A Conference Key Scheme Based on the Diffie-Hellman Key Exchange

Li-Chin Huang[1] and Min-Shiang Hwang[2,3]
(Corresponding author: Min-Shiang Hwang)
Department of Information Management, Executive Yuan, Taiwan (ROC)[1]
Department of Computer Science and Information Engineering, Asia University[2]
No. 500, Lioufeng Raod, Wufeng Shiang, Taichung, Taiwan
Department of Medical Research, China Medical University Hospital, China Medical University[3]
(Email: mshwang@nchu.edu.tw)
(Received May 31, 2018; revised and accepted Aug. 10, 2018)

Abstract

Secure group communication is becoming more and more important in internet. In order to provide a secure and reliable communication among the members of a conference over a public network, all group members must have the ability to establish a common secret key. We call this kind of the public key distribution system is a conference key distribution system (CKDS). Our protocol bases on the two-party Diffie-Hellman protocol to build intermediate keys from each subgroups gradually until the entire conference key is obtained. The process of forming the entire conference key will constructed a ripple structure which reduce the times of encryption and decryption than butterfly scheme key distribution systems. Our protocol promote the efficiency of a conference key distribution system.

Keywords: Conference Key; Key Authentication; Secure Group Communication

1 Introduction

A public key distribution system called public key distribution system (PKDS) is developed firstly [4]. However, this system provides only one pair of communication parties to share a particular pair of encryption and decryption keys [3, 9, 11, 13, 14, 21, 30]. The public key distribution system is applied in a conference key distribution system (CKDS) to permit any legitimate parties to share the same encryption and decryption keys. Hence, a conference key distribution system (CKDS) [2,29] is a scheme which generates a conference key and then spreads this key to all legitimate participants for establishing a secure communication.

Imgresson *et al.* [16] proposed a conference key distribution system (CKDS) without authentication on a ring network. For authenticate legitimate members using member's identification information (such as member's name and address) in cryptosystems, Shamir and Fiat proposed identify-based signature schemes [5], and Okamoto proposed an identity-based scheme [23]. An identity-based system applies to generating a conference key with authentication [23], called an identity-based conference key distribution system (ICKDS). Koyam and Ohta [17] applied Identity-based CKDS (ICKDS) on a ring network, complete graph network, and a star network. Many conference key schemes had been proposed for various enterprise organizations [1, 6–8, 15, 20, 25, 27, 31]. In this paper, we will propose a new hierarchical approach, the ripple scheme, to improve a conference key distribution.

The remaining of the paper is organized as follows. Section 2 gives a brief overview of the butterfly scheme. The proposed method is described in Section 3. Sections 4, 5, and 6 discuss the encryption and decryption cost, performance comparisons, and security analysis. Section 7 closes the paper with the conclusion.

2 Review of The Butterfly Scheme

In the butterfly scheme [26], the users generate a share keys for small subgroups, and furthermore these subgroups form larger subgroups and establish new subgroup keys by previous subgroup keys. These steps of key generation are repeated until the whole group constructs a share key for all users. The butterfly scheme is shown in Figure 1.

The initiation phase: Each user u_j chooses a random secret integer $\alpha_j \in Z_p^*$, where Z_p^* denotes the nonzero elements of the integers mod a prime p.

A conference key generation and distribution phase:

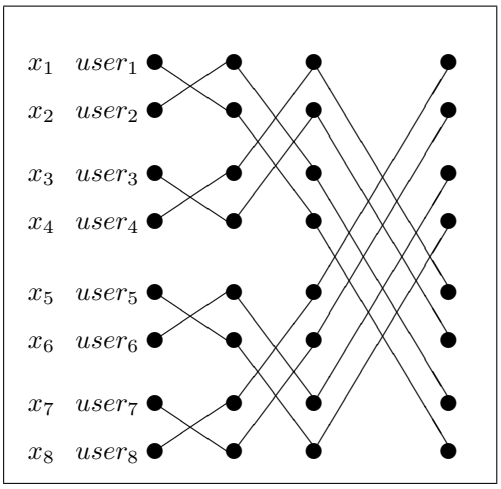

Figure 1: The butterfly scheme

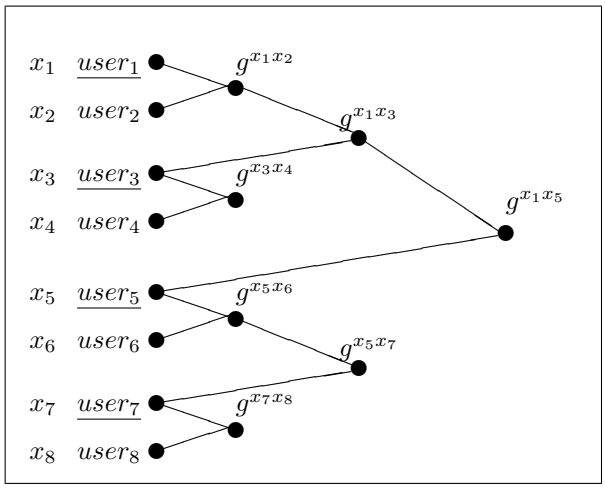

Figure 2: The Ripple scheme

1) In the first pairing, the members of a pair exchange $g^{x_j^0}$, where $x_j^0 = \alpha_j$. For example, u_1 sends $g^{x_1^0}$ to u_2, and u_2 sends $g^{x_2^0}$ to u_1. Then the users u_{2j-1} and u_{2j} compute $x_j^1 = g^{x_{2j-1}^0 x_{2j}^0} = g^{\alpha_{2j-1}\alpha_{2j}}$, where $x_j^1 \in Z_p^*$. Therefore, the members of a pair have established a conventional Diffie-Hellman key exchange.
2) In the second pairing, we may pair the pairs $u_j^1 = \{u_{2j-1}, u_{2j}\}$ into a second level of pairs. For Instance, $u_1^2 = \{u_1^1, u_2^1\}$, and a general rule $u_j^2 = \{u_{2j-1}^1, u_{2j}^1\}$. Consequently, the second level of pairings consists of 4 users in a pair. Each user u_{2j-1}^1 and u_{2j}^1 exchange $g^{x_{2j-1}^1}$ and $g^{x_{2j}^1}$. Every member in u_j^2 can compute $x_j^2 = g^{x_{2j-1}^1 x_2^1 j}$.
3) In the third pairing, consisting of 8 users may be formed. Each user u_{2j-1}^2 and u_{2j}^2 exchange $g^{x_{2j-1}^2}$ and $g^{x_{2j}^2}$. Then, every member in u_j^3 can compute $x_j^3 = g^{x_{2j-1}^2 x_{2j}^2}$. Reasoning from above the principle, we may produce a general rule, $u_j^k = \{u_{2j-1}^{k-1}, u_{2j}^{k-1}\}$ and $x_j^k = g^{x_{2j-1}^{k-1} x_{2j}^{k-1}}$.

3 The Proposed Scheme

We propose a different hierarchical approach to improve a conference key distribution. In our scheme, the users form keys for small subgroups using Diffie-Hellman scheme [4], and these subgroups act as single entities and chose a user as their manager to establish subgroup key that form larger subgroups and establish new keys using the manager's key chosen by in the previous subgroup keys. The process repeats until the entire group has formed a key that was shared by all members. For simplicity, we suppose the ripple scheme for establishing a group key for 8 users as shown in Figure 2. Our scheme describe as follow.

The initiation phase:

Each $user_i$ chooses a random secret integer $x_i \in Z_P^*$, $i \in \{1, 2, 3, 4, 5, 6, 7, 8\}$.

A conference key generation phase:

1) In the first round, the members of a pair exchange g^{x_i}, $i \in \{1, 2, 3, 4, 5, 6, 7, 8\}$ to establish a conventional Diffie-Hellman key as their subgroup key (SK for short). Thus the users form keys for small subgroups, and these subgroups as single entities. For example, $user_1$ sends g^{x_1} to $user_2$, and $user_2$ sends g^{x_2} to $user_1$. Then, $user_1$ and $user_2$ establish a subgroup key $SK_{12} = g^{x_1x_2}$ mod p. Therefore, the $user_{2k-1}$ and $user_{2k}$ calculate a subgroup key $SK_{(2k-1)(2k)} = g^{(x_{2k-1})(x_{2k})}$ mod p , $k \in \{1, 2, 3, 4\}$, respectively.
2) In the second round, to form larger subgroups and establish new subgroup keys. The new subgroup key is formed by the manager's key which is selected from each subgroup. For example, $user_1$ and $user_2$ form a small subgroup which subgroup key SK_{12} is $g^{x_1x_2}$ mod p. And immediately they select a group manager $user_1$. As the same way, $user_3$ and $user_4$ get a small subgroup key $SK_{34} = g^{x_3x_4}$ mod p, and then select a group manager $user_3$. Afterwards, they take the manager's key g^{x_1} and g^{x_3} to form a larger subgroup key $SK_{1234} = g^{x_1x_3}$ mod p for $\{user_1, user_2, user_3, user_4\}$. Obviously, $\{user_5, user_6\}$ and $\{user_7, user_8\}$ can separately get $SK_{56} = g^{x_5x_6}$ mod p and $SK_{78} = g^{x_7x_8}$ mod p. They also select the manager $user_5$ and $user_7$ to from $SK_{5678} = g^{x_5x_7}$ mod p.

   Finally, the conference key $SK_{12345678} = g^{x_1x_5}$ mod p is formed by the manager $user_1$

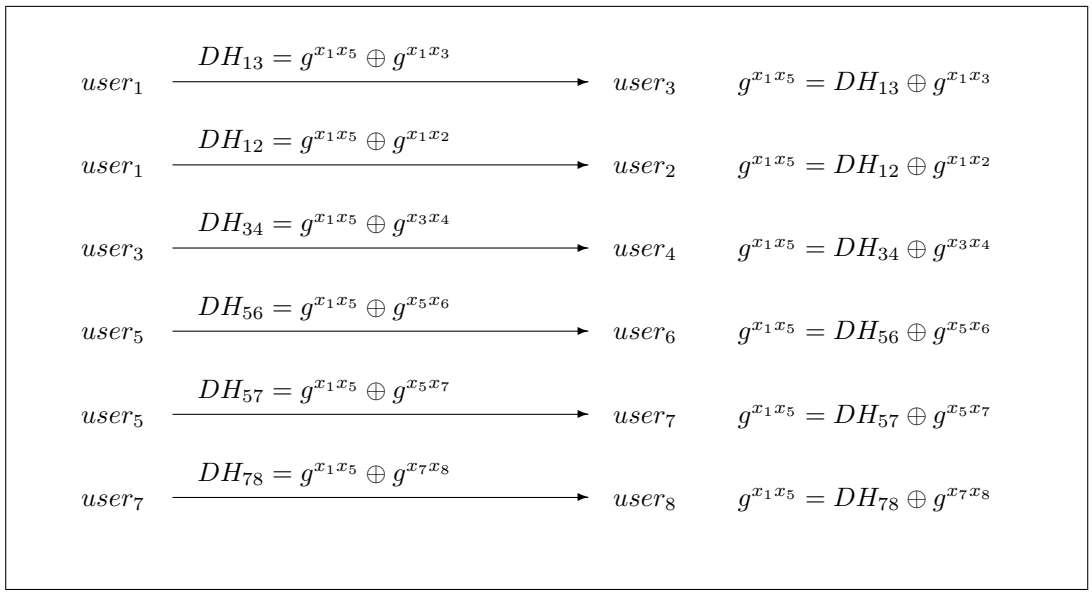

Figure 3: The conference key distributes to each member

and $user_5$ from $\{user_1, user_2, user_3, user_4\}$ and $\{user_5, user_6, user_7, user_8\}$.

The conference key distributes to each member phase:

The group manager $user_1$ and $user_5$ possess the conference key (for short CK).

1) $user_1$ send $DH_{13} = g^{x_1x_5} \bigoplus g^{x_1x_3}$ to $user_3$. As the same time, $user_1$ also sends $DH_{12} = g^{x_1x_5} \bigoplus g^{r_1r_2}$ to $user_2$.
2) $user_2$ gets the conference key $g^{x_1x_5}$ by computing $DH_{12} \bigoplus g^{x_1x_2}$.
3) $user_3$ gets the conference key $g^{x_1x_5}$ by $DH_{13} \bigoplus g^{x_1x_3}$. Then $DH_{34} = g^{x_1x_5} \bigoplus g^{x_3x_4}$ is computed by $user_3$ and sends it to $user_4$.
4) $user_4$ gets the conference key $g^{x_1x_5}$ by $DH_{34} \bigoplus g^{x_3x_4}$.
5) As the same way, $user_5$ send DH_{56} and DH_{57} to $user_6$ and $user_7$, respectively.
6) $user_6$ gets the conference key by $DH_{56} \bigoplus g^{x_5x_6}$.
7) $user_7$ gets the conference key by $DH_{57} \bigoplus g^{x_5x_7}$ and send DH_{76} to $user_8$.
8) $user_8$ gets the conference key by $DH_{78} \bigoplus g^{x_7x_8}$.

4 Encryption and Decryption Costs

As indicated in the previous section, our protocol is superior to the others with respect to exponentiation operations. With respect to conference key generation time, the total cost is 22 exponents and 12 XOR. We describe as follow.

The conference key generation phase:

Table 1 is the cost of the conference key generation phase.

1) **The first round:** Each $user_i$ and $user_{i-1}$ establishes a secret key $SK_{(2i-1)(2i)}$ based on Diffie-Hellman key exchange. There are 16 exponents in this round.
2) **The second round:** There are 2 exponentiation to form SK_{1234} secret key for $\{user_1, user_2, user_3, user_4\}$. As the same way, to construct SK_{5678} secret key requires 2 exponents.
3) **The third round:** To construct $SK_{12345678}$ (i.e. conference key) requires 2 exponents.

The conference key distributes to each member phase:

Figure 3 describes the conference key distributes to each member. Obviously, $user_i$ will generates $DH_{ij} = g^{x_1x_5} \oplus g^{x_ix_j}$ and send to $user_j$, the pair of $(i, j) \in \{(1, 3), (1, 2), (3, 4), (5, 6)\}$. Finally, $user_j$ get conference key $g^{x_1x_5}$.

Therefore, if n members require $n \times 2 + \sum_{i=1}^{log_2 n-1} \frac{1}{2i}$ exponents and $2log_2 n$ XOR to construct a conference key. In tree based conference key distribution systems require $2nlogn$. Our protocol is more efficient obviously.

5 Performance Comparison

In this section, we shall compare the computational complexity of our scheme with that of the butterfly scheme. To analyze the computational complexity, we first define the following notations.

Table 1: The Cost of the Conference Key Generation Phase

Round	Operation	Exponents per $user_i$	Times of exponents
1	$user_i$ compute g^{x_i} $i \in \{1,2,3,4,5,6,7,8\}$	$user_i$ $i \in \{1,2,3,4,5,6,7,8\}$	8
	$SK_{(2i-1)(2i)} = g^{(2x_i)(2i)}$ $i \in \{1,2,3,4,5,6,7,8\}$	$user_i$ $i \in \{1,2,3,4,5,6,7,8\}$	8
2	$SK_{1234} = g^{(x_1)(x_3)}$	$user_1$, $user_2$	2
	$SK_{5678} = g^{(x_5)(x_7)}$	$user_5$, $user_7$	2
3	$SK_{12345678} = g^{(x_1)(x_5)}$	$user_1$, $user_5$	2
Total			22

T_{MUL}: the time for computing modular multiplication.

T_{EXP}: the time for computing modular exponentiation.

T_{XOR}: the time for computing exclusive OR.

n: the number of participants in the conference.

In the conference key generation stage of our scheme, n members generate an entire conference key. Each member chooses a random secret key x_i and computes the corresponding public key g^{x_i}. Then, the members of a pair exchange their secret g^{x_i}, i $\in \{1, ..., n\}$ to construct a conventional Diffie-Hellman key as their subgroup key. Obviously, round 1 requires $2n \times T_{EXP}$. In the second Round, $\frac{n}{2}$ subgroup keys form larger subgroups by a convention Diffie-Hellman key, which require $\frac{n}{2} \times T_{EXP}$. As the same way, round i requires $\frac{n}{2^{i-1}} \times T_{EXP}$. Total computational complexity in this stage is required 3n-2n $\times (\frac{1}{2})^{(\log_2 n)} \times T_{EXP}$.

After generating the entire conference key (CK), each member enters the conference key distribution stage. $User_1$ computes $DH_{1,(\frac{n}{2^2}+1)} = CK \bigoplus g^{x_1 x_{\frac{n}{2^2}+1}}$ and send to $user_{(\frac{n}{2^2}+1)}$. Then $user_{(\frac{n}{2^2}+1)}$ obtains the conference key by $DH_{1,(\frac{n}{2^2}+1)} \bigoplus g^{x_1 x_{\frac{n}{2^2}+1}}$. As the same way, $user_{\frac{n}{2}+1}$ computes $DH_{(\frac{n}{2}+1),(\frac{n}{2}+\frac{n}{2^2}+1)} = CK \bigoplus g^{x_{\frac{n}{2}+\frac{n}{2^2}+1}}$ and send to $user_{(\frac{n}{2}+\frac{n}{2^2}+1)}$. $User_{(\frac{n}{2}+\frac{n}{2^2}+1)}$ also gets the conference key by $DH_{(\frac{n}{2}+1),(\frac{n}{2}+\frac{n}{2^2}+1)} \bigoplus g^{x_{\frac{n}{2}+\frac{n}{2^2}+1}}$. Therefore, round 1 requires $4 \times T_{XOR}$. The round 2 requires $8 \times T_{XOR}$ and round i requires $2^{(i+1)} \times T_{XOR}$. Total computation complexity in the key distribution stage is required $(2^{(\log_2 n)+1} - 4) \times T_{XOR}$.

In the computational complexity of the butterfly scheme, each user u_j chooses a random secret integer α_j and the members of a pair exchange x_j^0 to get the Diffie-Hellman key u_j^1 in the first round. Therefore, round 1 requires $2n \times T_{EXP}$. In the round 2, the members form $u_j^2 = \{u_{2J-1}^1, u_{2J}^1\}$ and require $n \times T_{EXP}$. The round i requires $n \times T_{EXP}$. Therefore, the butterfly scheme requires $n(\log_2 n + 1) \times T_{EXP}$.

According to Table 2, our scheme is more efficient than the butterfly scheme obviously.

6 Security

The security level of the proposed CKDS is based on Discrete Logarithmic Problem. Assume p is a large prime and g is a generator for Z_p^*. If b $\in X_p^*$ is publicly known, it is still hard to find the a such that b = g^a mod p. In our scheme, we extend two party Diffie-Hellman key exchange to construct a conference key, that is, two users such as $user_1$ (with private key x_1 and public key $b_1 = g^{x1}$ mod p) and $user_2$ (with private key x_2 and public key $b_2 = g^{x2}$ mod p) can calculate the shared key SK = g^{x1x2} mod p. Any user except $user_1$ and $user_2$ can not calculate SK even they know x_1 and x_2. Although the Diffie-Hellman key exchange exits a Man-in-Middle attack, many solutions [10, 12, 18, 22, 24, 28] are proposed to solve this problem.

7 Conclusions

In this paper, we show a different group key that the users produce a group key by the two-party Diffie-Hellman protocol. Our new scheme is more efficient than the butterfly scheme [26].

References

[1] T. Y. Chang and M. S. Hwang, "User-anonymous and short-term conference key distribution system via link-layer routing in mobile communications", *International Journal of Mobile Communications*, vol. 9, no. 2, pp. 144–158, 2011.

[2] T. Y. Chang, M. S. Hwang, W. P. Yang, "Cryptanalysis of the Tseng-Jan anonymous conference key distribution system without using a one-way hash function", *Information & Security: An International Journal*, vol. 15, no. 1, pp. 110–114, 2004.

[3] T. Y. Chang, W. P. Yang, M. S. Hwang, "Simple authenticated key agreement and protected password change protocol," *Computers & Mathematics with Applications*, vol. 49, pp. 703–714, 2005.

Table 2: Computational complexities of the butterfly scheme and our scheme

Round	Round 1	Round 2	...	Round i ($2 \le i \le \log_2 n$)	total
Our scheme					
key generation	$2n\times\ T_{EXP}$	$\frac{n}{2}\times\ T_{EXP}$	...	$\frac{n}{2^{i-1}}\times\ T_{EXP}$	3n-2n $\times(\frac{1}{2})^{(\log_2 n)}\times\ T_{EXP}$
key distribution	$4\times\ T_{XOR}$	$8\times T_{XOR}$	...	$2^{(i+1)}\times T_{XOR}$	$(2^{(\log_2 n)+1}-4)\times T_{XOR}$
The butterfly scheme	$2n\times\ T_{EXP}$	$n\times\ T_{EXP}$	...	$n\times\ T_{EXP}$	$[n(\log_2 n-1)+2n]\times T_{EXP}$ $= n(\log_2 n+1)\times\ T_{EXP}$

[4] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. IT-22, pp. 644–654, Nov. 1976.

[5] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *Proceedings of Crypto'86*, pp. 175–184, 1987.

[6] C. Guo, C. C. Chang, "A novel threshold conference-key agreement protocol based on generalized Chinese remainder theorem," *International Journal of Network Security*, vol. 17, no. 2, pp. 165-173, 2015.

[7] L. Harn, G. Gong, "Conference key establishment protocol using a multivariate polynomial and its applications," *Security and Communication Networks*, vol. 8, no. 9, pp. 1794–1800, 2015.

[8] C. L. Hsu, T. W. Lin, H. C. Lu, T. H. Chuang, Y. H. Chen, "Privacy-preserved conference key distribution protocol," in *12th International Conference on Digital Information Management*, pp. 127–132, 2018.

[9] M. S. Hwang, "Dynamic participation in a secure conference scheme for mobile communications", *IEEE Transactions on Vehicular Technology*, vol. 48, no. 5, pp. 1469–1474, Sept. 1999.

[10] M. S. Hwang, S. K. Chong, H. H. Ou, "On the security of an enhanced UMTS authentication and key agreement protocol," *European Transactions on Telecommunications*, vol. 22, no. 3, pp. 99–112, 2011.

[11] M. S. Hwang, C. W. Lin, and C. C. Lee, "An improved Yen-Joye's authenticated multiple-key agreement protocol," *Electronic Letters*, vol. 38, no. 23, pp. 1429–1431, 2002.

[12] M. S. Hwang, J. W. Lo, C. H. Liu, "Enhanced of key agreement protocols resistant to a denial-of-service attack," *Fundamenta Informaticae*, vol. 61, no. 3, pp. 389–398, 2004.

[13] M. S. Hwang, W. G. Tzeng, "A conference key distribution scheme in a totally-ordered hierarchy", *Lecture Notes in Computer Science*, vol. 2662, pp. 757–761, 2003.

[14] M. S. Hwang and W. P. Yang, "Conference key distribution protocols for digital mobile communication systems", *IEEE Journal on Selected Areas in Communications*, vo1. 13, no. 2, 1995, pp. 416–420.

[15] T. Hyla, J. Pejaś, "A fault-tolerant authenticated key-conference agreement protocol with forward secrecy," *Lecture Notes in Computer Science*, vol. 9842, pp. 647–660, 2016.

[16] I. Ingemarsson, D. T. Tang, and C. K. Wong, "A conference key distribution system," *IEEE Transactions on Information Theory*, vol. IT-28, September.

[17] K. Koyama and K. Ohta, "Identity-based conference key distribution system," in *Lecture Notes in Computer Science*, vol. 293, pp. 181–187, 1986.

[18] C. C. Lee, M. S. Hwang, and L. H. Li, "A new key authentication scheme based on discrete logarithms," *Applied Mathematics and Computation*, vol. 139, no. 2, pp. 343–349, 2003.

[19] C. C. Lee, T. C. Lin, M. S. Hwang, "A key agreement scheme for satellite communications," *Information Technology and Control*, vol. 39, no. 1, pp. 43–47, 2010.

[20] J. S. Lee, C. C. Chang, and K. J. Wei, "Provably secure conference key distribution mechanism preserving the forward and backward secrecy," *International Journal of Network Security*, vol. 15, no. 5, pp. 405–410, 2013.

[21] J. W. Lo, J. Z. Lee, M. S. Hwang, Y. P. Chu, "An improvement of a simple authenticated key agreement algorithm," *Journal of Internet Technology*, vol. 11, no. 7, pp. 997–1004, 2010.

[22] J. W. Lo, S. C. Lin, M. S. Hwang, "A parallel password-authenticated key exchange protocol for wireless environments," *Information Technology and Control*, vol. 39, no. 2, pp. 146–151, 2010.

[23] E. Okamoto, "Proposal for identity-based key distribution systems," *Electron. Letters*, vol. 22, pp. 1283–1284, 1986.

[24] H. H. Ou, M. S. Hwang and J. K. Jan, "A cocktail protocol with the authentication and key agreement on the UMTS," *Journal of Systems and Software*, vol. 83, no. 2, pp. 316–325, 2010.

[25] J. Ribeiro, G. Murta, S. Wehner, "Fully device-independent conference key agreement," *Source: Physical Review A*, vol. 97, no. 2, 2018.

[26] W. Trappe, Y. Wang, and K. J. R. Liu, "Group key agreement using divide-and-conquer strategies," in *Conference on Information Sciences and Systems*, 2001.

[27] C. Tselikis, C. Douligeris, S. Mitropoulos, N. Komninos, G. Tselikis, "Adaptation of a Conference Key Distribution System for the wireless ad hoc network,"

in *IEEE International Conference on Communications*, 2017.

[28] C. C. Yang, T. Y. Chang, M. S. Hwang, "Cryptanalysis of simple authenticated key agreement protocols," *IEICE Transactions on Foundations*, vol. E87-A, no. 8, pp. 2174–2176, 2004.

[29] C. C. Yang, T. Y. Chang, M. S. Hwang, "A new anonymous conference key distribution system based on the elliptic curve discrete logarithm problem", *Computer Standards and Interfaces*, vol. 25, no. 2, pp. 141–145, May 2003.

[30] C. C. Yang, J. W. Li, M. S. Hwang, "A new mutual authentication and key exchange protocol with balanced computational power for wireless settings," *European Transactions on Telecommunications*, vol. 15, no. 2, pp. 91–99, 2004.

[31] C. N. Yang, J. M. Li, Y. S. Chou, "On the analysis of k-secure t-conference key distribution scheme," in *ACM 7th International Conference on Communication and Network Security*, pp. 91–95, 2017.

Biography

Li-Chin Huang received the B.S. in computer science from Providence University, Taiwan, in 1993 and M.S. in information management from Chaoyang University of Technology (CYUT), Taichung, Taiwan, in 2001; and the Ph.D. degree in computer and information science from National Chung Hsing University (NCHU), Taiwan in 2001. Her current research interests include information security, cryptography, medical image, big data, and mobile communications.

Min-Shiang Hwang received B.S. in electronic engineering from National Taipei Institute of Technology, Taipei, Taiwan, in 1980; M.S. in industrial engineering from National Tsing Hua University, Taiwan in 1988; and Ph.D. degree in computer and information science from National Chiao Tung University, Taiwan in 1995. He also studied applied mathematics at National Cheng Kung University, Taiwan, from 1984-1986. Dr. Hwang passed the National Higher Examination in field "electronic engineer" in 1988. He also passed National Telecommunication Special Examination in field "information engineering", qualified as the first class advanced technician in 1990. From 1988 to 1991, he was the leader of the Computer Center at Telecommunication Laboratories, Ministry of Transportation and Communications. He was also the Chairman of Department of Information Management, CYUT, Taiwan, during 1999-2002. He was a professor and Chairman of the Department of Management Information Systems, NCHU, during 2003-2009. He was also a visiting professor with University of California (UC), Riverside and UC. Davis (USA) during 2009-2010. He was a distinguished professor of Department of Management Information Systems, NCHU, during 2007-2011. He obtained the 1997, 1998, 1999, 2000, and 2001 Excellent Research Award of National Science Council (Taiwan). Dr. Hwang was a dean of College of Computer Science, Asia University (AU), Taichung, Taiwan. He is currently a chair professor with Department of Computer Science and Information Engineering, AU. He is a member of IEEE, ACM, and Chinese Information Security Association. His current research interests include information security, electronic commerce, database and data security, cryptography, image compression, and mobile computing. Dr. Hwang has published over 300+ articles on the above research fields in international journals.