Sensors and Materials, Vol. 31, No. 2 (2019) 355–364
MYU Tokyo

S & M 1776

Elliptic Curve Cryptosystems-based Date-constrained Hierarchical Key Management Scheme in Internet of Things

Tsung-Chih Hsiao,[1] Tzer-Long Chen,[2] Tzer-Shyong Chen,[3*] and Yu-Fang Chung[4]

[1]School of Arts, Southeast University, Nanjing 211189, China
[2]Department of Information Technology, Ling Tung University, Taichung 40852, Taiwan
[3]Department of Information Management, Tunghai University, Taichung 40704, Taiwan
[4]Department of Electrical Engineering, Tunghai University, Taichung 40704, Taiwan

(Received May 23, 2018; accepted October 18, 2018)

Keywords: mobile agent, mobile security, Internet of Things, key management, date-constrained

In this paper, we propose a new mechanism to improve the disadvantage of the security mechanism proposed by a scholar and then fulfill the demands of Internet of Things (IoT) to go through the decentralized environment access control functions. We also propose the date-constrained hierarchical key management scheme for mobile agents. With elliptic curve cryptosystems (ECCs) and discrete logarithms, the proposed scheme is flexible. Moreover, the duration of access for each security class is restricted with a certain authorized discrete time period. We demonstrate the mathematical derivation and arguments for our scheme and further conduct a numerical trial. The constructed scheme could meet security needs and be more space-efficient.

1. Introduction

Owing to the advantages of elliptic curve cryptosystems (ECCs), the scheme can recognize an access control object faster with a small key storage space. In this study, the scheme was constructed to reduce the work of key management, minimize key storage, enhance the computation time of construction and derivation phrases, and provide high flexibility and security.

An ECC was proposed by Koblitz(1) and Miller(2) in 1985. To improve the existing cryptosystems,(3) the proposed ECC was used to reduce system parameters, public key certificates, bandwidth usage, power consumption, and hardware processor requirement, and for rapid implementation. Thus, the ECC with its advantages is useful for building a cryptosystem with high security and efficiency.(4) The mathematical illustration is described below.(4,5)

Elliptic curves are categorized into two families: prime and binary curves. Prime curves (Z_p) are suitable for use in software applications since they do not need to be extended for bit-fiddling operations. On the other hand, binary curves [GF(2n)] are for hardware applications since they need a small number of logic gates to build a cryptosystem. With the property of elliptic curves, the efficiency of ECC computing operation increases.

[*]Corresponding author: e-mail: arden@thu.edu.tw

https://doi.org/10.18494/SAM.2019.2085

ISSN 0914-4935 © MYU K.K.
https://myukk.org/

2. Previous Work

In 1998, Volker and Mehrdad(5) designed a tree-structure-based security scheme of securing a safer place for mobile agents. The functions of this mechanism are distinguished into three categories: mobile agent authorization, key management, and access control. Thus far, Jeng and Wang,(6) Chung *et al.*,(7) Nikooghadam *et al.*,(8) and Lin and Hsu(9) have contributed to the incipience of a reliable and effective scheme for mobile agents.

In 2006, Jeng and Wang(6) applied an ECC as a key management scheme to efficiently solve hierarchical access control problems. In the key derivation phase, a predecessor accesses the authorized files to derive encryption/decryption keys. We can use not only a secret key that is private to itself, but also the successor-related public information. Nevertheless, the scheme proposed by Jeng and Wang(6) had a loophole in security, which makes it possible for any outsider to derive an unauthorized encryption key. The relationship between any security classes was updated. The scheme proposed by Lin and Hsu(9) indicated such a flaw in the Jen and Wang(6) scheme by suggesting that an adversary could further derive the encryption key $k_{j,2}$ of the security class $k_{j,2} = f_j(v_{l,j})$ without knowing any secret information.

In 2008, another key management scheme was introduced by Chung *et al.*(7) This key management scheme was a novel and efficient solution to the dynamic access control problems in a user hierarchy by means of ECCs and one-way hash functions. The scheme introduced by Chung *et al.*(7) was different from that of Jen *et al.* in the application of polynomials. In Jen *et al.*'s scheme, each security class selected its own secret key and then sent the secret key to the Certificate Authority (CA) via a secure way, whereas in Chung *et al.*'s scheme, the same public polynomials were used in key generation and derivation phases. The CA was responsible for selecting all the secret parameters and sending them to the corresponding security classes via a secure way. On the other hand, constructing the interpolating polynomials requires both tremendous storage accommodation and a colossal amount of computational overhead. According to Knuth,(10) the cost of constructing an interpolating polynomial of degree m is derived by m additions, $2m^2 + 2$ subtractions, $2m^2 + m - 1$ multiplications, and $m + 1$ divisions. With respect to expenditure, Chung *et al.*'s scheme requires large computational expenditures and this suggests the considerable consumption of system resources to access confidential files. Hence, Nikooghadam *et al.*(8) introduced an ECC-based improved method for access control and key management.

In 2009 and 2011, an improved version was raised by Jeng *et al.*'s scheme. They replaced $(\tilde{A}(n_jP_i)),K_i)$, which was proposed in Jeng *et al.*'s scheme with $(h(r \,||\, \tilde{A}(n_jP_i)),K_i)$ using a random number, r, and the one-way hash function $h(\cdot)$. This equation implies that $\tilde{A}(n_jP_i)$ is not a solution of $\tilde{f}_i(x) - f_i(x) = 0$ anymore. The preference for this method over Jen *et al.*'s scheme is due to the fact that it can effectively eliminate the security flaw mentioned above.

The elliptic curve discrete logarithm problem (ECDLP) is significantly more difficult and has a larger computational complexity than the integer factorization or discrete logarithm problem.(11) To satisfy security requirements, the ECC needs a comparatively smaller key size than the other cryptosystems. ECC-based access control schemes enjoy high security performance at the expense of bulky mobile agent codes and excessive calculations for encryption/decryption keys.

This mechanism based on the ECC theory, therefore, is more efficient and less computationally complex with respect to key generation and derivation. Compared with the other published schemes, in addition to using elliptic curve cryptography, our scheme also incorporates the concepts of elliptic curve digital signature and data constraint. The purpose of using the concept of elliptic curve digital signature is to ensure that a private key is generated for a user only at a legal time granule as a data-bound warrant.

The purpose of using the Internet of Things (IoT) is to be able to share resources and information. IoT, itself, provides an open and public manipulation environment. The heterogeneity of data enables the management and sharing of resources and information. However, ensuring the confidentiality, correctness, and availability of the legally stored information definitely becomes a challenge for sharing information from the past to the present. Since the environment of the internet is unpredictable, this often leads to security problems, such as unauthorized access requirements, data being compromised or unauthorized access, and privacy disclosure; these issues can reveal the necessity and importance of the access control mechanism. Simultaneously, based on some scholars, Volker and Mehrdad(5) suggested some methods of access control to the acting agent and key management mechanism. These will consume the agent's time and cause security problems. When the application of mobile agents roams the internet, it may be attacked by unfriendly agents or the host or the agents will arrive at an unfamiliar or unknown host. This situation will lead to tampering or inaccurate execution of delivered tasks, resulting in private information being peeped or stolen. In this paper, therefore, an access control mechanism is proposed. The application of a mobile agent who is in a hierarchical relationship structure can adequately use the one-way hash function, the concept of time series, and ECC to ensure the security of the key; simultaneously, it can give the permission classification in order to achieve security.

3. Proposed Scheme

Akl and Taylor's proposed access control scheme(12) was based on a hierarchical structure model, which was obtained by assigning each user to a security class, which can be represented as $SC = \{SC_1, SC_2, SC_3, \ldots, SC_k\}$. On the basis of the hierarchical structure, the access relationship between one security class and another can be denoted by $SC_i \geq SC_j$. For instance, the class of SC_i is at a higher hierarchy than SC_j and their relational representation is $SC_i \geq SC_j$. The higher the hierarchy, the more authority to access the information. This means that the user SC_i has the authority to access the information available to SC_j. As the hierarchy network grows, SC_i would have to accommodate a growing number of private keys held by groups at a lower hierarchy. It is considered that a lower hierarchy would cause key management problems and security issues. Thus, Akl and Taylor raised the concept of superkey in place of key. In this manner, key management issues can be resolved. On the basis of the determined hierarchical structure of $SC_i \geq SC_j$, the user SC_i uses mathematical operations to obtain the SC_j's superkey with his superkey.

Figure 1 shows an illustration of an improved version of Akl and Taylor's structure. Among the leaf nodes of a hierarchical structure, we shall offer an explanation regarding the access of

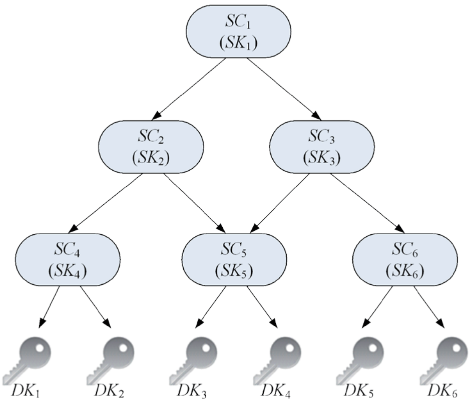

Fig. 1. (Color online) Structure of decryption keys for mobile agent.

confidential files. $file_j$ is an encrypted confidential file with SC_i as an internal node, which also represents a user. SK_i represents the secret key held by the user SC_i. When SK_i is authorized to have access to some encrypted files, it can obtain them from their corresponding leaf nodes. To elucidate the matter, we shall take the node SC_2 as an example. SC_2 holds the secret key SK_2 and the structural hierarchy suggests that it has authorization to access $file_1$, $file_2$, $file_3$, and $file_4$.

Before a mobile agent executes its assignments by being linked to the Internet, the mobile agent user must decide on the host to be visited by the mobile agent and on the information to be accessed by the visited host. Afterwards, the mobile agent user has his/her own access policy. This is based on the fact that he/she will construct an accessible network and assigns a different secret key to the corresponding internal node. A user can thus use his/her secret key to obtain the encryption/derivation key of confidential files. The concept of date constraint control is introduced so that access control in a hierarchical structure will become more effective. The effectiveness is achieved by the set-up scheme that will allow a predetermined time interval usage of his/her secret key by the user. Therefore, if the time interval is not the one that it should preset, the secret key holder will not be able to access the file with the key.

We propose a scheme that offers a secure, robust, and efficient hierarchical key management for a mobile agent against interior and exterior attacks. This scheme also has the advantage of allowing each visited host to maintain only one secret key that is used to derive the decryption key of confidential files. This means that the key management costs will be reduced. The cost of the ECC is low and the key size is small. Therefore, the application of the ECC to the mobile devices becomes appropriate. The limitation of bandwidth and storage space can be resolved by their common constraint. The mobile agent can follow the steps below to construct an accessible network.

3.1 Initialization phase

The following steps are executed by the CA in the initialization phase.

Step 1. The CA uses $y^2 = x^3 + ax + b \pmod{p}$ to define an elliptic curve $E_p(a, b)$ with the coefficients a and b satisfying $4a^3 + 27b^2 \pmod{p} \neq 0$, and p being a large prime number.

Step 2. The CA selects a base point $G = (x, y)$ on $E_p(a, b)$.

Step 3. Suppose that two users, SC_i and SC_j, obey the rule $SC_i \geq SC_j$. The CA uses $PK_i = SK_iG \pmod p$ to realize the assignment of the private key SK_i to SC_i and to realize the computation and publication of the public key PK_i of SC_i.

3.2 Key assignment phase

Step 1. In addition to the traveling route and access control of the agent, the CA establishes the lifetime Z and the servers of the mobile agent.

Step 2. The CA selects a random number a or s to establish the legal time interval $[t_1, t_2]$ and then computes the published T_b and T_e from $T_b = H^{t_1}(a)$ and $T_e = H^{Z-t_2}(a)$.

Step 3. The CA selects a secret random number k and applies the parameters T_b and T_e calculated in the previous step to acquire a signature date-bound warrant $W = (T_b, T_e)$, and then calculates and publishes the public parameters R, r, and s illustrated below:
$R = k \times G = (x_1, y_1)$,
$r = x_1 \pmod p$,
$s = k^{-1} \times (H(W) - SK_i \times r) \pmod p$.

3.3 Key derivation phase

The private key SK_j is generated by the mobile agent for SC_j, which can transmit information to SC_i for SK_j to acquire the private files of SC_j at the time granule t.

Step 1. The server SC_i computes $H^t(a)$ and $H^{Z-t}(a)$ at the time granule t. If the equation $(H^t(a), H^{Z-t}(a)) = (H^{t-t_1}(T_b), H^{Z-t_2}(T_e))$ is established, then t is a legal time; otherwise, t is not a legal time.

Step 2. The public parameters r, s, and W are the elements of the server SC_i for calculating the secret number k as $k = s^{-1} \times [(H(W) - SK_i \times r)]^{-1} \pmod p$.

Step 3. The server SC_i generates the private key SK_j for SC_j from $SK_j = H(k, ID_j) \oplus H^t(a) \oplus H^{Z-t}(a)$ with ID_j as the public identity of SC_j.

3.4 Time warrant key signature verification phase

The server SC_i calculates $V_1 = r \times PK_i + s \times R \pmod p$ and $V_2 = H(W) \times G$. At $V_1 = V_2$, the signature for the time warrant $W = (T_b, T_e)$ is confirmed to be valid.

Solution of the equation:

$$
\begin{aligned}
V_1 &= r \times PK_i + s \times R \\
&= r \times PK_i + (k^{-1} \times (H(W) - SK_i \times r)) \times R \ (\because s = k^{-1} \times (H(W) - SK_i \times r) \pmod p) \\
&= r \times PK_i + (k^{-1} \times (H(W) - SK_i \times r)) \times (k \times G) \ (\because R = k \times G) \\
&= r \times PK_i + (H(W) - SK_i \times r) \times G \\
&= r \times PK_i + (H(W) \times G - SK_i \times r \times G) \\
&= r \times PK_i + (H(W) \times G - PK_i \times r) \ (\because PK_i = SK_iG \pmod p) \\
&= H(W) \times G \\
&= V_2.
\end{aligned}
$$

4. Analysis of Security

This section provides a security analysis to examine the security of practical applications. Four types of attack that are likely to impact system security are under examination.

4.1 Reverse attack

If a user attempts to use his private key and other public information in order to derive a user's private key of higher access authority than his own private key, a reverse attack will occur. In the proposed scheme, the CA assigns a private key SK_i to the user SC_i. If the server SC_i intends to use his private key SK_i and other public parameters in the hope of generating the private key SK_j of the server SC_j, the server will be blocked because the hierarchy $SC_i \geq SC_j$ indicates that the lower hierarchy has the one-way property of the hash function. The private key SK_j of SC_j cannot be used to derive the private key SK_i of the user SC_j. Likewise, the public key PK_i of SC_j cannot be used to derive the private key SK_i of the user SC_i because solving an ECDLP is difficult.

4.2 Collusion attack

If a group of users colludes to share the knowledge of their private keys and other public information in the hope of deriving the private key of a user with a higher hierarchy than any member of the group, a collusion attack will occur. We note that each attacker receives his private key either directly from the CA or a user with higher authority. The collusion attack will never occur because the pooled and common knowledge is not practical. Neither can the attackers break the one-way property of a hash function nor can they resort to their pooled knowledge to solve the ECDLP.

4.3 External collective attack

If a group of unassociated attackers with the hierarchy attempts to collectively combine their resources and efforts to derive the private key of a user within the hierarchy, then an external collective attack will occur. We note that any user of the hierarchy has knowledge different from that of external attackers. It is impossible for any user of the hierarchy to implement a successful reverse attack; thus, implementing a successful external collective attack is also out of the question.

4.4 Date alteration attack

If the user SC_i intends to inspire a private key SK_j in place of SC_j as the equation $SC_i \geq SC_j$ suggests, which leads to the deviation from the legal time internal [T_1, T_2] specified by the CA, then a date alteration attack will occur. We note that if a time granule A deviates from the legal time internal [T_1, T_2], then the equation of $H^{t-t1}(T_b)$ on $H^{t2-t}(T_e)$ in step 1 of the key derivation phase will be invalid. The scheme that we propose has an embedded data-constraint mechanism so the idea of date alteration attack is only theoretical.

4.5 Analysis of performance

In this section, we compare the computation complexities(13–17) and storage requirements in Chung *et al.*'s,(7) Nikaooghadam *et al.*'s,(8) and Lin and Hsu's(9) schemes and in our proposed scheme. Table 1 shows the computation complexities and storage requirements in the four schemes mentioned above.

Chung *et al.*'s scheme states that the CA requests a computation time of $\sum_{i=1}^{k} v_i(T_{EC\_MUL}+T_{hash})$ to compute all $s_iG_j=(x_{j,i} \parallel y_{j,i})$ and h($x_{j,i} \parallel y_{j,i}$), and that of $\sum_{i=1}^{k} v_iT_{EC\_MUL}$ to construct *n* polynomials. Deriving the successor's encryption keys takes a computation time of $T_{EC\_MUL} + v_{\mathrm{i}}T_{MUL} + T_{hash}$ for each class.

Nikooghadam *et al.*'s scheme contends that a computation time of $\sum_{1 \le i \le k} v_i(T_{EC\_MUL}+T_{MUL})$ is required to determine the public parameters $k_{i,j}G$ from the secret parameters k_i. mT_{EC_MUL} is the computation time for the calculation of c_iG. To further calculate $M_{i,j}$, $F_{i,j}$, and $s_{i,}$,

Table 1
Analysis of computation complexity.

	Key generation/ derivation	Complexity	Storage of public parameters	Storage of private keys
Chung *et al.* (2008)	$(2\sum_{1\le i\le k} v_i+1)T_{EC\_MUL}$ $+\sum_{1\le i\le k} v_iT_{MUL}$ $+\sum_{1\le i\le k} (v_i+1)T_{hash}$	$O(k^2)$ in modular exponentiation on elliptic curve *E*	$(3k+\sum_{1\le i\le k} v_i+2)len$	2 *len*
Nikooghadam *et al.* (2010)	$(m+6\sum_{1\le i\le k} v_i)T_{EC\_MUL}$ $+(4\sum_{1\le i\le k} v_i)T_{MUL}$	$O(k^2)$ in modular exponentiation on elliptic curve *E*	$(k+3\sum_{1\le i\le k} v_i+3)len$	2 *len*
Lin and Hsu (2011)	$3kT_{EC\_MUL}+kT_{INV}$ $+(2k+2\sum_{1\le i\le k} v_i)T_{MUL}$ $+(3k+2\sum_{1\le i\le k} v_i+2)T_{hash}$	$O(k)$ in modular exponentiation on elliptic curve *E*	$(3k+\sum_{1\le i\le k} v_i+4)len$	*len*
Proposed	$(k+1)T_{EC\_MUL}$ $+(\sum_{1\le i\le k} v_i+3z+2)T_{hash}$ $+(k+2)T_{MUL}+(k+1)T_{INV}$	$O(k)$ in modular exponentiation on elliptic curve *E*	$(2k+\sum_{1\le i\le k} v_i+4)len$	*len*

a computation time of $(2\sum_{1\le i\le k} v_i)T_{EC\_MUL}$ + $(3\sum_{1\le i\le k} v_i)T_{MUL}$ is required. An additional computation time of $3v_iT_{EC\_MUL}+v_iT_{MUL}$ is required in the derivation of all secret keys authorized by the corresponding hosts. Therefore, the concluding computation time in Nikooghadam *et al.*'s scheme is $[m+6\sum_{1\le i\le k} v_i]T_{EC\_MUL}+[4\sum_{1\le i\le k} v_i]T_{MUL}$. As for Lin and Hsu's scheme, a computation time of $(3kT_{EC\_MUL}+2kT_{MUL}+3kT_{hash}+kT_{INV})$ is required for the CA to compute all secret parameters v_i and all $k_{i,2}$. Furthermore, constructing n polynomials takes a computation time of $\sum_{i=1}^{k} v_i(T_{MUL}+T_{hash})$. Each security class derives the successor's encryption keys by using a computation time of $(2T_{hash}+v_iT_{MUL})$. To conclude from the implications of the above equations in Lin and Hsu's scheme, the total computation time of $3kT_{EC\_MUL}+kT_{INV}+(2k+2\sum_{i=1}^{k} v_i)T_{MUL}+(3k+2\sum_{i=1}^{k} v_i+2)T_{hash}$ is required.

We propose a scheme where the CA requires a computation time of $(k+1)T_{EC\_MUL}$ $+(z+k)T_{hash}+(k+1)T_{MUL}+kT_{INV}$ to compute PK_i, (T_b, T_e), and the public parameters r, s, and R. Furthermore, to construct n polynomials, an additional computation time of $(\sum_{i=1}^{k} v_i+Z)T_{hash}$ is considered. Each security class derives the successors' encryption keys by using a computation time of $(T_{INV}+T_{MUL}+(Z+2)T_{hash})$. We therefore conclude that the total computation time of $(k+\sum_{i=1}^{k} v_i)T_{EC\_MUL}+(\sum_{i=1}^{k} v_i+3z+2)T_{hash})+(k+2)T_{MUL}+(k+1)T_{INV}$ is required.

Chung *et al.*'s scheme and Nikooghadam *et al.*'s scheme have the same computation complexity $O(k^2)$ in modular exponentiation on an elliptic curve E. On the other hand, Lin and Hsu's scheme and our proposed scheme have the same computation complexity $O(k)$ in modular exponentiation on an elliptic curve E.

We now consider the computation times of the four schemes required by the key derivation phase. Figure 2 follows a structure similar to that of Fig. 3, showing us the plots of the computation times. As the number of hierarchy numbers increases above 1200, the respective computation times of our proposed scheme, Lin and Hsu's scheme, Chung *et al.*'s scheme,

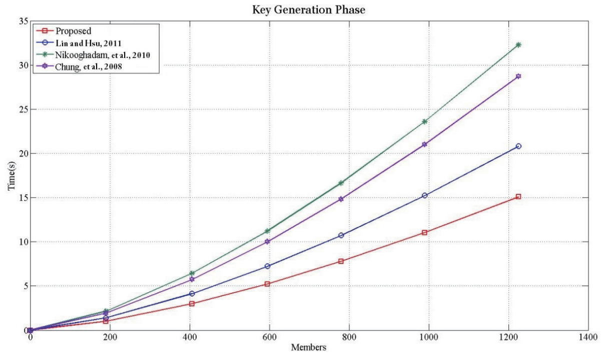

Fig. 2. (Color online) Key derivation phase.

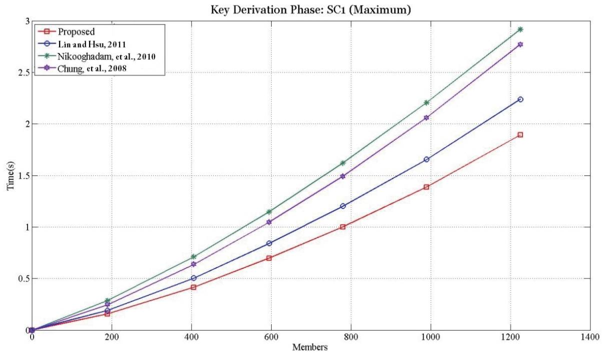

Fig. 3. (Color online) Key generation phase.

and Nikooghadam *et al.*'s scheme are 1.89, 2.24, 2.77, and 2.92, respectively. Owing to the effectiveness of the overall access control performance in the proposed scheme, the performance characteristics of the other three schemes in terms of key generation and key derivation phases are eclipsed.

5. Conclusions

Nowadays, the internet environment is conducive to the adaptation of mobile agents since mobile agents can efficiently use network resources and help improve organizational efficiency by reducing a variety of costs. Mobile agents have potential in the market because of their various applications. For example, mobile agents are important to e-commerce. However, the security problems and threats of mobile agents remain and are yet to be solved. Thus, our primary tasks are to minimize security problems, accelerate system operation, and reduce the required storage room of mobile agents. A more complete security system structure of mobile agents is welcomed.

We introduce a date-constrained hierarchical key management scheme, in which a date constraint is imposed on a key. That is, once a key exceeds the preset expiry date, the key user cannot continue to access information with the key anymore, which makes the key management system more secure. Moreover, with the help of elliptic curve cryptography, the access space of keys and key generation calculations can be reduced. In terms of security, the use of ECCs to generate keys also makes a mobile agent more secure. The ECDLP is known for its complexity and difficulty; in comparison, ECCs can use comparatively short keys for considerable protection. Our proposed scheme also has the advantage of reducing key generation calculations, which is helpful in lowering the system load. We comprehensively analyzed four different possible security attacks in order to make our proposed scheme verifiable. The results demonstrate that our proposed scheme can be used in practice, being applicable on the Internet and insusceptible to attacks by malicious users. The users of our proposed scheme can rest assured that the data transmitted through a mobile agent platform is encrypted.

References

1 N. Koblitz: Math. Comput. **48** (1987) 203.
2 V. S. Miller: Proc. Crypto'85 **218** (1986) 417.
3 H. B. Chen, W. B. Lee, C. W. Liao, and C. H. Huang: 1st Int. Workshop Privacy and Security in Agent-based Collaborative Environments (2006) 120–127.
4 S. T. Lin: Doctoral Dissertation, National Taiwan University of Science and Technology, Taipei, Taiwan (2005).
5 R. Volker and J. S. Mehrdad: Comput. Graphics **22** (1998) 457.
6 F. G Jeng and C. M. Wang: Syst. Software **79** (2006) 1161.
7 Y. F. Chung, H. H. Lee, F. Lai, and T. S. Chen: Inf. Sci. **178** (2008) 230.
8 M. Nikooghadam, F. Safaei, and A. Zakerolhosseini: Int. Conf. Parallel, Distributed and Grid Computing (IEEE, 2010).
9 Y. L. Lin and C. L. Hsu: J. Syst. Software **84** (2011) 679.
10 D. E. Knuth: The Art of Computer Programming (Addison-Wesley, Reading, MA, 1998) 3rd ed.
11 G. Liu, L. Li, J. Zheng, and Z. Li: Comput. Des. Appl. (ICCDA) **5** (2010) 581.
12 S. G. Akl and P. D. Taylor: ACM Trans. Comput. Syst. **1** (1983) 239.
13 C. L. Hsu and T. S. Lin: Comput. Secur. **22** (2003) 453.

14 J. Yeh: ACM Int. Conf. Information and Knowledge Management (2005) 285–286.
15 T. C. Hsiao, T. L. Chen, C. H. Liu, C. M. Lee, H. C. Yu, and T. S. Chen: Math. Prob. Eng. **2014** (2014) Article ID 910820.
16 Y. F. Chung, T. C. Hsiao, and S. C. Chen: Wireless Pers. Commun. **79** (2014) 1063.
17 Y. L. Lin, C. L. Hsu, T. C. Lin, S. L. Yen, and C. L. Tseng: 43rd Annu. 2009 Int. Carnahan Conf. (2009) 335–338.