논문 15-40-05-08 The Journal of Korean Institute of Communications and Information Sciences '15-05 Vol.40 No.05
http://dx.doi.org/10.7840/kics.2015.40.5.833

다항식 상등성 영지식 증명의 일반화

김 명 선[◆°], 강 보 람[*]

Generalization of Zero-Knowledge Proof of Polynomial Equality

Myungsun Kim[◆°], Bolam Kang[*]

요 약

본 논문에서는 미리 알려진 임의의 다항식과 암호화된 다항식의 곱셈을 수행한 후, 해당 곱셈이 정당하게 수행되었음을 보이기 위해 증명자 (Prover)와 검증자 (Verifier)간의 다항식 상등성 영지식증명 (Zero-knowledge Proof) 프로토콜을 일반화할 수 있는 방법을 다룬다. 이를 위하여 다항식의 상등성을 증명하는 일반화된 프로토콜을 제시하고 랜덤오라클 (Random Oracle) 모델에서 안전성을 증명한다. 이러한 기법은 안전한 집합연산 기법을 포함하여 다항식에 기반한 다자간 연산기법 (Secure Multi-party Computation)에 적용될 수 있다.

Key Words : Zero-knowledge proofs, Set operations, Polynomials

ABSTRACT

In this paper, we are interested in a generalization of zero-knowledge interactive protocols between prover and verifier, especially to show that the product of an encrypted polynomial and a random polynomial, but published by a secure commitment scheme was correctly computed by the prover. To this end, we provide a generalized protocol for proving that the resulting polynomial is correctly computed by an encrypted polynomial and another committed polynomial. Further we show that the protocol is also secure in the random oracle model. We expect that our generalized protocol can play a role of building blocks in implementing secure multi-party computation including private set operations.

Ⅰ. 서 론

Kissner와 Song이 Crypto 2005에서 다항식을 이용하여 집합연산을 안전하게 수행하는 효율적인 방법을 제시하였다[6]. 그들은 제안하는 집합연산 기법이 능동 공격자 (Maclicous Adversary) 모델에서도 안전함을 증명하기 위하여 몇 가지 영지식증명 (Zero-knowledge Proof) 기법을[3,4] 이용할 것을 함께 제안하였다. 이 중에서 두 다항식의 곱셈의 결과가 옳다는 것을 증명하기 위해 다항식의 상등성 (Polynomial Equality)을 증명하는 기법을 포함한다. 특히 이러한 다항식의 항등성은 Zero-Knowledge Proof (ZKP) 방식으로 수행될 필요가 있다. Prover가 소유한 다항식을 숨긴 채 임의의 다른 다항식과 곱셈의 결과가 정당하다는 것을 증명해야하기 때문이다. 이러한 방식으로 다항식의 상등성을 증명하는 방식을 다항식 상등성 영지식 증명 (PE-ZKP)라고 부른다.

PE-ZKP를 좀 더 구체적으로 설명하면 prover P가 동형암호를 사용하여 암호화한 후 공개된 다항식과 자신의 소유한 임의의 다항식에 대하여 곱셈연산을

※ 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 산학협력 특성화 지원사업의 연구결과로 수행되었음 (IITP-2015-R0346151001000001002)

◆° First and Corresponding Author : The Univ. of Suwon, Dept. of Infomation Security, msunkim@suwon.ac.kr, 정회원
* The Univ. of Suwon, Dept. of Infomation Security, 학사과정, teapack@suwon.ac.kr
논문번호 : 2015-03-066, Received March 23, 2014; Revised April 24, 2015; Accepted April 24, 2015

수행한 후, 암호화된 다항식을 얻게 되는데 이 결과 다항식이 암호화된 다항식과 자신의 다항식의 곱셈을 정직하게 수행하여 얻은 결과임을 verifier V에게 증명하고자 하는 것이다.

1.1 문제 정의

이미 언급한 바와 같이 Kissner와 Song은 PE-ZKP를 저자들이 제안하는 모든 집합연산 기법에서 폭넓게 사용하고 있다. 그러나 저자들은 그들이 사용하는 ZKP 기법에 대하여 자세한 동작을 구체적으로 제시하지 않았다. 물론 수신한 암호문을 송신자가 알고 있음을 증명하는 ZKP of Plaintext Knowledge (ZKPK)와 같이 이미 잘 알려진 ZKP 기법을 자세하게 제시할 필요는 없으나, PE-ZKP와 같이 이전에 구체적으로 제시된 바가 없는 ZKP 기법을 구체적으로 제시하여 동작이 완전하게 이루진다는 것을 확인할 필요가 있다. 이러한 필요성에서 출발하여 본 연구진은 다음과 같은 의문을 추가로 갖게 되었다. 즉,

이러한 PE-ZKP를 일반적인 형태로 표현할 수 없는가?

만약 이것이 가능하다면 PE-ZKP를 수행하는데 필요한 연산의 복잡도를 일반적인 형태로 제시할 수 있고, 나아가 어떠한 부분에서 연산이나 통신의 복잡도를 개선할 수 있는지 파악하는 것이 용이할 것이다.

본 연구는 이러한 두 가지 측면에서 이루어진 것으로 기여점을 구체적으로 제시하면 다음과 같다.

1.2 본 논문의 기여점

본 논문의 기여점은 다음과 같이 두 가지로 요약할 수 있다.

(1) Kissner와 Song에 의하여 제안된 PE-ZPK 프로토콜을 구체적으로 제시한다. PE-ZKP 기법을 구체적으로 자세히 제시하여 향후 Kissner-Song 집합연산 기법을 구현하려는 연구에서 정확한 동작을 구현할 수 있도록 지원할 수 있다.

(2) 나아가 PE-ZKP 기법을 일반화하고 이 기법이 랜덤오라클 모델에서 안전하다는 것을 증명한다. 이러한 결과를 응용하여 향후 PE-ZKP 기법의 통신량과 연산량의 복잡도를 개선하는데 이론적 기법을 제시할 수 있을 것이다.

본 논문은 먼저 Ⅱ장에서 필요한 표기법 및 암호학적 도구를 소개하고, Ⅲ장에서 Kissner와 Song에 의하여 구체적으로 제시되지 않은 PE-ZKP 기법을 완전하게 기술한 후, Ⅳ장에서 이러한 PE-ZKP의 일반화된 기법을 제시하고 이것이 랜덤 오라클 모델에서 안전하다는 것을 증명한다. 마지막으로 Ⅴ장의 맺음말로 본 논문을 마무리하고자 한다.

Ⅱ. 배경지식

본 장에서는 다항식과 다항식의 암호화된 표현 그리고 다항식 암호화에 사용되는 동형암호 기법을 소개한다.

먼저 R은 임의의 ring이라 하자. 그리고 $R[x]$는 모든 계수가 R에 속하는 polynomial ring이라 하자. 편의를 위하여 모든 다항식의 미지수 x는 특별히 언급하지 않는 이상 종종 생략하여 사용한다.

2.1 집합의 다항식 표현과 집합연산

집합 $S=\{\alpha_1,...,\alpha_k\}$가 주어질 때 집합의 각 원소를 다항식의 근으로 표현하는 것을 집합의 다항식 표현이라 부른다. 즉, 집합 S는 차수가 k인 다항식 $f\in R[x]$에 의하여 다음과 같이 표현할 수 있다.

$$\begin{aligned} f(x) &= (x-\alpha_1)(x-\alpha_2)\cdots(x-\alpha_k) \\ &= \Pi_{i=1}^{k}(x-\alpha_i) \\ &= \Sigma_{i=0}^{k}a_i x^i \end{aligned}$$

집합을 다항식으로 표현하는 경우, 특히 유리한 점은 다항식의 덧셈과 곱셈이 집합의 교집합이나 합집합 연산에 대응시킬 수 있다는 것이다.

2.1.1 교집합

집합 $S=\{\alpha_1,...,\alpha_k\}, T=\{\beta_1,...,\beta_\ell\}$에 대하여 각각을 다항식 f,g로 나타내면 $f=\sum_{i=0}^{k}a_i x^i$이며 $\varphi=\sum_{j=0}^{\ell}b_j x^j\in R[x]$일 것이다. 두 다항식에 대하여 $f+\varphi$를 수행하면 임의의 다항식 $\phi\in R[x]$에 대하여 $f+\varphi=\gcd(f,\varphi)\cdot\phi$로 쓸 수 있다. 특히 흥미로운 것은 $\gcd(f,\varphi)$의 모든 근이 $S\cap T$에 해당한다는 것이다.

2.1.2 합집합

비슷하게 두 집합 S, T의 합집합을 다항식 연산을 이용하여 구하려는 경우에는 간단하게 다항식 곱셈을 수행하면 된다. 즉, $\phi=f\cdot\varphi$의 모든 근은 두 집합의 합집합 $S\cup T$에 해당한다는 것을 쉽게 짐작할 수 있다.

물론 암호학적으로 사용하기 위해서는 기술적인 보완이 필요하지만, 개념적으로는 동일하다. 좀 더 자세

한 것은 그들의 논문[6]을 참조하라.

2.2 동형 암호

본 장에서는 다항식을 암호화한 후 계수의 암호화 및 연산에 적용되는 동형암호를 정의한다. 추가로 본 논문에서 사용하는 동형암호인 additive El Gamal 암호기법을 간략히 소개한다.

먼저 공개키 암호기법은 E=(KG,E,D)에 의하여 정의되며, 여기서

- $KG(1^{\lambda})$는 입력으로 security parameter λ를 입력으로 받아 (pk,sk)를 생성한다.
- $c \leftarrow E(pk,m)$은 암호화 알고리즘으로 공개키 pk와 평문 m, 난수 r을 사용하여 암호문 c를 출력한다.
- $m \leftarrow D(sk,c)$는 복호화 알고리즘으로 비밀키 sk와 암호문 c를 받아 평문 m을 출력한다.

본 논문에서는 표기의 편의를 위하여 공개키 pk를 생략하고 암호 알고리즘 $E(m)$을 종종 사용한다. 공개키 암호기법 E가 두 암호문 c_1,c_2에 대하여 평문의 합을 암호문상의 직접 연산으로 계산할 수 있으면 덧셈형 동형암호라고 부른다. 즉,

$$c_1 \oplus c_2 = E(m_1) \oplus E(m_2) = E(m_1 + m_2)$$

를 지원하고 임의의 평문 s에 대하여

$$s * E(m) = E(m \cdot s)$$

를 가능하는 연산 $(\oplus, *)$를 허용하면 덧셈동형암호인 것이다. 이러한 덧셈동형암호는 Paillier 암호기법[7]과 El Gamal 암호기법[2]의 변형기법[1]을 예로 들 수 있는데 본 논문은 기술적 이유로 additive El Gamal 암호 기법을 사용한다.

2.2.1 Additive El Gamal Encryption

- $(pk,sk) \leftarrow KG(1^{\lambda})$: 먼저 λ를 이용하여 충분히 큰 소수 p를 생성하고 Z_p^*의 order q subgroup G_q를 결정하고; 생성자 g를 찾는다. 이제 Z_q에 속하는 임의의 값 x를 선택한 후 $h = g^x$를 계산하여 $pk = h, sk = x$로 설정한다.
- $c \leftarrow E(pk,m)$: 평문 $m \in G_q$에 대하여 난수 $r \in Z_q$를 선택한 후 $c = (g^r, g^m h^r)$을 암호문으로 출력한다.
- $m \leftarrow D(sk,c)$: 암호문을 $c = (A,B)$의 형태로 분할한 후 $m = BA^{-x}$를 계산한다.

Additive El Gamal 암호기법은 복호화를 수행하여도 g^m이고 평문이 지수에 유지되므로 평문 m을 계산하는 것이 비효율적이나 집합연산은 m을 직접 복호화할 필요가 없기 때문에 문제되지 않는다.

2.3 영지식 증명 (Zero-Knowlede Proof)

영지식 증명 또는 ZKP는 임의의 사용자가 자신의 비밀정보를 숨긴 채, 자신의 행위가 정당하게 이루어졌음을 증명하기 위한 도구로서 Goldwasser, Micali와 Racoff에 의하여 제안된 후, 가장 광범위하게 사용되는 암호학적 도구이다[3].

그 후, 많은 암호연구자들에 의하여 ZKP 자체에 대한 연구뿐만 아니라[4], 다양한 응용이 제시되었다. ZKP는 이 자체로 매우 흥미로운 주제이나, 그 내용이 방대하여 지면의 제약을 고려하여 본 논문에서는 Goldreich의 도서[5]를 참고할 것을 제안하며 간략히 마무리 한다.

Ⅲ. Kissner와 Song 기법에서 이용되는 PE-ZKP 기법의 상세

본 장에서는 Kissner와 Song이 자세하게 제시하지 않은 PE-ZKP 기법을 구체적을 기술하려고 한다. 이렇게 하는 이유는 다음 장에서 설명할 PE-ZKP 기법의 일반화의 동작에 대한 이해를 돕고 일반화된 PE-ZKP 기법을 좀 더 간략하게 기술하기 위함이다.

이를 위하여 암호화되어 공개된 다항식을 $E(\varphi)$라 하고 prover가 소유한 다항식을 f라 하자. 또한 암호화된 다항식과 f의 차수를 모두 k라 하고, $f = \sum_{i=0}^{k} a_i x^i, \varphi = \sum_{j=0}^{k} b_j x^j$일 때 $f := (a_0,...,a_k)$로 표시하고 비슷하게 $E(\varphi) := (E(b_0),...,E(b_k))$로 나타내자. 다항식 $p = f\varphi$에 대하여 $p = \sum_{l=0}^{2k} c_l x^l$라 할 때, $c_l = \sum_{l=i+j; 0 \le i,j \le k} a_i b_j$이다. 그래서 암호화된 다항식 $p := (c_0,c_1,...,c_{2k})$로 표현한다.

3.1 Prover의 PE-ZKP

동형암호를 사용하는 것을 제외하면 암호화된 다항식의 곱셈이 특별한 것이 없기 때문에, prover가 알고 있는 다항식 f와 암호화된 다항식 $E(\varphi)$의 곱 역시, $f * E(\varphi) = E(f \cdot \varphi)$로 나타낼 수 있다. 이미 언급한 바와 같이 $*$는 암호문과 상수의 곱을 의미하며 $\cdot$은 평문 간의 곱셈이다. 그래서 $E(p)$역시 $(E(c_0),E(c_1),...,E(c_{2k}))$로 쓸 수 있다.

좀 더 구체적으로 $E(b_j)=(g^{t_j},g^{b_j}h^{t_j})$라 하고 prover가 re-randomize에 $r_{i,j},\gamma_{i,j}$을 사용한다고 하자. 그러면 각 l에 대하여 다음을 계산할 것이다.

즉, $E(c_l)=\sum_{l=i+j}a_j * E(b_j)$이므로 $E(c_l)=(E_l,F_l)$이라면 E_l,F_l은 다음과 같다:

$$E_l=g^{t_la_0+\ldots+t_0a_l+\sum_{i=0}^{l}r_{l,i}}g^{\sum_{i=0}^{l-1}\gamma_{l,i}} \quad (1)$$

$$F_l=g^{b_la_0+\ldots+b_0a_l}h^{t_la_0+\ldots+t_0a_l+\sum_{i=0}^{l}r_{l,i}}h^{\sum_{i=0}^{l-1}\gamma_{l,i}} \quad (2)$$

이제 prover는 verifier에게 자신이 곱셈을 정확하게 수행하였음을 증명하기 위하여 commitment를 만들고, verifier로부터 challenge를 받아 response를 보내면 된다. 그래서 commitment를 Γ로 나타내고 challenge를 ρ로, response는 Σ로 나타내자. 그러면 곱셈 연산을 위하여 계산하는 모든 값을 $l(0\le l\le 2k)$에 관하여 정리하면 다음과 같다:

$\Gamma_l=(G_l,H_l,I_l,J_l)$이고 여기서 각 값은 다음이다.

$$G_l=g^{u_{s_l}}, \quad (3)$$

$$H_l=g^{a_l}h^{u_{s_l}}, \quad (4)$$

$$I_l=g^{t_0u_{s_l}+\cdots+t_lu_{s_0}+\sum_{\ell=0}^{l-1}u_{r_{l,\ell}}+\sum_{\ell=0}^{l-1}u_{\gamma_{l,\ell}}}, \quad (5)$$

$$J_l=g^{b_0u_{a_l}+\cdots+b_lu_{a_0}}h^{t_0u_{s_l}+\cdots+\sum_{\ell=0}^{l}u_{r_{l,\ell}}+\sum_{\ell=0}^{l-1}u_{\gamma_{l,\ell}}} \quad (6)$$

당연한 것이지만 $l\ge k+1$에 대해서는 G_l,H_l을 계산할 필요없다. 그래서 $l=2k$인 경우의 I_l,J_l을 별도로 표시하면 다음과 같다:

- $I_{2k}=(g^{t_k})^{u_{a_k}+u_{r_{2k,0}}}$
- $J_{2k}=(h^{t_k})^{u_{a_k}+u_{r_{2k,0}}}g^{b_ku_{a_k}}$.

Commitment ρ에 대하여 Σ_l은 다음과 같이 계산한다. $\Sigma_l=(\sigma_{s_l},\sigma_{a_l},\sigma_{r_{l,0}},\ldots,\sigma_{r_{l,l}},\sigma_{\gamma_{l,0}},\ldots,\sigma_{\gamma_{l,l-1}})$이고 여기서

$$\sigma_{s_l}=u_{s_l}-\rho\cdot s_l\ \sigma_{a_l}=u_{a_l}-\rho\cdot a_l,$$
$$\sigma_{r_{l,0}}=u_{r_{l,0}}-\rho\cdot r_{l,0},\ldots,\sigma_{r_{l,l}}=u_{r_{l,l}}-\rho\cdot r_{l,l},$$
$$\sigma_{\gamma_{l,0}}=u_{\gamma_{l,0}}-\rho\cdot\gamma_{l,0},\ldots,\sigma_{\gamma_{l,l-1}}=u_{\gamma_{l,l-1}}-\rho\cdot\gamma_{l,l-1}$$

앞 절에서 기술한 바와 동일한 이유로 $l\ge k+1$이면 사용하는 난수의 개수가 점점 줄어서 $l=2k$일 경우 $u_{r_{2k,0}}$만 사용되므로

$$\Sigma_{sk}=\sigma_{r_{sk,0}}=u_{r_{2k,0}}-\rho\cdot r_{2k,0}$$

임을 쉽게 짐작할 수 있다.

주의할 것은 연산의 효율성과 사용자간의 통신 횟수를 줄이기 위하여 비대화식으로 ZKP를 변형하여 사용하는데, 이 경우 prover가 다음과 같이 challenge 값 ρ를 계산한다.

$\rho=H(g,h,\langle A_i,B_i\rangle_{i=0}^{k},\langle C_i,D_i\rangle_{i=0}^{k},\langle E_i,F_i\rangle_{i=0}^{2k})$

로서 H는 랜덤오라클이며 (A_i,B_i)는 a_i의 commitment 값으로 여기서는 단순하게 암호화된 값을 commit한다고 가정하고 $(A_i,B_i):=E(a_i)$이고 비슷하게 $(C_i,D_i)=E(b_i)$이다. E_i,F_i는 수식 (1), (2)에서 계산된 값들이다.

그러면 prover는 자신이 PE-ZKP를 수행하기 전에 미리 commit한 모든 $\langle A_i,B_i\rangle$와 미리 공개된 모든 $\langle C_i,D_i\rangle$에 대하여 두 다항식의 곱셈의 결과가 $\langle E_i,F_i\rangle_{i=0}^{2k}$이고 이것을 PE-ZKP로 증명하는 증거 Φ를

$$\Phi=(\rho,\Sigma_0,\ldots,\Sigma_{2k}) \quad (7)$$

verifier에게 전송한다. $l=2k$인 경우에는 약간의 주의가 필요한데, 이 경우에는 $G_k=H_k=0$으로 설정한다.

3.2 Verifier의 PE-ZKP

Verifier가 prover에 의하여 미리 공개한 $\left(\langle A_i,B_i\rangle_{i=0}^{k},\langle C_i,D_i\rangle_{i=0}^{k},\langle E_i,F_i\rangle_{i=0}^{2k}\right)$과 전송된 수식 (7)의 $\Phi=(\rho,\Sigma_0,\ldots,\Sigma_{2k})$를 이용하여 곱셈이 맞게 이루어진 것인지 검증해야 한다. 이를 위하여 모든 $l(0\le l\le 2k)$에 대하여 다음과 같은 방식으로 $\overline{\Gamma_l}=(\overline{G_l},\overline{H_l},\overline{I_l},\overline{J_l})$을 계산한다. 여기서

$\overline{G_l}=g^{\sigma_{s_l}}$,

$\overline{H_l}=g^{\sigma_{a_l}}B_l^{\rho}h^{\sigma_{s_l}}$,

$\overline{I_l}=\prod_{i=0}^{l}C_i^{\sigma_{a_{l-i}}}E_l^{\rho}g^{\sum_{i=0}^{l}\sigma_{r_{l,i}}+\sum_{i=0}^{l-1}\sigma_{\gamma_{l,i}}}$,

$\overline{J_l}=h^{\sum_{i=0}^{l}\sigma_{r_{l,i}}+\sum_{i=0}^{l-1}\sigma_{\gamma_{l,i}}}F_l^{\rho}\prod_{i=0}^{l}D_i^{\sigma_{a_{l-i}}}$이다.

위에서 언급한 바와 같이 $\overline{G_{2k}}=\overline{H_{2k}}=0$일 것이다.

이렇게 계산된 값을 랜덤오라클 $H(\cdot)$에 입력하여 얻은 값을 ρ와 비교하여 검증한다.

정리 1. 위에서 제시한 PE-ZKP 기법은 랜덤오라클 모델에서 ZKP protocol이다.

증명. 다음 장의 일반화된 PE-ZKP의 증명에 의하여 위 정리가 참임을 알 수 있다. □

Ⅳ. PE-ZKP 기법의 일반화

이 장에서는 전 장에서 기술한 PE-ZKP 기법을 일반화하고 안전성을 증명한다. 먼저 이해를 돕기 위하여 prover가 알고 있는 값과 공개된 값을 다시 정리하고 증명해야 하는 내용이 무엇인지 확인한 후, prover를 위한 PE-ZKP와 verifier를 위한 PE-ZKP로 나누어 설명한다.

우선 $a_i \in Z_p$에 대하여 선형식 $L(\vec{x})$를 다음과 같이 정의하자.

$$L(\vec{x}) := \sum a_i x_i,$$

여기서 $\vec{x}=(x_1,x_2,...,x_k)$라 하자. 이제 prover만 알고 있는 값은

- $(a_0,...,a_k),(s_0,...,s_k),(r_0,...,r_{2k^2-1})$이고prover가 사전에 암호화하여 공개한 값[1]은
- $(A_0,B_0),...,(A_k,B_k)$로서 각 값을 위 선형식을 이용하여 표시하면 $A_i=g^{L(\vec{s_i})}, B_i=g^{L(\vec{a_i})}h^{L(\vec{s_i})}$이며 여기서 $\vec{a_i}=(a_{i_1},...,a_{i_t}) \subset (a_0,...,a_k)$.

그리고 prover가 이용할 공개값은

- $(C_0,D_0),...,(C_k,D_k)$이다.

Ⅲ장에서 설명한 바와 같이 prover가 계산한 값은

- $(E_0,F_0),...,(E_{2k},F_{2k})$로서 모든 $0 \le i \le 2k$와 모든 $0 \le j,\ell \le k$에 대하여 $E_i = C_j^{L(\vec{a_\ell})} g^{L(\vec{r_i})}$이고 $F_i = D_j^{L(\vec{a_\ell})} h^{L(\vec{r_i})}$이다.

이때 첨자의 사용에 매우 주의가 필요한데, 그 이유는 $C_j^{L(\vec{a_\ell})}$가 의미하는 것은 $i=j+\ell$이 되도록 계수간의 곱셈과 덧셈이 이루어져야 한다. 물론 주의 깊게 조정하여 Ⅲ장에서 제시한 것과 같이 기술한 것이다.

마지막으로 prover가 증명하고 싶은 것은 $\vec{a},\vec{s}$와 $\vec{r}=(r_0,...,r_{2k})$에 대하여 다음 4개의 선형식이 모두 만족됨을 보이는 것이다.

- $L(\vec{a_i})-a_i=0$
- $L(\vec{s_i})-s_i=0$
- $\vec{b_j}L(\vec{a_\ell})-c_i=0$
- $L(\vec{r_i})-r_i=0$

이제 prover와 verifier로 구분하여 PE-ZKP가 어떻게 수행되는지 제시한다.

4.1 Prover의 일반화된 PE-ZKP

일반적인 3-라운드 ZKP 기법과 같이 Commitment, Challenge 및 Response 단계로 나누어 설명한다.

4.1.1 Commitment 계산

먼저 $u^{(\alpha)}$는 prover가 알고 있는 값 α를 commit하기 위해 이용하는 난수를 의미한다.

이제 $L(\overrightarrow{u_i^{(a)}})-u_i^{(a)}=0$을 만족하는 $(u_0^{(a)},...,u_k^{(a)})$과 $L(\overrightarrow{u_i^{(s)}})-u_i^{(s)}=0$를 만족하는 $(u_0^{(s)},...,u_k^{(s)})$를 선택한 후, $(G_0,H_o),...,(G_k,H_k)$를 계산한다, 여기서

$G_i = g^{L(\overrightarrow{u_i^{(s)}})}, H_i = g^{L(\overrightarrow{u_i^{(a)}})} h^{L(\overrightarrow{u_i^{(s)}})}$이다.

이어서 모든 $(I_0,J_0),...,(I_{2k},J_{2k})$를 모두 계산하는데

$I_i = C_j^{L(\overrightarrow{u_\ell^{(a)}})} g^{L(\overrightarrow{u_i^{(r)}})}$,

$J_i = D_j^{L(\overrightarrow{u_\ell^{(a)}})} h^{L(\overrightarrow{u_i^{(r)}})}$

을 만족하고 추가로 $\vec{b_j}L(\overrightarrow{u_\ell^{(a)}})-\overrightarrow{u_i^{(c)}}=0$도 만족하면서 선형 방정식 $L(\overrightarrow{u_i^{(r)}})-u_i^{(r)}=0$을 만족하는 $(u_0^{(r)},...,u_{2k^2-1}^{(r)})$을 선택한다.

4.1.2 Challenge 계산

Prover의 challenge 값 ρ는 다음과 같이 계산한다.

$$\begin{aligned}\rho = H(&g,h,A_0,B_0,...,A_k,B_k,\\ &C_0,D_0,...,C_k,D_k,\\ &E_0,F_0,...,E_{2k},F_{2k},\\ &G_0,H_0,I_0,J_0,...,G_{2k},H_{2k},I_{2k},J_{2k})\end{aligned}$$

4.1.3 Response 계산

Prover는 commitment와 challenge를 사용하여 다음 response 값들을 계산한다.

1) 사용자가 임의의 값을 commit하는 다양한 방법이 있으나, 본 연구에서는 암호화 기법을 commitment 기법으로 사용한다. 검증은 평문 m과 난수 r을 공개하면 검증 가능하다.

- $\sigma_i^{(a)} = L(\overrightarrow{u_i^{(a)}}) - \rho \cdot L(\overrightarrow{a_i})$
- $\sigma_i^{(s)} = L(\overrightarrow{u_i^{(s)}}) - \rho \cdot L(\overrightarrow{s_i})$
- $\sigma_j^{(r)} = L(\overrightarrow{u_j^{(r)}}) - \rho \cdot L(\overrightarrow{r_j}),\ 0 \le i \le k, 0 \le j \le 2k$

4.1.4 Proof 전송

마지막으로 prover는 PE-ZKP의 proof 값으로 다음 π를 verifier에게 전송한다.

$$\pi = (\rho, \sigma_0^{(a)},...,\sigma_k^{(a)}, \sigma_0^{(s)},...,\sigma_k^{(s)}, \sigma_0^{(r)},...,\sigma_{2k}^{(r)}) \quad (8)$$

4.2 Verifier의 일반화된 PE-ZKP

검증자는 수식 (8)과 같이 계산된 proof π를 수신하여 아래 4개의 값 $(\overline{G_i}, \overline{H_i}, \overline{I_j}, \overline{J_j})$을 모든 첨자 $0 \le i \le k, 0 \le j \le 2k$에 대하여 차례로 계산한 후, 랜덤오라클을 수행하여 $\overline{\rho} = H(\cdot)$을 얻은 후, ρ와 비교한다. 만약 같으면 PE-ZKP를 수락하고 다른 경우 실패를 알린다.

- $\overline{G_i} = g^{\sigma_i^{(s)}} A_i^{\rho}$
- $\overline{H_i} = h^{\sigma_i^{(s)}} B_i^{\rho} g^{\sigma_i^{(a)}}$
- $\overline{I_j} = g^{\sigma_{\ell_j}^{(r)}} E_j^{\rho} \prod_{j=v+w} C_w^{\sigma_v^{(a)}}$
- $J_j = h^{\sigma_{\ell_j}^{(r)}} F_j^{\rho} \prod_{j=v+w} D_w^{\sigma_v^{(a)}}$.

4.3 일반화

이제 일반화의 단계이다. 4.2장까지의 내용은 사실 Ⅲ장의 내용을 변수를 바꾸어 정리한 것 뿐이다. 좀 더 개념적으로 설명하면 내용만 다른 4개의 3라운드 ZKP를 수행한다고 할 수도 있다. 그러므로 하나의 선형식 (Linear Equation)에 대한 PE-ZKP만 있다면 앞 장에 서술된 모든 내용을 포괄할 수 있을 것이다. 결국 다항식의 일반화는 선형식의 상등성을 증명하는 것으로 볼 수 있다. 그래서 이것을 L2E-ZKP (ZKP of Linear Equation Equality)라 부를 것이다.

이제 L2E-ZKP를 기술할 것인데, prover는 먼저 다음 값들의 목록을 공개한다:

$(A_1,...,A_n) := (g^{L(\overrightarrow{a_1})},...,g^{L(\overrightarrow{a_n})})$,

여기서 $\overrightarrow{a_i} = (a_{i_1},...,a_{i_k}) \in (Z_p)^k$로서 집합으로 표시할 때, $\overrightarrow{a_i} \subset \overrightarrow{a} = (a_0,...,a_\ell)$이다.

이제 이 표기와 앞 장들의 내용을 바탕으로 본 논문의 최종 목표인 L2E-ZKP 기법을 제시한다.

4.3.1 Prover의 L2E-ZKP

(1) Commitment

모든 $1 \le i \le n$에 대하여 $L(\overrightarrow{x_i}) = 0$을 만족하는 $(u_1,...,u_\ell) \in (Z_p)^\ell$을 선택한 후 $\Gamma_1 = g^{L(\overrightarrow{u_1})},...,\Gamma_n = g^{L(\overrightarrow{u_n})}$을 계산한다.

(2) Challenge

랜덤오라클 H를 이용하여 $\rho = H(g, A_1,...,A_n, \Gamma_1,...,\Gamma_n)$를 계산한다.

(3) Response

모든 $1 \le j \le \ell$에 대하여 $\sigma_j = u_j - \rho \cdot a_j$를 계산한 후, $\pi = (\rho, \sigma_1,...,\sigma_\ell)$을 전송한다.

4.3.2 Verifier의 L2E-ZKP

모든 $0 \le j \le \ell$에 대하여 $\Lambda_i = g^{L(\sigma_i)} A_i^{\rho}$을 계산한 후, $\overline{\rho} = H(g, A_1,...,A_n, \Lambda_1,...,\Lambda_n)$의 결과인 $\overline{\rho}$와 수신한 ρ를 비교하여 같은 경우에만 증명을 수락한다.

정리 2. 랜덤오라클 모델에서 위 L2E-ZKP 기법은 선형식의 상등성에 대한 ZKP 기법이다.

증명. Zero-knowledge proof임을 증명하기 위해서는 완전성 (Completeness), 건전성(Soundness), 영지식성 (Zero-knowledgeness)을 각각 증명해야 한다. 먼저 가장 쉬운 완전성부터 살펴보자.

(1) 완전성

완전성이 의미하는 것은 prover가 실제로 아는 값을 사용하여 프로토콜을 수행하면 언제나 성공한다는 것을 보장해준다. 간략히 다음만 보이면 충분하다.

$$\begin{aligned}\Lambda_i &= g^{L(\overrightarrow{\sigma_i})} A_i^{\rho} \\ &= g^{\sum_{j=0}^{k}(u_{i_j} - \rho \cdot a_{i_j})} g^{\rho \cdot L(\overrightarrow{a_i})} \\ &= g^{\sum_{j=0}^{k}(u_{i_j} - \rho \cdot a_{i_j})} g^{\rho \cdot \sum_{j=0}^{k} a_{i_j}} \\ &= g^{\sum_{j=0}^{k}(u_{i_j} - \rho \cdot a_{i_j} + \rho \cdot a_{i_j})} \\ &= g^{\sum_{j=0}^{k} u_{i_j}} \\ &= g^{L(\overrightarrow{u_i})} = \Gamma_i.\end{aligned}$$

(2) 건전성

건전성은 악의적인 prover가 모르는 값을 이용하여 증명을 성공적으로 수행하는 것을 방지하는 것을 보장한다.

이를 위하여 $(a_1,...,a_\ell)$을 알지 못하는 악의적 prover $\tilde{P}$가 verifier의 검증을 통과할 수 없어야 한다. 먼저 g,H는 공개된 것으로 하자. $\tilde{P}$는 P가 알고 있는 $(a_1,...,a_\ell)$을 모르기 때문에 임의의 값으로 $(\alpha_1,...,\alpha_\ell)$을 선택해야 한다. 표기를 좀 더 간략하게 하면 $\vec{a_i}$대신 $\vec{\alpha_i}$를 사용하고 $\vec{u_i}$ 대신 $\vec{v_i}$를 사용한다고 하자. 그러면 $\tilde{P}$의 response에 대하여

$$L(\vec{\sigma_i})+\rho\cdot L(\vec{\alpha_i})=L(\vec{v_i}) \tag{9}$$

를 만족할 것이며, 각 A_i에 대하여

$g^{L(\vec{\sigma_i})}A_i^{\rho}=g^{L(\vec{\beta_i})}$일 것이므로

$$L(\vec{\sigma_i})+\rho\cdot L(\vec{a_i})=L(\vec{\beta_i}). \tag{10}$$

식 (9)에서 식 (10)을 빼면

$$\rho\cdot(L(\vec{\sigma_i})-L(\vec{a_i}))=L(\vec{v_i})-L(\vec{\beta_i}) \tag{11}$$

을 얻게 된다. 위 식 (11)에서 $L(\vec{\alpha_i})\neq L(\vec{a_i})$이라면

$$\rho=\frac{L(\vec{v_i})-L(\vec{\beta_i})}{L(\vec{\alpha_i})-L(\vec{a_i})}$$

를 의미한다. 그런데 랜덤오라클 모델에서 ρ는 정규분포에 따라 결정되며 H의 입력과 독립적으로 그 값이 결정된다. 그러므로 이렇게 랜덤하게 선택한 값을 사용하여 동일한 ρ를 출력하는 것은 무시할 만한 확률로 가능하다.

(3) 영지식성

영지식성은 prover P와 verifier V간에 주고받는 통신의 내용이 prover와 verifier가 아는 값을 전혀 모르는 상태에서 실제 P,V간에 주고받은 통신과 구별할 수 없는 통신내용을 만들어내는 것이 가능하다 것을 보장하므로 실제 통신내용이 비밀정보를 누설하지 않는다는 것을 보장한다. 이를 위하여 이러한 통신내용을 만드는 다항식 시간 simulator를 보이면 된다.

이제 $\vec{a_i}$를 모르는 상태에서 악의적 verifier $\tilde{V}$의 view를 만드는 simulator Δ를 제시한다. Δ가 랜덤오라클도 simulate해야 하므로 $\tilde{V}$가 랜덤오라클에 query할 경우 이전 query에 사용하지 않은 새로운 난수값을 생성하고 저장한 후 $\tilde{V}$에게 돌려준다. 만약 이전에 사용한 값으로 query하면 난수를 저장하는 테이블을 탐색하여 이전에 돌려준 값을 다시 전송한다. 만약 정직한 prover P가 $g,\vec{a}$를 사용하여 L2E-ZKP에 참여하였기 때문에 Δ는 임의의 $\rho,\vec{\sigma}$를 생성할 것이다. 이어서 $A_i=g^{L(\vec{\sigma_i})}A_i^{\rho}$를 모든 i에 대하여 계산한 후, $(g,A_1,...,A_n,A_i,...,A_n)$ 값에 대한 랜덤오라클의 출력으로 지정한다. 그러면 Δ가 만든 proof는 $\tilde{V}$를 통과할 것이고 이때의 view는 $\tilde{V}$가 보는 view이다. 또한 Δ는 다항식 횟수만큼 수행되므로 정리 2가 증명된다.

□

Ⅴ. 결 론

본 논문에서는 Kissner와 Song에 의하여 제안된 집합연산 기법에서 폭넓게 사용된 PE-ZKP 기법을 구체적으로 기술하고 이것을 일반화하는 방법에 대하여 살펴보았다. 특히 일반화된 PE-ZKP인 L2E-ZKP를 랜덤오라클 모델에서 안전하다는 것을 증명할 수 있다. 본 논문에서 제시한 일반화된 PE-ZKP의 연산량과 통신량을 개선하기 위한 연구가 추가 연구로서 남아있다.

References

[1] R. Cramer, M. Franklin, B. Scheonmakers, and M. Yung, “Multi-authority secret-ballot elections with linear work,” *Advances in Cryptology-Eurocrypt LNCS*, vol. 1070, pp. 72-83, 1996.

[2] T. ElGamal, “A public-key cryptosystem and a signature scheme based on discrete logarithms,” *Advances in Cryptology-Crypto LNCS,* pp. 10-18, 1985.

[3] S. Goldwasser, S. Micali, and C. Rackoff, “The knowledge complexity of interactive proof systems,” *SIAM J. Comput.*, vol. 18, no. 1, pp. 186-208, 1989.

[4] O. Goldreich, S. Micali, and A. Wigderson,

"Proofs that yield nothing but their vality," *J. ACM*, vol. 38, no. 3, pp. 690-728, 1991.

[5] O. Goldreich, *Foundations of cryptography vol. 1*, Cambridge Press, 2004.

[6] L. Kissner and D. Song, "Privacy-preserving set operation," *Advances in cryptology-Crypto LNCS*, vol. 3621, pp. 241-157, 2005.

[7] P. Paillier, "Public-key crytpsystems based on composite degree residuosity classes," *Advances in cryptology-Eurocrypt LNCS*, vol. 1592, pp. 223-238, Apr. 1999.

김 명 선 (Myungsun Kim)

1994년 8월 : 서강대학교 컴퓨터공학과 졸업

2002년 2월 : KAIST 컴퓨터공학부 졸업

2012년 8월 서울대학교 수리과학부 졸업

2012년 9월~현재 : 수원대학교 정보보호학과

<관심분야> 암호학 및 안전한 다자간 연산

강 보 람 (Bolam Kang)

2011년 3월~현재 : 수원대학교 정보보호학과 학사과정

<관심분야> 정보보안