05

아이소제니 기반 암호 구현

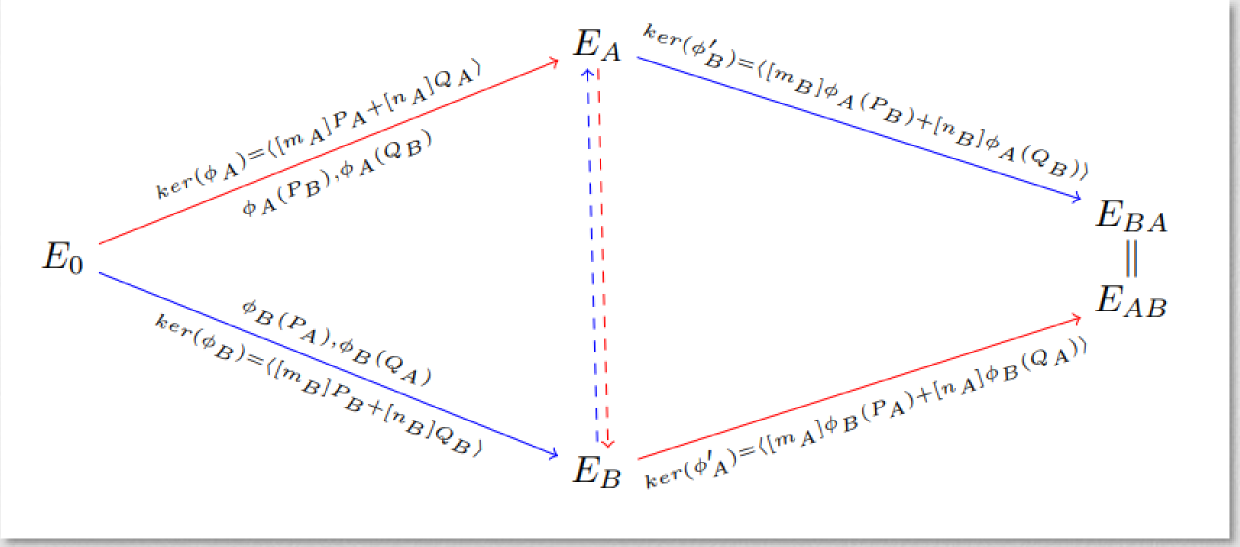
목차

1

SIDH 기반 암호 구현

SIDH

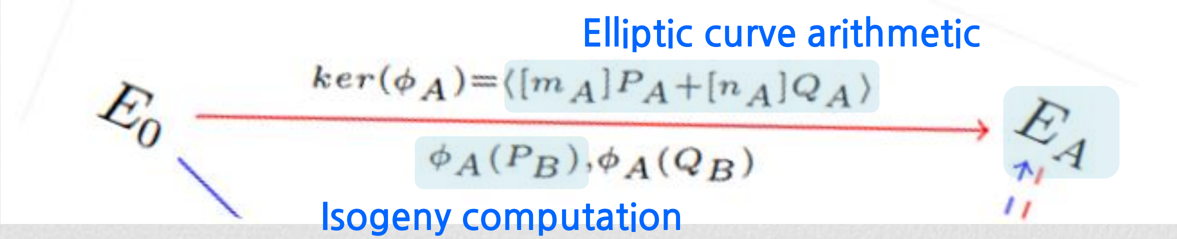
Recall



E_A
E_0
E_B
E_{BA}
$\|$
E_{AB}
$ker(\phi_A)=\langle[m_A]P_A+[n_A]Q_A\rangle$
$\phi_A(P_B),\phi_A(Q_B)$
$\phi_B(P_A),\phi_B(Q_A)$
$ker(\phi_B)=\langle[m_B]P_B+[n_B]Q_B\rangle$
$ker(\phi'_B)=\langle[m_B]\phi_A(P_B)+[n_B]\phi_A(Q_B)\rangle$
$ker(\phi'_A)=\langle[m_A]\phi_B(P_A)+[n_A]\phi_B(Q_A)\rangle$

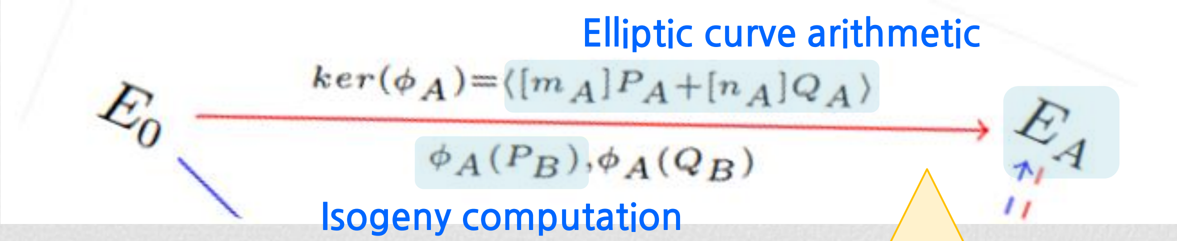
SIDH

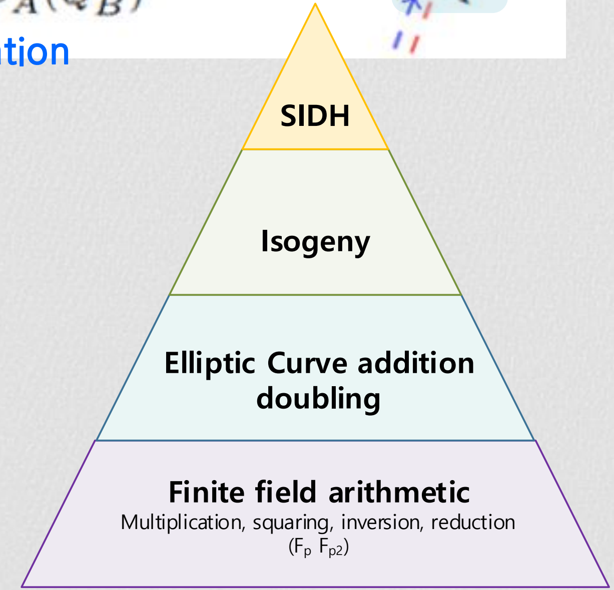
Building blocks



SIDH

Building blocks





SIDH

Isogeny

- SIDH

$$\boldsymbol{E} \xrightarrow[\boldsymbol{ker\phi = \langle m_A P_A + n_A Q_A \rangle}]{\boldsymbol{\phi}} \boldsymbol{E_A}$$

Kernel P w/ order 2^{372} ➔ 연산량 많음

- Idea
  - Isogenies used in SIDH is a separable isogeny
  - $\phi = \phi_n \circ \cdots \circ \phi_1$
  - Isogeny of degree 2^{372}➔ $O(2^{372})$
  - 2−isogeny 372 times ➔ $372 \cdot O(2)$

SIDH

Isogeny computation on Alice side

Kernel P w/ order 2^{372}

Isogeny of degree 2^{372}

$$E \xrightarrow{\phi} E_A$$

① $[2^{371}]P$ 연산 ➔ order 2

② 2-isogeny ϕ_0 where $\ker \phi_0 = \langle [2^{371}]P \rangle$

① $[2^{370}]\phi_0(P)$ 연산 ➔ order 2

② 2-isogeny ϕ_1 where $\ker \phi_1 = \langle [2^{370}]P \rangle$

Isogeny – Evaluation

General formula – Montgomery curves

- $\phi: (x, y) \to \big(f(x), yf'(x)\big)$ for degree $d = 2s + 1$

$$f(x) = x \prod_{i=1}^{s} \left(\frac{x \cdot x_i - 1}{x - x_i} \right)^2$$

$$\langle P \rangle = \{O, P, -P\} = \{O, (x_3, y_3), (x_3, -y_3)\}$$

Isogeny – Evaluation

General formula – Montgomery curves

- Example : 3-isogeny
  - $P = (x_3, y_3) \in E$, 3-torsion point in E $([3]P = O)$
  - $\phi: E \to E' = E/\langle P \rangle$
  - For a point $Q \in E$, $x(\phi(Q)) \in E$ is computed as

$$x(\phi(Q)) = x\left(\frac{x \cdot x_3 - 1}{x - x_3}\right)^2$$

Isogeny – Evaluation

General formula – Montgomery curves

- Example : 3-isogeny
  - In projective coordinates, $x_3 = \frac{X_3}{Z_3}, x = \frac{X}{Z}$

$$\frac{X'}{Z'} = \frac{X}{Z} \cdot \left(\frac{XX_3 - ZZ_3}{XZ_3 - X_3Z}\right)^2$$

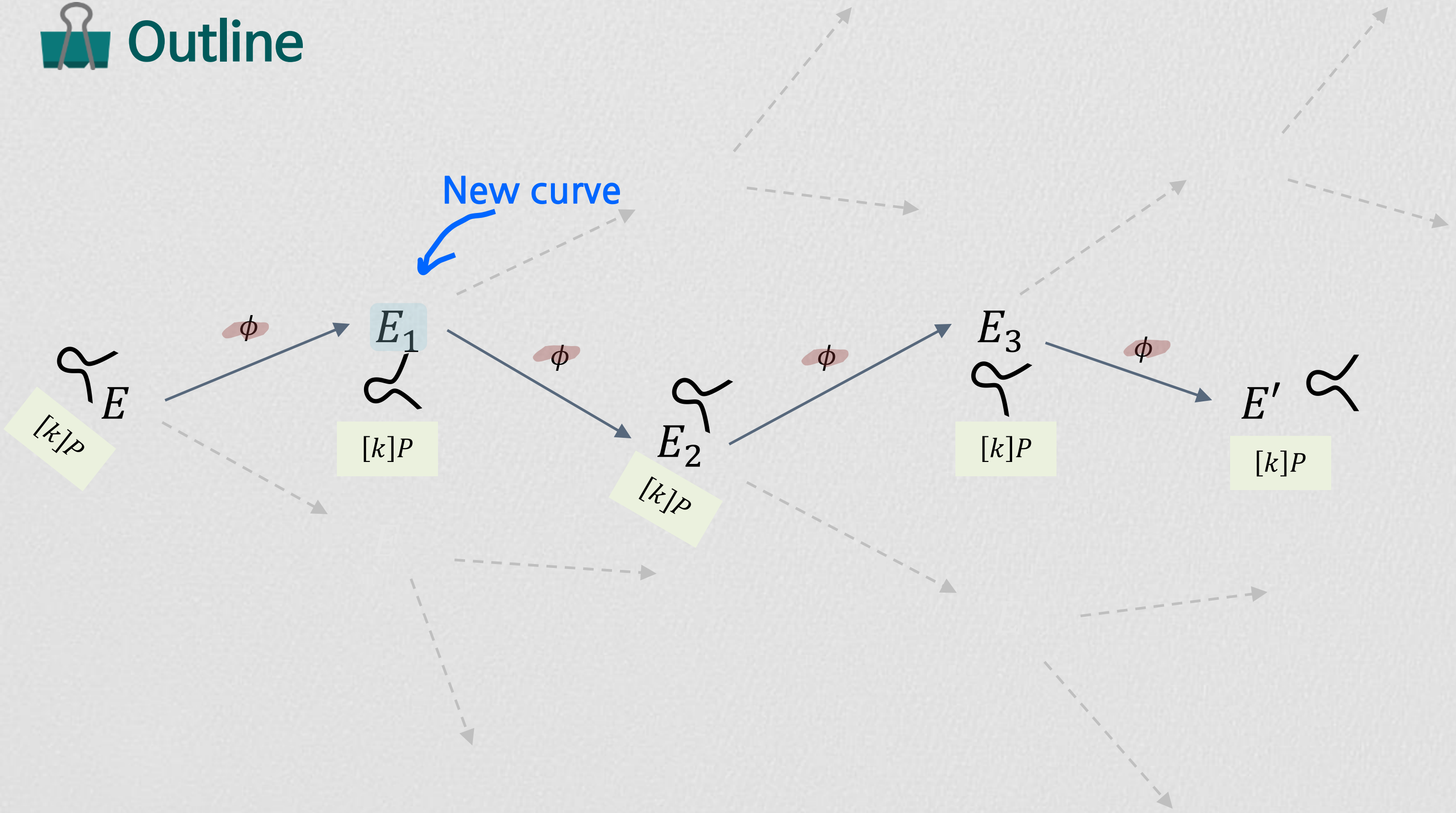
$$F = (X - Z)(X_3 + Z_3) = XX_3 + XZ_3 - ZX_3 - ZZ_3$$
$$G = (X + Z)(X_3 - Z_3) = XX_3 - XZ_3 + ZX_3 - ZZ_3$$
$$F + G = 2(XX_3 - ZZ_3)$$
$$F - G = 2(XZ_3 - ZX_3)$$

➔ **COST : 2M**

➔ **TOTAL COST : 4M+2S**

Isogeny – Coefficients

Outline



Isogeny – Coefficients

Image curve 에서의 계수 복원

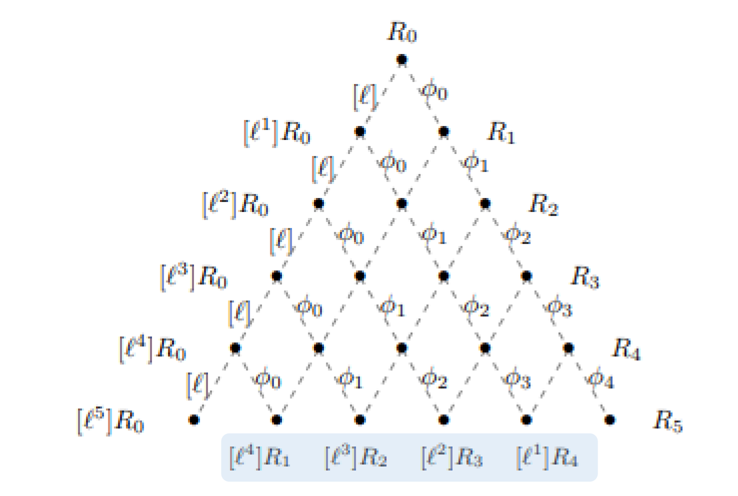
- Example : 3-isogeny

$$E: y^2 = x^3 + Ax^2 + x \xrightarrow[\boldsymbol{ker\phi} = \langle \boldsymbol{P} \rangle]{\boldsymbol{\phi}} E: x_3^2 y^2 = x^3 + \left(A + \frac{6}{x_3} - 6x_3\right)x^2 + x$$

- 타원곡선의 curve coefficient
  - n 차 division polynomial 을 이용해 n-torsion point의 좌표로 표현 가능
    - A 를 x_3 이용해 표현 가능
  - Curve coefficient 도 분수 형태로 표현됨
    - 연산 효율을 위해 projective version 사용
    - 기존 ECC 구현과 다르게 projective curve coefficient 이용
    - 타원곡선 연산 공식도 이에 맞게 변경

Others

Strategies in SIDH



연속적인 ℓ -isogeny 연산을 위해 필요

ℓ -isogeny 연산량과 $[\ell]P$ 연산량 비교를 통해 계산

Others

Public parameter 교환 시

- 공개키 $P, Q, R = P - Q$ 사용
  - 개인키로 $P + [s]Q$ 연산 시 Montgomery ladder 사용 위해
- Public parameter 교환 시
  - $\phi(P), \phi(Q), \phi(R)$ 만 교환
  - 타원곡선 계수 전달하지 않음
  - $\phi(P), \phi(Q), \phi(R)$ 로 타원곡선 계수 변환 가능

SIDH Implementation Summary

Summary

- 주어진 안전강도에 맞는 소수 찾기
  - $p = \ell_A^{e_A} \ell_B^{e_B} f - 1$
- 타원곡선 형태 및 base elliptic curve 설정
  - Montgomery curve
  - $y^2 = x^3 + 6x^2 + x$
- 구현!
  - Projective coordinate/ projective curve coefficient
  - Montgomery XZ-coordinate

2

CSIDH 기반 암호 구현

CSIDH

Recall

Alice	Bob
Choose a secret $[a]$	Choose a secret $[b]$
Compute $E_A = [a]E$	Compute $E_B = [b]E$

$$\xrightarrow{\boldsymbol{E_A}}$$

$$\xleftarrow{\boldsymbol{E_B}}$$

Compute $[a]E_B$	Compute $[b]E_A$

Shared Secret $\boldsymbol{[a][b]E = [a]E_B = [b]E_A}$

CSIDH

CSIDH 기반 암호의 핵심연산

- Computation of $[\alpha]E$
  - Naïve way
    - Example : $[\alpha] = [2]^3[3]^5$
      - E 에서 2−torsion point 선택 Velu 공식 이용해서 $E_1 = [2]E$ 연산
      - E_1 에서 2−torsion point 선택 Velu 공식 이용해서 $E_2 = [2]E_1$ 연산
      - E_2 에서 2−torsion point 선택 Velu 공식 이용해서 $E_3 = [2]E_2$ 연산
      - E_3 에서 3−torsion point 선택 Velu 공식 이용해서 $\mathrm{E}_4 = [3]E_3$ 연산

CSIDH

CSIDH 기반 암호의 핵심연산

- Computation of $[\alpha]E$
  - Problem 1
    - $[\alpha] = \ell_1^{e_1} \cdots \ell_n^{e_n}$ 에 대해서 $\sum e_i$ 번의 랜덤 point를 F_p 에서 선택해야함
    - 작은 torsion point일 수록 실패 확률 존재
    - Costly operation
  - Problem 2
    - $\ell_i^{e_i}$ 에서 e_i 가 음수일 경우 랜덤 point를 F_{p^2} 에서 선택해야함
    - 마찬가지로 순차적으로 isogeny 연산 수행 경우 실패 확률 존재

CSIDH

Isogeny computation in CSIDH

- IDEA
  - $[\alpha] = \ell_1^{e_1} \cdots \ell_n^{e_n}$ 에서 e_i 의 부호가 같은 것 끼리 연산

- Algorithm – STEP 1: Random point selection
  - F_p 에서 랜덤한 x 좌표 선택
  - $x^3 + Ax^2 + x = r$ 연산
  - r square ➔ 해당 점 F_p 에 존재
  - r non-square ➔ 해당 점 F_{p^2}에 존재

CSIDH

Isogeny computation in CSIDH

- IDEA
  - $[\alpha] = \ell_1^{e_1} \cdots \ell_n^{e_n}$ 에서 e_i 의 부호가 같은 것 끼리 연산

- Algorithm - STEP 2: Torsion point generation
  - r square ➔ 해당 점 F_p 에 존재
    - e_i 가 음수에 해당하는 소수를 다 곱해 k 구함 ➔ $k = \prod \ell_i \ s.t. e_i < 0$
    - $Q = [k]P$ 연산
  - r non-square ➔ 해당 점 F_{p^2} 에 존재
    - e_i 가 양수에 해당하는 소수를 다 곱해 k 구함 ➔ $k = \prod \ell_i \ s.t. e_i > 0$
    - $Q = [k]P$ 연산

CSIDH

Isogeny computation in CSIDH

- IDEA
  - $[\alpha] = \ell_1^{e_1} \cdots \ell_n^{e_n}$ 에서 e_i 의 부호가 같은 것 끼리 연산

- Algorithm – STEP 3: Isogeny computation
  - r square ➔ 해당 점 F_p 에 존재
    - e_i 가 양수에 해당하는 소수에 대한 isogeny 연산
    - 해당 소수를 제외한 소수를 곱해 torsion point 생성 ➔ Velu 공식 이용해 isogeny 연산

CSIDH

Isogeny computation in CSIDH

- IDEA
  - $[\alpha] = \ell_1^{e_1} \cdots \ell_n^{e_n}$ 에서 e_i 의 부호가 같은 것 끼리 연산

- Algorithm - STEP 3: Isogeny computation
  - r square ➔ 해당 점 F_p 에 존재
    - e_i 가 양수에 해당하는 소수에 대한 isogeny 연산
    - 해당 소수를 제외한 소수를 곱해 torsion point 생성 ➔ Velu 공식 이용해 isogeny 연산
  - Example
    - e_i 가 양수에 해당하는 소수가 2, 3, 5 일 경우
    - $[6]Q$ 연산 ➔ 5-isogeny 수행
    - 무한원점일 경우 skim

CSIDH

Algorithm

Algorithm 2: Evaluating the class-group action.

Input: $A \in \mathbb{F}_p$ and a list of integers $(e_1, \ldots, e_n)$.

Output: B such that $[\mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n}]E_A = E_B$ (where $E_B : y^2 = x^3 + Bx^2 + x$).

While some $e_i \neq 0$ **do**

- Sample a random $x \in \mathbb{F}_p$.
- Set $s \leftarrow +1$ if $x^3 + Ax^2 + x$ is a square in $\mathbb{F}_p$, else $s \leftarrow -1$.
- Let $S = \{i \mid e_i \neq 0,\ \mathrm{sign}(e_i) = s\}$. **If** $S = \emptyset$ **then** start over with a new x.
- Let $k \leftarrow \prod_{i \in S} \ell_i$ and compute $Q \leftarrow [(p+1)/k]P$.
- **For each** $i \in S$ **do**
  - Compute $R \leftarrow [k/\ell_i]Q$. **If** $R = \infty$ **then** skip this i.
  - Compute an isogeny $\varphi : E_A \to E_B : y^2 = x^3 + Bx^2 + x$ with $\ker \varphi = R$.
  - Set $A \leftarrow B$, $Q \leftarrow \varphi(Q)$, $k \leftarrow k/\ell_i$, and finally $e_i \leftarrow e_i - s$.

Return A.

Summary

SIDH vs CSIDH

- 유한체 연산
  - CSIDH 는 F_p, SIDH 는 F_{p^2} 연산
  - 소수의 특성상 CSIDH 는 일반 Montgomery reduction 사용
  - SIDH는 $p = 2^{e_A}3^{e_B}f \pm 1$ 의 형태로 유한체 연산이 비교적 효율적

- 아이소제니 연산
  - SIDH 는 3−, 4− isogeny 사용
  - CSIDH 는 소수를 구성하는 홀수 차수 아이소제니 사용