Anonymous and Dynamic Conference-key Distribution system

MaoJane
National Key Lab. Of ISN, Xidian Univ., Xian , China

Yangbo
National Key Lab. Of ISN, Xidian Univ., Xian , China

Abstract—**In order to hold secure electronic conference in communication networks via insecure channels, a conference key distribution system should be constructed. The conference key distribution system (CKDS) is used for distributing a conference key shared among the participants of the conference and hence secure communications are achieved. In this paper, by using the secret sharing scheme based on the MDS code and the Diffie-Hellman Key exchange scheme as the basic component, we propose an efficient and anonymous conference-key distribution scheme that supports conference membership changes dynamically. We also show that, based on the Diffie-Hellman (DH) and the one-way assumption, the proposed CKDS is secure against impersonation and conspiracy attacks, and the unattended ones reveal no useful knowledge about the conference key. In addition, the proposed CKDS allows for user anonymity.**

Keywords: secret sharing, broadcast channel, MDS codes, key distribution

I. INTRODUCTION

Key distribution is a major component of the security subsystem of distributed systems, communication systems, and data networks. With the increase in bandwidth, size, usage, and applications of such systems, holding electronic conference among a group of users becomes a growing application area in network. In order to hold secure electronic conference in computer and communication networks via insecure channels, a conference key distribution system (CKDS) should be established. The system is used for distributing a conference key shared among the attending users of the conference to let them use the conference key to encrypt/decrypt communicating messages during the conference, while unattended members reveal no useful knowledge about the conference key, and hence secure communications are achieved. From the point of view of security and efficiency, an electronic conference should be equipped with a point-to-multipoint environment, such as broadcasting channel (BC channel), in that anyone connected to the network will have access to all the information that flows through it.

Sometimes, for instance, the conference may have discussions on privacy or highly sensitive topics. To have the attending participants make unbiased decisions without influence from unknown pressures, the identities of the attending participants should be anonymous to each other, and even anonymous to the unattended one except for the conference organizer.

In many cases, the conference membership changes dynamically, i.e., some new members want to take part in while some old members want to leave. Thus the conference key shall change dynamically to ensure both forward secrecy and backward secrecy of the conference session. The forward secrecy is maintained if an old member who has been excluded from the current session cannot access the communication of the current session, and the backward secrecy is guaranteed if a new member of the current session cannot recover the communication of past sessions. This requires conference key should to be dynamically distributed to current conference members.

In 1976, Diffie and Hellman [1] proposed a KDS for distributing a common secret key shared between two communicating participants. The Diffie-Hellman KDS is suitable for the point-to-point environment in essence. However, it may be inefficient to iteratively employ the original Diffie-Hellman KDS for distributing a conference key shared among a group of communicating participants in the point-to-multipoint environment, since several interactions for point-to-point protocols with respect to each pair of principals are required. Since then, several CKDSs have been proposed in the literature [2-17]. However, most of the previously proposed CKDSs are performed through modular exponentiation, which makes them inefficient in practical usage, especially in the case that the participants have less computing power. Except for the secure lock schemes in [8,15], the CKDSs cited above cannot achieve user anonymity. However, these secure lock schemes suffer from the sophisticated computational complexities of the Chinese remainder theorem (CRT) and encryption/decryption algorithms, and are often not acceptable in the applications with medium or large number of participants. Therefore, the design of an efficient CKDS that is suitable for broadcasting channels and allowing for user anonymity is required.

In this paper, we propose an efficient CKDS with user anonymity based on the MDS code and the CKDS supports conference membership changes dynamically. We also show that under the assumption of Diffie-Hellman problem (DHP) and the one-way hash (OWH), the proposed CKDS is secure against impersonation and conspiracy attacks.

Supported by NSFC No.60073052

II. MODEL

A user in the system is a probabilistic polynomial-time Turing machine. The system has a public directory that records the system's public parameters and each user's public information that can be accessed by everyone. All users are connected on an authenticated broadcast network so that the messages sent on the network can be identified and cannot be altered, blocked, or delayed. Therefore, everyone can send and receive the message on the network without interruption. No private channel exists between users. Conference organizer (U_T) distributes the conference key to the participants of the conference.

There are two attacks on the CKDS,

- Impersonation attack: Single adversary replays the intercepted broadcast message for originating a conference by the name of U_T.
- Conspiratorial attack: Conspiratorial participants replay the intercepted broadcast message for originating a conference by the name of U_T.

Our CKDS has the following three components:

1. **System initialization:** System and users select related parameters and send public parameters to the public directory.
2. **Conference key distribution:** Conference organizer selects conference key K and shares the conference key with all participants by means of the secret sharing scheme under the broadcast channel. Every participant can recover the conference key from the broadcast message and his/her private information, while unattended ones will get no information from the broadcast message.
3. **Conference key recovery:** Participants evaluate and verify the conference key from the received message and their private information.

In this paper, we use MDS code as basic component of the secret sharing scheme [18][19][20].

III. OUR PROTOCOL

Preliminaries

Let G denote the generator matrix of a (n,k) linear code over $GF(q)$. Here n denotes the length of the codewords and k is the dimension of the code, i.e. G is a $(n \times k)$ matrix with elements in $GF(q)$, q is a prime power. The number of codewords is q^k and the set of codewords consists of all linear combinations of the rows of G. If $\vec{d} \in GF\ (q^k)$ denotes a vector of k information symbols, they will be encoded into $\vec{c} = \vec{d}G$.

A MDS code is usually defined by the condition that the minimum distance of the code is $n-k+1$. Reader could see [25] for a general introduction to MDS codes. From the property that every set of k columns in the generator matrix of a MDS code is independent it follows that a codeword is uniquely determined by any k elements in the codeword. It also follows that knowledge of less than k elements of a codeword reveals no information about another element.

Basic Protocol

C1. System initialization:

Let m be the number of users in the system, and ID_i be the identity of the users U_i. Denote by $|x|$ for the bit length of x. Publish a one-way hash function $H(x)$, which accepts a variable-length input string of bits and produces a fixed-length output string of bits (for example, 128 bits). Define and publish the following system parameters:

p is a large prime and $p > 2^{512}$

q is a prime factor of $p-1$ and $q > 2^{|H(\cdot)|}$

g is a generator with order q in $GF(p)$

User U_i selects private key $x_i \in Z_q^*$ (kept secret) and computes $y_i = g^{x_i} \bmod q$ (published).

Without loss of generality, let U_T be the conference organizer, and $\overline{U} = \{U_1, U_2, \cdots, U_m\}$ be the set of all users in the system.

U_T selects and publishes the matrix G, which is the generator matrix of $(N, m+1)-MDS$ codes (where, $q \geq N \geq m+1$) over $GF(2^{|q|})$ (In our protocol, we use system code). Obviously, $2^{|q|} > N$, so suitable codes will always exist.

For $\forall i, 1 \leq i \leq m$, U_T Computes $k_{ti} = y_i^{x_t} \bmod p$; U_i Compute $k_{it} = y_t^{x_i} \bmod p$.

C2. Conference key distribution:

Let $P = \{U_1, U_2, \cdots U_n\}$ be the set of attending members and $F = \{U_{n+1}, U_{n+2}, \cdots, U_m\}$ be the set of unattended members of the conference.

U_T performs the following steps:

Step1: Let $Y = \{x | x = H(k_{ti} \| ID_l), l \in \{1,2,...n\}\}$; Construct an one-one correspondence $f: Y \to \{1,2,\cdots n\}$, and publish f.

Step2: Get a timestamp T from the system.

Step3: Randomly select a conference key $K \in Z_q^*$.

Step4: Construct $D = (K, d_1, d_2, \cdots, d_i, \cdots, d_n, \cdots, d_m)$, where $d_i = H(k_{ti} \| ID_T \| ID_i \| T)$ $(1 \leq i \leq n)$ and t satisfies that $i = f(H(k_{ti} \| ID_t))$; the last (m-n) symbols of D are random elements in $G(q)$.

Step5: Compute $B = DG = (K, d_1, d_2, \cdots d_m, b_1, b_2, \cdots b_{N-m-1})$ (According to the definition of system codes, the first n+1 symbols of the codeword remain constant), and get

$\vec{b} = (b_1, b_2, \cdots, b_n)$ from B.

Step6: Compute the characteristic value of K at T as $V = H(K \| ID_T \| T)$.

Step7: Broadcast $\{ID_T, \vec{b}, V, T\}$.

C3. Conference key recovery:

On receiving $\{ID_T, \vec{b}, V, T\}$, each U_i performs the following steps for recovering the conference key K:

Step1: Check the expiration of the received timestamp T. If it is out-of-date then the broadcast message may be replayed by some malicious attacker, and hence terminate the recovery stage.

Step2: Compute $l = f(H(k_{il} \| ID_i))$;

Step3: Compute $d_l = H(k_{il} \| ID_T \| ID_i \| T)$;

Step4: According to the definition of $(N, m+1) - MDS$ codes, $B = (K, d_1, d_2, \cdots, d_n, b_1, b_2, \cdots b_{N-m-1})$ can be recovered from $(d_l, b_1, b_2, \cdots, b_n)$ and G by means of the decoding algorithm. Then get the conference key K.

Step5: Check the attendance of the conference by verifying that $H(K \| ID_T \| T) = V$.

Dynamic protocol

In the case that new members join or old members leave, protocol performs as following:

- Updates set P and set F.
- U_T performs C2.
- Users in set P perform C3.

IV. SECURITY ANALYSIS

In this chapter, we show security of the above protocol in correctness, and withstanding the attack of passive adversaries and impersonators.

4.1 Correctness and Security against Passive Attacks

Theorem 1 (Correctness): If attending members follow the protocol, they compute a common conference key K.

Proof. From the broadcast message $\forall U_i \in P$ can compute:

$k_{il} = k_{li} = g^{x_i x_l} \bmod p = g^{x_l x_i} \bmod p = y_l^{x_i} \bmod p = y_i^{x_l} \bmod p$;

$j = f(H(k_{il} \| ID_i))$;

$d_j = H(k_{il} \| ID_T \| ID_i \| T)$;—the j-th symbol of codeword B

According to the property of MDS codes, the U_i can uniquely determine the codeword B from $\vec{b} = (b_1, \cdots, b_n)$ and d_j. The $n+1$-th symbol of B is the conference key that we need. □

Diffie-Hellman Problem(DHP)[22][23]: Let G be a finite abelian group, $a, b \in N, g \in G$. Given g^a, g^b compute g^{ab}.

Lemma 1 (DHP assumption): Any adversary cannot compromise the principals' secret keys and their common secret keys form the public keys.

Lemma 2 (OWH assumption): It is infeasible to find x such that $H(x) = y$ for a given y, and it is infeasible to find a pair (x, x') such that $x \neq x'$ and $H(x) = H(x')$.

Theorem 3 (The passive attack): An eavesdropper cannot obtain the conference key K from the intercepted message.

Proof: Under DHP assumption, given $U_i \in P$, $\forall U_j \notin P$ cannot obtain x_T, x_i, k_{ti}. So U_j cannot compute $d_i = H(k_{ti} \| ID_T \| ID_i \| T)$.

U_j can only get $\vec{b} = (b_1, b_2, \cdots, b_n)$ from the intercepted message. According to the definition of $(2n+1, n+1) - MDS$ codes, U_j get no information about the codeword B. By lemma 2, it is infeasible for U_j to compute K from $V = H(K \| ID_i \| T)$. So U_j cannot obtain the conference key from the intercepted message. □

Theorem 4 (Anonymity): The identities of the attending participants are anonymous to each other, and even anonymous to the unattended one except for the conference organizer.

Proof: By lemma 1 and lemma 2, $\forall U_i \in \overline{U}/U_T$ cannot obtain the identity of the other users from $Y = \{x | x = H(k_{il} \| ID_l), l \in U_n\}$. □

4.2 Security analysis

In this section, we will show that under the assumption of Diffie-Hellman problem (DHP) and the one-way hash (OWH), the proposed CKDS is secure against impersonation and conspiracy attacks.

Theorem 4 (The impersonation attack): Any adversary cannot successfully replay the intercepted message $\{ID_T, \vec{b}, V, T\}$ for originating a conference by the name of U_T.

Proof: The expiration of T will be effectively verified by step1 of the conference key recovery stage. To pass the verification of T, the adversary should set a new acceptable T. Consequently, the adversary should forge all valid $d_i = H(k_{il} \| ID_T \| ID_i \| T)$ for the attending members, so that the attending members can recover the right K from $\vec{b}, G$. Under the OWH assumption, the adversary can forge all valid d_i only if he knows the x_is or k_{ti}s. However, by lemma1, the x_is and k_{ti}s are protected under the DH assumption. □

Theorem 5 (The conspiratorial impersonation attack): Conspiratorial participants cannot successfully replay the intercepted message $\{ID_T, \vec{b}, V, T\}$ for originating a conference by the name of U_T.

The proof is the same as the Theorem 4.

V. PERFORMANCE ANALYSIS

The complexity of a CKDS includes communication complexity, storage complexity and computation complexity. The communication complexity is usually measured by the number of data bits that need to be transmitted from the U_T to conference members to covey information of conference keys, while the storage complexity is measured by the number of data bits that the U_T and conference members need to store to obtain conference keys, and the computation complexity is measured by the computation the U_T and conference members need to perform to derive conference keys.

Here we show the performance of our protocol:

- **Communication complexity:** In order to distribute the conference key, U_T should broadcast ID_T and m symbols of the codeword B; U_T should broadcast n hash values to ensure the anonymity of the conference; and U_T should broadcast timestamp T to against active attack. So the number of bits that U_T should broadcast is $m|q|+n|H(\cdot)|+|T|+|ID_i|$.
- **Storage complexity:** U_T needs to store its secret key x_t and the share with the system members $k_{ti},(U_i \in \overline{U})$; the current conference member U_i only needs to store his/her secret key x_i and the share with U_T. So U_T should store $m|p|+|q|$ bits; and conference member U_i needs to store $|q|+|p|$.
- **Computation complexity:** In the initialization phase, U_T needs m modular exponentiation operations and the current conference member U_i only need one modular exponentiation operations. In the key distribution phase, U_T needs 2n hashing operations and one encoding operation. In the key recovery phase, U_i needs one hashing operation and one decoding operation.

Both the encoding and decoding operations for a $(N, m+1)-MDS$ code only need $O(m^2)$ arithmetic operations if standard encoding and erasure decoding algorithms are used. Fast encoding and decoding algorithms only need $O(m \log m)$ operations [26].

VI. CONCLUSION

In this paper, we propose an efficient and practical CKDS with user anonymity based on the MDS codes. Under the Diffie-Hellman (DH) assumption and the one-way hash (OWH) cryptographic assumptions, we also show that the proposed CKDS can withstand passive attack, impersonation attack and conspiracy attack. In addition, our protocol hides the identities of the attending members efficiently by means of one-way hash function and a one-one correspondence, So that the attending participants can make unbiased decisions without influence from unknown pressures. The protocol distributes conference key to current conference members dynamically with low additional overhead and ensures both forward secrecy and backward secrecy of the conference sessions.

Our protocol is efficient because only two rounds are required to compute a common conference key if all attending members follow the protocol properly. Nevertheless, the size of messages that each participant sends is proportional to the number of attending members. It is interesting to design a provable secure conference key distribution protocol with both round and message efficiency.

REFERENCES

[1] Diffie,W., and Hellman, M.E.: "New directions in cryptography", IEEE Trans. Info. Theory, 1976, 22, (6), pp.644-654

[2] Ingemarsson, I.,Tang, T.d., and Wong, C.K.: "A conference key distribution system", IEEE Trans. Inf. Theory, 1982, 28, (5), pp. 714-720

[3] Berkovits, S.: "How to broadcast a secret". Advances in Cryptology-EUROCRYPT'91, 1991, pp. 535-541

[4] Chang, C.C., Wu, T.C., and Chen, C.P.: "The design of a conference key distribution system". Advances in Cryptology- AUSCRYPT'92, 1992, pp.459-466

[5] Chen, J.L., and Hwang, T.: "Identity-based conference key broadcast schemes with user authentication", Comput. Secrutity, 1994, 13, (1), pp.53-57

[6] Chikazawa, T., and Inoue, T.: "A new key sharing system for global telecommunications". Proc. GLOBECOM'90, 1990, pp.1069-1072

[7] Chikazawa, T., and Yamagishi, A.: "Improved identity-based key sharing system or multiaddress", Electron. Lett., 1992, 28, (11), pp.1015-1017

[8] Chiou, G.H., and Chen. W. T.: "Secure broadcasting using the secure lock", IEEE Trans. Softw. Eng., 1989, 15, (8), pp. 929-934

[9] Hwang, M. S., and Yang, W.P.: "Conference key distribution schemes for secure digital mobile communications", IEEE J. Sel. Areas commun., 1995, 13, (2), pp.416-420

[10] Hwang, T., and Chen, J.L.: "Identity-based conference key broadcast systems", IEE Proc. Comput. Digit. Tech., 1994, 141, (1), pp. 57-60

[11] Koyama, K.: "Identity-based conference key distribution system", Advances in Cryptology-EUROCRYPT'87, 1987, pp. 175-184

[12] Koyama, K., and Ohta, K.: "Identity-based conference key distribution system", Advances in Cryptology-EUROCRYPT'87, 1987, pp. 175-184

[13] Koyama, K., and Ohta, K.: "Security of improved identity-based conference key distribution system", Advances in Cryptology-EUROCRYPT'88, 1988, pp. 11-19

[14] Laih, C.S., Harn, L., and LEE. J.Y.: "A new threshold scheme and its application in designing the conference key distribution system", Inf. Process. Lett., 1989, 32, (3), pp. 95-99

[15] Lin, C.H., Chang, C.C., and Lee, R.C.T.: "A conference key broadcasting system using scaled locks", Inf. Syst., 1992, 17, (4), pp. 323-328

[16] Wu, T.C., and Yeh, Y.S.: "A conference key distribution system based on cross-product", Comput. Math. Appl., 1993, 25, (4), pp. 39-46

[17] Wu, T.C.: "Conference key distribution system with user anonymity based on algebraic approach", IEE Proc.: Computers and Digital Techniques, vol. 144, no. 2, pp.145-148,1997

[18] Ingemarsson, I. And Simmons, G.J., "A protocol to Set up Shared Secret Schemes without the Assistance of a Mutually Trusted Party," Proc. Advances in Cryptology-Eurocrypt'90,

[19] F.J.MacWilliams and N.J.A.Sloane, The Theory of Error Correcting Codes, North-Holland, New York, 1977.

[20] R.Blom, "An Optimal Class of Symmetric Key Generation Systems,"Proc. Advances in Cryptology-Cryto'84, pp.335-338, 1985.

[21] Merkle, R.C.: "One way hash functions and DES", Advances in Cryptology, Crypto'89,1989, pp.218-238

[22] D.Boneh and R.Venkatesan, "Hardness of Computing the Most Significant Bits of Secret Keys in Diffie-Hellman and Related Problems," Proc. Advances in Crytology-Crypto'96, pp.129-142, 1996.

[23] V.Shoup, "Lower Bounds for Discrete Logarithms and Related Problems," Proc. Advances in Cryptology-Eurocrypt'97, pp.256-266, 1997.

[24] Simbo, A., and Kawamura, S.: "Cryptanalysis of several conference key

distribution systems", Advances in Cryptology-AISACRYPT'91, 1991, pp. 265-276.
[25] F.J. MacWilliams and N.J.A. Sloane, The Theory of Error Correcting Codes, North-Holland, New York, 1977.
[26] F.J.McEliece and D.V.Sarwate, "On sharing Secrets and Reed-Solomon Codes", Communications of ACM, 26(9), 583-584, Sep. 1981.