추적 가능한 디지털 서명의 개선

최승걸[O] 박근수
서울대학교 전기,컴퓨터공학부
{sgchoi[O], kpark}@theory.snu.ac.kr

Short Traceable Signatures

Seung Geol Choi[O] Kunsoo Park
School of Computer Science & Engineering, Seoul National University

요 약

그룹 디지털 서명(Group Signatures)은 그룹의 회원인 서명자의 익명성을 보장하는 디지털 서명 방법이다. 최근에 이 서명방법에 비해 향상된 기능을 제공하는 추적 가능한 디지털 서명 방법이 제안되었다[9]. 본 논문은 타원곡선 암호의 Pairing 기법으로 구현될 수 있는 Bilinear map을 이용한 추적 가능한 디지털 서명 방법을 소개한다. 이 서명 방법은 321 바이트를 차지하기 때문에, 기존의 방법[9]이 약 1238 바이트를 차지하는 데 비해 효율적인 서명방법이라 할 수 있다.

1 서 론

1.1 그룹 디지털 서명(Group Signature)

그룹 디지털 서명은 D. Chaum과 E. van Heyst가 처음 제안한 디지털 서명[7]으로, 서명을 통해 서명자가 그룹의 회원인 사실만 확인할 수 있고, 서명자의 정확한 신원은 알 수 없는 디지털 서명 방식이다. 법적인 분쟁이 일어날 경우에는 그룹 관리자(Group Manager)가 해당 서명을 열어봐서(open), 서명자의 신원을 확인할 수 있다. 기본적인 개념이 참신함에도 불구하고 이 논문에서 제안된 기법은 회원의 규모에 따른 확장성이 보장되지 못해서 실제적이지 못하였다.

따라서, 실제적이고 효율적인 그룹 디지털 서명 기법에 대한 연구가 활발이 진행되었으며, 2000년 Ateniese, Camenisch, Joye, Tsudik이 최초로 확장성있고(scalable), 안전성이 증명된 방법을 소개하였다[2]. 최근들어 Boneh, Boyen, Schacham이 타원 곡선(Elliptic Curve)의 Pairing을 이용한 그룹 디지털 서명을 고안함으로써, 서명의 크기를 혁신적으로 줄이는데 성공하였다[6].

1.2 추적 가능한 디지털 서명(Traceable Signatures)

추적 가능한 디지털 서명[9]은 고전적인 그룹 디지털 서명(Group Signatures)기법을 확장하여, 익명성 관리에 대한 더욱 효율적인 공정성 메커니즘(fairness mechanism)을 제공함으로써, 그룹 디지털 서명이 비효율적으로 기능하는 상황을 효율적으로 대처할 수 있다. 예를 들어, 이 서명 방식에 추가된 추적 연산(tracing operation)을 이용하면 악의적인 회원의 신원이 알려졌을때, 이 회원이 만든 서명을 매우 효율적으로 추적할 수 있다. 고전적인 그룹 디지털 서명의 경우 이 작업을 위해서는 모든 서명을 열어보아야 하므로, 매우 비효율적이다.

본 논문에서는 [6]에서 제안된 고전적인 그룹 디지털 서명을 확장하여 만든 추적 가능한 디지털 서명 알고리즘을 소개한다. 이 방식을 사용하면, 321 바이트만으로 서명을 만들 수 있기 때문에, 기존의 방식[9]이 약 1238 바이트를 차지하는데 비해 효율적이다.

2 기본지식

2.1 Bilinear Groups

먼저, bilinear map에 관련된 몇가지 개념을 살펴보자. 표기법은 [6]의 방식을 따랐다.

- G_1, G_2 는 prime order p를 갖는 두개의 (multiplicative) cyclic group 이다.
- g_1, g_2는 각각 G_1, G_2의 generator이다.
- ψ 는 $\psi(g_2) = g_1$을 만족하는 계산 가능한 isomorphism이다.
- $e : G_1 \times G_2 \rightarrow G_T$ 는 다음과 같은 성질을 갖는 계산 가능한 맵이다.
  - Bilinearity : 모든 $u \in G_1$, $v \in G_2$, $a, b \in Z$ 에 대해서, $e(u^a, v^b) = e(u, v)^{ab}$ 이 성립한다.
  - Non-degeneracy: $e(g1, g2) \neq 1$

위의 성질을 만족하는 맵 e는 타원곡선의 Pairing을 이용하면 얻을 수 있다[11, 12]. 여기에서는 G_1의 크기가 약 2^{170}으로 G_1의 원소를 171 비트로 나타낼수 있다는 사실만 이용한다.

2.2 복잡도에 대한 가정(Complexity Assumptions)

이 논문에서는 [6]에서 사용하고 있는 q-Strong Diffie-Hellman Problem (q-SDH)과, Dicision Linear Problem in G_1 이 어렵다는 가정을 그대로 사용한다. 자세한 내용은 [6]을 참고하기 바란다.

3 Short Traceable Signatures

이 절에서는 우리가 제안한 방식의 추적 가능한 디지털 서명(Traceable Signature) 방법을 소개한다.

3.1 KeyGen(n)

키를 생성하는 연산이다. [6]에서처럼, 신뢰할 수 있는 키 발급자(Trusted Key Issuing Entity)가 존재한다고 가정한다.

3.1.1 시스템 공개 키

이 키는 모두가 공유하고 있는 공개 키이다.

- $g_1, u_1, v_1, h_1, u_2, v_2, h_2 \in G_1$: 이 값들은 다음과 같은 등식을 만족한다. $u_1^{\xi_{11}} = v_1^{\xi_{12}} = h_1, u_2^{\xi_{21}} = v_2^{\xi_{22}} = h_2$
- $g_2, w_1, w_2 \in G_2$
- isomorphism ψ : $\psi(g_2) = g_1$

3.1.2 키 발급자 비밀키

이 키는 키 발급자만이 알고 있는 비밀키로서, $w_1 = g_2^{\gamma_1}, w_2 = g_2^{\gamma_2}$ 를 만족하는 γ_1, γ_2 이다.

3.1.3 GM 비밀키

이 키는 그룹 관리자(Group Manager)만이 알고 있는 비밀키로서, 법률적인 문제 발생시 디지털 서명을 열어볼때(Open) 사용한다. gmsk=(ξ_{11}, ξ_{12})가 GM 비밀키에 해당한다.

3.1.4 Tracing Agent 비밀키

그룹 관리자는 문제가 있는 디지털 서명에 대해, 열어보기(Open) 연산을 통하여 서명을 만든 회원을 확인할 수 있다. 만약, 그 회원이 만든 서명을 모두 확인하고 싶을때는 추적(Trace) 연산을 통해서 효율적으로 확인할 수 있다. 추적 연산을 수행하는 Agent들을 Tracing Agent라고 하며, 이들은 추적을 위해서 gtsk=(ξ_{21}, ξ_{22})를 가지고 있다.

3.1.5 회원 비밀키

그룹의 각 회원은 서명을 만들기 위한 비밀키를 가지고 있다. 이 비밀키는 gsk[i] = (A, B, x, x')으로 다음과 같은 조건을 만족한다.

$$A, B \in G_1, x, x' \in Z_p, A^{x+\gamma_1} = g_1, B^{x+x'+\gamma_2} = g_2$$

각 회원은 아래 수식을 통해, 받은 비밀키가 적법한지 확인할 수 있다.

$$e(A, w_1 g_2^x) = e(g_1, g_2), e(B, w_2 g^{x+x'})_2 = e(g_1, g_2)$$

3.2 Sign(gsk[i], M) / Verify(M, σ)

메시지 M과 회원 i의 비밀키 gsk[i]를 이용하여, 추적 가능한 디지털 서명을 만들어내는 연산이다. 먼저 SDH에 대한 영지식 프로토콜(Zero-Knowledge Protocol)[6]을 확장한 버전을 소개한다. Fiat-Shamir 변환[1,8]을 이용하면, 이 프로토콜을 디지털 서명으로 변환할 수 있다.

3.2.1 SDH에 대한 영지식 프로토콜의 확장

Alice(prover)는 $\alpha, \beta \in Z_p$ 를 랜덤하게 선택하여, 아래 값을 계산한다.

$$T_1 = u_1^{\alpha}, \quad T_2 = v_1^{\beta}, \quad T_3 = A h_1^{\alpha+\beta}$$
$$T_4 = u_2^{\alpha}, \quad T_5 = v_2^{\beta}, \quad T_6 = B h_2^{\alpha+\beta}$$

그런 다음 아래 등식들을 만족하는 $(\alpha, \beta, x, x', \delta_1 = x\alpha, \delta_2 = x\beta, \delta'_1 = x'\alpha, \delta'_2 = x'\beta)$ 에 대한 Proof of Knowledge를 수행한다.

$$T_1 = u_1^{\alpha}, \quad T_2 = v_1^{\beta}, \quad T_4 = u_2^{\alpha}, \quad T_5 = v_2^{\beta}$$
$$T_1^{x} u_1^{-\delta_1} = 1, \quad T_2^{x} v_1^{-\delta_2} = 1, \quad T_4^{x'} u_2^{-\delta'_1} = 1, \quad T_5^{x'} v_2^{-\delta'_2} = 1$$
$$e(T_3, g_2)^{x} e(h_1, w_1)^{-\alpha-\beta} e(h_1, g_2)^{-\delta_1-\delta_2} = e(g_1, g_2) / e(T_3, w_1)$$
$$e(T_6, g_2)^{x+x'} e(h_2, w_2)^{-\alpha-\beta} e(h_2, g_2)^{-\delta_1-\delta_2-\delta'_1-\delta'_2} = e(g_1, g_2) / e(T_6, w_2)$$

자세한 과정은 아래와 같다.

ㄱ. Alice는 아래 값을 구해서 Bob에게 보낸다.

$$R_1 = u_1^{r_\alpha}, \quad R_2 = v_1^{r_\beta}$$
$$R_3 = e(T_3, g_2)^{r_x} e(h_1, w_1)^{-r_\alpha - r_\beta} e(h_1, g_2)^{-r_{\delta_1} - r_{\delta_2}}$$
$$R_4 = T_1^{r_x} u_1^{-r_{\delta_1}}, \quad R_5 = T_2^{r_x} v_1^{-r_{\delta_2}}, \quad R_6 = u_2^{r_\alpha}, \quad R_7 = v_2^{r_\beta}$$
$$R_8 = e(T_6, g_2)^{r_x + r_{x'}} e(h_2, w_2)^{-r_\alpha - r_\beta} e(h_2, g_2)^{-r_{\delta_1} - r_{\delta_2} - r_{\delta'_1} - r_{\delta'_2}}$$
$$R_9 = T_4^{r_{x'}} u_2^{-r_{\delta'_1}}, \quad R_{10} = T_5^{r_{x'}} v_2^{-r_{\delta_2}}$$

ㄴ. Bob은 challenge 값 c를 랜덤하게 선택해서 Alice에게 보낸다.

ㄷ. Alice는 아래 값을 계산하여 Bob에게 보낸다.

$$s_\alpha = r_\alpha + c\alpha, \quad s_\beta = r_\beta + c\beta, \quad s_x = r_x + cx$$
$$s_{x'} = r_{x'} + cx', \quad s_{\delta_1} = r_{\delta_1} + c\delta_1, \quad s_{\delta_2} = r_{\delta_2} + c\delta_2$$
$$s_{\delta'_1} = r_{\delta'_1} + c\delta'_1, \quad s_{\delta'_2} = r_{\delta_2} + c\delta'_2$$

ㄹ. Bob은 아래 등식이 모두 성립하면 accept한다.

$$u_1^{s_\alpha} = T_1^{c} R_1, \quad v_1^{s_\beta} = T_2^{c} R_2$$
$$u_2^{s_\alpha} = T_4^{c} R_6, \quad v_2^{s_\beta} = T_5^{c} R_7$$
$$e(T_3, g_2)^{s_x} e(h_1, w_1)^{-s_\alpha - s_\beta} e(h_1, g_2)^{-s_{\delta_1} - s_{\delta_2}} = (e(g_1, g_2) / e(T_3, w_1))^{c} R_3$$
$$e(T_6, g_2)^{s_x + s_{x'}} e(h_2, w_2)^{-s_\alpha - s_\beta} e(h_2, g_2)^{-s_{\delta_1} - s_{\delta_2} - s_{\delta'_1} - s_{\delta'_2}} = (e(g_1, g_2) / e(T_3, w_2))^{c} R_8$$
$$T_1^{s_x} u_1^{-s_{\delta_1}} = R_4, \quad T_2^{s_x} v_1^{-s_{\delta_2}} = R_5$$
$$T_4^{s_{x'}} u_2^{-s_{\delta'_1}} = R_9, \quad T_5^{s_{x'}} v_2^{-s_{\delta_2}} = R_{10}$$

3.2.2 Fiat-Shamir 변환을 이용한 디지털 서명

ㄱ. 위의 프로토콜대로, T_1, ..., T_6, R_1, ..., R_{10}을 구한다.

ㄴ. challenge값 c=H(M, T_1,..., T_6, R_1, ..., R_{10})을 구한다.

ㄷ. 디지털 서명 σ 는$(T_1,\Lambda\ ,T_6,c,s_\alpha,s_\beta,\Lambda\ ,s_{\delta'_1},s_{\delta'_2})$ 이 된다. 각 요소는 171비트를 차지하므로 약 서명의 크기는 약 321바이트가 된다.

ㄹ. Verify는 [6]과 매우 유사하게 수행할 수 있다.

3.3 Open(gmsk, M, σ)

그룹 관리자는 gmsk = (ξ_{11},ξ_{12}) 를 알고 있으므로, 서명에서 $A = T_3 / T_1^{\xi_{11}} T_2^{\xi_{12}}$ 를 얻어낼 수 있다. 회원들의 비밀키 중에서 A와 일치하는 사용자를 찾아내서 서명자의 신원을 알아낸다.

3.4 Reveal(i)

회원 i의 비밀키 gsk[i] = (A, B, x, x') 중에서 B를 리턴한다.

3.5 Trace(gtsk, B, σ)

Tracing Agent는 비밀키 gtsk =(ξ_{21},ξ_{22})을 알고 있으므로, 서명에서 $B' = T_6 / T_3^{\xi_{21}} T_4^{\xi_{22}}$ 를 구할 수 있다. 인자로 주어진 B와 구한 B' 이 같으면 true를 리턴하고, 아니면 false를 리턴한다. 이 연산은 회원들의 비밀키를 모두 검색해봐야하는 Open 연산에 비해 매우 효율적이다.

3.6 Claiming(gsk[i], M, σ)

이 연산을 통해, 디지털 서명 σ 를 생성한 회원 i는 서명의 주인이 자신임을 주장할 수 있다. i의 서명이 맞다면, 다음 값들을 정확하게 구할 수 있다.

$$h^{\alpha+\beta} = T_3 / A, \quad rx = s_x - cx$$

아래의 프로토콜을 Fiat-Shamir 변환[1,8]을 이용하여 Claiming 연산을 non-interactive 버전으로 바꿀수 있다.

ㄱ. Alice는 k를 랜덤하게 선택한다.

ㄴ. Alice는 다음 값들을 구해서 Bob에게 준다.

$$X_1 = T_3^{r_x}, \quad X_2 = T_3^{x}$$
$$X_3 = (h^{\alpha+\beta})^k, \quad X_4 = (h^{\alpha+\beta})^{k\alpha}$$
$$X_5 = e(T_3, g_2)^{r_x}, \quad X_6 = R_3^{\ k}$$

ㄷ. 다음 등식들을 만족하는 (k, x, r_x) 에 대한 Proof of Knowledge를 수행한다. 자세한 과정은 3.1과 유사하므로 생략한다.

$$X_1 = T_3^{r_x}, \quad X_2 = T_3^{x}, \quad X_4 = X_3^{\ x}$$
$$X_5 = e(T_3, g_2)^{r_x}, \quad X_6 = R_3^{\ k}$$
$$X_6 = X_5^{\ k} e(h^{-k(s_\alpha+s_\beta)} X_3^{\ c}, w_1) e(h^{-k(s\delta_1+s\delta_2)} X_4^{\ c}, g_2)$$

4 결론 및 향후 과제

본 논문에서는 기존의 추적 가능한 디지털 서명[9]의 크기를 대폭 줄인 서명 방식을 소개하였다. 하지만, [9]에서는 신뢰할 수 있는 키 발급자를 가정하고 있지 않으므로 완벽하게 일치하는 시스템이라고 할 수는 없다. 향후 이 가정을 제거할 수 있는 안전한 Join 연산을 고안하는 것이 과제라고 할 수 있겠다.

5 참고 문헌

[1] M. Abdalla, J. An, M. Bellare, and C. Namprepre. From identification to signatures via the Fiat-Shamir transform: Minimizing assumptions for security and forward-security. In EUROCRYPT 2002, volume 2332 of LNCS, Springer.

[2] G. Ateniese, J. Camenish, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In Crypto 2000, volume 1880 of LNCS, Springer.

[3] M. Bellare, D. Micciancio, and B. Warinshci. Foundations of group signatures: Formal definitions, siimplified requirements, and a construction based on general assumptions. In EUROCRYPT 2003, volume 2656 of LNCS, Springer.

[4] M. Bellare, H. Shi, C. Zhang. Foundations of group signatures: The case of dynamic groups. Cryptology ePrint Archive, Report 2004/077, http://eprint.iacr.org/.

[5] D. Boneh, B. Lynn, and H. Shacham. Short signatures from Weil paring. In Asiacrypt 2001, volume 2248 of LNCS, Springer.
Full paper: http://crypto.standard.edu/~dabo/pubs.html

[6] D. Boneh, X. Boyen, and H. Shacham. Short Group Signatures. In Crypto 2004, LNCS, Springer.

[7] D. Chaum and E. van Heyst. Group signatures. In EUROCRYPT 1991, volume 547 of LNCS, Springer.

[8] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Crypto 1986, volume 263 of LNCS, Springer.

[9] A. Kiayias, Y. Tsiounis, and M. Yung. Traceable Signatures. In EUROCRYPT 2004, volume 3027 of LNCS, Springer.

[10] A. Kiayias and M. Yung. Group signatures: Efficient constructions and annymity from trapdoor-holders. Cryptology ePrint Archive, Report 2004/076. http://eprint.iacr.org/.

[11] A. Miyaji, M. Nakabayashi, and S. Takano. New explicit conditions of elliptic curve traces for FR-reduction. IEICE Trans. Fundamentals, E84-A(5):1234-43 May 2001.

[12] K. Rubin and A. Silverberg. Supersingular Abelian varieties in cryptology. In Crypto 2002, LNCS, Springer.