情報保護學會論文誌
第13券 第4號, 2003. 8

사용자 익명성을 위한 Distributed Signcryption*

곽 동 진 **, 하 재 철***, 문 상 재**

A Distributed Signcryption for User Anonymity

Dong-Jin Kwak**, Jae-Cheol Ha***, Sang-Jae Moon**

요 약

Distributed signcryption은 특정한 집단에게 서명과 암호가 동시에 된 메시지를 분산시키는 목적으로 설계되었다. 이 signcryption은 unsigncryption 시, 서명자의 공개키를 미리 알고 있어야 서명에 대한 검증을 할 수 있기 때문에 서명자의 익명성을 요하는 전자상거래 응용에서는 부적합하다. 따라서 본 논문에서는 계산량의 증가없이 서명자의 익명성과 신뢰기관에 의한 부인방지(non-repudiation) 기능을 동시에 제공하는 distributed signcryption을 제안하고, 그 안전성과 연산량을 분석한다. 이에 더하여 제안하는 signcryption의 확장으로 집단서명(group signature)의 성질을 가지는 효율적인 집단 signcryption을 제안한다.

ABSTRACT

Distributed signcryption was specially designed for distributing a signcrypted message to a designated group. Since a verifier of this signcryption should know the signer's public key in advance, it cannot provide the signer's anonymity. This study adds anonymity and non-repudiation by trusted party to the distributed signcryption with almost the same computational load. We also analyze security and computational loads of the proposed scheme. In addition, we extend our scheme to an efficient group signcryption.

keyword : *전자상거래 보안, 디지털 서명, singcryption, 집단서명(group signature), 익명성*

Ⅰ. 서 론

디지털 서명과 암호를 동시에 하면서 계산상·통신상의 효율성을 극대화 시키는 방법으로 signcryption scheme이 개발되었다.[1] 기밀성과 사용자 인증이 동시에 요구되는 응용이 많아짐에 따라 그 사용이 점점 더 많아 질 것으로 예상된다. 응용의 요구조건이 다양해짐에 따라 signcryption은 정해진 한명의 검증자를 대상으로 서명 및 암호를 하는 것이 아니라 다수의 검증자를 상대로 혹은 집단을 상대로 자신이 소속된 집단을 대표해서 서명 및 암호를 하는 방식도 제안되고 있다.[2~4]

예를 들어, 비공개 입찰(sealed-bid)의 경우 입찰을 하려는 사람은 각각의 입찰금액을 기입한 후 서명과 암호를 해서 입찰에 응한다. 그리고 공개적으로 최종 입찰금액을 서로가 공정히 그 결과를 수용할 수 있는 프로토콜을 설계하려할 때, 다수 혹은 집단을 대상으로 하는 signcryption 방식들이 유용할 것이다.

본 논문는 distributed signcryption[4]이 서명자의 익명성 결핍으로 인해 익명성을 요하는 전자상거래 응

* 본 논문은 2003년도 영남지부 학술대회 우수논문임.
본 연구는 대학 IT 연구센터 육성·지원 사업의 연구 결과로 수행되었음.
** 경북대학교 대학원 전자공학과(neverdid@palgong.knu.ac.kr, sjmoon@knu.ac.kr)
*** 나사렛대학교 정보통신학과(jcha@kornu.ac.kr)

용에서는 부적합하다는 것을 지적한다. 이 지적을 바탕으로 계산량의 증가없이, 서명자의 익명성과 부인방지 기능을 동시에 제공하는 새로운 distributed signcryption을 제안하고 그 안전성과 연산량을 분석한다. 이에 더하여, 제안하는 signcryption의 확장으로 집단서명(group signature)의 성질을 가지는 집단 signcryption을 제안한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 기본이 되는 이론을 먼저 살펴보고, 3장에서는 기존의 distributed signcryption에 대해 언급한다. 제 4장에서 익명성이 추가된 새로운 프로토콜을 제시하며, 그 안전성과 연산량을 분석한다. 5장에서는 제안된 프로토콜을 기반으로 하여 새로운 집단 signcryption으로 확장한 후, 마지막으로 결론을 맺는다.

II. Preliminaries

이 장에서는 기존의 Signcryption과 대리자 복호(delegated/proxy decryption)에 대해 간단히 살펴보기로 한다. 본 논문에서는, p를 큰 소수, Z_p^*를 위수 q ($q=p-1$)인 곱셈군(multiplicative group), $g(g\in Z_p^*)$를 생성자(generator), 그리고 (sk_a, pk_a)와 (sk_b, pk_b)는 각각 Alice와 Bob의 비밀키 · 공개키 쌍을 나타낸다.

2.1 Signcryption

Signcryption scheme은 기존의 서명 후 암호를 하는 방식에서 서명과 암호를 동시에 하면서 필요한 계산상의 로드를 줄이는 암호학적 방법으로 *Crypto'97*에 Y. Zheng[1]에 의해 처음 소개 되었다. 이 signcryption은 NIST의 DSS[6]를 기반한 약간의 수정으로 계산상의 효율성을 꾀하고 있으며, SDSS의 두 가지 형태로 나타낼 수 있다. 이 후, 다양한 형태의 변형된 signcryption scheme과 그 응용이 나오고 있다.[7~10] Alice가 Bob에게 메시지 m을 전송하려고 할 때, SDSS1과 SDSS2를 포함한 전체 signcryption은 다음과 같이 표현될 수 있다. 여기서 $H_k(\cdot)$는 키 k인 해쉬함수, $E_k(D_k)$는 대칭키 암호 혹은 복호를 나타낸다.

Alice[서명자]

$x\in Z_q$

$k=pk_b^x \bmod p=k_1\|k_2$

$r=H_{k_2}(m)$

$s=x(r+sk_a)^{-1}\bmod q$ if SDSS1

$=x(1+r\cdot sk_a)^{-1}\bmod q$ if SDSS2

$c=E_{k_1}(m)\rightarrow(c,r,s)\rightarrow$

Bob[검증자]

$k=(pk_a\cdot g^r)^{s\cdot sk_b}\bmod p$ if SDSS1

$k=(pk_a^r\cdot g)^{s\cdot sk_b}\bmod p$ if SDSS2

$m=D_{k_1}(c)$

$r?=H_{k_2}(m)$

2.2 대리자 복호 방식

대리자 복호 혹은 분산암호(distributed encryption)는 *Cryptography & Coding '99*에서 Yi Mu[5]에 의해 발표된 공개키 방식으로 하나의 공개키로 암호화한 암호문을 여러 개의 다른 복호키로서 복호를 함으로써, 특정 당사자의 부재 시 그 당사자를 대리할 수 있는 같은 집단의 다른 사람도 암호문을 복호할 수 있도록 하는 공개키 방식이다.

2.2.1 초기 설정

관리자는 n개의 비밀키 $x_i\in Z_q$를 생성하여 그 상수 $\{\alpha_i\}\in Z_q$를 식 (1)과 같이 계산한다.

$$f(x)=\prod_{i=1}^{n}(x-x_i)=\sum_{i=0}^{n}\alpha_i x^i \tag{1}$$

생성자 g를 정의한 뒤 $g^{\alpha_i}\bmod p$를 g_i로 치환하면, 식 (1)로부터 식 (2)와 같은 방정식을 얻는다.($\because j\in\{1,\ldots,n\}$, $f(x_j)=0$)

$$F(x_j)=g^{f(x_j)}=\prod_{i=0}^{n}g_i^{x_j^i}=1\bmod p \tag{2}$$

공개키를 $\{g_i\}$로 하면, 공개키의 공개 시 비합법적인 제3자에 의한 공개키 조작이 가능하다. 따라서 전체 안전도에 문제가 생길 수 있음으로 다음과 같이 새로운 α_i'를 다음과 같이 설정한다.[5]

$$\alpha_0'=\alpha_0,\ \alpha_1'=\ldots.=\alpha_{n-1}=\sum_{i=1}^{n-1}\alpha_i,\ \alpha_n'=\alpha_n$$

그리고 $\beta_i\leftarrow g^{\alpha_i'}$, $A_j\leftarrow\sum_{i=1.e=1.i\neq e}^{n-1}\alpha_e x_j^i$라 하면 모

든 $x_j(j=1,2,\ldots n)$에 대해서 식 (2)는 식(3)을 여전히 만족한다.

$$F'(x_j)=g^{-A_j}\prod_{i=0}^{n}\beta_i^{x_j^i}=1 \bmod p \quad (3)$$

다시 관리자는 난수 $\gamma\in Z_q$를 선택하여 γ^{-1}와 $\rho_j=-\gamma A_j \bmod q$를 계산한다. 이를 바탕으로, 관리자는 $n+2$의 tuple인 $\{\beta_0,\ldots,\beta_{n+1}\}=\{\beta_0,\ldots,\beta_n,g^{\gamma^{-1}}\}$를 그 집단의 공개키로 하고 집단 내 n명의 사용자에게 각각 비밀키쌍으로 (x_j,ρ_j)를 보내준다.

2.2.2 암호 및 복호

Alice가 메시지 m을 암호하기 위해 임의의 세션키 $k\in Z_q$를 선택하고, m을 해쉬함수에 대입해서 $w=H(m)$를 계산한다. 그리고 보내려고 하는 그룹의 공개키를 바탕으로 다음과 같은 암호문 $c=(c_1,c_2)$를 만들어 그 집단에 전송한다.

$$c_1\leftarrow\{a_0,\ldots,a_{n+1}\}\leftarrow\{g^k\beta_0^w,\beta_1^w,\ldots,\beta_{n+1}^w\}$$
$$c_2=mg^k \bmod p$$

Alice가 보낸 집단에 소속된 n명 누구든지 다음과 같은 방식으로 복호를 할 수 있다.

$$c_1'\leftarrow a_0\Big(\prod_{i=0}^{n}a_i^{x_j^i}\Big)a_{n+1}^{\rho_j}=g^k\Big(\prod_{i=0}^{n}g^{a_i w x_j^i}\Big)$$
$$=g^k g^{wf(x_j)}=g^k \bmod p$$
$$m=c_2/c_1'=(mg^k)/g^k$$

Ⅲ. 기존의 distributed signcryption

3.1 Distributed signcryption

대리자 복호방식을 기반으로 여러 명으로 구성된 집단이 검증 및 복호를 동시에 할 수 있는 signcryption을 distributed signcryption라 한다.[4] 먼저 집단의 관리자(manager)가 대리자 복호의 초기설정과 같이 집단 구성원들의 비밀정보 $\{(x_j,\rho_j)\}$를 생성 후 나누어 주고, 집단의 공개키 $\{\beta_0,\ldots,\beta_{n+1}\}$를 공개한다. Alice가 Bob이 속해있는 특정집단에게 서명 및 암호 기능을 한 메시지 m을 보내고 싶다고 가정하면 다음과 같은 과정으로 (c_1,c_2,r,s)를 생성한 후 전송한다.

$$x\in Z_q$$
$$k=g^x \bmod p=k_1\|k_2$$
$$r=H_{k_2}(m)$$
$$s=x(rk+sk_a)^{-1} \bmod q \text{ if SDSS1}$$
$$=x(k+r\cdot sk_a)^{-1} \bmod q \text{ if SDSS2}$$
$$w=H(m)$$
$$c_1\leftarrow\{a_0,\ldots,a_{n+1}\}$$
$$=\begin{cases}\{g^{kr}\beta_0^w,\beta_1^w,\ldots,\beta_{n+1}^w\} & \text{if } SDSS1\\ \{g^k\beta_0^w,\beta_1^w,\ldots,\beta_{n+1}^w\} & \text{if } SDSS2\end{cases}$$
$$c_2=E_{k_1}(m)$$

Bob 혹은 그 집단의 구성원이 메시지에 대한 복원 및 검증을 원한다면, Alice에게 받은 (c_1,c_2,r,s)와 그의 공개키 pk_a와 자신의 비밀정보 (x_j,ρ_j)를 가지고 다음과 같은 과정으로 먼저 k를 복원 및 동시에 검증한다.

$$k\leftarrow(pk_a a_0(\prod_{i=1}^{n}a_i^{x_j^i})a_{n+1}^{\rho_j})^s \text{ if } SDSS1$$
$$k\leftarrow(pk_a^r a_0(\prod_{i=1}^{n}a_i^{x_j^i})a_{n+1}^{\rho_j})^s \text{ if } SDSS2$$

k의 복원 후 메시지를 복호하여 원하는 메시지 m을 얻는다.

3.2 확장된 distributed signcryption

위의 distributed signcryption은 집단의 구성원이 그 집단을 대표해서 서명 및 암호를 할 수 있는 집단 signcryption으로 확장될 수 있다. 먼저 두 집단 G_A와 G_B가 있고, Alice는 G_A, 그리고 Bob은 G_B에 속해 있다고 가정한다. Alice와 Bob이 각각 $f(x_a)=0 \bmod p$와 $f(x_b)=0 \bmod p$를 만족시키는 비밀정보 x_a와 x_b를 가지며, 3.1에서와 같은 방법으로 관리자는 G_A의 집단 공개키 $\{\beta_0^*,\cdots,\beta_{n+1}^*\}$ 그리고 G_B의 집단 공개키 $\{\beta_0,\cdots,\beta_{n+1}\}$을 만든다. Alice가 G_A를 대표해서 Bob이 속한 G_B에게 메시지 m을 보내고 싶다면 다음과 같은 과정으로 $(c_1,c_2,c_3,r,s_1,\cdots,s_n)$을 G_B에 보낸다.

$$x\in Z_q$$

$k = g^x \bmod p = k_1 \| k_2$

$w = H(m)$

$u_j \leftarrow \beta_j^{*w}$ for j=1,...,n

$r = H_{k_2}(m)$

$s_j = x(x_a^j - ru_j) \bmod q$ for j=1,...,n

$c_1 \leftarrow \{a_0, \ldots, a_{n+1}\} \leftarrow \{g^{kr}\beta_0^w, \beta_1^w, \ldots, \beta_{n+1}^w\}$

$c_2 \leftarrow \{a_0^*, u_1, \ldots, u_n, a_{n+1}^*\}$

$\leftarrow \{g^{x-rk}\beta_0^{*w}, u_1, \ldots, u_n, \beta_{n+1}^{*w\rho_a}\}$

$c_3 = E_{k_1}(m)$

G_B에 속한 Bob이 $(c_1, c_2, c_3, r, s_1, \cdots, s_n)$을 받은 후, 다음과 같은 과정으로 세션키를 복원하여 복호함으로써, signcryption을 검증한다.

$$k = [a_0(\prod_{i=1}^{n} a_i^{x_b'})a_{n+1}^{\rho_b}][a_0^*(\prod_{j=1}^{n} u_j^{ru_j'}\beta_j^{*s_j'})a_{n+1}^*]$$

$$= [g^{kr}(\prod_{i=0}^{n}\beta_i^{wx_b'})][g^{x-rk}(\prod_{j=0}^{n}\beta_j^{*wx_a'})]$$

$$= g^x \bmod p$$

Bob은 복원한 세션키로서 a_0와 a_0^*가 합당한 값인지를 검증할 수도 있다.

Ⅳ. 익명성을 가진 distributed signcryption

이 장에서는 사용자 익명성을 제공하는 새로운 distributed signcryption을 제안한다. 기존의 signcryption이 복호 및 검증 시, 반드시 signcryption한 당사자의 공개키를 미리 알고 있어야 그 서명에 대한 검증이 가능하다. 이런 상황은 사용자 익명성을 요구하는 응용에서는 사용할 수 없다. 이런 문제점을 고려하여 다음과 같은 프로토콜을 설계할 수 있다.

4.1 제안된 프로토콜

집단의 관리자는 기존 signcryption과 같이 초기 설정을 하고 나머지 설정도 기존방식과 동일하다고 가정하면, 다음과 같은 방식으로 Alice는 (c_1, c_2)를 생성하여 자신의 익명성을 Bob이 소속된 특정 집단을 제외한 제3자로부터 보장 받을 수 있다. 여기서 $Cert_a$는 Alice의 공개키 pk_a를 포함한 인증서이다.

$x \in Z_q$

$k = g^x \bmod p = k_1 \| k_2$

$r = H_{k_2}(m)$

$s = x(r + sk_a)^{-1} \bmod q$ if SDSS1

$= x(1 + r \cdot sk_a)^{-1} \bmod q$ if SDSS2

$w = H(m)$

$c_1 \leftarrow \{a_0, \ldots, a_{n+1}\} = \{k\beta_0^w, \beta_1^w, \ldots, \beta_{n+1}^w\}$

$c_2 = E_{k_1}(m \| r \| s \| Cert_a)$

Bob 혹은 그 집단의 구성원은 Alice에게 받은 (c_1, c_2)와 자신의 비밀정보 (x_j, ρ_j)를 가지고 다음과 같은 과정으로 먼저 k를 복원 후 검증한다.

$k \leftarrow a_0(\prod_{i=1}^{n} a_i^{x_j'})a_{n+1}^{\rho_j} = g^x \bmod p = k_1 \| p_2$

$D_{k_1} = m \| r \| s \| Cert_a$

$r? = H_{k_2}(m)$, $g^x? = (pk_a \cdot g^r)^s$ if SDSS1

$g^x? = (g \cdot pk_a^r)^s$ if SDSS2

4.2 안전성 및 연산량 분석

4.2.1 위조방지

악의를 지닌 검증자가 메시지 m과 서명에 관련된 $r, s, Cert_a$를 알 수 있기 때문에 sig- ncryption을 위조하는데 가장 유리한 조건을 가지고 있다. 따라서 이 검증자의 입장에서 위조 가능한 경우의 수를 고려해 보는 것이 가장 현실적이다. 이 검증자가 위조에 성공하기 위해서는 서명값을 만족시키는 또 다른 메시지 m'를 찾거나 혹은 새로운 $m' \| r' \| s' \| Cert_a'$를 찾아야한다. 전자의 경우 $H_k(\cdot)$가 난수발생기의 역할을 함으로 r에 합당한 m'를 찾기 불가능하고, 후자의 경우 위조자가 m'와 r'값은 계산할 수 있지만 sk_a를 모르는 상황에서 합당한 s'값을 생성할 수 없다.

4.2.2 부인방지

만약 분쟁이 발생했을 시 Bob은 제 3의 신뢰기관에 자신이 복호한 $D_{k_1} = m \| r \| s \| Cert_a$를 전송한다. 이 메시지를 받은 신뢰기관은 받은값들을 바탕으로 다음과 같이 k를 복구하여 검증하므로, 부인방지 기능이 제공된다고 볼 수 있다.

$k=(pk_a \cdot g^r)^s = k_1 \| k_2$ if SDSS1

$k=(g \cdot pk_a^r)^s = k_1 \| k_2$ if SDSS2

$r? = H_{k_2}(m)$

4.2.3 익명성 및 기밀성

c_2안에 서명자의 공개키가 같이 암호화가 되어 있음으로 이 암호를 복호하도록 정해져 있는 당사자가 아니고서는 누가 이 메시지를 보내었는지를 알 수가 없다. 따라서 사용자 익명성이 보장된다고 볼 수 있다.

Alice가 보내는 전체 메시지 (c_1, c_2)가 전부 대칭키나 공개키 암호방식으로 메시지, 서명값, 그리고 세션키가 암호화되어 있다. 따라서,부가적인 연산으로 Zheng의 signcryption[1]보다 강한 기밀성을 제공한다. 기존의 distributed signcryption와 비교하면 동일한 기밀성을 제공한다고 볼 수 있다.

4.2.4 연산량 분석

기존의 signcryption과 제안한 프로토콜을 모듈라 연산의 관점에서 연산량을 비교하면 [표 1]과 같다. 대칭키 암·복호 및 해쉬는 모듈라 연산에 비해 적음으로 무시하고 모듈라 곱셈,역원 및 멱승 연산을 중점으로 연산량을 분석하였다. 서명 시 익명성을 제공하지 않는 기존의 signcryption보다 다소 우수함을 볼 수 있다. 그러나, 집단의 크기 n이 늘어남에 따라 거의 비슷한 연산량을 가진다고 볼 수 있다.

〔표 1〕 연산량 분석

연산량		signcryption		unsigncryption	
		SDSS1	SDSS2	SDSS1	SDSS2
기존 프로토콜	I_q	1	1	-	-
	M_q	3	2	-	-
	M_p	1	1	n+2	n+2
	E_q	-	-	n-1	n-1
	E_p	n+4	n+4	n+2	n+3
제안한 프로토콜	I_q	1	1	-	-
	M_q	1	2	-	-
	M_p	1	1	n+2	n+2
	E_p	-	-	n-1	n-1
	E_p	n+3	n+3	n+3	n+3
I_q는 mod q에서 역원, M_q(M_p)는 mod q(p)에서 곱셈, E_p(E_q)는 mod p(mod q)에서 멱승이다.					

V. 집단 signcryption으로의 확장

Yi Mu 역시 distributed signcrypion을 집단 signcryption으로의 확장[4]을 시도하였지만, 매우 복잡하고 집단 관리자가 각 구성원의 비밀정보를 생성하여 나누어 주는 방식임으로 그 관리자를 대상으로 공격이 성공할 시 그 파급효과가 엄청나다. 그리고 기존의 프로토콜과 마찬가지로 unsigncryption시 송신집단의 공개키를 미리 알아야 된다는 약점을 가지고 있다. 이 장에서는 이런 점들을 개선한 새로운 집단 signcryption을 제안한다.

5.1 초기 설정

각 집단의 구성원은 각각 자신의 비밀정보 x_j를 생성하여 관리자가 정한 $a \in Z_p^*$를 바탕으로 y_j $(= a^{x_j} \bmod p)$를 자신의 멤버쉽키로 생성하여 안전한 채널로 자신의 집단 관리자에게 y_j를 전송한다. 비밀정보 x_j 대신 y_j를 사용함으로써, 각 구성원은 자신의 관리자에게 비밀정보를 알려줄 필요가 없음으로, 각 구성원의 비밀정보 누출을 방지한다. 그리고 관리자에게 각 구성원의 비밀정보가 집중되는 것을 막는다. 각 집단 관리자는 구성원들의 y_j를 받아서 관리자의 RSA 서명 $v_j (= y_j^{e_{GM}^{-1}} \bmod n_{GM})$를 생성한다.[11] 여기서 e_{GM}은 관리자 RSA 서명 검증키, n_{GM}은 두 합성수의 곱인 관리자의 RSA 공개정보이다. 그리고 각 구성원들의 $\{y_j\}$를 바탕으로 다항식 $f(y) = \prod_{i=1}^{n}(y - y_j) = \sum_{i=0}^{n} \alpha_i y^i$를 만들어 그의 상수 α_i를 계산한다. 2장과 같이 $\{\alpha_i'\}$, $\{\beta_i\}$를 치환하고, $A_j = \sum_{i=1, i \neq e}^{n-1} \alpha_e y_j^i$라 하면 다음과 같은 방정식이 만족한다.

$$F(y_j) = g^{-A_j} \prod_{i=1}^{n} \beta_i^{y_j^i} = g^{-A_j} g^{\sum_0^n \alpha_i' y_j^i} = 1 \bmod p$$

다시, 관리자는 임의의 난수 $\gamma \in Z_q$와 그의 역수 γ^{-1}를 계산하여 $\rho_j (= -\gamma A_j \bmod q)$를 생성한다. 각 구성원에게 (v_j, ρ_j)를 주고 집단의 공개키 $\{\beta_0, \ldots, \beta_{n+1}\} = \{\beta_0, \ldots, \beta_n, g^{\gamma^{-1}}\}$와 공개정보 a, g, e_{GM}을 공개한다.

5.2 확장된 프로토콜

초기 설정이 잘 되었다고 가정하면, 다음과 같은

방식으로 Alice는 자신의 x_a, v_a, 그리고 y_a를 바탕으로. 자기가 속한 집단 G_a를 대표하고 자신의 신분을 숨긴 채 자기가 보내기 원하는 Bob이 속한 집단 G_b에게 (c_1, c_2)를 생성하여 보낸다.

$z, t \in_R Z_q$

$k = g^z \bmod p = k_1 \| k_2$

$r = H_{k_2}(m)$

$s = z(r + x_a \cdot t)^{-1} \bmod q$ if SDSS1

$\quad = z(1 + r \cdot x_a \cdot t)^{-1} \bmod q$ if SDSS2

$w = H(m)$

$c_1 \leftarrow \{a_0, \cdots, a_{n+2}\}$

$\quad = \{k\beta_0^{wy_a}, \beta_1^{wy_a}, \cdots, \beta_{n+1}^{wy_a}, g^{(t^{c_{n+1}})y_a}\}$

$c_2 = E_{k_1}(ID_{G_a} \| m \| r \| s \| \delta_a (= g^{x_a t}) \| tv_a)$

Bob이 속한 G_b의 구성원은 Alice에게 받은 (c_1, c_2)와 자신의 비밀정보 (x_b, ρ_b)를 가지고 다음과 같은 과정으로 먼저 k를 복원 후, G_a에서 보낸 메시지임을 알 수 있다. 복호 후, ID_{G_a}를 참조하여 G_a 집단의 관리자의 공개 검증키 e_{GM}을 가지고 r, s, a_{n+2}값들을 검증한다.

$k \leftarrow a_0 (\prod_{i=1}^{n} a_i^{y_b}) a_{n+1}^{\rho_b} = g^z \bmod p = k_1 \| k_2$

$D_{k_1}(c_2) = ID_{G_a} \| m \| r \| s \| \delta_a \| tv_a$

$r' = H_{k_2}(m)$, $g^{z'} = (\delta_a \cdot g^{r'})^s$ if SDSS1

$\quad g^{z'} = (g \cdot \delta_a^{r'})^s$ if SDSS2

$a_{n+2}' = g^{(tv_a)^{c_{n+1}}}$

위와 같은 방법으로 송신자의 x_a, y_a, v_a의 값이 올바른 값인지 아닌지 검증할 수 있다.

이 집단 signcryption은 집단서명과 달리 집단서명과 대칭키 암호를 동시에 하는 방식임으로 집단의 크기가 늘어남에 따라 그 효율성이 떨어지는 것은 피할 수 없는 성질이다. 제안한 프로토콜은 Yi Mu의 확장된 집단 signcryption[4]보다 간단하면서도 집단 서명의 요구조건을 만족시키는 효율적인 집단 signcryption이다.

5.3 안전성

확장된 프로토콜은 집단 signcryption으로서 signcryption과 집단서명의 기능을 함께 가지고 있으므로, 아래와 같은 기능들을 동시에 만족해야한다.[1,11~14]

- 정확성; 특정집단의 구성원이 생성한 signcryption 메시지가 수신집단의 구성원에 의해서 정확히 unsigncryption이 되어야한다는 성질로서, 5.2의 프로토콜의 동작과정을 살펴봄으로써, 정확성을 지님을 알수있다.

- 위조방지; 4.1에 제안된 프로토콜에서의 경우와 마찬가지로, $H_k(\cdot)$가 난수발생기의 역할을 하고 각 집단 구성원들의 비밀키 x_i가 자신을 제외한 누구에게도 노출되지 않는다. 따라서 제3자의 위조로 인한 공격은 불가능하다.

- 익명성; 수신자가 송신한 signcryption을 자신의 비밀정보로 복호한 후에도 송신집단의 어떤 구성원으로부터 받은 지를 확인하는 것은 계산상 불가능하다. 송신자의 정보가 $\delta_a (= g^{x_a t})$와 tv_a형태로 숨겨져 있으므로, 이산대수문제의 어려움과 세션마다 변하는 난수를 알기 어렵다는 것에 기인하여 수신자는 누가 송신한지를 알아내기는 계산상 불가능하다.

- 기밀성; 4.2.3의 기밀성과 동일한 기밀성을 가진다.

- Unlikability; 유효한 두개의 복호된 메시지($ID_{G_a} \| m \| r \| s \| \delta_a \| tv_a$)와 ($ID_{G_a} \| m' \| r' \| s' \| \delta_a' \| tv_a'$)로부터 송신자가 동일한지 아닌지 알아낼 수가 없다. 이 기능은 익명성의 경우와 유사하게, 어느 누구도 송신자의 정보를 지닌 δ_a와 tv_a에서 매번 변하는 난수 t를 알지 못하므로, 두 메시지간의 연계성을 알 수 없다.

- Exculpability; 어떤 집단구성원 및 관리자도 다른 집단의 구성원을 대신해서 위조를 하지 못해야한다는 성질이다. 이 프로토콜의 경우, 이산대수문제에 근거하여 어떤 구성원 및 관리자도 δ_a로부터 각 구성원의 비밀키 x_a를 알아낼 수 없음으로 위조가 불가능하다. 심지어 그 집단의 관리자 역시, 자신의 집단구성원을 대신해서 서명 및 암호를 할 수가 없다.

- 부인봉쇄(Traceability); 분쟁이 발생 시 메시지를 받

은 G_b의 구성원은 G_a의 관리자에게 c_1과 복호를 한 메시지 $D_{k_1}(c_2)$를 전송한다. 그 관리자는 자신이 알고 있는 $\{y_j\}$정보를 바탕으로 $a_i? = (\beta_i^{w})^{y_w}$인지를 확인하여 누가 signcryption하였는지를 알아내어 분쟁을 해결한다.

- Coalition-resistance; 집단의 구성원 여러 명이 담합하여도 위조를 할 수 없어야 된다는 성질이다. 이 프로토콜의 경우, 관리자가 구성원으로부터 y_a를 받아 RSA 서명인 v_a를 그 구성원에게 전달한다. 어떤 경우의 담함에도 관리자나 유효한 구성원의 도움없이는 유효한 x_a, y_a, v_a를 생성할 수가 없다.

5.4 연산량

기존의 확장된 signcryption[4]과 제안한 확장된 프로토콜을 모듈라 연산의 관점에서 연산량을 비교하면 [표 2]와 같다. [표 1]과 같이 대칭키 암·복호 및 해쉬는 모듈라 연산에 비해 작음으로 무시하고 모듈라 곱셈,역원 및 멱승 연산을 중점으로 연산량을 분석하였다. 약간의 역원계산 및 RSA 연산이 첨가되었지만, 전체적인 연산량 측면에서 서명 및 복호 시 기존의 확장된 집단 signcryption보다 효율적임을 볼 수 있다.

〔표 2〕 확장 프로토콜의 연산량 분석

연산량		signcryption		unsigncryption	
		SDSS1	SDSS2	SDSS1	SDSS2
기존 확장 프로토콜	M_q	$2n$+3	$2n$+3	n	n
	M_p	1	1	$3n$+3	$3n$+3
	E_q	n-1	n-1	n-1	n-1
	E_p	n+7	n+7	$3n$+1	$3n$+1
제안한 확장 프로토콜	I_q	1	1	-	-
	M_q	n+4	n+5	-	-
	M_p	1	1	n+2	n+2
	E_p	-	-	n-1	n-1
	E_p	n+5	n+5	n+4	n+4
	$M_{n_{GM}}$	2	2	-	-
	$E_{n_{GM}}$	1	1	1	1

I_q는 mod q에서 역원, M_q(M_p)는 mod q(p)에서 곱셈, E_p(E_q)는 mod p(mod q)에서 멱승, $M_{n_{GM}}$은 RSA 곱셈, $E_{n_{GM}}$은 RSA 멱승이다.

V. 결 론

이 논문에서는 계산상 증가없이 사용자 익명성을 가지면서 부인방지가 가능한 새로운 distributed signcryption을 제안하고, 이를 바탕으로 기존의 프로토콜이 가지고 있는 집단 관리자의 비밀정보의 집중을 막고, 사용자 익명성, 그리고 분쟁발생 시 부인방지를 할 수 있는 집단 signcryption을 제안하였다. 이에 더하여 이 프로토콜을 집단 signcryption으로 확장시켰다. 제안된 프로토콜은 전자상거래 및 비공개 입찰과 같은 응용에 효율적으로 쓰일 수 있다.

Acknowledgement

유용한 조언을 해주신 익명의 심사자들에게 감사드립니다.

참 고 문 헌

[1] Y. Zheng, "Digital Signcryption or How to Achieve Cost(Signature & Encryption) << Cost(Signature) + Cost(Encryption)", *In Advances in Cryptology - CRYPTO'97*, LNCS 1294, pp.165~179, 1997.

[2] Y. Zheng, "Signcryption and its appli cations in efficient public key solutions", *In Proceedings of 1997 Information Security Workshop(ISW'97)*, LNCS, 1997.

[3] J. Koo, H. Kim, I. Jeong, D. Lee, J. Lim, "Jointly unsigncrypptable signcryption schemes", *In proceed ings of WISA2001*. 2001.

[4] Y. Mu and V. Varadharajan, "Distri buted signcryption", *In Progress in Cryptology-INDOCRYPT'2000*, LNCS 1977, pp.155~164, 2000.

[5] Y. Mu, V. Varadharajan, and K. Q. Nguyen, "Delegated decryption", *In Proceedings of Cryptography and Coding*, LNCS, 1999.

[6] National Institute of Standards and Technology, "Digital Signature Standard", Federal Information Processing Standars Publication FIPS PUB 186 U.S. Department of Commerce, May 1994.

[7] F. Bao and R. H. Deng, "A signcryption scheme with signature directly verifiable by public key", *In Proceeding of PKC'98*, LNCS 1431, pp.55~59, 1998.

[8] K. Lee, S. Moon, W. Jeong, and T. Kim, "A 2-pass Authentication and Key Agreement Protocol for Mobile Communications", *In proceedings of ICISC'99*, pp.143~155, 1999.

[9] K. Lee and S. Moon, "AKA Protocols for Mobile Communications", *In proceedings of ACISP 2000*, pp.400~411, 2000.

[10] D. Kwak, J. Ha, H. Lee, H. Kim, and S. Moon, "A WTLS Handshake Protocol with User Anonymity and Forward Secrecy", *In proceeding of CIC '2002*, LNCS Vol.2524, pp.219~230, 2002.

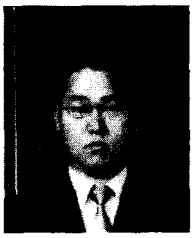
[11] J. Camenisch and M. Stadler, "Efficient group signature schemes for large groups", *In Advances in Cryptology-CRYPTO'97*, LNCS 1294, pp.410~424, 1997.

[12] G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik, "A practical and provably secure coalition- resistant group signature scheme", *In Advances in Cryptology-CRYPTO '2000*, LNCS 1880, pp.255~270, 2000.

[13] E. Bresson and J. Stern, "Efficient revocation in group signcryption", *In Proceeding of PKC'2001*, LNCS 1992, pp. 190-206, 2001.

[14] Y. Lyuu and M. Wu, "Convertible group undeniable signatures", *In proceeding of ICISC' 2002*, LNCS Vol 2587, pp.46~61, 2002.

〈著者紹介〉



곽 동 진 (Dong-Jin Kwak) 학생회원
1998년 2월 : 경북대학교 전자공학과 졸업(학사)
2000년 2월 : 경북대학교 대학원 전자공학과 졸업(석사)
2000년 3월~현재 : 경북대학교 대학원 전자공학과 박사과정
<관심분야> 공개키 암호 프로토콜, 이동네트워크 정보보호

하 재 철 (Jae-Cheol Ha) 종신회원
1989년 2월 : 경북대학교 전자공학과 졸업(학사)
1993년 8월 : 경북대학교 대학원 전자공학과 졸업(석사)
1998년 2월 : 경북대학교 대학원 전자공학과 졸업(박사)
1998년 3월~2000년 2월 : 나사렛대학교 전자계산소장
1998년 9월~2002년 2월 : 나사렛대학교 학술정보관장
1998년 3월~현재 : 나사렛대학교 정보통신학과 조교수
<관심분야> 정보 보호, 네트워크 보안, 스마트 카드 보안

문 상 재 (Sang-Jae Moon) 종신회원
1972년 2월 : 서울대학교 공업교육(전자)과 졸업(학사)
1974년 2월 : 서울대학교 대학원 전자공학과 졸업(석사)
1984년 6월 : 미국 UCLA 전자공학과 졸업(박사)
1984년 7월~1985년 6월 : UCLA Postdoctoral 근무
1984년 7월~1985년 6월 : 미국 OMNET 컨설턴트
1974년 12월~현재 : 경북대학교 공과대학 전자전기컴퓨터학부 교수
2000년 8월~현재 : 경북대학교 이동네트워크 정보보호기술 연구센터 소장
2002년 2월~현재 : 한국정보보호학회 명예회장
<관심분야> 정보보호, 디지털 통신, 이동 네트워크