# Notes on "Polynomial-based Key Management for Secure Intra-Group and Inter-Group Communication"

Chin-Chen Chang[1, 2], Lein Harn[3], and Ting-Fang Cheng[2]
(Corresponding author: Chin-Chen Chang)

Department of Information Engineering and Computer Science, Feng Chia University[1]
No. 100, Wen-Hwa Rd., Taichung, Taiwan, 40724, R.O.C
Department of Computer Science and Information Engineering, Asia University[2]
No. 500, Lioufeng Rd., Wufeng, Taichung, Taiwan, 41354, R.O.C.
Department of Computer Science Electrical Engineering, University of Missouri- Kansas City[3]
550K Flarsheim Hall, 5100 Rockhill Rd., Kansas City, MO 64110, USA
(Email: alan3c@gmail.com)

(Received Aug. 7, 2013; revised and accepted Sep. 11, 2013)

## Abstract

In 2012, Piao et al. proposed a polynomial-based key management for secure intra-group and inter-group communication. In this notes, we point out that there are some security weaknesses of Piao et al.'s intra-group key distribution scheme. One main problem is that their scheme cannot prevent a group member to obtain other members' secret keys shared with the controller. In addition, their scheme is suffered from the replay attack and cannot achieve the objectives of both perfect forward and backward secrecy. We provide a simple modified scheme to overcome these security weaknesses.

*Keywords: Perfect forward and backward secrecy, polynomial-based key distribution, replay attack, secure intra-group communication*

## 1 Introduction

Group communication has been widely developed in various communication applications and environments such as conferences [1, 6], wireless sensor networks [2, 7, 10, 14], and ad hoc networks [5]. In a secure group communication, a dynamic group key needs to be shared among all group members. All group members will use the group key to protect their information. The polynomial-based key distribution [1, 2, 7, 8, 11, 12, 13] is one of the group key distributions. Wang et al. [11, 12, 13] proposed a polynomial-based inter-group key sharing to use a *controller* to distribute personal key shares for inter-group communication. As shown in Figure 1, an *l*-degree polynomial $P_{i,j}(x)$ constructed by the controller in $G_i$ is used to distribute a secret group key such that a node (i.e., the variable $x$) in $G_j$ and members of $G_i$ can communicate with each other using the key. In 2012, Piao et al. [8] proposed a more efficient generation method of $P_{i,j}(x)$ based on [11, 12, 13] such that each member can construct $P_{i,j}(x)$ by her/himself. They also developed another polynomial $F(x)$ for intra-group key distribution such that all group members can efficiently retrieve the intra-group key from the broadcast message sent by the controller. Unfortunately, in this notes, we point out that there are some security weaknesses of Piao et al.'s intra-group key distribution scheme [8]. One main problem is that their scheme cannot prevent a group member to obtain other members' secret keys shared with the controller. Furthermore, their scheme is suffered from the replay attack and cannot achieve the objectives of both perfect forward and backward secrecy.

The remainder of this paper is organized as follows. In Section 2, we briefly review Piao et al.'s intra-group key sharing and re-keying, followed by the cryptanalysis on their scheme in Section 3. Subsequently, we provide a simple modified scheme and its detailed security analysis in Sections 4 and 5, respectively. Finally, we make conclusions in Section 6.

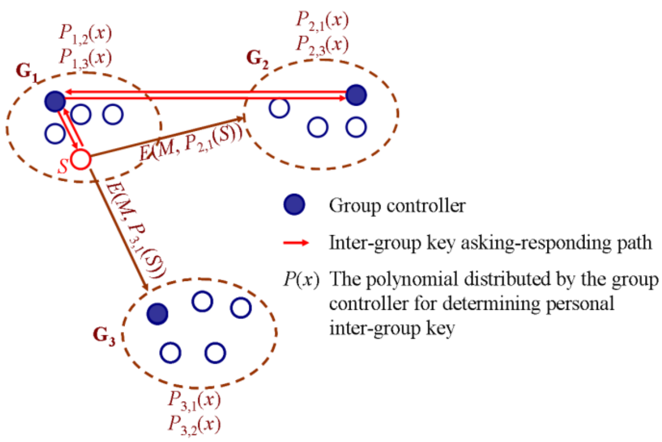


Figure 1: An example of Wang et al.'s inter-group key sharing: a member $S$ in $G_1$ sends message to all of members in both $G_2$ and $G_3$.

## 2 Piao et al.'s Scheme

Here, we briefly review the key management scheme of intra-group key sharing and re-keying proposed in [8]. The notations used in this scheme are defined as follows.

$G_k$: the $k$-th group

$GK_k$: the intra-group key for members of $G_k$

$SK_t$: the pre-distributed secret key shared between the group controller and a member $t$ in the same group

$F(x)$: the polynomial function in a finite field $GF(p)$ used for deriving intra-group key $GK_k$, where $p$ is a large prime

### 2.1 Intra-Group Key Sharing

In a group $G_k$ with $n$ members, assume that each member has pre-shared a secret key $SK_t$ with the group controller through a secure channel. If the group controller wants to distribute the intra-group key $GK_k$ to all members for a secure group communication, the controller has to perform the following procedures.

Step 1. The group controller selects an intra-group key $GK_k$ for all members in the group.

Step 2. The group controller uses all secret keys shared with the group members to generate a polynomial $F(x)$ and conceal the intra-group key $GK_k$ in it i.e., $F(x) = (x\text{-}SK_1)(x\text{-}SK_2)\ldots(x\text{-}SK_n)+GK_k$. The controller then broadcasts $F(x)$ to all members.

Step 3. Upon receiving the polynomial $F(x)$, every member can use her/his own secret key $SK_t$ to retrieve the intra-group key $GK_k$ by computing $F(SK_t)$.

### 2.2 Re-Keying

If there is any change in the group membership, the group controller has to renew the intra-group key for the sake of forward/backward secrecy. Based on the example given in Subsection 2.1, the re-keying for member joining and leaving are described in the following processes, respectively.

#### 2.2.1 For Member Joining

Assume that a new member $W$ wants to join in the group $G_k$. The group controller has to perform the following procedures.

Step 1. The group controller gives a share key $SK_W$ to $W$ in a secure channel.

Step 2. The group controller generates a new intra-group key $GK'_k$ and constructs a new polynomial as $F'(x) = (x\text{-}SK_1)(x\text{-}SK_2)\ldots(x\text{-}SK_n)(x\text{-}SK_W)+GK'_k$. The controller then broadcasts $F'(x)$ to all members.

Step 3. Upon receiving the polynomial, every member can use her/his own secret key $SK_t$ to retrieve the intra-group key $GK'_k$ by computing $F'(SK_t)$.

#### 2.2.2 For Member Leaving

If a member $Z$ leaves the group $G_k$, the group controller has to perform the following procedures.

Step 1. The group controller generates a new intra-group key $GK'_k$ and constructs a new polynomial as $F'(x)= (x\text{-}SK_1)(x\text{-}SK_2)\ldots(x\text{-}SK_{Z-1})(x\text{-}SK_{Z+1})\ldots(x\text{-}SK_n)+GK'_k$. The controller broadcasts $F'(x)$ to members remained in the group.

Step 2. Upon receiving the polynomial, every member can use her/his own secret key $SK_t$ to retrieve the intra-group key $GK'_k$ by computing $F'(SK_t)$.

## 3 Security Problems of Piao et al.'s Scheme

In this section, we analyze Piao et al.'s scheme [8] and point out some security problems.

### 3.1 One-Time Use of Pre-Shared Secrets

In [8], the polynomial $F(x)$ in a finite field $GF(p)$, where $p$ is a large prime, is used to distribute the intra-group key. As pointed out in [4], if the modulus used for the polynomial operation is a prime, it may suffer from the so-called ***internal attack***. The internal attack is launched by a legitimate group member who knows the group key. But, the attacker tries to obtain the secrets of other group members shared with the controller. For example, if a dishonest member, Ivy, in a group has received a polynomial $F(x) = (x\text{-}SK_1)(x\text{-}SK_2)\ldots(x\text{-}SK_I)\ldots(x\text{-}SK_n)+GK_k$ from the group controller and retrieved the intra-group key $GK_k$, she can further deduce another polynomial in $GF(p)$ as $H(x) = (F(x)\text{-}GK_k)/(x\text{-}SK_I) = (x\text{-}SK_1)(x\text{-}SK_2)\ldots(x\text{-}SK_{I-1})(x\text{-}SK_{I+1})\ldots(x\text{-}SK_n)$. It is computationally feasible for solving the roots of the polynomial $H(x)$, which are other members' secret keys $SK_k$'s. Thus, the pre-shared secrets of members in Piao et al.'s scheme can only be used for a one-time group communication.

### 3.2 Replay Attack

If an attacker, Eve, intercepts the polynomial $F(x)$ sent by the group controller, she can easily mount the replay attack by replaying it. This is because group members do not verify the freshness of the group key. Assume that Eve has recorded the polynomial $F_{\_old}(x) = (x\text{-}SK_1)(x\text{-}SK_2)\ldots(x\text{-}SK_n)+GK_{k\_old}$. Following, we consider different scenarios associated with this attack.

***Case 1.*** *Assume that the group membership does not change after the group key $GK_{k\_old}$ being compromised by Eve.*

Assume that the group controller sends a new key distribution message $F_{\_new}(x) = (x\text{-}SK_1)(x\text{-}SK_2)\ldots(x\text{-}SK_n)+GK_{k\_new}$ to members of the same group, where $GK_{k\_new}$ is the new intra-group key. The attacker, Eve, can replay the polynomial $F_{\_old}(x)$ corresponding to the group key $GK_{k\_old}$

which has been compromised by Eve already. Obviously, all members in the group can retrieve the intra-group key $GK_{k\_old}$ and use the key $GK_{k\_old}$ to communicate with each other. However, Eve knows the content of all future communications.

***Case 2.*** *Assume that a new member has joined in the group after $GK_{k\_old}$ being used for the group communication; but the group key $GK_{k\_old}$ has not been compromised by Eve.*

Assume that the group controller sends a new key distribution message $F_{\_new}(x) = (x\text{-}SK_1)(x\text{-}SK_2)\ldots(x\text{-}SK_n)(x\text{-}SK_W)+GK_{k\_new}$ when a new member, William, has joined in the group, where $SK_W$ is William's secret key shared with the controller and $GK_{k\_new}$ is the new intra-group key. The attacker, Eve, can replay the polynomial $F_{\_old}(x)$ to all members in the group. After receiving the replayed message, only William cannot retrieve the corresponding intra-group key $GK_{k\_old}$ from $F_{\_old}(x)$. This is because that the replayed polynomial $F_{\_old}(x) = (x\text{-}SK_1)(x\text{-}SK_2)\ldots(x\text{-}SK_n)+GK_{k\_old}$ is generated before William joining in the group. Obviously, when William computes $F_{\_old}(SK_W)$ using his own secret key $SK_W$, he cannot get the same group key as other members' obtained. This will end up an unsuccessful group communication.

### 3.3 No Forward/Backward Secrecy

Assume that a dishonest user, David, who used to be a group member knowing the group key $GK_{k\_old}$, attempts to obtain the content of communications which he is not authorized to.

#### 3.3.1 No Forward Secrecy

Assume that David has stored the polynomial $F_{\_old}(x) = (x\text{-}SK_1)(x\text{-}SK_2)\ldots(x\text{-}SK_n)+GK_{k\_old}$ and known the key, $GK_{k\_old}$, since he was a member of this group. Afterward, he just replays the polynomial disguised as a new key distribution message to all members in the group. He can easily learn the traffic of the group communications which he is not authorized to. This is because they will use the same key $GK_{k\_old}$ to communicate with each other. It is noteworthy that such attack is based on the assumption that the group membership does not change after David leaving the group. As a result, Piao et al.'s intra-group key distribution scheme [8] does not possess perfect forward secrecy.

#### 3.3.2 No Backward Secrecy

Assume that David intends to deduce the previous intra-group key which he is not authorized to. He can launch the attack through following procedures.

Step 1. David intercepts the polynomial $F(x)$ sent by the group controller and then joins in the group. Assume that the polynomial he intercepted is $F(x) = (x\text{-}SK_1)(x\text{-}SK_2)\ldots(x\text{-}SK_n)+GK_k$.

Step 2. After David joining in the group, the group controller gives a share key $SK_D$ to him through a secure channel. In addition, the controller generates a new intra-group key $GK_{k\_new}$ and constructs a new polynomial as $F_{\_new}(x) = (x\text{-}SK_1)(x\text{-}SK_2)\ldots(x\text{-}SK_n)(x\text{-}SK_D)+GK_{k\_new}$. The controller sends $F_{\_new}(x)$ to all members including David.

Step 3. Upon receiving the polynomial, David can use his own secret key $SK_D$ to retrieve the intra-group key $GK_{k\_new}$ by computing $F_{\_new}(SK_D)$. Then, he computes a new polynomial $H(x) = (F_{\_new}(x) - GK_{k\_new})/(x\text{-}SK_D)$.

Step 4. David can deduce the previous intra-group key GKk by computing F(x) - H(x). Obviously, the key he deduced is valid since H(x) = (F_new(x) - GKk_new)/(x-SKD) = (x-SK1)(x-SK2)…(x-SKn).

Thus, Piao et al.'s intra-group key distribution scheme [8] does not possess perfect backward secrecy.

## 4 A Simple Modification

In this section, we provide a simple modified scheme to overcome security weaknesses as mentioned in last section. In order to prevent the internal attack as we have described previously, we use a composite number as the modulus i.e., $N = pq$, where $p$ and $q$ are large primes used in RSA scheme [9]. In addition, we add random challenges of group members to overcome the replay attack.

Similarly, we assume that in a group $G_k$ with $n$ members, all members have pre-shared their secret keys $SK_t$'s with the group controller. The group controller performs following procedures to distribute the intra-group key $GK_k$ to all members.

Step 1. The group controller broadcasts a group communication message to all members.

Step 2. After receiving the message, each member randomly chooses a challenge Ct from ZN* and sends it back to the controller.

Step 3. The group controller generates an intra-group key GKk for all members in the group communication. Then, the controller uses all challenges and secret keys shared with the group members to generate a polynomial F(x) with modulus N (i.e., N = pq, where p and q are large primes) as F(x) = (x-(SK1⊕C1))(x-(SK2⊕C2))…(x-(SKn⊕Cn))+GKk.

Finally, the controller computes an authentication message as Auth = h(GKk) and sends it along with F(x) to the members, where h(.) is a secure one-way hash function.

Step 4. Upon receiving the key distribution message, every member can use her/his own secret key $SK_t$ and challenge $C_t$ to retrieve the intra-group key $GK_k$ by computing $F(SK_t \oplus C_t)$. After retrieving the intra-group key, every member also can authenticate the group key by checking $h(GK_k)$ ?= $Auth$.

## 5 Discussions

Here, we analyze our modified scheme with respect to each security problem as we have mentioned in Section 3.

### 5.1 Security of Pre-Shared Secrets

In our proposed scheme, we replace the prime modulus $p$ by the composite number $N$, where $N = pq$ as used in RSA [9]. Assume that a dishonest member, Ivy, in a group has received the polynomial $F(x) = (x\text{-}(SK_1 \oplus C_1))(x\text{-}(SK_2 \oplus C_2))\ldots(x\text{-}(SK_I \oplus C_I))\ldots(x\text{-}(SK_n \oplus C_n))+GK_k$ from the group controller and retrieved the intra-group key $GK_k$. Ivy tries to obtain the secret keys $SK_t$'s of other group members shared with the controller by deducing another polynomial in $Z_N$* as

$$H(x) = (F(x) - GK_k)/(x\text{-}(SK_I \oplus C_I)) = (x\text{-}(SK_1 \oplus C_1))(x\text{-}(SK_2 \oplus C_2))\ldots(x\text{-}(SK_{I\text{-}1} \oplus C_{I\text{-}1}))(x\text{-}(SK_{I+1} \oplus C_{I+1}))\ldots(x\text{-}(SK_n \oplus C_n)).$$

She needs to solve the roots of the polynomial $H(x)\equiv 0 \pmod N$ in order to find the secrets. In other words, Ivy needs to solve two separate equations as $H(x)\equiv 0 \pmod p$ and $H(x)\equiv 0 \pmod q$. Nevertheless, this approach is impossible since it is computationally infeasible to factor $N$ (i.e, the factorization assumption in RSA [9]). It is noteworthy that Coppersmith has shown that finding small roots of an univariate polynomial equation modulo an integer $N$ of unknown factorization is easy [3]. However, the secret in our application is at least 100 bits so it is not a "small solution". Thus, the algorithm described in [3] cannot be used to solve the pre-shared secrets of members in our proposed scheme.

### 5.2 Replay Attack

Assume that there is an adversary, Eve, who has intercepted messages transmitted publicly. If Eve intercepts the polynomial $F(x)$ sent by the group controller and attempts to mount the replay attack by replaying it, no matter whether the group membership has changed or not, this attack cannot work properly. This is because the polynomial $F(x)$ involves each member's random challenge $C_t$ which is refreshed for each key distribution. The group members can verify the freshness of the group key. Following, we give an example to analyze the replay attack in detail.

Assume that the group controller sends a new key distribution message $F_{\_new}(x) = (x\text{-}(SK_1 \oplus C_{1\_new}))(x\text{-}(SK_2 \oplus C_{2\_new}))\ldots(x\text{-}(SK_n \oplus C_{n\_new}))+GK_{k\_new}$ with the authentication message $Auth\_new = h(GK_{k\_new})$ to the members of the same group. Note that $GK_{k\_new}$ is the new intra-group key and $C_{t_new}$s are new challenges of all members used in current session. If Eve intends to launch the replay attack, she must replay the polynomial

$$F_{\_old}(x) = (x\text{-}(SK_1 \oplus C_{1\_old}))(x\text{-}(SK_2 \oplus C_{2\_old}))\ldots (x\text{-}(SK_n \oplus C_{n\_old})) + GK_{k\_old}$$

and the corresponding authentication message $Auth\_old = h(GK_{k\_old})$ which has been stored by her already. After receiving the replayed key distribution message, every member uses her/his own secret key $SK_t$ and current challenge $C_{t\_new}$ to compute the intra-group key $GK'_k = F_{\_old}(SK_t \oplus C_{t\_new})$. Obviously, the computed $GK'_k$ is different from $GK_{k\_old}$ which was concealed in $F_{\_old}(x)$. Hence, every member can verify that this key is incorrect by checking $h(GK'_k)$ ?= $Auth\_old$ and then asks the controller to resend another key distribution message. As a result, the random challenges and the authentication message used in our proposed scheme can overcome the replay attack.

### 5.3 Forward/Backward Secrecy

Assume that an adversary (i.e., a dishonest member or an external attacker), Eve, who has compromised the group key $GK_{k\_old}$ in the polynomial $F_{\_old}(x)$, attempts to obtain the content of communications that she is not authorized to. We consider different scenarios associated with forward/backward secrecy.

***Case 1.*** *Assume that Eve intends to destroy forward secrecy by replaying old key distribution messages $F_{\_old}(x)$ and Auth_old.*

Assume that the group controller sends a new key distribution message $F_{\_new}(x) = (x\text{-}(SK_1 \oplus C_{1\_new}))(x\text{-}(SK_2 \oplus C_{2\_new}))\ldots(x\text{-}(SK_n \oplus C_{n\_new}))+GK_{k\_new}$ and the authentication message $Auth\_new = h(GK_{k\_new})$ to members of the same group, where $GK_{k\_new}$ is the new intra-group key and $C_{t\_new}$'s are new challenges of all members used in current session. In order to make group members to use old key $GK_{k\_old}$ to communicate with each other, Eve replays the polynomial $F_{\_old}(x)$ and $Auth\_old$ disguised as a new key distribution message to all members in the group. By following our proposed scheme, this attack cannot work properly. As mentioned in Subsection 5.2, the key retrieved by every member is different from $GK_{k\_old}$. This is because each member's random challenge $C_t$ associated with the polynomial $F(x)$ is refreshed for each key distribution. Hence, all members would not use the retrieved key for group communication. The forward secrecy is achieved.

***Case 2.*** *Assume that Eve intends to destroy forward/backward secrecy by deducing previous/following intra-group key from $GK_{k\_old}$ in the polynomial $F_{\_old}(x)$.*

In our proposed scheme, the intra-group keys $GK_k$'s used in different sessions are all independent. Obviously, it is impossible to reveal other keys from a compromised key $GK_{k\_old}$. Furthermore, if Eve intends to deduce the previous/following intra-group key from the polynomial $F_{\_old}(x)$, she may deduce another polynomial in $Z_N$* (i.e., $N = pq$ as used in RSA [9]) as $H(x) = F_{\_old}(x) - GK_{k\_old} = (x\text{-}(SK_1 \oplus C_{1\_old}))(x\text{-}(SK_2 \oplus C_{2\_old}))\ldots(x\text{-}(SK_n \oplus C_{n\_old}))$. Then, she tries to obtain the secret key $SK_t$ of each group member shared with the group controller from $H(x)$. As explained in Subsection 5.1, it is computationally infeasible for solving the roots of the polynomial $H(x)$. Hence, Eve cannot obtain the secret $SK_t$ of each group member. Obviously, Eve cannot deduce the corresponding intra-group key from the previous/following polynomial without knowing group members' secret keys $SK_t$'s. As a result, our proposed scheme can achieve perfect forward/backward secrecy.

## 6 Conclusions

In this paper, we have described the security problems of Piao et al.'s intra-group key distribution scheme [8]. In their proposed scheme, a prime is used as the modulus for the polynomial operation. This setting cannot prevent a group member to obtain other members' secret keys shared with the controller. In addition, Piao et al.'s scheme is suffered from the replay attack and cannot achieve the objectives of both perfect forward and backward secrecy. We also provided a simple modified scheme to overcome these security problems. Detailed security analysis is also included.

## References


[1] C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly secure key distribution for dynamic conferences," *Information and Computation*, vol. 146, no. 1, pp. 1-23, 1998.

[2] Y. Cheng and D. P. Agrawal, "An improved key distribution mechanism for large-scale hierarchical wireless sensor networks," *Ad Hoc Networks*, vol. 5, no. 1, pp. 35-48, 2007.

[3] D. Coppersmith, "Small solutions to polynomial equations, and low exponent RSA vulnerabilities," *Journal of Cryptology*, vol. 10, no. 4, pp. 233-260, 1997.

[4] L. Harn and C. Lin, "Authenticated group key transfer protocol based on secret sharing," *IEEE Transactions on Computers*, vol. 59, no. 6, pp. 842-846, 2010.

[5] D. Huang and D. Medhi, "A secure group key management scheme for hierarchical mobile ad hoc networks," *Ad Hoc Networks*, vol. 6, no. 4, pp. 560-577, 2008.

[6] 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.

[7] D. Liu, P. Ning, and R. Li, "Establishing pairwise keys in distributed sensor networks," *ACM Transactions on Information and System Security*, vol. 8, no. 1, pp. 41-77, 2005.

[8] Y. Piao, J. Kim, U. Tariq, and M. Hong, "Polynomial-based key management for secure intra-group and inter-group communication," *Computers and Mathematics with Applications*, vol. 65, no. 9, pp. 1300-1309, 2013.

[9] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120-126, 1978.

[10] P. Sarkar and A. Saha, "Security enhanced communication in wireless sensor networks using Reed-Muller codes and partially balanced incomplete block designs," *Journal of Convergence*, vol. 2, no. 1, pp. 23-30, 2011.

[11] W. Wang and B. Bhargava, "Key distribution and update for secure inter-group multicast communication," in *Proceedings of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp. 43-52, Alexandria, VA, Nov. 2005.

[12] W. Wang and T. Stransky, "Stateless key distribution for secure intra and inter-group multicast in mobile wireless network," *Computer Networks*, vol. 51, no. 15, pp. 4303-4321, 2007.

[13] W. Wang and Y. Wang, "Secure group-based information sharing in mobile ad hoc networks," in *Proceedings of IEEE International Conference on Communications*, pp. 1695-1699, Beijing, China, May 2008.

[14] B. Zhou, S. Li, J. Wang, S. Yang, and J. Dai, "A pairwise key establishment scheme for multiple deployment sensor networks," *International Journal of Network Security*, vol. 16, no. 3, pp. 229-236, 2014.


**Chin-Chen Chang** received his Ph.D. degree in computer engineering from National Chiao Tung University. His first degree is Bachelor of Science in Applied Mathematics and master degree is Master of Science in computer and decision sciences. Both were awarded in National Tsing Hua University. Dr. Chang served in National Chung Cheng University from 1989 to 2005. His title is Chair Professor in Department of Information Engineering and Computer Science, Feng Chia University, from Feb. 2005. He is a Fellow of IEEE and a Fellow of IEE, UK. His research interests include database design, computer cryptography, image compression and data structures.

**Lein Harn** received his BS degree in Electrical Engineering from the National Taiwan University in 1977. In 1980, he received his MS in Electrical Engineering from the State University of New York-Stony Brook and in 1984 he received his Ph.D degree in Electrical Engineering from the University of Minnesota. He joined as an Assistant Professor in the department of Electrical and Computer Engineering at the University of Missouri-Columbia in 1984 and in 1986, he moved to Computer Science and Telecommunication Program (CSTP) of University of Missouri-Kansas City (UMKC). His research interests are cryptography, network security and wireless communication security.

**Ting-Fang Cheng** received her Ph.D. degree in computer science from National Tsing Hua University, Hsinchu, Taiwan in 2013. She received the BS and MS degrees in information engineering and computer science from Feng Chia University, Taichung, Taiwan in 2005 and 2007, respectively. Now she is as a post doctor in computer science and information engineering, Asia University, Taichung, Taiwan. Her current research interests include electronic commerce, information security, cryptography, mobile communications, and cloud computing.