i standard [11] has an online server to select a group key and transport it to each group member. Laih *et al.* [12] proposed the first group key protocol using a (t, n) secret sharing scheme. The advantage of using a secret sharing scheme is its efficiency. The KGC uses a secret sharing scheme to broadcast the group key to all group members as compared with traditional approach in which KGC uses one-to-one encryption by sending a group key to each member separately. Using the secret sharing approach, every group member needs to share a secret key with the KGC initially. We assume that l members want to establish a secure group communication. The KGC constructs a secret polynomial passing through l shared keys with group members and a randomly selected group key. The degree of this polynomial is l. Then, the KGC broadcasts l public points on the polynomial. Knowing l public points and one addition point (i.e. the shared key with the KGC initially) on the polynomial, each group member can recover the polynomial and thus obtains the group key. However, knowing only l points, any non-member cannot recover the group key. There are two major differences in using a secret sharing scheme in comparing with using a public-key encryption algorithm or a conventional encryption scheme, which are: (i) using secret sharing is a broadcast transmission while the other is one-to-to transmission and (ii) using secret sharing provides information-theoretic security while the other provides computational security. There are some papers [13–15] following the same concept to distribute messages to multiple users. Harn *et al.* [16–18] proposed new group key distribution schemes using multivariate polynomials. However, all these protocols can only establish a single group key or distribute a single message. This is because after establishing a single group key the pairwise shared secret key between each member and the KGC will be revealed to all group members. Harn and Lin [19] proposed a solution, which uses a RSA modulus as compared with the original prime modulus to enable multiple group keys to be established. A new centralized group key transfer protocol [20] has been proposed recently, which uses a linear secret sharing scheme and factoring assumption. In summary, the main disadvantage of the centralized group key establishment protocols is the need of a mutually trusted KGC. In some applications, the KGC will become the traffic bottleneck. Furthermore, a mutually trusted KGC cannot be identified in some networks. For example, users in an Internet chatroom want to hold a secret group communication. But users cannot identify a trusted KGC in such environment. A centralized group key establishment protocol without mutually trusted party [21] is proposed in a recent paper.

The public-key broadcast encryption [22–30] in which any public-key user can act as a KGC to broadcast a group key to all group members is a solution without needing a trusted KGC. The broadcast encryption can transmit a message secretly to all receivers in a broadcast channels. If the transmitted message is a group key, a broadcast encryption can be easily converted into a centralized group key establishment protocol. The disadvantage of using this approach to establish a group key is due to the complexity of public-key computation.

There are trade-offs between centralized and distributed group key establishment protocols. Centralized group key establishment approach enjoys its efficiency in both communication and computation; but it needs a mutually trusted KGC. On the other hand, distributed group key establishment approach enjoys its flexibility without needing a mutually trusted KGC; but it is inefficient in both communication and computation. An alternative approach, which we propose in this paper, can enjoy advantages of both centralized and distributed approaches is very desirable.

1.2. Our contribution

We propose a novel group key establishment protocol without a mutually trusted KGC. Our protocol is a hybrid of DH public-key scheme and a secret sharing scheme. The DH scheme allows any initiator (a group member) to establish a one-time secret key with all group members. The secret sharing scheme is used as the encryption tool to transfer a group key to group members. Since public-key encryption using a larger modulus (say at least 1024 bits) as compared with the secret sharing operation using a smaller modulus (say only 160 bits), our proposed approach is faster than the broadcast encryption in public-key setting. Furthermore, the security of the secret sharing broadcast encryption is perfect but the security of public-key broadcast encryption is computationally secure.

Here, we summarize our contribution below:

- A novel group key establishment protocol without a mutually trusted KGC is proposed.
- The protocol is a hybrid of DH public-key scheme and a secret sharing scheme.
- The protocol is non-interactive so it is very efficient in communication.
- Since the secret sharing operation involves polynomial operations using a smaller modulus (say only 160 bits) than modulus used in most public-key encryptions (say 1024 bits in RSA scheme), our proposed approach is faster than most public-key broadcast encryptions.
- The protocol can provide key secrecy, key authentication and key independence.

The organization of this paper is as follows. In Section 2, we present the model of our group key establishment protocol including protocol description, types of adversaries and security of group keys. In Section 3, we present an authenticated group key establishment protocol. The analysis of our protocol is in Section 4 and performance is included in Section 5. We conclude in Section 6.

2. 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.

2.1. Protocol description

In our proposed protocol, there has no mutually trusted KGC. The group key is determined by an initiator of the group communication and broadcasts the group key to all group members. The initiator can be any member in a group communication. Each group key is used for only one communication session. When a new group communication session is established, a new group key will be generated by an initiator.

In our protocol, each member needs a pair of long-term DH private and public keys and the long-term DH public key has been digitally signed by a trusted Certificate Authority (CA). The digital certificate of public keys of group members will be used by an initiator to assure that the group key can only be decrypted by legitimate group members but not by any non-members. Furthermore, the digital certificate of public key of the initiator will be used by all group members to assure that the group key is transferred from a legitimate initiator. In our protocol, the initiator needs to select a random group key. Then, the initiator constructs a polynomial passing through one-time DH shared keys with group members and the randomly selected group key. The initiator uses the secret sharing scheme as encryption tool to broadcast message to all group members. Later, each group member can decrypt the group key individually without interaction with other members. Since polynomial computation uses a smaller modulus (say 160 bits only) as compared with public-key computations using a larger modulus (say 1024 bits at least), polynomial computations are much faster than public-key computations. Thus, our protocol is more efficient than most public-key-based key distribution protocols.

The number of communication rounds is another important factor, which affects the efficiency of a protocol. Since our protocol is non-interactive, it is very efficient.

2.2. Type of attackers

2.2.1. *Insider attacker*

The inside attacker is a legitimate member who knows the group key. But inside attacker may try to recover other members' secrets (long-term private keys). After knowing each long-term private key, the inside attacker is able to reveal other group keys that he is not authorized to know or the attacker is able to impersonate other members in a secure group communication.

2.2.2. *Outsider attacker*

The outside attackers are non-members of a communication group. But outside attackers may try to recover the secret group key that he is unauthorized to know and then use the group key to wiretap the content of a secret group communication. This attack is related to the secrecy of group keys. In general, outside attackers can act either passively or actively. An active attacker can try to impersonate to be a group member participated in the key establishment scheme without being detected. Since a public-key digital certificate is needed for each member, our proposed scheme can prevent active attack. On the other hand, an attacker can try to derive the group key passively from the public information transmitted during the key establishment process. We will prove it later in Section 4 that our proposed scheme can prevent this type of attack as well.

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

2.3. Security of group keys

A sequence of group keys is denoted as $K = \{K_i | i = 1,2,\ldots,n\}$. We consider the following security objectives of group keys:

(a) *Key secrecy*: It is computationally infeasible for any outside attacker to discover any group key, U_i.
(b) *Key authentication*: Group members can authenticate the recovered group key, which is transmitted from an initiator.

(c) *Key independence*: Knowing a subset of group keys, $K' \subset K$, any unauthorized user cannot discover any other group keys, $K'' = K - K'$.

3. PROPOSED PROTOCOL

3.1. Notations

For ease of reference, Fig. 2 lists the notations used in this paper.

3.2. System setup

Our protocol needs following parameters:

- Let p be a large prime (say 1024 bits) where $p - 1$ contains a small prime factor q (say 160 bits and $q|p - 1$).
- Let g be a generator of Z_q. The public key of user U_i is $y_i = g^{x_i} \bmod p$, where $x_i \in Z_q$, is the private key of user, U_i.

Note that under this arrangement, the DH public-key computations are performed in Z_p, which is the field of a large modulus and the polynomial computations are performed in Z_q, which is the field of a smaller modulus.

We consider a set of n users, $U = \{U_i | U_i \in U, i = 1,2,\ldots,n\}$. Each user U_i follows DH public-key scheme to select a private key, $x_i \in Z_q$, and compute the public key, $y_i = g^{x_i} \bmod p$. Note that each public key needs to be digitally signed by a CA to generate a digital certificate. The digital certificate of each user is used to authenticate the ownership of the public key of the user.

3.3. Encryption of the group key

We assume that an initiator in the set, U, wants to transmit a secret group key, $K \in Z_q$, to a subset of l group members, $\{U_{r_i} | U_{r_i} \in U, i = 1, 2, \ldots, l\}$, where $1 \le l \le n$, in the set, U. The proposed scheme is described in Fig. 3.

4. ANALYSIS

4.1. Correctness

It is obvious that in Step 2 of the encryption and in Step 1 of the decryption processes, we have $k_{r_i} = \{y_{r_i}^{x_s+s} \bmod p\} \bmod q = \{(y_s \cdot r)^{x_{r_i}} \bmod p\} \bmod q$. Since the degree of polynomial, $f(x)$, is l, knowing the point, (ID_{r_i}, k_{r_i}), and l public values, $\{(i,f(i)) | i = 1,2,\ldots,l\}$, on the polynomial, $f(x)$, each U_{r_i} can follow the Lagrange interpolation formula to compute the polynomial, $f(x)$, as

$$k_{r_i} \prod_{j=1,}^{l} \frac{x - j}{ID_{r_i} - j} + \sum_{j=1}^{l} f(j) \frac{x - ID_{r_i}}{j - ID_{r_i}} \prod_{v=1, v \neq j}^{l} \frac{x - v}{j - v} \bmod q.$$

In Step 2, since $K = f(0)$, we have

$$f(0) = k_{r_i} \prod_{j=1,}^{l} \frac{-j}{ID_{r_i} - j} + \sum_{j=1}^{l} f(j) \frac{0 - ID_{r_i}}{j - ID_{r_i}} \prod_{v=1, v \neq j}^{l} \frac{0 - v}{j - v} \bmod q = K.$$

4.2. Security

In this section, we prove that the polynomial encryption in our protocol is unconditionally secure and the protocol provides key secrecy, key authentication and key independence.

Notations	Meanings
p	a large prime (say 1024 bits) where $p-1$ contains a small prime factor q
q	a small prime (say 160 bits and $q\|p-1$)
g	a generator
x_i	DH private key (say 160 bits)
y_i	DH public key (say 1024 bits)
K,	a group key (say 160 bits)
s	a one-time secret (say 160 bits)
r	a one-time public value (say 1034 bits)
k_{r_i}	a one-time shared secret (say 1034 bits)
TS	time stamp
U_i	a communication group
ID_{r_i}	public information with the member, U_{r_i}

FIGURE 2. Notations.

Step 1. The initiator with his private and public keys, (x_s, y_s), selects a one-time random secret, $s \in Z_q$, and computes, $r = g^s \bmod p$.

Step 2. The initiator accesses public keys of all group members, $\{y_{r_i} | i = 1,2,...,l\}$, and computes one-time shared keys, $k_{r_i} = \{y_{r_i}^{x_s+s} \bmod p\} \bmod q, i = 1,2,...,l$.

Step 3. The initiator solves a polynomial, $f(x)$, passing through $l+1$ points, $\{(0,K),(ID_{r_i}, k_{r_i}) | i = 1,2,...,l\}$, where ID_{r_i} is the public information associated with the member, U_{r_i}. Following the Lagrange interpolation formula, the polynomial, $f(x)$, can be computed as $f(x) = K\prod_{j=1,}^{l} \frac{x - ID_{r_j}}{-ID_{r_j}} + \sum_{i-1}^{l} k_{r_i} \frac{x}{ID_{r_i}} \prod_{j=1, j\neq i}^{l} \frac{x - ID_{r_j}}{ID_{r_i} - ID_{r_j}} \bmod q$. Note that the degree of polynomial, $f(x)$, is l.

Step 4. The initiator computes l public values, $\{f(i), i = 1,2,...,l\}$, where $i \neq ID_{r_i}, i = 1,2,...,l$, and $h(TS\|K)$, where TS is the time stamp and $h(TS\|K)$ is the one-way function of the concatenation of the time stamp, TS, and the group key, K. The initiator broadcasts $\{r, TS, h(TS\|K), (i, f(i) | i = 1,2,...,l)\}$, to all group members.

Decryption of the group key

Step 1. Each group member, U_{r_i}, in the subset uses his private key, x_{r_i}, to compute $k_{r_i} = \{(y_s \cdot r)^{x_{r_i}} \bmod p\} \bmod q$.

Step 2. Knowing the private value, k_{r_i}, and l public values, $f(i), i = 1,2,...,l$, each U_{r_i} can obtain the secret group key, K', by computing $k_{r_i}\prod_{j=1,}^{l} \frac{-j}{ID_{r_i} - j} + \sum_{j=1}^{l} f(j) \frac{0 - ID_{r_i}}{j - ID_{r_i}} \prod_{v=1, v\neq j}^{l} \frac{0 - v}{j - v} \bmod q = K'$.

Step 3. Each U_{r_i} verifies the time stamp, TS, and checks whether $h(TS\|K') \stackrel{?}{=} h(TS\|K)$. If it is verified, the group key, K, is authenticated. Therefore, K is used for the group communication.

FIGURE 3. Proposed scheme.

THEOREM 1. *The security of polynomial encryption in proposed protocol is unconditionally secure.*

Proof. The degree of the polynomial, $f(x)$, is l. For each group member (i.e. receiver), U_{r_i} knows l public points, $\{(i, f(i)) | i = 1,2,\ldots,l\}$, and one additional shared secret point, (ID_{r_i}, k_{r_i}), on the polynomial, $f(x)$, U_{r_i} has enough information to recover the polynomial, $f(x)$, and thus obtains $f(0)$. However, there are only l public points, $\{(i, f(i)) | i = 1,2,\ldots,l\}$, available for any outside attacker. This information is insufficient to recover the polynomial, $f(x)$. The security of this polynomial encryption of a group key does not depend on any computational assumption.

Let us consider the following insider attack. After a group communication, any legitimate member knows the group key and the polynomial, $f(x)$. Thus, any group member also obtains all shared keys, $k_{r_i} = \{y_{r_i}^{x_s+s} \bmod p\} \bmod q$, $i = 1, 2, \ldots, l$. However, since each shared key is only a one-time secret, knowing any one-time shared key does not reveal any secret of other communication sessions. □

THEOREM 2. *The proposed protocol can provide key secrecy, key authentication and key independence.*

Proof. Key secrecy: In Step 1 of the Decryption process, each group member, U_{r_i}, uses his long-term private key, x_{r_i},to compute the shared key $k_{r_i} = \{y_{r_i}^{x_s+s} \bmod p\} \bmod q$. This shared point, (ID_{r_i}, k_{r_i}),on the polynomial, $f(x)$, assures that only group member can recover the secret group key, $K = f(0)$.

Key authentication: Note that the polynomial, $f(x)$, used to generate the broadcast ciphertext is a function of the shared keys, $k_{r_i} = \{y_{r_i}^{x_s+s} \bmod p\} \bmod q$, $i = 1, 2, \ldots, l$. Each shared key, $k_{r_i} = \{y_{r_i}^{x_s+s} \bmod p\} \bmod q = \{(y_s \cdot r)^{x_{r_i}} \bmod p\} \bmod q$, can only be computed either by the initiator, U_s, who knows the long-term private key, x_s and one-time secret, s, or by the group member, U_{r_i}, who knows the long-term private key, x_{r_i}. Since each group member, U_{r_i}, did not compute the broadcast ciphertext, $\{\mathrm{TS}, h(\mathrm{TS}\|K), (i, f(i) | i = 1, 2, \ldots, l)\}$, the ciphertext must be generated by the initiator, U_s. Thus, the group key can be authenticated by each group member, U_{r_i}, if $h(\mathrm{TS}\|K') \stackrel{?}{=} h(\mathrm{TS}\|K)$ in Step 3 of Decryption process.

Key independence: It is obvious that the group key will be different for each communication session since each group key is a function of a random integer, $s \in Z_q$, chosen by the initiator as shown in Step 1 of Encryption process. □

4.3. Properties

In the following, we summarize properties of our proposed protocol.

(1) Our group key establishment protocol is a non-interactive protocol, which can provide key secrecy and key authentication.
(2) The proposed protocol is a hybrid of DH public-key scheme and the secret sharing scheme. The DH scheme enables a pairwise one-time shared key between any pair of users, which eliminates the need of a mutually trusted KGC. The secret sharing encryption with a smaller modulus is computationally faster than public-key-based broadcast encryption with a larger modulus.
(3) In decryption of the group key, there is no need of any interaction among group members. In other words, each member can decrypt the group key individually without the assistance of other members.
(4) The security of the secret sharing encryption is unconditionally secure.

5. PERFORMANCE AND COMPARISON

We discuss the performance of our protocol in terms of storage requirement, communication and computational costs and membership changes.

5.1. Storage requirement

The memory storage of each user is only the DH private key and the public-key digital certificate. This requirement is the same as any public-key cryptographic algorithm. There is no centralized KGC existed in our protocol to register users.

5.2. Communication cost

In our protocol, the initiator needs to broadcast a message to all group members at once. There is no interaction among group members in order to recover the group key.

5.3. Computational cost of initiator

The initiator of a group communication needs to compute $l + 1$ modular exponentiations as specified in Steps 1 and 2, where $l + 1$ is the number of group members in the group communication. Then, in Step 3, the initiator needs to compute a polynomial interpolation passing through $l + 1$ points. For each point, there has to compute l multiplications. However, these multiplications are very fast since the operands are differences of users' IDs (say 20 bits). In Step 4, the initiator needs to evaluate l different polynomials. Horner's rule [31] can be used to evaluate polynomials. From Horner's rule, evaluating a polynomial of degree l needs l multiplications and $l + 1$ additions. We can ignore the computational time needed for polynomial computations as compared with the computational time needed for modular exponentiations. This is due to the fact that the polynomial computations are executed in a smaller modulus q as compared with the modular exponentiations in a larger modulus p. In addition, each public-key modular exponentiation requires $\sim 1.5 \log_2 p$ multiplications. Overall, the computational cost to establish a group key with $l + 1$ members, the initiator needs to compute $l + 1$ modular exponentiations. This computational cost is the minimum among all existing public-key-based group key establishment protocols.

5.4. Computational cost of each receiver

Each receiver (group member) of a group communication needs to compute one modular exponentiation as specified in Step 1. Then, in Step 2, the receiver needs to compute a polynomial interpolation passing through $l + 1$ points. For each passing point, it has to compute l multiplications. However, these multiplications are very fast since operands are differences of users' IDs. Similarly, we can ignore the computational time needed for the polynomial computation in Step 2 as compared it with the computational time needed for the modular exponentiation in Step 1. Overall, the computational cost to recover a group key with $l + 1$ members, the group member needs to compute one modular exponentiation. The computational cost of our protocol is the minimum among all existing public-key-based group key establishment protocols.

5.5. Membership change

If a new user joins the group communication, the new user can just submit his public-key to the CA and makes his public-key digital certificate available to the public without affecting certificates of existing users. On the other hand, if there is departing user from the application, the departing user only to revoke his digital certificate through the CA.

In Table 1, we have compared our proposed scheme with two other schemes [10, 19], one scheme [10] is a distributed group DH protocol using the secret sharing scheme and the other scheme [19] is a centralized group key establishment scheme using secret sharing scheme with RSA modulus.

TABLE 1. Comparison among our scheme and other related schemes.

	KGC	Computational complexity	Communication complexity	Storage requirement
Proposed scheme	No	Initiator—n modular exponentiations Others—one modular exponentiation • Small modulus (160 bits) in polynomial evaluation	One broadcast transmission	One private key and one public-key certificate
Harn and Lin [19]	Yes	No modular exponentiation • Large modulus (1024 bits) in polynomial evaluation	Three round transmissions	One coordinate point
Harn and Lin [10]	No	Each user—$2n$ modular exponentiations	Three round transmissions	One private key and one public-key certificate

Since the modular exponentiation is the most time-consuming operation, we only include the number of modular exponentiations needed in the comparison.

6. CONCLUSION

We have proposed a non-interactive group key establishment protocol with properties of both key confidentiality and key authentication. Our protocol is a hybrid of the DH public-key scheme and the secret sharing scheme, which takes advantages of both schemes by removing the need of a centralized KGC and by reducing computational cost. We believe that our approach opens a new research direction in the design of group key establishment protocols.

REFERENCES

[1] Diffie, W. and Hellman, M.E. (1976) New directions in cryptography. *IEEE Trans. Inform. Theory*, **IT-22**, 644–654.

[2] Ingemarsson, I., Tang, D.T. and Wong, C.K. (1982) A conference key distribution system. *IEEE Trans. Inform. Theory*, **IT-28**(5), 714–720.

[3] Steer, D.G., Strawczynski, L., Diffie, W. and Wiener, M.J. (1988) A Secure Audio Teleconference System. In *Proc. Crypto '88*, LNCS, **Vol. 403**, pp. 520–528.

[4] Burmester, M. and Desmedt, Y. (1995) A Secure and Efficient Conference Key Distribution System. In *Proc. Eurocrypt '94*, LNCS, **Vol. 950**, pp. 275–286.

[5] Steiner, M., Tsudik, G. and Waidner, M. (1996) Diffie–Hellman Key Distribution Extended to Group Communication. In *Proc. 3rd ACM Conf. Computer and Comm. Security (CCS '96)*, pp. 31–37.

[6] Bresson, E., Chevassut, O., Pointcheval, D. and Quisquater, J.-J. (2001) Provably Authenticated Group Diffie–Hellman Key Exchange. In *Proc. ACM Conf. Computer and Comm. Security (CCS '01)*, pp. 255–264.

[7] Bohli, J.M. (2006) A Framework for Robust Group Key Agreement. In *Proc. Int'l Conf. Computational Science and Applications (ICCSA '06)*, LNCS, **Vol. 3982**, pp. 355–364.

[8] Bresson, E., Chevassut, O. and Pointcheval, D. (2007) Provably-secure authenticated group Diffie–Hellman key exchange. *ACM Trans. Inform. Syst. Secur.*, **10**, 255–264.

[9] Katz, J. and Yung, M. (2007) Scalable protocols for authenticated group key exchange. *J. Cryptol.*, **20**, 85–113.

[10] Harn, L. and Lin, C (2014) Efficient group Diffie–Hellman key agreement protocols. *Comput. Electric Eng.*, **40**, 1972–1980.

[11] 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.

[12] Laih, C., Lee, J. and Harn, L. (1989) A new threshold scheme and its application in designing the conference key distribution cryptosystem. *Inform. Proces. Lett.*, **32**, 95–99.

[13] Berkovits, S. (1991) How to Broadcast a Secret. In *Proc. Eurocrypt '91*, LCNS, **Vol. 547**, pp. 536–541.

[14] Li, C.H. and Pieprzyk, J. (1999) Conference Key Agreement From Secret Sharing. In *Proc. 4th Australasian Conf. Information Security and Privacy (ACISP '99)*, LNCS, **Vol. 1587**, pp. 64–76.

[15] Saze, G. (2003) Generation of key predistribution schemes using secret sharing schemes. *Discrete Appl. Math*, **128**, 239–249.

[16] Harn, L. and Gong, G. (2015) Conference key establishment protocol using a multivariate polynomial and its applications. *Secu. Commun. Netw.*, **8**, 1794–1800.

[17] Harn, L. and Xu, C.F. (2015) Predistribution scheme for establishing group keys in wireless wensor networks. *IEEE Sens. J.*, **15**, 5103–5108.

[18] Harn, L. and Xu, C.F. (2016) New design of secure end-to-end routing protocol in wireless sensor networks. *IEEE Sens J.*, **16**, 1779–1785.

[19] Harn, L. and Lin, C. (2010) Authenticated group key transfer protocol based on secret sharing. *IEEE Trans. Comput.*, **59**, 842–846.

[20] Xu, C.F., Harn, L., He, T. and Zhang, M. (2016) Efficient group key transfer protocol for WSNs. *IEEE Sens. J.*, **16**, 4515–4520.

[21] Harn, L. and Xu, C.F. Centralized group key establishment protocol without mutually trusted party. *Mobile Netw. Appl.*. doi:10.1007/s11036-016-0776-7.

[22] Berkovits, S. (1991) How to Broadcast a Secret. In *Proc. Eurocrypt'91*, Brighton, UK, April, LNCS 547, pp. 535–541. Springer

[23] Fiat, A. and Naor, M. (1994) Broadcast Encryption. In *Proc. Crypto '94*, California, USA, August, LNCS 839, pp. 480–491. Springer.

[24] Bellare, M., Boldyreva, A. and Micali, S. (2000) Public-Key Encryption in a Multi-User Setting: Security Proofs and Improvements. In *Proc. Eurocrypt 2000*, Bruges, Belgium, May, LNCS 1807, pp.259–274. Springer.

[25] Dodis, Y. and Fazio, N. (2002) Public Key Broadcast Encryption for Stateless Receivers. In *Proc. Digital Rights Management 2002*, Washington, DC, USA, LNCS 2696, pp. 61–80. Springer.

[26] Kurosawa, K. (2002) Multi-Recipient Public-Key Encryption With Shortened Ciphertext. In *Proc. Public Key Cryptography*, Pairs, France, February, LNCS 2274, pp. 48–63. Springer

[27] Fan, C.I., Huang, L. and Ho, P.H. (2010) Anonymous multi-receiver identity-based encryption. *IEEE Trans. Comput.*, **59**, 1239–1249.

[28] Wang, H., Zhang, Y., Xiong, H. and Qin, B. (2012) Cryptanalysis and improvements of an anonymous multi-receiver identity-based encryption scheme. *IET Inform. Secur.*, **6**, 20–27.

[29] Harn, L., Chang, C.-C. and Wu, H.-L. (2013) An anonymous multi-receiver encryption based on RSA. *J. Netw. Secur.*, **15**, 307–312.

[30] Wu, Q., Qin, B., Zhang, L., Domingo-Ferrer, J. and Manjón, J. A. (2013) Fast transmission to remote cooperative groups: a new key management paradigm. *IEEE/ACM Trans. Netw.*, **21**, 621–633.

[31] Donald, K. (1998) The Art of Computer Programming, Volume 2: Semi Numerical Algorithms. Addison-Wesley.