정 보 보 호 학 회 지
제 33 권 제3호, 2023. 6

KpqC 공모전에 제출된 Fiat-Shamir with aborts 구조의 격자 기반 서명 기법 분석

홍 가 희*, 우 주*, 박 종 환**

요 약

양자 컴퓨팅의 발전으로 기존의 전자서명 기법에 사용되던 소인수분해 문제와 이산로그 문제가 다항 시간 내에 풀린다. 그에 따라 국내외에서는 양자 컴퓨팅 환경에서도 안전한 암호 기법에 대한 연구가 활발히 진행되고 있다. 미국 국립 표준 기술 연구소에서 양자 내성 암호 기법의 표준을 설립하고자 Post-Quantum Cryptography Standardization Process를 진행하였으며 전자서명 기법으로는 CRYSTALS-Dilithium, FALCON, SPHINCS+가 표준으로 선택되었다. 국내에서도 양자 내성 암호 표준 수립을 위하여 KpqC 공모전이 개최되었다. 본 논문에서는 KpqC 공모전 Round 1에 제안된 격자 기반 전자서명 기법 중 Dilithium과 같이 Fiat-Shamir with aborts paradigm 구조로 설계된 3개의 기법, HAETAE, GCKSign, NCC-Sign을 분석하고 Dilithium과 함께 비교하였다.

Ⅰ. 서 론

다양한 종이 문서가 전자화됨에 따라 수기 서명을 대체할 수 있는 전자서명이 요구된다. 전자서명은 수기 서명과 동일한 효과를 가지면서 온라인에서 수행될 수 있기 때문에 인터넷 뱅킹이나 온라인 결제 시스템, 전자문서 공증 등에 사용된다.

전자서명 기법은 서명 생성과 검증에 사용할 키 쌍을 생성하고, 서명자가 본인만 알고 있는 서명키 (비밀키)를 이용하여 서명값을 생성하면 검증자가 서명자의 검증키 (공개키)로 해당 서명이 올바른 서명인지를 검증하는 절차로 구성되어 있다. 이때 올바른 서명이란 서명한 메시지의 내용이 위변조되지 않고, 서명자가 자신만 알고 있는 비밀키로 서명한 정당한 서명자임을 의미한다. 이러한 전자서명 기법은 소인수분해문제, 이산로그 문제 등이 어렵다는 것에 기반하여 설계되며 RSA, ElGamal 등이 그 예시이다.

컴퓨팅 기술의 발전으로 인해 양자컴퓨터가 등장하면서 기존의 난제인 소인수분해 문제, 이산로그 문제 등이 다항시간 내에 풀린다[1]. 즉, 양자 컴퓨팅 기술이 더욱 발전된 미래에는 기존의 전자서명 기법은 더 이상 안전하지 않게 되므로 기존의 전자서명을 대체할 수 있는 양자 컴퓨팅 환경에서도 안전한 전자서명 기법이 요구된다.

양자 내성 암호 기법은 격자 기반, 코드 기반 등의 기존과는 다른 난제를 사용하여 양자 컴퓨팅 환경에서도 안전하도록 설계되었다. 미국 국립 표준 기술 연구소 (National Institute of Standards and Technology, NIST)에서 양자 내성 암호 기법의 표준을 설립하고자 Post-Quantum Cryptography (PQC) Standardization Process를 진행하였으며, 전자서명 기법으로 CRYSTALS-Dilithium[2], FALCON[3] 등이 표준으로 선택되었다. 한편, 국내에서도 양자 내성 암호 표준 수립을 위하여 KpqC 공모전이 개최되었다.

본 논문에서는 KpqC 공모전 Round 1에 제안된 격자 기반 전자서명 기법 중 Dilithium과 같이 Fiat-Shamir with aborts paradigm[4] 구조로 설계된 3개의 기법, HAETAE[7], GCKSign[13], NCC-Sign[5]을 분석하고 Dilithium과 함께 비교하였다.

본 논문의 구성은 다음과 같다. 2장에서는 배경 지식 및 기존 연구를 살펴본다. 3장에서는 HAETAE와 GCKSign, NCC-Sign을 소개한다. 4장에서는 해당 기법들을 분석하고 Dilithium과 함께 비교한다. 마지막

본 연구는 고려대 암호기술 특화연구센터(UD210027XD)를 통한 방위사업청과 국방과학연구소의 연구비 지원으로 수행되었습니다.
* 고려대학교 정보보호대학원 정보보호학과 (대학원생, hongh@korea.ac.kr, 대학원생, woojoo0121@korea.ac.kr)
** 상명대학교 컴퓨터과학과 (교수, jhpark@smu.ac.kr)

으로 5장에서는 결론을 맺는다.

Ⅱ. 배경 지식 및 기존 연구

본 논문에서 사용할 기호들을 다음과 같이 정의한다. 정수 $k \in Z$에 대하여 $k' = k \bmod^{\pm} \alpha$는 $k = k' \bmod \alpha$를 만족하는 $[-\lfloor \alpha/2 \rfloor, \lfloor \alpha/2 \rfloor)$에서의 유일한 값이다. $Z_q = Z/qZ$는 modulus q에 대한 quotient ring이다. R과 R_q는 각각 $Z[x]/\phi(x)$와 $Z_q[x]/\phi(x)$로 Dilithium과 HAETAE, GCKSign 에서는 $n = 2^k$인 $\phi(x) = x^n + 1$을 사용하고, NCC-Sign은 $\phi(x) = x^p - x - 1$을 사용한다. 환의 원소들로 이루어진 벡터는 $\boldsymbol{a} = (a_1, a_2, \dots, a_m) \in R_q^m$과 같이 소문자의 굵은 글씨로 표기되며, 행렬은 대문자의 굵은 글씨로 표기된다. n-1차 다항식으로 구성된 환의 원소 $a = a_0 + a_1 x + \dots + a_{n-1} x^{n-1} \in R$에 대하여 L_2 $norm$은 $\| a \|_2 = (a_0^2 + a_1^2 + \dots + a_{n-1}^2)^{1/2}$이고 L_∞ $norm$은 $\| a \|_\infty = \max \left| a_i \bmod^{\pm} q \right|$이다. 이는 벡터로도 확장이 가능하다. S_η는 모든 계수가 $[-\eta, \eta] \cap Z$ 구간 내 있는 R_q의 원소들의 집합이다. $R_{\mathbb{R}} = \mathbb{R}[x]/(x^n + 1)$에 대하여 $B_{R,m}(r, \boldsymbol{c}) = \left\{ \boldsymbol{x} \in R_{\mathbb{R}}^m, \ \| \boldsymbol{x} - \boldsymbol{c} \|_2 < r \right\}$는 dimension이 m이고 중심이 $\boldsymbol{c} \in R^m$, 반지름이 r인 continuous hyperball 이다. 양의 정수 N에 대하여 $B_{(1/N)R,m}(r, \boldsymbol{c}) = (1/N) R^m \cap B_{R,m}(r, \boldsymbol{c})$은 discretized hyperball이다. $r' = r \bmod q$에 대하여, $Low_q(r, \alpha)$는 $r' \bmod^{\pm} \alpha$이고 $High_q(r, \alpha)$는 $(r' - Low_q(r, \alpha))/\alpha$이다.

2.1. Fiat-Shamir with aborts[4][6]

2009년 Lyubashevsky에 의해 제안된 Fiat-Shamir with aborts[4]는 rejection sampling을 이용한 격자 기반 전자서명 기법이다. 공개키는 격자 기반 난제인 SIS를 이용하여 생성된다. 임의의 벡터 $\boldsymbol{a} \in R_q^m$와 짧은 길이의 벡터 $\boldsymbol{s} \in S_\eta^m$에 대해 공개키는 $(\boldsymbol{a}, \boldsymbol{t} = \boldsymbol{as} \bmod q)$, 비밀키는 $\boldsymbol{s}$로 주어진다. 짧은 길이의 해시값 c에 대하여, 메시지 μ의 서명은 $(c, \boldsymbol{z}) = (H(\boldsymbol{ay} \bmod q, \mu), \boldsymbol{y} + \boldsymbol{s}c)$이다. $\boldsymbol{z}$를 통해 $\boldsymbol{s}$의 값을 유추할 수 없도록 적당히 큰 $\boldsymbol{y}$를 뽑아서 $\boldsymbol{z} = \boldsymbol{y} + \boldsymbol{s}c$과 같이 선형적으로 비밀 값을 가려준다. 이때, 서명 값 $\boldsymbol{z}$는 비밀 값 $\boldsymbol{s}$에 대한 정보를 노출하지 않도록 rejection sampling을 통해 $\boldsymbol{s}$와 독립적인 분포를 가진다. 마지막으로 서명 검증 시에는 $\boldsymbol{s}$가 충분히 작고 $c = H(\boldsymbol{az} - \boldsymbol{t}c \bmod q, \mu)$을 만족하는지 확인한다.

rejection sampling 방법은 난수 $\boldsymbol{y}$의 분포에 따라 달라진다. 먼저 [4]에서는 난수 $\boldsymbol{y}$를 $\| \boldsymbol{y} \|_\infty \leqq B$를 만족하는 균등분포 (uniform distribution)에서 뽑는다. $\boldsymbol{s}c$의 크기가 $\| \boldsymbol{s}c \|_\infty \leqq \beta$를 만족하도록 정의되어 있을 때 $\boldsymbol{z}$의 모든 원소가 $\| \boldsymbol{z} \|_\infty \leqq B - \beta$를 만족해야 $\boldsymbol{z}$로부터 비밀 값 $\boldsymbol{s}$가 노출되지 않는다. $\boldsymbol{z}$가 $\| \boldsymbol{z} \|_\infty \leqq B - \beta$를 만족할 확률은 $\{(2(B-\beta)-1)/2B\}^m$이다. 한편, 해당 기법은 Generalized Compact Knapsack (GCK) 함수의 충돌쌍을 찾는 것으로 리덕션 되는데, 안전성 증명에서 GCK 함수의 충돌쌍은 위조 서명 $\boldsymbol{z}$로부터 얻어진다. 즉, $\boldsymbol{z}$가 GCK 함수의 충돌쌍이 되므로 B가 작을수록 security bits는 증가한다.

반면, 2012년 Lyubashevsky에 의해 제안된 논문[6]에서는 난수 $\boldsymbol{y}$를 가우시안 분포 (gaussian distribution)에서 뽑아 rejection sampling을 확률적으로 수행한다. $\boldsymbol{y} + \boldsymbol{s}c$의 분포를 g, 비밀 정보를 노출하지 않는 $\boldsymbol{s}$와 독립적인 분포를 f 라고 할 때, $\boldsymbol{y} + \boldsymbol{s}c$의 분포를 f로 만드는 것이 목표이다. g의 분포를 따르는 값을 $f(z)/(M \cdot g(z))$의 확률로 출력하는 것은 f의 분포를 $1/M$의 확률로 출력하는 것과 동일한 분포를 갖는다[6]. 이때, M은 반복 기대 횟수를 의미하므로 M은 작을수록 서명 생성 시간이 줄어든다. 한편, 안전성 증명과정에서 서명 위조 공격자에 대해 비밀키 없이 서명 쿼리에 응답해야 한다. 비밀키를 사용하지 않고 f 분포를 $1/M$의 확률로 출력하면 공격자는 비밀키 없이 만든 서명임을 알아채지 못한다[6]. 즉, f 분포를 서명 생성의 simulator로 사용할 수 있다. 이러한 f를 타겟분포 (target distribution), g를 생성분포 (source distribution) 라고 한다.

2.2. CRYSTALS-Dilithium[2]

2.2.1. Dilithium의 설계원리

2009년과 2012년에 Lyubashevsky가 제안한 논문에서 공개키의 구조를 격자 기반 난제 Module-LWE (MLWE)로 변경하였다. [알고리즘 1] Sign 5와 같이 정의된 서명 값[20]으로부터 해시함수의 입력 값은 다

음과 같이 변경된다.

$$\boldsymbol{Az}-\boldsymbol{t}c=\boldsymbol{A}(\boldsymbol{y}+c\boldsymbol{s_1})-(\boldsymbol{As_1}+\boldsymbol{s_2})c=\boldsymbol{w}-c\boldsymbol{s_2} \quad (1)$$

$c\boldsymbol{s_2}$의 값은 매우 작은 값으로 하위비트에만 영향을 미친다. 해시함수의 입력값이 조금만 달라도 해시값이 완전히 달라진다는 점에서 두 입력값이 같아야 하므로, $\boldsymbol{Az}-\boldsymbol{t}c$의 상위비트와 $\boldsymbol{w}$의 상위비트를 해시함수의 입력값으로 사용한다.

더불어, 공개키의 크기를 줄이기 위하여 $\boldsymbol{t}$의 하위 d 비트를 잘라낸 나머지 $\boldsymbol{t_1}$을 공개키로 사용한다. 이때의 해시함수 입력값은 다음과 같다.

$$\boldsymbol{Az}-c\boldsymbol{t_1}\cdot 2^d=\boldsymbol{w}-c\boldsymbol{s_2}+c\boldsymbol{t_0} \quad (2)$$

알고리즘 1. CRYSTALS-Dilithium

KeyGen(1^λ)

1. $(\boldsymbol{s_1},\boldsymbol{s_2})\leftarrow S_\eta^l\times S_\eta^k$
2. $\boldsymbol{A}\leftarrow R_q^{k\times l}$
3. $\boldsymbol{t}:=\boldsymbol{As_1}+\boldsymbol{s_2} \bmod q$
4. $\boldsymbol{t_1}\cdot 2^d+\boldsymbol{t_0}=\boldsymbol{t}$
5. return $sk=(\boldsymbol{A},\boldsymbol{s_1},\boldsymbol{s_2},\boldsymbol{t}), vk=(\boldsymbol{A},\boldsymbol{t_1})$

Sign(sk,μ)

1. $\boldsymbol{y}\leftarrow S_{\gamma_1}^l$
2. $\boldsymbol{w}:=\boldsymbol{Ay} \bmod q$
3. $\boldsymbol{w_1}=High_q(\boldsymbol{w},2\gamma_2)$
4. $c:=H(\mu\|\boldsymbol{w_1})$
5. $\boldsymbol{z}:=\boldsymbol{y}+c\boldsymbol{s_1}$
6. if $\|\boldsymbol{z}\|_\infty\geq\gamma_1-\beta$ or $\|c\boldsymbol{t_0}\|_\infty\geq\gamma_2$ or $\|Low_q(\boldsymbol{w}-c\boldsymbol{s_2},2\gamma_2)\|_\infty\geq\gamma_2-\beta$ then go to step 1.
7. else $h:=High_q(\boldsymbol{w}-c\boldsymbol{s_2}+c\boldsymbol{t_0},2\gamma_2)-High_q(\boldsymbol{w}-c\boldsymbol{s_2},2\gamma_2)$
8. return $\sigma=(\boldsymbol{z},h,c)$

Verify(vk,μ,σ)

1. $\boldsymbol{w_1}':=High(\boldsymbol{Az}-c\boldsymbol{t_1}\cdot 2^d,2\gamma_2)-h \bmod (q-1)/\alpha$
2. return $[\|\boldsymbol{z}\|_\infty<\gamma_1-\beta]\wedge[c=H(\mu\|\boldsymbol{w_1}')]$

$\|Low_q(\boldsymbol{w}-c\boldsymbol{s_2},2\gamma_2)\|_\infty<\gamma_2-\beta$, $\|c\boldsymbol{t_0}\|_\infty<\gamma_2$ 이므로 $\|Low_q(\boldsymbol{w}-c\boldsymbol{s_2}+c\boldsymbol{t_0},2\gamma_2)\|_\infty>\gamma_2$ 인 경우 $\boldsymbol{w}-c\boldsymbol{s_2}$와 $\boldsymbol{w}-c\boldsymbol{s_2}+c\boldsymbol{t_0}$의 상위비트는 달라진다. 이때, $\|Low_q(\boldsymbol{w}-c\boldsymbol{s_2}+c\boldsymbol{t_0},2\gamma_2)\|_\infty<2\gamma_2$를 만족하므로 상위비트의 마지막 한 비트에만 영향을 미친다. [알고리즘 1] Sign 7과 같이 $\boldsymbol{w}-c\boldsymbol{s_2}$와 $\boldsymbol{w}-c\boldsymbol{s_2}+c\boldsymbol{t_0}$의 상위비트의 오차를 힌트값 h로 정의하여 서명과 함께 제공하고 해시함수의 입력값으로는 $\boldsymbol{w}$와 $\boldsymbol{w}-c\boldsymbol{s_2}+c\boldsymbol{t_0}$의 상위비트를 사용한다.

2.2.2. Dilithium의 안전성 증명

Dilithium은 MLWE를 기반으로 공개키를 생성하여 공개키로부터 비밀키의 복구에 대한 안전성은 MLWE의 어려움에 의존한다. quantum random oracle model (QROM)에서 서명 위조에 대한 안전성은 신규 난제인 SelfTargetMSIS[2][11]의 어려움에 근거하고 있다.

Ⅲ. KpqC 전자서명 기법

KpqC에 제안된 전자서명 기법 중 Fiat-Shamir with aborts[4]구조의 전자서명 기법들을 살펴본다.

3.1. HAETAE[7]

HAETAE는 서울대학교와 Crypto Lap, 프랑스의 Ecole Normale Supérieure de Lyon과 Institut Universitaire de France, 독일의 Ruhr Universit¨at Bochum 소속의 연구진이 개발한 전자서명 기법으로, hyperball bimodal rejection sampling을 도입하여 서명과 공개키의 크기를 줄였다.

3.1.1. HAETAE 설계원리

HAETAE는 난수 $\boldsymbol{y}$를 hyperball의 균등분포에서 선택하고, 서명 생성 과정에서 hyperball bimodal rejection sampling을 수행하여 공개키와 서명의 크기를 줄였다[8].

BLISS[9]는 최초로 제안된 bimodal 형태의 격자 기반 전자서명 기법으로, 가우시안 분포에서 뽑은 난수 $\boldsymbol{y}$에 대하여 서명의 형태를 [알고리즘 2] **Sign** 6

과 같이 정의하였다. 서명의 분포가 bimodal 가우시안 분포로 정의됨에 따라, 고정된 반복 횟수 M에 대해 서명 분포의 표준편차를 줄일 수 있어 서명의 크기가 감소한다.

$b=1$일 때, 서명 검증 과정에서 $\boldsymbol{Ay}=\boldsymbol{Az}-\boldsymbol{t}c$를 만족하지 않는다. 따라서 modulus를 q에서 $2q$로 변경하고 공개키는 $\boldsymbol{t}=q\boldsymbol{j}$ $(\boldsymbol{j}=(1,0,...,0)\in R^k)$ 를 만족하도록 설계하였다. 해당 기법은 bimodal rejection sampling을 통해 표준편차를 줄임으로써 unimodal 가우시안 분포로 이루어져 있는 2012년도 Lyubashevsky의 기법[6]보다 modulus q와 서명, 공개키의 크기가 작아졌다. 그러나 가우시안 분포는 서명 크기를 작다는 이점에도 불구하고 확률적인 rejection sampling[6]으로 인해 구현 상의 어려움이 있다.

HAETAE는 난수 $\boldsymbol{y}$를 hyperball에서 sampling하고 bimodal 형태의 서명 구조를 선택하여, 단순화된 rejection 조건으로 구현 상의 어려움을 줄이면서 서명의 크기는 작게 유지하고자 하였다. 서명의 크기와 반복 횟수 M은 서명의 분포가 bimodal hyperball에서의 균등분포일 때 가장 작음[8]을 이용하였다. 가우시안 분포는 bimodal로 바꿈으로써 공개키와 서명의 크기는 감소하지만 rejection sampling의 확률을 계산할 때 비밀값 $\boldsymbol{s}$에 대한 연산을 수행해야 하므로 부채널 공격에 취약하다. 반면, hyperball에서의 균등분포는 bimodal로 바꿨을 때, [알고리즘 2] **Sign** 8, 9와 같이 rejection sampling에서 비밀값 $\boldsymbol{s}$에 대한 연산을 수행하지 않으면서 bimodal 가우시안 분포만큼 서명의 크기를 줄일 수 있다[8]. 그러나 hyperball sampling에서 구현 상의 어려움이 있다는 한계가 존재한다.

한편, 서명의 크기를 줄이기 위하여 $\boldsymbol{Az}$에서 $2\boldsymbol{ID_k}$와 곱해지는 부분을 $\boldsymbol{z_2}$로 정의하고 $\boldsymbol{z}=(\boldsymbol{z_1},\boldsymbol{z_2})$중 $\boldsymbol{z_1}$만 서명 값으로 사용한다. 검증자에게 서명 값의 일부만 제공되므로 $\boldsymbol{Ay}$의 상위비트를 해시값 c의 입력으로 하고, $\boldsymbol{w}$의 상위비트와 $\boldsymbol{w}-2\boldsymbol{z_2}$의 상위비트 차이를 힌트로 제공하여 서명 검증이 가능하도록 하였다.

또한, hyperball의 분포를 이용함에 따라 hypercube로 encoding할 경우 entropy loss가 발생한다. hyperball에 최적화된 크기로 encoding 할 수 있도록 range Asymmetric Numeral System encoding(rANS encoding)[10]를 서명 $\boldsymbol{z_1}$의 상위비트와 힌트 $\boldsymbol{h}$에 적용하여 서명의 크기를 감소시켰다.

알고리즘 2. HAETAE

KeyGen(1^λ)

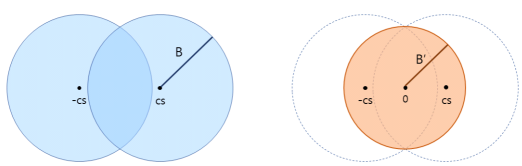
1. $(\boldsymbol{a}|\boldsymbol{A_{gen}})\leftarrow R_q^{k\times l}$
2. $(\boldsymbol{s_{gen}},\boldsymbol{e_{gen}})\leftarrow S_\eta^{l-1}\times S_\eta^k$
3. $\boldsymbol{b}:=\boldsymbol{a}+\boldsymbol{A_{gen}}\cdot\boldsymbol{s_{gen}}+\boldsymbol{e_{gen}} \mod q$
4. $(\boldsymbol{b_0},\boldsymbol{b_1}):=(Low_q(\boldsymbol{b},d),High_q(\boldsymbol{b},d))$
5. $\boldsymbol{A}:=(2(\boldsymbol{a}-\boldsymbol{b_1})+q\boldsymbol{j}|2\boldsymbol{A_{gen}}|2\boldsymbol{ID_k}) \mod 2q$
6. $\boldsymbol{s}:=(1,\boldsymbol{s_{gen}},\boldsymbol{e_{gen}}-\boldsymbol{b_0})$
7. if $f(\boldsymbol{s})>n\beta^2/\tau$ then go to 2
8. return $sk=(\boldsymbol{s}),vk=(\boldsymbol{A},\boldsymbol{b_1})$

Sign(sk,μ)

1. $\boldsymbol{y}\leftarrow U(B_{(1/N)R,(k+l)}(B))$
2. $\boldsymbol{w}:=\boldsymbol{Ay}$
3. $\boldsymbol{w_1}:=High_{2q}(\boldsymbol{w},\alpha_h)$
4. $b,b'\leftarrow\{0,1\}$
5. $c:=H(High_{2q}(\boldsymbol{w},\alpha_h),LSB(\boldsymbol{y_o}),\mu)$
6. $\boldsymbol{z}:=(\boldsymbol{z_1},\boldsymbol{z_2})=\boldsymbol{y}+(-1)^b c\cdot\boldsymbol{s}$
7. $\boldsymbol{h}:=\boldsymbol{w_1}-High_{2q}(\boldsymbol{w}-2\boldsymbol{z_2},\alpha_h) \mod^+ \frac{2(q-1)}{\alpha_h}$
8. if $\|\boldsymbol{z}\|_2\geq B'$ then go to 1
9. else if $\|2\boldsymbol{z}-\boldsymbol{y}\|_2<B \wedge b'=0$ then go to 1
10. return
$\sigma=(Encode(High_q(\boldsymbol{z_1},256)),Low_q(\boldsymbol{z_1},256),$
$Encode(\boldsymbol{h}),c)$

Verify$(vk,\mu,\sigma=(x,\boldsymbol{v},h,c))$

1. $\widetilde{\boldsymbol{z_1}}=Decode(x)\cdot a+\boldsymbol{v}$ and $\tilde{\boldsymbol{h}}=Decode(h)$
2. $\boldsymbol{w_1}=\tilde{\boldsymbol{h}}+High_{2q}(\boldsymbol{A_1}\widetilde{\boldsymbol{z_1}}-qc\boldsymbol{j}) \mod^+ 2(q-1)/\alpha_h$
3. $w'=LSB(\widetilde{\boldsymbol{z_0}}-c)$
4. $\widetilde{\boldsymbol{z_2}}=\left[\boldsymbol{w_1}\cdot\alpha_h+w'\boldsymbol{j}-(\boldsymbol{A_1}\widetilde{\boldsymbol{z_1}}-qc\boldsymbol{j})\right]/2 \mod^\pm q$
5. $\tilde{\boldsymbol{z}}=(\widetilde{\boldsymbol{z_1}},\widetilde{\boldsymbol{z_2}})$
6. return $[c=H(\boldsymbol{w_1},w',\mu)]\wedge[\|\tilde{\boldsymbol{z}}\|<B'']$



g f

〔그림 1〕 HAETAE의 서명 rejection sampling에서 사용하는 bimodal hyperball 분포이다. g는 생성분포로 $cs,-cs$를 중심으로 반지름이 B인 두 개의 구로 구성되어 있고, f는 타겟분포로 O를 중심으로 반지름이 B'인 구이다.

$\| c\boldsymbol{s} \|_2$의 바운드를 줄일수록 서명의 크기가 줄어든다는 점에서, 키 생성과정에서 추가적인 비밀키의 rejection sampling 과정을 수행한다. [알고리즘 2] KeyGen 7과 같이 주어진 $\boldsymbol{s}$와 임의의 c에 대해 $\| c\boldsymbol{s} \|_2$가 충분히 작은지 확인한다. 이러한 비밀키 전처리 과정을 통해 조건을 만족하는 비밀키만 선택적으로 사용한다.

3.1.2. HAETAE 안전성 증명

HAETAE의 안전성 증명은 MLWE와 BimodalSelfTargetMSIS의 어려움에 의존하며, 증명 과정은 두 단계로 구성된다. 서명의 영지식성에 의해 서명 값이 난수 값과 구분이 불가능하다는 점을 이용하여 UF-CMA에서 UF-NMA로의 리덕션 관계[11][12]를 보인다. 공개키로부터 비밀키의 정보를 알아내지 못하는 것은 MLWE의 어려움에 의존하고 서명 위조 불가능성은 BimodalSelfTargetMSIS의 어려움에 의존하여, HAETAE의 UF-NMA 문제를 MLWE와 BimodalSelfTargetMSIS로 리덕션한다.

3.2. GCKSign[13]

GCKSign은 고려대학교와 상명대학교, 한성대학교, 국민대학교 소속 연구진이 개발한 격자 기반 전자서명 기법으로, 기존 기법[4]의 안전성 증명에서 요구되는 추가적인 조건을 제거하기 위해 Generalized Compact Knapsack (GCK) 함수의 신규 난제 Target-Modified One-wayness (TMO) 문제를 정의하고 이를 기반으로 설계되었다.

3.2.1. GCKSign 설계원리

GCKSign은 2009년에 제안된 Lyubashevsky[4]의 격자 기반 전자서명 기법과 동일한 형태로 설계되되어 Dilithium[2]보다 간단하다. 기법의 구성 요소가 다양해질수록 부채널 공격의 위험도가 높아진다. 따라서 가장 간단한 형태의 기법을 유지하면서 서명의 크기를 줄이고자 하였다. 기존의 기법[4]에 기반 난제를 새롭게 정의하여 적용함으로써 기존 기법의 안전성 증명에서 요구되는 추가적인 조건이 불필요하도록 설계하였다.

알고리즘 3. GCKSign

KeyGen(1^λ)

1. $\boldsymbol{a} = (a_1, \dots, a_m) \leftarrow R_q^m$
2. $\boldsymbol{s} = (s_1, \dots, s_m) \leftarrow S_\eta^m$
3. $t \leftarrow F_{\boldsymbol{a}}(\boldsymbol{s}) = \sum_{i=1}^{m} (a_i \cdot s_i)$
4. return $sk = (\boldsymbol{a}, \boldsymbol{s}), vk = (\boldsymbol{a}, t)$

Sign(sk, μ)

1. $\boldsymbol{y} \leftarrow S_B^m$
2. $v \leftarrow F_{\boldsymbol{a}}(\boldsymbol{y}) = \sum_{i=1}^{m} (a_i \cdot y_i)$
3. $c \leftarrow H(v, \mu)$
4. $\boldsymbol{z} := \boldsymbol{y} + \boldsymbol{s}c$
5. if $\boldsymbol{z} \notin S_{B-L_s}^m$, then go to 1
6. else return $\sigma = (\boldsymbol{z}, c)$

Verify(vk, μ, σ)

1. $w := F_{\boldsymbol{a}}(\boldsymbol{z}) - tc = \sum_{i=1}^{m} a_i z_i - tc$
2. if $\boldsymbol{z} \notin S_{B-L_s}^m \vee c \neq H(w, \mu)$, then return 0
3. else return 1

기존 기법[4]의 안전성 증명은 GCK 함수의 충돌쌍을 찾는 문제로 리덕션된다. 안전성 증명과정은 다음과 같다. 먼저, 임의로 생성한 비밀키 $\boldsymbol{s}$에 대하여 공개키 $t = F_{\boldsymbol{a}}(\boldsymbol{s})$를 생성하여 공격자에게 $(\boldsymbol{a}, t)$를 준다. rewind technique를 이용하여 공격자로부터 동일한 $y = F_{\boldsymbol{a}}(\boldsymbol{y})$에 대해 서로 다른 위조 서명 $(c, \boldsymbol{z})$와 $(c', \boldsymbol{z}')$을 얻는다. 이때 $y = F_{\boldsymbol{a}}(\boldsymbol{z}) - \boldsymbol{t}c = F_{\boldsymbol{a}}(\boldsymbol{z}') - \boldsymbol{t}c'$을 만족한다. $\boldsymbol{x} := \boldsymbol{z} - \boldsymbol{s}c, \boldsymbol{x}' := \boldsymbol{z}' - \boldsymbol{s}c'$이 같지 않다면 $\boldsymbol{x}$와 $\boldsymbol{x}'$은 GCK함수의 충돌쌍이 된다. $\boldsymbol{x}$와 $\boldsymbol{x}'$이 같지 않기 위해서는 실제 비밀키 $\boldsymbol{s}$와 공격자가 서명 위조시 사용한 비밀키 $\boldsymbol{s}'$이 달라야 한다. $\boldsymbol{s}$와 $\boldsymbol{s}'$이 높은 확률로 다르기 위해서는 비밀키 집합 내에서 GCK함수의 충돌쌍이 많아야 하므로 비밀키가 다음의 식 (3)을 만족해야 하며 그때 $\boldsymbol{s} \neq \boldsymbol{s}'$일 확률은 $1 - 1/2^{128}$이다. 따라서 비밀키 계수의 범위인 η를 큰 값으로 정의해야하고 비밀키의 크기가 커짐에 따라 서명의 크기도 증가하게 된다.

$$2^{128} \cdot q^n < (2\eta + 1)^{mn} \tag{3}$$

한편, 새로운 난제인 GCK-TMO문제로 리덕션하면

식 (3)을 만족하지 않는 범위의 비밀키를 이용하여도 증명에 문제가 없다. 위의 조건이 불필요해지므로 η를 줄임으로써 간단한 기법[4]을 유지하며 서명의 크기를 감소시켰다.

3.2.2. GCKSign 안전성 증명

GCK-TMO 문제는 [알고리즘 4]와 같이 정의된다. 안전성 증명에서 rewind technique를 이용하여 두 개의 위조 서명 $(\boldsymbol{z},\boldsymbol{c}),(\boldsymbol{z'},\boldsymbol{c'})$을 생성하였을 때 $y=F_{\boldsymbol{a}}(\boldsymbol{z})-tc=F_{\boldsymbol{a}}(\boldsymbol{z'})-tc'$을 만족한다. 이때 $\boldsymbol{z}\neq\boldsymbol{z'}$, $\boldsymbol{c}\neq\boldsymbol{c'}$이므로 $F_{\boldsymbol{a}}(\boldsymbol{z}-\boldsymbol{z'})=(c-c')t$이다. $\boldsymbol{x}=\boldsymbol{z}-\boldsymbol{z'}$, $\tilde{c}=c-c$는 $\alpha=2, \beta=2(B-L_s)$에 대하여 GCK-TMO 문제의 답이 된다. 즉, GCKSign의 EUF-CMA문제는 GCK-TMO문제로 리덕션이 가능하다.

새로운 난제를 정의하였을 때, 기존의 난제 GCK-OW (Onewayness), GCK-CR (Collision Resistance) 문제와의 리덕션 관계를 보임으로써 새로운 난제의 어려움을 정의할 수 있다. 해당 기법에서는 GCK-TMO 문제가 GCK-CR 문제로 리덕션됨을 보였다. 먼저 $\|\boldsymbol{x}c^{-1}\|_\infty \leq \gamma$ $(n\alpha\gamma \leq \beta)$인 경우 $F_{\boldsymbol{a}}(\boldsymbol{x}c^{-1})=t$이므로 $GCK-OW_{n,m,\gamma}$로 리덕션 하였다. $\gamma<\beta$이므로, $GCK-OW_{n,m,\beta}$로 리덕션 하고, 최종적으로는 $GCK-CR_{n,m,\beta}$로 리덕션 관계를 보였다. 반면, $\|\boldsymbol{x}c^{-1}\|_\infty>\gamma$일 경우 $GCK-CR_{n,m,\beta}$로 리덕션이 되어 결론적으로 $GCK-TMO$문제는 $GCK-CR_{n,m,\beta}$문제의 어려움에 기반함을 보였다.

알고리즘 4. GCK-TMO 문제
Given $\boldsymbol{a}=(a_1,\ldots,a_m)\in R^m$ and $t\in R$,
find $\boldsymbol{x}\in R_q^m, c\in R_q$
s.t $\|c\|_\infty \leq \alpha$, $\|\boldsymbol{x}\|_\infty \leq \beta$, and $F_{\boldsymbol{a}}(\boldsymbol{x})=c\cdot t$

3.3. NCC-Sign[5]

NCC-Sign은 국가 수리 과학 연구소의 암호기술연구팀 소속 연구진이 개발한 격자 기반 전자서명 기법으로, Dilithium과 동일한 기법에 대하여 non-cyclotomic ring을 적용하여 부채널 공격 관점에서 안전하도록 설계되었다.

3.3.1. NCC-Sign 설계원리

기존의 암호 기법들은 주로 $n=2^k$를 만족하는 $R_q=Z_q[x]/<x^n+1>$에서 혹은 소수 p에 대해 $R_q=Z_q[x]/<x^p-1>$에서 설계되었다. 즉, 대부분 cyclotomic polynomial을 사용하였는데 이는 체 $Q[x]/\phi(x)$의 부분체[14][15], 작은 갈루아 그룹[16]과 $Z_q[x]/\phi(x)$에서의 환 준동형사상 (ring homomorphism)과 같은 다양한 대수적 구조를 가지고 있다. 또한, $n=2^k$인 cyclotomic polynomial에서는 Number Theoretic Transform (NTT)를 사용할 수 있어 효율적인 다항식 곱 연산이 가능하다. 그러나 이러한 장점들은 부채널 공격 관점에서 공격의 주요 대상이다.

NCC-Sign에서는 소수 p에 대하여 p차 갈루아 그룹의 non-cyclotomic $\phi(x)=x^p-x-1$과 inert modulus q를 사용하는 NTRU Prime field[17]를 사용하여 기법을 설계하였다. $n=2^k$를 사용하지 않음으로써 파라미터 선택의 폭이 넓어졌고 Dilithium의 반복 기대 횟수와 동일하도록 파라미터를 세밀하게 조정하였다. 또한, NTT를 사용하지 못하므로 Toom-Cook and Karatsuba의 다항식 곱셈 연산을 사용하였다.

NCC-Sign 알고리즘은 Module 구조에서 Ring 구조로 변경된 것 이외에 Dilithium의 알고리즘 [알고리즘 1]과 동일하므로 생략하였다.

3.3.2. NCC-Sign 안전성 증명

NCC-Sign은 공개키로부터 비밀키의 정보가 노출되지 않는 것은 RLWE의 어려움에 근거하고 서명 위조 불가능성은 SelfTargetRSIS의 어려움에 근거한다. Dilithium과 동일한 증명구조로 UF-CMA는 UF-NMA로 리덕션이 가능하고 UF-NMA는 RLWE와 SelfTargetRSIS로 리덕션 된다.

Ⅳ. KpqC 전자서명 기법 비교 및 분석

이 장에서는 앞서 소개한 KpqC 전자서명 기법들에 대해 고찰하여 본다.

4.1. HAETAE

HAETAE는 bimodal hyperball 분포를 도입하여 공개키와 서명의 크기를 줄이고자 하였다. bimodal 형태를 유지하기 위해 challenge c의 계수가 {-1,0,1}로 구성된 Dilithium과 달리 {0,1}로 이루어져 있다. 해시함수의 entropy가 충분히 확보되지 않음에 따라 c의 해밍 웨이트 (hamming weight)가 기존의 Fiat-Shamir with aborts 기반 기법들보다 커졌다. 그로 인해 서명의 크기가 증가하였고 비밀키의 rejection sampling을 통해 미리 정해놓은 $\|\boldsymbol{s}c\|_2$의 바운드를 만족하는 비밀키만 사용함으로써 서명의 크기를 줄였다. 그러나 키 생성과정에서 비밀키에 대한 rejection sampling을 수행하면서 키 생성이 느려졌다.

서명 생성과정에서는 난수 $\boldsymbol{y}$를 hyperball에서 sampling 하는데 많은 시간이 소요되고 있다. discrete hyperball uniform sampling을 하기 위해서는 continous hyperball uniform sampling을 수행해야 하고, continous hyperball uniform sampling을 위해 가우시안 sampling을 수행해야 한다. 즉, 난수 $\boldsymbol{y}$의 모든 원소에 대한 계수들을 가우시안 sampling으로 선택해 주어야 하므로 discrete hyperball uniform sampling의 연산량이 많다.

현재 안전성 증명은 HAETAE의 일부 저자들이 최근 제안한 논문[12]과 유사한 논리로 증명하였다. QROM에서 UF-CMA와 UF-NMA의 리덕션 관계를 보이기 위해 commitment $\boldsymbol{y}$의 min-entropy가 높다는 것과, 서명 기법을 interactive 프로토콜로 변경하여 시뮬레이션 해줄 때 witness $\boldsymbol{w}$,challenge c, response $\boldsymbol{z}$의 분포가 비밀키를 이용하여 정직하게 생성한 $(\boldsymbol{w},c,\boldsymbol{z})$의 분포가 다르지 않음을 이용하여 증명한다. 해당 논문[12]에서 제안하는 정리의 조건을 만족시킴에 따라 HAETAE의 서명 위조에 대한 안전성을 증명하고자 하였다. 안전성 증명과정이 일부만 제시되어 있어, 향후 추가되는 안전성 증명에서 일부의 비밀키만 사용하는 것으로부터 서명 쿼리의 응답 $(\boldsymbol{w},c,\boldsymbol{z})$의 분포와 실제 $(\boldsymbol{w},c,\boldsymbol{z})$의 분포에 대해 유의미한 차이 발생하지 않음이 논의되어야 한다.

4.2. GCKSign

GCKSign은 2009년도에 Lyubashevsky[4]의 격자 기반 전자서명 기법에서 기반 난제를 GCK-TMO로 변경하여 안전성 증명에서 필요한 추가적인 조건을 제거하였다. 그러나 새롭게 정의한 난제 GCK-TMO와 기존의 난제 GCK-CR간의 리덕션 관계를 증명하는 단계에서 오류가 발생하여 안전성 증명의 수정이 필요하다.

GCK-TMO 문제의 안전성 증명과정에서 $GCK-OW_{n,m,\beta}$는 $GCK-OW_{n,m,\gamma}$로 리덕션 됨을 이용하였다. 주어진 $\boldsymbol{a}\in R^m$과 $t\in R$에 대하여 $F_{\boldsymbol{a}}(\boldsymbol{x})=t$와 $\|\boldsymbol{x}\|_\infty \leqq \gamma$를 만족하는 $\boldsymbol{x}$를 찾는 것이 $GCK-OW_{n,m,\gamma}$문제이다. $GCK-OW_{n,m,\gamma}$의 해답 $\boldsymbol{x}$는 $n\alpha\gamma \leqq \beta$에 대하여 $\|\boldsymbol{x}\|_\infty \leqq \beta$를 만족하므로, $GCK-OW_{n,m,\beta}$문제의 답이 될 수 있다. 따라서 $GCK-OW_{n,m,\beta}$에서 $GCK-OW_{n,m,\gamma}$로 리덕션 관계가 성립할 것으로 보일 수 있다. $(2\beta+1)^{nm} \gg q^n$을 만족하여 $GCK-OW_{n,m,\beta}$문제에서는 답이 항상 존재한다. 그러나 $GCK-OW_{n,m,\gamma}$에서는 $(2\gamma+1)^{nm}$가 q^n보다 훨씬 작기 때문에 low-density SIS 문제가 되어 답이 존재하지 않는 경우도 있다. $GCK-OW_{n,m,\beta}$에서는 답이 있던 문제가 $GCK-OW_{n,m,\gamma}$에서는 답이 없는 문제가 되는 경우를 고려해야 하므로, 위와 같은 리덕션 관계는 성립하지 못한다. 따라서 $GCK-TMO$ 문제는 $GCK-OW_{n,m,\gamma}$ 와 $GCK-CR_{n,m,\beta}$ 문제 모두로 리덕션 되어야 할 것이다. 기존의 기반 난제였던 RSIS에 RLWE가 추가됨에 따라 SIS와 LWE 문제 모두에 대해 security bits를 만족시켜야 한다. 현재 GCKSign이 제안한 parameter set Ⅱ, Ⅲ, Ⅴ에 대하여 LWE 문제에 대한 security bits는 65, 56, 121 bits[21]이다. NIST에서 규정한 security level을 만족해야 하므로 파라미터 수정이 불가피할 것으로 예상된다. 따라서 Dilithium과 KpqC 기법들을 비교하는 4.4장에서는 제외하였다.

4.3. NCC-Sign

NCC-Sign은 non-cyclotomic polynomial을 이용하여 cyclotomic의 불필요한 대수적 구조를 제거하고 NTT를 사용하지 않음으로써 부채널 공격의 위험성을 낮추었다. 효율적인 다항식 곱셈 연산을 위해 Toom-Cook 알고리즘을 사용하여 속도 저하를 최소화하였으나 여전히 Dilithium과 HAETAE에 비해 속도가 느리다는 점에서 한계가 있다.

〔표 1〕 파라미터 및 공개키와 서명 크기(bytes) 비교표

Security Level	파라미터 크기	Dilithium	HAETAE	NCC-Sign
Ⅱ	q	8380417	**64513**	8339581
	$\|\| vk \|\|$	1312	**992**	1564
	$\|\| \sigma \|\|$	2420	**1463**	2458
	$\|\| vk \|\| + \|\| \sigma \|\|$	3732	**2455**	4022
Ⅲ	q	8380417	**64513**	8376649
	$\|\| vk \|\|$	1952	**1472**	1997
	$\|\| \sigma \|\|$	3293	**2337**	3605
	$\|\| vk \|\| + \|\| \sigma \|\|$	5245	**3809**	5602
Ⅴ	q	8380417	**64513**	8343469
	$\|\| vk \|\|$	2592	**2080**	2663
	$\|\| \sigma \|\|$	4595	**2908**	5055
	$\|\| vk \|\| + \|\| \sigma \|\|$	7187	**4988**	7718

NTRU Prime Field를 사용함으로써 불필요한 대수적 구조를 줄여 부채널 공격의 위험성을 낮추었다. 그러나 NTRU Prime KEM과 달리 서명 생성 과정에서 rejection sampling을 수행해야 하므로p, q가 NTRU Prime KEM의 p, q보다 크며 modulus q는 약 23비트로 Dilithium과 유사하다. 또한, non-cyclotomic polynomial을 사용하여 서명과 키의 크기가 크다.

한편, NCC-Sign에서는 NTT를 Toom-Cook and Karatsuba 알고리즘으로 대체하여 다항식의 곱셈 연산을 수행한다. 부채널 공격 관점에서 NTT가 공격의 대상이 된다는 점에서 Toom-Cook and Karatsuba 알고리즘으로 대체하면 NTT에 대한 부채널 공격에 대한 위험성이 감소한다. 그러나 Toom-Cook에 대한 부채널 공격의 위험성[18][19] 역시 존재하므로 안전성에 대해 고려해봐야 할 것이다.

Dilithium과 동일한 기법을 사용하고 환의 구조만 변경되었다는 점에서 Dilithium과 유사도가 높다.

4.4. 전체 비교

HAETAE와 GCKSign, NCC-Sign, Dilithium의 security bits 측정 방법은 모두 유사하다. 각 기법의 안전성은 LWE와 SIS 문제로 리덕션이 된다. primal attack, dual attack, SIS attack에 대해 BKZ-b 알고리즘을 이용하여 SVP 문제를 푸는 비용으로 security bits를 측정하였다.

[표 1]은 NIST security level Ⅱ, Ⅲ, Ⅴ에 대하여 modulus q와 bytes를 단위로 한 공개키와 서명 크기 비교표이다. 공개키와 서명의 크기는 HAETAE가 가장 작다. HAETAE는 uniform bimodal hyperball distribution을 채택하여 작은 modulus를 사용할 수 있고 rANS encoding 기법을 사용하면서 공개키와 서명의 크기가 감소하였다. 반면, NCC-Sign은 non-cyclotomic polynomial을 사용하면서 rejection 확률을 Dilithium과 유사하게 유지하기 위해 modulus q를 Dilithium과 유사한 약 23 bits로 설정하였고 그 결과 Dilithium과 공개키, 서명의 크기가 유사하다.

[표 2]는 각 기법의 키 생성, 서명 생성, 서명 검증 과정을 1,000회 진행했을 때의 평균 cycle 횟수로 단위는 k cycle이다. cycle 측정에 사용한 프로세서의 사양은 Intel(R) Core(TM) i7-8,700K CPU @ 3.70GHz 16.0GB이다. 3개의 기법 중 Dilithium의 cycle 수가 가장 적다. HAETAE는 비밀키의 rejection sampling 과정과 난수 $\boldsymbol{y}$의 hyperball sampling 과정으로 인해 많은 시간이 소요되었다. NCC-Sign의 경우 non-cyclotomic polynomial을 이용함에 따라 속도 저하가 발생하였다. HAETAE와 NCC-Sign이 구현 최적화 작업이 아직 수행되지 않은 점을 고려했을 때, 향후 최적화된 구현 코드로 정확한 비교가 가능할 것이다.

〔표 2〕 Cycle 비교표 (k cycle)

Security Level	알고리즘	Dilithium	HAETAE	NCC-Sign
Ⅱ	KeyGen	**265**	2,348	2,472
	Sign	**1,221**	17,546	41,542
	Verify	**285**	3,764	4,770
Ⅲ	KeyGen	**266**	5,265	4,842
	Sign	**1,228**	18,725	73,429
	Verify	**286**	7,052	9,667
Ⅴ	KeyGen	**275**	8,200	8,313
	Sign	**1,252**	50,777	122,087
	Verify	**286**	10,746	16,445

Ⅴ. 결 론

KpqC 1 round에 제안된 Fiat-Shamir with aborts paradigm 기반의 전자서명 기법 HAETAE와 GCKSign, NCC-Sign은 기존의 기법을 일부 수정하여 효율적이고 작은 크기의 서명 생성을 추구하였다.

HAETAE는 bimodal hyperball distribution을 도입하여 서명의 크기를 줄였으나 hyperball sampling으로 인하여 서명생성 시간이 Dilithium보다 길었다. 또한, 비밀키를 선택적으로 사용함으로써 키생성 시간이 길고 안전성 증명에서 영지식성 만족 여부에 대한 논의가 필요하다. GCKSign은 기존의 간단한 기법[4]에서 기반 난제를 GCK-TMO로 변경하여 안전성 증명에서 필요한 추가적인 조건을 제거하고자 하였으나, 새롭게 정의한 난제 GCK-TMO와 기존의 난제 간의 리덕션 관계를 증명하는 단계에서 오류가 발생하여 수정이 필요하다. NCC-Sign은 non-cyclotomic polynomial을 이용하여 불필요한 대수적 구조를 제거하고 NTT를 사용하지 않음으로써 부채널 공격의 위험성을 낮추었으나 Dilithium과 HAETAE에 비해 속도가 느리고 Dilithium과 유사도가 높다는 점에서 한계가 있다.

GCKSign은 안전성 증명과정에서 rewind technique을 사용하고 있어 QROM에서의 안전성을 보이지 못했다. HAETAE의 안전성 증명은 SelfTargetMSIS를 기반으로 하며, NCC-Sign은 Dilithium과 동일한 논리로 SelfTargetRSIS에 기반하고 있다. SelfTargetMSIS (RSIS)에서 MSIS (RSIS)로의 리덕션 관계가 QROM에서는 증명되지 않았기 때문에 HAETAE와 NCC-Sign 역시 QROM에서의 안전성을 엄밀히 보이지 못하였다. 세 기법 모두 QROM에서의 안전성 증명이 보완될 필요가 있다.

참 고 문 헌

[1] Shor, P. W. “Algorithms for quantum computation: discrete logarithms and factoring.”, In Proceedings 35th annual symposium on foundations of computer science (pp. 124-134). IEEE, 1994.

[2] Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D. “Crystals-dilithium: A lattice-based digital signature scheme.” IACR Transactions on Cryptographic Hardware and Embedded Systems, 238-268. 2018

[3] Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, Zhenfei Zhang. "Falcon: Fast-Fourier lattice-based compact signatures over NTRU." Submission to the NIST’s post-quantum cryptography standardization process 36.5 2018.

[4] Vadim Lyubashevsky. “Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures.”, In Mitsuru Matsui, editor, Advances in Cryptology –

ASIACRYPT, pages 598–616. Springer, 2009.

[5] Kyung-Ah Shim, Jeongsu Kim, Youngjoo An. “NCC-Sign:A New Lattice-based Signature Scheme using Non-Cyclotomic Polynomials”, Submission to the KpqC 1 round, 2022.

[6] Vadim Lyubashevsky. “Lattice signatures without trapdoors.”, In David Pointcheval and Thomas Johansson, editors, Advances in Cryptology – EUROCRYPT, pages 738–755. Springer, 2012.

[7] Jung Hee Cheon, Hyeongmin Choe, Julien Devevey, Tim Güneysu, Dongyeon Hong, Markus Krausz, Georg Land, Marc Möller, Damien Stehlé, MinJune Yi. “HAETAE: Shorter Lattie-Based Fiat-Shamir Signatures”, Cryptology ePrint Archive, Paper 2023/624, 2023. https://eprint.iacr.org/2023/624

[8] Julien Devevey, Omar Fawzi, Alain Passel`egue, and Damien Stehlé. “On rejection sampling in lyubashevsky’s signature scheme.”, Cryptology ePrint Archive, Number 2022/1249, 2022.

[9] Léo Ducas, Alain Durmus, Tancr`ede Lepoint, and Vadim Lyubashevsky. “Lattice signatures and bimodal gaussians.”, In Ran Canetti and Juan A. Garay, editors, Advances in Cryptology – CRYPTO, pages 40–56. Springer, 2013.

[10] Jarek Duda. “Asymmetric numeral systems: entropy coding combining speed of huffman coding with compression rate of arithmetic coding”, Cryptology ePrint Archive, Paper 2013. https://arxiv.org/abs/1311.2540.

[11] Eike Kiltz, Vadim Lyubashevsky, and Christian Schaffner. “A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model.”, In Advances in Cryptology – EUROCRYPT, pages 552–586. Springer, 2018.

[12] Julien Devevey, Pouria Fallahpour, Alain Passel`egue, Damien Stehlé. “A detailed analysis of Fiat-Shamir with aborts.”, Cryptology ePrint Archive, Paper 2023/245, 2023. https://eprint.iacr.org/2023/245.

[13] Woo Joo, Kwangsu Lee, Jong Hwan Park. “GCKSign: Simple and Efficient Signatures from Generalized Compact Knapsacks”, Cryptology ePrint Archive, Paper 2022/1665, 2022. https://eprint.iacr.org/2022/1665

[14] Bauch, J., Bernstein, D.J., Valence, H.d., Lange, T., Vredendaal, C.v. “Short gen┐erators without quantum computers: the case of multiquadratics.”, In: Annual Inter┐national Conference on the Theory and Applications of Cryptographic Techniques. pp. 27–59. Springer ,2017.

[15] Albrecht, M., Bai, S., Ducas, L. “A subfield lattice attack on overstretched NTRU assumptions.”, In: Annual International Cryptology Conference. pp. 153–178. Springer, 2016.

[16] Campbell, P., Groves, M., Shepherd, D.: Soliloquy. “A cautionary tale.” In: ETSI 2nd Quantum-Safe Crypto Workshop. vol. 3, pp. 1–9, 2014.

[17] Bernstein, D.J., Chuengsatiansup, C., Lange, T., Vredendaal, C.v. “NTRU prime: reducing attack surface at low cost.”, In: International Conference on Selected Areas in Cryptography. pp. 235–260. Springer, 2017

[18] Li, Y., Zhu, J., Huang, Y., Liu, Z., Tang, M. “Single-Trace Side-Channel Attacks on the Toom-Cook: The Case Study of

Saber”,. IACR Transactions on Cryptographic Hardware and Embedded Systems, 285–310. 2022.
[19] Catinca Mujdei, Arthur Beckers, Jose Maria Bermudo Mera, Angshuman Karmakar, Lennert Wouters, Ingrid Verbauwhede. “Side-Channel Analysis of Lattice-Based Post-Quantum Cryptography: Exploiting Polynomial Multiplication”, ACM Transactions on Embedded Computing Systems, 2022.
[20] Shi Bai, Steven D. Galbraith. “An improved compression technique for signatures based on learning with errors.”, In CT-RSA, pages 28–47, 2014.
[21] Minkyu Kim, Han Sol Ryu, Ho Chang Lee. “KpqC-bulletin board: Analysis of GCKSign”, 2022.

〈저 자 소 개〉

홍 가 희 (Ga Hee Hong)
2023년 2월 : 성신여자대학교 수학과 졸업
2023년 3월~현재 : 고려대학교 정보보호대학원 석사과정
<관심분야> 양자내성암호, 전자서명

우 주 (Joo Woo)
2017년 2월 : 고려대학교 임상병리학과 졸업
2019년 8월 : 고려대학교 정보보호대학원 석사
2019년 9월~현재 : 고려대학교 정보보호대학원 박사과정
<관심분야> 함수암호, 양자내성암호,

박 종 환 (Jong Hwan Park)
1999년 2월 : 고려대학교 수학과 졸업
2005년 2월 : 고려대학교 정보보호학과 석사
2008년 8월 : 고려대학교 정보보호학과 박사
2013년 9월~2019년 8월 : 상명대학교 컴퓨터과학과 조교수
2019년 9월~현재 : 상명대학교 컴퓨터과학과 부교수
<관심분야> 함수암호, 양자내성암호, 영지식 증명