Efficient Date-constraint Access Control and Key Management Scheme for Mobile Agents

*Yu-Fang Chung, Ming-Hsien Kao, Tzer-Long Chen, and Tzer-Shyong Chen

***Abstract*—A mobile agent has become a competent software nowadays because of its significant abilities. It can roam freely in different Internet environments, adapt itself to various types of online servers totally associated to the characteristics of said server, and detect its environment and automatic adaptation to the environment while executing the assigned tasks by users. Due to these competencies, mobile agents have been widely used in Internet business, wireless communications, information security technologies and so forth. In 1983, Akl and Taylor suggested the concept of super-key to resolve the key management issues. It's the kind of thing where a mobile agent faces. Thus in 1998, Volker and Mehrdad proposed a tree-based mobile agent model to deal with the access control problem. The proposed scheme is based upon Elliptic Curve Cryptosystem (ECC), which enhances the operational ability of a mobile agent because of shorter key length and higher efficiency on encryption and decryption processes. For a mobile agent, a said user will be forced to log out of the system due to error or change in the user's rights or after a period of time. As is the case, the system must keep modifying the original access rights of the key so as to prevent a user from continuing to use his old key. However, this action could cause unnecessary error and risks, in addition to the large amount of computations that the system needs to perform. Hence, this paper proposes a Date-constraint key management scheme, where a date is attached to a key, so as to give its validity period of the key. Thus, key management can be more efficient.**

***Keywords*—Access Control, Key Management, Mobile Agent, Elliptic Curve Cryptosystem, Date-constraint Key Management**

I. Introduction

The popular and widespread use of personal computers over the Internet has become a part of our daily life. As is the case, ensuring secure transmission of data over open network environments to prevent malicious eavesdropping of the data is naturally an important matter of discussion. In this paper, we integrate all of the mobile agents, which can autonomously detect their environments and adapt themselves to the environments, with a hierarchical structure. Obviously, the ability of a mobile agent to adapt itself to its environment makes it suitable for the current Internet environment.

Yu-Fang Chung is currently an associate professor in the department of Electrical Engineering at Tunghai University. (e-mail: yfchung@thu.edu.tw)

Ming-Hsien Kao is a M.S. student in the Electrical Engineering Department of Tunghai University.

Tzer-Long Chen is a Ph.D student in the Information Management Department of National Taiwan University.

Tzer-Shyong Chen is currently a professor in the department of Information Management at Tunghai University.

And furthermore, under a hierarchical structure each user holds a different encryption key based on his/her access right so as to ensure the security of the structure. Presently, many businesses and government organizations are using the hierarchical structure along with suitable cryptography or related technology to ensure data security. The hierarchical structure used in our work can be integrated with those presently used by businesses and government organizations. In addition, our system sets different rights for different users based on their different transaction needs. The encryption key can ensure that the data will be securely transmitted upon requests. However, when the users under the hierarchical structure decide to log off from the system or switch the authority level for access. The system needs to cancel the right of the previously assigned key so as to avoid the key being illegally used to access the data. But, this could cause the system to spend more on computation to keep the key updated.

To resolve the above-mentioned problem, this paper will propose a Date-Constraint key management scheme on a hierarchical structure of mobile agents to make the encryption key of a user being valid for use only within its date constraint. In other words, users cannot access data if the keys are no longer valid. Because of this characteristic, the system does not need to keep updating keys. The proposed method aims to enhance the key management of a hierarchical mobile-agent structure and to make it more efficient.

There are many hidden and potential risks in accessing data; hence, cryptography technologies are needed for protecting data during transmission. This paper applies Elliptic Curve Cryptosystem (ECC) to generate the users' keys. The greatest feature of ECC is its key-size, which is comparatively smaller than that of the currently used RSA cryptosystem on the same level of security. For instance, the key-size of ECC said 160 bits has the equivalent security level to RSA with a key-size of 1024 bits. The feature allows ECC with the advantages of lower memory requirements and greater execution speeds. In addition, the related researches on mobile-agent security are also extremely important. In our paper, we will discuss how to protect the system from malicious attacks issued by unauthorized users.

When the mobile agents execute their tasks on an open network, they will communicate or exchange information with each other [3, 4]. Below are four examples of security threats which a mobile agent may face [5]:

(1) Unauthorized accesses to a server by a partner.
(2) Attacks on a server by malicious agents.
(3) Attacks on an agent by the other agents.
(4) Attacks on an agent by a malicious server.

In order to prevent the above-stated security threats, this

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

paper applies elliptic curve cryptography to enhance the security of the mobile agents and also to keep it away from malicious attacks.

The rest of the paper is organized as follows. Section 2 introduces the concept of the mobile agents and the advantages it can provide; following, we shall explain the mathematical background of elliptic curve cryptosystem and time-bound key management. Section 3 explains our proposed scheme. Section 4 contains the security analysis of our proposed scheme. Finally, the conclusions are furnished in Section 5.

II. Related Research

2.1 Mobile Agent

A mobile agent is a software program that can roam freely in the Internet environments from a local host to other remote hosts and can execute tasks assigned by its user. It reflects the actual situation during the migration and sends the result back to its user. The migration not only involves transferring of agent codes and results, it also includes the status-transferring of the program. It is therefore suitable to be used in the distributed and wireless network environments.

In addition, the mobile agent is capable of analyzing a situation, determining a solution, and executing tasks all by itself even if the connection with the target server is not constant. What has been described above indicates that the agent program is designed to be self-managing, self-controlling, and self-resolving, because it can control its behaviors and status and will not be interfered by other systems or users.

Today mobile agents are not only applied in distributed computations or data searches at open environments, it is also used in the networking management or the workflow system. Advantages of a mobile agent are shown as follows [6]:

(1) Reducing network loads: The distributed system relies heavily on the transmission medium for message exchange, especially when a security protocol is used. However, frequent exchange of messages can cause heavy network traffic. A mobile agent does not require a constant connection with the target server. Its user can encapsulate the instructions before sending it to the target server. The mobile agent initiates interaction and communicates with the target server only on its arrival at the server, thus network loads would be reduced due to lessened communication between mobile agents and servers over the Internet.

(2) Decreasing network delay time: Network delay often occurs when a large number of systems require immediate responses via the Internet. Using a mobile agent, these tasks can be transferred to a remote server for execution. As a result, there is no need to maintain a constant connection between the source and the target server. Also, the number of times to do the connections is lessened, and thus network delay time is obviously reduced.

(3) Packaging protocol: When using traditional distributed systems, a fixed protocol is needed for data exchange. However, on different operating systems, each server must set up its own protocol. This is inefficient, and at the same time also proffers security problems. If one of the servers was not quick enough with its update, incompatibility issues or delays can easily occur. When mobile agent is migrating to a remote server, its communication protocol can be packaged for migration, and the same can be re-established later for a network connection. This resolves the connection issue, which could be caused by the data migration.

(4) Adapting to a dynamic environment: Mobile agents are capable of moving freely around different network environments. Therefore, depending on the capability of a visited host server, the mobile agent must adjust its own capability to it. Thus, we say that the mobile agent is able to detect the surrounding environment and automatically adapt to the environment.

(5) Asynchronous and autonomous execution method: In a network environment, a constant connection is often not maintained. Often, due to natural calamity or unknown reason, it may not be possible to connect to the desired server. Thus when the cost of internet connection is rising and it is not suitable to have a constant connection, the mobile agent can be a useful tool since it completes tasks and returns the result in an asynchronous manner. This can reduce the Internet connection cost and achieve non-simultaneous and spontaneous executions.

(6) Innate heterogeneity: Network computing can be of diverse nature in terms of both hardware and software. The mobile agent is independent from the computer and the network transmission layer. It is like a computer is installed in another computer and they are communicating on the same machine. In this way, it does not matter what the characteristics of the original computer are, it can still connect, interact and exchange information with each other. The mobile agent can relate to any environment. Therefore, a mobile agent is suitable for system integration.

(7) Stability and fault tolerance: Mobile agent can also be used to manage exceptions and this gives it good stability and higher level of fault tolerance.

(8) Expandability: Mobile agent structure allows for great expandability and flexibility in adjustments between the source and the target server.

2.2 Elliptic Curve Cryptosystem

In 1985, Elliptic Curve Cryptography (ECC) was proposed by Neal Koblitz [7] and Victor Miller [8] independently. ECC can improve the existing cryptogram systems in terms of having smaller system parameters, smaller public-key certificates, lower bandwidth usage, faster implementations, lower power requirements, and smaller hardware processor requirements [9]. Therefore, using ECC to build a cryptosystem is commendable due to its high security and its efficiency [10]. The mathematic settings of ECC are described in the following [10, 11].

First, elliptic curves basically can be divided into two families: prime curves and binary curves. Prime curves (Z_p)

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

are good to use in software applications, because it does not require extended bit-fiddling operations, which binary curves require. Binary curves ($GF(2^n)$) are best for hardware applications as they require a few logic gates to build a powerful cryptosystem. Second, the variables and coefficients of elliptic curves are limited to the elements of a finite field. Because of these limitations, ECC would increase the efficiency while doing the computational operations.

In a finite field Z_p, defined modulo prime p, an elliptic curve is represented to be $E_p(a,b): y^2 = x^3 + ax + b \pmod p$, where $(a,b) \in Z_p$, and $4a^3 + 27b^2 \bmod p \neq 0$. The condition, $4a^3 + 27b^2 \bmod p \neq 0$, is necessary for ensuring that $y^2 = x^3 + ax + b \pmod p$ has no repeated factors, which means that the set $E_p(a,b)$ can define a finite abelian group [12]. Included in the definition of an elliptic curve, a point at infinity denoted as O is called the zero point. The point O is the third point of the intersection of any straight line with the curve, so that there are points including (x, y), $(x, -y)$, and O on the straight line.

For points on the set $E_p(a,b)$, we define a certain addition on them, denoted "+". The addition rules are given 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 $Q = (x_q, y_q)$ in the set $E_p(a,b)$, the elliptic curve addition operation denoted as $P + Q = R = (x_r, y_r)$ satisfies the following rules.

$$x_r = (\lambda^2 - x_p - x_q) \bmod p$$
$$y_r = (\lambda(x_p - x_r) - y_p) \bmod p$$

$$\text{where } \lambda = \begin{cases} \left(\frac{y_q - y_p}{x_q - x_p}\right) \bmod p, \text{if } P \neq Q \\ \left(\frac{3x_p^2 + a}{2y_p}\right) \bmod p, \text{if } P = Q \end{cases}$$

P and Q are two points on the elliptic curve cryptosystem, and n is a constant value, such that $P = n \times Q$. If n is very large, any one given the two points, P and Q still is unable to guess n because the problem he got to deal with is the Elliptic Curve Discrete Logarithm Problem (ECDLP) [13].

Example

Given an equation of the form denoted as $E_{23}(1,1): y^2 = x^3 + x + 1 \bmod 23$, $a = 1, b = 1 \in Z_p$, and $4a^3 + 27b^2 = 31 \bmod 23 \neq 0$, points over the elliptic curve $E_{23}(1,1)$

Let $P = (0,1)$ and $Q = (1,7)$ in $E_{23}(1,1)$. When $P \neq Q$, we must derive λ before calculating $P + Q$, as follows:

$$\lambda = \left(\frac{7-1}{1-0}\right) \bmod 23 \equiv 6$$

So, when $\lambda = 6$, x_r and y_r can be derived as shown below:

$$x_r = (6^2 - 0 - 1) \bmod 23 \equiv 35 \bmod 23 \equiv 12$$
$$y_r = (6(0 - 12) - 1) \bmod 23 \equiv -73 \bmod 23 \equiv 19$$

Thus, $P + Q = R = (12, 19)$.

To calculate $2P$, $P = (0, 1)$, we must first derive λ as follows:

$$\lambda = \left(\frac{3 \times 0^2 + 1}{2 \times 1}\right) \bmod 23 \equiv \left(\frac{1}{2}\right) \bmod 23 \equiv 12$$

So, when $\lambda = 12$, x_r and y_r can be derived as shown below:

$$x_r = (12^2 - 0 - 0) \bmod 23 \equiv 144 \bmod 23 \equiv 6$$
$$y_r = (12(0 - 6) - 1) \bmod 23 \equiv -73 \bmod 23 \equiv 19$$

Thus, $P + P = 2P = (6, 19)$.

2.3 Overview of Volker and Mehrdad's Scheme

Fig. 1 illustrates Volker and Mehrdad's agent structure. There are two defects in Volker and Mehrdad's scheme, and they are explained as follows:

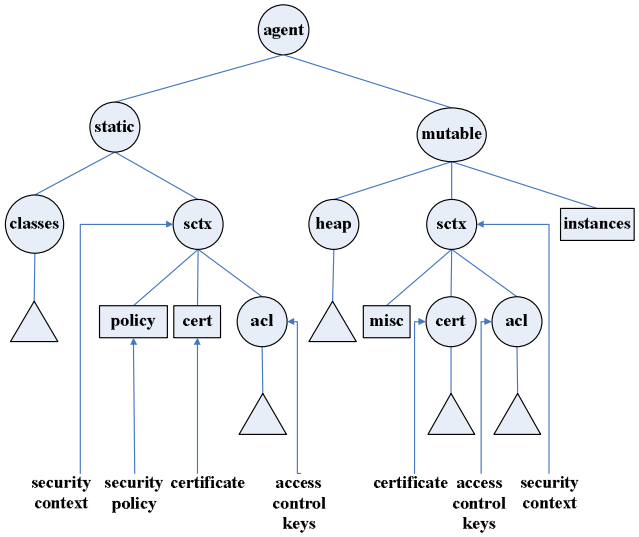

Fig. 1 Framework of mobile agent based on tree structure

(1) A large amount of storage space required inside the mobile agent for storing the generated keys: Based on this scheme, a decryption key can be duplicated in a lot of different servers. As shown in Fig. 2, servers S_1, S_2, and S_3 have the duplicated decryption key DK_2. Consequently, DK_2 would be continuously duplicated when a new server requires. This causes the wastage of storage space and makes the mobile agent cumbersome.

(2) Excessive computations on the public key generation: Because the decryption keys are repetitively stored under the folder of static/sctx/acl/, a mobile-agent user must use more resources and time to compute the public key encryption in order to ensure the security of the folder.

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

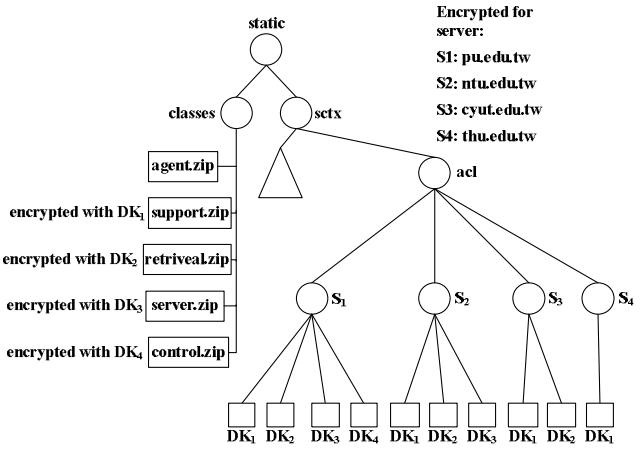

Fig. 2 An example of Volker and Mehrdad's access control and key management

Fig. 2 is used to illustrate Volker and Mehrdad's agent structure. This figure simply shows the process of how the static branch functions. On the left side of this figure, there are five zipped files: *agent.zip*, *support.zip*, *retrieval.zip*, *server.zip*, and *control.zip*. As the figure shows, *agent.zip* is not encrypted, but the rest of files are encrypted by DK_1, DK_2, DK_3, and DK_4 correspondingly. On the right side, the nodes, S_1, S_2, S_3 and S_4, represent different servers under access control keys. If these servers are authorized to access the specified files, the decryption keys will be copied to the corresponding server folders. For example, S_1 is granted access to all the encrypted files; thus, the keys (DK_1/DK_2/DK_3/DK_4) will be copied to the folder of S_1. Based upon the figure's decryption, S_2 is admitted to access the three encrypted files: *support.zip*, *retrieval.zip*, and *server.zip*; therefore, the keys (DK_1/DK_2/DK_3) will be duplicated to the folder of S_2. In addition, S_3 is only allowed to access the two encrypted files: *support.zip* and *retrieval.zip*; accordingly, only the keys, DK_1 and DK_2, are in the folder of S_3. Finally, S_4 is permitted to access the first encrypted zipped file, *support.zip*, and then it will just have DK_1 for accessing the designated file.

III. Research Method

Akl and Taylor presented an access control scheme [1] based upon a hierarchical structure model in 1983. In their scheme, each user is assigned to a specific security group called C_i, which is an element of the set $C = \{C_1, C_2, C_3, \ldots, C_m\}$. On the basis of a hierarchical structure, the access relationship between one security group and the other is denoted as $C_i \geq C_j$, which means that group C_i is at a higher level of the hierarchy than group C_j. This meaning can be described that all the users in group C_i have a greater authority able to access the information that is available to group C_j. However, when the hierarchy becomes larger, C_i would have to store a lot of decryption keys that are held by the groups at a lower level. And this would cause a security issue when managing the keys. The solution is to assign each user a key. Thus, Akl and Taylor came up with the concept of super-keys to solve this key management issue. Under the determined hierarchical relation of $C_i \geq C_j$, a user in C_i can use his super-key to obtain C_j's decryption key through mathematic operations.

Fig. 3 is an illustration of an improved version of Akl and Taylor's structure. In the following, we will explain how to access leaf nodes containing confidential files in a hierarchical structure. $file_j$ is the encrypted confidential file; C_i is the internal node, and it also represents a user; ski_i represents the secret key held by the user. When ski_i has permission to access some encrypted files, it can obtain confidential file from their corresponding leaf nodes. In Fig. 3, taking node C_2 as example, C_2 holds secret key sk_2, and it can access $file_1$, $file_2$, and $file_3$ based on its access right.

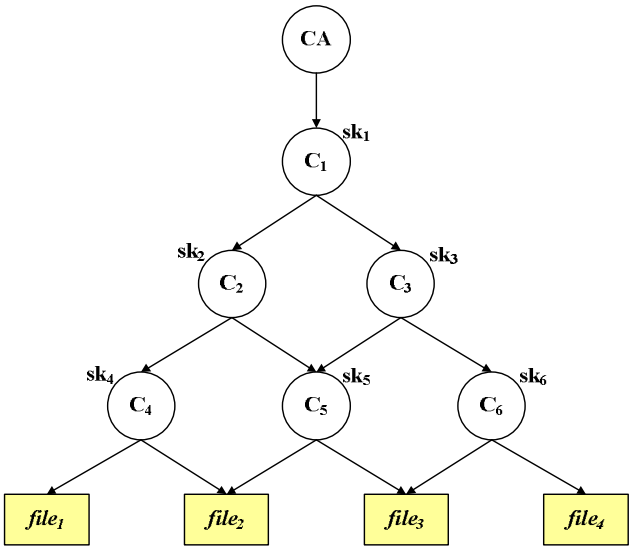

Fig. 3 Structure of decryption keys for mobile agent

Before the mobile agent is linked to the Internet to execute its assignments, mobile agent user must decide which host would be visited by the mobile agent and what kind of information can be accessed by the visited host. Afterward, he will construct an accessible network in relation to his access policy and gives a different secret key to every internal node. The secret keys are used to encrypt confidential files. A user can use his/her secret key to obtain the confidential file. In order to make access control on a hierarchical structure more effective, the concept of Date-constraint control will be introduced. The main point is to set up a scheme that will allow the user to use his/her existing decryption keys only at a predetermined time lot. In the incorrect time lot, the decryption key holder will be not able to use the key to access the file.

Mobile agent can construct the accessible network through the following steps.

3.1 Initialization phase

CA assigns a private key to each user, and performs the following steps:

Step 1: Define an elliptic curve group $E_p(a, b)$, where $y^2 = x^3 + ax + b \pmod{p}$. Note that a as well as b must

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

satisfy $4a^3 + 27b^2 \neq 0 \pmod p$, and p is a large prime number.

Step 2: Choose a base point $G = (x, y)$ on $E_p(a, b)$.

Step 3: Suppose two groups C_i and C_j with the relation $C_i \geq C_j$, and then the calculations for their public keys of C_i and C_j are shown below:

$$pk_i = sk_iG \bmod p$$
$$pk_j = sk_jG \bmod p$$

3.2 Key assignment phase

CA performs the following steps for group C_j.

Step 1: Randomly choose six random numbers, v1, v2, v3, v4, v5, and v6.

Step 2: Generate j= j1|| j2|| j3|| j4|| j5|| j6.
Here $j1=H^y(v1)$, $j2=H^{100-y}(v2)$, $j3=H^m(v3)$, $j4=H^{12-m}(v4)$, $j5=H^d(v5)$, and $j6=^{H31-d}(v6)$; y, m, and d, represent year, month, and date, respectively, of the expiry date, and "||" is the concatenation operator.

Step 3:Calculate $t= H^{100}(v1)|| H^{100}(v2)|| H^{12}(v3)|| H^{12}(v4)|| H^{31}(v5) || H^{31}(v6)$.

Step 4: CA chooses another secret random number k, and uses the previously calculated parameters t and j to calculate the signature of Date-bound warrant W =(t, j), and to generate the public parameters (r, s, R), and the private key x_j of user C_j. The calculations are as follows:

$$R = k \times G= (x_1, y_1) \text{ and}$$
$$r = x_1 \bmod p$$
$$s = k^{-1} \times (H(W)- sk_i \times r) \bmod p,$$

sk_i is the private key of C_i. $sk_j = H(k, ID_j)$, where ID_j is the public unique code of identity of C_j.

3.3 Key derivation phase

If $C_i \geq C_j$, then C_i can obtain C_j's private key in the following steps:

Step 1: Use public parameters (r, s, W) to calculate secret parameter k, as follows:
$k = ((s + sk_i \times H(W)) \times r \bmod p)^{-1}$,
where ski is the private key of C_i.

Step 2: Calculate C_j's private key $sk_j = H(k, ID_j)$.

3.4 Key expiration check phase

Step 1: Calculate $t'=H^{100-y}(v1)|| H^y(v2)|| H^{12-m}(v3)|| H^m(v4)|| H^{31-d}(v5)|| H^d(v6)$.

Step 2: If t' is not equal to t, then the key is already expired.

3.5 Key signature check phase

Use the following equations.

$R = k \times G= (x_1, y_1)$
$r = x_1 \bmod p$
$s = k^{-1} \times (H(W)-sk_i \times r) \bmod p$

And then calculate $V_1= r \times pk_i + s \times R$ and $V_2= H(W) \times G$. Judge whether V_1 is equal to V_2. If these two are equal, then the signature is true.

Proof:

$$V_1= r \times pk_i + s \times R$$
$$= r \times pk_i + (k^{-1} \times (H(W)- sk_i \times r) \times R$$
$$(\because s = k^{-1} \times (H(W)-sk_i \times r) \bmod p)$$
$$= r \times pk_i + (k^{-1} \times (H(W)-sk_i \times r) \times (k \times G) \quad (\because R = k \times G)$$
$$= r \times pk_i + (k^{-1} \times (H(W)- pk_i \times r) \times k$$
$$(\because pk_i = sk_iG \bmod p)$$
$$= r \times sk_iG + k^{-1} \times (H(W)- pk_i \times r \times k$$
$$= k^{-1} \times (H(W) \times R)$$
$$= k^{-1} \times (H(W) \times k \times G) \quad (\because R = k \times G)$$
$$= H(W) \times G$$
$$= V_2$$

IV. SECURITY ANALYSES

In this section, security analyses are performed to examine whether the proposed scheme is secure or not for practical applications. The analyses focus upon four types of attacks that may impact the system security.

4.1 Reverse Attacks

Reverse attacks can be defined as follows. When the relationship $C_j \leq C_i$ exists between the internal nodes C_j and C_i or they are at the same hierarchical level, host C_j tries to use its public key pk_j to derive C_i's public key pk_i and attempts to steal data accessible by C_i.

On a disjoint-entity hierarchy, any two internal nodes can be seen as independent units. For example, C_j does not have the right to access the data that is only available to C_i, because it is very difficult to obtain C_i's public key when CA is generating the function $pk_i = sk_iG \bmod p$ of the public key. Also, the security of key generation is based on the ECDLP problem, which is not easy to solve. Hence, C_j's reverse attacks would fail. Therefore, the users in a lower level can never derive the public key of the user in his upper level.

4.2 Collusion Attacks

A collusion attack occurs when the lower-level users work together to steal the information only accessible by the upper-level user in an accessible network. In the case of Fig. 3, users in C_5 and C_6 decide to work together against C_1 by using the obtained information to steal C_1's public key. However, this kind of attack would fail, because ECDLP would make the public key difficult to decode.

Another form of collusion attacks is a joint attack by upper-level users in an attempt to steal the data of the user on their same security level. However, the attack would not succeed, because the upper-level user's information is not embedded in the structure of leaf nodes, which only the users in the low-level leaf node can access.

4.3 External Collective Attacks

The intruder attempts to obtain the public key of an internal node to steal or modify the protected data of published system parameters. This is identified as an external collective attack. However, although the users in the hierarchy possess more information than external intruders, it is still difficult for them to derive a user's public key. Therefore, it would be even more difficult for an

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

external intruder who has even less data. Hence, it is impossible for the intruder to succeed, because ECC-based structure will prevent it from happening.

4.4 Date Alteration Attacks

Suppose group C_i attempts to extend the validity of the key by using the overdue encryption or continues to use the original encryption of the key that is in the date-bound warrant W=(t, j). In our proposed method,

$H^{100-y}(v1)||H^{y}(v2)||H^{12-m}(v3)||H^{m}(v4)||H^{31-d}(v5)\ ||H^{d}(v6)$
$=H^{100}(v1)||H^{100}(v2)||H^{12}(v3)||H^{12}(v4)||H^{31}(v5)\ ||H^{31}(v6)$
$=t$

Note that $j1=H^{y}(v1)$, $j2=H^{100-y}(v2)$, $j3=H^{m}(v3)$, $j4=H^{12-m}(v4)$, $j5=H^{d}(v5)$,and $j6=H^{31-d}(v6)$. Suppose a user tries to extend the key, to use the original encryption key continuously, or to alter system parameter j and public parameters j1, j2, j3, j4, j5, and j6. However, the expiry date of the key was originally defined by CA, and it is protected by the six secret random numbers, v1, v2, v3, v4, v5, v6, and one-way hash function. Suppose group C_i tries to extend the expiry year from y to y', and C_i can calculate $j1'=H^{y'}(v2)$ from $j1=H^{y}(v1)$. However, C_i cannot calculate $H^{100-y'}(v2)$. Therefore, no entity shall be able to alter the value of t, and the Date-bound warrant W cannot be changed at will. Hence, this type of attack can be avoided.

V. CONLCUSIONS

In today's Internet environment, mobile agent has a number of superior advantages. Furthermore, these advantages result from the mobile agent's efficient use of network resources, and it will thus be helpful in improving the efficiency of organizations by reducing various costs. Moreover, mobile agent plays an important role in e-commerce, and its contributive value in applications is expected to increase. However, security problems and threats on mobile agent remains a challenging subject. Thus, steps to minimize security problems, increase system operative speed, and reduce needed storage space, need to be taken for mobile agent. Therefore, a more complete mobile agent security system structure is desirable.

In this paper, a Date-Constraint key management scheme is proposed, in which each key is attached a date showing its validity of period. Once a key is expired, the key holder will no longer be able to use the key and further to access information. The proposed scheme makes key management system more efficient. With the help of elliptic curve cryptography (ECC), the proposed scheme can reduce the access space of keys and make key generation calculations lower. In addition, utilizing ECC to generate the system keys makes the mobile agent more secure. Based on the Elliptic Curve Discrete Logarithm Problem (ECDLP), ECC provides its system keys with more complexity and greater protection. Thus, the proposed scheme achieves the abilities of reducing key generation calculations and decreasing the system loads. In the security analyses, four different probable security attacks on the proposed scheme are analyzed in depth. The result shows that the proposed scheme is feasible when it is applied in Internet and is not easily susceptible to the attacks actuated by malicious users. Our proposed scheme is proven able to keep the transmitted data safe going through the mobile agent platforms.

ACKNOWLEDGEMENT

This work was supported partially by National Science Council of Republic of China under Grants NSC 98-2221-E-029 -025.

REFERENCES

[1] S.G. Akl and P.D. Taylor, "Cryptographic Solution to a Problem of Access Control in a Hierarchy," ACM Transactions on Computer Systems, Vol. 1, Issue 3, pp. 239– 248, 1983.

[2] R. Volker and J.S. Mehrdad, "Access Control and Key Management for Mobile Agents," Computer Graphics, Vol. 22, Issue 4, pp. 457–461, 1998.

[3] A. Karmouch, "Mobile Software Agents for Telecommunications," Guest Editorial, IEEE Communications Magazine, Vol. 36, No. 7, July, pp.24-25, 1998.

[4] I. C. Lin, H. H. Ou and M. S. Hwang, "Efficient Access Control and Key Management Schemes for Mobile Agents," Computer Standards & Interfaces, Vol. 26, No. 5, pp.423–433, 2004.

[5] F. Hohl, "A Model of Attacks Malicious Hosts Against Mobile Agents," Proceedings of the 4th Workshop on Mobile Object Systems: Secure Internet Mobile Computations, Brussels, Belgium, pp. 105–120, 1998.

[6] D.B. Lange and M. Oshima, "Programming and Deploying Java Mobile Agents with Aglets, Addison-Wesley Press," Massachusetts, USA, 1998.

[7] N. Koblitz, "Elliptic Curve Cryptosystems," Mathematics of Computation, Vol. 48, No. 177, pp. 203-209, 1987.

[8] V. S. Miller, "Use of Elliptic Curves in Cryptography," Advances in Cryptology: Proceedings of Crypto '85, Vol. 218, pp. 417-426, 1986.

[9] S. T. Wu, "Authentication and Group Secure Communications Using Elliptic Curve Cryptography," Doctoral Dissertation, National Taiwan University of Science and Technology, Taipei, 2005.

[10] Y. F. Chung, H. H. Lee, F. Lai and T. S. Chen, "Access Control in User Hierarchy Based on Elliptic Curve Cryptosystem," Information Sciences, Vol. 178, No. 1, pp. 230-243, 2008.

[11] C. W. Shieh, "An Efficient Design of Elliptic Curve Cryptography Processor," Master's Thesis, Tatung University, Taipei, 2006.

[12] K. H. Huang, Y. F. Chung, C. H. Liu, F. Lai and T. S. Chen, "Efficient Migration for Mobile Computing in Distributed Networks," Computer Standards & Interfaces, Available online 2007, in press.

[13] D. J. Guan and L. H. Jen, "Study and Implementation of Elliptic Curve Cryptosystem," Master's Thesis, National Sun Yat-Sen University of Technology, Kaohsiung, 2005.