JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 29 No. 8, pp. 103-112, August 2024
https://doi.org/10.9708/jksci.2024.29.08.103

# Input Certification protocol for Secure Computation

Myoungin Jeong*

*Assistant Professor, Dept. of Mathematics, Korea Military Academy, Seoul, Korea

## [Abstract]

This study was initiated with the aim of authenticating that inputs have not been tampered with without disclosing them in the case of computations where multiple inputs are entered by participants using the same key. In general, in the authentication stage, authentication is performed after the input value is disclosed, but we do not want to reveal the inputs until the end. This is a case of deviating from the traditional security model in which malicious participants exist in cryptography, but it is a malicious attack method that can actually occur enough. Privacy infringement or distortion of calculation results can occur due to malicious manipulation of input values. To prevent this, this study studied a method that can authenticate that the message is not a modified message without disclosing the message using the signature system, zero-knowledge proof, and commitment scheme. In particular, by modifying the ElGamal signature system and combining it with the commitment scheme and zero-knowledge proof, we designed and proved a verification protocol that the input data is not a modified data, and the efficiency was improved by applying batch verification between authentication.

▸ **Key words**: ElGamal signature scheme, commitment scheme, Zero-Knowledge proof of Knowledge, input certification, batch verification

## [요 약]

본 연구는 계산에 참여하는 참가자가 다수의 입력값을 동일한 키를 사용하여 입력하는 계산의 경우 입력값을 공개하지 않으면서 입력값이 변형되지 않았다는 인증을 목적으로 시작하였다. 일반적으로 인증 단계에서는 입력값을 공개 후 인증을 실시하나 입력값을 끝까지 공개하지 않고자 하는 것이다. 이는 암호학에서 악의적인 참가자가 존재하는 전통적인 보안 모델을 벗어나는 경우이지만, 실제로 충분히 일어날 수 있는 악의적인 공격 방법이다. 악의적인 의도를 가진 입력값 조작으로 프라이버시 침해, 또는 계산 결과의 왜곡이 일어날 수 있다. 본 연구에서는 이를 방지하기 위해 서명 체계, 영지식증명, commitment scheme을 이용하여 메시지를 공개하지 않고 해당 메시지가 변조되지 않은 메시지임을 인증(input certification)하는 방법에 대해 연구하였다. 특히, ElGamal 서명 체계를 수정하여 commitment scheme과 영지식 증명과 결합하여 입력된 데이터가 변조되지 않은 데이터라는 인증이 가능한 프로토콜을 설계하여 증명하였고, 인증 간에 batch verification을 적용하여 효율성을 향상시켰다.

▸ **주제어**: ElGamal 서명, commitment scheme, 영지식증명, 입력 인증, batch verification

• First Author: Myoungin Jeong, Corresponding Author: Myoungin Jeong
*Myoungin Jeong (mangjj@kma.ac.kr), Dept. of Mathematics, Korea Military Academy
• Received: 2024. 06. 18, Revised: 2024. 07. 24, Accepted: 2024. 07. 29.


http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

## I. Introduction

컴퓨팅 기술의 발전으로 클라우드 환경이 보편화되고 장비의 연산 속도가 빨라지면서 과거에 기술의 한계로 실용화되지 못했던 기술들이 최근 많이 실용화되고 있다. 그 중 하나가 안전한 다자간 계산으로, 이것은 여러 명의 참여자가 각자의 입력을 입력하여 이를 공개하지 않고 원하는 연산의 결과를 얻을 수 있게 하는 방법의 하나이다. 예를 들면, 안전한 다자간 계산을 이용하면 민감한 개인정보를 다루는 병원, 공공기관, 금융기관 등이 타 기관과 협업할 때, 자기 기관의 정보를 타 기관에 모두 공개하지 않으면서 원하는 계산 결과만 도출할 수 있게 할 수 있다. 자기 정보를 모두 공개하지 않는다는 프라이버시 보호를 달성하는 것뿐만 아니라 정확한 계산 결과를 도출하는 것도 중요한 목적 중 하나이다. 여러 기관의 민감한 정보를 다수가 참여하여 다루기 때문에 프라이버시 보호가 가능한 안전한 계산 방법에 관한 연구가 이루어져야 하고, 또한 참여자의 정보를 다른 참여자에게 모두 공개하지 않기 때문에 참여자들이 다른 참여자들의 입력이 조작되지 않은 진실한 입력이라는 확신을 가질 수 있어야 한다. 이것은 암호학적으로 악의적인 의도를 가진 참여자가 있는 전통적인 보안 모델을 벗어난 경우지만 심각한 보안 위협이 될 수 있다. 본 연구는 전통적인 ElGamal 서명 체계를 수정하여 영지식증명(Zero-Knowledge proof of Knowledge, ZKPK), Commitment scheme, batch verification을 적용하여 다량의 입력이 있는 경우 입력 데이터의 변조를 방지할 수 있는 효율성이 향상된 입력 인증 프로토콜에 대해 제안하고자 한다.

기존의 입력 인증에 관한 연구는 게임이론과 Garbled Circuit에 기반을 두고 주로 진행되었다. 2004년 Halper과 Teague[1], 2013년 Wallrabenstein와 Clifton[2]은 게임이론을 이용하여 합리적인 참여자가 입력값을 진실하게 입력하는 방법에 관해 연구하였고, 2016년 Blanton과 Bayatbabolghani[3]에 의해 Garbled Circuit(GC)에 기반을 둔 입력 인증에 관해 연구하였다. 그러나 GC는 대수적인 구조를 갖춘 서명 체계나 인증 기법과 결합하는 데 어려움이 있어서 대수적 구조를 바탕으로 한 일반적인 비밀 공유 기법을 기반으로 한 설정에서의 입력 인증 문제에 관한 연구가 필요하다. 2023년 Jeong[4]은 CL 서명 체계에 바탕을 둔 입력 인증 프로토콜에 관해 연구하였다. 이는 대수적 구조를 바탕으로 암호학적 기초 요소를 접목한 입력 인증 연구라고 할 수 있다. 본 연구에서는 일반적으로 사용되는 전통적인 ElGamal 서명 체계를 바탕으로 하여 입력을 인증하는 방법에 관해 연구한다. 구체적으로 2장에서 ElGamal 서명체계의 정의, commitment scheme, 영지식증명과 서명체계의 안전성을 증명할 수 있는 실험 등에 대해 정의하며 3장에서 ElGamal 서명 체계에 기반한 입력 인증 스킴과 배치 인증에 대해 제안하고 계산량을 비교한 뒤 마지막으로 결론을 맺는다.

## II. Preliminaries

함수 $\epsilon : \mathbb{N} \rightarrow \mathbb{R}_{\geq 0}$이 모든 양의 다항식 $p(\cdot)$에 대해 $x > N$일 때 $\epsilon(x) < \frac{1}{p(x)}$을 만족하는 $N$이 존재하면 $\epsilon$을 무시할 수 있는 함수(negligible function)라 하며 negl로 표기하자. 여기에서 $x$는 보안 매개변수(security parameter)이다.

### 1. ElGamal Signature Scheme

서명 체계는 메시지에 디지털 서명을 생성하고 검증하는 알고리즘으로 메시지의 유효성을 검사하고 보낸 메시지에 대한 부인방지 기능으로 제공한다. 전자서명은 공개키 암호 기술을 이용하여 서명을 생성하고 검증할 수 있다. 일반적으로 서명 체계는 키를 생성하는 키 생성(Key generation), 서명을 생성하는 서명(Sign), 서명 검증(Verification) 이렇게 세 개의 알고리즘으로 구성된다.

*Definition 1. (Signature Scheme)*

**KeyGen**: 확률적 다항시간(probabilistic polynomial-time, PPT) 알고리즘. 공개키, 개인 키 쌍 $(pk, sk)$을 생성

**Sign**: 개인 키 $sk$, 메시지 $m$을 입력으로 하는 PPT 알고리즘. 서명 $\sigma$을 생성

**Verify**: 공개키 $pk$, 메시지 $m$, 서명 $\sigma$을 입력으로 하는 결정적 다항시간(deterministic polynomial-time) 알고리즘. 한 비트(0 또는 1)를 출력

$\Pi = (\mathrm{KeyGen}, \mathrm{Sign}, \mathrm{Verify})$을 서명 체계라 한다.

서명 체계의 안전성은 "PPT 공격자에 의한 선택된 메시지 공격(chosen message attack, CMA) 하에서 실존적 위조(existential forgery)가 어렵다는 것"으로 정의된다. 실존적 위조란 합법적인 서명자로부터 서명된 적 없는 적어도 하나의 메시지, 서명 쌍 $(m, \sigma)$을 생성하는 것을 말한다. 서명 체계의 안전성을 정의하기 위해 다음과 같은 실험을 이용한다.

*Experiment* $\mathrm{ForgeSig}_{\mathrm{A},\Pi}(x)$ :

1. 도전자는 공개키, 비밀키 쌍 $(pk, sk) \leftarrow \mathrm{KeyGen}(1^x)$을 생성하여 공개키 $pk$를 $\mathcal{A}$에게 보낸다.
2. $\mathcal{A}$는 오라클 $\mathrm{Sign}_{sk}(\cdot)$에 접근할 수 있고 $\mathcal{A}$가 오라클에 질의하는 각 메시지 $m$은 리스트 $Q$에 저장되며 $\mathcal{A}$는 $\mathrm{Sign}_{sk}(\cdot)$로 생성된 서명 $\sigma = \mathrm{Sign}_{sk}(m)$를 안다. $\mathcal{A}$는 최종적으로 $(m^*, \sigma^*)$를 출력한다.
3. $\mathrm{Verify}_{pk}(m^*, \sigma^*) = 1$이고 $m^* \notin Q$일 경우 실험의 출력은 1이며, 이 외에는 0이다.

위의 실험을 이용하여 서명 체계의 안전성을 다음과 같이 정의한다.

*Definition 2. (Security of a signature scheme)*

만약 모든 PPT adversary $\mathcal{A}$에 대해

$$\Pr[\mathrm{ForgeSig}_{\mathcal{A},\Pi}(x) = 1] \leq \mathrm{negl}(x)$$

을 만족하는 무시할 수 있는 함수 negl이 존재하면 서명 체계 $\Pi = (\mathrm{KeyGen}, \mathrm{Sign}, \mathrm{Verify})$는 adaptive chosen message attack에 대해 실존적으로 위조 불가능하다고 한다.

ElGamal 서명 체계는 1985년 Taher ElGamal이 고안한 이산로그 문제에 기반한 서명 체계이다[5]. 본 연구에서 위조 가능성이 있는 원래의 서명 체계 대신 Pointcheval과 Stern[6][7]의 수정 서명 체계를 사용한다. 수정된 서명 체계에서는 소수 $q$를 사용하는 것보다 이산로그 문제의 어려움을 기반으로 할 때 좀 더 일반적인 $\alpha$-hard 소수 $p$를 정의한다. $\alpha$-hard 소수란 어떤 고정된 $\alpha$에 대해 $R \leq |p|^{\alpha}$와 $p - 1 = qR$을 만족하는 소수 $q$를 말한다. 이 서명 체계에서는 위조 불가능성이 증명된 랜덤 오라클 해시함수 $H$를 사용한다. ElGamal 서명 체계의 **KeyGen, Sign, Verify** 알고리즘은 다음과 같다.

**KeyGen**: 입력: 보안 매개변수 $1^x$, $\alpha$-hard 소수 $p$와 $\mathbb{Z}_p^*$의 제너레이터 $g$를 선택. 랜덤넘버 $x \in \mathbb{Z}_{p-1}$을 선택한 후 $y = g^x \bmod p$ 계산. 출력: 비밀키 $sk = x$, 공개키 $pk = (p, g, y)$

**Sign**: 입력: 메시지 $m$, $sk = x$, $pk = (p, g, y)$. 랜덤넘버 $k \in \mathbb{Z}_{p-1}^*$을 선택한 후 $t = g^k \bmod p$와 $s \equiv (H(m\|t) - \quad - xt)k^{-1} \bmod p - 1$계산. 출력: 서명 $\sigma = (t, s)$

**Verify**: 입력: $m$, $pk = (p, g, y)$, $\sigma = (t, s)$. $1 < t < p$와 $g^{H(m,t)} \equiv y^t t^s \bmod p$인지 확인. 출력: 만약 둘 다 만족하면 1을, 그렇지 않으면 0을 출력

## 2. Commitment scheme

commitment scheme은 메시지 $m$을 공개하지 않고 메시지에 커밋을 하여 commitment를 생성하여 메시지 대신 생성된 commitment를 계산에 이용한다. 한 번 커밋이 이루어지면 committer는 처음 사용한 메시지 $m$을 임의로 바꿀 수 없다. 즉, $m$에 대한 commitment가 주어지면 메시지 $m$은 비공개로 유지되며, $m$ 이외의 값으로는 그것을 열 수 없다. commitment scheme의 이러한 속성을 숨기기(hiding)와 묶기(binding)라고 한다. commitment scheme은 메시지 $m$에 대한 커밋이 이루어지는 **Commit** 알고리즘과 commitment가 메시지 $m$에 대한 것인지 확인하는 **Open** 알고리즘으로 구성된다. **Commit** 알고리즘에서 커밋이 이루어질 때 랜덤 넘버 $r$을 이용하기 때문에 $com(m, r)$로 표기한다. 본 논문에서는 잘 알려진 이산로그에 기반한 Pedersen commitment scheme[8]을 사용한다. Pedersen commitment는 소수 order $q$를 갖는 그룹 $G$와 그룹 $G$의 두 개의 제너레이터 $g, h$에 메시지 $m \in \mathbb{Z}_q$과 랜덤 넘버 $r \in \mathbb{Z}_q$을 이용하여 $com(m, r) = g^m h^r$로 정의된다. **Open** 알고리즘에서는 $r$과 $m$을 밝혀야 한다. Pedersen commitment scheme은 이산로그 문제의 어려움에 기반하여 information theoretically hiding과 computationally binding이 성립한다.

## 3. Zero Knowledge Proof of Knowledge

영지식증명은 prover와 verifier 간의 양자 간 상호작용 프로토콜로 prover가 공개하고 싶지 않은 정보를 공개하지 않으면서 verifier에게 자신의 진술(statement)이 진실하다는 것을 증명할 수 있다. 공개하고 싶지 않은 변수(variables)와 진술을 사용한 ZKPK는 다음과 같이 표기한다: $PK\{(\mathrm{variables}): \mathrm{statement}\}$. 변수는 공개하지 않고 prover만 알고 있는 정보이고, 진술은 prover와 verifier 모두가 알고 있다. 이 프로토콜이 성공한다면 prover는 verifier에게 변수가 무엇인지 공개하지 않으면서 자신이 해당 변수를 알고 있다는 사실을 증명하는 데 성공한 것이다. 본 연구에서는 이산로그에 기반한 영지식증명의 테크닉을 이용한다[9].

## 4. Batch Verification

여러 개의 서명을 동시에 인증해야 할 때 Batch Verification[10]을 사용하여 한 번에 인증함으로써 효율

성을 높일 수 있다. 본 논문에서는 여러 개의 기관이 계산에 참여할 때 자신의 메시지들은 모두 하나의 키를 사용하여 서명한 경우에 대해 다루었다.

*Definition 3. (Batch verification of signatures[11])*
보안 매개변수 $x$에 대한 서명 체계 $\Pi = (\mathrm{KeyGen}, \mathrm{Sign}, \mathrm{Verify})$을 이용하여 $n$명의 참여자 $P_1, P_2, \cdots, P_n$이 $\mathrm{KeyGen}(1^x)$ 알고리즘을 통해 만든 키 쌍들을 $(pk_1, sk_1), \cdots, (pk_n, sk_n)$라 하고, 공개키들을 $PK = \{pk_1, \cdots, pk_n\}$라 하자. $(pk_i, m_i, \sigma_i)$들을 입력으로 하는 PPT 알고리즘 **Batch**는 출력으로 한 비트를 출력하며, 다음 성질을 만족하면 batch verification이다.
- 모든 $i \in [1, n]$에 대해 $pk_i \in PK$이고 $\mathrm{Verify}(pk_i, m_i, \sigma_i) = 1$이면 $\mathrm{Batch}((pk_1, m_1, \sigma_1), \cdots, (pk_n, m_n, \sigma_n)) = 1$이다.
- 모든 $i \in [1, n]$에 대해 $pk_i \in PK$이고 적어도 하나의 $i \in [1, n]$에 대해 $\mathrm{Verify}(pk_i, m_i, \sigma_i) = 0$이면서 $\mathrm{Batch}((pk_1, m_1, \sigma_1), \cdots, (pk_n, m_n, \sigma_n)) = 1$일 최대 확률은 $2^{-x}$이다.

유효하지 않은 하나의 서명이 포함된 여러 개의 서명들을 인증할 때 Batch verification을 사용하면 verifier는 적어도 $1 - 2^{-x}$의 확률로 이를 발견할 수 있으며, 이것은 서명들을 각각 인증하는 것보다 속도가 빠르다.

## 5. Signature scheme with privacy

일반적으로 서명을 검증할 때는 원본 메시지 $m$과 서명에 사용한 랜덤넘버 $r$을 공개한다. 본 연구에서는 서명 체계의 인증 단계에서 서명 생성에 사용된 원본 메시지를 공개하지 않고 서명을 인증하는 방법에 관해 연구한다. 이러한 서명 체계를 signature scheme with privacy라고 부르고, 원본 메시지를 공개하지 않는 인증을 전통적인 인증 방법과 구분하기 위해 private verification이라 부른다. 이 서명 체계는 서명 자체만 가지고는 원본 메시지에 대한 정보를 알 수 없다는 특징을 갖는다. signature scheme with privacy를 다음과 같이 정의한다.

*Definition 4. (Signature Scheme with privacy)* Signature scheme with privacy는 다음과 같은 3개의 PPT 알고리즘으로 구성된다.

**KeyGen**: 보안 매개변수 $1^x$인 PPT 알고리즘. 공개키, 개인 키 쌍 $(pk, sk)$을 생성

**Sign**: 입력이 $sk$, 메시지 $m$인 PPT 알고리즘. 서명 $\sigma$과 추가로 $x_\sigma$를 생성할 수 있음

**PrivVerify**: Prover와 Verifier 간의 대화형(interactive) 알고리즘. Prover: 메시지 $m$과 $(\sigma, x_\sigma)$를 알고 있으며, $x_m$과 서명 $\tilde{\sigma}$(수정될 수 있음)을 verifier에게 보냄. Verifier: 한 비트(0 또는 1)를 출력

**Sign** 알고리즘에서 추가로 생성할 수 있는 $x_\sigma$는 인증 단계에서 사용하는 $x_m$을 계산하는 데 랜덤넘버로 쓰일 수 있으며, **Sign** 알고리즘에서 $x_\sigma$이 생성되지 않는 경우 서명 $\sigma$과 메시지 $m$에 접근이 가능한 누구라도 $x_m$을 계산할 수 있다.

위조 불가능성(unforgeability)을 달성하기 위해서는 **PrivVerify** 단계에서 **Verify**와 마찬가지로 서명 $\tilde{\sigma}$를 검증하고 Prover가 서명 $\tilde{\sigma}$를 생성하는 데 사용한 메시지 $m$과 $x_m$을 생성하는 데 사용한 메시지가 같은지 확인하여야 한다. 본 연구에서 $x_m$은 랜덤넘버 $r$을 이용한 $m$에 대한 commitment 형태 $com(m, r)$이기 때문에 새로운 security definition에 commitment를 적용한다.

위조 불가능성을 정의하기 위한 signature scheme with privacy에 관한 실험은 앞서 서명 체계의 안전성을 정의하기 위해 사용한 ForgeSig와 유사하나 두 가지 측면에서 다르다. 첫째, 챌린지 페어 $(\tilde{\sigma}^*, x_{m^*})$를 선택한 뒤 공격자 $\mathcal{A}$는 $x_{m^*}$가 이전에 쿼리된 적 없는 서명의 메시지로 생성된 것인지 영지식증명으로 증명해야 한다. 둘째, 서명 위조 실험은 **Verify** 알고리즘 대신 수정된 인증 알고리즘인 **PrivVerify**를 호출한다. 서명이 commitment $x_{m^*} = com(m^*, r)$에 대한 것인지 검증하여야 하고, prover는 commitment 생성에 사용된 $m^*$의 정보를 알고 있는지도 증명해야 한다.

*Experiment* $\mathrm{ForgePrivSig}_{A,\Pi}(x)$ :
1. 도전자는 공개키, 비밀키 쌍 $(pk, sk) \leftarrow \mathrm{KeyGen}(1^x)$을 생성하여 공개키 $pk$를 $\mathcal{A}$에게 보낸다.
2. $\mathcal{A}$는 오라클 $\mathrm{Sign}_{sk}(\cdot)$에 접근할 수 있고 $\mathcal{A}$가 오라클에 질의하는 각 메시지 $m$은 리스트 $Q$에 저장되며 $\mathcal{A}$는 $(\sigma, x_\sigma) = \mathrm{sign}_{sk}(m)$를 안다.
3. 도전자와 $\mathcal{A}$는 **PrivVerify** 알고리즘을 진행하고, 이 과정에서 $\mathcal{A}$는 challenge pair $(x_{m^*}, \tilde{\sigma}^*)$를 공개한다. $\mathcal{A}$는 영지식증명으로 commitment $x_{m^*} = com(m^*, r)$의 opening

$(m^*, r)$을 알고 있음을 증명한다.($m^* \notin \boldsymbol{Q}$)

4. **PrivVerify**의 결과가 1이고 모든 다른 증명이 성공하면 출력은 1이며, 이 외에는 0이다.

Private verification을 정의하기 위해서 다음과 같이 signature scheme with privacy $\Pi = (\mathrm{KeyGen}, \mathrm{Sign}, \mathrm{PrivVerify})$를 위한 message indistinguishability 실험을 정의한다.

*Experiment* $\mathrm{MesInd}_{\mathrm{A},\Pi}(x)$ :

1. 도전자는 공개키, 비밀키 쌍 $(pk, sk) \leftarrow \mathrm{KeyGen}(1^x)$을 생성하여 공개키 $pk$를 $\boldsymbol{\mathcal{A}}$에게 보낸다.
2. $\boldsymbol{\mathcal{A}}$는 오라클 $\mathrm{Sign}_{\mathrm{sk}}(\cdot)$에 접근할 수 있고, 자신이 선택한 메시지에 대한 알고리즘의 출력을 안다. $\boldsymbol{\mathcal{A}}$는 $(m_0, m_1)$ 쌍을 출력한다.
3. 도전자는 랜덤 비트 $b \in \{0,1\}$를 뽑고, $\boldsymbol{\mathcal{A}}$의 요청에 따라 $(\sigma_b, x_{\sigma_b}) \leftarrow \mathrm{Sign}_{sk}(m_b)$ 알고리즘을 실행한다. 도전자는 $x_{m_b}$를 계산하고 $(\tilde{\sigma_b}, x_{m_b})$를 리턴한다($\tilde{\sigma_b}$는 $\sigma_b$로부터 생성). 만약 $x_{m_b}$ 또는 $\tilde{\sigma_b}$가 확률적(probabilistic)이라면 같은 서명에 대해 $\boldsymbol{\mathcal{A}}$는 여러 개의 $(\tilde{\sigma_b}^{(i)}, x_{m_b}^{(j)})$를 요청할 수 있다($i, j \in \mathbb{N}$). 이러한 서명 검증 요청은 원하는 만큼 반복할 수 있다.
4. $\boldsymbol{\mathcal{A}}$는 결과로 한 비트 $b'$을 출력한다. 만약 $b = b'$이면 실험의 출력은 1이고, 이 외의 경우 0이다.

*Definition 5. (Private Verification)*

만약 모든 PPT 공격자 $\boldsymbol{\mathcal{A}}$에 대해

$$\Pr[\mathrm{MesInd}_{A,\Pi}(x) = 1] \leq \frac{1}{2} + \mathrm{negl}(x)$$

을 만족하는 무시할 수 있는 함수 negl이 존재하면 서명 체계 $\Pi = (\mathrm{KeyGen}, \mathrm{Sign}, \mathrm{PrivVerify})$는 비공개 검증(private verification)을 달성했다고 한다.

기존의 영지식증명을 사용하여 privacy-preserving 서명 체계에 관한 연구[12][13]에서 사용한 정의는 서명된 메시지에 대한 정보가 드러나지 않고, 두 개의 서명에 대한 지식을 증명하는 프로세스를 연결할 수 없게(unlinkability) 하는 데 중점을 두었다. 위의 Private verification 정의는 같은(혹은 다른) 서명 체계가 서로 다른 시간에 검증된다는 사실을 숨기려 하지 않고, 궁극적으로 완전하게 보호하려고 하는 것은 서명된 메시지 그 자체라는 점에서 이전의 연구들과 차이가 있다. 본 연구에서 고려하고 경우는 사용자가 여러 계산에 자신의 데이터를 사용할 수 있고 동일한 데이터를 여러 번 사용했다는 사실을 숨길 필요가 없으므로 서명들의 unlinkability는 일반적으로 고려하지 않아도 된다. 따라서 서명된 값 자체만 보호하면 되고, 이것은 보안 수준을 유지하면서 서명 검증 프로세스를 더 빨리 진행할 수 있게 한다.

## III. The Proposed Scheme

안전한 계산을 위해 ElGamal 서명 체계를 바탕으로 commitment scheme과 결합한 구조를 제안하고자 한다. Pointcheval[6]이 제안한 adaptably chosen message 공격에 안전한 수정된 ElGamal 서명 체계를 사용하여 서명 검증 단계에서 메시지를 오픈하지 않고도 검증할 수 있도록 수정한다. 그리고 여러 개의 서명을 검증할 때 효율성을 향상시키기 위해 batch verification 방법 중 모든 서명이 유효할 때만 검증에 성공하는 small exponents 방법을 사용하는 알고리즘을 제안한다.

### 1. Construction based on ElGamal Signature

서명 검증 단계에서 메시지를 공개하지 않고 검증이 가능한 구조로 ElGamal 서명 체계를 수정하기 위해서 원본 메시지가 아닌 메시지 $m$의 commitment를 사용한다. 본 연구에서는 ElGamal 서명 체계를 그룹 $\mathbb{Z}_p^*$가 아닌 소수 $q$를 order로 갖는 $\mathbb{Z}_p^*$의 subgroup에서 정의한다. 이것을 통해 기존의 $\mathbb{Z}_p^*$에서의 ElGamal 서명 체계와 같은 수준의 안전성을 보장받으면서 서명 체계를 좀 더 단순화할 수 있고 다른 primitives를 적용할 수 있게 한다. 예를 들면, batch verification에서 사용할 small exponent 방법의 경우 소수가 아닌 order를 가진 그룹에서는 사용할 수 없는데[10], $\mathbb{Z}_q^*$에서의 ElGamal을 사용하면 small exponents 방법을 사용할 수 있다. 메시지 공개 없이 검증할 수 있게 검증알고리즘을 **PrivVerify**로 수정한 ElGamal 서명 체계 $\Pi = (\mathrm{KeyGen}, \mathrm{Sign}, \mathrm{PrivVerify})$를 다음과 같이 정의한다.

**KeyGen:** 입력: 보안 매개변수 $1^x$, 큰 소수 $q$를 order로 갖는 그룹 $G$와 제너레이터 $g$를 선택. 랜덤넘버 $x, u \in \mathbb{Z}_q$을 선택하여 $y = g^x, h = g^u \bmod q$를 계산. 출력: $sk = x$, $pk = (q, G, g, y, h)$

**Sign**: 입력: $m$, $sk = x$, $pk = (q, G, g, y, h)$. 랜덤넘버 $k, r \in \mathbb{Z}_q$을 선택하여 $t = g^k, x_m = com(m, r) = g^m h^r$, $s \equiv (H(x_m \| t) - xt)k^{-1} \pmod{q}$ 계산. 출력: $\sigma = (t, s)$, $x_\sigma = r$. (수신자는 $com(m, x_\sigma)$을 계산하여 서명 검증)

**PrivVerify**: prover는 $m$, $x_\sigma$, 그리고 $x_m = g^m h^{x_\sigma}$에 대한 서명 $\sigma = (t, s)$을 알고 verifier에게 $\sigma$와 $x_m$을 보냄. prover와 verifier 사이에 영지식증명 실행: $PK\{(\mu, \gamma): x_m = g^\mu h^\gamma\}$. 출력 : ZKPK를 통과하고 $g^{H(x_m \| t)} = y^t t^s$이 성립하면 1, 그렇지 않으면 0

수정된 ElGamal 서명 체계에서는 서명자가 $g$와 $h$를 선택하므로 메시지 $m$이 아닌 $m$에 대한 commitment인 $com(m, r)$을 공개할 수 있다. 만약 이렇게 하는 것이 보안상 취약하다면 $h$는 독립적인 기관 또는 서명자가 $g$에 대한 $h$의 이산로그를 알 수 없도록 다른 참여자가 선택해야 한다.

이 서명 체계가 unforgeable 하다는 사실은 Ⅱ장에서 기술한 ForgePrivSig 실험과 ***Definition 2***를 이용하여 증명할 수 있다. 직관적으로 설명하면, prover는 commitment에 대한 서명을 가지고 commitment의 opening인 메시지 $m$을 알고 있다는 증명을 해야 하고, prime order 그룹을 이용하여 [6]의 증명을 단순화할 수 있다.

***Theorem 1.*** 수정된 ElGamal 서명 체계는 랜덤 오라클 모델에서 adoptive chosen-message attack에 대해 실존적으로 위조할 수 없다(existentially unforgeable).

***Proof*** Pointcheval 등[6][7]은 $\alpha$-hard 소수를 모듈리로 사용하는 랜덤 오라클 모델에서 ElGamal 서명 체계가 안전함을 증명하였다. 소수 order를 사용하는 그룹에서 수정된 ElGamal 서명 체계도 마찬가지로 이 가정을 만족한다.

앞선 연구 [7]의 증명은 두 단계로 나눌 수 있는데, 우선 no-message 공격에 대한 안전성을 보이고, 다음으로 forking lemma를 이용하여 서명자가 구분할 수 없는 분포로 시뮬레이션할 수 있음을 보임으로서 adoptively chosen-message 공격에 대한 안전성을 보였다. 첫 번째 단계의 증명을 살펴보면, 두 가지 경우로 나눌 수 있다. 서명 $\sigma = (t, s)$의 $t$가 modulo $p-1$로 $p$와 서로소일 때와 그렇지 않을 때이다. 우리의 경우 계산이 modulo $q$로 이루어지기 때문에 두 번째 경우가 존재하지 않아서 첫 번째 경우만 고려하면 된다. 연구 [7]에서의 두 번째 단계의 증명도 우리의 경우 modulo $qR = p-1$이 아닌 지수에서 modulo $q$를 사용하기 때문에 단순화할 수 있다. 따라서 소수 order를 갖는 그룹 세팅을 이용하면 안전성을 더 쉽게 보일 수 있다.

**PrivVerify**를 위해 메시지 $m$이 아닌 메시지 $m$에 대한 commitment $x_m$을 사용하여 서명 검증을 하므로 위의 증명에서 $m$을 대신하여 $x_m$을 사용한다. 또한, $H$가 랜덤 오라클로 모델링되기 때문에 출력으로 설정할 수 있는 값에 융통성이 있다. 검증 단계에서는 prover가 $x_m$을 생성하는 데 사용한 $m$을 알고 있어야만 할 수 있는 $x_m$의 이산로그 표현의 proof of knowledge를 해야 한다. 이것은 security definition에서 요구하는 바와 같이 $\mathcal{A}$가 위조 과정에서 출력인 $m^*$과는 다른 signing 오라클에 쿼리했던 $x_m$을 생성하는데 사용된 모든 메시지 $m$에 대해 영지식증명을 할 수 있음을 의미한다. □

다음으로 ***definition5***를 이용하여 수정된 ElGamal 서명의 privacy 성질에 대해 논한다.

***Theorem 2.*** 수정된 ElGamal 서명 체계는 signature scheme with privacy이다.

***Proof*** $\mathcal{A}$가 signing 오라클에 접근이 가능한 수정된 ElGamal 서명 체계를 공격하는 PPT 공격자라 하자. $\mathcal{A}$가 챌린지로 $(m_0, m_1)$을 쿼리하고 임의의 랜덤넘버 $r$을 이용한 commitment $x_{m_b} = com(m_b, r)$과 이에 대한 ElGamal 서명 $\sigma_b$ 쌍인 $(\sigma_b, x_{m_b})$를 받았다고 하자. 이 서명 체계에서 공격자 $\mathcal{A}$가 관측할 수 있는 서명 $\sigma_b$와 commitment $x_{m_b}$는 $\mathcal{A}$가 여러 번 **PrivVerify** 알고리즘을 수행하더라도 변하지 않는 값이다. 따라서 **PrivVerify** 알고리즘을 수행하는 동안에 같은 $\sigma_b$와 $x_{m_b}$를 사용하고 ZKPK 과정만 다르다. 기본 구성 요소의 속성에 따라, ZKPK를 통해서는 정보를 얻을 수 없으며, commitment scheme은 정보 이론적(information-theoretically)으로 hiding을 달성한다는 것을 알고 있는데, 이것은 이산로그 문제의 난이도를 가정할 때 $\mathcal{A}$가 $m_b$이 무엇인지 결정하는 데 사용할 만한 어떠한 정보도 non-negligible한 확률로 가질 수 없다는 것을 의미한다. 다시 말해, 우리가 챌린지에서 $m_b$를 랜덤하게 선택한 메시지로 대체하면($\mathcal{A}$는 랜덤하게 추측하는 것 외에 다른 전략이 없다) **PirvVerify** 알고리즘은 실제 랜덤 챌린지와 구분할 수 없다.

$\mathcal{A}$가 signing 오라클에 접근 가능하다고 하였기 때문에 자신의 챌린지 $m_0$과 $m_1$에 대한 서명을 챌린지가 시작되

기 전에 얻을 수 있다. 이렇게 사전에 생성하여 얻은 $m_0, m_1$에 대한 서명은 $\mathcal{A}$가 관찰할 수 있는 것이 $m_b$에 대한 commitment뿐이고 이것은 정보 이론적으로 hiding을 달성하고 다른 메시지에 대해 매번 다른 랜덤넘버를 사용하기 때문에 $\mathcal{A}$가 챌린지에 대한 답을 도출하는 데 도움이 되지 않는다. □

**PrivVerify** 알고리즘에 포함된 ZKPK는 다음과 같은 과정을 거친다. Prover는 랜덤하게 $v_1, v_2 \in \mathbb{Z}$ 를 선택하여 $T = g^{v_1} h^{v_2}$를 계산하여 Verifier에게 전송한다. Verifier는 challenge $e$를 선택하여 Prover에게 전송하고, 그러면 Prover는 $r_1 = v_1 + em \bmod q$, $r_2 = v_2 + er \bmod q$를 계산하여 Verifier에게 보낸다. 만약 $g^{r_1} h^{r_2} = x_m^e T$ 가 성립하면 Verifier는 ZKPK를 받아들이고, Prover는 ZKPK에 성공한 것이다. 여기서 ZKPK를 위해 5번의 modulo exponentiation, 서명 검증을 위해 3번의 mod exp가 사용되어 총 8번의 mod exp가 필요하다.(만약 commitment $x_m$을 저장하지 않고 매번 계산한다면 또 다른 1번의 보통 mod exp와 1번의 짧은 mod exp가 필요하다.)

## 2. Batch Verification of Modified ElGamal Signature

본 장에서는 수정된 ElGamal 서명 체계를 이용하여 생성된 $n$개의 서명 검증을 위한 batch verification에 대해 알아본다. 본 연구에서는 같은 서명자에 의해 생성된 $n$개의 서명 검증을 목표로 하므로 같은 키를 사용하여 생성된 $n$개의 서명에 대한 Batch Verification 방법에 대해 알아본다. Batch verification 방법 중 small exponent test [10]를 사용하였으며, 여기에서 verifier는 보안 매개변수 $l_b$를 선택한다. 보안 매개변수는 만약 유효하지 않은 서명이 포함된 여러 개의 서명이 batch verification을 성공적으로 통과했을 때 유효하지 않은 서명이 포함될 확률이 최대 $2^{-l_b}$ 임을 뜻하며, $l_b = 60$ 또는 80으로 설정한다. 수정된 ElGamal 서명 체계의 batch verification을 시행하는 **Batch** 알고리즘은 다음과 같이 정의한다.

**Batch**: prover는 $n$개의 메시지 $m_i \in \mathbb{Z}_q$ $(i = 1, \cdots, n)$에 대해 랜덤넘버 $x_{\sigma_i} \in \mathbb{Z}_q$를 사용하여 생성한 commitment $x_{m_i} = com(m_i, x_{\sigma_i}) = g^{m_i} h^{x_{\sigma_i}}$와 이에 대한 서명 $\sigma_i = (t_i, s_i)$를 알고 있으며 prover와 verifier는 모두 공개키 $pk = (q, G, g, y, h)$를 알고 있다.

1. prover는 서명 $\sigma_i$와 commitment $x_{m_i}$를 verifier에게 보낸다.
2. prover와 verifier는 다음 영지식증명 $PK\{(\mu_1, \cdots, \mu_n, \gamma_1, \cdots, \gamma_n): x_{m_1} = g^{\mu_1} h^{\gamma_1} \wedge \cdots \wedge x_{m_n} = g^{\mu_n} h^{\mu_n}\}$을 시행한다. 만약 영지식증명에 실패한다면 verifier는 0을 출력하고 프로토콜을 중지한다.
3. verifier는 랜덤넘버 $\delta_1, \cdots, \delta_n \in \{0,1\}^{l_b}$를 선택하고 $u_1 = \sum_{i=1}^{n} H(x_{m_i} \| t_i)\delta_i$와 $u_2 = \sum_{i=1}^{n} t_i \delta_i$을 계산하여 $g^{u_1} = y^{u_2} \prod_{i=1}^{n} t^{s_i \delta_i}$가 성립하는지 확인한다. 만약 성공한다면 verifier는 1을 출력하고 그렇지 않을 경우 0을 출력한다.

*Theorem 3.* **Batch** 알고리즘은 수정된 ElGamal 서명 체계의 batch 검증알고리즘이다.

*Proof* 먼저 $\mathrm{PrivVerify}(pk, m_1, \sigma_1) = \cdots = \mathrm{PrivVerify}(pk, m_n, \sigma_n) = 1$이면 $\mathrm{Batch}(pk, (m_1, \sigma_1), \cdots, (m_n, \sigma_n)) = 1$을 의미함을 보이기 위해 $n$개의 서명이 각각 검증되었다고 가정하자. 즉, 모든 $i = 1, \cdots, n$에 대해 $g^{H(x_{m_i} \| t_i)} = y^{t_i} t_i^{s_i}$이다. 그러면

$$g^{u_i} = g^{\sum_{i=1}^{n} H(x_{m_i} \| t_i)\delta_i} = \prod_{i=1}^{n} \left(g^{H(x_{m_i} \| t_i)}\right)^{\delta_i}$$

$$= \prod_{i=1}^{n} \left(y^{t_i} t_i^{s_i}\right)^{\delta_i} = \prod_{i=1}^{n} y^{t_i \delta_i} \prod_{i=1}^{n} t_i^{s_i \delta_i}$$

$$= y^{\sum_{i=1}^{n} t_i \delta_i} \prod_{i=1}^{n} t_i^{s_i \delta_i} = y^{u_2} \prod_{i=1}^{n} t^{s_i \delta_i}$$

원하는 결과를 도출하였다. 반대 방향의 증명을 위해 **Batch** 알고리즘의 검증이 통과했다고 가정하자. 그러면 $k_i \in \mathbb{Z}_q$에 대해 $t_i = g^{k_i}$로 계산할 수 있다. 따라서

$$g^{\sum_{i=1}^{n} H(x_{m_i} \| t_i)\delta_i} = y^{\sum_{i=1}^{n} t_i \delta_i} \prod_{i=1}^{n} t_i^{s_i \delta_i} = g^{\sum_{i=1}^{n} x_i t_i \delta_i} \prod_{i=1}^{n} g^{k_i s_i \delta_i}$$

$$= g^{\sum_{i=1}^{n} x_i t_i \delta_i + \sum_{i=1}^{n} k_i s_i \delta_i}$$

가 되어

$$\sum_{i=1}^{n} H(x_{m_i}||t_i)\delta_i - \sum_{i=1}^{n} x_i t_i \delta_i - \sum_{i=1}^{n} k_i s_i \delta_i \equiv 0 \pmod{q}$$

임을 알 수 있다. $\beta_i = H(x_{m_i}||t_i) - x_i t_i - k_i s_i$라 하자. 그러면 위의 식은 $\sum_{i=1}^{n} \delta_i \beta_i \equiv 0 \pmod{q}$라 쓸 수 있다. 이제 **Batch** 알고리즘 과정 중 **PrivVerify**에 입력되는 $(pk, m_i, \sigma_i)$에 대해 적어도 하나의 입력에서 0을 반환받았는데 **Batch** 알고리즘의 출력은 1이라 가정하자. **PrivVerify**에서 0을 반환받은 입력을 $i = 1$라 하자. 이것은 $g^{H(x_{m_1}||t_1)} \neq y^{t_1} t_1^{s_1}$을 의미하고, 따라서 $\beta_1 = H(x_{m_1}||t_1) - x_1 t_1 - k_1 s_1 \neq 0$이다. $G$는 소수 order $q$를 갖는 순환군이기 때문에 $\beta_1$은 역수 $\alpha_1$을 가지며 $\beta_1 \alpha_1 \equiv 1 \pmod{q}$이다. 따라서 위의 식을 다시 쓰면, $\delta_1 \beta + \sum_{i=2}^{n} \delta_i \beta_i \equiv 0 \pmod{q}$로 쓸 수 있으며, $\beta_1$ 대신 $\alpha_1^{-1}$을 사용하면 $\delta_1 \alpha_1^{-1} + \sum_{i=2}^{n} \delta_i \beta_i \equiv 0 \pmod{q}$이 된다. 이것을 통해 $\delta_1 \equiv -\alpha_1 \sum_{i=2}^{n} \delta_i \beta_i \pmod{q}$임을 알 수 있다.

$E$를 $\mathrm{PrivVerify}(pk, m_1, \sigma_1) = 0$이지만 $\mathrm{Batch}(pk, (m_1, \sigma_1), \cdots, (m_n, \sigma_n)) = 1$인 사건, 벡터 $\Delta = (\delta_2, \cdots, \delta_n)$라하고 $|\Delta|$를 $\Delta$가 가질 수 있는 값의 개수라 하자. $\Delta$이 고정되어 있다면 위의 식에 의해 오직 하나의 $\delta_1$이 존재함을 알 수 있으며, 그것이 사건 $E$가 일어나는 경우이다. 즉, $\Delta$가 고정되어 있을 때 랜덤하게 선택된 $\delta_1$에 대해 사건 $E$가 일어날 확률은 $\Pr[E|\Delta'] = 2^{-l_b}$이다. 따라서 랜덤하게 선택한 $\delta_1$에 대해 사건 $E$가 일어날 확률은 모든 가능한 $\Delta$의 경우를 모두 더하는 것에 바운드된다. 다시 말하면, $\Pr[E] \leq \sum_{i=1}^{|\Delta|} (\Pr[E|\Delta] \cdot \Pr|\Delta|)$이다. 따라서

$$\Pr[E] \leq \sum_{i=1}^{2^{l_b(n-1)}} \left(2^{-l_b} \cdot 2^{-l_b(n-1)}\right) = \sum_{i=1}^{2^{l_b(n-1)}} \left(2^{-l_b n}\right) = 2^{-l_b}$$

이다. 그러므로 유효하지 않은 서명이 포함된 여러 개의 서명이 batch verification을 성공적으로 통과하려면 유효하지 않은 서명이 포함될 최대 확률은 $2^{-l_b}$ 이하여야 한다.

### 3. Comparison of computation cost of protocol

계산에 입력하는 입력의 인증 프로토콜의 비용 분석을 위해서는 서명을 인증하는데 소요되는 계산량의 정확한 분석이 필요하다. 앞서 살펴본 수정된 ElGamal 서명 체계와 batch verification에 필요한 계산량을 분석하면 다음과 같다.

수정된 ElGamal 서명 체계를 이용하여 한 개의 서명을 인증하려면 ZKPK에 5번의 mod exp와 $g^{H(x_m||t)} = y^t t^s$ 확인에 3번의 mod exp 연산이 필요하다. 따라서 수정된 ElGamal 서명체계의 **PrifVerify**에는 총 8번의 mod exp 연산이 필요하다.

**Batch** 알고리즘의 영지식증명은 ElGamal 서명의 **PrivVerify**의 영지식증명을 $n$번 수행하는 것이다. 따라서 $n$개의 메시지에 대한 batch verification은 영지식증명에 $5n$번의 mod exp와 서명 검증에 $n+2$번의 mod exp가 필요하여 총 $6n+2$번의 mod exp가 필요하다. 만약 commitment를 저장하지 않고 다시 계산한다면 $n$번의 보통 mod exp와 또 다른 $n$번의 짧은 mod exp가 필요하다.

본 연구에서 고려하는 상황은 서명자가 여러 개의 메시지에 서명하여 계산에 사용하는 것이기 때문에 각 메시지마다 별도의 commitment를 생성하는 것이 아닌, 여러 개의 메시지를 이용하여 한 개의 commitment를 생성하는 방법을 사용할 수 있다. 수정된 ElGamal 서명 체계를 사용하는 경우 이 방법은 여러 개의 메시지에 단 하나의 서명만을 생성하는 것이기 때문에 검증의 효율성을 획기적으로 향상시킬 수 있다. 즉, 서명자는 모든 메시지에 대한 하나의 commitment $x_m = com(m_1, \cdots, m_n, r)$를 계산하고, 이에 대해 서명 하나를 생성하여서 검증하면 된다. 이것은 **Batch** 알고리즘의 1단계에서 1개의 서명과 commitment만 생성하기 때문에 communication 횟수를 획기적으로 줄일 수 있고, 2단계에서는 $x_m$ 하나에 대한 영지식증명만 하면 되고, 3단계에서는 $\sigma_i$들의 사용 없이 하나의 서명 검증만 하면 되기 때문에 계산량이 확연히 줄어든다. 이것을 통해 2단계에서는 필요한 계산이 $2n+3$ mod exp로 감소하고 **Batch**의 전체 계산은 $2n+6$ mod exp로 감소하여 성능 향상에 큰 영향을 미친다(만약 commitment를 저장하지 않는 경우 1번의 보통 mod exp와 $n$번의 짧은 mod exp 계산이 추가로 필요하다).

본 연구에서 제안한 수정된 ElGamal 서명 체계를 이용하고 commitment가 사전 계산되어 저장되어 있을 때 싱글 메시지에 서명할 경우, $n$개의 메시지에 $n$개의 commitment를 생성하여 batch verification을 사용할

경우, $n$개의 메시지에 하나의 commitment를 생성할 경우의 인증에 필요한 계산량은 다음 표와 같다.

Table 1. Performance of private verification for a single signature and a batch of size $n$.

| case | Modified ElGamal |
|---|---|
| single signature | 8 mod exp |
| Batch with $n$ commitments | $6n+2$ mod exp |
| Batch with 1 commitment | $2n+6$ mod exp |

인증 단계에서 각 메시지별로 commitment를 생성하는 방식에 비해 여러 메시지를 이용하여 하나의 commitment를 생성하는 것이 연산량을 감소시키는 데 큰 영향을 미친다는 것을 알 수 있다.

## IV. Conclusions

본 연구에서는 여러 명의 참여자가 참여하는 계산의 안전성을 위협하는 상황 중 전통적인 보안 모델을 벗어나는 경우인 악의적인 입력 조작을 방지하기 위한 입력 인증의 한 방법으로 수정된 ElGamal 서명 체계를 사용하여 batch verification을 사용하는 방법에 대해 알아보았다. 수정된 ElGamal 서명 체계와 commitment scheme을 사용하여 입력 인증을 강제할 수 있고, batch verification과 단일 commitment 생성 방법을 이용하여 인증에 필요한 계산량을 획기적으로 낮출 수 있음을 확인하였다. 이 방법을 통해 같은 키를 사용하여 대량의 메시지를 입력하는 사용자의 경우에도 입력 검증을 효율적으로 실시할 수 있다. 차후 연구에서는 계산 간에 전수조사로 인해 안전성 확보가 어려운 작은 메시지 공간에서 입력 인증이 필요할 때 안전성을 유지하면서 인증의 효율성을 향상시킬 수 있는 입력 인증 방법에 관해 연구를 확장하여 실생활에 접목하는 것을 목표로 한다.

## ACKNOWLEDGEMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(2020R1G1A1A01100862).



## REFERENCES

[1] J. Halpern, V. Teague, "Rational secret sharing and multiparty computation," in Proceedings of the thirty-sixth annual ACM symposium on Theory of computing, 2004., https://doi.org/10.1145/1007352.1007447

[2] J. Wallrabenstein, C. Clifton, "Equilibrium concepts for rational multiparty computation," in Decision and Game Theory for Security: 4th International Conference, TX, 2013., https://doi.org/10.1007/978-3-319-02786-9_14

[3] M. Blanton, F. Bayatbabolghani, "Efficient server-aided secure two-party function evaluation with applications to genomic computation," in Proceedings on Privacy Enhancing Technologies, 2016., https://doi.org/10.1515/popets-2016-0033

[4] M. Jeong, "Efficient and Secure Signature Scheme applicable to Secure multi-party Computation," in Journal of The Korea Society of Computer and Information, 2023., https://doi.org/10.9708/jksci.2023.28.07.077

[5] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," in IEEE Transactions on Information Theory, 1985., 10.1109/TIT.1985.1057074

[6] D. Pointcheval, J. Stern, "Security proofs for signature schemes," in International conference on the theory and applications of cryptographic techniques, 1996, https://doi.org/10.1007/3-540-68339-9_33

[7] D. Pointcheval, J. Stern, "Security arguments for digital signatures and blind," in Journal of cryptology, 2000., https://doi.org/10.1007/s001450010003

[8] T. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in Annual international cryptology conference, 1991., https://doi.org/10.1007/3-540-46766-1_9

[9] J. Camenisch, M. Stadler, "Proof systems for general statements about discrete logarithms," in Technical Report/ETH Zurich, Department of Computer Science, 1997., https://doi.org/10.3929/ETHZ-A-006651937

[10] M. Bellare, J. Garay, T. Rabin, "Fast batch verification for modular exponentiation and digital signatures," in Advances in Cryptology—EUROCRYPT'98: International Conference on the Theory and Application of Cryptographic Techniques Espoo, 1998., https://doi.org/10.1007/BFb0054130

[11] J. Camenisch, S. Hohenberger, "Batch verification of short signatures," in Advances in Cryptology-EUROCRYPT 2007: 26th Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2007., https://doi.org/10.1007/978-3-540-72540-4_14

[12] J. Camenisch, A. Lysyanskaya, "A signature scheme with efficient protocols," in Security in Communication Networks: Third International Conference, SCN 2002 Amalfi, 2003., https://doi.org/10.1007/3-540-36413-7_20

[13] J. Camenisch, A. Lysyanskaya, "Signature schemes and anonymous credentials," in Annual international cryptology conference, 2004., https://doi.org/10.1007/978-3-540-28628-8_4

## Authors

Myoungin Jeong received the B.S. degree in Dept. of Mathematics from Korea Military Academy, Korea, in 2004. She received M.S. degree in Dept. of Mathematical Science from Seoul National University, Korea, in 2008. And received Ph.D. degree in Dept. of Mathematics from University at Buffalo, United States, in 2018. Dr. Jeong joined the faculty of the Department of Mathematics at Korea Military Academy, Seoul, Korea, in 2018. She is currently an Assistant Professor in the Department of Mathematics, Korea Military Academy. She is interested in information security, cryptography, and scientific combat training system.