情報保護學會論文誌
第11券 第5號, 2001. 10

Forward Secrecy를 제공하는 Signcryption 기법들

정 희 윤*, 이 동 훈**, 임 종 인**

Signcryption Schemes with Forward Secrecy

Hee Yun Jung*, Dong Hoon Lee**, Jong In Lim**

요 약

Y. Zheng은 기존의 서명 후 암호화(signature-then-encryption)기법을 기반으로하여 디지털 서명 기법과 대칭키 암호 기법을 결합시킨 signcryption이라는 새로운 기법을 제안하였다[8]. Signcryption은 기밀성과 인증성을 동시에 만족시키면서 계산량과 통신비용을 줄여 효율성을 높인 방법이다. 또한 C. Gamage등은 대리 서명 기법과 signcryption 기법을 효율적으로 결합시킨 proxy-signcryption 기법을 제안 하였다[2]. 그러나 이 두 기법에서 송신자의 개인키가 드러날 경우 그 개인키를 알게되는 사람은 과거에 송신자가 생성한 signcrypt된 문서를 복호화 할 수 있게된다. 즉, 기존의 signcryption 기법은 송신자의 개인키에 대한 forward secrecy의 성질을 제공하지 못한다. 본 논문에서는 Zheng의 기법을 변형하여 forward secrecy를 제공하는 signcryption기법과 proxy-signcryption을 제안한다.

ABSTRACT

Y. Zheng introduced a new type of cryptograghic primitive as "signcryption", which combines a function of digital signature scheme with a symmetric key encryption algorithm. Signcryption doesn't only provide authenticity and confidentiality in a single step, but also give more efficient computation than the traditional "signature-then-encryption". And C. Gamage proposed a proxy-signcryption that efficiently combines a proxy signature with the signcryption. But, in the proposed signcryption schemes, one who obtains the sender's private key can recover the original message of a signcrypted text. That is, forward secrecy is not offered by the signcryption scheme with respect to the sender's private key. In this paper, we will propose a modified signcryption of Zheng's signcryption and a variant of proxy-signcryption with forward secrecy.

keyword : *signcryption, proxy-signcryption, forward secrecy*

Ⅰ. 서 론

통신상에서 다른 사람에게 문서를 전송할 때, 수신자가 송신자의 신원을 확인할 수 있는 인증성을 만족하기 위하여 디지털 서명기법을 사용하고, 문서를 비밀리에 안전하게 송신하기 위해서는 암호화 과정을 거쳐서 문서를 전송한다. 이러한 두가지 인증성과 기밀성을 동시에 만족하기 위해서 문서에 서명 단계를 거친 후에 그것을 다시 암호화하는 "서명 후 암호화 기법(Signature-Then-Encryption)"을 사용하였다. Y. Zheng은 서명 후 암호화 기법을 변형하여 signcryption 이라는 새로운 기법을 소개하였다. 이 기법은 디지털 서명과 암호화 기법이 제공하는 성질인 기밀성과 인증성을 동시에 제공하면서

* 고려대학교 정보보호기술 연구센터(CIST)(hyun@cist.korea.ac.kr)
** 고려대학교 정보보호기술 연구센터(CIST)({donghlee, jilim}@tiger.korea.ac.kr)

기존의 서명 후 암호화 기법보다 계산량과 통신비용을 줄인 효율적인 기법이다[8]. 그러나 Zheng의 기법에서는 기존의 서명후 암호화 기법에서 제공되는 송신자에 대한 forward secrecy의 성질을 만족하지 못한다. 즉, signcrypt된 문서를 생성하는 송신자가 자신의 개인키를 사고로 분실하였거나, 공격자가 키 저장장소에 침입함으로써 송신자의 개인키가 드러나게 되는 경우에 문제가 발생하게 된다. 송신자의 개인키를 알게되는 사람은 이전에 송신자가 이 키를 사용하여 생성했던 signcrypt된 문서로부터 원본 문서를 복구해낼 수 있기 때문이다. Forward secrecy의 정의는 다음과 같다.[9]

<정의1 : forward secrecy>

Long term 키의 분실이 long term 키의 분실 이전에 생성된 프로토콜 P의 세션키의 분실을 의미하지 않는다면 프로토콜 P는 *"forward secrecy"*를 제공한다고 말한다.

암호 프로토콜에서 forward secrecy의 성질은 중요하다. Long term key(-프로토콜의 세션키를 생성하기 위해 사용되는 키로 사용기간이 세션키 보다긴 상위계층의 키이다.-)의 의도하지 않은 공개가 그것으로 생성된 프로토콜의 세션키의 공개를 의미한다면 매우 치명적인 약점이다. Signcryption 기법은 기밀성과 인증성을 동시에 제공하므로 키 공유 기법(key agreement)에 사용될 수 있으며, 실제로 Y. Zheng은 signcryption 기법을 키 전송(key transport)기법과 키 교환(key exchange) 기법에 적용하였다[10]. 이러한 기법 안에서의 forward secrecy는 더욱 중요한 사항이다. 본 논문에서는 signcryption 기법의 forward secrecy를 송신자의 개인키(Ⅱ장에서는 A의 개인키 x_a, Ⅲ장에서는 대리서명키 x_{ap})에 한정하여 고려한다. 즉, 송신자의 개인키가 의도하지 않게 공개됐을 때, 이전에 생성했던 signcryption의 세션키가 누출(제 3자의 공격에 의한 세션키의 공개 등)되면 그때 사용한 signcryption 기법은 forward secrecy를 제공하지 않는다고 한다.

본 논문에서는 위의 문제점을 보완하는 두가지 변형된 signcryption 기법을 제안한다. 첫 번째 방법은 Zheng의 기법보다 모듈로 지수연산을 한번 더 요구하지만 기존의 서명 후 암호화 기법보다 효율적이며, 두 번째 방법은 proxy-signcryption에 적용하여 지수 연산의 추가 없이 forward secrecy 성질을 갖는다. 또한 두 번째 방법은 기존 proxy-signcryption 기법[2]에서의 서명 의뢰인이 대리서명자로 위장하여 대리 서명을 수행 할 수 있다는 단점을 보완하여 대리 서명자를 보호 할 수 있는 기법으로 바꾸어 준다.

Ⅱ. Signcryption

본 장에서는 Zheng이 제안한 signcryption 기법과 signcryption을 구성하기 위한 서명 기법들을 살펴본다[8].

2.1 SDSS1과 SDSS2

Zheng의 signcryption 기법에 사용되는 서명 기법은 DSS(Digital Signature Standard)를 변형하여 서명의 길이를 줄인 SDSS(shortened DSS)이다.

공개변수

p : 큰 소수 ($2^{511+64t}<p<2^{512+64t}$, $0\le t\le 8$)

q : $p-1$을 나누는 큰 소수 ($2^{159}<q<2^{160}$)

g : 위수가 q인 Z_p^*의 원소

$hash$: 일방향 해쉬함수

$x_a \in_R Z_q^*$: A의 개인키

$y_a = g^{x_a} \bmod p$: A의 공개키

[표 1] SDSS

	m에 대한 서명(r, s)	서명 확인
SDSS1	$r = hash(g^x \bmod p, m)$ $s = x/(r + x_a) \bmod q$	$k = (y_a \cdot g^r)^s \bmod p$ $hash(k, m) \stackrel{?}{=} r$
SDSS2	$r = hash(g^x \bmod p, m)$ $s = x/(1 + x_a r) \bmod q$	$k = (g \cdot y_a^r)^s \bmod p$ $hash(k, m) \stackrel{?}{=} r$

2.2 Zheng의 signcryption

A는 signcryption 과정을 통해 문서 m을 signcrypt된 문서로 만들어 기밀성과 인증성을 제공하도록 하여 B에게 보내며, B는 unsigncryption 과정을 통하여 문서를 복구한다.

공개변수

p : 큰 소수

q : $p-1$을 나누는 큰 소수 ($q|(p-1)$)
g : 위수가 q인 Z_p^*의 원소
$hash$: 일방향 해쉬함수
KH : keyed-일방향 해쉬함수
(E, D) : 대칭키 암호 시스템의 암호화, 복호화 알고리즘
A와 B의 개인키, 공개키 쌍들은 2절과 동일하다.

〔표 2〕 Zheng의 Signcryption

A에 의한 Signcryption	⇒	B에 의한 Unsigncryption
$x \in_R [1, 2, \cdots, q-1]$ $k = hash(y_b^x \bmod p)$ $k_1 \| k_2 = k$, $c = E_{k_1}(m)$ $r = KH_{k_2}(m)$ $s = \frac{x}{(r+x_a)} \bmod q$	c, r, s	$k = hash((y_a g^r)^{sx_b} \bmod p)$ $k_1 \| k_2 = k$, $m = D_{k_1}(c)$ $KH_{k_2}(m) = r$을 만족하면 m을 받아들인다

이 기법은 Zheng이 제안한 signcryption기법[8]으로, SDSS(Shortened Digital Signature Standard)와 대칭키 암호를 결합시켜 구성하였다. 기존의 DSS(Digital Signature Standard)-Elgamal 암호 기법을 기반으로 한 서명 후 암호화 기법에서는 송신자의 서명 과정과 암호화 과정에서 3번, 수신자의 서명 확인 과정과 복호화 과정에서 3번의 모듈로 지수 연산을 요구하나, signcryption기법에서는 전체적으로 3번의 모듈로 지수 연산을 요구한다. 또한 통신량도 $2|q|+2|p|$에서 $|KH(\cdot)|+|q|$로 약 84%정도 절감 시켰다. ($|p|=512, |q|=144, |KH(\cdot)|=72$를 기준으로 할 때)

그러나 이 기법은 서명 후 암호화 기법에서 제공되는 forward secrecy를 제공하지 못한다.

Unsigncrytion의 키 계산 과정을 살펴보면 해쉬함수 안의 연산이 다음을 만족한다.

$$(y_a \cdot g^r)^{s \cdot x_b} \bmod p = (y_b^{x_a + r})^s \bmod p$$

정당한 수신자 B는 자신의 개인키 x_b를 사용하여 등식의 왼쪽수식을 사용하여 키를 계산할 수 있게된다. 그러나 A의 개인키 x_a가 공개될 경우 이 x_a를 알게되는 제 3자는 등식의 오른쪽 수식을 이용하여 x_b값을 알지 못하더라도 세션키 k_1을 계산할 수 있게되며, 따라서 signcrypt된 문서를 복구할 수 있게된다. 즉, 이 기법은 송신자의 개인키 x_a에 대한 forward secrecy를 제공하지 못한다.

본 논문에서는 이러한 단점을 보완하는 변형된 signcryption 기법을 소개한다.

2.3 Forward Secrecy를 만족하는 Signcryption

먼저 변형된 signcryption을 구성하는데 사용되는 MSDSS(Modified-SDSS)를 소개한다.

2.3.1 변형된 SDSS(MSDSS)

공개변수와 A, B의 키 쌍은 SDSS와 같고 r과 s의 계산은 SDSS1과 동일하다.

서명자는 $R = g^r \bmod p$를 계산하여 (m, R, s)를 메세지의 서명으로 한다. 수신자는 $k = (y_a \cdot R)^s \bmod p$를 계산하여 $R = g^{hash(k, m)} \bmod p$ 이 성립하는지 확인함으로써 서명을 검증한다.

2.3.2 변형된 signcryption

2.3.1의 MSDSS를 적용하여 구성한 변형된 signcryption 기법을 소개한다. 공개변수와 A, B의 키 쌍은 Zheng의 기법과 동일하다〔표 3〕.

Forward Secrecy 의 만족 :

2.2절의 Zheng의 기법에서 살펴본 방법과 같이 해쉬함수 안의 연산이 다음 식을 만족함을 알 수 있다.

$$(y_a \cdot R)^{s \cdot x_b} \bmod p = (y_b^{x_a + r})^s \bmod p$$

그러나 A의 개인키 x_a만을 알아낸 제 3자는 r을 알지 못하고는 키를 복구해 낼 수 없다. R로부터 r을 계산할 수 있는 사람은 $\log_g R$을 계산할 수 있는 사람이다. 따라서 이산대수 문제에 기반하여 forward secrecy를 보장할 수 있게 된다.

〔표 3〕 변형된 signcryption

A에 의한 Signcryption	⇒	B에 의한 Unsigncryption
$x \in_R [1, 2, \cdots, q-1]$ $k = hash(y_b^x \bmod p)$ $k_1 \| k_2 = k$ $c = E_{k_1}(m)$, $r = KH_{k_2}(m)$ $R = g^r \bmod p$ $s = \frac{x}{(r+x_a)} \bmod q$	c, R, s	$k = hash((y_a R)^{sx_b} \bmod p)$ $k_1 \| k_2 = k$, $m = D_{k_1}(c)$ $R = g^{KH_{k_2}(m)} \bmod p$을 만족하면 m을 받아들인다.

연산량 비교 : Zheng의 signcryption 기법[표 2]과 비교하면, 변형된 signcryption 기법은 $R=g^r \bmod p$와 $g^{KH_k(m)} \bmod p = R$을 계산하기 위해 지수 연산이 두 번 증가하고, 수신자가 행하는 해쉬함수 안의 연산에서 지수 연산이 한 번 감소하므로 결과적으로 Zheng의 기법보다 한 번의 지수연산이 증가한다.
연산량이 동일한 일반적인 두 서명 기법을 비교할 때, 서명자에 의한 서명은 단 한번 생성되고 서명 확인 과정은 필요에 의해 여러번 수행 될 수 있기 때문에, 서명 과정의 연산량 보다 서명 확인 과정의 연산량이 적은 기법이 더 효율적이다. 이러한 측면을 signcryption 기법에 적용하면, 송신자의 signcryption 과정은 한번 수행되고 수신자의 unsigncryption은 여러번 수행될 수 있다. 변형된 signcryption 기법을 Zheng의 signcryption 기법과 비교하면, signcryption 과정에서만 지수 연산이 한번 증가하게 되므로, 제안한 기법이 Zheng의 기법에 비해 효율성이 크게 떨어지지 않는다. 또한 변형된 기법은 기존의 서명 후 암호화 기법보다 효율적이며, 기존에 제안되었던 다른 signcryption 기법들에도 적용이 가능하다.
안전성에 대해서는 Ⅳ장에서 논의한다.

[표 4] 효율성 비교

	지수 연산 갯수	통신량	Forward Secrecy
DSS+ElGamal을 기반으로 하는 서명후 암호화 기법	6	$2\|q\|+2\|p\|$	○
Zheng의 signcryption	3	$\|KH(\cdot)\|+\|q\|$	×
변형된 signcryption	4	$\|q\|+\|p\|$	○

Ⅲ. Proxy-Signcryption

본 장에서는 대리 서명 방식을 이용한 signcryption 기법을 살펴보고 forward secrecy 성질을 제공하는 변형된 기법을 제시한다.

3.1 기존의 기법

3.1.1 SDSS를 적용한 대리 서명 기법

대리 서명(proxy signature)방식은 본인이 부재중에 지정한 대리인으로 하여금 자신을 대신하여 서명을 이행할 수 있도록 하는 방식이다.[4] 다음 서명

[표 5] 대리 서명 기법

A에 의한 대리 서명 키 생성	→	P에 의한 대리 서명키 확인
$x \in_R [1,2,\cdots,q-1]$ $K=g^x \bmod p$ $x_{ap}=x_a+xK \bmod q$	K, x_{ap}	$g^{x_{ap}} \stackrel{?}{=} (y_a \cdot K^K) \bmod p)$
P에 의한 서명 생성	⇒	B에 의한 대리 서명 확인
$x' \in_R [1,2,\cdots,q-1]$ $r'=hash(g^{x'} \bmod p, m)$ $s'=\frac{x'}{(r'+x_{ap})} \bmod q$	m, r', s', K	$y_{ap}=y_a \cdot K^K \bmod p$ $k=(y_{ap} \cdot g^{r'})^{s'} \bmod p$ $hash(k,m)=r'$을 만족하면 m을 받아들인다.

기법은 SDSS를 적용한 대리서명 기법으로 proxy signcryption을 구성하기 위한 중간 단계이다.

A는 대리 서명자 P에게 자신을 대신하여 서명을 하도록 위임하고, 대리 서명자는 메세지 m에 대한 서명을 생성하여 B에게 보낸다. 공개변수와 키 쌍들은 앞의 기법에서와 동일하며, →은 안전한 채널(secure channel)을 의미한다[표 5].

프로토콜의 전체적인 구성은 다음과 같다.

A는 대리 서명키 x_{ap}를 생성하여 대리 서명자 P에게 은밀하게 전달하고, P는 대리 서명키를 확인하여 문서 m에 대한 서명을 생성하여 B에게 보낸다. B는 대리 서명을 확인하고 문서를 받아들인다.

대리 서명 기법은 다음의 사항을 만족해야한다.[4]

- 위조 불가능성(Unforgeability) : 서명 위임자와 지정된 대리 서명자 이외의 사람은 정당한 대리 서명을 생성할 수 없다.
- 대리 서명자의 위반(Proxy signer's deviation) 불가능성 : 대리 서명자는 서명 위임자의 서명권을 위임 받지 않고서는 대리 서명을 수행할 수 없어야 한다.
- 서명 위임자의 위반(Original signer's deviation) 불가능성 : 서명 위임자는 대리 서명자의 참여없이 대리 서명을 생성할 수 없어야 한다.
- 서명 구별 가능성(Verifiability) : 정당한 대리 서명은 일반 서명과 구별 될 수 있어야 한다.
- 대리 서명자의 신원 확인(Distinguishability) : 서명 위임자는 대리 서명으로부터 대리 서명자의 신원을 확인할 수 있어야 한다.
- 부인 불가능성(Undeniability) : 대리 서명자가 서명 위임자에 대한 대리 서명을 생성한 후에는

서명 사실을 부인할 수 없어야 한다.

위의 서명 기법은 서명 위임자의 위반 불가능성을 만족하지 못한다. P의 대리 서명과정에서 P에 관한 어떠한 비밀 정보도 요구되지 않으므로 P의 수행과정을 A가 대신할 수 있다(대리 서명자 비 보호). 또한 B는 전송 받은 서명으로부터 서명 위임자 A를 분명히 확인 할 수 있지만 대리 서명자 P의 신원은 확인할 수 없다. 이러한 단점은 다음절에 설명되는 대리 signcryption 기법에서도 살펴볼 수 있다. 정직하지 않은 서명 위임자로부터 대리서명자를 보호할 수 있는 기법을 2절에서 제안한다.

3.1.2 Proxy-Signcryption

본 절에서는 C.Gamage, J. Leiwo, Y.Zheng이 제안한 proxy signcryption 기법[2]을 설명한다.

Proxy-Signcryption은 대리 서명 방식을 signcryption 기법에 적용한 것으로 서명자가 지정한 대리인으로 하여금 정당한 signcryption을 수행할 수 있게 한다.

A는 대리 서명자 P에게 자신의 서명 능력을 위임하고, 대리 서명자 P는 signcryption을 실행 한 후에 B에게 보낸다. 공개변수와 A, B의 키 쌍들은 Zheng의 signcryption 기법과 동일하다.

〔표 6〕 C.Gamage의 Proxy-Signcryption

A에 의한 대리 서명 키 생성	→	P에 의한 대리 서명키 확인
$x\in_R[1,2,\cdots,q-1]$ $K=g^x \bmod p$ $x_{ap}=x_a+xK \bmod q$	K, x_{ap}	$g^{x_{ap}} \stackrel{?}{=} (y_a \cdot K^K) \bmod p$
P에 의한 proxy-signcryption	⇒	B에 의한 unsigncryption
$x'\in_R[1,2,\cdots,q-1]$ $k=y_b^{x'} \bmod p$ $k_1 \| k_2 = k$ $c=E_{k_1}(m)$ $r'=KH_{k_2}(m)$ $s'=\frac{x'}{(r'+x_{ap})} \bmod q$	c, r', s', K	$y_{ap}=y_a \cdot K^K \bmod p$ $k=(y_{ap}g^{r'})^{s'x_b} \bmod p$ $k_1 \| k_2 = k$ $m=D_{k_1}(c)$ $KH_{k_2}(m)=r'$을 만족하면 m을 받아들인다.

II. 2 절에서 언급한 방법과 유사하게 unsigncrytion 과정에서

$$(y_{ap} \cdot g^{r'})^{s' \cdot x_b} \bmod p = (y_b^{x_{ap}+r'})^{s'} \bmod p$$

이 성립하므로 대리 서명키 x_{ap}가 드러날 경우 x_b값을 모르는 B 이외의 다른 사람도 키 k를 계산할 수 있게 되며, 따라서 signcrypt된 문서를 복구할 수 있게 된다. 즉, 이 기법은 대리 서명키인 x_{ap}에 대한 forward secrecy를 제공하지 못한다. x_{ap}를 소유하는 사람은 A와 P 두 사람이므로 참여자들의 부주의에 의해 노출될 확률이 특정인의 개인키보다 높다.

또한 이 기법은 대리 서명 위임자가 대리 서명자로 위장하여 대리 서명자가 서명을 수행하는 것처럼 보이게 하는 위조 공격으로부터 대리 서명자를 보호하지 못한다(대리 서명자 비 보호). 2.2절에서 대리 서명자를 보호하면서, forward secrecy를 제공하는 새로운 기법을 소개한다.

3.2 새로운 기법

본 절에서는 forward secrecy를 제공하면서 동시에 대리 서명 위임자가 대리 서명자로 가장하는 위조 공격으로부터 대리 서명자를 보호할 수 있는 방법을 제안한다.

3.2.1 대리 서명자를 보호할 수 있는 대리 서명 기법

먼저 새로운 대리 signcryption 기법을 구성하기 위해 사용되는 변형된 대리 서명 기법을 소개한다.

공개변수 및 A, B의 키 쌍들은 앞의 대리 서명 방식과 동일하며, 대리 서명자의 키는 다음과 같다.

x_p : 대리서명자의 개인키

$y_p=g^{x_p} \bmod p$: 대리서명자의 공개키

〔표 7〕 변형된 대리 서명 기법

A에 의한 대리 서명 키 생성	→	P에 의한 대리 서명키 확인
$x\in_R[1,2,\cdots,q-1]$ $K=g^x \bmod p$ $x_{ap}=x_a+xK \bmod q$	K, x_{ap}	$g^{x_{ap}} \stackrel{?}{=} (y_a \cdot K^K) \bmod p$
P에 의한 대리 서명 생성	⇒	B에 의한 대리 서명 확인
$x'\in_R[1,2,\cdots,q-1]$ $r'=hash(g^{x'} \bmod p, m)$ $s'=\frac{x'}{(x_p r'+x_{ap})} \bmod q$	m, r', s', K	$y_{ap}=y_a \cdot K^K \bmod p$ $k=(y_{ap}y_p^{r'})^{s'} \bmod p$ $hash(k, m)=r'$을 만족하면 m을 받아들인다.

이 기법에서는 대리 서명자의 서명 생성 과정에 대리 서명자의 개인키 x_p를 요구하기 때문에 대리 서

명자 이외의 다른 사람은 대리 서명자로 가장하여 대리 서명을 생성할 수 없다. 그러므로 x_p를 알지 못하는 서명 위임자에 의한 대리 서명 위조 공격이 불가능하다.

3.2.2 변형된 SDSS를 적용한 Proxy-Signcryption

3.2.1의 변형된 대리서명 방식을 이용하여 대리 서명키 x_{ap}에 대한 forward secrecy의 성질을 만족하면서 동시에 서명 위임자가 대리 서명자의 서명을 위조할 수 없도록 하는 signcryption 기법을 제안한다.

공개변수 및 참여자들의 키 쌍들은 앞의 Proxy-Signcryption 기법에서와 동일하다.

〔표 8〕 변형된 대리 signcryption

A에 의한 대리 서명 키 생성	→	P에 의한 대리 서명키 확인
$x \in_R [1,2,\cdots,q-1]$ $K = g^x \bmod p$ $x_{ap} = x_a + xK \bmod q$	K, x_{ap}	$g^{x_{ap}} \stackrel{?}{=} (y_a \cdot K^K) \bmod p$
P에 의한 proxy-signcryption	⇒	B에 의한 unsigncryption
$x' \in_R [1,2,\cdots,q-1]$ $k = y_b^{x'} \bmod p$ $k_1 \| k_2 = k$ $c = E_{k_1}(m)$ $r' = KH_{k_2}(m)$ $s' = \frac{x'}{(x_p r' + x_{ap})} \bmod q$	c, r', s', K	$y_{ap} = y_a \cdot K^K \bmod p$ $k = (y_{ap} y_p^{r'})^{s' x_b} \bmod p$ $k_1 \| k_2 = k$ $m = D_{k_1}(c)$ $KH_{k_2}(m) = r'$을 만족하면 m을 받아들인다.

이 기법은 대리 서명자의 개인키 x_p와 공개키 y_p를 사용하여 3.1.2절에서 언급한 기존의 기법과 유사한 효율성으로(지수 연산량과 통신량의 증가 없이) forward secrecy 성질을 강화시켰을 뿐만 아니라 동시에 정직하지 않은 대리 서명 위임자로부터 대리 서명자를 보호할 수 있는 기법이다.

키 생성 과정에서

$$(y_{ap} \cdot y_p^{r'})^{s' \cdot x_b} \bmod p = (y_b^{x_{ap} + x_p r'})^{s'} \bmod p$$

이 성립하며, 정당한 수신자는 자신의 개인키 x_b를 사용하여 등식의 좌변을 사용하여 키를 계산하지만, 정당하지 않은 공격자는 3.1.2에서와 같이 우변의 식을 이용하여 공격을 시도할 것이다. 그러나 대리 서명키 x_{ap}가 드러나는 경우에도 대리서명자의 개인키 x_p를 알지 못하는 사람은 키를 계산할 수 없다. 반대로 x_p가 드러나는 경우에도 x_{ap}를 알지 못하는 사람은 키를 계산할 수 없다. 물론 x_{ap}와 x_p가 모두 드러나면 키를 계산할 수 있지만, 두 키가 모두 드러날 경우는 x_{ap}(혹은 x_p) 하나만 드러날 경우에 비하여 낮은 확률로 발생한다. 또한 x_p를 알지 못하는 사람은 대리 서명을 수행할 수 없으므로 대리 서명자로 가장할 수 없으며 수신자의 서명 확인에 y_p가 요구되므로, 수신자 B는 대리 서명자의 신원을 명확하게 확인할 수 있다. 다음의 표는 기존의 proxy-signcryption 기법과 변형된 proxy-signcryption 기법의 효율성을 비교한 것이다.

〔표 9〕 효율성 비교

	지수연산갯수 및 통신량	Forward Secrecy	대리서명자 보호
Proxy-Signcryption	7번 $3\|p\| + 2\|q\|$	×	×
변형된 Proxy-Signcryption	7번 $3\|p\| + 2\|q\|$	○	○

Ⅳ. Signcryption의 안전성

Signcryption 기법은 서명 기법과 암호 기법에서 고려해야하는 기본적인 성질을 모두 만족해야 한다.

위조 불가능(unforgeability)

- 공격자는 송신자(A)로 가장하여 메세지 m에 대한 signcrypt된 타당한 메세지를 생성해 내는 것이 계산적으로 불가능 해야 한다. 여기서 공격자는 정직하지 않은 수신자(B)가 될 수도 있다.

기밀성(confidentiality)

- A와 B를 제외한 제 3자는 signcrypt된 메시지로부터 원본 메세지 m에 대한 정보를 계산할 수 없어야 한다.

기존의 signcryption 기법이 이와 같은 성질을 만족하는지 살펴보고, 이것을 바탕으로 변형된 signcryption 기법이 위의 성질을 만족함을 보인다.

4.1 위조 불가능성

Y.Zheng은 〔8〕에서 Signcryption 기법의 위조

불가능성을 언급하였다. 본 논문에는 보다 구체적으로 signcryption의 위조 불가능성을 살펴본다.

완전 공격(total break)

완전 공격은 공격자가 송신자 A의 비밀키 x_a를 계산할 수 있게 되는 것을 의미한다. Unsigncryption 과정에서 x_a를 포함하는 식은 $s=x/(r+x_a) \bmod q$ 이며 이 식에는 알려지지 않은 두 개의 값 x와 x_a가 있다. 이때 공격자는 x를 알지 못하고는 x_a에 관한 어떠한 정보도 알 수 없으므로 x_a를 계산하기 위해서는 공개된 값 y_a를 가지고 $\log_g(y_a)$를 계산해야만 하며 이것은 이산 대수 문제를 해결해야 한다는 것을 의미한다.

위조공격(forgery attack)

공격자는 송신자 A로 가장하여 메시지 m에 대한 signcrypt된 메세지를 생성하는 것이 목적이다. 이때 송신자 A로부터 받은 signcrypt된 메시지를 바로 확인 할 수 있는 수신자 B가 가장 강력한 공격자가 될 수 있다. B는 자신의 개인키 x_b를 사용하여 A가 보낸 (c, r, s)로부터 $k=hash((y_a \cdot g^r)^{s \cdot x_b} \bmod p)$를 계살 할 수 있으며 이것으로 메세지 $m=D_{k_1}(c)$를 생성할 수 있다. 따라서 B는 (m, r, s)를 가지고 signcrypt된 문서의 생성을 시도할 것이다. 이것은 signcryption에서 복호화 과정을 통과한 값으로 signcryption 기법의 "위조 불가능성"은 전적으로 SDSS에 의존한다는 것을 의미한다. 그러므로 signcryption 기법의 위조 공격 가능성 여부는 SDSS의 위조 가능성 여부로 판단해도 무리가 없다. 본 논문에서는 SDSS1에 대해서만 고려한다. SDSS2는 이와 유사한 방법으로 보일 수 있다.

모든 메세지에 대해 가능한 위조 공격(universal forgery attack)

공격자는 임의의 메세지에 대하여 x를 랜덤하게 선택하고 $r=hash(g^x \bmod p, m)$을 계산한다. 서명 생성을 위하여 타당한 s의 구성을 시도할 것이다.(또는 이와 반대로 할 수 도 있다.)

서명 확인 과정에서 $(y_a \cdot g^r)^s \bmod p = g^x \bmod p$가 성립해야 하므로 타당한 s를 계산하기 위해서는 $\log_{(y_a g^r)} g^x$을 계산해야 만 한다. 따라서 메세지 m에 대한 SDSS1의 적절한 서명을 위조하는 것은 이산 대수 문제를 해결하는 것 만큼은 어렵다.

임의 메세지에 대한 위조공격(existential forgery of messages)

서명 생성 과정에 해쉬 함수를 사용하므로 사실상 ElGamal 서명과 DSS는 특정 메시지에 대한 위조 공격이 성립되지 않는다[6]. 만일 해쉬 함수를 사용하지 않는 다면 SDSS1도 공격이 가능하다. 해쉬 함수를 사용하지 않은 SDSS1의 서명값은 r과 s 각각이 $r=g^x \bmod p$, $s=x/(r+x_a \cdot m) \bmod q$ 이고 서명 확인 과정이 $r=(g^r \cdot y_a^m)^s \bmod p$가 된다.

이때는 다음과 같은 임의 메세지에 대한 위조 공격이 성립하게 된다.

공격자는 $w, u \in Z_q^*$를 선택하고 $r=g^u y_a^w \bmod p$와 $s=u \cdot r^{-1} \bmod q$를 계산한다. 이것은 메세지 $m=w \cdot s^{-1} \bmod q$에 대한 타당한 서명이 된다.

그러나 해쉬 함수를 사용하여 이러한 공격을 막을 수 있다.

공격자는 w와 u를 선택하고, $y_a^w \cdot g^u \bmod p$을 계산한다. 해쉬 함수 안의 계산에서 $r=hash(y_a^w \cdot g^u \bmod p, m)$을 계산해야 한다. 이 식을 만족하는 r을 계산하기 위해서는 r을 계산하기 이전에 위조가 가능하도록 메시지 m을 구성해야 하므로 이와 같은 r을 계산할 수 없다. 만일 m을 임의로 선택하여 r을 계산하고 $s=u \cdot r^{-1} \bmod q$로 서명을 생성한다면 이것은 타당한 서명이 성립되지 않는다. 서명 확인 과정에서 $hash(y_a^w \cdot g^u \bmod p, m) \neq hash((y_a^m \cdot g^r)^{ur^{-1}}, m)$가 되기 때문이다.

SDSS1과 SDSS2는 각각 Meta-ElGamal 서명 기법에서 ElGamal 서명의 변형인 EGⅣ.1 과 EGⅣ.6 이다.[3] 위의 사실들로 signcryption 기법은 위조 공격에 강함을 알 수 있다.

이제 변형된 signcryption 기법의 안전성을 보이기 위해 [7]에 언급된 정의를 사용한다.

<정의 2: 서명 기법의 동치>

두 서명 기법 S_1과 S_2가 송신자 A의 비밀키 x_a없이 임의의 메세지 m에 대하여 S_1의 서명으로부터 S_2의 서명을 생성할 수 있고, S_2의 서명으로부터 S_1의 서명을 생성할 수 있다면 이 서명들은 서로 동치(equivalence)라고 말한다.

이 정의를 만족하는 두 서명 기법은, 하나의 서명이 위조 가능할 때 다른 서명 또한 위조 가능하다. 반대로 하나의 서명이 위조 공격에 안전하다면 나머지 다른 서명 또한 위조 공격에 안전하다는 것을 알 수 있다.

<정리 1> SDSS1과 MSDSS는 서로 동치 이다.

<증명>

(r, s)를 메시지 m에 대한 SDSS1의 서명이라고 하자. 이때 $r = hash(g^x \bmod p, m)$, $s = x/(r + x_a) \bmod q$를 만족할 것이다.

그러면 $(g^r \bmod p, s)$는 메시지 m에 대한 MSDSS의 서명이 된다.

반대로 (R, s)를 MSDSS의 서명이라고 하자. $(r = hash((y_a \cdot R)^s \bmod p, m), s)$는 SDSS1의 서명 확인과정의 $g^x = (y_a \cdot g^r)^s \bmod p$을 만족하므로 SDSS1의 메세지 m에 대한 서명이 된다. ■

따라서 SDSS1으로부터 MSDSS를 생성할 수 있고, MSDSS로부터 SDSS1을 생성할 수 있으므로 SDSS1이 위조 공격에 안전하다면 MSDSS도 위조 공격에 안전함을 알 수 있다. 또한 앞에서 밝힌 바와 같이 SDSS1의 위조에 대한 안전성은 이산 대수 문제를 해결 하는 것 만큼 어려우므로 MSDSS도 이산 대수 문제를 해결하는 것 만큼 어렵다는 것을 알 수 있다.

4.2 기밀성

Zheng의 signcryption기법의 기밀성은 [8]에 언급되어 있다. 변형된 signcryption 기법이 메시지 기밀성에 대하여 Zheng의 기법과 동일한 안전성을 제공함을 보이기 위하여 다음을 정의한다.

<정의 3: 기밀성에 대한 동치>

두 signcryption기법 SE_1, SE_2에서 수신자의 개인키 x_b 없이 SE_1의 세션키 k_1으로부터 SE_2의 세션키 k_2를 계산할 수 있고, SE_2의 세션키 k_2으로부터 SE_1의 세션키 k_1을 계산할 수 있다면 이 두 signcryption 기법은 메세지의 기밀성에 대하여 동치(equivalence)이다.

위의 정의를 만족하는 두 signcryption 기법은 하나의 키를 계산할 수 있는 사람이면 다른 기법의 키를 계산할 수 있는 사람이며, 반대로 하나의 키도 계산 할 수 없는 사람이라면 다른 기법의 키도 계산 할 수 없게 된다.

<정리 2>

Zheng의 signcryption 기법과 변형된 signcryption 기법은 메시지 기밀성에 대하여 동치이다.

<증명>

k_{zheng}을 Zheng의 기법의 키, k_{msig}을 변형된 기법의 키라고 하자.

$$k_{zheng} = hash(y_b^x \bmod p) = k_{msig}$$

이 성립하므로 두 기법은 메시지 기밀성에 대하여 자연스럽게 동치가 된다. ■

대칭키 암호 알고리즘의 안전성(메세지 기밀성)은 전적으로 키에 의존하므로 사용되는 암호화 복호화 알고리즘이 안전하다면 위의 정의와 정리에 의하여 변형된signcryption 기법은 Zheng의 기법과 동일한 안전성을 가짐을 알 수 있다.

V. 결 론

Signcryption이 Y. Zheng에 의해 제안된 이후로 signcryption 기법을 응용한 여러 프로토콜이 제안되어 왔으나 이러한 기존의 signcryption 기법은 송신자의 개인키에 대한 forward secrecy의 성질은 제공하지 못한다. 본 논문에서는 forward secrecy의 성질을 제공하는 두 가지 signcryption 기법들을 소개하였다. 첫 번째 기법은 Y. Zheng의 기법을 변형한 것으로 Y. Zheng의 기법보다 한번의 지수 연산의 증가를 추가하여 송신자에 대한 forward secrecy를 제공할 수 있는 기법이며, 두 번째 기법은 C. Gamage등이 제안한 proxy-signcryption 기법[2]의 변형으로 C. Gamage의 기법과 유사한 효율성(지수 연산량 및 통신량에 대한)을 가지고 forward secrecy를 제공하는 동시에 대리 서명자 비보호 기법을 대리 서명자 보호 기법으로 바꾸어 준다.

참 고 문 헌

[1] F. Bao and R. H.Deng, "A Signcryption Scheme with Signature Directly Verifiable by Public Key", Proc. of PKC'98, LNCS, Vol. 1431, Springer-Verlag, pp. 55~59, 1998.

[2] C. Gamage, J. Leiwo and Y. Zheng, "An Efficient Scheme for Secure Message

Transmission using Proxy-Signcryption," Proceeding of the Twenty Second Australian Computer Science Conference, pp.18~21, Jan, 1999.

[3] P. Horster, M. Michels, and H. Petersen, "Meta-ElGamal signature schemes". In Proceedings of the second ACM Conference on Computer and Communications Security, The Association for Computer Machinery, pp. 96~107, November 1994.

[4] M. Mambo, K. Usuda and E. Okamoto, "Proxy signatures for delegating signing operation", Proc. Third ACM Conference on Computer and Communications Security, pp. 48~57, 1996.

[5] A. J. Menezes, P. C. van Oorschot and S. A. Vanstone, "Handbook of Applied Cryptography", pp. 490~524, 1997.

[6] N.I.S.T. National Technical Information Service, "Digital Signature Standard(DSS)", FIPS 186, U.S. Department of Commerce, Springfield, Virginia, 1994.

[7] K. Nyberg, R. Rueppel, "Message recovery for signature schemes based on the discrete logarithm problem", Preproceedings of Eurocrypt '94, pp. 175~190, 1994.

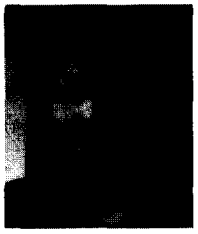
[8] Y. Zheng, "Digital signcryption or how to achieve cost(signature and encryption) ⟨⟨cost(signature)+cost(encryption)", Advances Cryptology-CRYPTO'97, LNCS Vol. 1294, Springer-Verlag, pp. 165~179, 1997.

[9] 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.

[10] Y. Zheng, "Compact and unforgeable session key establishment over an ATM network", In Proceedings of IEEE INFOCOM' 98, pp. 411~418, 1998.

[11] Y. Zheng, "Improved public key cryptiosystems secrue against chosen ciphertext attacks", Technical Report 94-1, University of Wollongong, 1994.

〈著者紹介〉



정 희 윤 (Hee-Yun Jeong)
2000년 2월 : 한양대학교 수학과 졸업
2000년 2월~현재 : 고려대학교 수학과 석사과정
〈관심분야〉 암호이론, 암호 프로토콜

이 동 훈 (Dong Hoon Lee) 정회원
1984년 : 고려대학교 경제학과 졸업
1987년 : Oklahoma Univ. 전산학과 석사
1992년 : Oklahoma Univ. 전산학과 박사
1993년~현재 : 고려대학교 전산학과 교수
2000년~현재 : 고려대학교 정보보호 대학원 교수
〈관심분야〉 암호이론, 암호 프로토콜, 정보이론

임 종 인 (Jong-In Lim) 정회원
1980년 : 고려대학교 수학과 졸업
1982년 : 고려대학교 수학과 석사
1986년 : 고려대학교 수학과 박사
1986년~현재 : 고려대학교 수학과 교수
2000년~현재 : 고려대학교 정보보호 대학원 원장
〈관심분야〉 암호이론, 암호 프로토콜, 정보이론