Joint Conference of 1996 International Computer Symposium
December 19~21, Kaohsiung, Taiwan, R.O.C.

# Supervising Secure Communications for Level-Based Hierarchy

Tzong-Sun Wu and Tzong-Chen Wu

Department of Information Management,
National Taiwan Institute of Technology,
Taipei, Taiwan 106, Republic of China

*Abstract*

*Following the Bell-Lapadula access control model, it is only allowed that two users with the same security level in the hierarchy can communicate secretly with each other. Meanwhile, if it is necessary for security and law enforcement considerations, the contents of the secure communications between any two communicating users can be supervised (or eavesdropped) by the users with higher security level. Based on the usage of the tamper-proof hardware and one-way hash function, we proposed a novel scheme to resolve the supervising problem stated above. By the proposed scheme, a session key shared by two users can be quickly established for secure communications. Also, this session key can be easily derived by the users with higher security level, and hence the supervising requirements can be achieved.*

***Keywords:*** *supervising secure communications, level-based hierarchy, tamper-proof hardware, one-way hash function.*

## 1. Introduction

In 1994, U.S. government published the "Escrowed Encryption Standard" (EES) to increase the effectiveness of the anti-crime measures for the law enforcement agency, such as the Federal Bureau of Investigation (FBI), for supervising secure communications under certain legitimate authorization procedures [2, 3, 5]. Often supervising secure communications for the security clearance hierarchy is intrinsically required in order to enhance the management issues with respect to the authorized relationships among users in the organization.

In 1976, Bell and Lapadula [1] introduced the concept of simple security property and *-property for handling the access control problem in security clearance hierarchy. These two properties are described as follows:

1. Simple security property (no read up): A user with lower security clearance cannot read any information that belongs to the other user with higher security clearance.
2. *-property (no write down): A user with higher security clearance cannot write any information to the other user with lower security clearance.

Following the Bell-Lapadula access control model, it is only allowed that two users with the same security clearance can communicate with each other. This access control model protects the confidential information from being intentionally or carelessly leaked out from the user with higher security clearance to the user with lower one. For achieving secure communications, any two communicating users should first establish a session key, and then use this session key to encrypt/decrypt communicating messages. Any secure communication between two users, who should be with the same security clearance, can be supervised by the users with higher security clearance under certain legitimate authorization procedures.

Consider the situation that the security clearance hierarchy for the users is formed as a level-based hierarchy as shown in Figure 1. Users with the same security level are gathered into a group and entitled to communicate with each other. Let "$\succ$" be the partial order operator. For $U_i \succ U_j$, we say that the security level of $U_i$ is higher than that of $U_j$. If $U_i$ and $U_j$ are in the same security level, it is denoted as $U_i \approx U_j$. The users with higher security level have the authority to supervise the communications of users with lower one, while the opposite is not allowed. For example, in Figure 1, $U_1$ and $U_2$ can communicate secretly with each other, since $U_1 \approx U_2$. With the same reason, any two of $U_3$, $U_4$ and $U_5$ can communicate secretly with each other, and $U_1$ or $U_2$ can supervise their communications; any two of $U_6$, $U_7$, $U_8$ and $U_9$ can communicate secretly with each other, and their communications can be supervised by any one of $U_1$, $U_2$, $U_3$, $U_4$ and $U_5$.

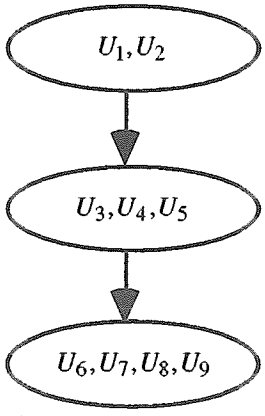


Figure 1. A level-based user hierarchy.

Inspired from Zheng's key agreement protocol [6], we propose a novel scheme to resolve the problem mentioned above. In the proposed scheme, each user is assigned a secret key and a public key. The keys used in the proposed scheme are represented in vector-form. The

secret key is stored in a tamper-proof chip. Except for the system manager (who is the initiator of the chip), no one knows the contents in the chip. The proposed scheme has the following characteristics:

1. Following the proposed key agreement protocol, any two users with the same security level can efficiently construct a session key shared by themselves, and further this session key can be efficiently derived by the users with higher security level.
2. The users with the same or lower security level cannot individually or conspiratorially obtain the session key shared by the communicating users.
3. The construction of the session key is rather efficient.
4. The amount of storage for the public parameters is small.

We sketch the organization of this paper as follows. In the next section, we give a brief review of Zheng's key agreement protocol. Then, the proposed scheme and the related definitions are described in Section 3. The security analysis and computational complexity of the proposed scheme are discussed in Section 4. Finally, we give conclusions in Section 5.

## 2. Brief Review of Zheng's Key Agreement Protocol

In 1995, Zheng [6] proposed an efficient key agreement protocol to construct a common key shared by two communicating users. Zheng's protocol is based on the well-known properties of one-way hash function. A one-way hash function $h$ has the following properties [4].

1. $h$ can be applied to any size of input.
2. $h$ produces a fixed-size output.
3. It is easy to compute $h(x)$ with knowing $x$.
4. Given $h(x)$, it is infeasible to determine $x$.
5. It is infeasible to find different $x$ and $y$ satisfy $h(x) = h(y)$.

For simplicity, we define $h^0(x) = x$ and $h^i(x) = h(h^{i-1}(x))$ for $i > 0$. Zheng's protocol is described in the following.

Initially, the system manager ($SM$) chooses an $m$-dimensional vector $X = [x_1, x_2, \ldots, x_m]$ as his secret key, where each $x_i$ is sufficient long, for instance more than 64 bits, to prevent it from exhaustive search attack. Note that $m$ is between $O(b^2 \log n)$ and $O(b^3 \log n)$, where $n$ is the total number of users and $b$ is the maximum number of dishonest users in the system. Meanwhile, $SM$ publishes a one-way hash function $h$ to all users in the system. Any user that wants to join the system should first register with $SM$. At the registration stage, $SM$ randomly selects an $m$-dimensional vector $V_i = [v_{i1}, v_{i2}, \ldots, v_{im}]$ as the public key for the registering user $U_i$, where $v_{ij}$'s are small integers. Afterwards, $SM$ personalizes a tamper-proof chip for $U_i$ simply by injecting $Y_i$, $V_i$ and $ID_i$ into the chip, where $ID_i$ is the identity for $U_i$ and $Y_i = [y_{i1}, y_{i2}, \ldots, y_{im}] = [h^{v_{i1}}(x_1), h^{v_{i2}}(x_2), \ldots, h^{v_{im}}(x_m)]$. It should be noticed that $Y_i$ is never seen by $U_i$.

After the registration stage, $U_i$ can obtain the common key shared with $U_j$ by presenting $ID_j$ and $V_j$ to his tamper-proof chip. The chip outputs the common key $k_{ij}$ as

$$k_{ij} = \begin{cases} h(h^{\delta_1}(y_{i1}) \| \ldots \| h^{\delta_m}(y_{im}) \| ID_i \| ID_j), & ID_i < ID_j \\ h(h^{\delta_1}(y_{i1}) \| \ldots \| h^{\delta_m}(y_{im}) \| ID_j \| ID_i), & ID_i > ID_j \end{cases} \quad (1)$$

where "||" denotes the concatenation operator, and $\delta_c$=0 if $v_{ic} > v_{jc}$ and $\delta_c = v_{jc} - v_{ic}$ otherwise (for $c$ = 1, 2, ..., $m$). In the similar way, $U_j$ can obtain the common key $k_{ji}$, which is identical to $k_{ij}$, shared with $U_i$ by presenting $ID_i$ and $V_i$ to his tamper-proof chip. Consequently, a secure channel between $U_i$ and $U_j$ is established by using the common key $k_{ij}$ (= $k_{ji}$).

## 3. The Proposed Scheme

In the proposed scheme, the secret keys and the public keys for users are represented as integer vectors. To improve the presentation of the proposed scheme, preliminaries with respect to integer vectors are first given. The proposed scheme is described in the subsequent subsection.

### 3.1. Preliminaries

Let $\mathbb{V} = \{V_1, V_2, \ldots, V_n\}$ be a set of $n$ $m$-dimensional integer vectors, where $V_i = [v_{i1}, v_{i2}, \ldots, v_{im}]$ for $i$ = 1, 2, ..., $n$.

**Definition 1.** (max vector) A vector $V_i \in \mathbb{V}$ is a *max vector* in $\mathbb{V}$ if and only if there exists some $c$ such that $v_{ic} > v_{jc}$ for all $V_j \in \mathbb{V}$ and $i \neq j$.

**Definition 2.** (max set) If every vector $V_i \in \mathbb{V}$ is a *max vector*, then $\mathbb{V}$ is a *max set*.

**Definition 3.** (derive) $V_i$ *derives* $V_j$, which is denoted as $V_i \Rightarrow V_j$, if and only if $v_{ic} \leq v_{jc}$ for all $c$.

It is to see that $V_j$ can be obtained by adding some nonnegative integer to each element of $V_i$ if $V_i \Rightarrow V_j$.

**Example 1.** Let $V_1$ = [3, 4, 7], $V_2$ = [5, 6, 9], $V_3$ = [4, 3, 6], $V_4$ = [8, 6, 3], and $\mathbb{V}$ = $\{V_1, V_2, V_3, V_4\}$. Then $V_2$ and $V_4$ are max vectors in $\mathbb{V}$. The set $\{V_2, V_4\}$ is a max set, however $\mathbb{V}$ is not a max set. Further, $V_1 \Rightarrow V_2$, since $V_2 = V_1 + [2, 2, 2]$.

**Definition 4.** (RMAX) The relative maximum elements of two vectors is defined as $\mathrm{RMAX}(V_i, V_j) = [\max(v_{i1}, v_{j1}), \max(v_{i2}, v_{j2}), \ldots, \max(v_{im}, v_{jm})]$.

**Definition 5.** (RMIN) The relative minimum elements of two vectors is defined as RMIN($V_i$, $V_j$) = [min($v_{i1}$, $v_{j1}$), min($v_{i1}$,$v_{j1}$), ..., min($v_{im}$, $v_{jm}$)].

**Theorem 1.** For a max set V, NOT($V_i \Rightarrow V_j$) if $\{V_i, V_j\} \subseteq V$.

Proof:

We use proof by contradiction. Assume that $V_i \Rightarrow V_j$, then $v_{ic} \le v_{jc}$ for all $c$ by definition. Since $V_i$ is a max vector, there exists some $d$ such that $v_{jd} < v_{id}$. That contradiction shows that NOT($V_i \Rightarrow V_j$). □

**Theorem 2.** For a max set V, NOT($V_e \Rightarrow$ RMAX($V_i$, $V_j$)) if $\{V_e, V_i, V_j\} \subseteq V$.

Proof:

Since $V_e$ is a max vector, there exists some $c$ such that $v_{ec} > \max(v_{ic}, v_{jc})$. Therefore, we have NOT($V_e \Rightarrow$RMAX($V_i$, $V_j$)) by definition. □

It should be noticed that the number of max vectors cannot outnumber the dimension of the vectors in a max set. A straightforward but effective algorithm for generating a max set is given in Figure 2.

**Algorithm MAX_SET_GENERATOR**

**Input:** LowerBound: the smallest value of the elements of vectors;
$m$: the dimension of vectors;
$n$: the number of generated vectors.

**Output:** a max set $\{V_1, V_2, ..., V_n\}$/* $V_i = [v_{i1}, v_{i2}, ..., v_{im}]$ */.

```
{
if (m < n) then STOP;
for i = 1 to n do
    for j = 1 to m do
        if j = i then v_{i,j} = LowerBound + 1;
        else v_{i,j} = LowerBound;
return ({V_1, V_2, ..., V_n});
}
```

Figure 2. The max-set assignment algorithm.

### 3.1. Supervising secure communications on hardware approach

We divide the proposed scheme into three stages: the initialization stage, the communication stage, and the supervising stage. Suppose there exists a system manager ($SM$) to handle the initial setup, and the users in the system are classified according to their security level. The keys used in the proposed scheme are represented in vector-form. Let $m$ be the dimension of vectors. The value of $m$ should be determined before the initialization stage and it should be larger or equal to the number of users in any security level. Assume that security level 1 is the highest security clearance. The proposed scheme is described in the following.

**The initialization stage** -- $SM$ performs the following steps to setup the system:

Step 1. Publish a one-way hash function $h$.

Step 2. Choose an $m$-dimensional vector $X = [x_1, x_2, ..., x_m]$ as the seed for $h$ and keep $X$ secretly.

Step 3. Assign the public keys and the secret keys for users, say $U_{i1}$, $U_{i2}$, ..., $U_{in}$, in security level $i$ by the following steps:

(3-1). Use Algorithm MAX_SET_GENERATOR($i$, $m$, $n$) to generate a max set $\{V_{i1}, V_{i2}, ..., V_{in}\}$, and $V_{ij} = [v_{ij1}, v_{ij2}, ..., v_{ijm}]$ is used as the public key for $U_{ij}$.

(3-2). Compute the secret key for $U_{ij}$ as

$$Y_{ij} = [y_{ij1}, y_{ij2}, ..., y_{ijm}] = [h^{v_{ij1}}(x_1), h^{v_{ij2}}(x_2), ..., h^{v_{ijm}}(x_m)], \quad (2)$$

inject $Y_{ij}$ into a tamper-proof chip, and send the chip to $U_{ij}$.

Step 4. Repeat Step 3 until all security levels in the hierarchy are examined.

Note that the public keys for users in the same security level forms a max set. Further, for any two users $U_i$ and $U_j$ in the hierarchy, $V_i \Rightarrow V_j$ if $U_i \succ U_j$.

**The communication stage** -- After the initialization stage, each user owns a tamper-proof chip. If $U_i$ wants to communicate with another user, say $U_j$, with the same security level, he can obtain the session key shared with $U_j$ by presenting $V_j$ and a public nonce $N_i$ to his tamper-proof chip. The chip outputs the session key $k_{ij}$ as

$$k_{ij} = h(h^{\delta_1}(y_{i1}) || ... || h^{\delta_m}(y_{im}) || N_i), \quad (3)$$

where $\delta_c = 0$ if $v_{ic} > v_{jc}$ and $\delta_c = v_{jc} - v_{ic}$ otherwise (for $c$ = 1, 2, ..., $m$). In the similar way, $U_j$ can obtain the session key $k_{ji}$, which is identical to $k_{ij}$, shared with $U_i$ by presenting $V_i$ and the public nonce $N_i$ to his tamper-proof chip. Consequently, a secure channel between $U_i$ and $U_j$ is established.

Note that in Eq. (3), the involvement of the public nonce $N_i$ is used to generate distinct session keys for different communication sessions. Further, the construction of $k_{ij}$ does not involve with $ID_i$ and $ID_j$ as that in Eq. (1). Since the public keys for users in the same security level form a max set, there exist no users $U_c$ and $U_d$ such that RMAX($V_c$, $V_d$) = RMAX($V_i$, $V_j$). That is, the input of $h$ will always be distinct for different communicating users. Hence, $ID_i$ and $ID_j$ are no longer required for the construction of session key in the proposed scheme.

**The supervising stage** -- Following the communication stage, suppose $U_e$ with higher security level wants to supervise the communications between $U_i$ and $U_j$. $U_e$ can obtain $k_{ij}$ by presenting RMAX($V_i$, $V_j$)

and $N_i$ to his tamper-proof chip. The chip outputs the session key $k_{ij}$ as

$$k_{ij} = h(h^{\rho_1}(y_{e1}) \| h^{\rho_2}(y_{e2}) \| \ldots \| h^{\rho_m}(y_{em}) \| N_i), \quad (4)$$

where $\rho_c = \max(v_{ac}, v_{bc}) - v_{sc}$ (for $c$ = 1, 2, ..., $m$).

**Lemma 1.** For any two users $U_i$ and $U_j$ with the same security level in the hierarchy, they can obtain the same session key $k_{ij}$ (= $k_{ji}$) in the communication stage.

Proof:

By Eqs. (2) and (3), we have

$$k_{ij} = h(h^{\sigma_1}(x_{i1}) \| \ldots \| h^{\sigma_m}(x_{im}) \| N_i), \quad (5)$$

where $\sigma_d = \max(v_{id}, v_{jd})$ (for $d$ = 1, 2, ..., $m$). It is trivial that $V_i \Rightarrow \text{RMAX}(V_i, V_j)$ and $V_j \Rightarrow \text{RMAX}(V_i, V_j)$. Hence, $U_i$ and $U_j$ can obtain the same session key $k_{ij}$ (= $k_{ji}$). □

**Lemma 2.** For any user $U_e$ in the hierarchy, if $(U_e \succ U_i)$ and $(U_e \succ U_j)$, then $U_e$ can obtain the same session key $k_{ij}$ (= $k_{ji}$) shared by $U_i$ and $U_j$ in the supervising stage.

Proof:

The readers can easily verify that the value of $k_{ij}$ in Eq. (4) is identical to that in Eq. (5). In Eq. (4), since $V_e \Rightarrow \text{RMAX}(V_i, V_j)$, we have $\sigma_c \geq 0$ (for $c$ = 1, 2, ..., $m$). Hence, $U_e$ can obtain the same session key $k_{ij}$ (= $k_{ji}$) shared by $U_i$ and $U_j$. □

**Theorem 3.** In the proposed scheme, the communication and the supervising stages work correctly.

Proof:

It can be proved by Lemma 1 and Lemma 2. □

**Example 2.** Consider the security level hierarchy in Figure 1. After the initialization stage, the public key $V_i$ and the secret key $Y_i$ for $U_i$ are listed in Table 1. Suppose $U_3$ wants to communicate secretly with $U_4$. $U_3$ presents $V_4$ = [2, 3, 2 ,2] and a public nonce $N_3$ to his tamper-proof chip, and then the chip outputs the session key $k_{34} = h(h^3(x_1) \| h^3(x_2) \| h^2(x_3) \| h^2(x_4) \| N_3)$. By the similar way, $U_4$ can obtain $k_{43}$ which is identical to $k_{34}$. After that, the secure communication channel is established. If $U_1$ wants to supervise such communication, he may present RMAX($V_3$, $V_4$) = [3, 3, 2 ,2] and $N_3$ to his tamper-proof chip, and obtain $k_{34}$. However, users with lower security level cannot derive $k_{34}$. For example, suppose $U_6$ presents RMAX($V_3$, $V_4$) and $N_3$ to his tamper-proof chip, the chip will output a value $h(h^4(x_1) \| h^3(x_2) \| h^3(x_3) \| h^3(x_4) \| N_3)$ which is not identical to $k_{34}$.

Table 1. The assignment of keys for users in the user hierarchy of Figure 1.

| $U_i$ | $V_i$ | $Y_i = [y_{i1}, y_{i2}, \ldots, y_{im}]$ |
|---|---|---|
| $U_1$ | [2, 1, 1, 1] | $[h^2(x_1), h^1(x_2), h^1(x_3), h^1(x_4)]$ |
| $U_2$ | [1, 2, 1, 1] | $[h^1(x_1), h^2(x_2), h^1(x_3), h^1(x_4)]$ |
| $U_3$ | [3, 2, 2, 2] | $[h^3(x_1), h^2(x_2), h^2(x_3), h^2(x_4)]$ |
| $U_4$ | [2, 3, 2 ,2] | $[h^2(x_1), h^3(x_2), h^2(x_3), h^2(x_4)]$ |
| $U_5$ | [2, 2, 3, 2] | $[h^2(x_1), h^2(x_2), h^3(x_3), h^2(x_4)]$ |
| $U_6$ | [4, 3, 3, 3] | $[h^4(x_1), h^3(x_2), h^3(x_3), h^3(x_4)]$ |
| $U_7$ | [3, 4, 3, 3] | $[h^3(x_1), h^4(x_2), h^3(x_3), h^3(x_4)]$ |
| $U_8$ | [3, 3, 4, 3] | $[h^3(x_1), h^3(x_2), h^4(x_3), h^3(x_4)]$ |
| $U_9$ | [3 ,3, 3, 4] | $[h^3(x_1), h^3(x_2), h^3(x_3), h^4(x_4)]$ |

## 4. Analyses

### 4.1. Crypto analysis

In this subsection, we will show that the proposed scheme is secure under the following two assumptions:

(1). The Tamper-Proof Chip (TPC) assumption: Except for *SM*, the tamper-proof chip can prevent anyone (even the legitimate user of the chip) from gaining access to the secret key contained therein.

(2). The One-Way Function (OWF) assumption: Consider the one-way hash function $h(x)$. Given $y$, it is computationally infeasible to find $x$ such that $h(x) = y$. That is, it is computationally infeasible to find $h^{i-1}(x)$ with given $h^i(x)$.

**Lemma 3.** Under the OWF assumption, anyone cannot reveal the secret key stored in the tamper-proof chip by using the information emitted from the chip.

Proof:

Given the emitted information $K = h(h^{v_{e1}}(y_{i1}) \| h^{v_{e2}}(y_{i2}) \| \ldots \| h^{v_{em}}(y_{im}) \| N)$ for some $V_e$ and $N$, it is computationally infeasible to find the secret key $Y_i = [y_{i1}, y_{i2}, \ldots, y_{im}]$ by the OWF assumption. □

**Lemma 4.** For users $U_e$, $U_i$ and $U_j$ in the hierarchy, if NOT($U_e \succ U_i$) and NOT($U_e \succ U_j$), then $U_e$ cannot obtain the session key $k_{ij}$ (= $k_{ji}$) share by $U_i$ and $U_j$ under the OWF assumption.

Proof:

Following the Bell-Lapadula access control model, the communicating users $U_i$ and $U_j$ should with the same security level in the hierarchy. Two cases will be considered.

Case 1. If $U_i \succ U_e$ and $U_j \succ U_e$, then NOT($V_e \Rightarrow$ RMAX($V_i$, $V_j$)).

Case 2. If $U_e \approx U_i$ and $U_e \approx U_j$, then NOT($V_e \Rightarrow$ RMAX($V_i$, $V_j$)).

In both of the cases, NOT($V_e \Rightarrow$RMAX($V_i$, $V_j$)) assures that $U_e$ cannot obtain $k_{ij}$ under the OWF assumption. □

**Theorem 4.** The conspiracy attack cannot succeed in deriving the session key of other users under the OWF and the TPC assumptions.
Proof:

If the security level of $U_a$ is higher than that of $U_b$, than $V_a \Rightarrow V_b$. Thus, it is no necessary to discuss the case that the conspiratorial users come from different security levels. Only two cases will be considered.

Case 1. The conspiratorial users work together to obtain the session key of users with higher security level: This attack cannot succeed under the OWF assumption.

Case 2. The conspiratorial users work together to obtain the session key of users with the same security level: The public keys for users in the same security level form a max set. By Theorem 2, for a max set { $V_e$, $V_i$, $V_j$ }, we have NOT($V_e \Rightarrow$RMAX($V_i$, $V_j$)). Thus, this attack cannot succeed under the TPC assumption. □

Note that if the key agreement protocol in the proposed scheme is implemented on software approach, then the possible conspiracy attack may succeed in deriving the secret key or session key of other users. We give an example to show such attack. The conspiratorial users, say $U_i$ and $U_j$, can work together to obtain a new secret key $[h^{\min(v_{i1},v_{j1})}(x_1), h^{\min(v_{i2},v_{j2})}(x_2), \ldots, h^{\min(v_{i3},v_{j3})}(x_m)]$ which contains lower order of $h$ as compared to their original secret keys. If there exists some other users, say $U_c$ and $U_d$ in the same security level as $U_i$ and $U_j$, satisfying RMIN($V_i$, $V_j$)$\Rightarrow$ RMAX($V_c$, $V_d$), then $U_i$ and $U_j$ may succeed in deriving the session key $k_{cd}$ shared by $U_c$ and $U_d$.

**Theorem 5.** Under the TPC and the OWF assumptions, the proposed scheme is secure.
Proof:

It is proved by Lemma 3, Lemma 4 and Theorem 4. □

### 4.2. Computational analysis and memory requirements

Let $T_h$ be the time for performing the one-way hash function $h$ and $T_A$ be the time for executing Algorithm MAX_SET_GENERATOR. Suppose $q$ is the number of security levels in the hierarchy, $n$ is the number of users in a security level, and $m$ is the dimension of vectors. The computational complexities of the proposed scheme are analyzed as below:

Time for the initialization stage

$= O(qT_A + nmq^2 \times T_h)$.

Time for the communication or supervising stage

$= O(mT_h)$.

For the memory requirements, the public directory stores an $m$-dimensional vector as the public key for each user. Since it is infeasible to derive $x$ with known $h(x)$, the public keys can be assigned with small integers without decreasing the system's security. Such assignment can reduce the storage of public keys as well as the time for performing one-way hash functions.

## 5. Conclusions

In this paper, we address a new problem that may exist in the network communications. Following the Bell-Lapadula access control model, it is only allowed that two users with the same security level can communicate with each other. Users with higher security level can supervise the communications of users with lower one. Based on the properties of one-way hash functions and the security of tamper-proof chips, we proposed a scheme to resolve the supervising secure communication problem for the level-based hierarchy. The security and the computational complexities of the proposed scheme are analyzed. With the small integers of public keys, we can reduce the operating time for performing one-way hash function and the storage space. To sum up, the proposed scheme has the following characteristics:

1. The construction of the session key is fast, since the computational operations of the proposed scheme are only comparison and one-way hash function.
2. The proposed scheme can withstand the conspiracy attack.
3. The storage space for the public directory is small since it only requires $m$ small integers for each user.
4. It is unnecessary for the *SM* to keep information for users after the initialization stage.

With the above analyses, we conclude that the proposed scheme can be implemented practically. However, the solution of the supervising secure communication for a general poset user hierarchy is left as an open problem.

## References

[1] Bell, D. E. and Lapadula, L. J., "Secure computer systems: Mathematical foundations and model", MITRE Report MTR 2547, November 1973.

[2] Desmedt Y., "Securing traceability of ciphertexts - Towards a secure key escrow system", *Advances in Cryptology - EUROCRYPT '95*, Springer-Verlag, Berlin, 1995, pp. 147-157.

[3] Lenstra, A. K., Winkler, P., and Yacobi, Y., "A key escrow system with warrant bounds", *Advances in Cryptology - CRYPTO '95*, Springer-Verlag, Berlin, 1995, pp. 197-207.

[4] Merkle, R. C., "One way hash function and DES", *Advances in Cryptology - CRYPTO '90*, Springer-Verlag, Berlin, 1990, pp. 428-446.

[5] National Institute of Standards and Technology, NIST FIPS PUB 185, "Escrowed Encryption Standard", U.S. Department of Commerce, February 1994.

[6] Zheng, Y., "On key agreement protocols based on tamper-proof hardware", *Information Processing Letters*, Vol. 53, No. 1, 1995, pp. 49-54.