http://dx.doi.org/10.3745/KIPSTC.2012.19C.4.225

Cocks' ID-based Scheme 기반 문턱 암호화 기술

Sergey V. Bezzateev[†] · 김 대 엽[††]

요 약

공개키 암호 시스템을 구현하기 위해서는 공개키 정보를 반드시 검증해야 된다. 이와 같은 단점을 극복하기 위하여 사용자 신원정보를 이용하여 공개키를 생성하는 기술들이 소개 되었다. 그러나 신원정보 기반 기술은 비밀키를 생성하는 별도의 생성자를 필요로 하기 때문에 이와 같은 생성자가 주요 공격 대상이 될 수 있다. 이러한 문제를 해결하기 위하여 문턱 암호기술을 접목시키는 기술들이 제안되었다. 본 논문에서는 Cocks가 제안한 신원정보 기반 암호 기술을 확장하여 별도의 생성자를 요구하지 않는 사용자 신원 정보기반의 문턱 암호 기술을 제안하고 제안된 기술이 chosen identity 공격에 안전함을 증명한다.

키워드 : 문턱 암호 시스템, 사용자 신원 정보 암호화 기술, Quadratic Residues

Threshold Encryption Scheme based on Cocks' IBE Scheme

Sergey V. Bezzateev[†] · Kim DaeYoub[††]

ABSTRACT

Since one of weak points of public crypto-systems is to require the verification of public key, identity based crypto-systems were proposed as an alternative. However, such techniques need a private key generator which can be a single point of failure. To improve such weakness, threshold identity-based crypto-systems were proposed. In this paper, we propose a new threshold identity-based encryption scheme which is constructed to extend an identity-based encryption scheme by Cocks. Since the proposed scheme is based on quadratic residues, it has smaller complexity of encryption. And we prove that the proposed scheme is secure against a chosen identity attack.

Keywords : Threshold Cryptography, Identity-based Encryption, Quadratic Residues

1. Introduction

The main feature of an Identity-Based Encryption (IBE) scheme is usage of an unique identifier of a node (an arbitrary string, e.g. an email address) as the public key [1]. Current implementations of IBE schemes are based on different mathematical techniques: [2], [3] use residues, [4], [5] and [6] use elliptic curves.

As opposite to the traditional public key crypto-systems, a sender node in IBE scheme does not need to obtain the public key and certificate of an receiver node from a trusted entity. The sender could generate an appropriate public key by himself from the publicly known identifier of the receiver node. Therefore, IBE scheme does not require a trusted certification authority in order to store/manage public keys and certificates for all nodes.

However, IBE scheme should requires an another trusted entity called as private key generator (PKG) which on demand generates private keys for nodes from their identifiers using a master secret key. PKG can be a single point of failure in IBE scheme. For example, if PKG is compromised, all earlier issued keys will be also compromised. And when PKG is unavailable, a new node could not join a domain/network, because the new node could not obtain its private key elsewhere.

To eliminate the aforementioned disadvantage of IBE scheme, a threshold identity-based encryption (TIBE) scheme was proposed. In TIBE scheme, each predefined size coalition of nodes could act as PKG. That is, assume that there are l nodes in a domain/network that can

[†] 비 회 원 : State University of Airspace Instrumentation(SUAI) 교수
[††] 종신회원 : 수원대학교 정보보호학과 전임강사

논문접수 : 2012년 3월 20일
수 정 일 : 1차 2012년 5월 22일
심사완료 : 2012년 6월 11일

* Corresponding Author : Kim DaeYoub(daeyoub69@suwon.ac.kr)

participate in an private key generation process. Then each coalition containing k $(k \leq l)$ such nodes could together perform PKG operations. The coalition could generate private keys for the new nodes, but none of the coalition members has complete information about the master secret key as well as about the generated private keys of the new nodes.

The main idea of TIBE is to distribute a master secret key between all l nodes using a secret sharing scheme. In general, a threshold key generation works as follows: A new node chooses coalition of k nodes and then sends them a request. Each coalition node individually makes a partial calculation using both its own share of the master secret key and the identifier of the new node, and then sends back the result (a private key share) to the new node. The new node combines all received shares for obtaining its own private key.

There were proposed several TIBE schemes ([7], [8]) based on the technique of the elliptic curves for IBE schemes of the same type ([4], [6]). For enhancing the performance of TIBE scheme, we propose a new TIBE scheme based on Cock's IBE scheme proposed in [2] which is based on the technique of the quadratic residues having smaller complexity of encryption then elliptic curve based scheme. This work is important because this is the first attempt to apply threshold scheme to a residual based IBE.

The paper is organized as follows: The following section briefly gives the definition of TIBE. Section 3 explains assumptions that will be used in a security proof. In Section 4, we propose a new TIBE and describe both the construction of the TIBE and its security analysis. The paper ends with some concluding remarks.

2. Definitions

A (k,l)-TIBE scheme consists of the following functions: *Setup, ShareKeyGen, ShareVerify, Combine, Encrypt, ValidateCT* and *Decrypt*. These functions are specified as:

1. Setup(l,k,σ): It takes as input the number of nodes l that could act as PKG members, a threshold k $(1 \leq k \leq l)$, and a security parameter $\sigma \in Z$. It gives as output a triple (PK, VK, SK), where PK is a public scheme parameters, VK is a verification key, and $SK = (SK_1, \cdots, SK_l)$ is a vector of master secret key shares. Each node i is given a master secret key share (i, SK_i).

2. ShareKeyGen$(PK, ID, (i, SK_i))$: It takes as input the public scheme parameters PK, an identifier ID, and a master secret key share (i, SK_i). It gives as output both a private key share H_i and a share verification key P_i for a node i.

3. ShareVerify(PK, VK, ID, H_i, P_i): It takes as input the public scheme parameters PK, the verification key VK, the identifier ID, a private key share H_i and its verification key P_i. It gives as output "valid" or "invalid".

4. Combine$(PK, ID, (H_{i_1}, \cdots, H_{i_k}))$: It takes as input the public scheme parameters PK, an identifier ID, and k private key shares $(H_{i_1}, \cdots, H_{i_k})$. It outputs a private key d_{ID} or a failure symbol $\perp$.

5. Encrypt(PK, ID, M): It takes as input the public scheme parameters PK, an identity ID and a message M. It outputs a ciphertext C.

6. ValidateCT(PK, ID, C): It takes as input the public scheme parameters PK, an identifier ID and a ciphertext C. It outputs "valid" or "invalid". If "valid", C is regarded as a valid result of Encrypt process under ID.

7. Decrypt(PK, ID, d_{ID}, C): It takes as input the public scheme parameters PK, an identity ID, a private key d_{ID} and a ciphertext C. It outputs a message M or a failure symbol $\perp$.

In these functions, ShareVerify and ValidateCT are not presented in our scheme: Since ShareVerfiry could be realized using the known techniques from [9], ShareVerify is not presented. Accordingly, Setup does not calculate the verification key VK and ShareKeyGen does not output the share verification key P_i; Because ValidateCT is required for a security proof in a chosen ciphertext attack model and the base of our TIBE (Cocks' scheme) is only secure in a chosen plaintext attack model [2], ValidateCT is not described.

3. TIBE security notion

In this section, we describe the security notion for TIBE scheme based on [7]. As mentioned in [7], the security of TIBE is defined by both a consistency of key generation and a security against chosen identity attack. In the chosen identity attack, an attacker tries to generate the private key for a node with ID. There are two models of the chosen identity attack: the first one is an adaptive-ID attack, when an adversary chooses a target identity adaptively; the second one is a selective-ID attack, when an adversary selects it in advance.

Consistency of key generation in our scheme follows from Shamir's secret sharing properties and expressions (9)-(11), so it seems obvious and is not considered in details. The proof of the consistency of key generation is not considered in our work, because *ValidateCT* is not provided in our scheme for the reasons described above.

The proposed scheme will be shown to be secure against the adaptive-ID attack in the Random Oracle Model. In our case, the security against the adaptive-ID attack is defined using the following game:

1. *Init* : The adversary outputs a set of $k-1$ PKG nodes $S \subset \{1, \cdots, l\}$ that it wishes to corrupt.

2. *Setup* : The challenger runs $\mathrm{Setup}(l, k, \sigma)$ to obtain a random instance of the scheme parameters (PK, SK), where $SK = (SK_1, \cdots, SK_l)$. It gives the adversary PK and all (j, SK_j) for $j \in S$.

3. *Query phase 1* : The adversary adaptively issues identity queries (ID, i), where $ID \in \{0,1\}^*$ and $i \in \{1, \cdots, l\}$. The challenger responds with the result of $ShareKeyGen(PK, ID, (i, SK_i))$.

4. *Challenge* : The adversary outputs two messages M_0 and M_1 of equal length and identifier ID^*, on which it wishes to be challenged. The challenger generates a random bit $b \in \{0,1\}$ and returns the encrypted message M_b.

5. *Query phase 2* : The adversary and the challenger interact as in *Query phase 1*.

6. *Guess* : The adversary has to return his guess of challenged bit b'. The advantage of adversary is

$$Adv_A^{TIBE}(l, k, \sigma) = \left| \frac{1}{2} - \Pr\{b = b'\} \right|. \qquad (1)$$

4. The Threshold IBE Scheme

4.1 Brief Review of Cocks' IBE

Cocks' IBE scheme consists of four algorithms: *Setup*, *Extract*, *Encrypt* and *Decrypt*.

1. *Setup* : It generates two large random prime numbers $p = 3 \mod 4$ and $q = 3 \mod 4$, and calculates $M = pq$. There should be defined a hash function $Hash: \{0,1\}^* \rightarrow Z_M$. According to the properties of p and q, $a = Hash(ID)$ holds $\left(\frac{a}{M}\right) = 1$, where $\left(\frac{a}{M}\right)$ is a Jacobi symbol. The public scheme parameters are defined as $PK = (M)$ and a master secret key consists of (p, q).

2. *Extract* : For a given node with an identifier ID, a public key is defined as $PK_{ID} = Hash(ID) = a$ and a private key is defined as $d_{ID} = r$, where r is a square root of a or $-a$ modulo M. Since nobody else knows p and q except PKG, only PKG can calculate the square root of a or $-a$ modulo M. For example, PKG can perform this in the following way:

$$r = a^{\frac{M+5-(p+q)}{8}} \mod M. \qquad (2)$$

As result, either $r^2 = a$ if a is the Quadratic Residue (QR) modulo M or $r^2 = -a$ if $-a$ is the QR modulo$M$.

3. *Encrypt* : To encrypt a message bit $x \in \{-1, 1\}$ for a receiver with the identifier ID, a sender first calculates the receiver's public key $PK_{ID} = Hash(ID) = a$ and generates two random values, t_1 and t_2, such that $\gcd(t_1, M) = 1$, $\gcd(t_2, M) = 1$ and $\left(\frac{t_1}{M}\right) = \left(\frac{t_2}{M}\right) = x$. Then the sender calculates

$$S_1 = t_1 + \frac{a}{t_1}, S_2 = t_2 - \frac{a}{t_2} \mod M. \qquad (3)$$

The ciphertext is a pair $C = (S_1, S_2)$.

4. *Decrypt* : The receiver decrypt $C = (S_1, S_2)$ using the secret key $d_{ID} = r$ as follows:

$$\hat{x} = \begin{cases} \left(\frac{S_1 + 2r}{M}\right), \mathrm{if}\ a\ \mathrm{is}\ QR \mod M \\ \left(\frac{S_2 + 2r}{M}\right), \mathrm{if}\ -a\ \mathrm{is}\ QR \mod M. \end{cases} \qquad (4)$$

4.2 Threshold Key for Cocks' IBE

In Cocks' IBE scheme, the calculations modulo $M = pq$ are used. When performing a threshold cryptography with such a compound modulo operation, a main problem is that if it is required to calculate $a^{xy^{-1}} \mod M$, then one should find

$y^{-1} \mod \phi(M)$, where $\phi(M)$ is Euler function of M. In [9], to overcome this obstacle, following two restrictions were added:

1. p and q are "safe" primes, i.e. $p = 2p' + 1$ and $q = 2q' + 1$, where p' and q' are also primes;

2. $\gcd(p', y) = \gcd(q', y) = 1$.

Fortunately, both these requirements conform with Cocks' scheme. So we also use ideas from [9] to securely share a master secret key in our TIBE as follows:

1. *Setup* : (A) It selects two "safe" primes $p = 2p' + 1$ and $q = 2q' + 1$, where p' and q' are also primes and $\frac{p'q'+1}{2}$ is odd. After that, it calculates both $M = pq$ and $m = p'q'$. It is easy to show that $\phi(M) = 4m$. The master secret key is calculated as

$$d = \frac{M + 5 - (p+q)}{8} = \frac{m+1}{2}. \qquad (5)$$

(B) The master secret key is represented as a sum

$$d = 4d_1 + d_2. \qquad (6)$$

Using Shamir's secret sharing scheme, both d_1 and d_2 are shared between l PKG nodes. For this purpose, two random polynomials $f(x)$ and $g(x)$ of degree $k-1$ are generated such that $f(0) = d_1$ and $g(0) = d_2$. All their coefficients are calculated by modulo m. Two shares are given to each PKG node i as follows:

$$SK_i^{(1)} = f(i)\Delta^{-1} \mod m \qquad (7)$$

and

$$SK_i^{(2)} = g(i)\Delta^{-1} \mod m, \qquad (8)$$

where $1 \le i \le l$ and $\Delta = l!$. l is a number of nodes. Therefore, $l < p'$, $l < q'$ and $\gcd(\Delta, m) = 1$.

(C) The public scheme parameters PK consist of M and e_2, where e_2 has the following property

$$d_2 e_2 \equiv 1 \mod \phi(M). \qquad (9)$$

2. *ShareKeyGen* : Upon receiving ID from a requesting node, each node of k coalition-nodes calculates two private key shares from its master key shares as follows: $a = Hash(ID)$, $H_i^{(1)} = a^{4SK_i^{(1)}} \mod M$, $H_i^{(2)} = a^{4SK_i^{(2)}} \mod M$, and outputs triple $\{i, H_i^{(1)}, H_i^{(2)}\}$.

3. *Combine* : Without loss of generality, lets consider that nodes of coalition S have numbers $1, \cdots, k$. Let $\lambda_{0,1}, \cdots, \lambda_{0,k}$ be the Lagrange coefficients for nodes in this coalition, and $\lambda'_{0.1} = \Delta\lambda_{0.1}, \cdots, \lambda'_{0.i} = \Delta\lambda_{0.i} \in Z_m$ be the modified Lagrange coefficients such that $f(0)\Delta = \sum_{i=1}^{k} \lambda'_{0,i} f(i)$. Then the private key could be calculated as follows:

$$\begin{aligned} d_{ID}^{(1)} &\equiv \prod_{i \in S} H_i^{(1)\lambda'_{0,i}} \equiv a^{4\Delta\Delta^{-1}\sum_{i \in S}\lambda_{0,i} f(i)} \\ &\equiv a^{4d_1} \mod M. \end{aligned} \qquad (10)$$

Since $\gcd(4, e_2) = 1$, using the public parameter e_2 and the extended Euclidean algorithm, the values of x and y satisfying $4x + e_2 y = 1$ could be always found. Using these x and y, it calculates:

$$\begin{aligned} d'^{(2)}_{ID} &\equiv \prod_{i \in S} H_i^{(2)\lambda'_{0,i}} \equiv a^{4\Delta\Delta^{-1}\sum_{i \in S}\lambda_{0,i} g(i)} \\ &\equiv a^{4d_2} \mod M, \\ d_{ID}^{(2)} &\equiv a^y d'^{(2)x}_{ID} \equiv a^{4d_2 x + y} \\ &\equiv a^{d_2} \mod M, \end{aligned} \qquad (11)$$

$$d_{ID} \equiv d_{ID}^{(1)} d_{ID}^{(2)} \equiv a^{4d_1 + d_2} \equiv a^d \mod M. \qquad (12)$$

4. *Encrypt* : To encrypt a message bit $x \in \{-1, 1\}$ for a receiver with the identifier ID, a sender calculates the receiver's public key $PK_{ID} = Hash(ID) = a$, generates two random values, t_1 and t_2, holding $\left(\frac{t_1}{M}\right) = \left(\frac{t_2}{M}\right) = x$. Then the sender finally calculates

$$\begin{aligned} S_1 &= t_1 + \frac{a}{t_1} \mod M, \\ S_2 &= t_2 - \frac{a}{t_2} \mod M. \end{aligned} \qquad (13)$$

The ciphertext is a pair $C = (S_1, S_2)$.

5. *Decrypt* : A receiver decrypt $C = (S_1, S_2)$ using the secret key $d_{ID} = r$ as follows:

$$\hat{x} = \begin{cases} \left(\dfrac{S_1 + 2r}{M}\right), \text{if } a \text{ is } QR \bmod M \\ \left(\dfrac{S_2 + 2r}{M}\right), \text{if } -a \text{ is } QR \bmod M. \end{cases} \tag{14}$$

4.3 Security Result

By definition from [7], adaptive-ID attack is stronger then selective-ID attack. So if the proposed scheme is shown as adaptive-ID secure, it is the more secure to selective-ID attack. To prove the semantic security of the proposed TIBE against an adaptive-ID attack in the Random Oracle Model, we assume that the adversary $\mathcal{A}$ has the advantage $Adv_{\mathcal{A}}^{TIBE}(k, l, \sigma) > \epsilon$ in attacking the proposed TIBE scheme for a given value of the security parameter σ. We now construct an algorithm $\mathcal{B}$ that attacks Cocks' IBE scheme with advantage ϵ using $\mathcal{A}$.

Assume that $\mathcal{B}$ has access to Cocks' encryption Oracle and knows only the public scheme parameter M. For using $\mathcal{A}$, $\mathcal{B}$ should simulate all interactions between $\mathcal{A}$ and TIBE Oracle. It means that the simulators for functions *Setup*, *Query phase 1* and *Random Oracle* (which simulates hash function $Hash(ID)$) should be developed for $\mathcal{B}$. These simulators and resulting game are as follows:

1. ***Initialization*** : $\mathcal{A}$ chooses a set S of $k-1$ PKG nodes that it wants to corrupt. Without lost of generality, let $S = \{1, 2, \cdots, k-1\} \subset \{1, \cdots, l\}$.

2. ***Setup simulator*** : $\mathcal{B}$ randomly generates an odd integer $e_2 \in Z_M$ and gives the public scheme parameters $PK = (M, e_2)$ to $\mathcal{A}$. For each of corrupted $k-1$ nodes, the share is generated randomly in following interval $(SK_i^{(1)}, SK_i^{(2)}) \in \{0, \cdots, \lfloor M/4 \rfloor - 1\}$. As in Shoup's scheme [9], statistical distance between uniform distribution on $\{0, \cdots, m-1\}$ and $\{0, \cdots, \lfloor M/4 \rfloor - 1\}$ is negligible. The obtained shares $SK' = (i, SK_i^{(1)}, SK_i^{(2)})$ for $1 < i \leq k-1$ are given to $\mathcal{A}$.

3. ***Random Oracle simulator*** : When $\mathcal{A}$ requests the random Oracle to calculate $a = Hash(ID)$, $\mathcal{B}$ captures the request and returns a fake value a' to $\mathcal{A}$, which is statistically indistinguishable from a real value. For that, the random Oracle simulator maintains a table, where it saves answers on previous requests (ID and a'). If ID is found in the table, then $\mathcal{B}$ returns corresponding a'. Otherwise $\mathcal{B}$ generates uniformly distributed random values $b \in \{0, 1\}$ and $r \in \{1, M-1\}$, and then calculates $a' = (-1)^b r^{2e_2\Delta^2} \bmod M$. The obtained pair (ID, a') is returned to $\mathcal{A}$ and saved in the table. Therefore, for the same ID, the simulator always returns the same a'. The values of a' have the following properties: they are uniformly distributed; their Jacobi Symbol is always equal to 1; $\Pr\{a' \text{ is } QR\} = \dfrac{1}{2}$.

4. ***Query phase 1 simulator*** : For a returned by the random Oracle simulator and any given $j \not\in \{1, \cdots, k-1\}$, this simulator generates a private key share $(H_j^{(1)}, H_j^{(2)})$. Let there is a polynomial $h(x)$ with coefficients defined as $h_i = 4f_i + g_i$. Then $h_0 = d$. The shares for $h(x)$ are calculated as follows:

$$\begin{aligned} SK_i^{(3)} &= h(i)\Delta^{-1} \bmod m \\ &= 4SK_i^{(1)} + SK_i^{(2)}. \end{aligned} \tag{15}$$

Since $r^2 = \pm a \bmod M$, $r = a^d \bmod M$. Hence, $r^{e_2} = a'^{d\Delta^2} \bmod M$. Therefore, it is possible to calculate

$$\begin{aligned} H_j^{(3)} &\equiv (r^{e_2})^{\lambda'_{j,0} + 2\Delta(\sum_{i=1}^{k-1} \lambda'_{j,i} SK_i^{(3)})} \\ &\equiv a'^{4SK_j^{(1)} + SK_j^{(2)}} \bmod M. \end{aligned} \tag{16}$$

Also, $a'^{d_2\Delta^2} = a \bmod M$. From this value, the missing private key share $(H_j^{(1)}, H_j^{(2)})$ could be calculated as follows:

$$\begin{aligned} H_j^{(2)} &\equiv (a'^{SK_j^{(2)}})^4 \equiv a'^{4SK_j^{(2)}} \bmod M \\ H_j^{(1)} &\equiv \frac{H_j^{(3)}}{a'^{SK_j^{(2)}}} \equiv a'^{4SK_j^{(1)}} \bmod M, \end{aligned} \tag{17}$$

where $a'^{SK_j^{(2)}}$ is calculated from

$$\begin{aligned} & a^{\lambda'_{j,0} + e_2\Delta(\sum_{i=1}^{k-1} \lambda'_{j,i} SK_i^{(2)})} \\ &\equiv a'^{SK_j^{(2)}} \bmod M. \end{aligned} \tag{18}$$

It is possible to show that the valid private key share is obtained as the result.

5. ***Challenge*** : $\mathcal{A}$ outputs two messages, M_0 and M_1, of equal length and an identifier ID, on which it wishes to be challenged. $\mathcal{B}$ forwards these values to

the Cocks' scheme challenger, receives a ciphertext and returns it to $\mathcal{A}$.

6. *Query phase 2* : $\mathcal{A}$ issues an additional queries as in Query phase1 and $\mathcal{B}$ replies to them as earlier.

7. *Guess* : Finally, $\mathcal{A}$ outputs a guess bit $b' \in \{0,1\}$. $\mathcal{B}$ forwards this bit to the Cocks' scheme challenger to finish its own game.

If $\mathcal{B}$ has generated valid e_2 during Setup simulator step, then from the $\mathcal{A}$ point of view, his interactions with $\mathcal{B}$ are statistically indistinguishable from his interactions with a real TIBE challenger. The probability of generating the valid e_2 is calculated as

$$\Pr\{e_2 : (m-d_2) \bmod 4 = 0,\ d_2 < m\} \qquad (19)$$
$$= \frac{m/2}{4m/2} = \frac{1}{4},$$

so the advantage could be calculated as

$$Adv_{B}^{IBE} = \frac{Adv_{A}^{TIBE}(k,l,\sigma)}{4} > \frac{\epsilon}{4}. \qquad (20)$$

We could conclude that the proposed TIBE scheme is as secure as its base, i.e. Cocks' IBE scheme.

5. Conclusion

We proposed TIBE scheme based on the Cocks' IBE scheme. Our scheme is non-interactive and is the first TIBE scheme proposed for a residual based IBE scheme. From security point of view, it is proven to be secure against an adaptive-ID attack in the Random Oracle Model.

References

[1] Shamir, A.: Identity-based cryptosystems and signature schemes. Proc. of Crypto 84, LNCS 196, pp.47 - 53, 1984.

[2] Cocks, C.: An Identity Based Encryption Scheme based on Quadratic Residues. LNCS 2260, pp. 360 - 363, 2001.

[3] Boneh, D., Gentry, C., Hamburg, M.: Space-Efficient Identity Based Encryption Without Pairings. Proc. FOCS 07, pp.647 - 657, 2007.

[4] Boneh, D., Franklin, M. : Identity Based Encryption from the Weil Pairing. Proc. Crypto 01, LNCS 2139, pp.213 - 229, 2001.

[5] Boneh, D., P., Boyen, X.: Efficient Selective-ID Secure Identity Based Encryption Without Random Oracles. Proc. Eurocrypt 04, LNCS 3027, pp.223 - 238, 2004.

[6] Waters, B.: Efficient Identity based Encryption without random oracles. Proc. Eurocrypt 05, LNCS 3494, pp.114 - 127, 2005.

[7] Boneh, D., Boyen, X.,Halevi, S.: Chosen Ciphertext Secure Public Key Threshold Encryption without Random Oracles. Proc. RSA-CT 06, LNCS 3860, pp.226 - 243, 2006.

[8] Li, J. , Wang, Y.: Threshold Identity Based Encryption Scheme without Random Oracles. Proc. WCAN 06, 2006.

[9] Shoup, V.: Practical threshold signatures. Proc. Eurocrypt 00, LNCS 1807, pp.207 - 220, 2000.

Sergey V. Bezzateev

e-mail : bsv@aanet.ru

1995년~현 재 러시아 세인트 피터스버그 State University of Airspace Instrumentation (SUAI) 교수

관심분야 : Coding Theory, Cryptography

김 대 엽

e-mail : daeyoub69@suwon.ac.kr

2000년 고려대학교 수학과(이학박사)

2000년~2002년 시큐아이닷컴 정보보호 연구소 PKI실 차장

2002년~2012년 삼성전자 종합기술원 전문연구원

2012년~현 재 수원대학교 정보보호학과 전임강사

관심분야 : 콘텐츠 보안, 미래 인터넷 보안