전방 비밀성을 제공하는 개선된 Signcryption 프로토콜

정회원 이 경 현*, 조 현 호**, 이 준 석***

An enhanced signcryption protocol for providing forward secrecy

Kyung-hyune Rhee*, Hyun-ho Cho**, Jun-Seok Lee*** *Regular Members*

요 약

RSA 기반 서명 후 암호화 기법은 forward secrecy를 제공하는 반면 총 4번의 모듈러 멱승 연산을 요구하며, Zheng에 의해 제안된 signcryption은 논리적으로 한 번에 서명과 암호화를 수행하여 기존의 서명 후 암호화에서 요구되는 계산 비용보다 더 적은 비용을 가지는 반면 forward secrecy를 제공하지 못한다. 본 논문에서는 RSA 기반 서명 후 암호화 기법보다 적은 계산 비용과 통신 오버헤드를 가지며, 또한 forward secrecy를 제공하는 개선된 signcryption 기법을 제안한다. 제안 방안은 수신자의 비밀키를 사용하지 않고 직접 검증 가능한 방안으로 쉽게 변형될 수 있다. 또한 제안 signcryption 방안을 응용하여 공개키 사용 프로토콜의 선결과제인 CRL 처리 문제를 해결하면서 명시적인 송신 부인 방지 서비스를 제공하는 서버 지원 서명된(server-supported signature) 변형 signcryption 방안을 제시한다. 서버 지원된 변형 signcryption 방안은 기존의 signcryption 기법에도 적용이 가능하다.

ABSTRACT

The signature-then-encryption based on RSA scheme provides forward secrecy, but requires 4 modulo exponentiation operations in total, and the signcryption scheme proposed by Zheng simultaneously fulfills both the functions of digital signature and symmetric key encryption in a logically single step, and with a computational cost significantly smaller than that required by the current standard signature-then-encryption, but it can not provide forward secrecy. In this paper, we propose an enhanced signcryption scheme which can provide forward secrecy with lower computational cost and lower communication overhead comparing with those of the signature-then-encryption based on RSA, and with a similar communication overhead of Zheng's scheme. The proposed scheme can be also easily modified to the direct signature verification scheme by the recipient without using the recipient's private key. Additionally, we suggest a new design protocol with server-supported signatures which solves the CRLs(Certificate Revocation Lists) burden and provides non-repudiation of origin. This protocol with server-supported signatures also can be applied to the original signcryption scheme proposed by Zheng in order to improve security.

I. 서 론

서명 후 암호화 방식은 RSA 서명에 기반한 방식과 ElGamal 서명, 그리고 Schnorr 서명에 기반한 방식 등이 있으며, 이러한 서명에 추가하여 대칭키 암호화를 수행한다. 이에 비해 Signcryption 기법은

* 부경대학교 전자컴퓨터정보통신공학부(khrhee@pknu.ac.kr), ** 동부산대학 컴퓨터정보학부(chohh@sb.dpc.ac.kr)
*** 부경대학교 전자계산학과(jsmagic@unicorn.pknu.ac.kr)
논문번호 : 010397-1217, 접수일자 : 2001년 12월 18일
※본 연구는 2001년 한국학술진흥재단의 지원(KRF-2001-013-E00064)에 의하여 연구되었음.

서명과 암호화를 논리적으로 한 번에 수행함으로써 기존의 서명 후 암호화(signature-then-encryption)기법들에서 요구되는 계산 비용보다 적은 비용을 가지는 새로운 암호학적인 기법이다[1].

Signcryption에서는 보안 파라메터(security parameter)들의 크기에 비례해서 비용 효율이 증가하기 때문에 높은 레벨의 보안을 요구하는 어플리케이션에 대해 더욱 큰 비용 효율을 기대할 수 있다. 하지만 기존 signcryption 기법에서 만일 signcryption측의 비밀키가 노출된다면, 그 키를 알고 있는 사람은 노출된 키에 의해 signcrypt된 암호문을 복구할 수 있게 되므로 Zheng이 제안한 signcryption 기법은 forward secrecy를 제공하지 못한다. forward secrecy에 대한 정의로서, 만일 암호 프로토콜에서 오랜 기간 사용되는 비밀키의 분실이 이전에 그 키를 사용하여 비밀 통신을 수행한 메시지(들)을 노출시키지 않는다면, 그 프로토콜은 forward secrecy를 제공한다라고 할 수 있다[2].

본 논문에서는 signcryption시의 비밀키로써 두 형태의 키가 사용되는 변형된 signcryption 기법을 제안하고 있으며, 서버 지원에 의한 서명을 하기 위해 RSA 키 쌍을, 그리고 기존의 signcryption 기법을 응용하기 위해 ElGamal 키 쌍을 이용한다. 비밀키들의 안전성(security)은 RSA 알고리즘에서의 소인수분해 문제의 어려움과 ElGamal에서의 이산대수 문제 어려움에 기반하고 있으며, 세션키의 안전성은 이산대수문제의 어려움에 기반한다. 제안하는 기법은 기존 기법과는 달리 forward secrecy를 제공하는 장점을 가지면서도 RSA에 기반한 서명 후 암호화 기법보다 계산 비용이 적고 통신 오버헤드를 작게 가진다. 또한 본 논문에서 보다 향상된 검증 방안으로 수신자의 비밀키를 사용하지 않고 직접 검증 가능한 변형 signcryption 기법을 고려한다. 이에 추가하여 제안 방안을 응용함으로써, 공개키 사용 프로토콜의 선결과제인 CRL 처리 문제를 해결하면서 명시적 송신 부인 방지 서비스를 제공하는 서버 지원 서명된(server-supported signature) signcryption 방안을 제안한다. 그리고, 기존 signcryption 기법이 forward secrecy를 제공하지 못하는 단점에 대한 대책으로 signcryption 비밀키를 매 세션마다 갱신하도록 서버 지원 서명된 기법을 응용할 수 있다.

본 논문의 구성은 먼저 Ⅱ절에서 forward secrecy를 제공할 수 있는 변형 signcryption 기법을 제안하며, 안전성(security)과 성능(performance) 측면의 분석을 Ⅲ절에서 설명한다. Ⅳ절에서는 수신자의 비밀키를 사용하지 않고 직접 서명 검증이 가능하도록 변형하는 방안에 대해 논하고, 또한 다수의 수신자에 대한 signcryption 기법인 multi-signcryption 방안을 제시한다. 이에 추가하여 제안한 기법을 응용하여 수신측에서 송신측의 공개키에 대해 CRLs (Certificate Revocation Lists) 질의를 수행할 필요가 없을 뿐만 아니라 높은 안전성과 명시적 송신 부인 방지 서비스를 제공할 수 있는 서버 지원 서명된 변형 signcryption 프로토콜을 제안한다. 그리고 Ⅴ절에서는 기존 signcryption 기법이 forward secrecy를 제공하지 못하는 단점에 대한 대책으로서 Ⅳ절에서 제안되는 서버 지원 서명된 기법을 적용하는 방안을 제시한다. 마지막으로 Ⅵ절에서 결론을 맺는다.

Ⅱ. 제안 signcryption 기법

제안 기법을 기술하기에 앞서 본 기법에서 사용되어지는 signcryption 키와 unsigncryption 키에 대한 정의를 하도록 한다. 먼저, 송신자(A로 표기)는 RSA 형태의 키 쌍(RSA 공개키와 비밀키)과 ElGamal 형태의 키 쌍(ElGamal 공개키 및 비밀키)을 이용하여 signcryption을 실행하므로, A의 비밀키는 RSA 및 ElGamal 비밀키가 되고, 이를 signcryption 키라고 한다. 그리고, 수신자(B로 표기)는 ElGamal(또는 Diffie-Hellman) 형태의 키 쌍(ElGamal 공개키와 비밀키)을 이용하여 unsigncryption을 실행하므로, B의 비밀키는 ElGamal 비밀키가 되고, 이를 unsigncryption 키라고 한다.

signcryption 절차에서 송신자 A는 RSA 공개키 및 비밀키를 다음과 같이 생성한다.

$$KA_e \cdot KA_d \equiv 1 \bmod \phi(n_A)$$

여기서, KA_e는 공개키, KA_d는 비밀키이고, p_A와 q_A는 큰 소수, $n_A = p_A \cdot q_A$, 그리고 $\phi(n_A) = (p_A - 1)(q_A - 1)$이다.

그 후, $\lambda | \phi(n_A)$인 큰 소수 λ를 선택하고, 또한 법 n_A상에서 λ를 위수로 가지는 (임의의) g_A를 선택한다. 주의할 것은 λ는 A만 알고 있는 비밀정보라는 것이다. 또한 $x_A \in [1, \cdots, \lambda - 1]$을 선택하여 ElGamal 비밀키로 하고, ElGamal 공개키 $y_A = g_A^{x_A} \bmod n_A$를 생성한다. 이제 송신자 A는 다음

과 같이 signcryption 절차를 수행한다.

단계 1. 비밀 정수값 $x \in [1, \cdots, \lambda-1]$를 임의로 선택한다.

단계 2. 다음의 식과 같이 세션키 k를 계산한다.

$$k = hash(y_B^{x \cdot KA_e} \bmod n_A), \quad k = k_1 \| k_2 \tag{1}$$

여기서, y_B는 수신자 B의 공개키로써 $y_B = g_A^{x_B} \bmod n_A$(x_B는 수신자 B의 비밀키)이고, $hash$는 일 방향 해쉬 함수이며, $\|$는 연접(concatenation)을 의미한다.

단계 3. 다음과 같이 메시지 암호화를 수행하고 서명값을 생성한다.

$$c = E_{k_1}(m)$$
$$r = KH_{k_2}(m)$$
$$s = (x - x_A \cdot KA_d)/(KA_d \cdot r + 1) \bmod \lambda$$

여기서, $E(\cdot)$는 대칭키 암호화 함수이고, $KH(\cdot)$는 keyed-해쉬 함수를 나타낸다.

단계 4. 마지막으로 (c, r, s)를 수신자 B에게 전송한다.

송신자 A로부터 (c, r, s)를 수신 받은 후, 수신자 B는 다음과 같이 unsigncryption 절차를 수행한다.

단계 1. 다음 식에 의해 세션키 k를 계산한다.

$$k = hash((y_A \cdot g_A^{(r+KA_e) \cdot s})^{x_B} \bmod n_A) \tag{2}$$
$$k = k_1 \| k_2$$

단계 2. 암호화된 메시지의 복호 및 검증 절차를 다음과 같이 수행한다.

$m' = D_{k_1}(c)$

$KH_{k_2}(m') = r$이면 m'을 송신자 A로부터 전송된 유효한 메시지로 받아들인다.

식 (2)에서 다음 식이 만족되기 때문에, 수신자 B는 세션키 k를 정확히 계산해낼 수 있다:

$$s \equiv \frac{x - x_A \cdot KA_d}{KA_d \cdot r + 1} \pmod{\lambda} \equiv \frac{KA_e \cdot x - x_A}{r + KA_e} \pmod{\lambda} \tag{3}$$

식 (2)에서 수신자 B는 세션키 k를 계산하기 위해 먼저 $v = y_A \cdot g_A^{(r+KA_e) \cdot s} \bmod n_A$를 계산하고, $v^{x_B} \bmod n_A$를 계산한다. 따라서, unsigncryption 절차에서 2번의 모듈로 멱승 연산이 요구된다.

Ⅲ. 제안 signcryption 기법의 분석

1. 제안기법의 안전성 분석

본 절에서는 제안 기법에 대한 안전성 분석을 수행한다. 분석 기법은 증명 가능한 방안은 아니나 기존의 수학적 어려움에 그 근간을 두고 있다.

제안 signcryption 기법에서는 기존 signcryption 기법과는 달리 signcryption 키들의 안전성을 소인수분해의 어려움과 이산대수의 어려움에 의해 보장하고 있으며, 또한 대칭키 암호화 및 서명키로 사용되는 세션키 $k = (k_1, k_2)$의 안전성은 이산대수문제의 어려움에 기반하고 있다. 따라서 제안 기법을 해독하기 위해서는 소인수분해문제 또는 이산대수문제를 풀어야만 한다.

그리고, 기존 signcryption 기법이 forward secrecy를 제공하지 못하는 이유는 세션키를 생성할 때 signcryption key(A의 비밀키)와의 연관성을 가지고 있기 때문이다. 그러므로, forward secrecy의 관점에서 이런 연관성을 분석하는 작업은 의미가 있다. 따라서, 본 제안 기법에서 사용되는 키들(RSA 비밀키, 세션키, ElGamal 비밀키) 간의 연관성 존재 여부를 몇 가지 경우를 가정하여 분석하고, 또한 본 제안 기법이 forward secrecy를 제공한다는 것을 보인다.

1) 송신자 A의 RSA 비밀키가 노출되었을 경우

권한 없는 제 3자에게 송신자 A의 RSA 비밀키가 노출되었다는 것은 RSA 시스템 파라메터들이 모두 노출되었다라고 할 수 있다. 이 경우 식 (3)에서 관건이 되는 것은 $KA_e \cdot x - x_A \pmod{\lambda}$에서 랜덤 비밀값 x와 $-x_A$의 계산 가능 여부이다. 결국 권한 없는 제 3자는 이 값들을 계산해내지 못하므로 세션키를 계산할 수 없다. 세션키를 계산하기 위해서는 반드시 수신자 B의 ElGamal 비밀키를 알아야만 한다. 따라서, 본 제안 기법은 forward secrecy를 제공한다. 또한, 정당치 못한 제3자는 송신자 A의 ElGamal 비밀키 및 수신자 B의 ElGamal 비밀키에 대한 어떠한 정보도 획득할 수 없다. 그 정보를 획득하기 위해서는 반드시 이산대수문제를 풀어

야만 한다.

2) 송신자 A의 ElGamal 비밀키가 노출되었을 경우

송신자 A의 ElGamal 비밀키가 노출되었을 경우 RSA 시스템 파라메터인 λ를 모르기 때문에 역원 계산이 불가능하다. 따라서 식 (3)에서 공개 파라메터를 제외한 어떠한 값도 계산해낼 수 없으므로 권한 없는 제 3자는 세션키를 계산해낼 수 없다. 세션키를 계산하기 위해서는 반드시 수신자 B의 ElGamal 비밀키를 알아야만 한다. 따라서, 가정 1과 마찬가지로 본 제안 기법은 forward secrecy를 제공한다. 또한, 정당치 못한 제3자는 송신자 A의 RSA 비밀키 및 수신자 B의 ElGamal 비밀키에 대한 어떠한 정보도 획득할 수 없다. 그 정보를 획득하기 위해서는 반드시 소인수분해문제 또는 이산대수문제를 풀어야만 한다.

3) 송신자 A의 RSA 및 ElGamal 비밀키 모두 노출되었을 경우

송신자의 두 비밀키가 모두 노출되었을 경우는 RSA 시스템 파라메터들이 다 노출되었다 라고 할 수 있으며, 또한 송신자의 ElGamal 비밀키 x_A 역시 노출되었기 때문에, 식 (3)에서 역원 계산이 가능하여 결국 수신자 B의 비밀키를 모른다하더라도 세션키를 계산해 낼 수 있다. 이 경우는 forward secrecy를 제공하지 못한다.

가정 1), 2), 3)에 의해 본 제안 기법은 송신자의 두 비밀키 모두 노출되지 않는 한 forward secrecy를 제공할 수 있으며, 본 기법에서 사용된 키들간의 배타적 지역성(mutual exclusive and local security)을 제공한다. 또한, 기존 signcryption에서 제공되는 위조 방지(unforgeability), 부인 방지(non-repudiation) 및 비밀성(confidentiality)이 제공된다.

2. 제안 기법의 성능 분석

본절에서는 제안 기법에 대해 계산 비용과 통신 오버헤드 비용 측면의 분석을 논의한다.

계산 비용 측면에서는, 표 1과 같이 RSA에 기반한 서명 후 암호화 기법은 총 4번(송신측 2번, 수신측 2번)의 모듈로 멱승 연산을, 그리고 기존 signcryption 기법은 총 3번(signcryption 시 1번, unsigncryption 시 2번)의 모듈러 멱승 연산을 필요로 하고 있으며, 본 논문에서 제안하는 변형 기법 역시 총 3번의 모듈러 멱승 연산을 요구하고 있다. 참고로 표 1의 비교 방식은 [1]에서의 방식을 적용하였다.

표 1. Signcryption 기법들의 비용 비교

Schemes	Computational cost	Communication overhead (in bits)
RSA 기반 서명 후 암호화 기법	Exp=2, Hash=1, Enc=1 (Exp=2, Hash=1, Dec=1)	$\|n_a\|+\|n_b\|$
기존 signcryption 기법	Exp=1, Mul=0, Div=1 Add=1, Hash=1, Enc=1 (Exp=2, Mul=2, Div=0 Add=0, Hash=1, Dec=1)	$\|KH(\cdot)\|+\|q\|$
변형 signcryption 기법	Exp=1, Mul=3, Div=1 Add=2, Hash=1, Enc=1 (Exp=2, Mul=2, Div=0 Add=1, Hash=1, Dec=1)	$\|KH(\cdot)\|+\|\lambda\|$

Exp : 모듈러 멱승 연산의 수
Mul : 모듈러 곱셈 연산의 수
Div : 모듈러 나눗셈 연산의 수(역원 연산)
Add : 모듈러 덧셈 또는 뺄셈 연산의 수
Hash : 일 방향 해쉬 또는 keyed-해쉬 연산의 수
Enc : 대칭키 암호 알고리즘을 사용한 암호화 연산의 수
Dec : 대칭키 암호 알고리즘을 사용한 복호화 연산의 수
괄호( ) 안의 파라메터들은 복호 후 검증(decryption-then-verification) 또는 unsigncryption에 연관된 연산들의 수를 나타냄

모듈러 멱승 연산 시 fast exponentiation 알고리즘을 사용할 경우 그 계산 비용은 지수 크기에 결정된다. 기존 기법에서는 송/수신측에서 모듈러 멱승 연산시 그 지수 크기는 $|q|$이지만, RSA 기반 서명 후 암호 기법은 RSA 암호 및 서명 기법을 사용하기 때문에 공개 및 비밀키의 크기만큼 비용이 발생하게 된다. 본 변형 기법에서는 송신측에서 기존 기법과 마찬가지로 지수의 크기가 $|\lambda|$이지만 수신측에서는 이 λ를 알지 못하기 때문에 표 2와 같이 한 번의 멱승 연산이 $|KA_e|+|\lambda|$만큼의 지수 크기를 가지게 된다.

표 2. 모듈러 멱승 연산 시의 지수크기

기존 signcryption 기법		RSA 기반 서명 후 암호 화 기법		변형 signcryption 기법	
송신측	$\|q\|$	송신측	$\|e_b\|, \|d_a\|$	송신측	$\|\lambda\|$
수신측	$\|q\|, \|q\|$	수신측	$\|e_a\|, \|d_b\|$	수신측	$\|KA_e\|+\|\lambda\|, \|\lambda\|$

$|q|$: 기존 signcryption 기법에서 사용되는 지수연산에서의 법 q의 비트 크기
$|e_a|, |e_b|$: RSA 기반 기법에서의 송/수신측의 RSA 공개키 비트 크기
$|d_a|, |d_b|$: RSA 기반 기법에서의 송/수신측의 RSA 비밀키 비트 크기
$|\lambda|$: 변형 기법에서 사용되어지는 지수연산에서의 법 λ의 비트 크기
$|KA_e|$: 변형 기법에서의 송신측의 RSA 공개키 비트 크기

이 세 기법에서 사용되고 있는 지수의 크기를 비교하기 위해, 기존 signcryption 기법에서의 p, RSA 기반 서명 후 암호화 기법에서의 n_a, 그리고 제안된 변형 기법에서의 n_A의 크기가 1024비트로 같다고 가정한다.

$$|p| = |n_a(\text{또는 } n_b)| = |n_A| = 1024\text{비트}$$

그리고, 기존의 signcryption 기법뿐만 아니라 DSS(Digital Signature Standard), 국내 표준인 KCDSA(Korean Certificate-based Digital Signature Algorithm)에서 이산대수의 어려움을 보장하기 위해 $|p|$=1024 비트일 때 최소 160 비트 이상인 q를 사용할 것을 권고하고 있으므로 다음을 가정한다.

$$|q| = |\lambda| = 160\text{비트}$$

이 때, 기존 기법과 비교하면, 제안 기법에서의 지수 크기는 수신측에서 한 번의 모듈러 멱승 연산만이 $|KA_a|$만큼 크며, 그 만큼의 더 많은 계산 비용을 가진다. 그러나, RSA 서명 후 암호화 기법과 비교하면, 송신측에서는 훨씬 더 적은 계산 비용을 가지며, 수신측에서 $|e_a| \approx |KA_a|$라고 한다면, RSA 기반 서명 후 암호화의 경우 총 $|e_a| + |d_b| \approx 1024$ 비트 크기인 반면, 제안 기법에서 $|KA_a| \approx 512$비트일 경우 $|KA_a| + 2|\lambda| \approx 832$ 비트 크기를 가지게 된다. 따라서 제안 기법은 RSA 기반 서명 후 암호화 기법보다 적은 계산 비용을 가지며, 수신측의 계산 비용을 고려하여 공개키의 크기를 작게 한다면 RSA 기반 서명 후 암호화 기법보다 훨씬 더 적은 계산 비용을 가질 수 있다. 참고로 RSA에서의 안전성을 고려하여 비밀키 d는 $\max\{p, q\} + 1 < d < \phi(n)$를 만족하여야 하기 때문에[11], 이에 대한 공개키 e의 크기는 $e \cdot d \equiv 1 \bmod \phi(n)$에 의해 상대적으로 작은 값을 가진다.

이를 검증하기 위해 본 논문에서 실험결과를 제시한다. 컴퓨터 시뮬레이션에 사용된 소수들은 DSS나 KCDSA에서의 생성방식을 따라 생성하여 소수 판정을 거쳤으며, 각 기법에서의 시스템 파라메터 및 키들은 각 기법에서 제시되는 조건 하에서 랜덤하게 생성하였다. 또한 모듈러 멱승 연산의 수행은 fast exponentiation 알고리즘을 구현하였으며, 사용되는 해쉬 함수의 출력값의 크기를 160비트로 가정하여 수행하였다. 표 3은 그에 대한 결과를 보이고 있다.

표 3. 모듈러 멱승 연산 수행 시간

		RSA 기반 서명 후 암호화 기법					변형 기법				기존 Signcryption 기법			
$\|p\|$	$\|q\|$	단위는 초(sec)이며, 정확도는 1/100 초로 측정												
$\|n_x\|$	$\|\lambda\|$	송신		수신		계	송신	수신		계	송신	수신		계
512	144	0.01	0.11	0.01	0.17	0.30	0.05	0.11	0.05	0.21	0.06	0.05	0.01	0.12
768	152	0.05	0.22	0.05	0.22	0.54	0.06	0.05	0.01	0.12	0.05	0.06	0.06	0.17
1024	160	0.11	0.44	0.06	0.49	1.10	0.06	0.11	0.11	0.28	0.05	0.05	0.11	0.21
1280	168	0.16	0.77	0.11	0.82	1.86	0.11	0.22	0.11	0.44	0.11	0.11	0.05	0.27
1536	176	0.22	1.37	0.22	1.32	3.13	0.11	0.38	0.16	0.65	0.17	0.16	0.11	0.44
1792	184	0.39	1.92	0.39	1.92	4.62	0.22	0.55	0.17	0.94	0.22	0.22	0.17	0.61
2048	192	0.55	2.91	0.55	2.86	6.87	0.22	0.77	0.22	1.21	0.27	0.27	0.27	0.81
2560	208	1.04	5.33	0.99	5.43	12.79	0.44	1.32	0.5	2.26	0.43	0.44	0.44	1.31

표 3에서 보여지는 Security Parameter에서의 비트 크기는 [1]을 근거로 하였으며, 시뮬레이션 시의 하드웨어 환경은 Pentium III 700MHz, RAM 192M이며, 사용언어는 C, C++, 컴파일러는 VC++6.0, 배정도연산 라이브러리는 Miracl을 사용하였다.

그림 1은 표 3에서의 수행시간을 Security Parameter를 기준으로 하여 그래프로서 비교한 것이다. 제안 기법은 기존 기법과 비교하여 차이가 근소하나, RSA 기반 서명 후 암호화 기법은 비트의 크기가 증가할수록 지수적으로 수행 시간이 증가하여 변형 기법이나 기존 signcryption 기법과는 많은 차이가 남을 알 수 있다.

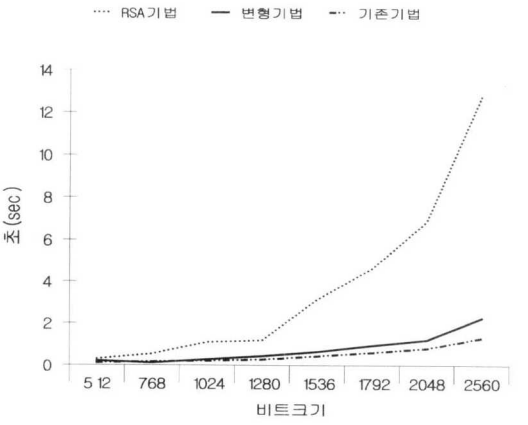


그림 1. Security Parameter의 크기에 따른 수행시간

통신 오버헤드 측면에서는 표 1에서 보이는 것과 같이, 기존 기법과 거의 같으며, RSA 기반 서명 후 암호화 기법보다는 훨씬 적은 통신 오버헤드를 가진다.

Ⅳ. 제안 기법의 변형 방안

본 절에서는 제안 기법의 응용 방안으로써 직접 서명 검증 가능한 signcryption 기법과 다수의 수신자에 대해 signcryption을 실시할 수 있는 multi-signcryption 방안, 그리고 공개키에 대한 유효성을 검증하기 위해 사용되는 CRLs(Certificate Revocation Lists) 처리를 해결한 서버 지원 서명된 변형 signcyrption 프로토콜을 제안한다.

1. 직접 검증 가능한 signcryption 기법

F. Bao 등은 기존 signcryption 기법을 응용하여 그림 2와 같이 수신자 B의 비밀키 없이 제 3자가 직접 서명 검증할 수 있는 기법을 제안하였다[8]. 본 논문에서 제안하는 기법 역시 Ⅱ절의 식 (1)과 (2)를 약간 수정하여 수신자 B의 비밀키 없이 직접 서명 검증할 수 있도록 구성할 수 있다.

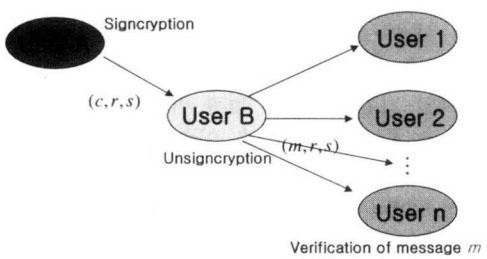


그림 2. 직접 검증 가능한 Signcryption 기법

먼저, 식 (1)에서, 세션키 k를 두 부분으로 나누어 다음과 같이 계산한다:

$$k_1 = hash(y_B^{x \cdot KA_r} \bmod n_A)$$
$$k_2 = hash(g_A^{x \cdot KA_r} \bmod n_A)$$

수신자 B는 각 세션키 k_1, k_2를 다음과 같이 계산한다.

$$t_1 = y_A \cdot g_A^{(r+KA_r) \cdot s} \bmod n_A, \quad t_2 = t_1^{x_B} \bmod n_A$$
$$k_2 = hash(t_1), k_1 = hash(t_2), \text{ and } m = D_{k_1}(c)$$

여기서 수신자 B가 제 3자에게 (m, r, s)를 전송한다고 가정할 경우, 그 제 3자는 메시지 m이 송신자 A에 의해 구성되었으며, 서명되었다는 것을 다음과 같이 검증함으로써 확신할 수 있다:

$$k_2 = hash(y_A \cdot g_A^{(r+KA_r) \cdot s} \bmod n_A)$$
$$r = KH_{k_2}(m)$$

앞서 설명된 세션키 k를 암호화 키 k_1과 서명키 k_2로 분리할 수 있기 때문에 수신자 B의 비밀키는 검증 시 더 이상 사용되지 않는다. 이 경우 수신측 B에 있어서의 계산 비용 측면에서는 signcryption 시 한 번의 모듈러 멱승 연산이 추가된다. 그러나, 이 경우에도 기존의 서명 후 암호화 기법들보다는 효율적이며, unsigncryption 절차에서는 계산 비용이 3절에서 제안된 방식과 동일하다.

2. Multi-signcryption 방안

지금까지 단지 한 명의 수신자에 대한 signcryption을 실시하는 방안에 대해 논의 하였으나, 안전하고 인증된 채널을 통해 다수의 수신자가 동일한 메시지를 수신할 수 있는 방안 역시 실제적으로 중요한 부분이므로, 본 절에서는 다수의 수신자에 대해 signcryption을 실시하는 multi-signcryption 방안을 제시한다.

먼저, 가정하는 것은 t명의 수신자 $R_1, R_2, \cdots, R_t$가 존재하고, 각 수신자 R_i의 unsigncryption 키(ElGamal 비밀키)는 각각 독립적으로 임의로 선택된 x_i를 가지고 있으며, 또한 이에 대한 공개키는 $y_i = g_A^{x_i} \bmod n_A$이다.

multi-signcryption 절차는 다음의 단계를 따른다.

단계 1. 임의의 k를 선택하고, 전송할 메시지 m에 대해 다음을 수행한다.

$$h = KH_k(m), \quad c = E_k(m \| h)$$

단계 2. 각 수신자 $i = 1, \cdots, t$에 대하여 3절에서 제안한 방식과 유사하게 세션키에 대한 signcryption을 다음과 같이 수행한다.

1) 임의의 수 $v_i \in [1, \cdots, \lambda - 1]$을 선택하고, 다음을 계산한다.

$$k_i = hash(y_i^{v_i \cdot KA_r} \bmod n_A), \quad k_i = k_{i,1} \| k_{i,2}$$

2) $c_i = E_{k_{i,1}}(k)$

3) $r_i = KH_{k_{i,2}}(m||h)$

4) $s_i = (v_i - x_A \cdot KA_d)/(KA_d \cdot r_i + 1) \bmod \lambda$

단계 3. 모든 수신자에게 전송할 메시지 $(c, c_1, r_1, s_1, \cdots, c_t, r_t, s_t)$를 생성하여 전송한다.

$(c, c_1, r_1, s_1, \cdots, c_t, r_t, s_t)$를 전송받은 각 수신자는 다음과 같이 unsigncryption 절차를 수행한다.

단계 4. $(c, c_1, r_1, s_1, \cdots, c_t, r_t, s_t)$로부터 (c, c_i, r_i, s_i)를 선택한다.

단계 5. 다음과 같이 k_i를 계산한다.

$k_i = hash((y_A \cdot g_A^{(r_i + KA_u) \cdot s_i})^{x_i} \bmod n_A), \quad k_i = k_{i,1}||k_{i,2}$

단계 6. 암호화된 c_i를 다음과 같이 복호화한다.

$k = D_{k_{i,1}}(c_i)$

단계 7. 암호화된 c를 다음과 같이 복호화한다.

$w = D_k(c), \quad w = m||h$

단계 8. $KH_k(m)$의 값을 계산하여 h와 같은지를 검사하고, 또한 $KH_{k_{i,2}}(w)$의 값을 계산하여 r_i와 같은지를 검사한다.

$KH_k(m) = h, \quad KH_{k_{i,2}}(w) = r_i$

단계 8이 만족될 경우에만 메시지 m을 송신자 A가 전송한 유효한 메시지로 받아들인다.

본 절에서 제안하는 multi-signcryption 방안 역시 2절에서 제안된 signcryption 방안과 같은 특성을 지니고 있으며, 기존의 signcryption에서 제공하는 위조 방지, 부인 방지 및 비밀성을 제공한다. 이에 추가하여, 다수의 수신자가 동일한 메시지를 수신하는 것을 보장하는 측면에서 악의적인 송신자가 특정 수신자를 제외시키는 것을 방지할 수 있으며, 이는 단계 1과 단계 2의 3), 4)에 의해 효과적으로 보장될 수 있다.

3. 서버 지원 서명된 응용 기법

제안된 기법에서, 모듈러 값 n_X (X는 사용자, 예를 들어 A, B, C를 의미)가 사용자마다 서로 다르기 때문에 수신자의 공개키(unsigncryption key) 역시 사용자마다 다르다. 이는 표 4의 값들을 포함하는 공개 디렉토리 서버를 도입함으로써 해결할 수 있다.

signcryption 기법뿐만 아니라 공개키를 사용하는 모든 프로토콜에서의 선결과제는 사용되는 공개키에 대한 유효성을 검증하기 위한 CRLs 처리 문제이다. N. Asokan 등은 서명 서버(signature server)를 도입하고 사용자의 비밀키에 대한 해쉬체인을 구성하여 송신 및 수신 부인 방지를 제공하는 방안을 제안하였다[9]. 이를 응용하여 본 제안 방식에서 도입되는 공개 디렉토리 서버를 이용하여 앞서 제시된 문제점을 해결하면서, 수신측에서 CRL 질의를 수행할 필요가 없을 뿐만 아니라, 또한 송신자에 대한 부인 방지 서비스를 제공할 수 있는 signcryption 시스템을 제안한다.

표 4. 공개 디렉토리 서버

signcryption / unsigncryption	사용자 A	사용자 B	사용자 C
사용자 A	·	$g_B^{x_A} \bmod n_B$	$g_C^{x_A} \bmod n_C$
사용자 B	$g_A^{x_B} \bmod n_A$	·	$g_C^{x_B} \bmod n_C$
사용자 C	$g_A^{x_C} \bmod n_A$	$g_B^{x_C} \bmod n_B$	·

제안하기 앞서, 가정해야 할 것은 인증기관(certificate authority)에 키를 등록하는 절차는 안전하게 이루어진다는 것이며, 또한 사용되는 공개 디렉토리 서버(이하 DS로 표기)는 인증기관에 의해 신뢰되며, 자신의 공개키, 비밀키 쌍 (PK_{ds}, SK_{ds})을 가지고 있고, 서명된 메시지에 대해 누구나 검증가능하고 메시지의 내용을 읽을 수 있는 그러한 서명을 생성할 수 있다는 것을 가정한다.

먼저 사용자 A는 자신의 ElGamal 비밀키 x_A를 선택하고, ElGamal 공개키 $y_A = g_A^{x_A} \bmod n_A$를 생성한다. 또한 x_A에 일 방향 해쉬 함수를 반복적으로 적용하여 다음과 같이 해쉬체인 $x_A^0, x_A^1, \cdots, x_A^i, \cdots, x_A^n$을 생성한다:

$x_A^0 = x_A, \quad x_A^i = hash^i(x_A) = hash(x_A^{i-1})$

그리고, 해쉬 함수를 n 번 적용시킨 x_A^n을 최상위 공개키(root public key)라 하고, 이를 인증기관(certificate authority)에 등록하고 다음의 절차를 수행한다.

단계 1. 사용자 A는 DS에게 사용자 B의 공개키를 요구하는 메시지 (A, i, x_A^i, B)를 전송한다. 여기서, A와 B는 사용자 A, B의 식별자를 의미하며, x_A^i는

i번째 signcryption 수행 시에 사용되는 RSA 공개키가 된다. i의 초기값은 n이며, 사용 시마다 1씩 감소하다가, 0이 되면 사용자 A는 새로운 x_A'를 생성하여 해쉬체인을 구성한다.

단계 2. 요구 메시지 (A, i, x_A^i, B)를 수신한 DS는 x_A^i를 $n-i$ 만큼 해쉬 함수를 반복적으로 적용한 값과 인증기관에 의해 제공되는 사용자 A의 최상위 공개키를 비교하여 사용자 A의 i번째 공개키임을 식 (4)와 같이 검증한다. 만일 현재 세션이 i라고 할 경우 i보다 작거나 클 경우, 또는 0일 경우의 요구 메시지에 대해서는 거부 메시지를 사용자 A에게 전송한다.

$$hash^{n-i}(x_A^i) = x_A^n \tag{4}$$

검증이 성공하면 DS는 사용자 A의 i번째 공개키 x_A^i가 사용되었다는 사실을 저장하고 $(i = i-1)$, 다음과 같이 B의 공개키 $y_B = g_A^{x_B} \bmod n_A$를 포함하는 메시지에 서명하고 사용자 A에게 전송한다:

$$[A, i, x_A^i, y_B]SK_{ds} \tag{5}$$

여기서 $[\cdot]SK_{ds}$는 메시지 $[\cdot]$를 DS 자신의 비밀키로 서명한 것을 의미한다.

단계 3. (5)를 수신한 사용자 A는 DS의 공개키로 수신된 메시지를 검증하고, 또한 검증된 i번째 공개키 x_A^i를 이용하여 다음 식을 만족하는 i번째 signcryption 키(RSA 비밀키) s_A^i를 생성한다:

$$x_A^i \cdot s_A^i \equiv 1 \bmod \lambda$$

이 RSA 비밀키 s_A^i와 ElGamal 비밀키 x_A을 이용하여 Ⅱ절에서 제안한 기법 또는 Ⅳ.1절의 직접 검증 가능한 기법과 같이 signcryption 절차를 수행하여 c, r, s를 생성하고, 다음과 같은 NRO (Non-Repudiation of Origin) 토큰을 생성, 추가하여 사용자 B에게 전송한다:

$$NRO = ([A, i, x_A^i, y_B]SK_{ds}, x_A^{i-1})$$
$$(c, r, s, NRO) \text{전송}$$

단계 4. 사용자 B는 NRO 토큰에 포함된 파라메터에 의해 사용자 A의 i번째 공개키의 유효성 여부를 검사하며, 또한 송신자가 사용자 A라는 것을 x_A^{i-1}에 해쉬 함수를 적용하여 x_A^i와 비교함으로써 검증한다. 검증 절차가 성공적이면, 수신 받은 c, r, s, x_A^i를 이용하여 Ⅱ절 또는 Ⅳ.1절의 unsigncryption 절차를 수행한다.

본 절에서 기술된 서버 지원 서명된 변형 기법의 수행 과정을 도식화하면 그림 3과 같다.

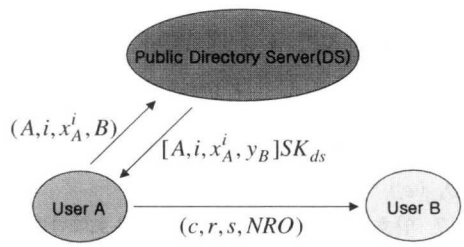


그림 3. 서버 지원 서명된 변형 signcryption

본 절에서 제안한 시스템에서 사용자 B는 사용자 A의 공개키들에 대한 CRL 질의를 할 필요가 없는데, 이는 DS와 사용자 A간의 프로토콜에서 해결해 주고 있기 때문이다. 단지 사용자 B는 DS의 서명에 대한 검증만을 수행하면 된다. 또한 매번 signcryption 절차를 수행할 때마다 signcryption 키가 바뀌며, 사용된 키는 다시 사용하지 않기 때문에 더욱 높은 수준의 안전성을 기대할 수 있다. 그리고 송신측의 송신 부인이 발생할 경우 사용자 B는 NRO 토큰을 중재자(또는 인증기관)에게 제출함으로써 송신 부인을 방지할 수 있다는 장점이 있다.

Ⅴ. 기존 signcryption 기법에 대한 서버 지원 서명 응용 방안

Ⅳ.3절에서 제안 signcryption 기법에 대한 서버 지원 서명된 응용 방안을 제시하여, 보다 높은 안전성을 제공하면서, 수신측의 CRLs 질의에 대한 오버헤드를 제거할 수 있고, 또한 송신 부인 방지 서비스를 제공할 수 있도록 하였다. 이러한 기법은 또한 기존의 signcryption 기법에 대해서도 적용 가능하다. 따라서 본 절에서는 기존 기법에 대한 서버 지원 서명 방안을 제시한다.

제안하기 앞서, 가정해야 할 것은 Ⅳ.3절에서와 마찬가지로 인증기관(certificate authority)에 키를

등록하는 절차는 안전하게 이루어진다는 것이며, 또한 사용되는 공개 디렉토리 서버(이하 DS로 표기)는 인증기관에 의해 신뢰되며, 자신의 공개키, 비밀키 쌍 (PK_{ds}, SK_{ds})을 가지고 있고, 서명된 메시지에 대해 누구나 검증가능하고 메시지의 내용을 읽을 수 있는 그러한 서명을 생성할 수 있다는 것을 가정한다. 먼저 사용자 A는 자신의 ElGamal 비밀키 $x_a \in [1, \cdots, q-1]$를 선택하고, 이 비밀키를 인증기관에 등록한다. 등록된 비밀키는 안전한 채널을 통해 DS로 제공되어질 수 있다는 것을 가정한다.

사용자 A는 비밀키 x_a를 일 방향 해쉬 함수를 반복적으로 적용하여 다음과 같이 signcryption시 비밀키로 사용되어질 키 해쉬체인 $x_a^0, x_a^1, \cdots, x_a^i, \cdots, x_a^n$을 생성한다:

$$x_a^0 = x_a, \quad x_a^i = hash^i(x_a) = hash(x_a^{i-1}) \tag{6}$$

또한, 인증 해쉬체인 $X^0, X^1, \cdots, X^i, \cdots, X^n$을 다음과 같이 생성한다:

$$\begin{aligned} X &= hash(A \| x_a) \\ X^0 &= X, \quad X^i = hash(X^{i-1} \| hash(x_a^{i-1})) \end{aligned} \tag{7}$$

여기서, A는 사용자 A의 식별자를 의미한다.

이렇게 두 해쉬체인을 구성한 후 다음의 단계를 수행한다.

단계 1. 사용자 A는 DS에게 자신의 i번째 signcryption시에 사용되는 공개키를 요구하는 메시지 (A, i, X^i)를 전송한다. 여기서, A는 사용자 A의 식별자를 의미하며, X^i는 DS가 일 방향 해쉬함수의 특성을 이용하여 사용자 A임을 검증할 수 있는 challenge 값이다. i의 초기값은 n이며, 사용 시마다 1씩 감소하다가, $i \leq 1$이 되면 사용자 A는 새로운 X'를 생성하여 해쉬체인을 구성한다.

단계 2. 요구 메시지 (A, i, X^i)를 수신한 DS는 X^i에 해쉬 함수를 반복적으로 적용한 값과 인증기관에 의해 제공되는 사용자 A의 비밀키에 일 방향 해쉬함수를 i번 반복적으로 적용한 값과 비교하여 사용자 A임을 식 (8)과 같이 검증한다. 이 때, 계산의 용이성을 위해 DS 역시 사용자 A와 마찬가지로 초기에 비밀키 x_a를 이용하여 식 (6), (7)과 같이 해쉬체인을 구성할 수 있다. 만일 현재 세션이 i라고 할 경우 i보다 작거나 를 경우, 또는 $i \leq 1$일 경우의 요구 메시지에 대해서는 거부 메시지를 사용자 A에게 전송한다.

$$hash(X^{i-1} \| hash(x_a^{i-1})) = X^i \tag{8}$$

검증이 성공하면 DS는 사용자 A의 X^i가 사용되었다는 사실을 저장하고 $(i = i-1)$, 다음과 같이 사용자 A의 공개키 $y_a^i = g^{x_a^i} \bmod p$를 포함하는 메시지에 서명하고 사용자 A에게 전송한다:

$$[A, i, X^i, y_a^i] SK_{ds} \tag{9}$$

여기서 $[\cdot] SK_{ds}$는 메시지 $[\cdot]$를 DS 자신의 비밀키로 서명한 것을 의미한다.

단계 3. (9)를 수신한 사용자 A는 DS의 공개키로 수신된 메시지를 검증을 실시한다. 여기서, DS에 의해 서명된 메시지 내에 포함된 공개키 $y_a^i = g^{x_a^i} \bmod p$에 대응되는 해쉬체인상의 키 x_a^i가 현 세션에서의 signcryption 키(비밀키)가 된다.

이 signcryption 키 x_a^i을 이용하여 Zheng에 의해 제안된 signcryption 절차를 수행하여 c, r, s를 생성하고, 다음과 같은 NRO(Non-Repudiation of Origin) 토큰을 생성, 추가하여 사용자 B에게 전송한다:

$$\begin{aligned} &NRO = ([A, i, X^i, y_a^i] SK_{ds}, X^{i-1}) \\ &(c, r, s, NRO) \text{ 전송} \end{aligned}$$

단계 4. 사용자 B는 NRO 토큰에 포함된 파라메터에 의해 사용자 A의 공개키 y_a^i의 유효성 여부를 검사(DS의 서명 검증)하며, 또한 송신자가 사용자 A라는 것을 X^{i-1}에 해쉬 함수를 적용하여 X^i와 비교함으로써 검증한다. 검증 절차가 성공적이면, 수신 받은 c, r, s, y_a^i를 이용하여 unsigncryption 절차를 수행한다.

본 절에서 제안되는 서비 지원 서명된 기존 signcryption 기법의 수행 과정을 도식화하면 그림 4와 같다.

기존 signcryption 기법은 계산 비용 측면에서 매우 효율적이나, forward secrecy를 제공하지 못하는 단점이 존재한다. 이에 대해 본 절에서 제안하는 방안과 같이 signcryption 키를 매 세션마다 서로 다

른 키를 사용하고, 또한 사용된 키는 다시 사용하지 않음으로써 한 단계 높은 안전성을 제공하여 forward secrecy를 제공하지 못하는 단점을 최소화 하는 방안이 하나의 대책이 될 수 있다. 또한 이러한 방안은 기존 signcryption 기법의 효율성을 보장할 뿐만 아니라 4.3절에서 제시된 제안 signcryption 기법의 서버 지원 서명 응용 방안의 특징과 마찬가지로 수신측에서는 송신측의 공개키에 대한 CRLs 질의를 할 필요가 없으며, 송신 부인 방지서비스를 제공하는 장점을 제공해 준다.

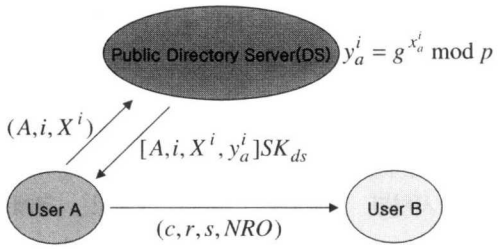


그림 4. 서버 지원 서명된 기존 signcryption

Ⅵ. 결 론

본 논문에서는 안전성과 비용을 향상시킨 변형 signcryption 기법을 제안하였다. 제안 기법은 forward secrecy를 제공하기 위해 RSA 방식과 ElGamal 방식을 통합한 형태를 취하고 있으며, 특히 송신자의 측면에서 볼 때 제안된 변형 기법은 기존의 다른 signcryption 기법들 보다 더욱 작은 비용을 가지고 있다. 계산량에서는 기존 signcryption 기법과 마찬가지로 총 3번의 모듈러 멱승 연산을 수행하며, RSA 기반 서명 후 암호화 기법보다 훨씬 적다. 통신 오버헤드 측면에서는 기존 기법과 같은 오버헤드를, 그리고 RSA 기반 서명 후 암호화 기법보다 훨씬 적게 가진다. 또한 본 논문에서는 signcrypt된 메시지에 대해 제 3자가 unsigncrytion측의 비밀키(unsigncryption key)를 사용하지 않고서도 메시지에 대한 검증을 수행할 수 있는 직접 검증 가능한 signcryption 기법을 제안하였고, 다수의 수신자에 대한 multi-signcryption 기법을 제시하였다. 또한 현재 공개키를 사용하는 모든 프로토콜에서의 선결과제인 CRL 처리문제를 해결하고, 보다 더욱 높은 수준의 안전성과 송신자 부인 방지 서비스를 제공할 수 있는 서버 지원 서명을 지닌 응용 방안을 제시하였다. 서버 지원 서명에 대한 응용 방안 역시 기존 signcryption 기법에 적용될 수 있다.

제안된 변형 signcryption 방안은 PDA, 이동 장비나 스마트 카드와 같이 컴퓨팅 능력이 떨어지는 light-weight 장치 등에서 암호화 및 인증의 용도로 활용될 수 있다. 특히 제안된 기법의 효율성과 안전성은 자동차 네트워크와 같은 특수 네트워크(Ad-hoc Network)를 기반으로 한 지능형 교통 시스템이나 WAP, 디지털 캐쉬 지불 시스템 등에 다양하게 적용될 수 있을 것으로 기대된다.

참 고 문 헌

[1] Y. Zheng, "Digital signcryption or how to achieve cost(signature and encryption)<<cost(signature) + cost(encryption)", *Advances in Cryptology, Proceedings of CRYPTO'97, LNCS Vol. 1294, Springer-Verlag*, pp. 165-179, 1997.

[2] Y. Zheng, "Signcryption and its application in efficient public key solutions", *Proc. of Information Security Workshop(ISW'97), LNCS Vol. 1396, Springer-Verlag*, pp. 291-312, 1998.

[3] Y. Zheng, "Compact and unforgeable session key establishment over ATM network", *In Proceedings of IEEE INFOCOM'98*, pp. 411-418, San Francisco, 1998.

[4] R. L. Rivest, A. Shamir and L. Adleman, "A method of obtaining digital signature and public key cryptosystem", ACM Communication, 21 No. 2, pp. 120-126, 1978.

[5] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithm", IEEE Trans. on Information Theory IT-31, pp. 469-472, 1995.

[6] "Proposed Federal Information Proceeding Standard for Digital Signature Standard(DSS)", Federal Register, Vol. 56, No.169 30, 1991.

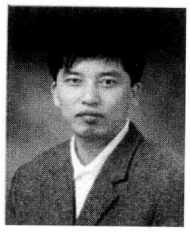
[7] A. J. Menezes P. C. van Oorschot and S. A. Vanstone, "Handbook of Applied Cryptography" 1997.

[8] F. Bao and H. Deng, "A signcryption scheme with signature directly verifiable by public key", Proceeding of Public Key Cryptography (PKC'98), LNCS Vol.1431, pp.55-59, 1998.

[9] N. Asokan, G. Tsudik and M. Waider,

"Server-Supported Signatures", Journal of Computer Security, Vol.5, No. 1, November 1997.
[10] RSA Security Inc. "http:\\www.rsa.com".
[11] 이민섭, "현대 암호학", 교우사, 2001.

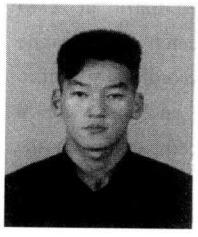
이 경 현(Kyung-hyune Rhee) 정회원



1982년 2월 : 경북대학교 수학교육과 졸업
1985년 2월 : 한국과학기술원 응용수학과 석사
1992년 8월 : 한국과학기술원 수학과 박사

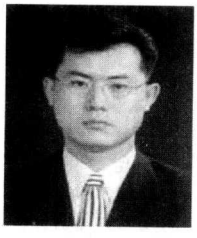
1985년 2월~1993년 2월 : 한국전자통신연구소 연구원, 선임연구원
1993년 3월~현재 : 부경대학교 전자컴퓨터정보통신공학부 전임, 조교수, 부교수
<주관심 분야> 암호이론, 암호프로토콜, 네트워크보안, 이동네트워크, 그룹키 관리

조 현 호(Hyun-ho Cho)



1997년 2월 : 부경대학교 전자계산학과 졸업
2002년 2월 : 부경대학교 전자계산학과 석사
2002년 3월~현재 : 동부산대학 컴퓨터정보학부 정보보안 안전공전임강사
<주관심 분야> Signcryption, 타원 곡선 암호, 셀룰라 오토마타

이 준 석(Jun-seok Lee)



1995년 2월 : 동의대학교 전자통신공학과 졸업
1998년 2월 : 동의대학교 전자공학과 석사
2001년 2월 : 부경대학교 전자계산학과 박사 수료
<주관심 분야> 셀룰라 오토마타, 정보보호, 암호이론, 부호이론