Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.31, NO.6, Dec. 2021 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2021.31.6.1137

CSIDH 성능 향상을 위한 Radical Isogeny 적용 분석*

김 수 리†‡
성신여자대학교 (조교수)

On the Use of Radical Isogenies for CSIDH Implementation*

Suhri Kim†‡
Sungshin Women's University (Assistant Professor)

요 약

CSIDH 기반 암호를 구현하는 데 있어서 가장 큰 단점은 Velu 공식을 활용하여 isogeny를 연산하기 위해 작은 소수 위수를 가지는 커널의 생성점을 선택하는 부분이다. 이 과정은 작은 위수의 경우 실패확률이 크기 때문에 연산량이 많이 들어서, 최근에 radical isogeny를 사용하는 부분에 관한 연구가 진행되었다. 본 논문에서는 CSIDH 기반 암호 구현에 있어서 radical isogeny를 사용하는 최적 방안에 대해 제시한다. 본 논문에서는 Montgomery 곡선과 Tate 곡선 사이의 변환을 최적하였으며, 2-, 3-, 5-, 7-isogeny에 대한 공식을 최적화하였다. 본 논문의 결과, CSIDH-512의 경우 radical isogeny를 7차까지 사용할 경우 기존 constant-time CSIDH에 비해서 15.3% 빠른 결과를 얻을 수 있었다. CSIDH-4096의 경우 radical isogeny를 2차까지 사용하는 것이 최적이라는 결론을 얻을 수 있었다.

ABSTRACT

The main obstacle for implementing CSIDH-based cryptography is that it requires generating a kernel of a small prime order to compute the group action using Velu's formula. As this is a quite painstaking process for small torsion points, a new approach called radical isogeny is recently proposed to compute chains of isogenies from a coefficient of an elliptic curve. This paper presents an optimized implementation of radical isogenies and analyzes its ideal use in CSIDH-based cryptography. We tailor the formula for transforming Montgomery curves and Tate normal form and further optimized the radical 2- and 3- isogeny formula and a projective version of radical 5- and 7- isogeny. For CSIDH-512, using radical isogeny of degree up to 7 is 15.3% faster than standard constant-time CSIDH. For CSIDH-4096, using only radical 2-isogeny is the optimal choice.

Keywords: Post-quantum cryptography CSIDH, Velu's formula, isogeny-based cryptography, radical isogeny

I. 서 론

양자 컴퓨터의 개발이 가시화 되면서 양자 컴퓨팅 능력을 가진 공격자의 공격에 대응하기 위한 공개키 암호에 관한 연구가 활발히 진행되고 있다. 최근 2018년 인텔은 49-qubit으로 이루어진 프로세서를 제안하였으며, 그해 구글은 72-qubit Bristlecone을 제안하였다. 최근 2020년 IBM는 Hummingbird라는

Received(10. 05. 2021), Modified(11. 16. 2021), Accepted(11. 16. 2021)
* 이 논문은 2021년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No. 2021-0-00518, 블록체인 데이터 암호화 기반의 프라이버시 보호 기술개발)
† 주저자, suhrikim@sungshin.ac.kr
‡ 교신저자, suhrikim@sungshin.ac.kr(Corresponding author)

65-qubit 프로세서를 제안하면서 양자 컴퓨터의 개발 속도가 가속화되고 있다. 만약 Shor 알고리즘이 양자 컴퓨터에 구현될 경우 더 이상 현재 사용하고 있는 RSA나 ECC와 같은 암호를 사용할 수 없으며, 이에 따라 고전 컴퓨터에서도 구현할 수 있지만 양자 컴퓨팅 능력을 갖춘 공격자에 안전한 양자 내성 암호 (후 양자 암호, post-quantum cryptography, PQC)와 관련된 연구가 활발히 진행되고 있다. 2016년 NIST의 PQC 공모전을 시작으로 현재는 Round 3단계에 있으며, 이 중 isogeny 기반 암호는 다른 PQC 암호에 비해 작은 키 사이즈로 주목받고 있다.

Isogeny 기반 암호는 Couveignes에 의해 처음으로 제안되었으며 후에 Rostovtsev와 Stolbunov에 의해 확장되어 현재는 CRS 라 한다 [10, 23]. 하지만, CRS 는 ordinary 타원곡선을 사용하기 때문에 endomorphism이 가환성을 가지고 있으며, 이를 이용한 양자 하지수시간의 공격이 존재한다 [7]. 하지만 더 큰 문제는 ordinary 타원곡선 사용으로 인해 파라미터 선택이나 효율적인 연산이 불가능해 실제 사용하기에 속도가 매우 느리다는 점에 있다. Isogeny 기반 암호는 후에 De Feo와 Jao 가 제안한 SIDH에 의해 다시 주목받기 시작했다 [15]. Ordinary 곡선을 사용하는 CRS와 달리 SIDH는 supersingular 타원곡선을 사용하기 때문에, endomorphism ring이 비가환성이여서, endomorphism ring의 가환성을 이용하는 [7]의 공격에 대응할 수 있을 뿐만 아니라, 훨씬 더 효율적인 성능을 제공한다. 현재까지고 SIDH에 대한 양자 공격은 지수시간의 복잡도를 가지고 있다. 한편, SIDH를 기반으로 둔 SIKE는 현재 NIST PQC 표준화 공모전의 Round 3의 대체후보이다.

한편, CRS 알고리즘 느리다는 단점이 있지만, non-interactive 키 교환 알고리즘 설계하는 데 적합하다는 장점이 있다. SIDH 경우 비가환적인 성질로 인해서 자신의 비밀 isogeny로 상대방의 공개키에 대한 값을 연산해서 전달해야 하는 부분이 존재한다. 비록 이 부분이 아직은 SIDH에 유의미한 공격으로 작용하지 않지만, 비밀값이 누출될 수 있다는 점이 단점으로 작용한다. 하지만 CSIDH의 경우 가환적인 성질을 가지기 때문에, 기존 Diffie-Hellman과 유사하게 별도의 부가정보 없이 자신의 연산값만 상대방에게 전달해주면 된다는 장점이 있다. 따라서 최근 CRS를 최적화하려는 연구가 활발히 진행되고 있으며, 특히 Castryck 등이 제안한 CSIDH로 다시 주목받기 시작했다 [5]. Castryck 등은 기존 CRS가 ordinary 곡선을 사용하면서 발생하는 문제를 supersingular 곡선을 사용하므로 해결하여 CSIDH (Commutative SIDH)를 제안하였다. CSIDH 키 교환 알고리즘은 4개의 group action으로 구성되어 있고, 한 group action의 평균 속도는 35ms 정도로 SIDH에 비해서 느리지만, 기존 isogeny 기반 암호의 단점 중 하나인 효율적인 전자 서명 알고리즘 부재를 해결해줌으로써 많이 연구되고 있다[3,16].

SIDH와 CSIDH 기반 암호의 공통된 장점으로는, 다른 PQC 기반 암호들보다 키 사이즈가 작다는 점이다. 하지만 32비트 이내의 행렬-벡터 연산으로 구성된 다른 PQC 기반 암호와 달리, isogeny 기반 암호는 400비트 이상의 큰 유한체에서 타원곡선 연산으로 구성되어있기 때문에, 다른 PQC 암호에 비해 느리다는 단점을 가진다. 따라서, isogeny 기반 암호의 최적화를 위해 많은 연구가 진행되어왔다. Isogeny 기반 암호 최적화 연구 중 한 갈래는, isogeny 연산 자체에 관한 최적화로, 이 경우 isogeny 연산이 빠른 다른 형태의 타원곡선을 사용하거나, 새로운 isogeny 공식을 연구하는 방향이 있다. [17,19] 에서는, 빠른 연산을 위해 Montgomery 곡선과 Edwards 곡선을 사용하는 hybrid 방식을 제안한다. Isogeny 연산 최적화로는 최근에 Bernstein 등이 [2]에서 n-isogeny를 연산하는데 기존 $O(n)$의 연산량에서 $O(\sqrt{n})$의 연산량으로 최적화하는 방안에 대해 제시하였다. Isogeny 기반 암호를 최적화하는 다른 갈래로는, 기존 scheme 구현이 빠르도록 변형시키는 방법이다. [8]에서 제안된 B-SIDH는 Alice가 $(p+1)$-torsion subgroup에서 연산하고, Bob이 $(p-1)$-torsion subgroup에서 연산하는 방안에 대해서 제시한다. B-SIDH는 SIDH에 대한 변형이라고 생각할 수 있으며, Alice 쪽에서 isogeny 연산을 SIDH 보다 감산 연산이 효율적인 유한체 위에서 수행할 수 있다는 장점을 가진다. CSIDH 기반 암호에 대해서는 [4]에서 제안된 CSURF가 있으며, 이는 floor에서 정의된 supersingular 곡선을 사용하는 기존 CSIDH과 달리 효율적인 2-isogeny 연산을 위해 surface에서 정의된 supersingular 곡선을 사용한다.

CSURF는 $p \equiv 7 \bmod 8$에서 endomorphism ring이 $\mathbb{Z}[(1+\sqrt{-p})/2]$인 supersingular 곡선을 사용한다. [4]에서는 이러한 supersingular 곡선은 tweaked Montgomery 곡선 (Montgomery- 곡선)으로 표현되고 있고, 이 곡선에서 타원곡선 연산 식

은 Montgomery 곡선과 유사하다. $p \equiv 7 \bmod 8$ 을 만족하는 유한체 F_p 위에서 소수 2는 $Q(\sqrt{-p})$에서 분해될 수 있어 수평적 2-isogeny 연산을 가능하게 한다. 2-isogeny의 연산은 거의 유한체 위의 지수연산 1번으로 구성되어있다고 볼 수 있어서, 기존 CSIDH의 개인키의 지수의 범위를 적절히 조절하면 더 효율적인 연산을 수행할 수 있다. [12]에서는 projective coordinate을 사용할 경우 CSURF가 기존 CSIDH보다 성능이 좋지 않다는 결과를 보였으나, 2-isogeny의 사용은 radical isogeny의 연구로 이어지게 되었다 [24].

CSIDH 기반 암호는 다양한 차수의 isogeny 연산이 필요하며, 이를 위해서 다양한 위수를 가지는 타원곡선 위의 점을 선택할 필요가 있다. 주어진 위수의 커널을 성성하기 위해서는 F_p에서 랜덤한 타원곡선 위의 점 Q를 선택하고 cofactor k에 대해서 scalar multiplication $[k]Q$ 연산을 수행한다. 랜덤한 타원곡선 위의 점을 선택하는데 대략 $1.5\log p$의 유한체 곱셈이 필요하고, $[k]Q$연산을 수행하는데 CSIDH 기반 암호에 대해서는 대략 $11\log p$연산이 필요하다. 만약 연산결과 $P=[k]Q$가 무한원점일 경우에 다른 랜덤한 점을 생성하고 위 연산을 반복한다. 따라서 특정 위수의 커널을 생성하는 것은 많은 연산량이 요구되며, 특히 위수가 작을 경우 실패 확률이 커지게 된다. 따라서 [24]에서는 radical isogeny라는 새로운 isogeny 연산 방법을 제안한다. Radical isogeny는 n^e-isogeny를 연산하는 데 기존 e번의 위수가 n인 점을 요구로 하는 방법과 달리, 한 개의 n-torsion 점으로 isogeny를 연산하는 방법을 제안한다. CSURF와 유사하게, radical isogeny를 사용하는 소수의 지수의 범위를 크게 하고 radical isogeny를 사용하지 않는 소수의 지수범위는 작게 하여 전반적인 알고리즘에서 랜덤한 점을 선택하는 횟수를 줄일 수 있다.

본 논문에서는 CSIDH 기반 암호 구현에 있어서 효율적인 radical isogeny 사용 방안에 대해 제안한다. 본 논문의 기여는 다음과 같이 정리할 수 있다.

- 본 논문에서는 기존 제안된 radical isogeny 공식을 최적화하였다. 3-isogeny의 경우 Onuki와 Moriya 가 제안한 방법을 추가적으로 최적화를 진행하였으며, 5-, 7-isogeny의 경우 Chi-Domiguez와 Reijnders가 제안한 방법을 추가적으로 최적화하였다.
- 본 논문에서는 radical isogeny를 이용한 CSIDH 구현결과를 제시한다. 128비트 고전 보안강도를 가지는 CSIDH-512의 경우 radical isogeny를 최대 7차까지 사용했을 경우 기존 CSIDH 보다 15.3% 빠르다. 128비트 양자 보안강도를 가지는 CSIDH-4096의 경우 radical isogeny는 최대 2차까지 사용하는 것이 최적이라는 결론을 얻었다.

본 논문은 다음과 같이 구성되어있다. 2장에서는 본 논문의 기반이 되는 개념에 관해 설명한다. 3장에서는 radical isogeny와 본 논문에서 제안된 최적화 방안에 대해 제시한다. 4장에서는 구현결과를 제시하고, 5장의 결론으로 마무리한다.

II. 배경지식

본 장에서는 먼저 두 종류의 Montgomery 곡선에 관해 설명하고, CSIDH와 radical isogeny에 관해 소개한다.

2.1 Montgomery 곡선과 뒤틀린 Montgomery 곡선

Characteristic이 2나 3이 아닌 유한체를 K라고 가정하자. K위에서 정의된 Montgomery 곡선과 뒤틀린 Montgomery 곡선은 다음의 식으로 각각 정의할 수 있다.

$$M_{a,b} : by^2 = x^3 + ax^2 + x \tag{1}$$

$$M_{a,b}^{-} : by^2 = x^3 + ax^2 - x \tag{2}$$

$M_{a,b}$에서 $a, b \in K$이며, $b(a^2-4) \neq 0$이다. 이와 유사하게 $M_{a,b}^{-}$에서 $a, b \in K$이며, $b(a^2+4) \neq 0$이다. 본 논문에서는 $M_{a,b}$ 형태의 타원곡선을 Montgomery$^+$ 곡선이라 한다. $b=1$일 경우 b를 생략하여 M_a로 나타내도록 한다. 마찬가지로 $M_{a,b}^{-}$ 형태의 타원곡선을 Montgomery$^-$곡선이라 정의하며, $b=1$일 경우에 M_a^{-}로 표현한다.

2.2 CSIDH

CSIDH (Commutative CSIDH)는 Castryck

등에 의해 제안된 isogeny 기반 키 교환 알고리즘으로, 기존 CRS에 supersingular 타원곡선을 적용하여 최적화하였다[5]. CSIDH는, 일반적으로 supersingular 타원곡선의 endomorphism ring은 비가환적이지만, F_p에서 정의될 경우 가환성이라는 점을 이용하였다. 이차 수체에서의 order를 $\mathcal{O}$를 로 정의하자. F_p위에 정의된 endomorphism ring을 $\mathcal{O}$로 하는 타원곡선들의 집합을 $E\ell\ell_p(\mathcal{O})$로 하자. 그러면 class group $Cl(\mathcal{O})$의 타원곡선에서의 group action은 $[a]E$로 표현할 수 있고, 이는 Velu의 공식을 이용하면 효율적으로 계산할 수 있다. 여기서 $E\in E\ell\ell_p(\mathcal{O})$이고, $[a]\in Cl(\mathcal{O})$이다.

CSIDH에서는 효율적인 group action 연산을 수행하기 위해서 작은 소수들의 곱 ℓ_i로 이루어진 소수 $p=4\ell_1\ell_2\cdots\ell_n-1$를 사용한다. E를 $End_p(E)=\mathbb{Z}[\pi]$인 F_p위에서 정의된 supersingular 타원곡선이라 하고, $End_p(E)$는 F_p에서 정의된 E의 endomorphism ring을 이라 하자. E는 supersingular 곡선이기 때문에 Frobenious trace는 0이 되어 $E(F_p)=p+1$을 만족한다. 따라서 이는 $\pi^2-1\equiv 0 \bmod \ell_i$을 의미하기 때문에, $\ell_i\mathcal{O}$는 $\ell_i\mathcal{O}=\mathfrak{l}_i\bar{\mathfrak{l}_i}$의 형태로 인수분해 될 수 있다. 이 때, $\mathfrak{l}_i=(\ell_i,\pi-1)$, $\bar{\mathfrak{l}}_i=(\ell_i,\pi+1)$이다. 이를 이용하면 group action은 Velu의 공식을 이용해서 효율적으로 연산할 수 있다.

CSIDH 키 교환 알고리즘은 다음과 같이 진행된다. Alice는 자신의 개인키에 해당하는 아이디얼을 $Cl(\mathcal{O})$에서 선택하는데, $[a]\in Cl(\mathcal{O})$는 $[a]=\mathfrak{l}_1^{e_1}\cdots\mathfrak{l}_n^{e_n}$ 형태로 표현될 수 있기 때문에, $(e_1,\cdots,e_n)\in\mathbb{Z}^n$의 벡터로 생각할 수 있다. 여기에서 e_i는 임의의 양의 정수 m에 대해서 $e_i\in[-m,m]$를 만족한다. Alice는 Velu의 공식을 이용해서 $E_A=[a]E$를 연산하고 E_A를 Bob에게 전달한다. Bob도 Alice와 동일하게 자신의 개인키 $[b]=(e_1,\cdots,e_n)\in\mathbb{Z}^n$를 선택해서 Velu의 공식을 이용해 group action을 연산한 뒤, Alice에게 $E_B=[b]E$를 전달한다. Bob으로부터 받은 값으로 Alice는 $[a]E_B$를 연산한다. Bob도 마찬가지로 Alice로부터 받은 값을 이용해 $[b]E_A$를 연산한다. F_p위에서 endomorphism ring은 가환이기 때문에 $[a]E_B=[b]E_A$를 만족하게 되어 서로 같은 비밀값을 공유할 수 있다.

2.3 Radical Isogenies

최근 Castryck, Decru, Vercauteren은 작은 차수의 isogeny를 효율적으로 연산하는 방법을 제안했다[24]. 타원곡선 $E(F_p)$에서 ℓ-isogeny를 연산하는 방법은 두 단계로 구성되어있다. 먼저 F_p에서 위수가 ℓ인 점 P를 생성한 다음, P를 커널로 하는 ℓ-isogeny를 Velu의 공식을 이용해 연산한다. 위수가 ℓ인 점 P를 생성하기 위해서는, 먼저 F_p에서 랜덤한 점 Q를 생성한 다음에, cofactor $k=\#E(F_p)/\ell$을 곱해서 $P=[k]Q$를 만든다. 만약에 P가 무한원점과 같다면, 다른 랜덤한 수를 선택한다음에 위 과정을 반복한다. 따라서, 이 방법은 특히 차수가 작은 isogeny 연산할 때 비효율적인데, 위수 ℓ에 대해서 P가 무한원점이 될 확률 (실패확률) 이 $1/\ell$이기 때문이다. 그렇기 때문에 $\prod_{i=1}^{n}\ell_i$-torsion 점을 선택한 다음에 isogeny를 반복적으로 연산하여 $\ell_1,\cdots\ell_n$-isogeny를 연산하는 방법이 더 효율적이다. 하지만, 작은 차수의 경우 이 방법도 실패 확률이 존재하기 때문에 랜덤한 점을 다시 선택할 경우가 존재한다.

[24]에서는 SIDH와 유사한 방법으로 작은 차수 isogeny의 경우 isogeny chain을 연산하는 아법을 제안한다. 타원곡선 E에 대해서 $\phi: E\to E'$를 n-isogney라 하고, n-torsion 점 P에 대해서 $\ker(\phi)=\langle P\rangle$라 하자. [24]의 아이디어는 E'에서의 n-torsion 점 P'를 E와 P의 계수로 표현하는 것이다. 이렇게 되면, ϕ와 isogeny $E'\to E'/\langle P'\rangle$의 합성은 n^2-isogeny가 된다. 더 구체적으로는, 유한체 K위에서 정의된 타원곡선 E와 위수가 $n\geq 4$인 K-rational 점 P는 다음의 Tate normal form으로 표현할 수 있다.

$$E: y^2+(1-c)xy-by=x^3-bx^2,\quad P=(0,0) \quad (3)$$

위 식에서 $b,c\in K$이다. 그 다음, Velu의 공식을 이용하면 isogenous한 곡선 $E'=E/\langle P\rangle$를 얻을

수 있다. E'에서 n-torsion 점 P'는 E와 P로 표현이 가능해서, 다음 합성함수

$$E \rightarrow E' \rightarrow E'/\langle P \rangle \quad (4)$$

는 n^2-isogeny가 된다. 이 방법을 반복적으로 사용하면 n^e-isogeny를 한 번의 n-torsion 점을 이용해서 연산할 수 있다.

III. Radical isogney의 최적화

본 장에서는 효율적인 CSIDH 구현에서 적용을 위해 radical isogeny를 최적화하는 방법에 관해 서술한다. 구체적으로, 본 장에서는 radical 2-, 3-, 5-, 7-isogeny에 관한 최적화 방안을 제시한다. Radical 7-isogeny를 최대로 사용하는 이유는 다음과 같다. 먼저, 2-, 4-isogeny와 다르게, n^e-isogeny 연산을 위해서는 n-torsion 점 하나는 반드시 필요하다. 따라서 m개의 서로 다른 radical isogeny 차수를 사용할 경우, m개의 torsion 점이 필요한 상황인데, 개인키의 지수의 최대값이 5인 기존 CSIDH를 고려해봤을 때 m이 증가할수록 (특히 $m \geq 5$) radical isogeny의 효율성은 감소한다. 두 번째 이유는, radical isogeny의 경우 차수가 올라갈수록 공식이 복잡해지고 연산량이 많아지기 때문에 CSIDH 구현 최적화를 위해서는 7차가 최대 차수이다.

또한, 최근 [25]에서 projective coordinate를 사용할 경우 radical isogeny 연산 시 역원연산을 감소시킬 수 있는 연구결과가 제시되었다. 해당 방법은 4-, 5-, 7차 radical isogeny에 적용이 가능하며, 본 논문에는 그 방법을 추가적으로 최적화하였다. 또한, 본 논문에서는 Montgomery 곡선과 Tate normal form 사이의 변환에 관한 최적화도 진행하였다.

3.1 Radical 2^e-isogeny

Radical 2-isogeny에 관한 설명을 위해 먼저 $p \equiv 7 \bmod 8$에 대해서 유한체 F_p에서 정의된 supersingular curve에 대해서 알아본다. 이 유한체 위에서는 supersingular curve E는 두 가지 그룹으로 분류하고 있다. 하나는 $\mathbb{Z}[\sqrt{-p}]$를 endormophism ring으로 가지는 floor에 존재하는 곡선이고, 다른 하나는 $\mathbb{Z}[(1+\sqrt{p})/2]$를 endomorphism ring으로 가지는 surface에 존재하는 곡선이다. Surface에 존재하는 곡선의 경우 위수가 2인 서로 다른 F_p-rational 점을 3개 가지는데, 이 점들은 다음과 같이 분류할 수 있다.

- P^-: P의 이등분 점의 x좌표는 F_p에 정의되지 않는다.
- P_1^+: P의 이등분 점은 F_p에 정의되지 않지만 x좌표는 F_p에 정의되어있다.
- P_2^+: P의 이등분 점은 F_p에 정의되어있다.

[4]의 Lemma 9에 제시된 것처럼, P_1^+ 이나 P_2^+ 그룹에 정의된 점을 사용하면 연속된 2-isogeny를 연산할 수 있다. 또한, [4]의 Proposition 14에 의하면, $\mathbb{Z}[(1+\sqrt{p})/2]$를 endomorphism ring으로 가지는 supersingular 곡선은 M_a^-와 F_p-isomorphic 하다. 본 논문에서는 [26]에 정의된 2-isogeny에 관한 최적화 공식을 사용하였다. [26]에 제시된 방법을 사용한 2^e차 isogeny의 연산량은 $3\mathbf{M}+2\mathbf{S}+3\mathbf{E}+(e-1)(1\mathbf{M}+1\mathbf{S}+1\mathbf{E})$이며, $\mathbf{E}$는 유한체 위에서 지수연산을 의미한다. 한편, 소수 p에 대해서 $p \equiv 3 \bmod 4$를 만족할 때, 유한체 F_p에서 $a \in F_p$에 대해 a의 역원은 a^{p-2}로 연산할 수 있으며, a의 제곱근은 $a^{(p+1)/4}$로 연산된다. 또한, 연산한 뒤 제곱근을 연속적으로 구할 때에는 $a^{(p+1)(p-2)/4 \bmod p-1}$로 연산할 수 있다. [26]에서 언급된 바와 같이 2^e-isogeny 연산 시에는 유한체에서의 역원과 제곱근 연산이 효율성에 연관을 미치게 된다. 따라서 해당 지수연산은 실험을 통해서 얻게 된 값인 window size 6을 활용하여 sliding window 방식을 이용해 구현하였다.

3.2 Radical 3^e-isogeny

최근에 Onuki와 Moriya는 Montgomery+ 곡선과 Montgomery- 곡선에서 효율적인 radical 3-, 4-isogeny 공식을 제시하였다 [27]. 기존 [24]에서 제안된 radical ℓ-isogeny 공식은 주어

진 Montgomery 곡선에서 Tate normal form E 로 변환이 필요하다. 이 때, 기존 Montgomery 곡선에서의 ℓ-torsion 점은 Tate normal form에서 점 $P=(0,0)$으로 변환된다. 이 후, Velu 공식을 이용해서 isogenous 한 곡선 $E'=E/\langle P\rangle$을 생성한다. E'의 곡선의 계수와 E'에서의 ℓ-torsion 점 P'는 E의 계수로 표현할 수 있어서 연속된 isogeny 연산을 가능하게 한다. 하지만 Montgomery 곡선에서 Tate normal form으로의 변화는 특정 isogeny 차수의 경우 까다롭다. [27]에서는 특히 3-, 4-isogeny에 대해서 효율적인 radical isogeny 공식을 제안한다. 제안하는 방법은 ℓ-torsion 점과 ℓ차 division polynomial의 관계를 이용하였다.

본 논문에서는 Montgomery+곡선에서 Onuki와 Moriya의 3-isogeny 곡선을 최적화한 공식을 제안한다. 먼저 Onuki와 Moriya의 공식은 다음과 같다. 유한체 F_p위에서 정의된 타원곡선 M_a에서 3-isogeny를 연산하는 방법은 3-torsion point인 점 P를 생성자로 하는 cyclic group을 커널로 하는 isogeny를 구하는 Velu의 공식을 따른다. 이 때, P의 x좌표를 t라 하면, 3-isogenous 한 곡선 $E'=E/\langle P\rangle$위에서의 3-torsion point의 x좌표는 다음 식으로 나타낸다.

$$3t\alpha^2+(3t^2-1)\alpha+3t^3-2t \tag{5}$$

여기에서 α는 $t(t^2-1)$의 세 제곱근이다. 따라서 초기 시작 타원곡선에서의 3-torsion point의 x좌표를 알면, 이를 활용하면 isogenous 곡선의 3-torsion point의 x좌표를 구할 수 있고, 이를 반복하면 3-isogeny chain을 연속적으로 구할 수 있다. 마지막으로 division polynomial로부터 Montgomery 곡선의 계수를 복원할 수 있다. 최종 곡선을 M_a라 하면, 계수 a는 다음 식을 따른다.

$$a=\frac{-3t'^4-6t'^2+1}{4t'^3} \tag{6}$$

여기에서 t'는 M_a에서의 3-torsion point의 x좌표이다. 이를 이용하며 Tate normal form으로의 변환이 필요없어 효율적인 isogeny 연산이 가능하다. 위의 식의 최적화 연산 알고리즘은 다음과 같다.

Algorithm 1. Computing 3^e-isogeny on M_a over F_p

Input : M_a and 3-torsion point P, where $x(P)=t$
Output : 3^e-isogenous curve $M_{a'}$

// isogeny computation
1. For $i=0$ to e do
2. $t0\leftarrow t^2$
3. $t1\leftarrow t0\cdot t$
4. $\alpha\leftarrow t1-t$
5. $t2\leftarrow\alpha+\alpha$
6. $t2\leftarrow t2+\alpha$
7. $\alpha\leftarrow\sqrt[3]{\alpha}$
8. $t2\leftarrow t2+t$
9. $t3\leftarrow t+\alpha$
10. $t3\leftarrow t3\cdot\alpha$
11. $t3\leftarrow t3\cdot t$
12. $t0\leftarrow t3+t3$
13. $t0\leftarrow t0+t3$
14. $t\leftarrow t0-\alpha$
15. $t\leftarrow t+t2$

// recovering the coefficient
16. $t0\leftarrow t^2$
17. $t1\leftarrow t0\cdot t$
18. $t1\leftarrow t1+t1$
19. $t1\leftarrow t1+t1$
20. $t2\leftarrow t0+2$
21. $t0\leftarrow t2\cdot t0$
22. $t2\leftarrow t0+t0$
23. $t2\leftarrow t2+t0$
24. $t2\leftarrow 1-t2$
25. $t1\leftarrow t1^{-1}$
26. $a\leftarrow t1\cdot t2$
return a

Fig. 1. Method of computing consecutive 3-isogenies on M_a

아래 표는 본 논문에서 Onuki와 Moriya의 공식을 최적화한 연산량과 [24]의 연산량을 비교한 표이다. 여기에서 $\boldsymbol{M}$는 유한체에서의 곱셈, $\boldsymbol{S}$는 유한체에서의 제곱, $\boldsymbol{E}$는 유한체에서의 지수 연산을 의미한다.

위 표와 [27]의 결과에서도 확인할 수 있듯이, isogeny 연산량 자체는 [24]가 더 효율적이다. 하지만 [24]의 공식은 Montgomery에서 Tate로 변환화는 과정에서의 연산량이 많아서, 특정 isogeny

Table 1. Comparison of the computational cost of radical 3-isogeny formula between [24] and our optimized version of [27]

	[24]	Ours
Transform from Montgomery	$4\boldsymbol{M}+2\boldsymbol{E}$	-
Isogeny	$2\boldsymbol{M}+1\boldsymbol{E}$	$3\boldsymbol{M}+1\boldsymbol{S}+1\boldsymbol{E}$
Transform to Montgomery	$16\boldsymbol{M}+4\boldsymbol{E}$	$3\boldsymbol{M}+1\boldsymbol{S}+1\boldsymbol{E}$

차수에서는 [27]의 공식이 더 효율적일 수 있다. 특히, CSIDH-512의 경우 3^e-isogeny를 연산한다 가정하자. CSIDH에 사용되는 소수 특성상 $1\boldsymbol{S}\cong 1\boldsymbol{M}$ 이고, $1\boldsymbol{E}$ 는 sliding window 방법을 사용하면 대략 $600\boldsymbol{M}$이 필요하다. 따라서 [27] 공식이 더 효율적이려면,

$$20\boldsymbol{M}+6\boldsymbol{E}+(2\boldsymbol{M}+1\boldsymbol{E})e\rangle 4\boldsymbol{M}+1\boldsymbol{E}+(4\boldsymbol{M}+1\boldsymbol{E})e$$

를 만족해야하고, 이는 $e<1508$을 의미한다. 보통 radical isogeny를 이용해서 CSIDH를 구현하는 경우 isogeny 차수는 100 이하로 하므로, 3-isogeny 연산량이 많지만 [27]을 사용하는 것이 더 효율적이라는 것을 알 수 있다.

3.3 Radical 5^e-isogeny

Radical 5-isogeny 연산은 다음과 같다. 주어진 5-torsion point Q에 대해서 Montgomery 곡선 M_a를 isomorphic한 다음 곡선으로 변환한다.

$$E: y^2+(1-b)xy-by=x^3-bx^2 \tag{7}$$

M_a의 Q는 E에서 $P=(0,0)$으로 대응된다. r을 Q의 x좌표라 하자. 이 경우

$$b=-\frac{(4ar^3+3r^4+6r^2-1)^3}{(4r(r^2+ar+1))^4} \tag{8}$$

이 된다. E에서 $\langle P\rangle$를 커널로 하여 Velu의 공식을 적용하면 5-isogenous 곡선 $E'=E/\langle P\rangle$를 얻을 수 있다. 5-isogeny를 연속적으로 계산하기 위해서는 E'에서의 5-torsion point Q'가 $(0,0)$으로 대응되야 한다. Q'를 $(0,0)$으로 변환했을 때 이에 대응하는 곡선은

$$E': y^2+(1-b')xy-b'y=x^3-b'x^2 \tag{9}$$

이 되고, 위 식에서 b'는 다음과 같이 계산된다.

$$b'=\alpha\frac{\alpha^4+3\alpha^3+4\alpha^2+2\alpha+1}{\alpha^4-2\alpha^3+4\alpha^2-3\alpha+1} \tag{10}$$

여기에서 $\alpha=\sqrt[5]{b}$ 이다. 연속적인 5-isogeny를 연산한 후에 E'를 다시 Montgomery 곡선으로 되돌려야 한다. Montgomery 곡선에서 Tate 곡선으로의 변환은 8M+1E가 소요되며, 5-isogeny 하나의 연산량은 5M+2E이다. 마지막으로 Weierstrass 곡선에서 Montgomery 곡선으로의 변환은 18M+4E의 연산량이 필요하다. 한편, [25]에서는 5-isogeny의 효율적인 연산을 위해 projective coordinate를 사용하여 역원연산을 없애는 방법을 제안하였다. $\alpha=X/Z$라 하면, $b'=X'/Z'$는 다음과 같이 연산할 수 있다.

$$X'=X(X^4+3X^3Z+4X^2Z^2+2XZ^3+Z^4) \tag{11}$$
$$Z'=Z(X^4-2X^3Z+4X^2Z^2-3XZ^3+Z^4) \tag{12}$$

이 후 $(X':Z')=(X'Z'^4:Z'^5)$을 이용하면, 역원연산 없이 5 제곱근 연산 한 번만으로 5-isogeny를 구할 수 있다.

3.4 Radical 7^e-isogeny

M_a에 주어진 7-torsion 점 Q에 대해서 Q의 x좌표를 r이라 할 때, Q를 $P=(0,0)$으로 보내는 곡선 E는 다음과 같이 정의된다.

$$E: y^2+(-N^2+N+1)xy+(-N^3+N^2)y = x^3+(-N^3+N^2)x^3 \tag{14}$$

위 식에서

$$N=\frac{(r^2(3r^2+4ar+6)-1)^3}{(2(r^2(r^2+2ar+6)+2ar+1))}$$

$$\cdot \frac{1}{(4r^2+4ar+4)^2(r^4-r^2)} \quad (14)$$

E에서 $\langle P\rangle$를 커널로 하여 Velu의 공식을 적용하면 7-isogenous 곡선 $E'=E/\langle P\rangle$을 얻게 된다. 5-isogeny와 마찬가지로, 연속적인 7-isogeny 연산을 위해서는 E'에서의 7-torsion 점 Q'를 $(0,0)$으로 보내야 한다. 이 경우 곡선 E'는 다음과 같다

$$E': y^2+(-N'^2+N'+1)xy \\ +(-N'^3+N'^2)y=x^3+(-N'^3+N'^2)x^2 \quad (15)$$

위 식에서 N'은 $\alpha=\sqrt[7]{N^5-N^4}$에 관한식으로 표현이 가능하지만, 본 논문에서는 생략하도록 한다. 5-isogeny와 마찬가지로 projective coordinate를 사용하면 역원연산을 줄일 수 있다. 다음 표는 [ ]에서의 구현과 본 논문에서의 최적화를 비교한 표이다. [Table 2]에서 [M+ to M-]은 Montgomery$^+$ 곡선에서 Montgomery$^-$ 곡선으로의 변화를 의미하고, [M- to M+]은 Montgomery$^-$ 곡선에서 Montgomery$^+$ 곡선으로의 변환을 의미한다. [M to Tate]는 Montgomery 곡선에서 Tate normal form으로의 변환을 의미하고, [Tate to M]는 Tate normal form에서 Montgomery 곡선으로의 변환을 의미한다. 또 P to A는 projective coordinate에서 Affine coordinate으로의 변환을 의미한다. 마지막으로 W to M은 Weierstrass 곡선에서 Montgomery 곡선으로의 변환을 의미한다.

Table 2. Comparison of the computational cost of radical isogeny formula between [25] and ours

Degree		[25]	Ours
2	M+ to M-	$2M+2E$	$2M+1E$
	Isogeny	$2M+1E$	$2M+1E$
	M- to M+	$6M+5E$	$5M+3E$
3	M to Tate	$4M+2E$	-
	Isogeny	$2M+1E$	$3M+1S+1E$
	Tate to M	$16M+4E$	$3M+1S+1E$
5	M to Tate	$10M+2E$	$8M+1E$
	Isogeny	$14M+1E$	$12M+1E$
	P to A	$4M+1E$	$4M+1E$
	W to M	$27M+5E$	$18M+4E$
7	M to Tate	$10M+2E$	$10M+1E$
	Isogeny	$26M+1E$	$18M+1E$
	P to A	$1M+1E$	$1M+1E$
	W to M	$29M+5E$	$20M+4E$

IV. 구현 결과

본 장에서는 radical isogeny를 이용해 constant-time CSIDH를 구현한 결과를 제시한다. 본 장에서 $CRADS_n$을 radical isogeny를 n 차까지 사용한 CSIDH 구현이라 명명한다. 본 논문에서는 $CRADS_n$을 사용했을 때의 성능을 CSIDH와 128-bit 고전 보안강도와, 128-bit 양자 보안강도에서 비교한다. Constant-time CSIDH 알고리즘으로는 OAYT-style 알고리즘을 사용한다 [22]. Constant-time $CRADS_n$에 대해서는 2, 3, 5, 7 radical isogeny에 대해서는 constant-time 연산을 수행하고, 나머지 홀수 차수 isogeny에 대해서는 OAYT-style 알고리즘을 사용한다. 측정에 사용한 CPU는 3.60GHz의 동작 주파수를 가지는 Intel Core i7-7700를 사용했으며, Ubuntu 20.04 LTS 운영체제상에서 최적화 옵션 -O3과 GCC 9.3.0 컴파일러를 이용했다.

4.1 Classical CSIDH

4.1.1 파라미터 선택

128-bit 고전 보안강도를 가지는 CSIDH 구현을 위해 다음 511비트 소수를 사용하였다.

$$p_{511}=2^4\cdot 3^3\cdot 5\cdot 7\cdot 11^2\cdot 13\cdots 373-1 \quad (16)$$

유한체 $F_{p_{511}}$위에서 supersingular Montgomery$^+$ 곡선

$$M_0^+: y^2=x^3+x \quad (17)$$

를 base 곡선으로 사용하였다. $CRADS_n$의 경우

$$M_0^- : y^2 = x^3 - x \quad (18)$$

Table 3. Private key exponent range and its security

	Exponent range	Security
CSIDH [5]	$[-5,5]^{73}$	252.54
$CRADS_2$	$[-58,58]$ $\times[-3,3]$ $\times[-4,4]^2$ $\times[-5,5]^{53}$ $\times[-4,4]^{15}$ $\times[-3,3]^2$	252.53
$CRADS_3$	$[-90,90]$ $\times[-62,62]$ $\times[-2,2]$ $\times[-3,3]^2$ $\times[-4,4]^4$ $\times[-5,5]^{49}$ $\times[-4,4]^{11}$ $\times[-3,3]^3$ $\times[-2,2]^2$	252.53
$CRADS_5$	$[-90,90]$ $\times[-65,65]$ $\times[-50,50]$ $\times[-3,3]^2$ $\times[-4,4]^2$ $\times[-5,5]^{38}$ $\times[-4,4]^{22}$ $\times[-3,3]^4$ $\times[-2,2]^3$	252.54
$CRADS_7$	$[-90,90]$ $\times[-65,65]$ $\times[-44,44]$ $\times[-29,29]$ $\times[-3,3]$ $\times[-4,4]^2$ $\times[-5,5]^{28}$ $\times[-4,4]^{34}$ $\times[-3,3]^2$ $\times[-2,2]^2$ $\times[-1,1]$	252.52

를 base 곡선으로 사용하였다. 개인키의 범위는 다음 테이블과 같다.

4.1.2 실험 결과

[Table 3]의 파라미터를 사용해서 constant-time group action의 결과는 다음과 같다. 본 논문의 constant-time 구현은 [1]에서 제안한 strategy 방법을 사용하지 않았으며, 100 차수 이상의 홀수 차수 isogeny 연산은 [2]에서 제안한 square-root Velu 공식을 사용하였다. [Table 4]는 100,000번의 group action의 cycle count를 평균 낸 것이다.

[Table 4]에 나와 있듯이, constant-time $CRADS_n$은 기존 CSIDH 보다 최대 15.3% 빠르다는 것을 알 수 있다. 속도 향상의 원인은, 비록 radical isogeny가 지수승 연산으로 기존 Velu의 공식을 이용한 isogeny 연산보다 isogeny 자체는 연산량이 많이 드나, 큰 유한체 위에서 랜덤한 특정 torsion point를 적게 선택한다는 점에서 성능 향상을 가져올 수 있다.

Table 4. Performance result of group action of CSIDH and $CRADS_n$

	Group action	Savings
CSIDH	336,343,562	-
$CRADS_2$	229,267,517	11.0%
$CRADS_3$	298,006,889	11.3%
$CRADS_5$	286,682,988	14.7%
$CRADS_7$	284,758,164	15.3%

4.2 Quantum CSIDH

최근 [28]에서는 CSIDH에 대해 보다 더 정확한 quantum analysis를 수행하였으며, CSIDH가 양자컴퓨팅 환경에서도 128 비트 보안강도를 가지기 위해서는 유한체의 크기가 4096비트가 되어야 한다는 결론을 제시했다. 본 논문에서는 4095 비트 유한체를 사용해서 radical isogeny를 적용한 결과를 제시한다.

4.2.1 파라미터 선택

128-bit 양자 보안강도를 가지는 CSIDH 구현을 위해 다음 4095 비트 소수를 사용하였다.

$$p_{4095} = 2^3 \cdot 3 \cdot 5 \cdot 7 \cdots 2879 \cdot 4603 - 1 \quad (19)$$

위 소수에 사용된 홀수 소수의 개수는 416이다. 이 유한체 위에서

유한체 $F_{p_{4095}}$위에서 supersingular Montgomery$^+$ 곡선

$$M_0^+ : y^2 = x^3 + x \quad (20)$$

를 base 곡선으로 사용하였다. $CRADS_n$의 경우

$$M_0^- : y^2 = x^3 - x \quad (21)$$

를 base 곡선으로 사용하였다. CSIDH-4096의 경우 radical isogeny를 2차까지만 사용하였는데, 그 이유는 radical isogeny 연산 시 필요한 지수연산은 유한체의 크기가 커질수록 증가하기 때문이다. 구현에 사용된 개인키의 범위는 다음 테이블과 같다.

4.2.2 실험 결과

〔Table 5〕의 파라미터를 사용해서 constant-time group action의 결과는 다음과 같다. CSIDH-512와 마찬가지로, 100차 이상의 isogeny 연산에는 square-root Velu 공식을 사용하였다. 〔Table 6〕는 10,000번의 group action의 cycle count를 평균 낸 것이다.

〔Table 6〕에서 확인할 수 있듯이, 4096비트 유한체 위에서 CSIDH와 $CRADS_2$의 속도는 거의 비슷하지만, $CRADS_2$가 CSIDH에 비해 1.5% 정도 느리다는 것을 알 수 있다. 〔28〕에 제시된 파라미터

Table 5. Private key exponent range and its security

	CSIDH 〔28〕	$CRADS_2$
Exponent Range	$[-1,1]^{139}$	$[-4,4] \times [-1,1]^{137}$
Security	220.31	220.31

Table 6. Performance results of a constant-time group action of CSIDH and $CRADS_2$ (in millions)

	CSIDH 〔28〕	$CRADS_2$
Group action	13,297.482	13,505.895
Savings	-	-1.5%

상, $CRADS_2$는 CSIDH에 비해 797차와 809차 isogeny 연산을 수행하지 않고도 동일한 보안강도를 수행할 수 있다. 797차와 809 isogeny 연산량의 합은 4,590M 의 연산량이 든다. 일반적으로 4096비트 유한체에서의 지수연산은 대략 4,543M 연산량이 들게 된다. 제시된 파라미터 상 해당 지수연산은 3번 일어나기 때문에 대략적으로 총 13,629M 연산량이 든다. 기존 CSIDH의 경우에는 2-isogeny가 특정 torsion point를 선택하지 않으면서 isogeny를 수행할 수 있어서 동일한 보안강도에서 더 효율적인 결과를 가져왔다. 이 이유는 2-isogeny 연산에서 제곱근 연산이 존재한다 하더라도 작은 torsion point를 생성하는데 있어서 실패하는 연산량보다 작기 때문에 효율적이였다. 하지만 radical 2-isogeny 연산에 필요한 지수승 연산이 유한체가 커짐에 따라 연산량이 많아져서 CSIDH-4096에서는 속도가 좋지 않음을 알 수 있다. 본 논문의 실험 결과 4096 비트 CSIDH에서는 radical isogeny를 2차까지만 사용하는 것이 최적이며, 따라서 유한체의 크기가 커질수록 radical isogeny의 성능은 낮아진다고 볼 수 있다.

V. 결 론

본 논문에서는 CSIDH 구현에 있어서 radical isogeny의 최적 사용에 대해 분석해보았다. 이를 위해 〔25〕에서 제시된 python 기반 radical isogeny를 C로 추가적인 최적화를 진행하였으며, 최근에 제안된 효율적인 3-isogeny를 적용하였다. 본 논문의 결과 128비트 고전 보안강도에서는 $CRADS_7$이 CSIDH 보다 15.3% 빠르다. 128비트 양자 보안강도에서는 radical isogeny 연산에 요구되는 지수승 연산이 유한체 크기가 커짐에 따라서 연산량이 많아져 최대 radical 2-isogeny 까지 사용이 적합하다. Radical isogeny의 경우 특히 곡선간의 변환이나, 적절한 근을 찾으면 isogeny 공식

을 최적화 할 수 있는데 향후에는 이에 대한 연구를 진행할 예정이다.

References

[1] J.J. Chi-Domiguez et al. "On new Velu's formulae and their applications to CSIDH and BSIDH constant-time implementations," IACR Cryptology ePrint Archive, 2020:1109, 2020

[2] D. Bernstein et al. "Faster computation of isogenies of large prime degree," IACR Cryptology ePrint Archive, 2020:341, 2020

[3] W. Beullens et al. "CSI-FiSh: efficient isogeny based signatures through class group computations," ASIACRYPT, LNCS 11921, pp. 227-247, Dec. 2019

[4] W. Castryck and T. Decru "CSIDH on the surface," PQCrypto, LNCS 12100, pp.111-129, April, 2020

[5] W. Castryck et al. "CSIDH: An efficient post-quantum commutaitve group action," ASIACRYPT, LNCS 11274, pp.395-427, Dec. 2018

[6] D. Cervantes-Vazquez et al. "Stronger and faster side-channel protections for CSIDH," LATINCRYPT, LNCS 11774, pp. 173-193, Sept. 2019

[7] A. Childs et al. "Constructing elliptic curve isogenies in quantum subexponential time," Journal of Mathematical Cryptology, vol. 8, no. 1, pp. 1-29, 2014

[8] C. Costello, "B-SIDH supersingular isogeny Diffie-Hellman using twisted torsion," ASIACRYPT, LNCS 12492, pp. 440-463, Dec. 2020

[9] C. Costello and H. Hisil, "A simple and compact algorithm for SIDH with arbitrary degree isogenies," ASIACRYPT, LNCS 10625, pp. 303-329, Dec. 2017

[10] J.M. Couveignes, "Hard homogenous spaces," IACR Cryptology ePrint Archive, 2006:291, 2006

[11] De Feo. et al. "Towards practical key exchange from ordinary isogeny graphs," ASIACRYPT, LNCS 11274, pp. 365-394, Dec. 2018

[12] D. Heo et al. "On the performance analysis for CSIDH-based cryptosystems," Applied Sciences, vol. 10, no. 19, 2020

[13] D. Heo et al. "Optimized CSIDH implementation using a 2-torsion point," Cryptography, vol. 4, no. 3, 2020

[14] A. Jalali, "Towards optimized and constant-time CSIDH on embedded devices," International Workshop on Constructive Side-Channel Analysis and Secure Design, pp. 215-231, 2019

[15] D. Jao, L. De Feo "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies," PQCrypto, LNCS 7071, pp. 19-34, Aug. 2011

[16] T. Kawashima, "An efficient authenticated key exchange from random self-reducibility on CSIDH," IACR Cryptology ePrint Archive, 2020:1178, 2020

[17] S. Kim et al. "New hybrid method for isogeny-based cryptosystems using Edwards curves," IEEE transactions on Information Theory, vol. 66, no. 3, pp. 1934-1943, 2020

[18] M. Meyer et al. "On lions and elligators: An efficient constant-time implementations of CSIDH", PQCrypto, LNCS 11505, pp. 307-325, 2019

[19] M. Meyer and S. Reith "A faster way to the CSIDH," INDOCRYPT, LNCS 11356, pp. 137-152, 2018

[20] M. Meyer et al. "On hybrid SIDH

schemes using Edwards and Montgomery curve arithmetic," IACR Cryptology ePrint Archive, 2017:1213, 2017

[21] D. Moody and D. Shumow, "Analogues of Velu's formula for isogenies on alternate models of elliptic curves," Mathematics of Computations, vol. 85, no. 300, pp. 1929-1951, 2016

[22] H. Onuki et al. "A constant-time algorithm of CSIDH keeping two points," IEICE Transactions on Fundamentals of Electronics, Communications, and Computer Sciences, vol. E103.A, no. 10, pp. 1174-1182, 2020

[23] A. Stolbunov, "Constructing public-key cryptographic schemes based on class group action on a set of isogenous ellitpic curves," Advances in Mathematics of Communication, vol. 4, no. 2, pp. 215-235, 2010

[24] W. Casrtyck, T. Decru, and F. Vercauteren, "Radical isogenies," ASIACRYPT, LNCS 12492, pp. 440-463, Dec. 2020

[25] J.J. Chi-Domiguez and K. Reijnders, "Fully projective radical isogenies in constant-time" IACR Cryptology ePrint Archive, 2021:259, 2021

[26] S. Kim, "On the use of twisted Montgomery curves for CSIDH-based cryptography", Journal of the Korea Institute of Infromation Security and Cryptology, 31(3), pp. 497-508, 2021

[27] H. Onuki and T. Moriya, "Radical isogenies on Montgomery curves", IACR Cryptology ePrint Archive, 2021:699, 2021

[28] J.J. Chi-Domiguez et al, "The SQALE of CSIDH: Square-root Velu quantum-resistant isogeny action with low exponents", IACR Cryptology ePrint Archive, 2020:1520, 2020

〈저자소개〉

김 수 리 (Suhri Kim) 정회원
2014년 2월: 고려대학교 수학과 이학사
2016년 8월: 고려대학교 정보보호대학원 공학석사
2020년 2월: 고려대학교 정보보호대학원 공학박사
2020년 3월~2021년 2월: 고려대학교 정보보호대학원 박사후연구원
2020년 3월~2021년 2월: KU Leuven ESAT/COSIC 박사후연구원
2021년 3월~현재: 성신여자대학교 수리통계데이터사이언스학부 조교수
〈관심분야〉 공개키 암호시스템, 후양자암호