Sensors and Materials, Vol. 30, No. 3 (2018) 413–422
MYU Tokyo

S & M 1503

Efficient and Scalable Access Management Scheme Based on Chinese Remainder Theorem

Tsung-Chih Hsiao,[1] Yin-Tzu Huang,[2] Yu-Min Huang,[3]
Tzer-Long Chen,[4] Tzer-Shyong Chen,[5*] and Sheng-De Wang[2]

[1]College of Computer Science and Technology, Huaqiao University,
No. 668 Jimei Avenue, Xiamen 361021, Fujian, China
[2]Department of Electrical Engineering, Taiwan University,
No. 1, Sec. 4, Roosevelt Rd., Taipei 10617, Taiwan
[3]Department of Statistics, Tunghai University,
No. 1727, Sec. 4, Taiwan Boulevard, Xitun District, Taichung 40704, Taiwan
[4]Department of Information Technology, Ling Tung University,
1, Ling tung Rd., Taichung 40852, Taiwan
[5]Department of Information Management, Tunghai University,
No. 1727, Sec. 4, Taiwan Boulevard, Xitun District, Taichung 40704, Taiwan

(Received July 5, 2017; accepted October 25, 2017)

Keywords: mobile agent, access control, key management, private key, Chinese remainder theorem

We introduce access management methods and their disadvantages after reviewing previous papers. On the basis of the Chinese remainder theorem (CRT), we have developed a mobile agent novel key management scheme that solves some of the defects. We demonstrate a mobile agent that can secure and collect information across different places. We also illustrate the mathematical computation of our proposed method. Moreover, our proposed method is flexible since the whole system is considered.

1. Introduction

With the rapid development of computer technology and the Internet, more and more resources are shared through the Internet. Problems regarding access control naturally arise as resources are shared over the Internet. It is worth noting that access control mechanisms are widely used in online video systems, wireless networks, electronic documents, and so on. Therefore, it is necessary to construct an access control mechanism to access data effectively and securely. Common access control problems are unauthorized access, data invasion or destruction, inconsistent permissions, privacy leakage, etc. These imply that the access control problem is worthy of further study.

The purpose of information security access control is to restrict data access to legitimate members. In the information transmission environment using sensor networks, this is essential because the sensor network is a public transport environment, meaning there are more security threats, so more security methods are needed to protect the data access of legitimate members.[1,2]

*Corresponding author: e-mail: arden@thu.edu.tw
http://dx.doi.org/10.18494/SAM.2018.1751

ISSN 0914-4935

We implement the Chinese remainder theorem (CRT)[3,4] in the first scheme to construct an access control scheme. We can comprehend CRT in a simple way, understand, and manipulate this research in various applications, such as online video systems.

This research examines potential internal threats and possible external attacks. According to previous issues on key management, schemes to meet the requirements of the design of the access control mechanism should be developed. We propose that access control schemes based on three mathematical theories can be used to provide secure communication under different application environments.

The idea of CRT is simple and can be used in building an access control scheme that provides high security. The security of cryptographic techniques mostly relies on the complexity of the encryption algorithm. Most solutions are due to public key cryptography[5] that offers to encrypt and decrypt operations. However, CRT is commonly implemented in computer science.[6]

2. Previous Work

The encryption algorithm has been widely used for secure data exchange. In order to prevent unauthorized attacks, it is necessary to provide measures for ensuring data security. CRT has been used for key management; one of its primary applications is data sequence encryption and decryption, and it can be used to generate the key pool and chain for predistribution. CRT has also been used for key generation in a variety of security protocols.

In 2004, Lin *et al.*[7] proposed a hierarchical key management scheme to filter out the demands of repeatedly storing cryptographic keys in a mobile agent. Compared with Volker and Mehrdad's scheme, Lin *et al.*'s scheme offers fewer public key computations. However, modular exponentiation is required during key generation and derivation.[8] In addition to being very expensive, it might also cause some problems with the bottleneck while serving several agents and visiting hosts at the same time.

According to Chen *et al.*[9] in 2009, only lightweight operations with one-way hash function and bitwise XOR operators were embedded. Since the number of public variables $\gamma_{\rightarrow j}$ was more strongly related to the parental nodes and descendent nodes, a larger storage space was required. We set ω to be the total number of nodes involved in the hierarchy and $\gamma_{\rightarrow j}$ the number of parents of the descendent node j. Let α be the number of descendent nodes that are larger than one parent, $0 \leq \alpha < \omega$, and the size stored in the mobile agent be $[256(\omega - 1 + \alpha)]$.

The aforementioned schemes require an excessive computation time. We propose a modified scheme that provides users with a better way of key accessing based on the mobile agent. Our improved scheme became more efficient and flexible since we extracted the advantages of CRT.

3. Proposed Work

We employ the hierarchical structure shown in Fig. 1. The bottom nodes represent the encrypted confidential files. The internal nodes represent the servers or users of the system. The confidential files are accessed by the authorized server when users attempt to connect to the server. In order to compute the superkey and a generating function for each server and encrypted confidential file, mobile agents are used to accomplish these tasks.

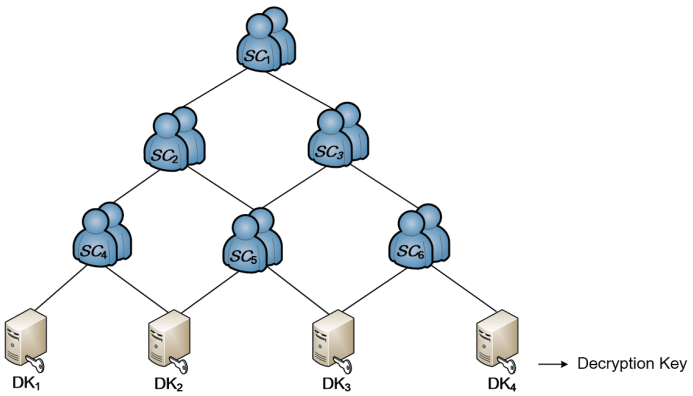

Fig. 1. (Color online) Hierarchical structure of mobile agents.

Since mobile agents are able to change their tasks, they can efficiently finish the computations and reduce the load of the system. The mobile agents can also work with different communication protocols within heterogeneous networks and overcome the incompatibility issue. In favor of CRT, a hierarchical access structure is built using the mobile agents. Each server can only access the protected confidential files based on their access level. The security of our scheme can prevent unauthorized access using the characteristics of CRT and one-way hash function.

Here, we introduce the logic and algorithm for the construction of our proposed scheme. For $1 \le j \le m$, DK_j is the secret symmetric key for encrypting and decrypting the jth confidential file, and a secret parameter r_j is also assigned to the jth confidential file. We identify DK_j with the jth confidential file in order to reduce the complexity of our presentation. The mobile agents compute a (secret) superkey SK_i for the ith server ($0 \le i \le k$) using public and secret parameters, such as t. CRT is utilized in the construction of superkey SK_i. Because of CRT mathematical properties, the constructed superkeys meet the requirements of a hierarchical structure.[10–13] Thus, if server SC_i has a higher access authorization than server SC_j (denoted by $SC_j \le SC_i$), then server SC_i can derive the superkey SK_j of server SC_j using the knowledge of its superkey SK_i and other public parameters. Moreover, each server can access only the confidential files in accordance with its position in the hierarchical structure.

The parameters are defined as follows (Table 1) before the proposed method is discussed in detail.

Defining key property of "$\le$":

Assuming that $SC_i \ne SC_j$, we cannot have both $SC_j \le SC_i$ and $SC_i \le SC_j$. If we set both $SC_j \le SC_i$ and $SC_i \le SC_j$, then $SC_1 \rightarrow SC_i \rightarrow SC_j$ exists and so does path $SC_1 \rightarrow SC_j \rightarrow SC_i$. Then, we can obtain the path $SC_1 \rightarrow SC_i \rightarrow SC_j \rightarrow SC_i \rightarrow SC_1$. This contradicts the definition of a tree (= a connected graph with no loop). This follows that the relation "$\le$" defines a partial order on the set $\{SC_1, \ldots, SC_k\}$, that is, $\le$ satisfies the following properties.

(1) Reflexive: $SC_i \le SC_i, \forall i$

(2) Transitive: $SC_q \le SC_j$ and $SC_j \le SC_i \Rightarrow SC_q \le SC_i$

$SC_q \le SC_j \Rightarrow \exists$ path $SC_1 \rightarrow SC_j \rightarrow SC_q$

$SC_j \le SC_i \Rightarrow \exists$ path $SC_1 \rightarrow SC_i \rightarrow SC_j$

Hence, $\Rightarrow \exists$ path $SC_1 \rightarrow SC_i \rightarrow SC_j \rightarrow SC_q$

That is $\Rightarrow \exists$ path $SC_1 \rightarrow SC_i \rightarrow SC_q \therefore SC_q \le SC_i$

Table 1
Parameters for constructing the system.

DK_u	DKu is the secret symmetric key for encrypting and decrypting the uth confidential file, $1 \le u \le k$.
SC_i	SC_i is the ith server, $1 \le i \le k$.
J_i	J_i is the collection of the subscripts of the confidential files to which server SC_i has accessed.
n_u	n_u is a large prime for each DK_u, $\forall u = 1, 2, \ldots, m$.
N_i	$N_i = \prod_{u \in J_i} n_u$ and note that gcd $(N_i/n_u, n_u) = 1$.
r_u	r_u is a secret parameter for DK_u, $1 \le u \le m$.
$W_{i,u}$	$W_{i,u}$ is the unique primitive multiplicative inverse of N_i/n_u (mod n_u), $1 \le u \le m$.
SK_i	SK_i is the secret superkey for SC_i, $1 \le i \le k$.
E	E is the encryption function for the confidential files.
D	D is the decryption function for the confidential files.

(3) Anti-symmetric: $SC_j \le SC_i$ and $SC_i \le SC_j \Rightarrow SC_i = SC_j$.

In the hierarchical structure, the decryption key DK_u, $(1 \le u \le m)$ in the bottommost layer is associated with encryption and decryption of the uth confidential file and the intermediate nodes represent the servers. The set $J_i = \{u: SC_i$ is authorized to access decryption key $DK_u\}$ denotes the set of subscripts of the confidential files to which server SC_i has accessed.

3.1 Key generation phase

Step 1 The mobile agent owner selects nonrepeated random integers $\{DK_1, DK_{2, \ldots,} DK_m\}$ as the symmetric encryption and decryption keys of confidential files and pairwise relative primes n_u for each DK_u, $\forall u \in \{1, 2, \ldots, m\}$. DK_u is kept secretly and n_u is a public parameter.

Step 2 The mobile agent constructs N_i for the internal node SC_i.

$$N_i = \prod_{u \in J_i} n_u \tag{1}$$

By constructing N_i, we have gcd$(N_i/n_u, n_u) = 1$, $\forall i \in \{1, 2, \ldots, k\}$.

Step 3 The mobile agent owner randomly selects distinct r_u for DK_u, which is kept secretly.

Step 4 The mobile agent owner calculates separately a unique primitive multiplicative inverse $W_{i,u}$ of $\frac{N_i}{n_u}$ modulo n_i. Thus, $W_{i,u}$ satisfies the following equation:

$$W_{i,u} \cdot \frac{N_i}{n_u} \equiv 1 \ (\text{mod } n_u), \forall u \in J_i, \forall a \le i \le k. \tag{2}$$

Step 5 The mobile agent owner calculates the superkey of SK_i of server SC_i as

$$S K_i = \sum_{u \in J_i} r_u \times W_{i,u} \times \left(\frac{N_i}{n_u}\right) (\text{mod } N_i), \forall i = 1, 2, \ldots, k. \tag{3}$$

SK_i is the unique primitive solution to the following system of congruence based on CRT.

$$y \equiv r_u \pmod{n_u}, \forall u \in J_i.$$

As a result, we obtain $SK_i \equiv r_u \pmod{n_u}, \forall u \in J_i$.

Step 6 Define and publish the one-way hash function $h(\ .\ )$. Define the generating function $f_u(x)$ for DK_u,

$$f_u(x) = \prod_{i:s\in J_i} [x - h(r_s \parallel S K_i)] + DK_u, \forall 1 \le u \le m. \quad (4)$$

The expanded form of Eq. (4), other than Eq. (4) itself, is published. We note that DK_u is embedded in the constant term of the expanded form of Eq. (4); this prevents the extraction of DK_u.

3.2 Key derivation phase

When server SC_i corresponding to an internal node attempts to access the leaf node DK_u, the following steps are performed:

Step 1 Server SC_i uses Superkey SK_i and the public parameter n_u to determine the secret parameter r_u. The formula is

$$r_u \equiv SK_i \pmod{n_u}, \forall u \in J_i.$$

Step 2 SC_i uses r_u and Superkey SK_i to compute $h(r_u \parallel SK_i)$, and then obtains DK_u using the public formula as follows:

$$DK_u = f_u(h(r_u \parallel SK_i)), \forall u \in J_i.$$

4. Analysis of Security

Our research has developed a scheme secured against potential attacks. The scheme is introduced in this section.

4.1 Reverse attack

Let SC_i and SC_j be two servers in the hierarchical structure with $SC_j \le SC_i$. By reverse attack, SC_j can use its superkey SK_j and other public parameters to compute the superkey SK_i of server SC_i.

Assuming server SC_j ($\le SC_i$), using its superkey SK_j and other public parameters, it attempts to obtain the superkey SK_i of user SC_i. To avoid triviality, we assume that J_j is a strict subset of J_i. User SC_j can use his superkey SK_j to retrieve the secret parameter r_u for the uth confidential

file for $u \in J_j$. However, SC_j is not able to obtain r_u for $u \in J_i - J_j$. SC_j without full knowledge of r_u for all $u \in J_j$ values is not also able to obtain the superkey SK_i of server SC_i. In this case, we propose that the scheme is protected against the reversal attack.

4.2 Collusion attacks

A collusion attack takes place if a group of servers SC_{j_1}, SC_{j_2}, ..., SC_{j_t} each of whom satisfies the relation $SC_{j_i} \leq SC_i$, using the pooled knowledge of their private keys SK_j and other public parameters, manage to obtain the superkey SK_i of user SC_i. Since each J_{j_l}, $1 \leq l \leq t$, is a set subset of J_i, the union $J_{j_1} \cup J_{j_2} \cup ... \cup J_{j_l}$ is still a strict subset of J_i, and without a full knowledge of all $r_u \in J_i$ values, the superkey SK_i cannot be obtained. Thus, our proposed scheme is secure against collusion attack.

4.3 External attacks

As shown below, we compare the performance of the proposed scheme with those of other schemes. We investigate the computational complexity and storage requirement of each case. The variables are defined and listed in Table 2.

We assume that an agent will visit k hosts and carry m confidential files. Let v_i be the number of files from the visited host that i can access, where $1 \leq i \leq k$.

The scheme proposed by Lin *et al.*[7] requires the storage of k private keys under/static/sctx/acl/. According to the Rivest–Shamir–Adleman (RSA) algorithm, the scheme proposed by Lin *et al.* requires storage with $1024k$ b for all private keys. One-way hash indicates the relationship between a specific node and an associated sole parent node.[9]

A hash function, such as the 256 b SHA-256, takes an arbitrary-length input and returns a fixed-length output. Thus, $256k$ b storage is required for the private keys generated by the scheme of Chen *et al.* In our proposed scheme also utilizing a hash function in the construction of the function $f_u(x)$, a storage space of 256 b is required for the private keys.

Table 2
Notation table.

Definition	Notation
k	Number of security classes
m	Number of files
ν_i	Degree of polynomial $f(x)$
len	Bit length of integer len
T_{MUL}	Time for modular multiplication
T_{INV}	Time for modular inversion
T_{EC_MUL}	Time for scalar multiplication on the elliptic curve E
T_{EC_ADD}	Time for addition/subtraction on the elliptic curve E
T_{exp}	Time for modular exponentiation
T_{hash}	Time for evaluating hash function
T_{mod}	Time for modular arithmetic operation
T_l	Time for evaluating interpolation polynomial

Now, we consider the storage spaces for public parameters. The storage spaces for public parameters required by the schemes proposed by Lin *et al.*, Chen *et al.*,[9] and our proposed scheme are $512(m + 1)$, $256(2k - 1)$, and $256m$ b, respectively. It is reasonable to assume that the number of security classes, k, is larger than that of confidential files, m, in a hierarchical structure. The storage space required by the scheme proposed by Chen *et al.* is larger than that required by our proposed scheme.

Then, we should calculate the computation time required by the key generation and derivation phases for the three schemes being compared. The scheme of Chen *et al.*[9] requires a computation time of $(k - 1 + \sum_{1 \le i \le k} v_i) T_{mod} + (k - 1 + 2 \sum_{1 \le i \le k} v_i) T_{hash}$. We should also establish the relationship among the user superkeys SK_i, $1 \le i \le k$, and require a computation time of $(k - 1)T_{hash}$ to derive all the symmetric encryption and decryption keys DK_u, $1 \le u \le m$. The scheme of Lin *et al.*[7] requires one modular exponentiation. For the scheme of Lin *et al.*,[7] a computation time of $(\sum_{1 \le i \le n} v_i) T_{hash}$ is required for both the key generation and derivation phases. Our proposed scheme demands a computation time of $(k) T_{mod} + (\sum_{1 \le i \le m} v_i) T_{hash}$ to generate all and a computation time of $(k) T_{mod} + (\sum_{1 \le i \le m} v_i) T_{hash}$ to derive all keys.

Table 3 shows the computation complexities and storage requirements of these three schemes. It is seen that the storage requirements of the private keys and public parameters for our proposed scheme are less than those for required by the other two schemes. The computation complexity of our proposed scheme is also smaller than those of the other two schemes. We note that the key operation used in the schemes of Chen *et al.* and Lin *et al.* is modular exponentiation and that used in our proposed scheme is the evaluation of a hash function. Both the schemes of Chen *et al.* and Lin *et al.* have a computation complexity of $O(k^2)$ in modular exponentiation. Our proposed scheme has a smaller computation complexity than the other two schemes because it takes less time to evaluate a hash function.

4.4 Analysis of performance

A numerical experiment is conducted to calculate the run time required for the key generation phase for each scheme under a given configuration. From the obtained results, we compare the actual performance characteristics of the three schemes.

Table 3
Analysis of computation complexity and storage.

	Key generation/derivation	Complexity	Storage for public data	Storage for keys
Lin *et al.* (2004)	$(2 \sum_{1 \le i \le k} v_i) T_{\exp}$	O(k^2) in modular exponentiation	512(m+1)	*len*
Chen *et al.* (2009)	$(k - 1 + \sum_{1 \le i \le k} v_i) T_{\mathrm{mod}}$ $+(2k - 2 + 2 \sum_{1 \le i \le k} v_i) T_{hash}$	O(k^2) in hashing	256($2k$ − 1)	*len*
The proposed	$(2k) T_{\mathrm{mod}} + (\sum_{1 \le i \le k} v_i + m) T_{hash}$	O(k^2) in hashing	256(m)	*len*

All the numerical calculations reveal that the mathematical package MATLAB (2011b version) on a personal computer with an Intel Core i7 2.67 GHz CPU and 8 GB memory is efficient. The operation system is Windows 7, 64 b version. According to the method proposed by N. Koblitz,[15] the calculation time of the parameters in the stage of generation and derivation is given. Each configuration is run twenty times and all the run times are recorded and plotted in order to obtain a higher numerical accuracy.

The numerical experiment is divided into two sections. In the first section, the run time for key generation is calculated. In the second section, the run time for key derivation is calculated. For the former, we access functions, and for the latter, we calculate the time it takes for server SC_1 to derive a symmetric encryption/decryption key. This is performed because SC_1 has the highest degree of authorization and will take more time to derive a symmetric encryption/decryption key than any other users who perform the same task.

A plot of the run time (y-axis) required by the key generation phase vs the number of servers (x-axis) is given in Fig. 2. In Fig. 2, the run times required by the key generation phase for the schemes of Chen *et al.* and Lin *et al.*, and our proposed scheme are represented by the blue, green, and red lines, respectively. It is seen from Fig. 2 that our proposed schemes perform better than the other two schemes. As the number of servers increases, the superiority of our proposed scheme becomes more pronounced (as the gap between the red line and the other two lines becomes wider). It is verified by the performance curves in Fig. 2 that our proposed scheme requires the least run time in the key generation phase.

In Fig. 3, we give the performance curve for the key derivation phase. For the hierarchy of 1200 servers, the times required for the key derivation phase are 3.04, 3.22, and 3.29 for our proposed scheme and the schemes of Chen *et al.* and Lin *et al.*, respectively. It is seen from Fig. 3 that the performance of the key derivation phase of our scheme is higher than those of the other two schemes.

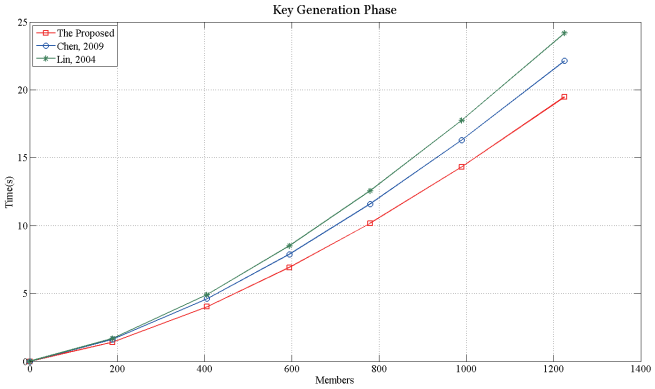

Fig. 2. (Color online) Key generation phase.

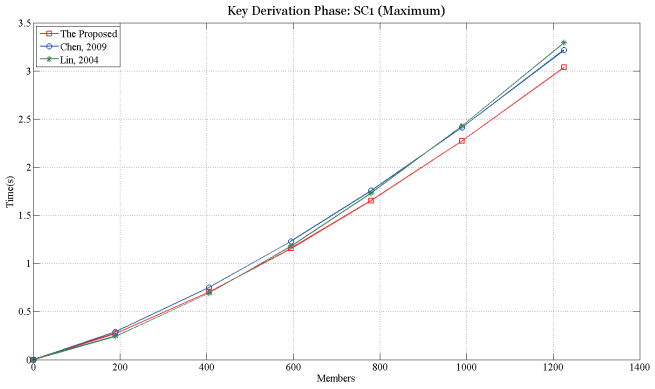

Fig. 3. (Color online) Key derivation phase.

5. Conclusion

In this research, a hierarchy-base information system that is suitable for various conditions is developed. The system is rounded by the Internet. Otherwise, we use the characteristics of the mobile agents to adopt the heterogeneous networks and roam among servers. Data can be collected from a diverse array of monitoring devices and useful information is also derived after mobile agents have communicated among servers and by utilizing the scalability and openness properties. The basis of the control system is CRT. The security analysis indicates that the proposed system is secure against various types of attack. Our proposed system is flexible and efficient since the mobile agents are assigned to send the results of computation.

Acknowledgments

This work was supported by the Natural Science Foundation of Fujian Province of China (No. 2017J01109), the Education Department of Fujian Province (No. JA15031), and the Science & Technology Planning Fund of Quanzhou (No. 2016T009).

References

1 G. J. Simmons: Contemporary Cryptology: The Science of Information Integrity (IEEE Press, Piscataway, 1992) pp. 177–288.
2 W. Jen, C. Chao, M. Hung, Y. Li, and Y. Chi: Int. J. Med. Inf. **76** (2007) 565.
3 T. C. Wu, T. S. Wu, and W. H. He: Comput. Syst. Sci. Eng. **10** (1995) 92.
4 X. Zou, B. Ramamurthy, and S. S. Magliveras: 3rd Int. Conf. Information Communications Security **2229** (2001) 381.
5 E. Bierman, T. Pretoria, and E. Cloete: Proc. Annu. Research Conf. Port Elizabeth, South Africa (2002) 141.
6 I. Ray and N. Narasimhamurthi: ACM Symp. Access Control Model Technologies (2002) 65.
7 I. C. Lin, H. H Ou, and M. S. Hwang: Comput. Stand. Interfaces **26** (2004) 423.
8 M. Markovic, Z. Savic, and B. Kovacevic: M-Health, Emerging Mobile Health Systems, R. Istepanian, S. Laxminarayan, and C. S. Pattichis, Eds. (Springer, 2006) p. 81.
9 H. B. Chen, C. W. Liao, and C. K. Yeh: WSEAS Trans. Commun. **8** (2009) 1106.
10 R. Sandhu: Inf. Process. Lett. **27** (1988) 95.
11 M. S. Hwang: Int. J. Comput. Math. **73** (2000) 463.
12 H. B. Chen, W. B. Lee, C. W. Liao, and C. H. Huang: 1st Int. Workshop on Privacy and Security in Agent-based Collaborative Environments, Future University-Hakodate, Japan (2006).
13 T. C. Hsiao, T. L. Chen, C. H. Liu, C. M. Lee, H. C. Yu, and T. S. Chen: Math. Prob. Eng. (2014) Article ID 910820.
14 Y. F. Chung, T. C. Hsiao, and S. C. Chen: Wireless Personal Commun. **79** (2014) 1063.
15 N. Koblitz, A. Menezes, and S. Vanstone: Des. Codes Cryptography **19** (2000) 173.

About the Authors

Tsung-Chih Hsiao received his Ph.D. in the Department of Computer Science and Engineering, National Chung Hsing University, Taiwan. He is currently an instructor in the College of Computer Science and Technology at Huaqiao University, China. Research fields include information security, cryptography, and network security.

Yin-Tzu Huang received her B.A. and M.S. degrees in the Department of Information Management at Tunghai University. She is currently a Ph.D. student in the Department of Electrical Engineering (Computer Science) at National Taiwan University. Research fields include information security, networks, and cloud computing.

Yu-Min Huang received her Ph.D. in the Department of Statistics at the University of Minnesota Twin Cities, United States. She is currently an assistant professor in the Department of Statistics at Tunghai University, Taiwan. Research fields include time series, regression, and data modeling.

Tzer-Long Chen received his Ph.D. in the Department of Information Management, National Taiwan University, Taiwan. He is currently an assistant professor in the Department of Information Technology at Lingtung University, Taiwan. Research fields include information security, cryptography, and network security.

Tzer-Shyong Chen received his Ph.D. in the Department of Electrical Engineering (Computer Science) at National Taiwan University, Taiwan. He is currently a professor in the Department of Information Management at Tunghai University, Taiwan. Research fields include information security, cryptography, and network security.

Sheng-De Wang received his Ph.D. in the Department of Electrical Engineering at National Taiwan University. He is currently a professor in the Department of Electrical Engineering (Computer Science) at National Taiwan University. His main research interests are on the design and analysis of embedded systems, cloud computing, and IoT data analysis.