Available online at www.sciencedirect.com
ScienceDirect
Computer Standards & Interfaces 31 (2009) 40–47
COMPUTER STANDARDS & INTERFACES
www.elsevier.com/locate/csi

Efficient migration for mobile computing in distributed networks

Kuo-Hsuan Huang [a], Yu-Fang Chung [b,*], Chia-Hui Liu [a], Feipei Lai [a,c,d], Tzer-Shyong Chen [e]

[a] Department of Electrical Engineering, National Taiwan University, Taiwan
[b] Department of Information Management, Chaoyang University of Technology, Taiwan
[c] Department of Computer Science and Information Engineering, National Taiwan University, Taiwan
[d] Graduate Institute of Biomedical Electronics and Bioinformatics, National Taiwan University, Taiwan
[e] Department of Information Management, Tunghai University, Taiwan

Received 12 February 2007; received in revised form 16 September 2007; accepted 10 October 2007
Available online 22 October 2007

Abstract

The speed and convenience of the Internet makes it advantageous to online applications. Basing on the elliptic curve cryptosystem, this study proposes a hierarchical mobile agent framework for handling key management and access control problems between mobile agent and host. It raises the security of key management, and also controls access to distributed environment in non-specific network. The proposed method successfully secures the accessing relationship between the mobile agent and the host while economizing the exhaust of storage space. Such an achievement lets the mobile agent operate efficiently, and puts in order a secure execution environment for mobile computing.

Keywords: Elliptic curve cryptosystem; Mobile agent; Access control; Mobility; Distributed environment

1. Introduction

The increasing popularity of the Internet has made distribution of information via the Network increasingly common. Therefore, networks have become complex and bulky. The work of the network administrator is also increasingly important. In the conventional centralized network administration method, a host must exchange messages and data with clients. However, large networks increase workload which in turn increases net dataflow. The heavy dataflow leads to a drop in work efficiency. Also, the frequent exchange of data between host and clients also uses up large amount of network bandwidth and eat into efficiency. Thus, the current system administration, which tends toward large and distributed networks, faces a considerably large problem. Questions regarding its dependability, expandability, interactivity, and inelasticity are raised.

A Mobile agent is a kind of software program that can migrate from one host to another in a heterogeneous network. It can communicate and interact with other agents and the distributed resource system on heterogeneous networks. A mobile agent can also decide when to migrate, and which nodes to access. After migrating to the desired host, the mobile agent resumes the execution of previously broken off or awaiting tasks. On completing the task, the mobile agent returns the result to the client. Therefore, the client need not be constantly connected to the server. This not only saves a lot of unnecessary transmission load, but also helps in the application of mobile calculation. The mobile agent can single-handedly execute all tasks assigned by the user. It can meet and interact with other agents when necessary while still executing its task. Therefore, a mobile agent can be viewed as an independent program. A manager need only assign tasks to a mobile agent. The agent then migrates to the remote network management servers to assign tasks to the servers. Once the tasks are completed, the mobile agent brings the result back to the manager for analyses and processing. With this kind of entrusted distributed network management server, increased workload which is often brought

* Corresponding author.
E-mail address: yfchung@cyut.edu.tw (Y.-F. Chung).

0920-5489/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.csi.2007.10.011

on by expansion of network can be prevented. Generally speaking, mobile agent has qualities like enhanced flexibility, network traffic reduction, support for disconnected operations, roaming ability on heterogeneous environment, and fault tolerance [1]. Because of the afore-mentioned qualities, mobile agent is currently widely in use in numerous areas like distribution information retrieval [2–4], electronic commerce [5–7], medical data transmission [8,9], and network management [10].

A mobile agent continuously visits other agents to exchange information while roaming around the servers in the networks carrying out its tasks. However, convenient as a mobile agent is, there are several potential security risks [11] in making contact with others in the Internet, especially for business activities. The security risks occur in different situations as described below.

One, protecting hosts from access by unauthorized parties. Two, protecting hosts from attacks by malicious agents, e.g., an agent might assume the identity of legal agent to request for services. Three, protecting agents from attacks by another agent, e.g., an agent tries to hinder another agent by constantly sending messages to the agent, causing the receiving end server to overload and the computation time to increase. Finally, protecting agents from attacks by malicious hosts, e.g., a host might impersonate an agent so as to deceive the agents or to ignore the requests of the agents. A malicious host deliberately delays an agent's request, or even terminates an agent's connection without warning causing other agents awaiting response from this agent to enter into a deadlock. Also, a malicious host can deliberately make an agent fail to carry out his task causing the agent to be live locked.

The first three risks are caused by the agents. They are resolvable through cryptography technologies that can control agents' access. The final risk is caused by a malicious host that controls the accessing and roaming activities of mobile agents. It is quite difficult to prevent a malicious server from attacking. Therefore, this work is intended as a solution to the attacks from the host authorities.

Numerous attempts to raise the security of mobile agents were made by scholars. Corradi et al. [12] presented a mobile agent structure called SOMA. The structure was built with an agent, agent server, management system, and security approach. With the same functions as SOMA, Karnik and Tripathi [13] proposed a structure called Ajanta. Besides, a tree structure was developed by Volker and Mehrdad [14], with the functions of mobile agent authorization, key management, and access control. However, the scheme did not take into consideration the efficiency of key management. Therefore, the scheme has two faults, bulky mobile agent codes and excessive calculations for encryption/decryption of keys. In recent years, Chang and Lin [15] proposed a key management scheme with hierarchically-based structure to reform the inefficient key management in Volker and Mehrdad's scheme. Although Chang and Lin corrected the faults in Volker and Mehrdad's scheme, their proposal remained inefficient as the scheme required the use of RSA exponential operations for key generation and derivation. Therefore, accessing confidential files consumes considerable system resources. To raise efficiency so as to make its application more convenient, we hope to be able to reduce its key derivation computation as well as the needed memory.

In 1985, Miller [16] and Koblitz [17] each proposed an elliptic curve algorithm-based crypto-technology for designing public key algorithm. This facilitated the development of many international standards for the Elliptic Curve Cryptosystem (ECC), such as ISO 11770-3, ANSI X9.62, IEEE P1363, FIPS 186-2, etc. The greatest feature of the ECC is that on the same level of security its key-size is comparatively smaller than that of the currently widely used RSA cryptosystem. For instance, the ECC whose key-size is 160 bits is as secure as the RSA with a key-size of 1024 bits. The smaller key-size of the ECC allows lower memory requirement and greater execution speed.

This study explores an approach to key management and access control for mobile agents, in which the framework is established in hierarchical environment. Besides, the proposed method will be based on the ECC whose small key-size and lower computations shall benefit the scheme by allowing the agents to execute the assigned tasks in optimal efficiency under reliable security measures.

After the introduction to the characteristics and the security risks of mobile agents, Section 2 examines the key management and access control method for mobile agents by Volker and Mehrdad with emphasis on performance efficiency. Section 3 shows the mathematic background of the elliptic curve cryptosystem, and put forth a new method to solve the key management problems of mobile agents. Section 4 discusses potential attacks that could damage the system, and interprets the measures of the proposed scheme against these security risks. Section 5 analyzes the performance efficiency from two points, the required space complexity for key storage and computation complexity for key derivation. Section 6 furnishes the conclusions.

2. Overview of Volker and Mehrdad's scheme

For access control and key management of mobile agents, Volker and Mehrdad [14] proposed a security method under tree-based structure, as illustrated in Fig. 1. The method is devoted to realizing agent authorization, dealing with key management, and controlling access of agents.

According to the data patterns — either static or mutable, the mobile agent framework is separated into two branches. The static branch stores constant data that will not change during the agent's lifetime, such as class codes, security policies, and so on. As to the mutable branch, the contents are alterable data like the return results, the instances of the classes, and the confidential contexts. Confidential information, whether static or mutable, always remains private. Also, after an agent achieves the target on a host, the host can alter the state of the agent and the information carried by the agent on the spot. Access control methods are competent so as to protect the secret resources from being accessed by unauthorized personnel; there are numerous similar related studies. In this section, the key management and access control method [14] presented by Volker and Mehrdad is given as an introduction to the subject, as follows.

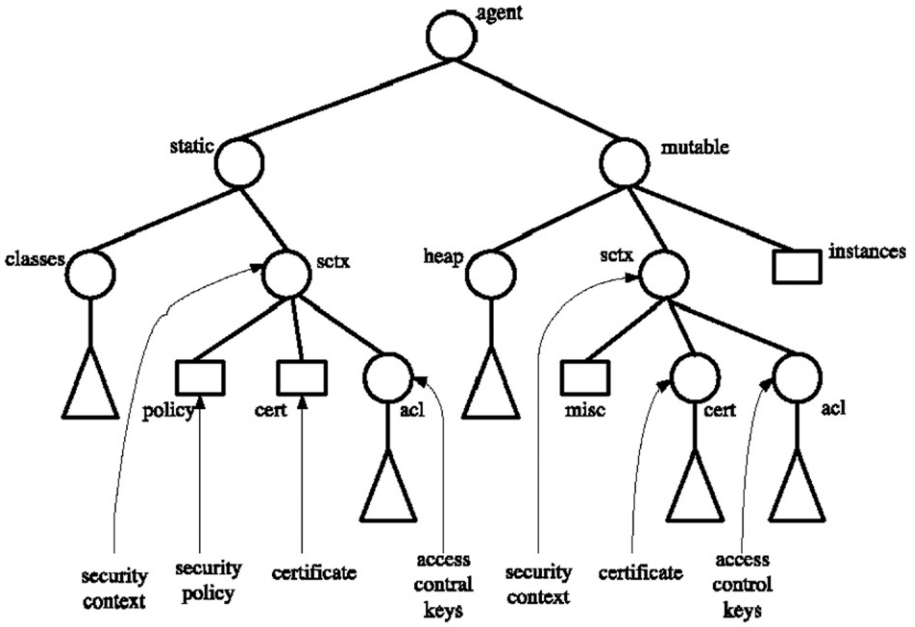

Fig. 1. Tree-based mobile agent framework.

The method by Volker and Mehrdad can be applied to public key cryptosystems, or symmetric encryption systems. Access control can be attained in the following manner. For the static branch, a folder is created for each visited host within the *static/sctx/acl* folder. If agent owner wants to send mobile agent onto the Internet to carry out assigned tasks like information retrieval or Internet transaction, the agent owner must first set the traveling route and access policies before sending the agent. The agent owner encrypts the confidential file with its matching secret key using a symmetric cryptosystem like the AES. Then, he assigns a secret key to each host according to the access policies, and stores these secret keys under the respective folders of each host. He also uses the public key of each host to encrypt their respective folders. When the mobile agent visits the hosts and communicates and interacts with them, the host can use its private key to decrypt its folder. From the folder, the host retrieves its secret key to decrypt the confidential file that is within its rights.

Fig. 2 illustrates the approach to solving problems of key management and access control for the method. In the diagram, there are five files in the folder of classes, including *agent.zip*, *support.zip*, *retrievel.zip*, *control.zip*, and *manage.zip*. Except for *agent.zip*, the rest of the four files remain in encryption, each of which corresponds to the decryption keys, DK_1, DK_2, DK_3, and DK_4, respectively. Moreover, there are six hosts, including *thu.edu.tw*, *ntu.edu.tw*, *pu.edu.tw*, *nchu.edu.tw*, *fcu.edu.tw*, and *ncyu.edu.tw*, called S_1, S_2, S_3, S_4, S_5, and S_6 for brevity.

Suppose a host is authorized to access a certain file. So, we should be able to find the required decryption key in the host's folder. For instance, S_1 has the decryption keys, DK_1, DK_2, DK_3, and DK_4, for decrypting the files: *support.zip*, *retrievel.zip*, *control.zip*, and *manage.zip*; S_2 has DK_1, DK_2, and DK_3 for decrypting the files: *support.zip*, *retrievel.zip*, and *control.zip*. All six folders, *static/sctx/acl/*S_1, *static/sctx/acl/*S_2, *static/sctx/acl/*S_3, *static/sctx/acl/*S_4, *static/sctx/acl/*S_5, and *static/sctx/acl/*S_6, must be encrypted, and only the corresponding host can

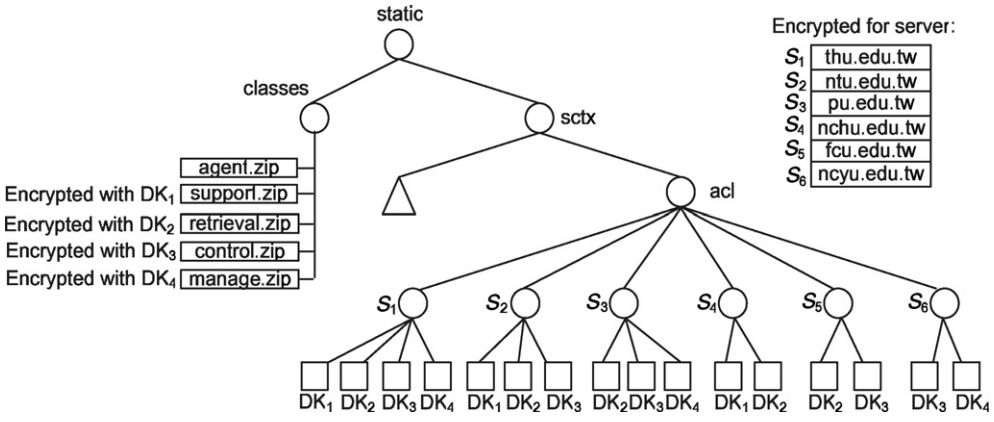

Fig. 2. Volker and Mehrdad's key management and access control method.

decrypt the respective folders. Such a technology is designed to protect the key management and access control method from being damaged.

The observations of the tree-based mobile agent structure are discussed in the following.

One is that the tree-based structure wastes a lot of memory storing the key, that is, the decryption key is repetitively stored under different folders. In the example shown in Fig. 2, DK_1 can be found in *static/sctx/acl/*S_1, *static/sctx/acl/*S_2, and *static/sctx/acl/*S_4. Similar consumption of memory occurs in archiving DK_2, DK_3, DK_4, DK_5, and DK_6. Clearly, the tree-based structure enlarges the size of a mobile agent.

The other is that the method costs a large computation amount for public key computation. Since the decryption key is restored in various hosts' folders *static/sctx/acl/*S_i, more computations for public key encryption are needed to maintain the confidentiality of the folder.

Minimizing storage space and public key calculations are in support of the target to make a mobile agent migrate freely in the Internet. For practical application, this study presents an efficient key management and access control method based on hierarchically-based mobile agent structure.

3. Overview of the proposed approach

3.1. Mathematic background of elliptic curve cryptography

The elliptic curves used in cryptography are of two types and defined as: the prime curves over Z_p for software applications and the binary curves over $GF(2^n)$ for hardware applications. In the finite field Z_p, an elliptic curve is represented as E: $y^2=x^3+ax+b$, where $(a,b)\in Z_p$ and $4a^3+27b^2\neq 0 \pmod p$. The elliptic curve indicates the integer points set that contains all points over the elliptic curve and a point of infinity O. The point of infinity O is the third point of intersection of any straight line with the curve, so that there are points including (x,y), $(x,-y)$, and O on the straight line. The necessity of $4a^3+27b^2\neq 0 \pmod p$ is to guarantee that the curve $y^2=x^3+ax+b \pmod p$ will not cause repeated factors. The set $E_p(a,b)$ defines a finite Abelian group, then the calculation in the finite Abelian group can be precisely executed because the occurrence of round off error in cryptographic application is disallowed.

The set of elliptic curve points forms a commutative finite group under the rules of addition operation, and it must satisfy the rules below:

1. $O+P=P$ and $P+O=P$, where O serves as the additive identity.
2. $-O=O$.
3. $P+(-P)=(-P)+P=O$, where $-P$ is the negative point of P.
4. $(P+Q)+R=P+(Q+R)$.
5. $P+Q=Q+P$.

For any two points $P=(x_P,y_P)$ and $Q=(x_Q,y_Q)$ over $Ep(a,b)$, the elliptic curve addition operation, which is denoted as $P+Q=R=(x_R,y_R)$, satisfies the following rules.

$$\begin{bmatrix} x_R=\lambda_2+x_P+x_Q \\ y_R=\lambda(x_P-x_R)-y_P \end{bmatrix}, \text{ where } \begin{cases} \lambda=\dfrac{y_Q-y_p}{x_Q-x_P} & \text{if} \quad P\neq Q \\ \lambda=\dfrac{3x_p^2+a}{2y_P} & \text{if} \quad P=Q \end{cases}$$

The elliptic curve multiplication operation by an integer over $Ep(a,b)$, which is represented as $Q=kP$, can be defined as repeated elliptic curve addition operations.

In a cryptosystem, security is constituted from resolving computationally infeasible mathematic problem in computer science. For instance, ECC-based security depends on the difficulty of solving the elliptic curve discrete logarithm problem (ECDLP), DLP-based security depends on the difficulty of solving the discrete logarithm problem (DLP), and RSA-based security depends on the difficulty of solving the integer factorization problem. Compared with the discrete logarithm cryptosystems, the addition operation in the ECC can be regarded as the counterpart of modular multiplication operation, and the multiplication operation in the ECC as that of

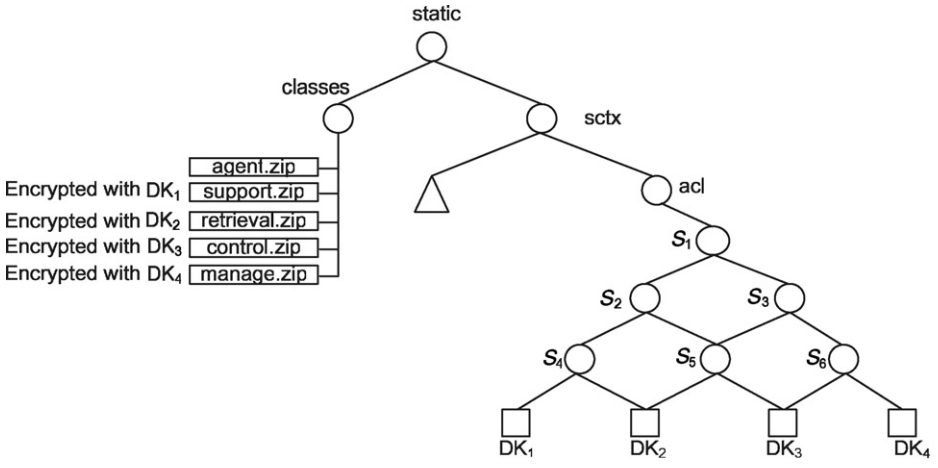

Fig. 3. Hierarchically-based framework for mobile agents to migrate.

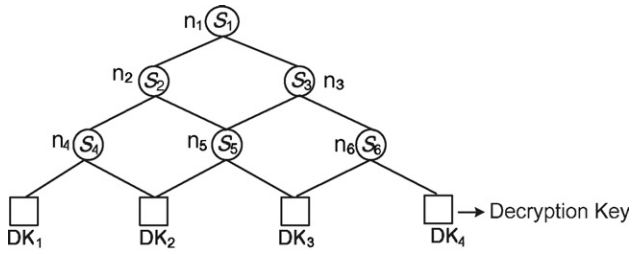

Fig. 4. Accessible network of decryption keys for mobile agents.

modular exponentiation operation. Classic cryptosystems, such as RSA and DSA, must employ a 1024-bit modulus to accomplish computationally infeasible security. In its place, the 160-bit-modulus ECC can achieve equal security. The security gap significantly increases as the modulus increases, for instance, the security gap between the 2048-bit-modulus RSA and DSA cryptosystems and 300-bit-modulus ECC is far larger.

3.2. Proposed hierarchically-based mobile agent framework

The proposed method adopts a hierarchical structure as shown in Fig. 3. Such a framework allows the decryption key to be exclusively located. Moreover, it applies elliptic curve cryptography which economizes agent size and reduces key-size.

In the hierarchy, the leaf node indicates the decryption key DK_j, where DK_j is used to decrypt the confidential files; the internal nodes represent the hosts S_i. Also, a mobile agent can form an accessible network to obtain the decryption key located at the leaf node. From the example shown in Fig. 4, the accessible network allows the root node S_1 to have authority over all decryption keys and S_2 to have authority over DK_1, DK_2, and DK_3.

If an agent owner wants to send a mobile agent onto the Internet to carry out assigned tasks, before he sends the agent, he must first decide which hosts are to be visited by the agent and what information can be accessed by the visited host. Next, he encrypts the confidential file using the key, DK_j. A symmetric cryptosystem like the AES can be used for the encryption. Then, he constructs the accessible network according to the access policies, and assigns a heterogeneous superkey n_i to each internal node (host). Each superkey is encrypted using public key cryptosystem, that is, the public key of the host is used to encrypt the superkey. When the agent visits the host and communicates and interacts with it, the host can use its private key to decrypt the superkey. Then, the superkey is used to derive the decryption key for decrypting the confidential file. A mobile agent constructs the accessible network as follows.

Step 1.1: Determine an elliptic curve equation $E_p(a,b)$, represented as $y^2=x^3+ax+b \pmod p$, where a and b satisfy $4a^3+27b^2\neq 0 \pmod p$ and p is a large prime number.
Step 1.2: Select a base point $G=(x,y)$ within the order n from $E_p(a,b)$, where n should be large enough such that $nG=O$.
Step 2.1: Select a one way hash function $f(x,y)$, by which a point can be transformed into a real number v.
Step 2.2: Declare $f(x,y)$ publicly.
Step 3.1: Randomly select a secret superkey n_i for each internal node.
Step 3.2: Generate the public parameters P_i and P_j, corresponding to n_i and n_j, respectively, as follows.

$$P_i = n_i G$$
$$P_j = n_j G$$

Step 4: Assign each leaf node an exclusive decryption key DK_j.
Step 5.1: Construct the polynomial $H_{DK_j}(x)$ for each leaf node; it should involve the public parameters of all hosts that have authority over the decryption key DK_j.

$$H_{DK_j}(x) = \prod_{S_j \le S_i} \left[x - f\left(n_j P_i\right)\right] + DK_j$$

Step 5.2: Declare $H_{DK_j}(x)$ publicly.
The relationship $S_j \le S_i$ means that a mobile agent can derive the decryption key DK_j through the host S_i using the superkey n_i and the public parameter P_j of S_j, as follows.

$$H_{DK_j}\left(f\left(n_i P_j\right)\right) = \prod_{S_j \le S_i} \left[f\left(n_i P_j\right) - f\left(n_i P_j\right)\right] + DK_j$$

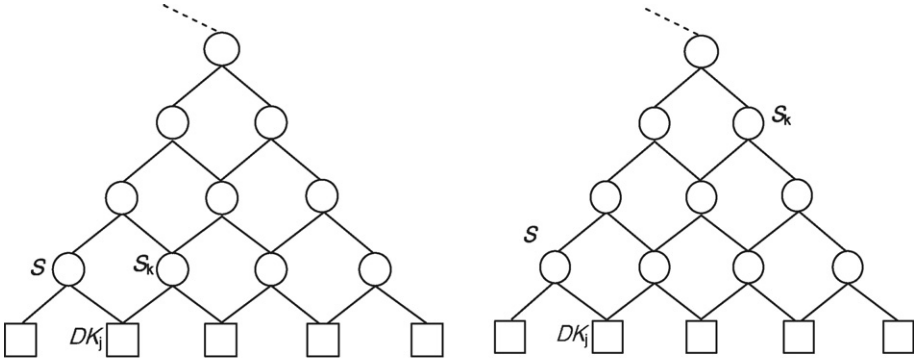

Fig. 5. Relationship risking equation breaking attack.

Table 1
Comparisons of Chang and Lin's Scheme with the proposed scheme

	Required storage space	Required computation cost	
		Time complexity	Rough estimation
Chang and Lin's scheme	$1024(r+m)$ bits	$2rT_{PK}+\left(m+r+\sum_{i=1}^{r}S_i\right)T_{EXP}$	$2rT_{PK}+240\left(m+r+\sum_{i=1}^{r}S_i\right)T_{MUL}$
The proposed scheme	$480r$ bits	$2rT_{PK}+\left(r+2\times\left(\sum_{i=1}^{r}S_i\right)\right)T_{EC\_MUL}$	$2rT_{PK}+29\left(r+2\times\left(\sum_{i=1}^{r}S_i\right)\right)T_{MUL}$

From the example in Fig. 3, $H_{DK_1}(x)$, $H_{DK_2}(x)$, $H_{DK_3}(x)$, and $H_{DK_4}(x)$ are constructed as follows.

$$\begin{aligned}
H_{DK_1}(x) &= [x-f(n_{DK_1}P_1)][x-f(n_{DK_1}P_2)][x-f(n_{DK_1}P_4)] \\
&\quad +DK_1 \\
H_{DK_2}(x) &= [x-f(n_{DK_2}P_1)][x-f(n_{DK_2}P_2)][x-f(n_{DK_2}P_3)] \\
&\quad \times[x-f(n_{DK_2}P_4)][x-f(n_{DK_2}P_5)]+DK_2 \\
H_{DK_3}(x) &= [x-f(n_{DK_3}P_1)][x-f(n_{DK_3}P_2))][x-f(n_{DK_3}P_3)] \\
&\quad \times[x-f(n_{DK_3}P_5)][x-f(n_{DK_3}P_6)]+DK_3 \\
H_{DK_4}(x) &= [x-f(n_{DK_4}P_1)][x-f(n_{DK_4}P_3)][x-f(n_{DK_4}P_6)] \\
&\quad +DK_4
\end{aligned}$$

To decrypt DK_4, S_1 takes n_1 and P_{DK_4} into the public polynomial $H_{DK_4}(x)$, as derived below.

$$\begin{aligned}
&H_{DK_4}(f(n_1P_{DK_4})) \\
&= [f(n_1P_{DK_4})-f(n_{DK_4}P_1)][f(n_1P_{DK_4})-f(n_{DK_4}P_3)] \\
&\quad \times[f(n_1P_{DK_4})-f(n_{DK_4}P_6)]+DK_4 \\
&= [f(n_1P_{DK_4}G)-f(n_{DK_4}n_1G)][f(n_1P_{DK_4})-f(n_{DK_4}P_3)] \\
&\quad \times[f(n_1P_{DK_4})-f(n_{DK_4}P_6)]+DK_4 \\
&= 0\times[f(n_1P_{DK_4})-f(n_{DK_4}P_3)][f(n_1P_{DK_4})-f(n_{DK_4}P_6)] \\
&\quad +DK_4 = DK_4
\end{aligned}$$

4. Discussion of security risks and relevant measures

Potential security risks and corresponding solutions are demonstrated in this section. The result shows that the proposed method is secure enough for practical applications.

4.1. Reverse attack

When two hosts S_i and S_m form a relationship such as $S_m \le S_i$, or when S_i and S_m are located on the same level in a hierarchy, an attempt by S_m to forcefully hack the superkey of S_i is called a reverse attack. In disjoint-entity hierarchy, any two hosts are independent from each other. When S_m, who does not have authority over S_i, tries a reverse attack on S_i, it must challenge $P_i=n_iG$ where the difficulty depends on the ability to solve the ECDLP. When information is sparse, a subordinate host cannot in reverse determine the superkey of the preceding host.

4.2. Conspiracy attack

This risk results when internal members cooperate to hack another host. In directed accessible network, it should benoted that an upper-level host might suffer a conspiracy attack from the lower-level subordinate hosts. Take for example the hosts in Fig. 3. Suppose S_4 and S_5 reach an agreement to conspire against S_1. Although it is possible to hack the desired superkey using known information, this attack shall fail because the problem is a kind of ECDLP.

Alternatively, another conspiracy attack arises when upper-level hosts conspire against a lower-level host who is not subordinate to them, such as the case where S_2 and S_4 conspire to hack S_6. This attack shall fail because the structure of leaf nodes accessible to lower-level users does not include the information of the upper-level users.

4.3. External collective attack

The external attack is caused by an intruder who tries to forcefully determine the superkeys of the internal hosts through the public parameters. With ECC-based structure, the intruder cannot possibly succeed in breaking it.

4.4. Equation breaking attack

The potential risk, equation breaking attack, is that in which a host tries to hack the secret parameters of the other hosts via their commonly shared decryption key. They can do so by cracking the public equation $H_{DK_j}(x)$. To be specific, supposing that both S_i and S_k have authority over DK_j, as illustrated in Fig. 5; the risk is that S_i will be able to guess the superkey n_k of S_k by cracking $H_{DK_j}(x)$.

From the example in Fig. 3, consider the possibility that S_2 hacks n_3 of S_3 through $H_{DK_3}(x)$, then S_2 might transform $H_{DK_3}(x)$ as follows.

$$\begin{aligned}
H_{DK_3}(x) &= [x-f(n_{DK_3}P_1)][x-f(n_{DK_3}P_2)][x-f(n_{DK_3}P_3)] \\
&\quad \times[x-f(n_{DK_3}P_5)][x-f(n_{DK3}P_6)]+DK_3 \\
H_{DK_3}(x)-DK_3(x) &= [x-f(n_{DK_3}P_1)][x-f(n_{DK_3}P_2)] \\
&\quad \times[x-f(n_{DK_3}P_3)][x-f(n_{DK_3}P_5)] \\
&\quad \times[x-f(n_{DK_3}P_6)]
\end{aligned}$$

$$[x-f(n_{DK_3}P_2)] = \frac{H_{DK_3}(x)-DK_3}{[x-f(n_{DK_3}P_1)][x-f(n_{DK_3}P_3)][x-f(n_{DK_3}P_5)][x-f(n_{DK_3}P_6)]}$$

When $x=f(n_2P_{DK_3})$, then the result of $f(n_2P_{DK_3})-f(n_{DK_3}P_2)$ can be derived as follows.

$$f(n_2P_{DK_3})-f(n_{DK_3}P_2) = \frac{H_{DK_3}(x)-DK_3}{[x-f(n_{DK_3}P_1)][x-f(n_{DK_3}P_3)][x-f(n_{DK_3}P_5)][x-f(n_{DK_3}P_6)]}$$

In this case, all S_2 can legally obtain is n_2 and DK_3 and no other. Without sufficient information, even though $H_{DK_3}(x)$ is eventually cracked by S_2 through the shared DK_3, S_2 shall only obtain: (1) n_{DK_3} through which S_2 can obtain DK_3, but because DK_3 is inherently accessible by S_2, so it does not matter; (2) P_3, but since P_3 is inherently a public parameter, S_2 cannot determine n_3, so it does not matter. Therefore, hacking n_3 fails because n_3 is not included in the structure of $H_{DK_3}(x)$.

5. Analysis of performance efficiency

According to Volker and Mehrdad's access control scheme, under *static/sctx/acl*, each host's folder must store and manage several decryptable confidential file's decryption key. The number of decryption keys depends on the amount of files that is accessible by the host. But Volker and Mehrdad's scheme neglected the problem of efficient key management. Because there are a lot of common files among the confidential files accessible to the visited hosts, a lot of space is wasted repeatedly storing decryption key causing the size of the mobile agent to be too large. This causes the mobile agent to consume excessive bandwidth while moving between the various hosts. Also, because the decryption key is repeatedly stored under the *static/sctx/acl* folder, agent owner must use more public key encryption operations to protect the confidentiality of the related folders. And the computation amount needed by the visited hosts to decrypt the decryption key is also increased. This could easily cause inefficiency problems when the host executes the mobile agent codes.

In recent years, Chang and Lin [15] proposed a key management scheme with hierarchically-based structure to reform the inefficient key management in Volker and Mehrdad's tree-based structure. In studies performed by Chang and Lin, it was proved that, be it the storage amount of decryption key or the calculations needed to derive the decryption key, their key management method far excels that of Volker and Mehrdad's. Therefore, we shall directly compare our scheme to Chang and Lin's scheme in the efficiency analysis.

Suppose the mobile agent will be visiting r hosts, VH_1, VH_2,..., VH_r, and carrying m confidential files, F_1, F_2,..., F_m. The number of confidential files accessible to the visited hosts is S_1, S_2,..., S_r. So, Chang and Lin's scheme need to store r superkeys and m public parameters. Because the scheme is based on the RSA cryptosystem, the length of the superkeys and public parameters must be 1024 bits. Therefore, the total storage space is 1024 $(r+m)$ bits. Our proposed scheme needs to store r superkeys and r public parameters, among which the length of the superkeys and the x-axis and y-axis of the elliptic curvepoint P_i is 160 bits. Hence, the total storage space is $480r$ bits.

Besides, in calculation, suppose T_{PK} denotes the time needed to encrypt or decrypt a public key, T_{EXP} denotes the time needed to execute a modulus exponentiation operation, T_{MUL} denotes the time needed to execute a modulus multiplication operation, and T_{EC_MUL} denotes the time needed to execute an elliptic curve multiplication operation. According to the original design of Volker and Mehrdad's structure, under the *static/sctx/acl* folder, the respective folder of each host is kept confidential. Therefore, each folder must be encrypted with the public key of the respective host. However, in public key cryptosystem, the length of the plaintext must be shorter than that of the public key. So, the superkeys must be encrypted/decrypted. Both Chang and Lin's scheme and our scheme require a key computation amount of $2rT_{PK}$. In Chang and Lin's scheme, agent owner needs to spend mT_{EXP} to generate m decryption keys and rT_{EXP} to calculate r superkeys; all hosts require $\sum_{i=1}^{r} S_i\ T_{EXP}$ to derive the decryption keys. The total computation amount is $2rT_{PK}+(m+r+\sum_{i=1}^{r} S_i)\ T_{EXP}$. In our proposed scheme, agent owner spends rT_{EC_MUL} to generate r public parameters and $\sum_{i=1}^{r} S_i\ T_{EC\_MUL}$ to construct m polynomials; all hosts spend $\sum_{i=1}^{r} S_i\ T_{EC\_MUL}$ to derive the decryption keys. The total computation amount is $2rT_{PK}+(r+2\times(\sum_{i=1}^{r} S_i))\ T_{EC\_MUL}$.

In Chang and Lin's scheme, the modulus exponentiation operation consumes the most time. In ours, it is the elliptic curve multiplication operation. These two operations require far greater time complexity as compared to other operations like multiplication, addition, hash function, etc. Therefore, the performance analysis performed above is based only on the modulus exponentiation operation and elliptic curve multiplication operation required by the two schemes. Also, consulting the studies of Koblitz et al. [18], in the operation of $g^k \bmod p$, k is a random integer of 160 bits, p is a prime number of 1024 bits, and the elliptic curve multiplication operation is used to calculate sG, where $G \in E(Z_p)$ and $p \approx 2^{160}$, where s is a random number of 160 bits. Integrating the above, the operation relations are as follows:

$$T_{EXP} \approx 240 T_{MUL}$$
$$T_{EC\_MUL} \approx 29 T_{MUL}$$

The performance analysis between Chang and Lin's scheme and our scheme is illustrated by Table 1. From the result of the comparison, we can see that be it in terms of storage space or computation amount, the performance of our scheme far excels that of Chang and Lin's.

6. Conclusions

In advanced networking research, issues on mobile agents have always been popular interest. Effectively utilizing resources over the Internet greatly enhances the efficiency of an organization and economizes computational overhead. With regard to practical concept, mobile agents can improve the efficiency of the software in hardwares. Mobile agents are currently challenged by execution barriers from security problems. Providing a cryptosystem with workable secure-control methods is essential for access activities in the Internet.

To resist potential security risks and enhance performance efficiency in application, this study presents for mobile agents an efficient key management and access control method based on the elliptic curve cryptosystem. Additionally, the procedures of key generation and operation are very simple; users with greater accessibility can directly access the decryption key of the subordinate members, but the latter is not allowed to access the decryption key of the former. The method not only successfully ensures the security of a mobile agent, but also economizes the exhaust of storage space. The achievement facilitates efficient operation of mobile agents, and provides a secure execution environment for mobile computing.

References

[1] P.P. Gian, Mobile agents: an introduction, Microprocessors and Microsystems 25 (2) (2001) 65–74.
[2] M.H. Shao, J. Zhou, Protecting mobile-agent data collection against blocking attacks, Computer Standards & Interfaces 28 (5) (2006) 600–611.
[3] J.W. Baek, H.Y. Yeom, A timed mobile agent planning approach for distributed information retrieval in dynamic network environments, Information Sciences 176 (22) (2006) 3347–3378.
[4] T. Lu, C. Hsu, Mobile agents for information retrieval in hybrid simulation environment, Journal of Network and Computer Applications 30 (1) (2007) 244–264.
[5] D.H. Shih, S.Y. Huang, D.C. Yen, A new reverse auction agent system for M-commerce using mobile agents, Computer Standards & Interfaces 27 (4) (2005) 383–395.
[6] W. Liu, J. Yang, L. Wei, A secure threshold proxy signature scheme for mobile agent-based electronic commerce transactions, Proceedings of the 7th International Conference on Parallel and Distributed Computing, Applications and Technologies, 2006, pp. 454–459.
[7] O. Esparza, J.L. Muñoz, M. Soriano, J. Forné, Secure brokerage mechanisms for mobile electronic commerce, Computer Communications 29 (12) (2006) 2308–2321.
[8] B. Arunachalan, J. Light, I. Watson, Mobile agent based messaging mechanism for emergency medical data transmission over cellular networks, Proceedings of the 2nd International Conference on Communication Systems Software and Middleware, 2007, pp. 1–6.
[9] B. Orgun, J. Vu, HL7 ontology and mobile agents for interoperability in heterogeneous medical information systems, Computers in Biology and Medicine 36 (7–8) (2006) 817–836.
[10] A. Koliousis, J. Sventek, A trustworthy mobile agent infrastructure for network management, Proceedings of the 10th IFIP/IEEE International Symposium on Integrated Network Management, 2007, pp. 383–390.
[11] A. Karmouch, Mobile software agents for telecommunications, IEEE Communications Magazine 36 (7) (1998) 24–25.
[12] A. Corradi, R. Montanari, C. Stefanelli, Security issues in mobile agent technology, Proceedings of the 7th IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS '99), IEEE Computer Society Press, Cape Town, South Africa, 1999, pp. 3–8.
[13] N.M. Karnik, A.R. Tripathi, A security architecture for mobile agents in Ajanta, Proceedings of the International Conference on Distributed Computing Systems, Taipei, Taiwan, 2000, pp. 402–409.
[14] R. Volker, J.S. Mehrdad, Access control and key management for mobile agents, Computers & Graphics (Pergamon) 22 (4) (1998) 457–461.
[15] C.C. Chang, I.C. Lin, A new solution for assigning cryptographic keys to control access in mobile agent environments, Wireless Communications and Mobile Computing 6 (1) (2006) 137–146.
[16] V.S. Miller, Use of elliptic curves in cryptography, advances in cryptology: proceedings of Crypto '85, Lecture Notes in Computer Science, vol. 218, 1986, pp. 417–426.
[17] N. Koblitz, Elliptic curve cryptosystems, Mathematics of Computation 48 (177) (1987) 203–209.
[18] N. Koblitz, A. Menezes, S.A. Vanstone, The state of elliptic curve cryptography, Designs, Codes and Cryptography 19 (2–3) (2000) 173–193.

Kuo Hsuan Huang received the B.S. and the M.S. degrees from Dayeh University in 2001 and 2003 respectively, both in Computer Science, Taiwan. He is currently a Ph.D. candidate in Computer Science of the Electrical Engineering Department in National Taiwan University, and doing research, i.e., information security, cryptography, and medical security.

Yu Fang Chung received the B.A. degree in English Language, Literature and Linguistics from Providence University in 1994, the M.S. degree from Dayeh University in 2003, and the Ph.D. degree from National Taiwan University in 2007, both in Computer Science, Taiwan. She is currently an Assistant Professor of Information Management at Chaoyang University of Technology, and doing research, i.e., information security and cryptography.

Chia Hui Liu received the B.S. degree from Dayeh University in 2002 and the M.S. degree from Chia Yi University in 2004, both in Computer Science, Taiwan. She is currently a Ph.D. student in Computer Science of the Electrical Engineering Department in National Taiwan University, and doing research, i.e., mobile communications, information security, cryptography, and network security.

Feipei Lai received a B.S.E.E. degree from National Taiwan University in 1980, and M.S. and Ph.D. degrees in computer science from the University of Illinois at Urbana-Champaign in 1984 and 1987, respectively.
He is a Professor in the Graduate Institute of Biomedical Electronics and Bioinformatics, the Department of Computer Science & Information Engineering and the Department of Electrical Engineering at National Taiwan University. He is a Vice Superintendent of National Taiwan University Hospital. He is the Chairman of Taiwan Network Information Center. He was a Visiting Professor in the Department of Computer Science and Engineering at the University of Minnesota, Minneapolis, USA. He was also a Guest Professor at the University of Dortmund, Germany and a Visiting Senior Computer System Engineer in the Center for Supercomputing Research and Development at the University of Illinois at Urbana-Champaign. Dr. Lai holds 6 Taiwan patents and 3 USA patents currently. His current research interests are SOC low power computing and medical information system.
Prof. Lai is one of the founders of the Institute of Information & Computing Machinery. He is also a member of Phi Kappa Phi, Phi Tau Phi, ACM, Chinese Institute of Engineers. Dr. Lai is the chairman of Taiwan Internet Content Rating Foundation. He received the Taiwan Fuji Xerox Research award in 1991. Dr. Lai is a senior member of IEEE and included in "Who's Who in Science and Engineering" and "Who's Who in the World."

Tzer Shyong Chen received the B.S. degree from Tunghai University in 1989, the M.S. degree from National Chiao Tung University in 1991, and the Ph.D. degree from National Taiwan University in 1996, all in Computer Science, Taiwan. He is currently a Professor of Information Management at Tunghai University, and doing research, i.e., information security, cryptography, and network security.