Bilinear Pairing을 이용한 서명 시스템 연구 동향

유지현*, 원동호*
*성균관대학교 소프트웨어학과
jhryu@security.re.kr, dhwon@security.re.kr

A Survey of Signature System using Bilinear Pairing

Jihyeon Ryu*, Dongho Won*
*Dept. of Software, Sungkyunkwan University

요 약

동형암호는 서버에 암호화된 데이터를 통해 연산할 수 있다는 장점으로 대용량의 데이터를 암호화하여 처리하는 시스템에 사용될 수 있어 주목된다. 동형암호의 방법 중 효율성과 실용성을 지니는 장점으로 인해 연구되고 있는 Bilinear Pairing을 사용하는 서명 및 인증 방법들은 DDH와 CDH 문제에 기반을 둔 방법으로, 많은 연구가 진행되어 왔다. 본 논문은 동형암호에서 사용되는 Bilinear Pairing의 핵심인 GDH 그룹과 타원곡선암호, Weil Pairing, SDH 문제를 기반으로 하는 서명 방식과 그룹 서명 방식, 랜덤오라클을 제외한 서명을 소개한다.

1. 서 론

대량의 데이터가 분포하고 수집되는 빅데이터의 시대가 도래함에 따라, 데이터를 처리하는 방법이 무수히 발전하고 있다. 특히, 대용량 데이터의 수집 및 학습이 필요한 머신러닝 모델과 많은 데이터를 처리해야 하는 통계 분석은 현대 사회에서 관심이 주목되는 분야이다. 이런 빅데이터를 처리하는 과정에서 원시데이터를 그대로 사용하는 것은 프라이버시 침해의 문제를 야기할 수 있다.

원시데이터 사용으로 인한 보안 문제를 해결하는 방안으로 동형암호가 많은 논문에서 연구되고 있다. 이는 메시지를 암호화할 수 있을 뿐만 아니라 암호화된 데이터끼리의 연산이 가능하다. 동형암호 중에는 암호화된 데이터끼리의 합을 연산할 수 있는 가법 동형암호와 암호화된 데이터끼리의 거듭제곱을 연산할 수 있는 승법 동형암호 등이 있다. 이런 유형의 동형암호는 복잡한 기능을 구현하기 어렵다. 위 두 가지 유형의 동형암호에 대한 대안은 완전동형암호이다. 준동형암호라고도 불리는 완전동형암호는 암호화된 데이터끼리의 합 연산과 거듭제곱 연산이 가능하다. 기능과 효율성의 균형을 위해 레벨 동형암호라는 다른 유형의 동형암호도 집중적으로 연구된다. 동형암호의 기법 중 Bilinear Pairing에 기반한 동형암호는 효율성과 실용성 모두를 지닌다[1]. 그러한 장점으로 인해 Bilinear Pairing을 사용한 동형암호 공개키 암호 시스템과 Bilinear Pairing을 사용한 동형암호 서명이 제안되었다[1, 2].

본 논문은 Bilinear Pairing을 사용한 동형암호 서명에 기반이 되는 지식과 관련 서명을 이해하기 쉽게 설명한다. 2장에서는 Bilinear Pairing을 사용한 서명의 배경 지식이 되는 GDH 그룹, 타원곡선암호, Weil Pairing과 SDH 문제에 관해 설명한다. 제 3장에서는 Bilinear Pairing을 사용한 세 가지 서명 방법을 소개한다. 그리고 마지막 4장에서는 결론으로 마무리한다.

2. 배경 지식

2.1 GDH 그룹

위수가 소수 p인 곱셈순환군 $G = <g>$에 대해 다음의 문제를 정의한다[3, 4]. 다음에서 나타나는 a, b, c는 Z_p^*의 원소이다.

Definition 1. a, b, c에 대해 (g, g^a, g^b, g^c)가 주어졌을 때 $c = ab$임을 결정하는 것을 DDH (Decision Diffie-Hellman)라고 한다.

Definition 2. a, b에 대해 (g, g^a, g^b)가 주어졌을 때, g^{ab}를 연산하는 것을 CDH (Computational Diffie-Hellman) 라 한다.

Definition 3. 그룹 G는 한 번의 그룹 액션 한 번 일어나고 DDH에 최대 τ시간이 걸린다고 할 때 그룹 G를 τ-decision group for Diffie-Hellman이라고 한다.

Definition 4. 그룹 G에서 CDH 문제를 푸는 연산을 알고리즘 A라 할 때, 이 알고리즘 A가 CDH 문제를 풀 때 걸리는 최대 시간이 t이고, 문제를 풀 확률이 ϵ이상일 때, 우리는 알고리즘 A를 G에서 (t,ϵ)-breaks Computational Diffie-Hellman라 한다.

Definition 5. τ-decision group for Diffie-Hellman인 그룹에 대해 (t,ϵ)-breaks Computational Diffie-Hellman을 만족하는 알고리즘이 없을 때, 소수 차수를 갖는 그룹 G를 (τ,t,ϵ)-GDH (Gap Diffie-Hellman) 그룹이라 한다.

2.2 타원곡선암호

타원곡선암호에서는 다음의 수식을 사용한다[5, 6]. 이때, a,b는 F_p의 원소이며, $4a^3+27b^2 \neq 0 \mod p$를 만족한다.

$$y^2 = x^3 + ax + b \mod p$$

타원곡선암호는 다음의 안전성 조건을 만족한다.

- ECCDHP (Elliptic Curve Computational Diffie-Hellman Problem) : xyP가 주어졌을 때, xP와 yP를 찾는 것은 불가능하다.
- ECDDHP (Elliptic Curve Decisional Diffie-Hellman Problem) : xP와 yP가 주어졌을 때, xyP를 찾는 것은 불가능하다.
- ECDLP (Elliptic Curve Discrete Logarithm Problem) : P와 xP가 주어졌을 때, x를 찾는 것은 불가능하다.

P는 F_p 상의 점이며, xP는 P를 x회 연산한 것이며, yP는 P를 y회 연산한 것이다. 또한, xyP는 P를 xy회 연산한 것이다.

2.3 Weil Pairing

맵 e에 대한 Weil Pairing은 다음의 성질을 만족한다.

- $e : E[q] \times E[q] \rightarrow F_{p^{l\alpha}}^*$
- Identity : $\forall R \in E[q], e(R,R) = 1$
- Bilinear : $\forall R_1, R_2 \in E[q], e(aR_1, bR_2) = e(R_1, R_2)^{ab}$
- Non-degenerate : 어떤 $R \in E[q]$에 대하여, 모든 $R' \in E[q]$에 대해 항상 $e(R,R') = 1$을 만족할 때, $R = O$이다.
- Computable : $\forall R_1, R_2 \in E[q]$에서 $e(R_1, R_2)$는 효율적으로 연산된다.

2.4 SDH 문제

SDH (The Strong Diffie-Hellman) 문제를 정의할 때, G_1, G_2가 소수 p를 위수로 갖는 순환군으로, $G_1 = G_2$가 가능하다. g_1이 G_1의 생성원이고, g_2가 G_2의 생성원이라 가정할 때, 다음의 문제를 고려한다[7].

(G_1, G_2)에서의 q-SDH 문제는 다음의 문제를 정의한다. $x \in Z_p^*$ 조건에서 $(q+2)$개의 튜플 $(g_1, g_2, g_2^{\gamma}, g_2^{\gamma^2}, \dots, g_2^{\gamma^q})$가 입력값이며, $(g_1^{1/(\gamma+x)}, x)$가 출력값일 때, 다음의 알고리즘 A는 (G_1, G_2)에서의 q-SDH를 풀 때 ϵ의 이점을 가지고 있다.

$$\Pr[A(g_1, g_2, \dots, g_2^{\gamma^q}) = (g_1^{1/(\gamma+x)}, x)] \geq \epsilon$$

Definition 6. (G_1, G_2)에서 t시간 안에 ϵ의 이점을 가지고 있을 q-SDH 문제를 풀 수 있는 알고리즘이 없을 때 (G_1, G_2)에서의 (q,t,ϵ)-SDH 가정이라 한다.

공개 정보 $g_1, u, v, h \in G_1$와 $g_2, w \in G_2$에 대해 어떤 비밀 정보 $\gamma \in Z_p$에 대한 값 $w = g_2^{\gamma}$이다. 또한, $A \in G_1$과 $x \in Z_p$인 쌍 (A,x)에 대해 $A^{x+\gamma} = g_1$를 만족한다. 즉 이 쌍은 $e(A, wg_2^x) = e(g_1, g_2)$를 만족하며, Schnorr의 프로토콜을 사용한다[8].

Protocol 1. 증명자는 지수 값 $\alpha, \beta \leftarrow Z_p$을 선택하고 A의 선형 암호를 다음과 같이 연산한다.

$$T_1 \leftarrow u^{\alpha},\ T_2 \leftarrow v^{\beta},\ T_3 \leftarrow Ah^{\alpha+\beta}$$

또한, 증명자는 도움 값 $\delta_1 \leftarrow x\alpha, \delta_2 \leftarrow x\beta$을 연산한다.

증명자와 검증자는 각 값 $(\alpha, \beta, x, \delta_1, \delta_2)$가 다음 다섯 관계를 만족하는지 확인해야 한다.

$$u^{\alpha} = T_1, v^{\beta} = T_2$$

$$e(T_3, g_2)^x \cdot e(h,w)^{-\alpha-\beta} \cdot e(h, g_2)^{-\delta_1-\delta_2} = e(g_1, g_2) / e(T_3, w)$$

$$T_1^x u^{-\delta_1} = 1, T_2^x v^{-\delta_2} = 1$$

본 증명은 다음의 과정으로 진행된다. 증명자는 Z_p상의 알려지지 않은 값 $r_\alpha, r_\beta, r_x, r_{\delta_1}, r_{\delta_2}$을 선택하고 다음의 값 R_1, R_2, R_3, R_4, R_5를 연산한다.

$$R_1 \leftarrow u^{r_\alpha}, R_2 \leftarrow v^{r_\beta}$$
$$R_3 \leftarrow e(T_3, g_2)^{r_x} \cdot e(h, w)^{-r_\alpha - r_\beta} \cdot e(h, g_2)^{-r_{\delta_1} - r_{\delta_2}}$$
$$R_4 \leftarrow T_1^{r_x} \cdot u^{-r_{\delta_1}}, R_5 \leftarrow T_2^{r_x} \cdot v^{-r_{\delta_2}}$$

증명자는 검증자에게 $(T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5)$를 보낸다. 검증자는 값 $c \in Z_p$를 증명자에게 보내고, 증명자는 값 $s_a = r_a + ca (a \in \{\alpha, \beta, x, \delta_1, \delta_2\})$를 연산한다. 최종적으로 검증자는 다음의 다섯가지 수식을 검증한다.

$$u^{s_\alpha} = T_1^c R_1$$
$$u^{s_\beta} = T_2^c R_2$$
$$e(T_3, g_2)^{s_x} \cdot e(h, w)^{-s_\alpha - s_\beta} \cdot e(h, g_2)^{-s_{\delta_1} - s_{\delta_2}} = (e(g_1, g_2)/e(T_3, w))^c \cdot R_3$$
$$T_1^{s_x} u^{-s_{\delta_1}} = R_4$$
$$T_2^{s_x} u^{-s_{\delta_2}} = R_5$$

3. Bilinear Map을 이용한 서명 방식

3.1 Boneh 등의 Weil Pairing을 이용한 짧은 서명

Weil pairing을 사용한 서명 중 가장 유명한 방법으로 2001년 Asiacrypt에 제안되었다[4].

Key generation 값 l이 주어지면 E/F_{3^l}을 해당 곡선으로 하고 q를 곡선 차수의 가장 큰 소인수로 둔다. 점 $P \in E/F_{3^l}$은 차수가 q인 점이다. 이때, 임의의 $x \in Z_q^*$를 선택하고 $R \leftarrow xP$를 설정한다. 여기서 (l, q, P, R)은 공개키이고 x는 개인키이다.

Signing 메시지에 서명하려면 0과 1로 이루어진 메시지 $M \in \{0,1\}^*$을 알고리즘 MapToGroup을 사용하여 $P_M \in \langle P \rangle$로 대응한다. $S_M \leftarrow xP_M$을 설정하고, S_M의 x좌표를 서명 $\sigma \in F_{3^l}$을 둔다.

Verification 공개키 (l, q, P, R)와 메시지 M, 서명 σ로 다음의 검증을 수행한다.

- x좌표가 σ에 대응하는 y 값 $y \in F_{3^l}$인 점 $S \in E/F_{3^l}$를 찾는다. 이 점이 존재하지 않으면 서명이 유효하지 않다고 간주한다.
- $u \leftarrow e(P, \phi(S)), v \leftarrow e(R, \phi(h(M)))$을 설정한다. 여기서 맵 e는 곡선 $E/F_{3^{6l}}$의 Weil pairing이고, $\phi : E \rightarrow E$는 곡선 E의 자기동형사상이다.
- $u = v$ 혹은 $u^{-1} = v$일 때 서명을 수락한다. 그렇지 않으면 서명을 거부한다.

MapToGroup Algorithm GDH 서명에 필요한 MapToGroup 알고리즘은 $h : \{0,1\}^* \rightarrow G^*$인 다음의 해시 함수를 의미한다.

- $M \in \{0,1\}^*$이 주어졌을 때, $i \leftarrow 0$으로 설정한다.
- $(x, b) \leftarrow h'(i \| M) \in F_{p^l} \times \{0,1\}$을 설정한다.
- $f(x)$가 F_{p^l}에서 quadratic residue라면 다음을 수행한다.
  - $y_0, y_1 \in F_{p^l}$이 $f(x)$의 두 square root라고 할 때, $b \in \{0,1\}$을 두 root중 하나 선택한다고 한다. y_0, y_1은 F_p에서 degree가 $l-1$인 다항식으로, y_0의 상수항은 y_1보다 크지 않다고 보증한다. (그렇지 않을 경우, y_0과 y_1의 값을 바꾼다.) 이때 점 $\widetilde{P_M} = (x, y_b) \in E/F_{p^l}$을 설정한다.
  - $P_M = (m/q)\widetilde{P_M}$을 연산하고 P_M이 G 내에 있는지 확인한다. 만약 P_M이 G^* 내에 있으면 $MapToGroup_{h'}(M) = P_M$으로 설정하고 동작을 중지한다.
- 그렇지 않을 경우, i를 증가시키고 (x, b)설정으로 올라간다. 만약 i가 2^l에 도달할 경우 실패한다.

3.2 Boneh 등의 SDH를 사용한 짧은 그룹 서명

Bilinear pairing을 사용한 서명 중 가장 유명한 그룹 서명 방법으로 2004년 CRYPTO에 제안되었다[7].

Key generation 그룹의 멤버 수 n을 입력으로 받으면 다음의 과정이 진행된다. $u^{\xi_1} = v^{\xi_2} = h$를 만족하는 $h \leftarrow G_1 \setminus \{1_{G_1}\}$, $\xi_1, \xi_2 \leftarrow Z_p^*$, $u, v \in G_1$를 선택한다. $\gamma \leftarrow Z_p^*$를 선택하고 $w = g_2^\gamma$를 설정한다. γ를 사용하여 각 사용자 $i, 1 \le i \le n$에 대하여 SDH 튜플 쌍 (A_i, x_i), $x_i \leftarrow Z_p^*$을 선택하고 $A_i \leftarrow g_1^{1/(\gamma + x_i)}$를 설정한다. 그룹 공개키 $gpk = (g_1, g_2, h, u, v, w)$이며, 그룹 메니저의 개인키는 $gmsk = (\xi_1, \xi_2)$이며, 각 사용자의 개인키는 $gsk[i] = (A_i, x_i)$이다.

Signing 그룹 공개키 $gpk = (g_1, g_2, h, u, v, w)$, 사용자의 개인키 $gsk[i] = (A_i, x_i)$와 메시지 $M \in \{0,1\}^*$가

주어질 때, 서명값 $\sigma = (T_1, T_2, T_3, c, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2})$은 2.4절에 의해 연산된다.

Verification 그룹 공개키 $gpk = (g_1, g_2, h, u, v, w)$와 메시지 $M \in \{0,1\}^*$, 그룹 서명 $\sigma = (T_1, T_2, T_3, c, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2})$가 주어질 때 검증은 2.4절에 의해 연산된다.

3.3 랜덤 오라클을 제외한 서명

랜덤 오라클을 제외한 서명은 2004년 EUROCRYPT에서 발표되었다[9]. G_1, G_2가 소수 p를 위수로 갖는 순환군으로, $G_1 = G_2$가 가능하다. g_1이 G_1의 생성원이고, g_2가 G_2의 생성원이라 가정하고, 메시지 $m \in Z_p^*$로 둘 때, 다음의 과정을 거친다.

Key generation 랜덤 값 $x, y \leftarrow Z_p^*$을 뽑고 $u \leftarrow g_2^x \in G_2$와 $v \leftarrow g_2^y \in G_2$를 연산한다. 이때 공개키는 (g_1, g_2, u, v)이고 비밀키는 (x, y)이다.

Signing 주어진 비밀키 $x, y \in Z_p^*$와 메시지 $m \in Z_p^*$, 랜덤 값 $r \in Z_p^*$을 선택한다. 이후, $\sigma \leftarrow g_1^{1/(x+m+yr)} \in G_1$을 연산한다. 여기서 $1/(x+m+yr)$은 $\bmod p$상에서 연산한 값이다. 따라서 $x+m+yr=0$이 된다면 랜덤 값 $r \in Z_p^*$을 다시 선택해야 한다. 서명 값은 (σ, r)이다.

Verification 공개키 (g_1, g_2, u, v)와 메시지 $m \in Z_p^*$ 그리고 서명 (σ, r)이 주어질 때 다음을 검증한다.

$$e(\sigma, u, g_2^m, v^r) = e(g_1, g_2)$$

4. 결 론

현대 사회에서의 대량의 데이터를 처리할 필요성이 높아지게 되며, 다양한 동형암호 프로토콜에 대한 연구가 활발하게 진행되고 있다. 또한, 동형암호의 내부 알고리즘 뿐만 아니라 다양한 응용분야인 통계 및 머신러닝 분야에서 사용하려는 시도가 증가하고 있다. 그중 Bilinear Pairing을 사용한 동형 암호는 효율성과 실용성을 지녔다고 알려져 있다. 본 논문은 Bilinear Pairing을 이용한 동형암호의 이해를 돕기 위해 기반 지식에 대해 정리하고, Bilinear Pairing을 사용한 서명에 대해 소개한다. 본 논문에서는 Bilinear Pairing을 이용한 서명과 이를 사용하기 위한 기반 지식인 GDH 그룹, 타원곡선암호, Weil Pairing과 SDH 문제를 다루었다. 또한, 본 논문에서는 Bilinear Pairing을 이용한 일반 서명과 그룹 서명, 랜덤 오라클을 제외한 서명이 소개된다.

ACKNOWLEDGEMENT

이 논문은 2021년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임. (No.2021-0-00558, 동형암호 기술을 활용한 국가통계 분석 시스템 개발)

참고문헌

[1] Nuttapong Attrapadung, Goichiro Hanaoka, Shigeo Mitsunari, Yusuke Sakai, Kana Shimizu, Tadanori Teruya, “Efficient Two-level Homomorphic Encryption in Prime-order Bilinear Groups and A Fast Implementation in WebAssembly”, Proc. of ASIACCS ’18, pp. 685 - 697.

[2] Dan Boneh, Eu-Jin Goh, Kobbi Nissim, “Evaluating 2-DNF Formulas on Ciphertexts.”, Proc. of TCC ’05, pp. 325 - 341.

[3] Tatsuaki Okamoto, David Pointcheval, “The Gap-Problems: A New Class of Problems for the Security of Cryptographic Schemes”, Proc. of PKC ’01, Lecture Nores in Computer Sciences, Vol. 1992, pp. 104 - 118.

[4] Dan Boneh, Ben Lynn, Hovav Shacham, “Short Signatures from the Weil Pairing”, Proc. of Asiacrypt ’01, Lecture Notes in Computer Sciences, Vol. 2248, pp. 514 - 532.

[5] Victor S. Miller, “Uses of Elliptic Curves in Cryptography”, Proc. of Crypto ’85, Advances in Cryptology, Vol. 218, pp. 417 - 426.

[6] Neal Koblitz, “Elliptic Curve Cryptosystems.” Mathematics of Computation, Vol. 47, pp. 203 - 209, 1987.

[7] Dan Boneh, Xavier Boyen, Hovav Shacham, “Short Group Signatures.” Proc. of Crypto ’04, Advances in Cryptology, pp. 41 - 55, 2004.

[8] C. P. Schnorr, “Efficient Signature Generation by Smart Cards.” Journal of Cryptology, Vol. 4, pp. 161 - 174, 1991.

[9] Dan Boneh, Xavier Boyen, “Short Signatures without Random Oracles.” Proc. of EUROCRYPT ‘04, Advances in Cryptology - EUROCRYPT 2004, pp. 56-73, 2004.