익명성을 보장하는 (비대칭) 공개키 공모자 추적

최은영[O] 황정연 이동훈
고려대학교 정보보호 대학원
{bluecey[O], hjy}@cist.korea.ac.kr, donghlee@korea.or.kr

An anonymous asymmetric public key traitor tracing scheme

Eun Young Choi[0], Jung Yeon Hwang, Dong hoon Lee
Center for Information Security Technologies(CIST)

요 약

본 논문은 , 내용 은닉 서명[1]과 시간-잠금 퍼즐[2]을 이용해서 익명성을 보장하는 [3]의 공모자 추적 스킴이 사실상 익명성을 갖지 않는다는 것에 대해서 보인다. 그 다음, 신뢰 기관을 이용하여 익명성을 보장하는 두개의 스킴을 제시한다. 첫 번째 스킴은 게시판과 [4]의 영지식에서의 증명을 이용해서 [3]에 제시된 스킴을 변형시켜 익명성을 보장한다. 두 번째 스킴은 [5]의 비대칭 공모자 추적 스킴을 이용해서 익명성을 보장하는 공개키 공모자 추적 스킴을 제공한다.

1. 서 론

최근에, 브로드 캐스트 암호화 스킴들은 네트워크 상에서의 디지털 컨텐츠(멀티미디어, 소프트웨어) 분배에 사용된다. 이러한 환경에서는 단, 권한을 부여받은 사람만이 디지털 컨텐츠에 접근하고, 어떤 사악한 사용자가 암호화된 내용의 브로드 캐스트로부터의 디지털 컨텐츠를 얻을 수 없어야 하는 것이 중요한 요구사항이다. 또한, 다른 중요한 요구사항은 공모자 추적이다. 공모자 추적 스킴은 사용자와 데이터 제공자가 비밀키를 공유하는 대칭성 스킴과 사용자만이 사용자 개인키를 알고 있는 비대칭으로 나누어진다. 그 후 Boneh 와 Franklin 은 하나의 공개 암호화 키를 사용하는 데이터 제공자와 각기 다른 개인키를 사용하여 복호화 할 수 있는 효율적인 공개키 공모자 추적 스킴을 제시했다.

익명성은 멀티미디어 컨텐츠를 구매할 때, 사용자의 정보가 누출되는 것으로 인해서 더 중요하게 되어, 익명성에 대한 다양한 연구가 이루어 졌으나 이 요구사항은 공모자 추적의 상반되는 개념이다. 그래서 익명성을 보장하는 공모자 추적 스킴을 구성하는 것은 쉬운 일이 아니다.

2. 사용하는 기저들

표시법: 메시지(m)을 key ek 로 암호화 하는 것은 $E_{ek}(m)$, $E_{ek}(m')$,$m)=E_{ek}(m') \| E_{ek}(m)$ 라 표시한다. 메시지(m)을 key sk 로 서명하는 것은 $m_{sign(sk)}$라고 표시한다. (이후로) G_q는 Z^*_p의 위수 q 의 곱셈 subgroup 라고 가정한다.

2.1 은닉 함수 값(Oblivious polynomial evaluation)

OPE 프로토콜[6]에서 보내는 사람 A는 P 함수를 비밀값으로 가지고 있으며, 받는 사람 B는 α를 비밀값으로 가지고 있다. 하지만 보내는 사람 A는 B의 비밀값 α를 모르지만, 받는사람 B가 P함수를 모르는 상태에서 $P(\alpha)$의 값을 계산할 수 있도록 프로토콜이 이루어진다. 그래서 $OPE(\alpha)$는 B에 의해서 A에게 전송되는 것을 표시하며, A에 의해서 B에게 전송되는 것을 $OPE(P(\alpha))$라고 표시한다. 특히 OPE는 유한체상에서 $OPE(\alpha+\alpha')$를 계산할 수 있는 융통성(malleable)을 가진다.

2.2 비대칭 공개키 공모자 추적[5]

kiayias 와 Yung 이 제시한 것으로 불법 디코더를 생성하는데 공모한 모든 사용자를 증명할 수 있는 효율적인 비대칭 공개키 암호이다.

1.초기화 단계: 시스템 관리자는 Z_q 에서 임의의 함수 $Q_1(x)=a_0+a_1x+\ldots+a_{2v}x^{2v}$, $b\in_R Z_q$ 선택하고, $y=g^{a_0}$, $h_0=g$, $h_1=g^{-a_1}$, $\ldots$, $h_{2v}=g^{-a_{2v}}$, $h'=g^{-b}$ 계산하고,시스템의 공개키로 $ek=\langle y, h_0, h_1, \ldots h_{2v}, h'\rangle$를 공개한다. 시스템 관리자는 $Q(x,y')=Q_1(x)+by'$을 비밀로 유지한다.

2.참여 단계: 시스템 관리자가 임의로 Z_u를 선택하고, $\alpha_u=\alpha_u^R+\alpha_u^C$를 생성한다. α_u^C는 사용자가 선택하고 위임(commitment)한 값이고, α_u^R는 시스템 관리자가 선택한 값이다. α_u^C의 위임은 $\langle C_u=g^{\alpha_u^C}, C_{usign(sk_u)}\rangle$ 형태이다. 참여 프로토콜은 OPE의 융통성에 의해서 이루어

진다. 사용자개인키: $K=\langle Q(Z_u, a_u), Z_u, \ldots, Z_u^{2v}, a_u\rangle$

3.암호화 단계: 데이터 제공자는 메시지(content) M을 임의의 r과 암호화 키 ek 이용해서 암호화 한다.

$$E(ek, M)=\langle y^r \cdot M,\ h_0^r, \cdots, h_{2v}^r, (h')^r\rangle$$

4.복호화 단계: 사용자는 암호문 $\overline{C}=\langle G, G_0, G_1, \ldots, G_{2v}, G'\rangle$을 개인키 $K=\langle \delta_0, \ldots, \delta_{2v}, \delta'\rangle$ 를 사용해서 복호화 한다. $D(\overline{C}, K)= G/((G')^{\delta'} \Pi_{j=0}^{2v} (G_j)^{\delta_j})$

5.공모자 추적 단계: 시스템 관리자는 대수학(algebraic) 적 코드를 기반으로 하는 공모자 추적 알고리즘을 사용한다.

입력값 : $\overrightarrow{K}=\sum_{l=1}^{t} \mu_l K_{u_l}$

$\{u_1, u_2, \ldots, u_t\} \subseteq \{1,2,\ldots,n\}$, $t(<2v+2)$: 공모자 수

출력값: $\overrightarrow{\nu}=\langle \nu_{u_1}, \ldots, \nu_{u_t}\rangle$, $\nu_{u_l}=\mu_l$ for $l=1, \ldots, t$.

그리고 $\nu_i=0$ for all $i\in\{1,2,\ldots,n\}-\{u_1, \ldots, u_t\}$

6.심사 단계: 시스템 관리자는 심사관에게 $\overrightarrow{\nu}$, $\overrightarrow{K}$, $a_{u_1}^R, \ldots, a_{u_t}^R$ 사용자의 위임(commitments)값 C_{u_1}: $sign_{u_1}(C_{u_1}), \ldots, C_{u_t}$: $sign_{u_t}(C_{u_t})$을 보낸다. 심사관은 다음을 테스트 한다.

$$\Pi_{l=1}^{t}(C_{u_l} g^{a_{u_l}^R})^{\nu_{u_l}} = ?\ g^{\overrightarrow{K}}$$

이 테스트를 통과한 경우, 심사관은 데이터 제공자가 공모자라고 주장한 것을 보증한다.

2.3 영지식의 증명 [4](proving in Zero-Knowledge)

증명자가 자신이 제공한 $y=g^x$에 대한 비밀값 x를 알고 있다는 것을 증명하는 과정이다. 증명자가 임의의 r값을 선택, $t=g^r$을 계산해서 확인자에게 보내고, 확인자는 임의의 C 값을 선택해서 그것을 증명자에게 보내고 , 증명자는 $s=r-cx$값을 확인자에게 보내고 , 확인자는 $g^s y^c = ^? t$ 를 통해 확인한다.

2.4 비상호작용 은닉전송
(non-interactive oblivious transfer:(nOT))

은닉 전송은 보내는 사람은 받는 사람이 두개의 비밀값 중 어떤 값을 받았는지 알 수 없는 특성을 가지고 있다. 이것을 데이터 제공자가 사용자에게 어떤 개인키를 줬는지 알 수 없게 하는데 이용한다. 구체적인 것은 논문 [7]을 참조하다.

3. [3] 스킴에 대한 문제점 , 익명성 보장의 해결책

3.1 [3] 스킴의 비익명성에 대한 설명

[3]의 스킴에서는 익명성 제공의 방법으로 시간-장금 퍼즐[2]을 사용한다. 하지만 이 것은 추적성의 성질과 익명성의 성질 , 즉 , 정반대의 개념 성질을 가져야 하기 때문에 결국 주어진 시간 잠금 퍼즐은 데이터 제공자가 정직한 사용자의 시간-잠금 퍼즐을 계산할 수 있으므로 익명성을 제공하지 못한다.

3.2 익명성 보장의 해결책(스킴)

우리는 익명성을 보장하기 위해서 신뢰 기관(TA), 영지식 증명, 게시판을 이용한다. 이것은 초기 단계와 키 생성과정, 키 생성 과정은 데이터 제공자의 초기화, 키 생성 과정으로 이루어진다. (ID:사용자 신원정보, Π: 거짓 신원정보(pseudo-identity))

표 1. 초기화 단계

사용자(sk_u; vk_u)	신뢰기관(dk_A;(g, ek_A))
$x\in_R Z_q, \Pi=g^x$	
$\omega_b\in_R Z_{q-1}, (b=0 or 1)$	
$a_b=g^{w_b}, a_{1-b}=C\cdot(g^{w_b})^{-1}$	$(\Pi,(a_0,a_1))$
$\langle \omega_b(a_0,a_1)\rangle$	게시판에 공고
$E_{ek_A}(\Pi_{sign(sk_u)}, ID)$, $E_{ek_A}((a_0,a_1)_{sign(sk_U)})$	

이 과정에서 TA는 $(ID, \Pi_{sign(sk_u)}, (a_0,a_1)_{sign(sk_u)})$을 저장하고, 이후에 공모자 추적에 사용한다.

표 2. 초기화 : 데이터 제공자

$f(x)=C_0+C_1x+\ldots+C_kx^k$: 공개키
S_0:$(u_0, f(u_0))$, S_1:$(u_1, f(u_1))$:복호화 키

표 3. 키 생성 과정

사용자	데이터 제공자
$r\in_R Z_q, t=g^r$ (Π, t) →	검사 $(\Pi,(a_0,a_1))$
← C	$C\in_R\{0,1\}^k$
$s=r-cx \bmod q$	
→ S	검사 $g^s y^c = ^? t$
← nOT (s_0, s_1)	$\beta_i=g^{y_i}, y_i\in_R Z_{q-1}$, $i\in\{0,1$
$r_i\oplus\gamma_i=s_i$	$\gamma_0=(a_0)^{y_0}, \gamma_1=(a_1)^{y_1}$
	$r_0=S_0\oplus\gamma_0$,
	$r_1=S_1\oplus\gamma_1$,

3.3 익명성과 안전성

이 스킴은 데이터 제공자가 사용자의 거짓 신원정보만을 알기 때문에 익명성이 보장되며, 동일 사용자를 보호한다. 또한 기존의 스킴과 동일한 안전성을 갖는다.

4. 익명성을 보장하는 (비대칭) 공개키 공모자 추적

이 스킴은 앞에서 제시된 익명성 보장의 스킴을 이용하여 사용자의 익명성을 제공하며, 다른 점은 신뢰 기관(TA)은 $(ID, \Pi_{sign(sk_u)}, \langle C_u=g^{a_u^C}, C_{usign(sk_u)}\rangle)$을 저장

하고, 데이터 제공자가 암호화 키 (dk_D, ek_D)사용한다.

표 4. 초기화 단계

사용자(sk_u; vk_u)	신뢰기관(dk_A;(g, ek_A))
임의로 α_u^C 선택	
$\langle C_u = g^{\alpha_u^C}, C_{usign(sk_u)} \rangle$	
$x \in_R Z_q, \Pi = g^x$	게시판 공고 Π
$E_{ek_1}(\Pi_{sign(sk_u)}, ID)$, $E_{ek_A}(\langle C_u = g^{\alpha_u^C}, C_{usign(sk_u)} \rangle)$	

표 5. 초기화 : 데이터 제공자

선택: $Q_1(x) = a_0 + a_1 x + \ldots + a_{2v} x^{2v}$, $b \in_R Z_q$

계산 $Q(x, y') = Q_1(x) + by'$: 비밀키

$y = g^{a_0}$, $h_0 = g$, $h_1 = g^{-a_1}$, $\ldots$, $h_{2v} = g^{-a_{2v}}$, $h' = g^{-b}$

공개 $\langle y, h_0, h_1, \ldots h_{2v}, h' \rangle$:공개키

표 6. 키 생성(참여 단계)

사용자		데이터 제공자
$r \in_R Z_q, t = g^r$	(Π, t) →	검사 Π
		선택: Z_u, α_u^R
C, Z_u,OPE(α_u^R)	←	$C \in_R \{0,1\}^k$
$s = r - cx \bmod q$		
E_{ek_0}(s ,OPE($\alpha_u^R + \alpha_u^C$))	→	검사 $g^s y^c = ? t$
key 획득	OPE($Q(Z_u, \alpha_u)$) ←	

앞의 2절에 설명했던 것과 유사한 방법으로 공모자 추적, 심사 과정을 실행하여 공모자를 찾아내고 확인하는 작업을 한다. [3] 에서의 사용자는 데이터 제공자로부터 받은 개인키를 확인하기 위해서 비상호작용 은닉 전송을 사용해야 하지만, 여기서는 공개키의 특성을 이용하기 때문에 사용자는 데이터 제공자와의 어떤 상호 작용도 필요 없다.

참고 문헌

[1].D.Chaum, Blind Signatures for Untraceable Payments, Advances in cryptology -- CRYPTO'82, Plenum Press, 1982, pp.199-203.

[2].R. Rivest, A. Shamir, and D. Wagner, Time-Lock Puzzles and Timed-Released Crypto, LCS Techinical *Mono MIT/LCS/TR-684, 1996.*

[3].E. M, P. K, V. C, An asymmetric traceability scheme for copyright protection without trust assumptions, Advances in cryptology -- EC_Web'2001, LNCS 2115, Springer-Verlag,2001, pp.186-195.

[4].J.Camenisch and M. Michels, Proving in Zero-Knowledge that a Number Is the Product of Two Safe Primes, Advancen in cryptology --EUROCRYPT'99, LNCS 1592,1999, pp 107-122.

[5].A. Kiayias and M.Yung, Breaking and repairing asymmetric public-key traitor tacing, ACM Conference on Computer and Communication Security ,ACM, 2002.

[6].M.Naor and B. Pinkas, Oblivious Transfer and Polynomial Evaluation, the 31th ACM Conference on Computer and Communication Security, ACM, 1999.

[7].M. Bellare, S. Micali, Non-Interactive Oblivious Transfer ,Applications, Advances in cryptology -- CRYPTO'89, LNCS 435,Springer-Verlag, 1990, pp.544-557.