工學碩士 學位論文

이동통신 환경에 적합한 효율적인 Proxy-signcryption 방식에 관한 연구

指導教授 朴 志 煥

이 論文을 工學碩士 學位論文으로 提出함.

2003年 2月

釜慶大學校 産業大學院

電 算 情 報 學 科

金 東 佑

金東佑의 工學碩士 學位論文으로 인준함

2002年 12月 14日

主 審 공학박사 김 영 봉 

委 員 이학박사 조 성 진 

委 員 이학박사 박 지 환 

<차 례>

<표 차례>

<그림 차례>

A Study on Efficient Proxy-signcryption Scheme for Mobile Communications

Dong - Woo Kim

Dept. of Computer and Information Graduate School of PuKyong National University

abstract

Zheng suggested a new concept called signcryption that provides confidentiality with digital signature properties. The signcryption scheme is more efficient than general method what we call first-sign-then-encrypt or first-encrypt-than-sign in computational and communicational cost. However, these

schemes do not gain the confidentiality in nonrepudiation. In this paper, we suggest an inproved signcryption with confidentiality in nonrepudiation.

Also, Based on the development of mobile communication, the future mobile communication systems are expected to provide higher quality of multimedia services for users than today's systems. Therefore, many technical factors are needed in this systems. Especially the confidentiality and security would be obtained through the introduction of the security for mobile communication. In this paper, we presents an efficient proxy-signcryption scheme that processes a user's digital signature and encryption using the proxy agent who has more computational power than origins on mobile communication. The proposed scheme provides non-repudiation and prevents creating illegal signature by the origin and proxy agent in a phase of proxy signature processing, this scheme also keep the confidentiality and security in the

mobile communication by means of confirming signature by the right receiver.

1. 서 론

송수신자가 공개 통신망에서 비밀리에 메시지를 교환하고자 할 때, 공개키 암호방식은 비밀키 암호방식과 비교하여 같은 길이의 메시지를 암호화하는데 많은 시간과 자원이 소요되기 때문에 키의 분배단계에서는 공개키 암호방식을 사용하고 실질적인 메시지 교환단계에서는 비밀키 암호방식을 이용하는 하이브리드 방식을 많이 사용하고 있다. 이런 방식의 대표적인 예로 임의의 난수로 키를 만들어 메시지를 암호화하고 RSA 암호시스템을 이용하여 수신자의 공개키로 암호키를 암호화한 후 상대방에게 보내는 방법이 있다.

이렇게 메시지를 교환함에 있어 메시지에 대한 전자서명은 별개의 과정에 따라 암호화되어 상대방에게 전해지는 것이 일반적이며 이런 방법을 first-sign-then -encryption이라고 한다. Zheng은 이런 두 개의 과정을 하나로 통합한 Signcryption 이라는 새로운 개념의 두가지 Signcryption 방식을 제안하였다[1,2]. 하지만, Zheng의 Signcryption 방식을 이용할 때, 수신자가 제3자(중재자 혹은 공공기관등)에게 송신자의 전자서명을 증명하기 위하여 관련정보를 공개하는 경우, 제3자는 이를 이용하여 송수신자간의 암호화된 메시지를 풀수 있는 키를 만들 수 있어 기밀성을 보장할 수 없다. 따라서 본 논문에서는 기존 Signcryption의 부인봉쇄 과정에서 생길수 있는 기밀성 문제점을 해결하고, Zheng의 Signcryption과 계산량 및 통신량이 비슷한 Signcryption방식을 소개한다. 또한 최근 네트워크의 발전과 정보통신 분야의 급속한 발전에 따라 사용자들은 네트워크에 직접 연결된 컴퓨터를 사용하지 않고도 이동 중에 소형 노트북이나 휴대폰, PDA등과

같은 휴대용 단말기를 이용, 인터넷에 접속 가능 하게 되었다. 이동통신 분야는 산업계에서 가장 빨리 성장하는 분야 중의 하나로서 많은 사람들이 이동통신 서비스를 통해 그 편리성과 유용성을 인지하고 있다. 그러나 이러한 이동 통신 관련 서비스들은 많은 보안상 문제점들에 노출될 수 있다. 즉, 이동통신에서 신호 교환은 무선 채널을 통해 수행되므로 도청자나 그 밖의 신뢰되지 못한 요소들로부터 위조나 불법적 변경 등과 같은 위협들에 대해서는 취약성을 지니고 있다. 뿐만 아니라 사용자 인증 및 부인봉쇄 등과 같은 여러 가지 문제가 발생할 수 있게 된다. 따라서 불법 가입자들로부터 기밀성과 안전성을 확보하고, 사용자의 인증성을 제공하기 위한 방법 중에 하나로서 '수신자 지정 서명' 기법이 제시되었다[3]. 이 기법은 네트워크 상에서 기밀성과 사용자 인증성을 동시에 제공하기 위해서 전자서명을 수행한 결과에 공개키 암호 방식을 사용하여 전송하게 된다. 그러나 디지털 서명이나 공개키 암호 방식은 모두 모듈라 멱승과 같은 많은 계산량을 요구하므로 상대적으로 계산 능력이 떨어지는 휴대용 단말기에 사용하기에는 어려운 점이 있다. Gamage등은 이러한 문제점을 해결하기 위해 Signcryption 방식과 대리 서명 방식[4,5]을 이용하여(Signcryption을 생성하는데 요구되는 계산을 상대적으로 계산 능력이 뛰어난 서버에 의뢰함으로써) 휴대용 단말기가 수행해야 할 계산량을 더욱 감소시킨 Proxy-signcryption 방식을 제안하였다[6]. 그러나 Gamage등이 제안한 방식을 실제로 이동 통신에 적용할 경우, 자신이 전송한 메시지에 대해 부인하는 것을 판단할 수 없으므로 proxy agent를 보호할 수 없다는 문제점이 있다. 이를 해결하기 위해 대리인 보호형 Proxy-signcryption 방식[7]이

제안되었으나, Proxy agent와 Bob사이에 forward secrecy를 제공하지 않으므로 메시지의 안전성 유지가 어렵다는 문제점이 있다. 여기서 forward secrecy란 송신자의 개인키를 알게 되는 사람이 이전에 송신자가 이 키를 사용하여 생성했던 signcrypt된 문서로부터 원본 문서를 복구해 낼 수 없는 것을 말한다. 따라서 본 논문에서는 적은 계산량으로 디지털 서명을 수행하면서 기존의 Proxy-signcryption이 가지고 있는 문제점을 해결하고, proxy agent에서 forward secrecy를 제공하는 개선된 Proxy-signcryption을 제안한다.

본 논문의 구성은 2장에서 이산대수에 기초한 서명과 암호 방식에 대해 간략하게 소개하고, 3장에서는 기존 Signcryption 방식의 문제점을 살펴보고, 이러한 문제점을 해결하는 Signcryption 방식을 소개한다. 4장에서는 대리서명 방식과 대리인 보호형 대리서명 방식을 살펴보고, 대리 서명을 수행할 경우 발생할 수 있는 사용자와 대리서명 agent의 부정 서명 생성 및 부인행위를 방지하며 동시에 정당한 수신자가 서명을 확인하도록 함으로서 이동 통신상에서 기밀성을 유지하는 효율적인 Proxy- signcryption 방식을 제안한 후, 5장에서 결론을 맺는다.

2. 이산대수에 기초한 서명과 암호

2.1 Elgamal 서명과 암호

Elgamal의 디지털 서명(Digital signature)과 암호(Encryption)은 이산대수 문제해결의 어려움에 기초 하고 있다[8].

[키 설정]

p : 큰 소수

q : $p-1$을 나누는 큰 소인수

g : 유한체 상의 위수 q인 Z_p에 속하는 정수

x_a, x_b : Alice와 Bob의 비밀키

y_a, y_b : Alice와 Bob의 공개키($y_i = g^{x_i} \bmod p$)

2.1.1 Signature

[표 1] Elgamal Signature

서명 생성	$x \in {}_R Z_q$ $r = g^x \bmod p$ $s = (hash(m) - x_a r) / x \bmod (p-1)$
서명 검증	$g^{hash(m)} = y_a^r \cdot r^s \bmod p$

2.1.2 Encryption

[표 2] Elgamal Encryption

암호 (Encryption)	$k = y_b^x \bmod p$ $c_1 = E_k(m)$ $c_2 = g^x \bmod p$
통신(Send)	$c_1,\ c_2$
복호(Recover)	$k = c_2^{x_b} \bmod p$ $m = D_k(c_1)$

이렇게 함으로 Alice가 메시지(m)에 대해 서명하고 암호화 하여 보내고자 하는 상대방에게 안전한 방법으로 보낼 수 있다. 또한 Elgamal encryption은 Schnorr sinature와 DSS(Digital Signature Standard)의 기초가 된다.

2.2 Schnorr 서명 방식

Schnorr 서명방식(Schnorr Signature Scheme)과 DSS는 Elgamal 서명방식(Elgamal signature scheme)의 변형된 형태[9]로 키설정은 다음과 같다.

[키 설정]

p : $p \geq 2^{512}$을 만족하는 큰 소수

q : $p-1$을 나누는 큰 소인수($q \geq 2^{144}$)

g : 유한체 상의 위수 q인 Z_p에 속하는 정수

$x_a,\ x_b$: Alice와 Bob의 비밀키

y_a : Alice의 공개키($y_a = g^{-x_a} \bmod p$)

2.2.1 Signature

[표 3] Schnorr Signature

서명 생성	$x \in {}_R Z_q$ $r = hash(g^x \bmod p,\ m)$ $s = x + x_a \cdot r \bmod q$
서명 검증	$r = hash((g^s\ y_a^r \bmod p),\ m)$

만약 여기서 공개키를 $y_a = g^{-x_a} \bmod p$ 대신에 $y_a = g^{x_a} \bmod p$을 사용한다면 $s = x - x_a \cdot r \bmod q$로 정의되고 서명의 검증은 동일하다.

2.3 DSS(Digital Signature Standard)

DSS방식은 공개키 $y_a = g^{-x_a} \bmod p$를 사용하지 않고 대신에 $y_a = g^{x_a} \bmod p$을 사용하며, 나머지 공캐키와 비밀키의 정의는 Schnorr Signature Scheme와 동일하다.

[키 설정]

p : $p \geq 2^{512}$을 만족하는 큰 소수

q : $p-1$을 나누는 큰 소인수($q \geq 2^{144}$)

g : 유한체 상의 위수 q인 Z_p에 속하는 정수

h : 일방향 해쉬함수($512 \leq |p| \leq 1024$, $|q| \leq 160$, *output* 160*bit*)

x_a, x_b : Alice와 Bob의 비밀키

y_a : Alice의 공개키($y_a = g^{x_a} \bmod p$)

2.3.1 Signature

[표 4] DSS Signature

서명 생성	$x \in {}_{R}Z_q$ $r = (g^x \bmod p) \bmod q$ $s = (hash(m) + x_a \cdot r)/x \bmod q$
서명 검증	$v = (g^{hash(m)/s} \cdot y_a^{r/s} \bmod p) \bmod q$ *check* if $v = r$, *accepts* (r, s)

2.3.2 Cost of Elgamal, Schnorr, DSS

[표 5]는 지금까지 살펴본 Elgamal, Schnorr, DSS의 각 서명과 암호방식에 대한 계산량과 통신량을 보여준다.

[표 5] Cost of Elgamal, Schnorr, DSS

구 분	계 산 량	통 신 량
Elgamal Encrpytion	EXP=2, ENC=1 (EXP=1, DEC=1)	| p |
Elgamal Signature	EXP=1, MUL=1, DIV=1, ADD=1, HASH=1 (EXP=1.25, MUL=1, DIV=0, ADD=0, HASH=1)	2 | p |
Schnorr Signature	EXP=1, MUL=1, ADD=1, HASH=1 (EXP=1.17, MUL=1, ADD=0, HASH=1)	| hash(·) | + | q |
DSS	EXP=1, MUL=1, DIV=1, ADD=1, HASH=1 (EXP=1.17, MUL=1, DIV=2, ADD=0, HASH=1)	2 | q |

- EXP = the number of modular exponentiations,
- MUL = the number of modular multiplications,
- DIV = the number of modular division,
- ADD = the number of modular addition or subtraction,
- HASH = the number of one-way or keyed hash operations,
- ENC = the number encryptions using a private key cipher,
- DEC = the number decryptions using a private key cipher

3. Signcryption 방식

3.1 기존 Signcryption의 문제점

본 장에서는 Zheng과 Michel의 Signcryption 방식의 문제점과 김성덕의 Signcryption을 살펴본다.

[기호 및 표기]

p : 큰 소수

q : $p-1$을 나누는 큰 소인수

g : 유한체 상의 위수 q인 Z_p에 속하는 정수

h : 일방향 해쉬 함수

KH : Keyed 일방향 해쉬 함수

E, D : 비밀키를 사용하는 암호화/복호화 알고리즘

x_a, x_b : Alice와 Bob의 비밀키

y_a, y_b : Alice와 Bob의 공개키($y_i = g^{x_i} \bmod p$)

3.1.1 Y. Zheng의 Signcryption 방식과 문제점

1997년 Zheng은 SDSS1과 SDSS2로 명명된 두 가지의 Signcryption 방식을 제안하였으며, 각각의 Signcryption 생성과 검증 과정은 [표 6]과 같다.

[표 6] Y.Zheng의 Signcryption 방식

생성	$x \in {}_R Z_q$ $k = h(y_b^x \bmod p)$ $k = k_1 \| k_2$ $r = KH_{k_2}(m)$ [*SDSS* 1] $s = x\,(r+x_a)^{-1} \bmod q$ [*SDSS* 2] $s = x\,(1+x_a r)^{-1} \bmod q$ $c = E_{k_1}(m)$
통신	$c,\ r,\ s$
검증 및 복호	[*SDSS* 1] $k = h((y_a\ g^r)^{s \cdot x_b} \bmod p)$ [*SDSS* 2] $k = h((y_a^r\ g)^{s \cdot x_b} \bmod p)$ $k = k_1 \| k_2$ $m = D_{k_1}(c)$ *check* if $r = KH_{k_2}(m)$

Zheng은 제안한 방식이 공개통신로상에서 전자 문서의 기밀성을 보장할 뿐만 아니라 전자 서명을 통해 부인봉쇄도 동시에 제공한다고 하였다. 하지만 제안된 Signcryption 방식은 Bob이 Carol에게 Alice의 Signcryption을 증명하는 과정에서 중간정보가 공개되는 경우에 Carol은 이 정보를 이용하여 y_{ab}를 계산할수 있게 되고, Carol은 이 y_{ab}를 이용하여 Alice와 Bob 사이에서 이루어지는 Signcryption에 의한 암호통신문을 언제든지 복호 할 수 있는 키를 생성할 수 있음을 Petersen 등이 지적하였으며, 그 과정은 다음과 같다[10].

① Carol에게 공개되는 정보 : k, r, s, y_b

② [SDSS 1] $y_{ab} = k^{s^{-1}} y_b^{-r} \bmod p = (y_b^x)^{s^{-1}} y_b^{-r}$

$= (y_b^x)^{r+x_a /x} y_b^{-r} = y_b^{r-r+x_a}$

$= y_b^{x_a} \bmod p$

③ 통신로 상의 정보 : s', r', c'

④ [SDSS 1] $k' = y_{ab}^{s'} y_b^{r's'} \bmod p$

$= y_b^{s'x_a} y_b^{r's'} = y_b^{s'(x_a + r')}$

$= y_b^{(x_a + r')x' /(x_a + r')}$

$= y_b^{x'} \bmod p$

⑤ $k' = k_1' \| k_2'$

⑥ $m' = D_{k_1'}(c')$

[SDSS 2]도 유사하게 계산할 수 있다.

즉, Zheng의 Signcryption 방식은 기밀성을 유지하지 못한다.

3.1.2 M. Michel의 Signcryption 방식과 문제점

Petersen 등은 Signcryption 방식의 단점을 지적하고 이를 보완한 새로운 Signcryption을 제안하였지만 이방식은 Signcryption의 생성과정에서 변수 사이의 연관성 결여로 수신자가 송신자의 Signcryption을 생성할 수 있기 때문에 부인봉쇄 기능을 보장할 수 없다는 단점이 있다.

Petersen의 공동 집필자인 Michel은 이런 사실을 인지하고 기존의 방식을 수정한 [표 7]와 같은 Signcryption 방식을 제안 하였다[11].

[표 7] M. Michel의 Signcryption 방식

생성	$x_1,\ x_2 \in {}_RZ_q$ $C = E_{x_1}(M \| H(M))$ $k = H(y_b^{x_2} \bmod p,\ C)$ $r = x_1 k \bmod q$ $s = x_2(r+x_a)^{-1} \bmod q$
통신	$s,\ r,\ C$
검증 및 복호	$K = y_b^{rs} y_a^{sx_b} \bmod p$ $k = H(K, C)$ $x_1 = rk^{-1} \bmod q$ $M \| H(M) = D_{x_1}(C)$

하지만 Michel의 Signcryption 방식 역시 Zheng의 Signcryption 방식과 유사한 문제점을 가지고 있음을 김성덕이 지적하였으며, 그 과정은 다음과 같다.[12]

① 공개정보 : $y_b,\ K,\ s,\ r$

② $y_{ab} = K^{s^{-1}} y_b^{-r} \bmod p = (y_b^{rs} y_a^{sx_b})^{s^{-1}} y_b^{-r}$

$= g^{rx_b + x_a x_b - rx_b} = g^{x_a x_b} \bmod p$

③ Signcryption : $s',\ r',\ C'$

④ $y_{ab}^{s'} y_b^{r's'} = y_b^{r's' + s'x_a} = y_b^{s'(x_a + r')}$

$= y_b^{(x_a + r')x'_2 / r' + x_a} = y_b^{x'_2} \bmod p$

$$k' = H(y_b^{x'_2} \bmod p,\ C)$$

⑤ $x_1' = r'k'^{-1}$,

$$M' = D_{x'_1}(C')$$

즉, Michel의 Signcryption 방식도 기밀성을 유지할 수 없다.

3.1.3 김성덕의 Signcryption 방식과 특징

김성덕은 Michel이 제안한 Signcryption 역시 Zheng의 Signcryption 방식과 유사한 문제점이 있음을 지적하고, 이를 보완한 [표 8]과 같은 새로운 Signcryption을 제안하였다.

[표 8] 김성덕의 Signcryption 방식

생성	$x \in Z_q^*$ $K = g^x \bmod p$ $r = y_b^x \bmod p$ $k = H(K, M) \bmod q$ $T = x - x_a k \bmod q$ $C = E_K(M \| T)$
통신	$r,\ C$
검증 및 복호	$K' = r^{x_b^{-1}} \bmod p$ $M' \| T' = D_K(C)$ $k' = H(K', M') \bmod q$ check if $K' = y_a^k g^T \bmod p$

김성덕 등이 제안한 Signcryption 방식의 특징을 살펴보면 y_b^x를 전자서명 값으로 직접 이용하여 제 3자가 y_{ab}를 계산하더라도 전자서명의 안전성을 유지할 수 있도록 하였고, r에서 g^x을 계산하기 위해서는 Bob의 x_b를 알아야 하므로 수신자 만이 이를 복호할 수 있다. 또한 s를 암호화하여 C에 포함하여 송부하므로 공격자는 s를 알 수 없다. 즉 r을 이용하여 암호문 C를 복호해야만 s를 알 수 있는 특징을 가지고 있다. 하지만 김성덕등의 방식은 서명후 암호화하는 방법에 비해서는 적은 통신량을 가지지만 기존의 Signcryption 방식에 비해서는 통신량이 많은 단점이 있다.

3.2 Signcryption 방식의 개선

Zheng과 Michel이 제안한 방식은 Signcryption을 Carol에게 증명하게 되면, Carol이 Alice와 Bob의 암호화된 송수신 메시지를 복호해 볼 수 있어 기밀성을 유지하지 못함을 살펴 보았다. 본장에서는 기존의 Signcryption 방식들이 가지고 있던 문제점을 해결한 [표 9]과 같은 Signcryption방식을 제안한다.

제안한 Signcryption 방식의 특징은 다음과 같다.

제안하는 방식은 ElGamal의 encryption[8]과 Schnorr의 signature기법[9]을 이용하였으며, Signcryption을 Carol에게 증명하는 과정에서 중간 정보들이 공개되어도 Carol은 Alice와 Bob사이의 암호문을 풀어볼 수 없는 특징이 있다.

[표 9] 제안하는 Signcryption 방식

생성	$x \in Z_q^*$ $k = h(y_b^x \bmod p)$ $r = KH_k(m)$ $c = E_k(m)$ $s = x - x_a r \bmod q$
통신	$s,\ r,\ c$
검증 및 복호	$k = h((g^s \cdot y_a^r)^{x_b} \bmod p)$ $m = D_k(c)$ check if $r = KH_k(m)$

3.2.1 기밀성(Confidentiality)

기존에 제안된 방식들은 별도의 정보를 이용하여 y_b^x를 복원하기 때문에 제 3자에게 정보가 공개 되는 경우 제 3자는 공개된 정보를 이용하여 y_{ab}를 계산할 수있고 이를 기반으로 암호문의 복호에 필요한 키를 만들 수 있다는 공통점이 있다. 제안하는 방식에서는 (s, c, r)을 동일하게 생성하고 제 3자에게 송신자의 전자서명을 증명하기 위해 관련정보를 공개하여 제 3자가 y_{ab}를 계산한다 하더라도 전자서명의 안전성을 유지하도록 하였다. 제안식의 암호 통신문 복호화 과정을 살펴보면 다음과 같다.

① 공개 정보 : k, r, s, y_b

② $y_{ab} = k^{s^{-1}} y_b^{-r} \bmod p = (y_b^x)^{s^{-1}} y_b^{-r} = y_b^{x/s - r}$

③ 통신로 상의 정보 : s', r', c'

④ $k' = y_{ab}^{s'} y_b^{r's'} \bmod p$

$= (y_b^{x/s - r})^{s'} y_b^{r's'} = y_b^{s'(x/s - r + r')} \neq k$

위와 같이 Carol은 y_{ab}를 계산하다 하더라도 $k' \neq k$ 이므로 Alice와 Bob사이에서 이루어지는 Signcryption에 의한 암호통신문을 복호할 수 없다.

3.2.2 계산량과 통신량

암호시스템과 전자서명 시스템의 효율성은 주로 계산량과 통신량을 기준으로 한다. [표 10]은 Zheng의 P1363 제출자료[1]를 기반으로 계산량과 통신량을 비교하였다. 통신량의 비교에서 암호문의 길이는 동일하므로 표에서는 생략한다.

제안한 Signcryption 방식은 서명후 암호화 하는 방식에 비해 모듈러 멱승이 2번 적고, 통신에 따른 오버헤드가 적음을 알수 있다. 또한 기존의 Zheng의 Signcryption 방식과 계산량 및 통신량에서 큰 차이 없이 Signcryption을 생성할 수 있다.

[표 10] Signcryption 방식 비교

구분		EXP	MUL	DIV	ADD	HASH	Com Overhead
SDSS 1	Sign	1	0	1	1	2	|H()| + |q|
	Verify	1.17	2	0	0	2	
SDSS 2	Sign	1	1	1	1	2	|H()| + |q|
	Verify	1.17	2	0	0	2	
Michel	Sign	1	2	1	1	2	|H()| +2 |q|
	Verify	1.17	4	1	0	2	
김성덕	Sign	2	1	1	1	1	|p| + |q|
	Verify	2.17	1	1	0	1	
제안방식	Sign	1	1	0	1	2	|H()| + |q|
	Verify	2.17	1	0	0	2	
Sign(Schnorr) Encryption	Sign	3	1	0	1	1	|H()| + |q| + |p|
	Verify	2.17	1	0	0	1	
Sign(DSS) Encryption	Sign	3	1	1	1	1	2|q| + |p|
	Verify	2.17	1	2	0	1	

4. 이동통신 환경에서 Proxy-signcryption 적용

4.1 Proxy-signcryption

Proxy-signcryption 방식이란 사용자가 지정한 대리인이 자신을 대신하여 정당한 Signcryption 메시지를 생성할 수 있도록 하는 방식으로 Signcryption을 생성하는데 요구되는 계산을 상대적으로 계산 능력이 뛰어난 proxy agent에 의존하는 것이다. 그 방식은 다음과 같다.

[시스템 설정]

- p : 512비트 이상의 큰 소수
- q : $q \mid p-1$인 큰 소수
- g : 위수가 q인 Z_p상의 원소
- x_A: $x_A \in Z_q$, Alice의 비밀키
- y_A : $y_A \equiv g^{x_A} \bmod p$, Alice의 공개키
- x_B: $x_B \in Z_q$, Bob의 비밀키
- y_B : $y_B \equiv g^{x_B} \bmod p$, Bob의 공개키
- x_P: $x_P \in Z_q$, proxy agent의 비밀키
- y_P : $y_P \equiv g^{x_P} \bmod p$, proxy agent의 공개키
- $KH()$: Keyed 해쉬 함수
- $E()$ / $D()$: 관용암호/ 복호 알고리즘

1) 대리서명용 키 생성

Alice는 $x \in Z_q$를 선택하고 $K \equiv g^x \bmod p$를 계산하여 대리 서명용 키 $x_{AP} \equiv x_A + x \cdot K \bmod q$ 를 생성하여 (x_{AP}, K)를 전송한다.

2) 대리서명용 키의 검증

Proxy agent는 $y_{AP} = g^{x_{AP}} = y_A \cdot K^K \bmod p$를 이용하여 자신이 받은 대리 서명용 키가 정당한지 확인한다.

3) Proxy agent에 의한 Signcryption 생성

Proxy agent는 비밀 랜덤수 $x' \in [1,2,\cdots,q-1]$를 선택하여 $k = y_B^{x'} \bmod p$를 계산한다.

$k = k_1 \| k_2$ 로 나누고 다음과 같은 메시지 m에 대한 Signcryption을 생성한다.

$$r' = KH_{k_2}(m)$$

$$s' = x' / (r' + x_{AP}) \bmod q$$

$$c = E_{k_1}(m)$$

메시지 m에 대한 Signcrypted 메시지 (c, r', s', K)를 Bob에게 전송한다.

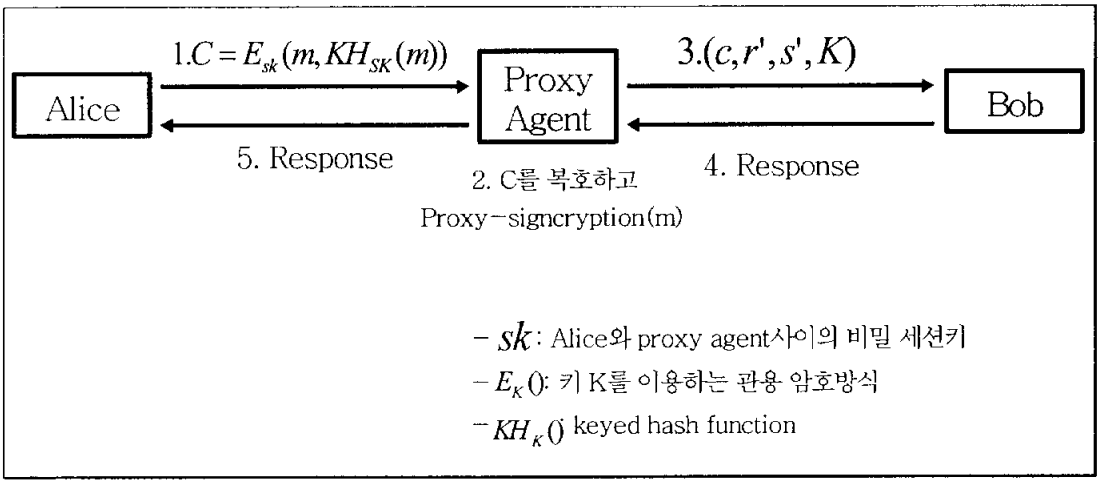
4) Proxy-signcryption의 검증

Bob은 $y_{AP} \equiv y_A \cdot K^K \bmod p$를 계산한 후, 자신의 비밀키를 이용하여 $k = (y_{AP} \cdot g^{r'})^{s' \cdot x_B} \bmod p$ 를 구한다. $k = k_1 \| k_2$로 나누고 다음과 같은 메시지를 복호한다.

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

단, $KH_{k_2}(m) = r'$ 인 경우에만 정당한 Signcryption 으로 받아들인다.

[그림 1] Proxy-signcryption 방식



Gamage 등이 제안한 Proxy-signcryption 방식[6]은 무선 통신에서 요구되는 기밀성, 인증성 및 유효성을 확보하고 있다. 그러나 이 방식은 서명 메시지 송신시 서명자와 대리 서명 agent간의 대리 서명용 키를 이용하게 된다. 따라서 서명자가 원할 경우 자신이 대리 서명 agent를 대신하여 정당한 서명을 생성할 수 있고, 대리 서명 agent 역시 서명자의 동의 없이 임의로 서명을 생성할 수 있다. 또한 이 방식은 서명자가 메시지 서명에 대한 부인봉쇄가 불가능하기 때문에 전자 상거래 등의 여러 분야에 적용하기에는 문제가 발생 될 수 있다.

4.2 대리인 보호형 Proxy-signcryption 방식

Gamage 등이 제안한 Proxy-signcryption방식은 대리인 비 보호형 대리 서명방식을 이용하므로 Alice가 임의의 메시지에 대해 Proxy-signcryption을 생성한 후 proxy agent가 생성한 것이라고 주장하는 경우에 제3자는 이를 판단할 수 없다는 문제점이 있다. 이를 해결하기 위하여 오수현 등은 대리인 보호형 대리 서명 방식을 이용하고, Alice가 전송한 메시지에 대해 부인봉쇄를 제공하기 위해 Asokan 등이 제안한 S^3(Server Supported Signatures)방식[13,14]을 이용하였다. 오수현 방식[7]은 다음과 같은 4개의 구성요소로 이루어진다.

- Alice : 계산 능력이 적은 단말기를 이용하여 인터넷을 통해 전자 상거래를 하려는 사용자
- 서명서버 G : 송신자 부인봉쇄를 제공하기 위해 후보 NRO(Non-Repudiation of Origin) 토큰을 생성하는 서버
- Proxy agent : 사용자의 위임에 의해 Proxy-signcryption을 생성하는 대리인
- Bob : 인터넷 쇼핑몰

[시스템 설정] 4.1장의 시스템 설정에 추가되는 항목들

- PK_A : 부인 봉쇄 토큰의 생성/검증에 사용하는 Alice의 루트 공개키,

$$PK_A = K_A^n = h^n(K_A)$$

- K_A^i : Alice의 $(n-i)$번째 공개키 ,

$$K_A^0 = K_A,\ K_A^i = h^i(K_A) = h(K_A^{i-1})$$

$K_A^0 = K_A,\ K_A^i = h^i(K_A) = h(K_A^{i-1})$

- $h()$: 일방향 해쉬 함수
- $Sigs()$: 서명 서버 G의 디지털 서명
- sk : Alice와 proxy agent 사이에 공유한 비밀키

[프로토콜]

1) Alice는 먼저 Bob에게 보내고자 하는 메시지 m_i를 생성하고 $(ID_A, h(m_i), i, K_A^i)$를 서명서버 G에게 전송한다.

2) 서명서버 G는 다음 식을 이용하여 Alice의 현재의 공개키 K_A^i를 검증한다.

$$h^{n-i}(K_A^i) = PK_A$$

3) 서명서버 G는 디지털 서명을 생성하여 다음과 같이 후보 NRO 토큰을 생성 한다.

$$Sigs(ID_A, h(m_i), i, K_A^i)$$

4) 서명서버 G는 생성한 후보 NRO 토큰을 Alice 에게 전송한다.

5) Alice는 서명을 검증하고 i를 $i-1$로 변경한 후 다음과 같이 실제 NRO 토큰을 생성한다.

$$NRO^i = (Sigs(ID_A, h(m_i), i, K_A^i), K_A^{i-1})$$

6) Alice는 Bob에게 전송할 메시지와 NRO 토큰을 Proxy agent와 사전에 공유한 비밀키를 이용 암호화하여 $C = E_{sk}(m_i, NRO^i)$를 Proxy agent 에게 전송한다.

7) Proxy agent는 암호문 C를 복호하고 NRO토큰을 저장한다. 그리고 사전에 위임받은 대리 서명용 키를 이용하여 m_i에 대한 대리인 보호형 Proxy- signcryption을 생성한다.

① Alice는 $x \in Z_q$를 선택하여 $K \equiv g^x \bmod p$를 계산하고 대리서명용 키 $x_{AP} \equiv (x_A + x \cdot K) \bmod p-1$를 생성하여 비밀리에 (x_{AP}, K)를 proxy agent에게 전송한다.

② Proxy agent는 $g^{x_{AP}} \equiv (y_A \cdot K^K) \bmod p$를 이용하여 자신이 받은 대리서명용 키가 정당한지 확인한 후, 다음과 같이 alternative proxy x_{AP}'를 생성한다.

$$x_{AP}' \equiv x_{AP} + x_P y_P \bmod q$$

③ Proxy agent는 $x' \in Z_q$를 선택하여 $k \equiv y_B^{x'} \bmod p$계산하고, $k = k_1 \| k_2$로 나누어 다음과 같이 메시지 m_i에 대한 Signcryption을 생성한다.

$$r' = KH_{k_2}(m_i)$$

$$s' \equiv (x_{AP}')^{-1} \cdot (x' - r') \bmod q$$

$$c = E_{k_1}(m_i)$$

(c, r', s', K)을 Bob에게 전송한다.

④ Bob은 $y_{AP}' = y_A \cdot y_P^{y_P} \cdot K^K \bmod p$를 계산하고, 비밀키를 이용하여 $k = ((y_{AP}')^{s'} \cdot g^{r'})^{x_B} \bmod p$를 구하고 $k = k_1 \| k_2$로 나눈 후 메시지를 복호한다.

$$m_i = D_{k_1}(c)$$

단, $KH_{k_2}(m_i) = r'$인 경우에만 정당한 Signcryption으로 받아들인다.

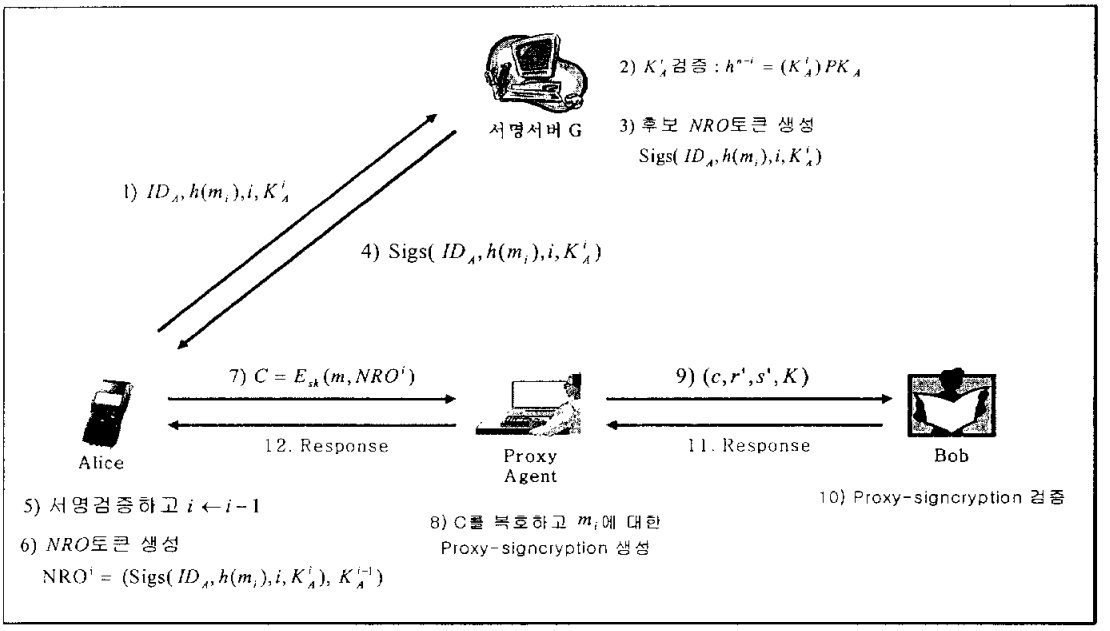
8) Proxy agent는 생성한 Proxy-signcryption (c, r', s', K)를 Bob에게 전송한다.

9) Bob는 Proxy-signcryption을 검증하여 Alice의 요구에 의해 Proxy agent가 보낸 메시지임을 확인한다.

10) Bob은 9)에 대한 응답을 Proxy agent에게 전송한다.

11) Proxy agent는 메시지가 성공적으로 전달되었음을 Alice에게 알려준다.

[그림 2] 대리인 보호형 Proxy-signcryption 방식



오수현 등이 제안한 방식은 Gamage등이 제안한 방식과 유사하게 서명의 생성과 같이 많은 계산량을 요구하는 부분은 서명서버와 Proxy agent에 의해 수행하고, Alice가 proxy agent에게 메시지를 전송하기 전에 서명서버로부터 해당 메시지에 대한 NRO토큰을 발급 받으므로 기존의 방식이 제공하지 못한 송신자 부인봉쇄 기능을 제공한다. 또한 송신자의 부인봉쇄 문제를 해결하기 위해 추가된 서명서버의 부정 검출기능도 함께 제공한다. 그러나 오수현 등이 제안한 방식은 아래와 같이 proxy agent와 Bob사이에 메시지의 안전성 유지가 어렵다는 문제점이 있다.[15]

① 가정 : 대리 서명키 x'_{AP}의 노출

② $k = ((y_{AP'})^{s'} \cdot g^{r'})^{x_B} = y_B^{x'_{AP} \cdot s' + r'}$

Bob 이외의 다른 사람이 키 k값을 계산할 수 있고, 이는 대리서명용 키 x'_{AP}에 대한 forward secrecy를 제공하지 못함을 의미하며, x'_{AP}는 특정인의 개인키가 아니므로 대리 서명인의 부주의로 각 개인의 개인키보다 노출 될 확률이 높다.

4.3 이동통신 환경에 적합한 효율적인 Proxy-signcryption 방식 제안

본 장에서는 이동 통신 환경에 응용하기 위하여 수신자 지정 서명 방식과 대리인 보호형 서명 방식을 적용하고, proxy agent에서 forward secrecy를 제공하는 개선된 Proxy-signcryption 방식을 제안한다.

[시스템 설정] 4.1, 4.2장의 시스템 설정에 추가되는 항목들

- x_A, x_B, x_P : Alice, Bob, Proxy agent의 비밀키
- y_A : $y_A \equiv g^{x_A} \bmod p$, Alice의 공개키
- y_B : $y_B \equiv g^{x_B} \bmod p$, Bob의 공개키
- y_P : $y_P \equiv g^{x_P} \bmod p$, Proxy agent의 공개키
- S : 위임서명자의 일회용 비밀서명정보
- T_i : time stamp(실시간 값)
- T' : Proxy agent가 암호문을 받는 시간
- ΔT : 채널의 최대한 지연시간

4.3.1 프로토콜

1) 위임정보 생성

이동 통신 단말기를 보유한 Alice는 proxy agent에게 서명생성을 위한 위임정보를 사전 계산(K는 위임정보 생성에 관여하지 않으므로 사전계산 가능)하여 휴대폰 단말기나 스마트카드에 저장한다. 이는 오프라인 상에서 이루어 질

수 있으므로 온라인 접속시에 연산 부하량과 시간을 단축시킬 수 있다.

$$x \in Z_q$$

$$R \equiv g^x \bmod p$$

$$K \equiv y_P^x \bmod p$$

$$S \equiv (x_A + x \cdot R) \bmod p-1$$

Alice는 이동 통신 단말기를 사용하지 않을 때 즉, 무선 네트워크를 사용하지 않거나 음성 통화를 하지 않는 빈 시간에 K, R, S를 계산하고 메시지를 전송하고자 할 때, 사전 저장되어 있는 K, R, S 값을 이용하여 다음과 같이 C를 계산한 후, Proxy agent에게 (R, C)를 전송한다.

$$C \equiv E_K(m \| S \| T_i)$$

Alice는 위의 과정을 반복 수행할 필요 없이 필요시 오프라인 상에서 사전 계산(K, R, S)하여 저장된 값을 이용하여 C를 1번만 수행하면 된다. 따라서 n번 반복 수행시 계산량은 C를 n번 수행한 계산량을 가진다. 또한 K에서 g^x를 계산하기 위해서는 proxy agent의 비밀 서명정보 x_P를 알아야 하므로 지정된 수신자만이 이를 복호할 수 있으며, S를 암호화한 후 C에 포함하여 송부하므로 공격자는 S를 알 수 없고, R을 이용하여 암호문 C를 복호해야만 S를 알 수 있다. 즉 외부에 공개되는 정보를 최소화 할 수 있다.

2) 위임정보의 확인 및 변환

Proxy agent는 수신된 정보를 기초로 세션키 K를 계산하고, 계산된 K를 이용하여 암호문을 복호화하여 time stamp(T_i)를 다음과 같이 검증한다.

$$K \equiv R^{x_P} \bmod p$$

$$m \| S \| T_i \equiv D_K(C)$$

$$T' - T_i \leq \Delta T$$

만약 검증이 만족되지 않으면 요청을 거절하고, 만족하면 위임 서명자의 정당성을 확인하고 다음과 같은 x_{AP}를 생성한다.

$$g^S \equiv (y_A \cdot R^R) \bmod p$$

$$x_{AP} \equiv S + x_P y_P \bmod q$$

3) Proxy agent에 의한 Signcryption 생성

$$x' \in Z_q$$

$$k \equiv h(y_B^{x'} \bmod p)$$

$$K' \equiv h(g^{x'} \bmod p)$$

$$r \equiv KH_K(m)$$

$$s \equiv x' - (x_P + x_{AP} \cdot r) \bmod q$$

$$c \equiv E_k(m)$$

Signcryption 메시지 (c, r, s, R)을 Bob에게 전송한다.

4) Proxy-signcryption 검증

Bob은 $y_{AP} \equiv y_A \cdot y_P^{y_P} \cdot R^R \bmod p$를 계산하고, 다음과 같이 메시지를 복호한다.

$$t_1 \equiv (y_{AP}^r \cdot y_P \cdot g^s) \bmod p$$

$$t_2 \equiv t_1^{x_B} \bmod p$$

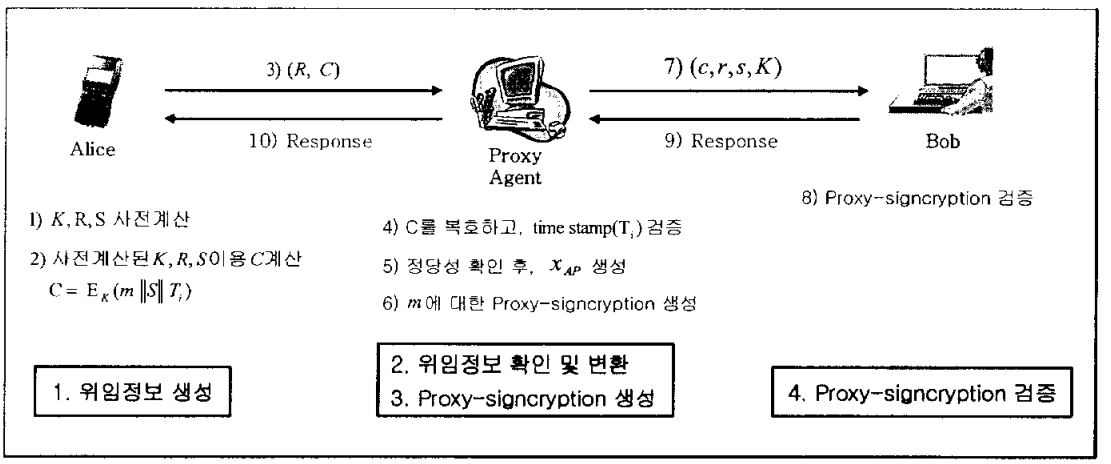
$$K' \equiv h(t_1)$$

$$k \equiv h(t_2)$$

$$m \equiv D_k(c)$$

단, $KH_K(m) \equiv r$인 경우에만 정당한 Signcryption으로 받아들인다.

[그림 3] 제안하는 효율적인 대리인 보호형 Proxy-signcryption 방식



제안방식에서 수신자 Bob이 y_{AP}를 계산하는 과정에서 Alice의 공개키 y_A와 proxy agent의 공개키 y_P를 동시에 사용하므로 Alice의 위임에 의해 proxy

agent가 생성한 Proxy-signcryption임을 확인 가능하다. 또한 s를 생성하는데 proxy agent의 비밀키 x_P가 사용되므로 이 값을 모르는 Alice는 정당한 Proxy-signcryption을 생성할 수 없다.

4.4 제안방식의 고찰

(1) 서명자 기밀성 확보

제안 방식은 수신자 지정 대리 서명방식을 적용함으로 수신자를 보호할 수 있다. 즉, 메시지의 진위여부를 판단하기 위해서는 수신자의 도움 없이는 불가능하다. 따라서 서명의 검증기능을 수신자의 통제하에 두어 수신자만이 이를 복호할 수 있으므로 기밀성을 보장한다고 할 수 있다.

(2) 인증성 제공

이동 통신에서 투명성을 높이기 위해서는 인증성 제공은 필수적이다. 제안 방식은 메시지의 송·수신시 출처가 누구이며, 전송 도중 제 3자로부터의 위조 및 변경내용 확인이 가능하므로 인증성을 제공한다.

(3) 부인봉쇄 기능

서명 생성시 proxy agent는 자신의 비밀 정보와 서명자의 위임 서명 정보를 함께 포함하여 대리 서명을 수행하게 되므로 Alice의 서명 생성에 대한 부인 방지가 가능하다.

(4) 유효성 획득

무선 이동 통신은 메시지 송·수신을 위해 일반 네트워크에 비해 상대적으로

계산능력이 떨어지는 무선 단말기를 사용한다. 따라서 서명 생성시 상대적으로 계산능력이 뛰어난 proxy agent를 이용하여 서명을 수행함으로 유효성을 확보할 수 있다.

(5) 안전성 제공

무선 이동 통신에서 메시지 송·수신에 참여하는 개체에 의한 위조 및 변조가 불가능해야 한다. 따라서 제안방식에서는 위임정보 전송시 Alice는 일회용 비밀 서명 정보를 제공하며, K에서 g^x를 계산하기 위해서는 proxy의 비밀 서명정보 x_P를 알아야 하므로($K \equiv (g^x)^{x_P} \bmod p$), 지정된 수신자만이 이를 복호할 수 있다. 또한 S를 암호화한 후 C에 포함하여 송부($C \equiv E_K(m \| S \| T_i)$)하므로 공격자는 S를 알 수 없고, $R \equiv g^x \bmod p$을 이용하여 암호문 C를 복호해야만 S를 알 수 있으므로 외부에 공개되는 정보를 최소화하고 있다. 서명 생성시 proxy agent는 자신의 비밀 정보를 생성하여 서명을 전송하므로 Alice 및 proxy agent에 의한 불법적인 서명 생성은 불가능하고, proxy agent에서 forward secrecy를 제공하는 개선된 Signcryption을 사용하므로 메시지에 대한 안전성(security)을 제공한다. Unsigncryption 과정에서

$$(y_{AP}{}^{r} \; y_P \; g^s)^{x_B} = y_B{}^{x_{AP} r + x_P + s}$$

이 성립하나, 대리 서명키 x_{AP}가 드러나는 경우에도 대리 서명자의 개인키 x_P를 알지 못하는 사람은 키를 계산할 수 없고, x_P를 알지 못하는 사람은 대리 서명을 수행할 수 없으므로 대리 서명 위임자의 위조 공격으로부터 대리 서명자를 보호할 수 있으므로 forward secrecy를 제공한다. 즉 x_{AP}와 x_P가 모두 드러나면 키를 계산할 수 있지만, 두 키가 모두 드러날 경우는 x_{AP}만 드러날 경우에 비해 매우 낮은 확률로 발생한다.

기존방식은 Bob의 unsigncryption 과정에서 수신자의 개인키(x_B) 정보가 필요하기 때문에 제3자에게 Proxy-signcryption의 정당성 여부를 증명하기 위해서는 영지식 증명(zero knowledge proof)과 같은 복잡한 계산 과정을 필요로 하거나 서명 서버와 같은 별도의 구성요소를 필요로 한다. 따라서 제안방식은 Bao & Deng[16]이 제안한 공개키에 의한 direct verifiability를 제공한다. 즉, Bob이 제3자에게 (m, r, s)를 보내면 제3자는 proxy agent의 공개키를 이용하여 $K' \equiv h((y_{AP}^{r} \cdot y_P \cdot g^{s}) \bmod p)$를 계산하고, $r \equiv KH_K(m)$이 성립하면 proxy agent가 서명을 생성하였음을 확인할 수 있으므로 공개키에 의한 direct verifiability를 제공한다.

(6) Alice의 계산량

제안하는 방식을 이동통신 환경에 적용할 경우 서명 후 암호화 하는 방식에 비해 단말기 사용자에게 요구되는 계산량을 감소시킬 수 있다. 또한, 모듈라 멱승과 같은 복잡한 연산을 오프라인에서 사전계산(pre-computation)을 통하여 감소시킴으로써 휴대용 단말기와 같은 낮은 연산 처리 능력을 가지는 시스템에 적합하도록 하였다. 즉, 제안방식은 Alice가 1번 수행시 ENC 1번의 계산량을 가지면서 forward secrecy 제공과 노출되는 정보를 최소화 하고, time stamp(T_i)를 사용하므로 실시간 값으로 검증이 가능하다.

Alice에게 요구되는 계산량을 기존의 방식과 비교하면 [표 11]과 같다.

[표 11] 각 방식별 특성 비교 분석

구 분	Proxy-signcryption 방식	대리인 보호형 Proxy-signcryption 방식	제안방식
서명자 기밀성	○	○	○
인증성	○	○	○
부인봉쇄	×	○	○
유효성	○	○	○
안전성	×	×	○
forward secrecy	×	×	○
구성요소 개수	3	4	3
Alice의 계산량	ENC: 1 EXP: 1	ENC: 1 EXP: 1	ENC: 1 [EXP: 2]

■ 단, EXP: 모듈라 멱승이고, ENC: 관용암호방식, [•] : 오프라인 연산

■ 구성요소 개수 : 각 방식에 사용되는 구성요소 즉, Alice, proxy agent, Bob, 서명서버 G

5. 결 론

지금까지 2장에서는 이산대수에 기초한 서명방식을, 3장에서는 기존 Signcryption 방식들이 부인봉쇄 과정에서 기밀성을 보장하지 못하는 문제점을 지적하고, 이를 해결한 개선된 Signcryption 방식을 소개하였으며, 이는 일반적인 서명 후 암호화하는 방식보다 효율적이며, 기존의 Signcryption 방식과 계산량과 통신량에서 크게 차이 없이 Signcryption을 생성하고, 기존의 방식에서 발생할 수 있는 문제점을 해결하였다. 또한 4장에서는 Gamage 방식이 대리 서명방식과 Signcryption 방식의 적용을 통해 기밀성, 인증성 및 유효성을 보장하고 있으나, 송신자와 대리 서명 Agent의 부정을 방지 못함으로서 부인 봉쇄 및 안전성을 만족하지 못하고, 오수현 등이 제안한 방식 또한 기밀성, 인증성, 유효성 및 부인봉쇄는 제공하고 있으나, proxy agent에서 forward secrecy를 제공하지 못하므로 메시지에 대한 안전성을 제공하지 못하는 문제점이 있음을 살펴 보았다.

따라서 본 논문에서는 수신자 지정 서명 방식과 대리인 보호형 대리 서명 방식 및 proxy agent에서 forward secrecy를 제공하여 원 서명자인 Alice도 proxy agent를 대신하여 정당한 Proxy-signcryption을 생성할 수 없고, Alice는 자신이 전송한 메시지에 대해 그 사실을 부인할 수 없도록 하였다. 또한 proxy agent는 사용자의 요구가 있는 경우에만 Proxy-signcryption을 생성할 수 있는 개선된 Proxy-signcryption 방식을 제안하였다. 이를 통해 제안방식은 기밀성과 효율성을 획득하고 있으며, 동시에 인증성, 부인봉쇄 및 안전성을 만족하고 있으므로 계산 능력이 낮은 이동 통신환경에 적용할 수 있다.

네트워크와 휴대용 단말기가 급속도로 발전함에 따라 사용자들이 이동 중에 휴대용 단말기를 이용하여 인터넷 전자 상거래등 여러종류의 서비스를 이용하는 경우가 많아지는 점을 감안할 때 이동 통신상에서 기밀성, 인증 및 부인봉쇄를 제공하는 효율적인 전자 서명 방식의 연구는 매우 중요하며 앞으로도 계속되어야 할 것이다.

[참고문헌]

[1] Y. Zheng, "Digital Signctyption or How to Achieve Cost(Signature & Enctyption) << Cost(Signature)+Cost(Enctyption)", Advances in Cryptology - CRYPTO'97, Springer-verlag, LNCS 1294, pp.165-179, 1997.

[2] Y. Zheng, "Signcryption and Its Applications in Efficient Public Key Solutions", Proceedings of 1997 Informatio Security Workshop (ISW'97), LNCS 1397, pp.291-312, Springer-verlag, 1998.

[3] S.J. Kim, S.J. Park and D.H. Won, "Nominative Signatures." Proc. of ICEIC'95, pp.Ⅱ-68-Ⅱ-71, 1995.

[4] M. Mambo, K. Usuda and E. Okamoto, "Proxy Signature : Delegation of the Power to Sign message", IEICE Transaction on Fundamentals, E79-A(9): 1338-1354, 1996.

[5] M. Mambo, K. Usuda and E. Okamoto, "Proxy Signatures for Delegation Signing Operation", Proc. Third ACM Conference on Computer and Communications Security, pp.48-57, 1996.

[6] C. Gamage, J. Leiwo and Y. Zheng, "An Efficient Scheme for Secure Message Transmission Using Proxy-Signcryption", Proc. of the 22nd Australasian Computer Science Conference, Jan. 1999.

[7] 오수현, 김현주, 원동호, "이동통신 환경에서의 전자 상거래에 적용할 수 있는 Proxy - Signctyption 방식", 한국정보보호학회 논문지, 제 10권 제 2호, 2000.6.

[8] T. ElGamal, "A public key cryptosystem and a Signature Scheme based on Discrete Logarithms", IEEE Transactions on Information Theory, IT-31(4) : 469-472, 1985.

[9] C.P. Schnorr "Efficient Signature Generation for Smart Cards", Advance in Cryptology-CRYPTO '89 Proceeding.

[10] H. Petersen and M. Michel. "Cryptanalysis and Improvement of Signcryption Schemes" IEE Computer and Digital Techniques 1998.

[11] H. Petersen and M. Michel. "Cryptanalysis and Improvement of Signcryption Schemes" IEE Computer and Digital Techniques 1998.(Revised Version)

[12] 김성덕, "다자전송 효율성을 가진 Signcryption 방식" 한국정보보호학회 논문지, 제 10권 제 3호, 2000.9.

[13] N. Asokan, G. Tsudik and M. Waidner, "Server Supported Signatures", Proc. of the Fourth European Symposium on Research in Computer Security (ESORICS), LNCS 1146, pp.131-143, Springer-Verlag, September 1996.

[14] N. Asokan, G. Tsudik and M. Waidner, "Server Supported Signatures", Journal of Computer Security, November 1997.

[15] 정희윤, 이동훈, 임종인, "Forward secrecy를 제공하는 Signcryption" ITRC Forum pp.(D1)11-44, 2001.

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

감사의 글

배우고 싶다는 생각에 시작한 대학원 생활이 어느듯 3년이라는 시간이 흘러 졸업이라는 문턱에 섰습니다. 뒤돌아보면 지난시간이 어떻게 지나갔는지 모를 많큼 너무나도 아쉬고 그리운 나날들을 보냈습니다. 직장생활과 학업을 병행하는 핑계로 연구실 세미나와 모임에 적극적이지 못했던 것이 아쉬움으로 남습니다.

아쉬움을 뒤로한채 졸업이라는 시간까지 아끼고 사랑하고 말없이 도움과 격려를 보내주신 모든 분들께 진심으로 감사드립니다.

가장먼저, 여러 가지 어려움 속에서도 졸업할수 있게 하신 하나님께 감사드립니다. 그리고 부족하고 부족한 저에게 언제나 조언과 격려로 한결같이 지도해 주신 박지환 교수님께 진심으로 감사의 마음을 전합니다. 또한 심사를 맡아주신 김영봉 교수님, 조성진 교수님, 언제나 깊은 관심으로 지켜봐 주신 윤성대 교수님, 정순호 교수님, 박승섭 교수님, 여정모 교수님, 이경현 교수님, 김창수 교수님, 박만곤 교수님, 박흥복 교수님께 감사 드립니다.

이 논문을 쓸 수 있게 도움을 주신 성한씨, 현호씨, 소진씨 그리고 멀티미디어 보호 및 응용 연구실의 일반대학원, 교육대학원, 산업대학원 식구들에게 감사를 드립니다. 대학원 생활에 깊은 애정과 많은 즐거움, 좋은 추억을 가질수 있도록 도움을 주신 산업대학원 '00학번 동기들과 저를 알고 격려해 주신 모든 분들께 감사를 드립니다.

직장과 대학원 생활이 바쁘다는 핑계로 가정에 소홀함에도 남편을 이해해주고 말없이 기도와 격려를 아끼지 않은 사랑하는 아내 미현과 딸 윤진, 그리고 배속에 있는 둘째, 멀리서나마 아들이 건강하고 잘되기만을 늘 기도하해 주시는 어머니, 동생 제호에게 감사드리며 이 논문을 바치고자 합니다.