情報保護學會論文誌
第14券 第2號, 2004. 4

Schnorr 전자서명을 이용한 효율적인 Threshold 서명 기법

양 대 헌[a)†‡], 권 태 경[b)]
InHa University[a)†‡], Sejong University[b)]

Efficient Threshold Schnorr's Signature Scheme

DaeHun Nyang[a)†‡], Taekyoung Kwon[b)]
InHa University[a)†‡], Sejong University[b)]

요 약

Threshold 전자 서명은 ad hoc network 등 인프라가 없는 곳에서 유용하게 사용될 수 있고, 이에 대한 연구가 최근 증가하고 있다. 현재까지 알려진 threshold 전자 서명 기법은 주로 RSA, DSA 등에 국한되어 왔다. Schnorr 서명은 계산량 및 통신량의 측면에서 매우 효율적이지만 비대화식 증명을 이용하는 구조로 인해 threshold 기법으로 응용하는데 어려움이 있다. 이 논문에서는 이를 해결하여 Schnorr의 전자 서명의 효율적인 threshold 방법을 제안한다. 제안하는 threshold 전자 서명 기법은 매우 확장성이 좋으며, 선처리에 의해 실행시간의 계산량을 줄일 수 있는 구조를 가진다.

ABSTRACT

Threshold digital signature is very useful for networks that have no infrastructure such as ad hoc network. Up to date, research on threshold digital signature is mainly focused on RSA and DSA. Though Schnorr's digital signature scheme is very efficient in terms of both computation and communication, its hard structure using interactive proof prevents conversion to threshold version. This paper proposes an efficient threshold signature scheme based on the Schnorr's signature. It has a desirable property of scalability and reduces runtime costs by precomputation.

Keywords: *Information Security, Threshold Cryptography, Schnorr Scheme, Digital Signature*

1. 서 론

Shamir에 의해서 소개된 threshold 암호[1]는 전체 n개의 멤버들 중 어떤 t개의 멤버로 구성된 부분 집합에 대해서 항상 유효한 암호 연산을 수행할 수 있도록 하지만, 그 보다 적은 수의 멤버들에게는 결코 허용하지 않는 특성이 있다. 따라서 관련 응용 분야를 넓히기 위한 연구가 다각적으로 진행되어 왔는데[2], 특히 최근에는 ad-hoc 네트워크에서의 유용성이 입증되면서 더 많은 관심을 불러 일으키고 있다. 예를 들면, ad-hoc 네트워크에서 가장 다루기 어려운 문제 중 하나로서 공개키 인증을 고려해 볼 수 있는데, 이것의 어려움은 언제든 비협조적인 노드로 바뀔 수 있는 어떠한 단일 노드에 키 인증과 같은 중대한 기능을 부여하는 것이 바람직하지 않다는 사실에서 기인한다. 따라서 이와 같이 중대한 기능을 다수의 노드에 공평하게 분산할 수 있는 기법이 필요

접수일: 2003년 12월 31일; 채택일: 2004년 4월 3일
† 주저자, ‡ 교신저자 : nyang@inha.ac.kr

하며, 바로 threshold 암호가 이를 위한 가장 유용한 도구라고 할 수 있다. 하지만 threshold 암호는 비효율적인 측면에서 실용화하기에는 다소 어려움이 따른다.

본 논문에서는 효율성과 확장성 면에서 많은 장점을 갖는 Schnorr 전자서명[3]을 위한 threshold 기법을 제안한다. Schnorr 전자서명은 서명 생성 단계에서 계산량이 적다는 이유로 특히 저전력 계산 장치를 위해서 유용하지만, 오히려 비대화식 영지식 증명에 기반한 본질적인 구조로 인하여 threshold 기법을 적용하기에는 어려움이 따른다. 본 논문에서는 이와 같은 문제를 쉽게 해결하기 위하여 두 개의 비밀 커브를 사용하도록 한다. 그 중 하나는 비밀 share 분산을 위한 것이고 나머지 하나는 random commitment share 분산을 위한 것으로서, 효과적으로 Schnorr 전자서명을 위한 threshold 기법을 제공하게 된다.

2. Schnorr 서명 기법

Schnorr 전자서명은 비대화식 증명에 기반하며 특히 서명 생성 단계에서 multi-precision 곱셈을 요구하지 않는 등 매우 효율적인 반면, 서명 검증을 위해서는 상대적으로 많은 계산량을 요구한다[3]. 따라서 다수의 서명자가 존재하는 응용 분야에 적용하기 위해서 threshold 기법으로 확장할만한 가치가 있다[4]. 본 절에서는 먼저 Schnorr 전자서명을 간략히 살펴보도록 한다.

먼저 초기화를 위해서는, 키인증기관(KAC: key authentication center)이 조건 $q \mid p-1$을 만족하는 두 개의 임의의 소수 $q \geq 2^{106}$와 $p \geq 2^{1024}$를 선택하고, 위수 q를 갖는 원소 $g \in Z_p - \{1\}$를 구한다. 값 g, p, q는 일방향해쉬함수 $H()$와 함께 공개된다. 각 사용자는 개인키 Z^*_q를 임의로 선택하고, 이에 대응하는 공개키 $V \equiv g^{-s} \bmod p$를 계산한다.

어떤 메시지 M에 대한 서명 생성을 위해서, 서명자는 난수 $c \equiv g^r \bmod p$를 생성하고 질문값 $e = H(M, c)$를 계산한다. 또한 서명자는 값 $y \equiv r + se \bmod q$를 계산하고, 최종적으로 서명값 (e, y)를 검증자에게 전송한다. 검증자는 값 $e \equiv H(M, c \equiv g^y) * V^e \bmod p$의 검증을 통해서 서명 검증을 수행한다.

3 Threshold Schnorr 전자 서명

3.1 Share 배포 방법

만약 부분적으로 t 명이 합의할 경우 함께 서명을 생성할 수 있지만, 그 이하의 참여자들은 서명 생성을 위한 비밀값에 대한 어떠한 정보도 얻을 수 없는, 그러한 n명의 참여자가 있다고 가정하자. $i = 1, \ldots, n$일 경우, 검증자를 v_i로 표기하며, 모든 참여자의 집합을 Γ이라 한다. 즉, 크기 t를 가지는 Γ의 어떠한 부분집합도($\Omega = \{P \mid P \subset \Gamma \text{ and } \mid P \mid = t\}$를 만족하는 $P \in \Omega$) 서명을 생성할 수 있다. 값 p, q, g, 는 Schnorr 기법에서와 같이 선택되고 공개된다[3].

단계 1: 신뢰할 수 있는 딜러는 개인키 $s \in Z^*_q$를 임의로 선택하고 이에 대응하는 공개키 $V \equiv g^{-s} \bmod p$를 계산한다. 또한 $a_i \in Z^*_q$와 $a_0 = s$ 를 만족하는 비밀 커브 $A(Z) = a_0 + a_1 z + a_2 z^2 + \cdots a_{t-1} z^{t-1} \bmod q$를 선택한 후, 점 $(x_{v_i}, A(x_{v_i}))$를 각 검증자 v_i에게 배포한다. $i = 1, \ldots, n$일 경우, 값 x_{v_i}는 공개값이며, $A(x_{v_i})$는 v_i에게 비밀리에 전송되는 비밀값이다.

단계 2: 참여자 v_i는 $s_{v_i} = l_{v_i} \cdot A(x_{v_i}) \bmod q$를 계산한 후 이것을 비밀리에 보관한다. 여기서 $l_{v_i} = \prod_{1 \leq k \leq t,\ k \neq i} \frac{x_{v_k}}{x_{v_k} - x_{v_i}} \bmod q$는 Lagrange 계수이며, 따라서 $s = \sum_{v_i \in P} s_{v_i} \bmod q$이다.

단계 3: 신뢰할 수 있는 딜러는 비밀 위탁값 $r \in Z^*_q$과 다른 임의의 커브 $B(z) = b_0 + b_1 z + b_2 z^2 \cdots + b_{t-1} z^{t-1} \bmod q$를 선택하는데, 이 때 $b_i \in Z^*_q$이며 $b_0 = r$이다. 딜러는 점 $(x_{v_i}, B(x_{v_i}))$를 참여자 v_i에게 배포하는데, 역시 값 x_{v_i}는 공개값이며, $B(x_{v_i})$는 v_i에게 비밀리에 전송되는 비밀값이다.

단계 4: 참여자 v_i는 $r_{v_i} = l_i \cdot B(x_{v_i}) \bmod q$를 계산한 후 이것을 비밀리에 보관한다. 여기서 $r = \sum_{v_i \in P} r_{v_i} \bmod q$이다.

보다 신속한 전자서명 생성을 위해서, 커브 $B(z)$의 share 값들을 실제로 서명을 생성하기 이전에 미리 배포할 수 있다. 또한 하나 이상의 커브 $B(z)$의 share 값들도 프로토콜의 연속적인 수행을 위해서 미리 배포할 수 있다. 이 경우 메시지에 대한 서명생성을 요청하는 참여자는 메시지의 앞에 사용할 commitment를 위한 커브의 일련번호를 붙여서 broadcasting하므로써 동기화 문제를 해결할 수 있다. 이와 같은 사전배포를 통해서, share 배포 단계를 프로토콜 수행 시간으로부터 분리할 수 있다.

3.2 서명 생성 및 검증

서명 생성 단계에서 그룹의 어떠한 참여자도 combiner의 역할을 수행할 수 있지만, 편의상 서명을 원하는 메시지의 소유자로 가정한다.

단계 1: 먼저 combiner는 서명될 메시지 M을 broadcasting한다.

단계 2: 모든 참여자는 각자 자신의 partial commitment 값인 $c_{v_i} \equiv g^{r_{v_i}} \bmod p$를 계산한 후 broadcasting한다. 이것은 사전 계산 단계에서 수행할 수도 있다.

단계 3: 참여자 v_i는 t개의 값 c_{vk}를 고른 후 값 $e = H(M, \prod_{v_k \in P} c_{v_k} \bmod p)$와 $y_{v_i} \equiv r_{v_i} + s_{v_i} e \bmod q$를 계산한다. 그리고 결과값 (e, y_{v_i})를 combiner에게 전송한다.

단계 4: 마지막으로 combiner는 t개의 부분 서명 (e, y_{v_i})를 고르는데, 이때 반드시 e값들은 모두 동일하여야 한다. 그리고 최종적인 서명값 (y, e)를 간단한 계산($y \equiv \sum_{v_i \in P} y_{v_i} \equiv \sum_{v_i \in P} r_{v_i} + e\sum_{v_i \in P} s_{v_i} \equiv r + es \bmod q$)을 통해서 얻어낸다.

서명값 (y, e)의 검증을 위해서 검증자는 간단히 Schnorr 기법을 따라서 $e \equiv H(M, c \equiv g^y \cdot V^e \bmod p)$의 동치 여부를 검사한다. 비록 commitment 배포를 위한 집합 P와 비밀 share 배포를 위한 집합이 서로 다르지만, 전자서명 생성은 명확하게 이루어진다.

3.3 Share 갱신 및 신뢰기관을 가정하지 않는 Share 배포

제안하는 방법에서 share의 배포는 Shamir의 secret sharing을 이용하고 있다. 따라서, 이 논문에서 제안하는 threshold 전자 서명은 다음의 의미에서 homomorphic하다고 할 수 있다: 어떤 비밀 s의 share 할당 방법 $(s_1, s_2, \ldots, s_n)$와 s'의 share 할당 방법 $(s'_1, s'_2, \ldots, s'_n)$에 대해 $(s_1 + s'_1, s_2 + s'_2, \ldots, s_n + s'_n)$가 $s + s'$의 share 할당 방법이 된다. Homomorphism의 성질을 이용하면 share의 배포를 위한 신뢰할 수 있는 배포자(Trusted Dealer)를 프로토콜에서 제거할 수 있다. 이 경우에도 배포된 share s_{v_i}가 abelian 그룹에 속하고, Shamir의 secret sharing이 완전하므로(perfect secret sharing), $t-1$명의 shareholder는 비밀키 s에 대해 아무런 정보도 얻지 못하게 된다[2]. 같은 방법으로, commitment 수 r도 신뢰할 수 있는 배포자가 없이 share 배포를 할 수 있게 된다.

또한 homomorphism을 이용하면, share를 갱신하는 작업을 공개키 V나 비밀키 s를 바꾸지 않고 원래의 share 할당 방법에 0의 share 할당방법을 더함으로써 쉽게 수행할 수 있다. 자세한 내용은 [2]를 참고하라.

4 안전성 증명

안전성 증명은 Schnorr 프로토콜을 이 논문에서 제안한 프로토콜로 reduce함으로써 보인다. 증명의 쉬운 이해를 위해, 다음의 lemma로 시작한다.

Lemma 1. Schnorr 서명 방법의 모든 transcript $[c \equiv g^r \bmod p, e, y \equiv r + e \cdot s, \bmod q]$에 대해서, 비밀을 s, commitment 수를 r로 가지는 제안한 서명 방법의 threshold transcript $[(c_{v_1}, c_{v_2}, \ldots, c_{v_t})], e, [(y_{v_1}, y_{v_2}, \ldots, y_{v_t})]$가 존재한다. 또한, r과 s를 이용하지 않고 threshold transcript를 구성할 수 있다.

증명 증명은 constructive하다. 임의의 다항식 $B(z) \equiv b_0 + b_1 z + b_2 z^2 + \ldots b_{t-1} z^{t-1} \bmod q$와 $D(z) \equiv d_0 + d_1 z + d_2 z^2 + \ldots d_{t-1} z^{t-1} \bmod q$를 선택한다. 이때 $b_0 = d_0 = 0$이다. $1 \le i \le n$에 대해 $k_{v_i} = l_{v_i} \cdot$

$B(x_{v_i}) \bmod q$.

$m_{v_i} = l_{v_i \cdot P}(x_{v_i})$라 하자. 이 때 $\sum_{v_i \in P} k_{v_i} \equiv 0 \bmod q$ 이다. 따라서 다음의 transcript

$$[(c_{v_1} = -g^{k_{v_1}} \cdot c^{1/t}, c_{v_2} \equiv g^{k_{v_2}} \cdot c^{1/t}, \cdots, c_{v_n} \equiv g^{k_{v_n}} \cdot c^{1/t}), e, (y_{v_1} \equiv y/t + m_{v_1} y_{v_2} \equiv y/t + m_{v_2} \cdots, y_{v_n} y/t + m_{v_n}$$

는 transcript (c, e, y)와 같은 commitment r, 같은 비밀키 s를 가진다. 이는

$$\Pi_{v_i \in P} c_{v_i} \equiv g^{\sum_{v_i \in P} k_{v_i}} \cdot (c^{1/t})^t \equiv g^0 \cdot c \equiv c \bmod p$$

그리고

$$\sum_{v_i \in P} y_{v_i} \equiv \sum_{v_i \in P} y/t + \sum_{v_i \in P} m_{v_i} \equiv t(y/t) + 0 \equiv y \bmod q$$

이기 때문이다. 또한, 위의 threshold transcript를 구성하는데 r과 s를 이용하지 않았다. □

정리 1. 만약 transcript $[(g^{r_{v_1}}, g^{r_{v_2}}, \ldots, g^{r_{v_t}}), e, (y_{v_1}, y_{v_2}, \ldots y_{v_t})]$로 부터 비밀키 s를 찾아내는 알고리즘이 존재한다면, 그 알고리즘은 Schnorr 서명의 transcript $[g^r, e, y]$로부터 비밀키 s를 찾아낼 수 있다. 여기서 $r \equiv \sum_{v_i \in P} r_{v_i} \bmod q$ $y \equiv \sum_{v_i \in P} y_{v_i} \bmod q$ $y \equiv r + es \bmod q$이다. 따라서 이 논문에서 제안하는 threshold Schnorr 서명 기법은 적어도 Schnorr의 전자 서명 만큼 안전하다.

증명 Threshold transcript $[(g^{r_{v_1}}, g^{r_{v_2}}, \ldots, g^{r_{v_t}}), e, (y_{v_1}, y_{v_2}, \ldots y_{v_t})]$로 부터 비밀키 s를 다항시간내에 계산하는 adversary A가 있다고 가정하자. 여기서 $r \equiv \sum_{v_i \in P} r_{v_i} \bmod q$ $y \equiv \sum_{v_i \in P} y_{v_i} \bmod q$ $y \equiv r + es \bmod q$ 이다. 이 adversary A를 이용하면, 우리는 다항시간내에 $[g^r, e, y]$로부터 비밀키 s를 계산하는 알고리즘 R을 다음과 같이 정의할 수 있다:

① R은 임의의 다항식 $B(z) \equiv b_0 + b_1 z + b_2 z^2 + \ldots b_{t-1} z^{t-1} \bmod q$를 선택한다. 여기서 $b_0 = 0$이다. 이를 이용해 $k_i = l_i \cdot B(x_{v_i})$를 계산한다. 이 때 $0 = \sum_{v_i \in P} k_{v_i}$임을 주목하라.

② $c \equiv g^r \bmod p$를 이용해서 R은 다음을 계산한다.

$$(c_{v_1} \equiv g^{k_{v_1}} \cdot c^{1/t} \bmod p, c_{v_2} \equiv g^{k_{v_2}} \cdot c^{1/t}, \ldots, c_{v_t} \equiv g^{k_{v_t}} \cdot c^{1/t})$$

③ ①에서와 마찬가지로 R은 임의의 다항식 $D(z) \equiv d_0 + d_1 z + d_2 z^2 + \ldots d_{t-1} z^{t-1} \bmod q$를 선택한다. 여기서 $d_0 = 0$이고 $m_{v_i} = l_{v_i} \cdot D(x_{v_i})$를 계산한다. $d_0 = 0$이므로 $0 = \sum_{v_i \in P} m_{v_i}$이 된다.

④ y를 이용해서 R은 다음을 계산한다.

$$y_{v_i} \equiv y/t + m_{v_1} \bmod q, \; y_{v+2} \equiv y/t + m_{v_2} \ldots y_{v_k} \equiv y/t + m_{v_t}$$

⑤ 이제 R은 입력을 $[(c_{v_1}, c_{v_2} \ldots c_{v_t}) i)]$ e, $[(y_{v_1}, y_{v_2}, \ldots, y_{v_t})]$로 해서 adversary A를 서브루틴으로 호출한다. Adversary A가 돌려주는 결과는 Lemma 1에 의해 s가 되며, R은 이 값을 출력한다.

Adversary A의 실행시간을 $poly_A$라고 하면, 알고리즘 R의 전체 수행시간은 $O(t \log q + poly_A)$가 되며 이는 다항시간 클래스에 포함된다. 여기서 $O(t \log q)$는 2에서 partial commitment를 계산하는데 드는 비용을 의미한다. 즉, c_{vi}하나를 계산하는데 드는 곱셈의 수가 $O(\log q)$, 그리고 t개의 c_{vi}를 계산해야하므로 $O(\log q)$의 실행 시간을 가진다. 다른 단계에서의 계산 시간은 이에 비해 무시할 수 있다. □

5. 성능 평가

x를 q의 크기에 대한 보안 매개변수, l을 p에 대한 보안 매개변수라고 하자. 만약 지수가 임의로 선택되고 Z_p^*의 q차 서브그룹에서 square-and-multiply 알고리즘이 사용된다면, 멱승계산에는 x번의 멱제곱과 평균 $\frac{x}{2}$번의 멱곱셈이 필요하게된

다. 이를 기준으로 계산량을 추정하면 다음과 같다.

초기화 과정에서는 신뢰할 수 있는 배포자가 V를 계산하기 위해 Z_p^*에서 $1.5x$번의 곱셈을 해야하고, $A(x_{v_i})$의 계산을 위해 Z_q^*에서 $(t-1)n$번의 곱셈을 수행해야한다. 프로토콜에 참여하는 각각의 참여자는 s_{v_i}의 계산을 위해 Z_q^*에서 평균 $32t-3$번의 곱셈을 수행해야한다. Commitment수를 미리 배분하기 위해서 m개의 커브를 미리 이용한다면, 신뢰할 수 있는 배포자는 m가지의 $B(x'_{v_i})$ 를 얻기 위해서 Z_q^*에서 $(t-1)nm$번의 곱셈을 해야한다. 이 때 각 참여자가 m가지의 r_{v_i}을 얻기위해서는 Z_q^*에서 $(32t-3)m$번의 곱셈만을 수행하면 된다.

서명을 생성하기 위해서, 각 참여자는 c_{v_i}를 계산하기 위해 Z_p^*에서 $1.5x$번의 곱셈을 할 필요가 있고, (e, y_{v_i})를 계산하기 위해서 Z_q^*에서 $t-1$번의 곱셈, 그리고 Z_q^*에서 한번의 곱셈을 수행해야 한다. 부분 서명을 모아서 결합하는 참여자는 Z_p^*에서 덧셈만을 수행하면 된다. 서명을 확인하기 위해서는, Schnorr의 서명과 마찬가지로 Z_p^*에서 $3x$번의 곱셈을 수행하면 된다.

따라서 Schnorr 서명 기법과 비교할 때, n명의 참가자가 참여하는 threshold 전자 서명의 경우 n배의 연산량의 증가가 필요하고, 이 n배의 연산은 각 참여자들이 동시에 수행하게 되므로 수행시간은 원래의 Schnorr 서명기법과 동일하다. 또한, 실시간에 필요한 연산의 수는commitment에 사용되는 커브를 미리 배포함으로써 줄일 수 있는데, 이렇게 하면 서명의 생성에 Z_p^*에서 $t-1$번의 곱셈만이 필요하다. 이는 참여하는 노드의 수에 비례하므로 매우 확장성이 좋다고 할 수 있다. 또한 서명의 검증에는 참여하는 노드의 수에 관계없이 Z_p^*에서 $3x+1$번의 곱셈만이 필요하다.

통신량의 측면에서 볼 때, 각 참여자는 $h$bit 길이의 서명할 메시지의 수신을 위해 h비트의 broadcasting 수신 채널, l비트의 broadcasting 송신 채널, 그리고 $2x$비트의 unicasting 송신 채널을 필요로 한다. 부분 서명을 결합하는 참여자의 경우 h비트의 broadcasting 송신 채널그리고 $2xn$의 unicasting 수신 채널을 필요로 한다.

본 논문에서 제안한 기법은 Z_p^* 에서 연산을 수행하며, 따라서 1996년 Gennaro, Jarecki, Krawczyk, Rabin이 제안한 Threshold DSS 기법과 효율성을 비교할 수 있다 [5]. Threshold DSS의 특징은 Joint Shamir Random Secret Sharing (JSRSS)을 사용하여 본 논문의 3.3절에서 언급한 바와 같이 신뢰 배포자를 제거한 것이다. Threshold DSS의 자세한 사항에 대해서는 [5]를 참고하라. 우선 본 논문에서 제안한 기법은 2개의 임의의 커브가 필요한 반면, Threshold DSS는 5개의 임의의 커브를 필요로 한다. 또한 서명 생성을 위해서 제안 기법은 Z_p^*에서 $1.5x+t-1$번의 곱셈을 필요로 하는 반면, Threshold DSS는 $(1.5t+4.5)x$번의 곱셈을 필요로 한다. 즉 Threshold DSS는 기본적으로 $4.5x$번의 곱셈 이외에도 t의 크기에 따라 $1.5x$배씩 늘어나서 매우 커지는 반면, 제안 스킴은 기본적으로 $1.5x$번의 곱셈 이외에 단순히 t에 대해서만 선형적으로 늘어나게 된다. 결과적으로 제안 기법이 효율적인 것을 쉽게 알 수 있다.

6. 결 론

이 논문에서는 잘 알려진 Schnorr의 전자 서명에 기초해서 효율적인 threshold 전자 서명 방법을 제안했다. 제안한 전자 서명 방법의 안전성은 Schnorr 전자 서명을 제안한 전자 서명으로 reduction하므로써 증명했다. 서명을 생성하는데 각 참여자들이 Z_p^*에서의 곱셈을 $(t-1)$번, 그리고 부분 서명을 합치는 참여자는 Z_q^*에서 $(t-1)$번의 덧셈만을 수행하게 된다. 여기서 Z_q^*에서의 덧셈은 Z_p^*에서의 곱셈에 비해 매우 빠르다. 따라서, 제안한 threshold 전자 서명 기법은 서명에 참여하는 참여자의 숫자에 비례해서 Z_p^*에서의 곱셈의 숫자가 늘어나므로 매우 확장성이 우수하다.

참고문헌

[1] A. Shamir, How to Share a Secret, Commun. of the ACM, Vol. 22, Nov. 1979, pp. 612613.

[2] Y. Desmedt, Some Recent Research Aspects of Threshold Cryptography,

Proceedings of Information Security, LNCS, Springer-Verlag, 1997, pp. 158173.
[3] C.P. Schnorr, Efficient Identification and Signatures for Smart cards, Advances in Cryptology : Proceedings of Crypto 89, LNCS, Springer-Verlag, 1989, pp. 239251.
[4] S. Micali, K. Ohta, L. Reyzin, Accountable-Subgroup Multisignatures, Proceedings of ACM Computer and Communications Security, 2001, pp. 245253.
[5] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, Robust threshold DSS signatures, Advances in Cryptology : Proceedings of Eurocrypt 96, LNCS 1070, Springer-Verlag, 1996, pp. 354371.

〈著者紹介〉

양 대 헌 (DaeHun Nyang) 종신회원
1994년 2월: 한국과학기술원 과학기술대학 전기 및 전자 공학과 졸업
1996년 2월: 연세대학교 컴퓨터 과학과 석사
2000년 8월: 연세대학교 컴퓨터 과학과 박사
2000년 9월~2003년 2월: 한국전자통신연구원 정보보호연구본부 선임연구원
2003년 2월 ~현재: 인하대학교 정보통신대학원 전임강사, 정보보호학회 편집위원, TTA 정보보호/인터넷보안 분과위원
〈관심분야〉 암호이론, 암호프로토콜, 인증 프로토콜, 무선 인터넷 보안

권 태 경 (Taekyoung Kwon) 종신회원
1992년: 연세대학교 컴퓨터과학과 졸업
1995년: 연세대학교 컴퓨터과학과 석사
1999년: 연세대학교 컴퓨터과학과 박사
1999년~2000년: U.C. Berkeley Post-Doc.(과학재단/삼성전자 지원)
2001년~현재: 세종대학교 컴퓨터공학부 컴퓨터소프트웨어학과 조교수, 정보보호학회 편집위원, TTA 암호분과 특별위원
〈관심분야〉 정보보호, 암호프로토콜, 네트워크 보안, 무선 네트워크 등