情報保護學會論文誌
第 23 卷 第 1 號, 2013. 2

래티스에서 ID 기반의 강한 지정된 검증자 서명 기법*

노 건 태,† 천 지 영, 정 익 래‡
고려대학교

Identity-based Strong Designated Verifier Signature Scheme from Lattices*

Geontae Noh,† Ji Young Chun, Ik Rae Jeong‡
Korea University

요 약

강한 지정된 검증자 서명은 서명자가 검증자를 지정하여 서명을 생성하며, 이렇게 생성된 서명은 지정된 검증자만이 서명자로부터 생성되었는지를 확신할 수 있다. 추가적으로, 지정된 검증자 이외의 어떤 누구도 이렇게 생성된 서명이 어떤 서명자에 의해 생성된 서명인지를 알아낼 수 없다는 특징을 가진다. 본 논문에서 우리는 사용자의 공개키가 ID라는 장점을 가지는 ID 기반의 강한 지정된 검증자 서명 기법을 제안한다. 우리가 제안하는 기법은 ID 기반의 강한 지정된 검증자 서명 분야에서 최초로 래티스에서 설계되었으며, 따라서 양자 컴퓨팅 환경에서도 안전하며 높은 연산 효율성을 가진다.

ABSTRACT

When a signer signs a message, strong designated verifier signature allows the signer to designate a verifier. Only the designated verifier can make sure that the signature is generated by the signer. In addition, no one except the designated verifier can know the signature generated by some signer. In this paper, we propose an identity-based strong designated verifier signature scheme where users' public keys are identities. Our proposed scheme is the first identity-based strong designated verifier scheme from lattices. Naturally, our proposed scheme is secure against quantum computing attacks and has low computational complexity.

Keywords: Lattice-based cryptography, Identity-based strong designated verifier signature

I. 서 론

지정된 검증자 서명은 1996년, M. Jakobsson 등에 의해 처음으로 소개되었다[1]. 지정된 검증자 서명은 서명자가 검증자를 지정하여 서명을 생성하며, 이렇게 생성된 서명은 지정된 검증자만이 서명자로부터 생성되었는지를 확신할 수 있다. 서명자와 지정된 검증자 이외의 어떤 누구도 이렇게 생성된 서명이 서명자에 의해 생성되었는지를 확신할 수 없으며, 단지 서명자와 지정된 검증자 중의 1명이 생성했다는 것만을 확인할 수 있다. 이러한 성질은 지정된 검증자 서명이 기본적으로 만족해야 하는 비전이성(non-transferability), 즉, 서명자뿐만 아니라 지정된 검증자도 서명자가 생성한 서명과 동일한 형태의 시뮬레이션된 서명을 생성할 수 있다는 것에 기인한다[1-3].

하지만 지정된 검증자 서명은 서명자와 지정된 검증자 사이의 익명성만을 제공한다. 다시 말하면, 서명자가 서명을 생성해서 지정된 검증자에게 보내는 도중

접수일(2013년 1월 4일), 수정일(1차: 2013년 1월 22일, 2차: 2013년 1월 29일), 게재확정일(2013년 1월 29일)
* 이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임 (2012R1A1A3005550)
† 주저자, oldos@korea.ac.kr
‡ 교신저자, irjeong@korea.ac.kr

에 악의적인 사용자가 이것을 가로챈다면, 그리고 악의적인 사용자가 이 서명이 지정된 검증자로부터 생성되지 않았다는 확신을 가지게 된다면, 악의적인 사용자는 이 서명이 서명자로부터 생성되었다는 것을 확신할 수 있게 된다. 따라서 지정된 검증자 서명에 추가적인 안전성을 제공하는 강한 지정된 검증자 서명이 연구되었다. 강한 지정된 검증자 서명은 지정된 검증자 서명과 마찬가지로 서명자가 검증자를 지정하여 서명을 생성하며, 이렇게 생성된 서명은 지정된 검증자만이 서명자로부터 생성되었는지를 확신할 수 있다. 추가적으로, 강한 지정된 검증자 서명으로부터 생성된 서명은 지정된 검증자 이외의 어떤 누구도 이 서명이 어떤 서명자에 의해 생성된 서명인지를 알아낼 수 없어야 한다는 서명자의 프라이버시(privacy of signer's identity)를 만족해야 한다. 최근에는 지정된 검증자 서명에 대한 연구보다는 강한 지정된 검증자 서명에 대한 연구가 보다 활발히 진행되고 있다[1,4-8].

공개키 기반의 강한 지정된 검증자 서명 기법이 S. Saeednia 등에 의해 제안된 후[4], 공개키 기반의 기법에서 공개키 분배 문제를 해결한 ID 기반의 강한 지정된 검증자 서명 기법을 설계하는 일반적인 방법(generic construction)이 W. Susilo 등에 의해 제안되었다[9]. ID 기반의 암호시스템에 대한 개념은 A. Shamir에 의해 처음으로 정립되었으며[10], 공개키 기반의 기법에서 발생하는 키 관리 문제를 해결함과 동시에 이메일 주소나 전화번호 등과 같은 정보를 공개키로 그대로 사용할 수 있다는 장점을 가진다.

2012년, F. Wang 등은 래티스 기반의 강한 지정된 검증자 서명 기법을 제안하였다[8]. 래티스에서 설계된 암호 및 서명 기법은 양자 컴퓨팅 환경에서도 높은 안전성을 가지며, 행렬 및 벡터들의 선형 연산들로만 구성되기 때문에 연산 효율성도 매우 높다는 장점을 가진다. 하지만 이것은 공개키에 기반을 둔 기법으로, ID 기반의 기법들이 가지는 다양한 장점들을 포함하지 못한다.

본 논문에서 우리는 래티스에서 ID 기반의 강한 지정된 검증자 서명 기법을 제안한다. 제안하는 기법은 최초로 래티스에서 설계되었으며, 래티스에서 설계된 기법이 가지는 장점들과 ID 기반의 기법들이 가지는 장점들을 모두 포함하는 최초의 기법이다. 즉, 사용자의 공개키로 임의의 난수가 아닌 자신의 이메일 주소나 전화 번호 등의 ID를 사용할 수 있고, 행렬에서의 선형 연산들로만 구성되기 때문에 겹선형 맵에서 설계된 기법들과는 달리 연산 효율성이 매우 높다. 또한 제안하는 기법은 래티스에서의 어려운 문제들인 SIS(Short Integer Solution) 문제와 LWE(Learning With Errors) 문제의 어려움에 안전성의 기반을 두고 설계되었기 때문에 양자 컴퓨팅 환경에서도 안전한 기법이다.

ID 기반의 강한 지정된 검증자 서명 기법을 설계하는 일반적인 방법이 W. Susilo 등에 의해 이미 제안되었음에도 불구하고 제안하는 기법이 의미가 있는 이유는 일반적인 설계 방법에서 반드시 필요한 ID 기반의 카멜레온 해시 함수가 래티스에서는 아직 알려지지 않았기 때문이다. 지금까지 제안된 ID 기반의 카멜레온 해시 함수는 겹선형 맵에서 설계되거나[11], RSA 서명 기법에 기반을 두고 설계되었다[12]. 비록 래티스에서 공개키 기반의 카멜레온 해시 함수는 존재하지만[13], 래티스에서 ID 기반의 카멜레온 해시 함수는 아직까지 존재하지 않는다. 따라서 W. Susilo 등이 제안한 일반적인 방법을 사용하여 래티스에서 강한 지정된 검증자 서명 기법을 설계하는 것은 아직까지는 불가능하다. 따라서 본 논문에서 제안하는 기법은 일반적인 방법을 따라 제안하는 기법이 아니며, 향후 래티스에서 ID 기반의 카멜레온 해시 함수가 제안되어진다 하더라도 일반적인 방법으로 설계하기 위해서는 반드시 ID 기반의 카멜레온 해시 함수와 ID 기반의 암호화 기법 등이 필요하며, 본 논문에서 제안하는 기법은 이러한 것들을 전혀 사용하지 않기 때문에 보다 효율적이다.

1.1 응용 환경

제안하는 강한 지정된 검증자 서명은 광범위한 분야에 적용될 수 있으며, 가격 경쟁 입찰, 전자 투표, 전자 경매 등에 유용하게 사용될 수 있다. 어떤 기관에서 물품 구매를 위한 가격 경쟁 입찰 시 지정된 검증자 서명 기법을 사용한다면, 입찰 참가자들은 다른 경쟁사들이 제시하는 금액을 볼 수 있을 것이다. 만약 지정된 검증자 서명 기법이 아닌 일반적인 서명 기법을 사용한다면, 다른 경쟁사가 제안한 금액보다 낮은 금액을 제시함으로써 낙찰 받으려고 할 것이기 때문에 이러한 상황에서는 지정된 검증자 서명 기법이 적합하다. 지정된 검증자 서명 기법을 사용하면 그 금액이 다른 경쟁사가 제시한 금액인지 아니면 낙찰가를 낮추기 위해 전략적으로 물품 구매 기관에서 제시한 금액인지에 대한 판단이 불가능하기 때문에 물품 구매 기

관 입장에서는 적정 가격에 낙찰되도록 하기 위한 적절한 시도가 가능하다.

하지만 강한 지정된 검증자 서명 기법이 아니라 지정된 검증자 서명 기법을 사용하였을 경우에는 서명자가 생성한 서명을 지정된 검증자가 받기 전에 가로채고, 지정된 검증자가 서명을 시뮬레이션하지 않는다는 확신을 가지게 된다면, 누구나 어떤 서명자가 이 서명을 생성했는지에 대한 정보를 얻을 수 있다는 문제점이 존재한다. 따라서 지정된 검증자 서명 기법을 사용하는 것보다 강한 지정된 검증자 서명 기법을 사용하는 것이 서명자의 프라이버시 측면에서 볼 때 보다 안전하다.

또한, 위와 같은 경우에 ID 기반의 강한 지정된 검증자 서명 기법을 사용한다면 공개키 기반의 기법에서 발생하는 키 관리 문제를 해결함과 동시에 이메일 주소나 전화번호 등과 같은 정보를 사용자의 공개키로 사용한다는 장점을 가지게 된다.

1.2 접근

2012년, F. Wang 등에 의해 래티스에서 공개키 기반으로 설계된 강한 지정된 검증자 서명 기법[8]과는 달리 ID 기반으로 설계하기 위해서는 ID를 기반으로 한 키 추출 기법에 대한 추가적인 설계가 필요하다. 즉, KGC(Key Generation Center)의 마스터 비밀키로부터 사용자의 비밀키를 위임(delegation)할 수 있도록 기법을 구성해야 한다. 따라서 KGC의 마스터 비밀키로부터 사용자의 비밀키를 위임할 수 있도록 하기 위해 우리는 2010년, D. Cash 등이 설계한 트랩도어 확장 알고리즘을 활용한다[13]. 트랩도어 확장 알고리즘을 사용하면 KGC의 마스터 비밀키를 기저(basis)로 가지는 래티스의 행렬과 사용자의 ID 정보를 접합시켜 확장된 래티스의 기저를 생성할 수 있으며, 이를 사용자의 비밀키로 활용할 수 있게 된다.

제안하는 기법에서는 ID 기반의 강한 지정된 검증자 서명 기법의 설계를 위해 먼저 KGC가 트랩도어 생성 알고리즘을 수행하여 래티스를 생성하는 행렬과 이것의 트랩도어를 생성한다. 래티스를 생성하는 행렬은 임의의 길이를 가지는 문자열을 행렬 형태로 출력해주는 해시 함수와 함께 공개 파라미터로 두고, 래티스의 트랩도어를 마스터 비밀키로 둔다. KGC는 사용자의 개인정보, 즉 ID에 대한 비밀키를 위임하기 위해서 공개 파라미터에 포함된 래티스를 생성하는 행렬과 ID를 해시한 값을 접합하고, 이렇게 접합된 행렬을 통해서 생성된 기저를 사용자(즉, 서명자와 지정된 검증자)의 비밀키로 위임한다. 비밀키를 알고 있는 정당한 서명자는 자신의 ID, 지정된 검증자의 ID, 그리고 난수를 포함하는 래티스를 구성할 수 있으며, 비밀키를 활용하여 래티스에서의 짧은 벡터를 샘플링하고, 난수는 LWE 문제의 어려움에 기반을 두어 지정된 검증자만이 난수 정보를 얻어낼 수 있도록 구성한다. 이렇게 생성된 서명은 난수를 얻어낼 수 있는 지정된 검증자만이 자신의 ID, 서명자의 ID, 그리고 난수를 포함하는 래티스를 구성할 수 있으며, 따라서 지정된 검증자만이 서명에 대한 검증이 가능하게 된다.

1.3 구성

본 논문의 구성은 다음과 같다. 2장에서는 본 논문과 관련된 배경 지식에 대해서 살펴본다. 3장에서 우리는 래티스에서 ID 기반의 강한 지정된 검증자 서명 기법을 제안하고, 4장에서 이를 분석한다. 마지막으로 5장에서 결론을 맺는다.

II. 배경 지식

2.1 표기법

본 논문에서 사용할 표기법은 다음과 같다: n은 시큐리티 파라미터이다. 행렬 A의 그램-슈미트 직교화(Gram-Schmidt orthogonalization)는 $\tilde{A}$로 표시한다. 두 벡터 a와 b, 또는 두 행렬 A와 B의 접합은 $a\|b$, 또는 $A\|B$로 표기한다. 벡터 a의 유클리디안 거리는 $\|a\|$로 표시하고, 행렬 A의 유클리디안 거리는 $\|A\|$로 표시한다. 빅오 표기법(big-O notation)을 사용한다. $\beta = poly(n)$은 어떤 양의 정수 c에 대해 $\beta \in \Theta(n^c)$를 의미한다. 충분히 큰 n과 어떤 $c > 0$에 대해 $|negl(n)| < 1/n^c$면, $negl(n)$는 무시할만하고(negligible), $1 - negl(n)$은 극단적이다(overwhelming). 모듈러 연산 $x \pmod q$는 정수 x를 $(-q/2, q/2]$로 매핑한다.

2.2 래티스

본 논문에서 우리는 m차원의 풀-랭크(full-rank) 정수 래티스 $\Lambda \subseteq Z^m$를 사용하며, 이것은 다음과 같이 m개의 선형 독립인 기저 벡터 $B = \{b_1, \cdots b_m\} \subset Z^m$의

선형 결합으로 정의된다:

$$\Lambda=\left\{Bc=\sum_{i\in\{1,\cdots,m\}}c_i b_i : c\in Z^m\right\} \quad (1)$$

본 논문에서 우리는 m차원 풀-랭크 정수 래티스의 특정한 형태를 사용한다. 이것은 패리티 검사 행렬(parity check matrix) $A\in Z_q^{n\times m}$로부터 생성되며, 다음과 같이 정의된다:

$$\Lambda^{\perp}(A)=\left\{Ax=\sum_{i\in\{1,\cdots,m\}}x_i a_i=0\in Z_q^n : x\in Z^m\right\} \quad (2)$$

여기서 $n\geq 1$과 $q\geq 2$는 정수이다. 그리고 $m=O(n\log q)$에 대해 균일하게 랜덤한 행렬 $A\in Z_q^{n\times m}$의 열벡터들은 선형 결합을 통해 극단적인 확률로 Z_q^n를 모두 표현할 수 있다[14].

2.2.1 가우시안 분포

차원 $m\geq 1$과 어떤 $s>0$에 대해서 가우시안 함수(Gaussian function) $\rho_s : R^m\rightarrow(0,1]$는 다음과 같이 정의된다:

$$\rho_s(x)=\exp(-\pi\|x\|^2/s^2) \quad (3)$$

어떤 래티스 Λ에 대해 이산 가우시안 분포(descrete Gaussian distribution) $D_{\Lambda,s}$는 각각의 $x\in\Lambda$에서 $\rho_s(x)$에 비례하는 확률을 가진다[13,15-17].

어떤 래티스 Λ와 양의 실수 ϵ에 대해, 가우시안 파라미터(Gaussian parameter) $\eta_\epsilon(\Lambda)$는 $\rho_{1/s}(\Lambda^*\setminus\{0\})\leq\epsilon$를 만족하는 가장 작은 s이다. 여기서 Λ^*는 Λ의 듀얼 래티스이다[16].

래티스에서의 가우시안 분포는 다음과 같은 성질을 가진다:

$$\Pr\left[\|x\|>s\sqrt{m} : x\leftarrow D_{\Lambda^{\perp}(A),s}\right]\leq negl(n) \quad (4)$$

$$\Pr\left[x=0 : x\leftarrow D_{\Lambda^{\perp}(A),s}\right]\leq negl(n) \quad (5)$$

여기서 $T\in Z^{m\times m}$는 $\Lambda^{\perp}(A)$의 기저이며, $A\in Z_q^{n\times m}$이고, $s\geq\|\tilde{T}\|\cdot\omega(\sqrt{\log n})$이다.

2.2.2 SIS 문제

본 항에서 우리는 SIS 문제를 살펴본다. 이것은 평균적인 경우에 어려운 문제(average-case hardness problem)로 알려져 있으며, M. Ajtai에 의해 최악의 경우에 어려운 문제(worst-case hardness problem)로 리덕션(reduction)되었다[18,19].

$SIS_{n,m,q,\beta}$ 문제는 어떤 $m=poly(n)$과 $q\geq 2$에 대해 균일하게 랜덤한 행렬 $A\in Z_q^{n\times m}$를 입력으로 받아서 $\|v\|\leq\beta$를 만족하는 래티스 $\Lambda^{\perp}(A)$의 0이 아닌 벡터 $v\in Z^m$를 찾는 것이다. 즉, $Av=0\in Z_q^n$이다.

2.2.3 LWE 문제

본 항에서 우리는 LWE 문제를 살펴본다. 이것은 평균적인 경우에 어려운 문제로 알려져 있으며, O. Regev에 의해 최악의 경우에 어려운 문제로 리덕션 되었다[14].

$LWE_{n,m,q,\alpha}$ 문제는 어떤 $m=poly(n)$과 $q\geq 2$에 대해 $(A_0, r_0=A_0^T s_0+x_0)$의 분포와 (A_1, r_1)의 분포를 구별하는 문제이다. 여기서 $A_0\in Z_q^{n\times m}$와 $A_1\in Z_q^{n\times m}$은 균일하게 랜덤한 행렬이고, $s_0\in Z_q^n$와 $r_1\in Z_q^m$은 균일하게 랜덤한 벡터이며, $x_0\in Z_q^m$는 가우시안 에러 분포 $\Psi_{q,\alpha}^m$로부터 샘플링된 벡터이다.

2.2.4 기본 알고리즘

본 논문에서 우리는 트랩도어 생성 알고리즘 $TrapGen(1^n,1^m,q)$, 가우시안 샘플링 알고리즘 $SampleD(A,T_A,y,s)$, 트랩도어 확장 알고리즘 $ExtBasis(T_A,A\|A')$을 사용한다. 따라서 우리는 본 항에서 이러한 기본 알고리즘들을 살펴본다.

- 트랩도어 생성 알고리즘: 트랩도어 생성 알고리즘 $TrapGen(1^n,1^m,q)$은 시큐리티 파라미터 n, 래티스의 차원 $m=O(n\log q)$, 양의 정수 q를 입력으로 받아서 균일하게 랜덤한 행렬 $A\in Z_q^{n\times m}$와 래티스 $\Lambda^{\perp}(A)$의 기저 $T_A\in Z^{m\times m}$를 생성한다. 여기서 T_A는 $\|\widetilde{T_A}\|=O(\sqrt{n\log q})$와 $AT_A=0\,(\mathrm{mod}\,q)$를 만족하며, 역행렬이 존재한다[20].
- 가우시안 샘플링 알고리즘: 가우시안 샘플링 알고리즘 $SampleD(A,T_A,y,s)$는 균일하게 랜덤한 행렬 $A\in Z_q^{n\times m}$, 래티스 $\Lambda^{\perp}(A)$의 기저 $T_A\in Z^{m\times m}$, 신드롬(syndrome) $y\in Z_q^n$, 가우시안 파라미터 s를 입력으로 받아서 벡터 $v\in Z^m$를 가우시안 샘플링한다. 여기서 v는

$Av = y \pmod q$와 $\|v\| \le s\sqrt{m}$를 만족한다. 추가적으로, 가우시안 샘플링 알고리즘 $SampleD(A, T_A, y, s)$의 정의역을 가우시안 분포로부터 샘플링할 수 있는 $SampleDom$ 알고리즘도 존재한다. $SampleDom$ 알고리즘으로부터 샘플링된 벡터 $x \in Z^m$는 $\|x\| \le s\sqrt{m}$를 만족한다[17].

- 트랩도어 확장 알고리즘: 트랩도어 확장 알고리즘 $ExtBasis(T_A, A\|A')$는 균일하게 랜덤한 행렬 $A \in Z_q^{n \times m}$, 래티스 $\Lambda^{\perp}(A)$의 기저 $T_A \in Z^{m \times m}$, 균일하게 랜덤한 행렬 $A' \in Z_q^{n \times m'}$을 입력으로 받아서 래티스 $\Lambda^{\perp}(A\|A')$의 기저 $T_{A\|A'} \in Z^{(m+m') \times (m+m')}$를 생성한다. 여기서 $T_{A\|A'}$는 $\|\widetilde{T_A}\| = \|\widetilde{T_{A\|A'}}\|$를 만족하며, 역행렬이 존재하고, A'의 위치가 반드시 A의 뒤에 올 필요는 없다[13].

2.3 ID 기반의 강한 지정된 검증자 서명

먼저, 서명자를 S라 하고, 지정된 검증자를 V라 하자. 그러면 ID 기반의 강한 지정된 검증자 서명은 다음의 5가지 알고리즘으로 구성된다:

- $Setup$: KGC는 공개 파라미터 $params$와 마스터 비밀키 msk를 생성한다.
- $Extract(msk, ID_S, ID_V)$: KGC는 마스터 비밀키 msk, 서명자 S의 ID ID_S, 지정된 검증자 V의 ID ID_V를 입력으로 받아서 서명자 S의 비밀키 sk_S와 지정된 검증자 V의 비밀키 sk_V를 생성한다.
- $Sign(ID_S, sk_S, ID_V, params, b)$: 서명자 S는 자신의 ID ID_S, 자신의 비밀키 sk_S, 지정된 검증자 V의 ID ID_V, 공개 파라미터 $params$, 그리고 서명할 메시지 b를 입력으로 받아서 서명 σ를 생성한다.
- $Vrfy(ID_V, sk_V, ID_S, params, b, \sigma)$: 지정된 검증자 V는 자신의 ID ID_V, 자신의 비밀키 sk_V, 서명자의 ID ID_S, 공개 파라미터 $params$, 서명된 메시지 b, 그리고 서명 σ를 입력으로 받아서 서명을 검증한다.
- $Simul(ID_V, sk_V, ID_S, params, b)$: 지정된 검증자 V는 자신의 ID ID_V, 자신의 비밀키 sk_V, 서명자 S의 ID ID_S, 공개 파라미터 $params$, 그리고 서명할 메시지 b를 입력으로 받아서 시뮬레이션된 서명 σ를 생성한다.

ID 기반의 강한 지정된 검증자 서명은 다음의 4가지 성질을 만족해야 한다:

- 정확성(correctness): 서명자 S가 $Sign(ID_S, sk_S, ID_V, params, b)$ 알고리즘을 사용하여 정당하게 서명 σ를 생성하였다면, 이 서명 σ는 $Vrfy(ID_V, sk_V, ID_S, params, b, \sigma)$ 알고리즘으로 검증되어야 한다.
- 위조 불가능성(unforgeability): 서명자 S와 지정된 검증자 V를 제외한 어느 누구도 정당한 서명 σ를 생성하는 것이 불가능해야 한다.
- 비전이성(non-transferability): 서명자 S가 $Sign(ID_S, sk_S, ID_V, params, b)$ 알고리즘을 사용하여 생성한 서명 σ와 지정된 검증자 V가 $Simul(ID_V, sk_V, ID_S, params, b)$ 알고리즘을 사용하여 생성한 시뮬레이션된 서명 σ가 서로 구별 불가능해야 한다.
- 서명자의 프라이버시(pravacy of signer's identity): 서명자 S와 지정된 검증자 V 사이의 서명 σ에 대해, 지정된 검증자 V를 제외한 어떤 누구도 어떤 서명자에 의해 생성된 서명인지를 알아낼 수 없어야 한다.

위의 4가지 성질 중에서 위조 불가능성은 챌린저와 공격자 사이의 게임으로 나타낼 수 있다. 먼저, $Setup$ 단계에서 챌린저는 공격자에게 공개 파라미터 $params$를 제공한다. 그러면 공격자는 다음의 3가지 오라클에 대한 접근이 가능하다.

- $Extract(msk, \bullet)$: 공격자는 자신이 원하는 ID ID_i에 대한 비밀키 sk_i를 얻을 수 있다.
- $Sign(\bullet, sk_{S_i}, \bullet, params, \bullet)$: 공격자는 자신이 원하는 서명자의 ID ID_{S_i}, 자신이 원하는 지정된 검증자의 ID ID_{V_i}, 자신이 서명하고자 하는 메시지 b_i에 대한 서명 σ_i를 얻을 수 있다.
- $Vrfy(\bullet, sk_{V_i}, \bullet, params, \bullet, \bullet)$: 공격자는 자신이 원하는 서명자의 ID ID_{S_i}, 자신이 원하는 지정된 검증자의 ID ID_{V_i}, 자신이 원하는 메시지 b_i, 자신이 원하는 서명 σ_i에 대해 검증해볼 수 있다.

마지막으로, 공격자는 $(ID_S^*, ID_V^*, b^*, \sigma^*)$를 챌린저에게 보낸다. $Vrfy(ID_V^*, sk_V^*, ID_S^*, params, b^*, \sigma^*)$ 알고리즘을 통해 검증이 되고, 공격자가 ID_S^*와 ID_V^*를

$Extract(msk, \bullet)$ 오라클에 질의하지 않았으며, (ID_S^*, ID_V^*, m^*)를 $Sign(\bullet, sk_i, \bullet, params, \bullet)$ 오라클에 질의하지 않았다면, 공격자는 위조 불가능성 게임에서 승리한다.

III. 제안하는 기법

본 장에서 우리는 래티스에서 ID 기반의 강한 지정된 검증자 서명 기법을 처음으로 제안한다. 우리가 제안하는 기법은 SIS 문제와 LWE 문제에 기반을 두어 설계되었으며, 랜덤 오라클 모델에서 안전하다.

먼저, 우리는 제안하는 기법에 사용될 파라미터들을 정의한다. 여기서 n은 시큐리티 파라미터이다. 그러면, 제안하는 기법에서 사용될 파라미터들은 다음과 같다:

- $m = O(n \log q)$
- $q > \beta \bullet \omega(\log n)$
- $\beta = poly(n)$
- $s = O(\sqrt{n \log q}) \bullet \omega(\sqrt{\log n})$

제안하는 기법에서 서명자를 S라 하고, 지정된 검증자를 V라 하자. 위에서 정의한 파라미터들을 사용하여, 제안하는 기법은 아래와 같이 구성된다:

- $Setup$: KGC는 다음과 같이 공개 파라미터 $params$와 마스터 비밀키 msk를 생성한다.
  1) 트랩도어 생성 알고리즘 $TrapGen(1^n, 1^m, q)$을 수행하여 (A_0, T_0)를 생성한다. 여기서 $A_0 \in Z_q^{n \times m}$이고, $T_0 \in Z^{m \times m}$이다.
  2) 세 개의 해시함수 $H_0 : \{0,1\}^* \rightarrow Z_q^{n \times m}$, $H_1 : \{0,1\}^* \times Z_q^n \rightarrow Z_q^n$, 그리고 $H_2 : \{0,1\}^* \times Z_q^n \rightarrow Z_q^{3m}$를 선택한다.
  3) 공개 파라미터는 $params = \{A_0, H_0, H_1, H_2\}$이고, 마스터 비밀키는 $msk = T_0$이다.
- $Extract(msk, ID_S, ID_V)$: KGC는 마스터 비밀키 msk, 서명자 S의 ID ID_S, 지정된 검증자 V의 ID ID_V를 입력으로 받아서 서명자 S의 비밀키 sk_S와 지정된 검증자 V의 비밀키 sk_V를 다음과 같이 생성한다.
  1) 트랩도어 확장 알고리즘 $ExtBasis(T_0, A_0 \| H_0(ID_S))$와 $ExtBasis(T_0, A_0 \| H_0(ID_V))$를 수행하여 $T_S \in Z^{2m \times 2m}$와 $T_V \in Z^{2m \times 2m}$를 각각 생성한다.
  2) 서명자 S의 비밀키는 $sk_S = T_S$이고, 지정된 검증자 V의 비밀키는 $sk_V = T_V$이다.
- $Sign(ID_S, sk_S, ID_V, params, b)$: 서명자 S는 자신의 ID ID_S, 자신의 비밀키 sk_S, 지정된 검증자 V의 ID ID_V, 공개 파라미터 $params$, 그리고 서명할 메시지 $b \in \{0,1\}^*$를 입력으로 받아서 서명 σ를 다음과 같이 생성한다.
  1) 난수 $t \in Z_q^n$와 $r' \in Z_q^n$를 랜덤하게 선택한다.
  2) 가우시안 에러 분포 $\Psi_{q,\alpha}^{2m}$를 따르는 벡터 $x \in Z_q^{2m}$를 샘플링 한다.
  3) 트랩도어 확장 알고리즘 $ExtBasis$ $(T_S, A_0 \| H_0(ID_S) \| H_0(ID_V))$를 수행하여 $T_{S,V} \in Z^{3m \times 3m}$를 생성한다.
  4) 가우시안 샘플링 알고리즘 $SampleD$ $(A_0 \| H_0(ID_S) \| H_0(ID_V), T_{S,V}, H_1(b,t), s)$를 사용하여 벡터 $v_{S,V} \in Z^{3m}$를 샘플링한다. 샘플링된 벡터 $v_{S,V}$는 극단적인 확률로 $\|v_S\| \le s\sqrt{3m}$를 만족한다. 만약 이를 만족하지 않는다면, 다시 샘플링한다.
  5) $\theta = v_{S,V} + H_2(b, r') \pmod q$를 계산한다.
  6) $r = (A_0 \| H_0(ID_V))^T \bullet r' + x \pmod q$를 계산한다.
  7) 서명은 $\sigma = \{\theta, r, t\}$이다.
- $Vrfy(ID_V, sk_V, ID_S, params, b, \sigma)$: 지정된 검증자 V는 자신의 ID ID_V, 자신의 비밀키 sk_V, 서명자의 ID ID_S, 공개 파라미터 $params$, 서명된 메시지 b, 그리고 서명 σ를 입력으로 받아서 서명을 다음과 같이 검증한다.
  1) $T_V^T \bullet r \pmod q = T_V^T \bullet x \pmod q = T_V^T \bullet x$를 계산한다.
  2) ${T_V^{-1}}^T \bullet T_V^T \bullet x = x$를 계산하여 x를 얻는다.
  3) $r = (A_0 \| H_0(ID_V))^T \bullet r' + x \pmod q$, x, 그리고 $(A_0 \| H_0(ID_V))$를 사용하여 r'을 얻는다.
  4) $\theta - H_2(b, r') = v_{S,V} \pmod q$를 계산하여 $v_{S,V}$를 얻는다.
  5) $(A_0 \| H_0(ID_S) \| H_0(ID_V)) \bullet v_{S,V} = H_1(b,t) \pmod q$와 $\|v_{S,V}\| \le s\sqrt{3m}$를 만족하는지 확인한다. 만약 모두 참이면 서명 σ는 정당하고, 그렇지 않다면 서명 σ는 정당하지 않다.
- $Simul(ID_V, sk_V, ID_S, params, b)$: 지정된 검증자 V는 자신의 ID ID_V, 자신의 비밀키 sk_V, 서명자 S의 ID ID_S, 공개 파라미터 $params$, 그리고 서명할 메시지 $b \in \{0,1\}^*$를 입력으로 받아서 시뮬

레이션된 서명 σ를 다음과 같이 생성한다.

1) 난수 $t \in Z_q^n$와 $r' \in Z_q^n$를 랜덤하게 선택한다.
2) 가우시안 에러 분포 $\Psi_{q,\alpha}^{2m}$를 따르는 벡터 $x \in Z_q^{2m}$를 샘플링한다.
3) 트랩도어 확장 알고리즘 $ExtBasis$ $(T_V, A_0 \| H_0(ID_S) \| H_0(ID_V))$를 수행하여 $T_{S,V} \in Z^{3m \times 3m}$를 생성한다.
4) 가우시안 샘플링 알고리즘 $SampleD$ $(A_0 \| H_0(ID_S) \| H_0(ID_V), T_{S,V}, H_1(b,t), s)$를 사용하여 벡터 $v_{S,V} \in Z^{3m}$를 샘플링한다. 샘플링된 벡터 $v_{S,V}$는 극단적인 확률로 $\|v_S\| \le s\sqrt{3m}$를 만족한다. 만약 이를 만족하지 않는다면, 다시 샘플링한다.
5) $\theta = v_{S,V} + H_2(b,r') \pmod q$를 계산한다.
6) $r = (A_0 \| H_0(ID_V))^T \cdot r' + x \pmod q$를 계산한다.
7) 시뮬레이션된 서명은 $\sigma = \{\theta, r, t\}$이다.

IV. 분 석

우리가 제안하는 기법은 정확성, 위조 불가능성, 비전이성, 서명자의 프라이버시를 만족한다. 본 장에서 우리는 제안하는 기법이 정확성, 위조 불가능성, 비전이성, 서명자의 프라이버시를 만족하는지에 대해 분석한다.

먼저, 제안하는 기법이 정확성을 만족하는지를 분석한다.

정리 1. 〔정확성〕 제안하는 기법은 정확성을 만족한다.

증명. 먼저, 서명자 S가 $Sign$ $(ID_S, sk_S, ID_V, params, b)$ 알고리즘을 사용하여 서명 $\sigma = \{\theta, r, t\}$를 생성하였다고 가정하자. 그러면 지정된 검증자 V는 $T_V^T \cdot r \pmod q = T_V^T \cdot x \pmod q = T_V^T \cdot x$와 $T_V^{-T} \cdot T_V^T \cdot x = x$를 계산하여 x를 구할 수 있다. 이것은 T_V가 기저이고, 역행렬이 존재하며, T_V와 x가 충분히 작은 값들로 구성되어있기 때문에 타당하다. 따라서 지정된 검증자 V는 r, x, $(A_0 \| H_0(ID_V))$를 모두 알 수 있으며, 이들과 $r = (A_0 \| H_0(ID_V))^T \cdot r' + x \pmod q$를 사용하여 r'을 구할 수 있다. 그러면 지정된 검증자 V는 θ, b, r'을 알기 때문에 $\theta - H_2(b,r') = v_{S,V} \pmod q$를 계산하여 $v_{S,V}$를 구할 수 있다. 만약 서명자 S가 서명 $\sigma = \{\theta, r, t\}$를 정당하게 생성하였다면,
$v_{S,V}$는 $(A_0 \| H_0(ID_S) \| H_0(ID_V)) \cdot v_{S,V} = H_1(b,t) \pmod q$와 $\|v_{S,V}\| \le s\sqrt{3m}$를 모두 만족하며, 따라서 지정된 검증자 V는 $Vrfy(ID_V, sk_V, ID_S, params, b, \sigma)$ 알고리즘을 사용하여 정당하게 생성된 서명 $\sigma = \{\theta, r, t\}$를 검증할 수 있다. □

그 다음으로, 제안하는 기법이 위조 불가능성을 만족하는지를 분석한다.

정리 2. 〔위조 불가능성〕 SIS 문제가 어렵다면, 제안하는 기법은 위조 불가능성을 만족한다.

증명. 우리는 여기서 챌린저와 공격자 사이의 게임을 통해 제안하는 기법이 위조 불가능성을 만족하는지를 증명한다. 챌린저는 SIS 문제를 풀고자 하고, 공격자는 제안하는 기법의 위조 불가능성을 공격하고자 한다. 제안하는 기법에서 사용되는 세 개의 해시함수 $H_0 : \{0,1\}^* \to Z_q^{n \times m}$, $H_1 : \{0,1\}^* \times Z_q^n \to Z_q^n$, 그리고 $H_2 : \{0,1\}^* \times Z_q^n \to Z_q^{3m}$는 모두 랜덤 오라클로 사용된다. $Setup$ 단계에서 챌린저는 $SIS_{n,3m,q,\beta}$ 문제의 입력으로 행렬 $A \in Z_q^{n \times 3m}$를 받아온다. 챌린저는 행렬 A를 $A = (A_0 \| A_1 \| A_2)$로 분해한다. 즉, $A_0 \in Z_q^{n \times m}$, $A_1 \in Z_q^{n \times m}$, $A_2 \in Z_q^{n \times m}$이다. 챌린저는 공격자에게 공개 파라미터 $params = \{A_0\}$를 제공한다. 그러면 공격자는 H_0, H_1, H_2에 대한 해시 오라클과 $Extract$, $Sign$, $Vrfy$ 오라클에 대한 접근이 가능하다.

$H_0(\cdot)$: 공격자는 자신이 선택한 ID ID_i를 최대 q_{H_0}번까지 질의할 수 있다. 만약 공격자가 예전에 H_0 오라클에 ID_i를 질의한 적이 있다면, 챌린저는 저장된 정보들을 사용하여 예전과 동일한 반환값으로 공격자에게 돌려준다. 챌린저는 q_{H_0}번 중의 2번에 한해서 ID_i에 대한 반환값으로 A_1과 A_2를 각각 공격자에게 돌려준다. 그리고 챌린저는 그 이외의 경우(최대 $q_{H_0} - 2$번)에 대해 트랩도어 생성 알고리즘 $TrapGen(1^n, 1^m, q)$을 수행하여 (A_i, T_i)를 생성하고, ID_i에 대한 반환값으로 A_i를 공격자에게 돌려준다. 여기서 $A_i \in Z_q^{n \times m}$이고, $T_i \in Z^{n \times m}$이다. 챌린저는 이때 생성한 정보들을 모두 저장한다. H_0 오라클은 $Extract$ 오라클과 $Sign$ 오라클을 통해서도 호출될 수 있으며,

이러한 경우에는 A_1 또는 A_2를 반환하지 않는다.

- $H_1(\bullet,\bullet)$: 공격자는 자신이 선택한 (b_i,t_i)를 질의할 수 있다. 만약 공격자가 예전에 (b_i,t_i)를 질의한 적이 있다면, 챌린저는 저장된 정보들을 사용하여 예전과 동일한 반환값으로 공격자에게 돌려준다. 챌린저는 $SampleDom$ 알고리즘을 수행하여 $v_i \in Z^{3m}$를 샘플링하고, $Av_i = u_i \pmod q$를 계산하여 (b_i,t_i)에 대한 반환값으로 u_i를 공격자에게 돌려준다. 여기서 $u_i \in Z_q^n$이다. 챌린저는 이때 생성한 정보들을 모두 저장한다.
- $H_2(\bullet,\bullet)$: 공격자는 자신이 선택한 (b_i,r'_i)를 질의할 수 있다. 만약 공격자가 예전에 (b_i,r'_i)를 질의한 적이 있다면, 챌린저는 저장된 정보들을 사용하여 예전과 동일한 반환값으로 공격자에게 돌려준다. 챌린저는 랜덤하게 $h_i \in Z_q^{3m}$를 선택하고, (b_i,r'_i)에 대한 반환값으로 h_i를 공격자에게 돌려준다. 챌린저는 이때 생성한 정보들을 모두 저장한다.
- $Extract(msk,\bullet)$: 공격자는 자신이 선택한 ID ID_i를 질의할 수 있다. 만약 공격자가 질의한 ID_i가 예전에 H_0 오라클에 질의했었고, 그것에 대한 반환값으로 A_1 또는 A_2를 공격자에게 돌려줬었다면, 챌린저는 여기서 게임을 종료한다. 만약 공격자가 예전에 H_0 오라클에 ID_i를 질의했었지만 위와 같은 경우가 없었다면, 챌린저는 H_0 오라클에 저장된 정보들을 사용하여 트랩도어 확장 알고리즘 $ExtBasis(T_i,A_0\|A_i)$를 수행하고, 이것에 대한 결과값인 $T_{0,i}$를 공격자에게 돌려준다. 만약 공격자가 선택한 ID_i가 $Extract$ 오라클 또는 H_0 오라클에 질의된 적이 없다면, 챌린저는 이 ID_i를 H_0 오라클에 질의하여 ID_i에 대응되는 (A_i,T_i)를 검색해오고, 트랩도어 확장 알고리즘 $ExtBasis(T_i,A_0\|A_i)$를 수행하여 이것에 대한 결과값인 $T_{0,i}$를 공격자에게 돌려준다. 챌린저는 이때 생성한 정보들을 모두 저장한다.
- $Sign(\bullet,sk_{S_i},\bullet,params,\bullet)$: 공격자는 자신이 선택한 (ID_{S_i},ID_{V_i},b_i)를 질의할 수 있다. 만약 공격자가 ID_{S_i}와 ID_{V_i} 둘 다 H_0 오라클에 질의했었고 ($Extract$ 오라클에 질의하여 H_0 오라클을 호출한 경우도 포함된다), 그것에 대한 결과값으로 (대응되는 순서와는 무관하게) A_1과 A_2를 반환했었다면, 챌린저는 여기서 게임을 종료한다. 그렇지 않다면, 챌린저는 ID_{S_i}와 ID_{V_i}를 모두 H_0 오라클에 질의하거나 저장된 정보들을 사용하여 A_{S_i}와 A_{V_i}를 얻고, 최소한 T_{S_i} 또는 T_{V_i} 중의 하나를 얻는다(위에서 A_1과 A_2 중에서 하나를 반환했던 경우라면 T_{S_i} 또는 T_{V_i} 중에서 하나만을 얻을 수 있고, 그렇지 않다면 T_{S_i}와 T_{V_i} 둘 다 얻을 수 있다). 그리고 챌린저는 트랩도어 확장 알고리즘 $ExtBasis(T_{S_i},A_0\|A_{S_i})$ 또는 $ExtBasis(T_{V_i},A_0\|A_{V_i})$를 수행하여 T_{0,S_i} 또는 T_{0,V_i}를 얻을 수 있다. 그러면 챌린저는 $Sign(ID_{S_i},T_{0,S_i},ID_{V_i},params,b_i)$ 알고리즘 또는 $Simul(ID_{V_i},T_{0,V_i},ID_{S_i},params,b_i)$ 알고리즘을 수행할 수 있고, 서명 또는 시뮬레이션된 서명 σ_i를 생성하여 공격자에게 반환한다.
- $Vrfy(\bullet,sk_{V_i},\bullet,params,\bullet,\bullet)$: 공격자는 자신이 선택한 $(ID_{S_i},ID_{V_i},b_i,\sigma_i)$를 질의할 수 있다. 만약 공격자가 예전에 $Extract$ 오라클에 ID_{V_i}를 질의한 적이 있다면, 챌린저는 $Extract$ 오라클에 저장된 정보들을 사용하여 ID_{V_i}에 대응되는 T_{0,V_i}를 얻는다. 그렇지 않다면, 챌린저는 $Extract$ 오라클에 ID_{V_i}를 질의하여 T_{0,V_i}를 얻는다. 그러면 챌린저는 $Vrfy(ID_{V_i},T_{0,V_i},ID_{S_i},params,b_i,\sigma_i)$ 알고리즘을 수행하여 서명 σ_i가 정당한지를 검증하고, 이 결과를 공격자에게 반환한다.

마지막으로, 공격자는 $(ID_S^*,ID_V^*,b^*,\sigma^*=(\theta^*,r^*,t^*))$를 챌린저에게 보낸다. 제안하는 기법의 위조 불가능성에 대한 공격을 성공한 공격자라면, $Vrfy(ID_V^*,sk_V^*,ID_S^*,params,b^*,\sigma^*)$ 알고리즘을 통해 검증이 되고, 공격자가 ID_S^*와 ID_V^*를 $Extract(msk,\bullet)$ 오라클에 질의하지 않았으며, (ID_S^*,ID_V^*,m^*)를 $Sign(\bullet,sk_i,\bullet,params,\bullet)$ 오라클에 질의하지 않았다. ID_S^*와 ID_V^*의 H_0 오라클에 대한 반환값이 (대응되는 순서와는 무관하게) A_1과 A_2가 아니라면, 챌린저는 여기서 게임을 종료한다. 그렇지 않다면 챌린저는 H_2 오라클에 저장된 모든 정보들을 확인하여 $b_i = b^*$인 경우의 모든 h_i들을 검색해오고, 이렇게 검색된 모든 h_i에 대해서 $\theta^* - h_i = v_{S_i,V_i} \pmod q$를 계산하고, 이렇게 계산된 v_{S_i,V_i} 중에서 $(A_0\|A_1\|A_2)\bullet v_{S_i,V_i}$

$= H_1(b^*, t^*) \pmod q$ 또는 $(A_0 \| A_2 \| A_1) \cdot v_{S, V_i} = H_1(b^*, t^*) \pmod q$와 $\| v_{S, V_i} \| \le s\sqrt{3m}$를 만족하는 것을 찾은 후, 이것을 v^*라고 둔다. 랜덤 오라클의 특성상 이것을 만족하는 것은 반드시 존재한다. 그러고 난 후, 챌린저는 H_1 오라클에 저장된 모든 정보들을 확인하여 $b_i = b^*$이고 $t_i = t^*$인 경우의 v_i를 검색해온다. 역시 랜덤 오라클의 특성상 이것을 만족하는 것은 반드시 존재한다. 위에서 $(A_0 \| A_1 \| A_2) \cdot v^* = H_1(b^*, t^*) \pmod q$인 경우 $(A_0 \| A_1 \| A_2) \cdot v^* = (A_0 \| A_1 \| A_2) \cdot v_i \pmod q$를 만족하며, 따라서 $(A_0 \| A_1 \| A_2) \cdot (v^* - v_i) = 0 \pmod q$을 만족한다. 위에서 $(A_0 \| A_2 \| A_1) \cdot v^* = H_1(b^*, t^*) \pmod q$인 경우도 간단한 치환만으로 동일한 결과를 얻을 수 있다. 그러면 챌린저는 $SIS_{n, 3m, q, \beta}$ 문제의 입력 $A = (A_0 \| A_1 \| A_2)$에 대한 출력으로 $(v^* - v_i)$를 출력한다.

위의 게임에서 게임이 중간에 종료되는 경우는 *Extract* 오라클, *Sign* 오라클, 그리고 마지막 단계에서 존재한다. 마지막 단계에서 ID_S^*와 ID_V^*의 H_0 오라클에 대한 반환값이 (대응되는 순서와는 무관하게) A_1과 A_2라면, *Extract* 오라클과 *Sign* 오라클에서 게임이 중간에 종료되는 경우는 발생하지 않는다. 따라서 우리는 마지막 단계에서 발생할 확률만을 고려한다. 마지막 단계에서 게임이 종료되지 않고 진행될 확률은 최소한 $\frac{2}{q_{H_0}^2 - q_{H_0}} - negl(n)$이며, 제안하는 기법의 위조 불가능성에 대한 공격자의 공격 성공 확률을 ϵ라고 할 때, 챌린저는 최소한 $\frac{2\epsilon}{q_{H_0}^2 - q_{H_0}} - negl(n)$의 확률로 $SIS_{n, 3m, q, \beta}$ 문제를 풀 수 있다. □

세 번째로, 제안하는 기법이 비전이성을 만족하는지를 분석한다.

정리 3. 〔비전이성〕 제안하는 기법은 비전이성을 만족한다.

증명. 제안하는 기법에서 서명자 S가 서명 σ를 생성하는 $Sign(ID_S, sk_S, ID_V, params, b)$ 알고리즘과 지정된 검증자 V가 시뮬레이션된 서명 σ를 생성하는 $Simul(ID_V, sk_V, ID_S, params, b)$ 알고리즘의 차이는 오직 3번째 단계에서만 존재한다. 3번째 단계에서 두 알고리즘들 모두 $T_{S, V} \in Z^{3m \times 3m}$를 생성하기 위해 트랩도어 확장 알고리즘 $ExtBasis(T_S, A_0 \| H_0(ID_S) \| H_0(ID_V))$와 $ExtBasis(T_V, A_0 \| H_0(ID_S) \| H_0(ID_V))$를 각각 수행한다. $ExtBasis(T_S, A_0 \| H_0(ID_S) \| H_0(ID_V))$ 알고리즘을 수행하여 생성된 $T_{S, V}$는 $\| \widetilde{T_{S, V}} \| = \| \widetilde{T_S} \| = \| \widetilde{T_0} \|$를 만족하며, $ExtBasis(T_V, A_0 \| H_0(ID_S) \| H_0(ID_V))$ 알고리즘을 수행하여 생성된 $T_{S, V}$는 $\| \widetilde{T_{S, V}} \| = \| \widetilde{T_V} \| = \| \widetilde{T_0} \|$를 만족한다. 각각의 알고리즘들을 통해 생성된 $T_{S, V}$는 모두 $\Lambda^{\perp}(A_0 \| H_0(ID_S) \| H_0(ID_V))$의 기저이며, 각각의 알고리즘을 통해 생성된 $T_{S, V}$는 모두 $\| \widetilde{T_{S, V}} \| = \| \widetilde{T_0} \|$를 만족하기 때문에 두 알고리즘들 모두 4번째 단계에서 가우시안 샘플링 알고리즘 $SampleD(A_0 \| H_0(ID_S) \| H_0(ID_V), T_{S, V}, H_1(b, t), s)$를 사용하여 동일한 분포로부터 $v_{S, V} \in Z^{3m}$를 샘플링할 수 있다. $v_{S, V}$를 샘플링하는 과정을 제외하고는 $Sign(ID_S, sk_S, ID_V, params, b)$ 알고리즘과 $Simul(ID_V, sk_V, ID_S, params, b)$ 알고리즘이 완전히 동일하며, 따라서 서명자 S가 $Sign(ID_S, sk_S, ID_V, params, b)$ 알고리즘을 사용하여 생성한 서명 σ와 지정된 검증자 V가 $Simul(ID_V, sk_V, ID_S, params, b)$ 알고리즘을 사용하여 생성한 시뮬레이션된 서명 σ를 서로 구별하는 것은 불가능하다. □

마지막으로, 제안하는 기법이 서명자의 프라이버시를 만족하는지를 분석한다.

정리 4. 〔서명자의 프라이버시〕 LWE 문제가 어렵다면, 제안하는 기법은 서명자의 프라이버시를 만족한다.

증명. 제안하는 기법에서 서명 $\sigma = \{\theta, r, t\}$을 생성하기 위해 $Sign(ID_S, sk_S, ID_V, params, b)$ 알고리즘에서 사용되는 $r = (A_0 \| H_0(ID_V))^T \cdot r' + x \pmod q$은 LWE 문제의 입력 형태와 동일하다. 다시 말하면, r은 LWE 문제의 어려움에 기반을 두고 있는 형태이기 때문에 지정된 검증자 V의 비밀키 $sk_V = T_V$를 모르는 어떤 누구도 r로부터 r'을 계산해낼 수 없다. r로부터 r'을 계산해내지 못한다면, 당연히 $\theta - H_2(b, r') = v_{S, V} \pmod q$로부터 $v_{S, V}$를 계산해낼 수 없게 된다. 따라서 r로부터 r'을 계산해낼 수 있는 지정된 검증자 V를 제외하고는 어떤 누구도 서명 $\sigma = \{\theta, r, t\}$가 어떤 서명자에 의해 생성된 서명인지를 알아낼 수 없다. □

V. 결 론

본 논문에서 우리는 래티스에서 ID 기반의 강한 지정된 검증자 서명 기법을 최초로 제안하였다. 우리가 제안한 기법은 래티스에서 정의된 SIS 문제와 LWE 문제에 기반을 두어 설계되었으며, 랜덤 오라클 모델에서 안전성이 증명되었다.

참고문헌

[1] M. Jakobsson, K. Sako, and R. Impagliazzo, "Designated verifier proofs and their applications," Advances in Cryptology, EUROCRYPT '96, LNCS 1070, pp. 143-154, May 1996.

[2] H. Lipmaa, G. Wang, and F. Bao, "Designated verifier signature schemes: attacks, new security notions and a new construction," Proceedings of the 32nd International Colloquium on Automata, Languages and Programming, LNCS 3580, pp. 459-471, Jul. 2005.

[3] Y. Li, W. Susilo, Y. Mu, and D. Pei, "Designated verifier signature: definition, framework and new constructions," Proceedings of the 4th International Conference on Ubiquitous Intelligence and Computing, LNCS 4611, pp. 1191-1200, Jul. 2007.

[4] S. Saeednia, S. Kremer, and O. Markowitch, "An efficient strong designated verifier signature scheme," Proceedings of the 6th Annual International Conference on Information Security and Cryptology, LNCS 2971, pp. 40-54, Nov. 2003.

[5] 구영주, 천지영, 최규영, 이동훈, "인증서가 없는 강한 지정된 검증자 서명기법," 정보보호학회 논문지, 18(6(A)), pp. 27-37, 2008년 12월.

[6] Q. Huang, G. Yang, D.S. Wong, and W. Susilo, "Efficient strong designated verifier signature schemes without random oracle or with non-delegatability," The International Journal of Information Security, vol. 10, no. 6, pp. 373-385, Aug. 2011.

[7] H. Tian, X. Chen, Z. Jiang, and Y. Du, "Non-delegatable strong designated verifier signature on elliptic curves," Proceedings of the 14th Annual International Conference on Information Security and Cryptology, LNCS 7259, pp. 219-234, Dec. 2012.

[8] F. Wang, Y. Hu, and B. Wang, "Lattice-based strong designate verifier signature and its applications," Malaysian Journal of Computer Science, vol. 25, no. 1, pp. 11-22, Jan. 2012.

[9] W. Susilo, F. Zhang, and Y. Mu, "Identity-based strong designated verifier signature schemes," Proceedings of the 9th Australasian Conference on Information Security and Privacy, LNCS 3108, pp. 313-324, Jul. 2004.

[10] A. Shamir, "Identity-base cryptosystems and signature schemes," Advances in Cryptology, CRYPTO '84, LNCS 196, pp. 47-53, Aug. 1984.

[11] F. Zhang, R. Safavi-Naini, W. Susilo, "ID-based chameleon hashes from bilinear pairings," IACR ePrint 2003-208, Sep. 2003.

[12] G. Ateniese and B. Medeiros, "Identity-based chameleon hash and applications," Proceedings of the 8th International Conference on Financial Cryptography, LNCS 3110, pp. 164-180, Feb. 2004.

[13] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert, "Bonsai trees, or how to delegate a lattice basis," Advances in Cryptology, EUROCRYPT '10, LNCS 6110, pp. 523-552, Jun. 2010.

[14] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," Journal of the ACM, vol. 56, no. 6, pp. 34:1-34:40, Sep. 2009.

[15] C. Peikert and A. Rosen, "Efficient colli-

sion-resistant hashing from worst-case assumptions on cyclic lattices," Proceedings of the 3rd Theory of Cryptography Conference, LNCS 3876, pp. 145-166, Mar. 2006.

〔16〕 D. Micciancio and O. Regev, "Worst-case to average-case reductions based on Gaussian measures," SIAM Journal on Computing, vol. 37, no. 1, pp. 267-302, Apr. 2007.

〔17〕 C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," Proceedings of the 40th Annual ACM Symposium on Theory of Computing, pp. 197-206, May 2008.

〔18〕 M. Ajtai, "Generating hard instances of lattice problems," Proceedings of the 28th Annual ACM Symposium on the Theory of Computing, pp. 99-108, May 1996.

〔19〕 M. Ajtai, "Generating hard instances of the short basis problem," Proceedings of the 26th International Colloquium on Automata, Languages and Programming, LNCS 1644, pp. 1-9, Jul. 1999.

〔20〕 J. Alwen and C. Peikert, "Generating shorter bases for hard random lattice," Proceedings of the 26th International Symposium on Theoretical Aspects of Computer Science, pp. 75-86, Feb. 2009.

〈著者紹介〉

노 건 태 (Geontae Noh) 학생회원
2008년 2월: 고려대학교 산업시스템정보공학과 졸업
2010년 2월: 고려대학교 정보경영공학과 석사
2010년 3월~현재: 고려대학교 정보보호학과 박사과정
〈관심분야〉 암호 이론, 프라이버시 향상 기술 (PET), 유비쿼터스 보안

천 지 영 (Ji Young Chun) 학생회원
1997년 2월: 이화여자대학교 수학과 졸업
2006년 2월: 고려대학교 정보보호학과 석사
2011년 8월: 고려대학교 정보경영공학과 박사
2011년 9월~현재: 고려대학교 정보보호연구원 연구교수
2012년 8월~현재: University of Illinois at Urbana-Champaign, Security Lab. 박사 후 연구원
〈관심분야〉 암호 이론, 프라이버시 향상 기술 (PET), 유비쿼터스 보안

정 익 래 (Ik Rae Jeong) 정회원
1998년 2월: 고려대학교 전산학과 졸업
2000년 2월: 고려대학교 전산학과 석사
2004년 8월: 고려대학교 정보보호학과 박사
2006년 3월~2008년 2월: 한국전자통신연구원 암호기술연구팀 선임연구원
2008년 3월~현재: 고려대학교 정보보호대학원 교수
〈관심분야〉 암호 이론, 프라이버시 향상 기술 (PET), 데이터베이스 암호