Implementing Isogeny-based Cryptography

June 02, 2021

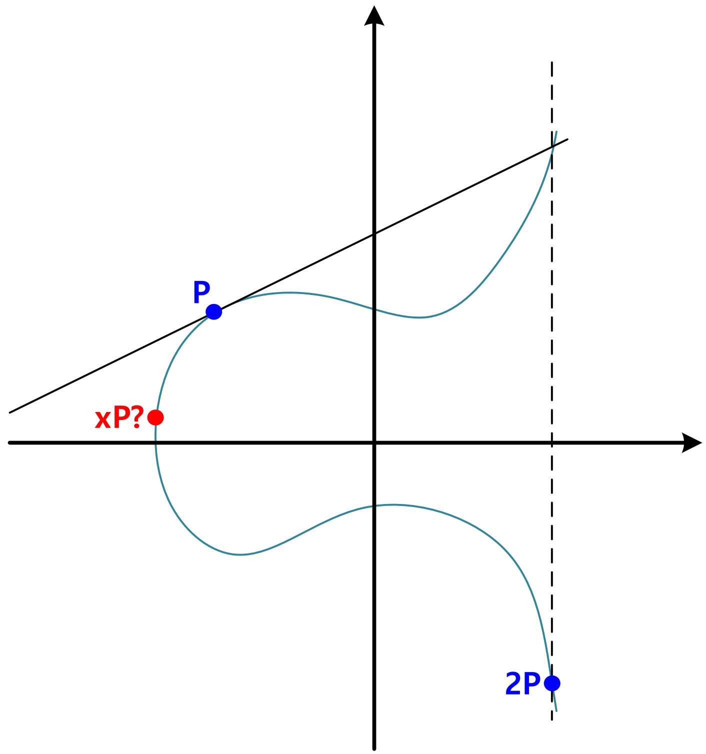
성신여자대학교 수리통계데이터사이언스학부

김수리

Contents

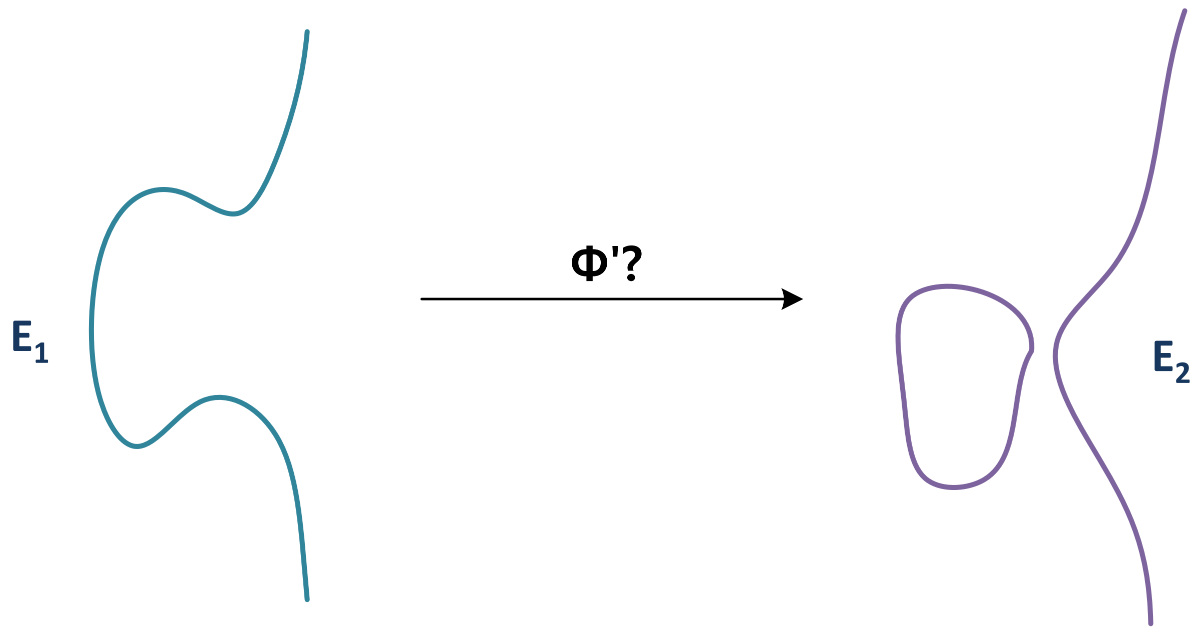
Introduction

- Standard Elliptic Curve Cryptography



Introduction

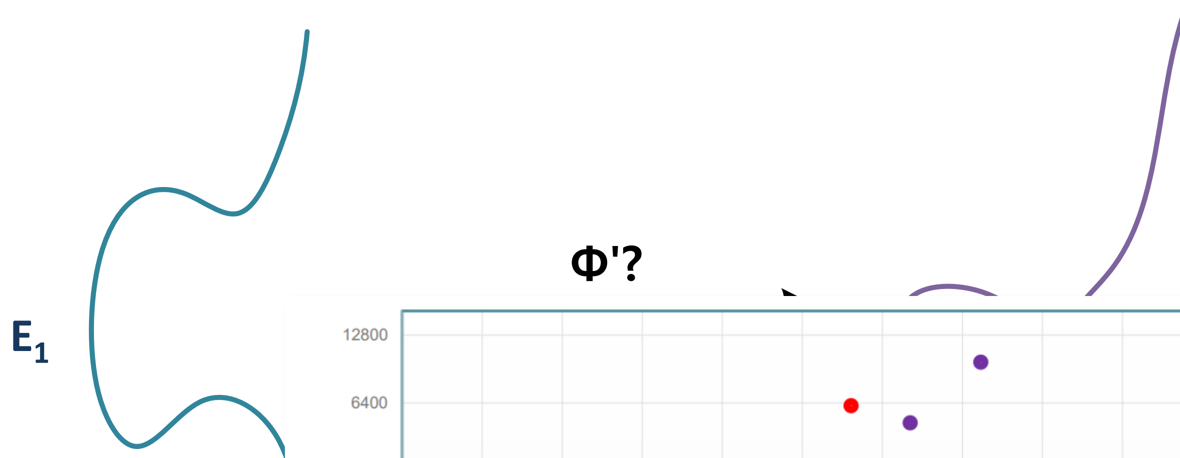
- Isogeny-based cryptography

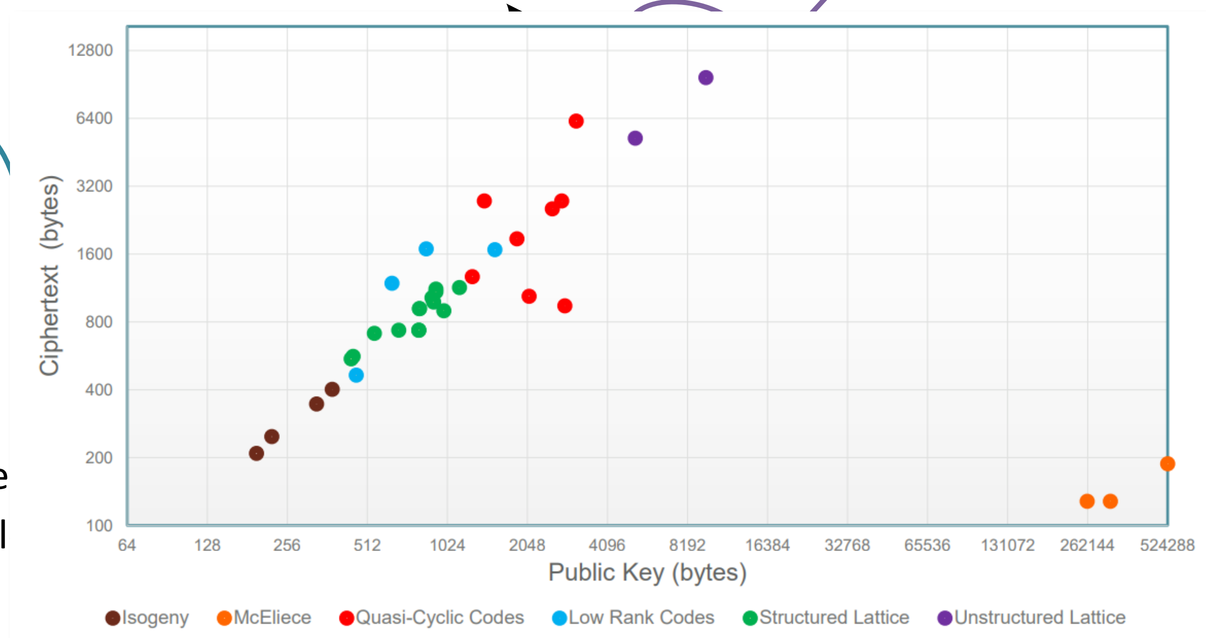


- Pros : Small key size compared to other PQC algorithms
- Cons : Magnitude slower than any other PQC algorithms

Introduction

- Isogeny-based cryptography





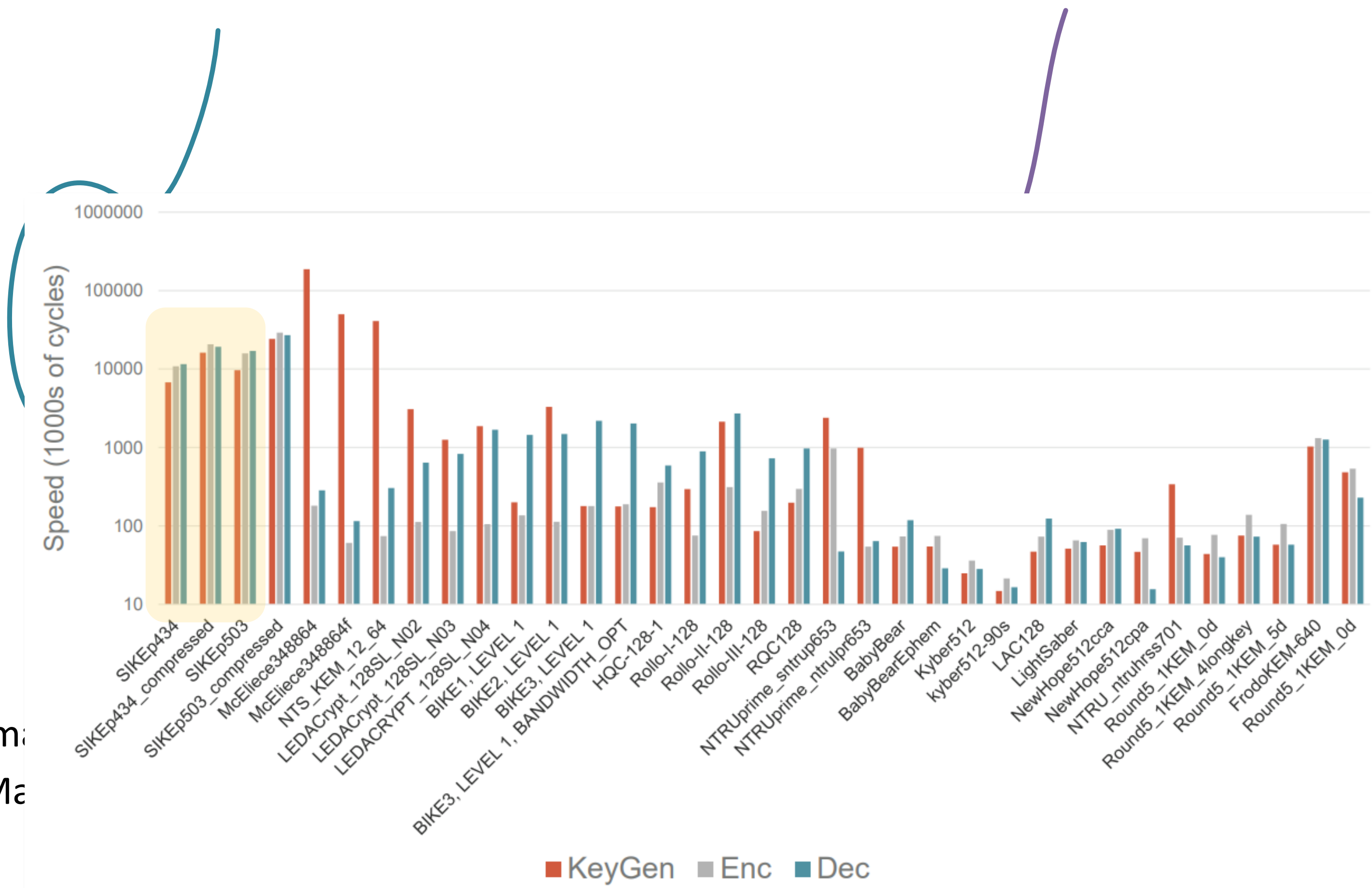
Opening Remarks PQC 2019, csrc.nist.gov

- Pros : Small key size
- Cons : Magnitude sl

Introduction

- Isogeny-based cryptography

E_1



- Pros : Sm
- Cons : Ma

Opening Remarks PQC 2019, csrc.nist.gov

History

2006
CRS Scheme
ordinary curves
- subexponential,
- inefficient

2011
SIDH
Jao, De Feo
supersingular curves
- exponential

2016
SIDH Library
Costello et al.
Practical implementation

2017
SIKE
NIST PQC Submission
Round 3 alternative candidate

2018
CSIDH
Castryck et al.
CRS using supersingular curve

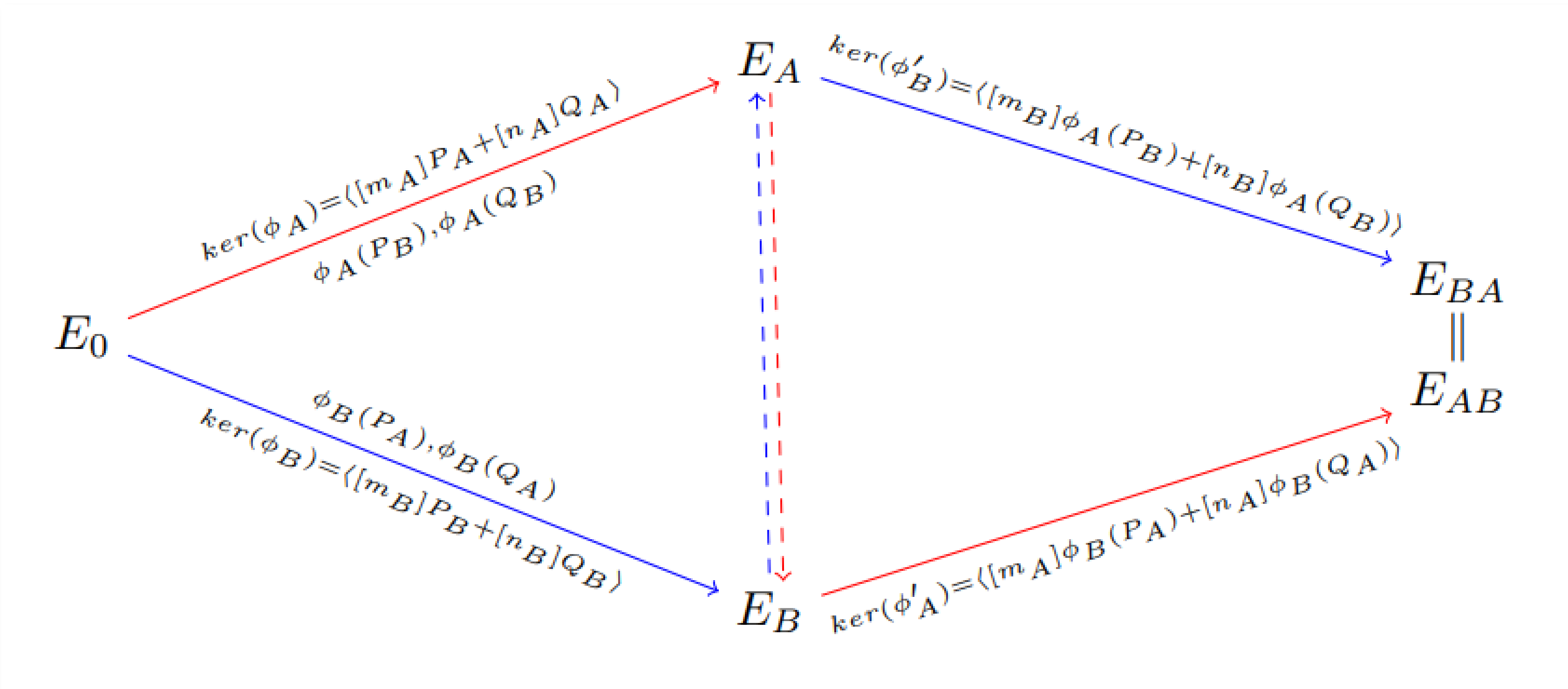
Implementing Isogeny-based Cryptography

Supersingular Isogeny Diffie Hellman (SIDH)

- Parameter Settings

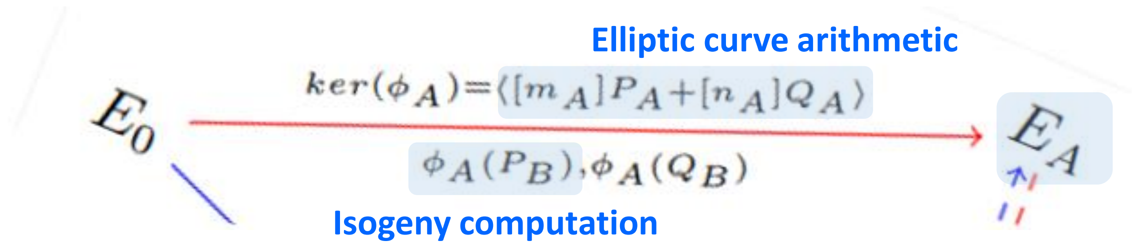
	SIDH	ECC
Prime	$p = 2^{e_A}3^{e_B} - 1$	$p = 2^{256} - 2^{224} + 2^{192} + 2^{96} + 1$
Field	F_{p^2}	F_p
Curve	Supersingular elliptic curve	Ordinary curve
Order of a curve	$(2^{e_A}3^{e_B})^2$	Near prime
Security	Hardness of finding isogeny between given two elliptic curve	Hardness of solving ECDLP
Private key	Isogeny (kernel)	d

Supersingular Isogeny Diffie Hellman (SIDH)



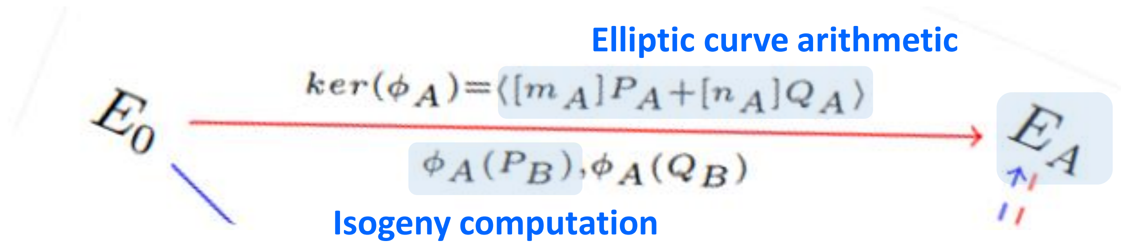
Implementing Isogeny-based cryptography

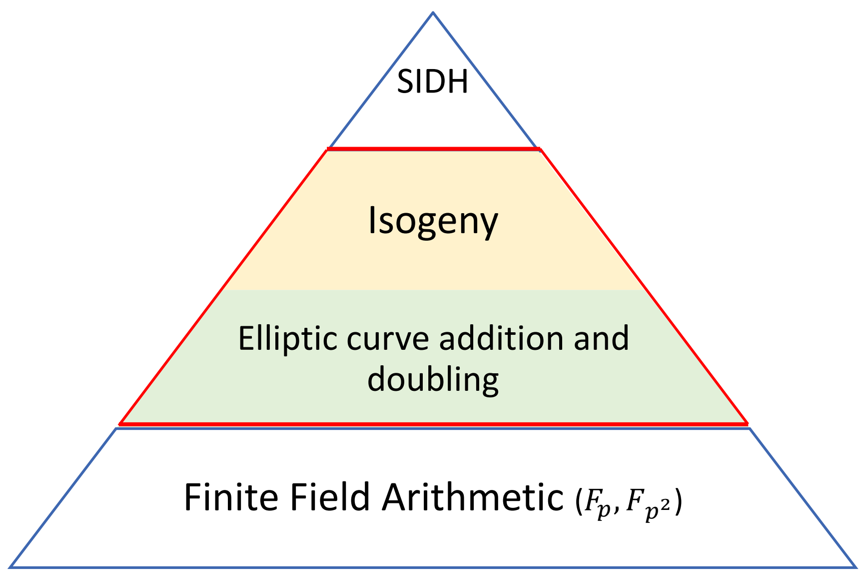
- Building blocks



Implementing Isogeny-based cryptography

- Building blocks





Isogeny

- Isogeny $\boldsymbol{\phi}: \boldsymbol{E_1} \rightarrow \boldsymbol{E_2}$ 모든곳에서 정의되는 유리함수
  - Non-constant morphism that maps the distinguished point of E_1 to the distinguished point of E_2

- Standard form of $\boldsymbol{\phi}$

$$\boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{y}) = \left(\frac{\boldsymbol{u}(\boldsymbol{x})}{\boldsymbol{v}(\boldsymbol{x})}, \frac{\boldsymbol{s}(\boldsymbol{x})}{\boldsymbol{t}(\boldsymbol{x})}\boldsymbol{y}\right)$$

  - Where $\big(u(x), v(x)\big) = 1, \big(s(x), t(x)\big) = 1$
  - $\deg\phi = \max\{\deg u, \deg v\}$

Isogeny

- Example ($\boldsymbol{F} = \boldsymbol{F_{109}}$)
  - $E_0 : y^2 = x^3 + 2x + 2 \xrightarrow{\boldsymbol{\phi}} E_1 : y^2 = x^3 + 34x + 45$

$$\boldsymbol{\phi(x, y) = \left(\frac{x^3 + 20x^2 + 50x + 6}{x^2 + 20x + 100}, \frac{x^3 + 30x^2 + 23x + 52}{x^3 + 30x^2 + 82x + 19} y \right)}$$

Isogeny

- Velu formula
  - 주어진 타원곡선 $E_1(\bar{K})$ 의 유한 subgroup $G \subset E_1(\bar{K})$ 를 kernel로 하는 isogeny ϕ 를 만들 수 있다 (Velu)
  - Order of such isogeny $\phi = ord\ G$
  - Complexity : $O(n), n = ord\ G$

모든 커널의 원소와 연산해야 함

$$\phi(P) = \left(x_P + \sum_{Q \in F - \{\infty\}} (x_{P+Q} - x_Q), y_P + \sum_{Q \in F - \{\infty\}} (y_{P+Q} - y_Q) \right).$$

Kernel

Isogeny

- Velu formula
  - 주어진 타원곡선 $E_1(\bar{K})$ 의 유한 subgroup $G \subset E_1(\bar{K})$ 를 kernel로 하는 isogeny ϕ 를 만들 수 있다 (Velu)
  - Order of such isogeny $\phi = ord\ G$
  - Complexity : $O(n), n = ord\ G$

모든 커널의 원소와 연산해야 함

$$\phi(P) = \left(x_P + \sum_{Q \in F - \{\infty\}} (x_{P+Q} - x_Q), y_P + \sum_{Q \in F - \{\infty\}} (y_{P+Q} - y_Q) \right).$$

Kernel

➔ Velu의 공식에 의해 임의의 subgroup 을 커널로 하는 아이소제니 생성 가능

➔ 함수값 연산하기 위해 커널의 모든 원소와 타원곡선 연산 수행해야함

➔ 효율성을 위해 **cyclic subgroup** 이용 : $\langle m_A P_A + n_A Q_A \rangle$

Isogeny

- SIDH

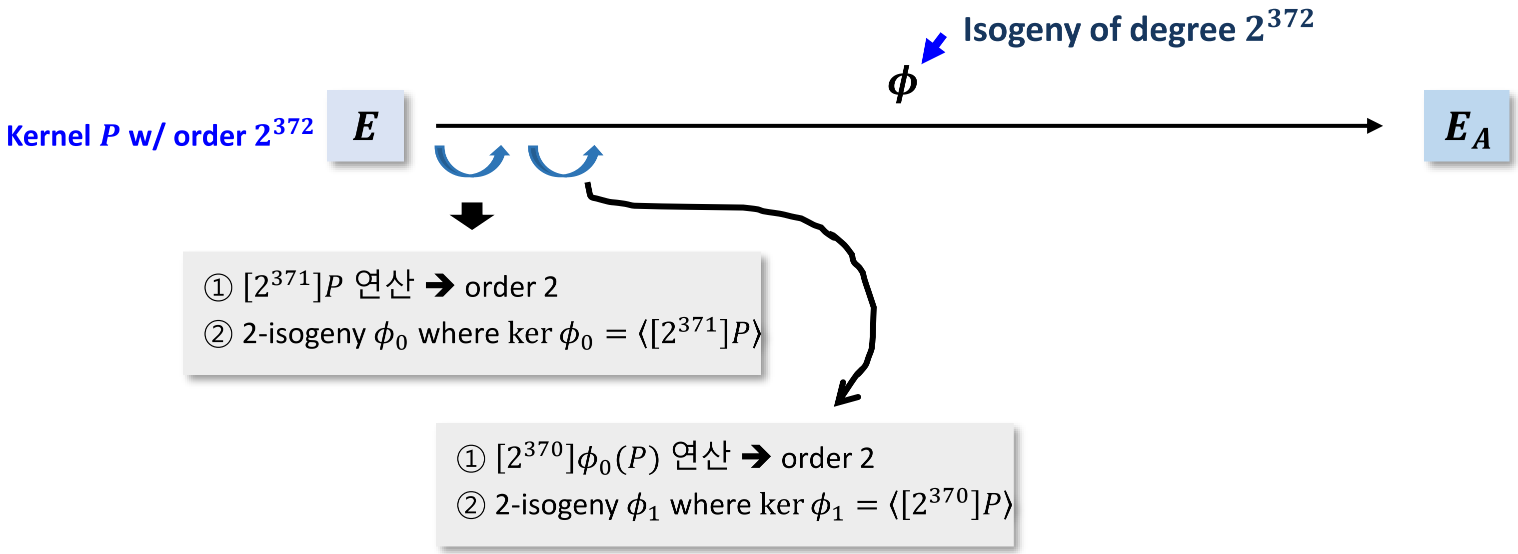
$$\boldsymbol{E} \xrightarrow[\boldsymbol{ker\phi} = \langle \boldsymbol{m_A P_A + n_A Q_A} \rangle]{\boldsymbol{\phi}} \boldsymbol{E_A}$$

Kernel $\boldsymbol{P}$ 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 ➔ $372 \cdot O(2)$

Isogeny

- Isogeny computation on Alice side



Isogeny - Evaluation

- General formula (Montgomery curves)
  - $\phi: (x, y) \rightarrow (f(x), yf'(x))$ for degree $d = 2s + 1$

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

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

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

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

Isogeny - Evaluation

- 3-isogeny

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

- In projective coordinates,
  - $x_3 = X_3/Z_3$, $x = X/Z$

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

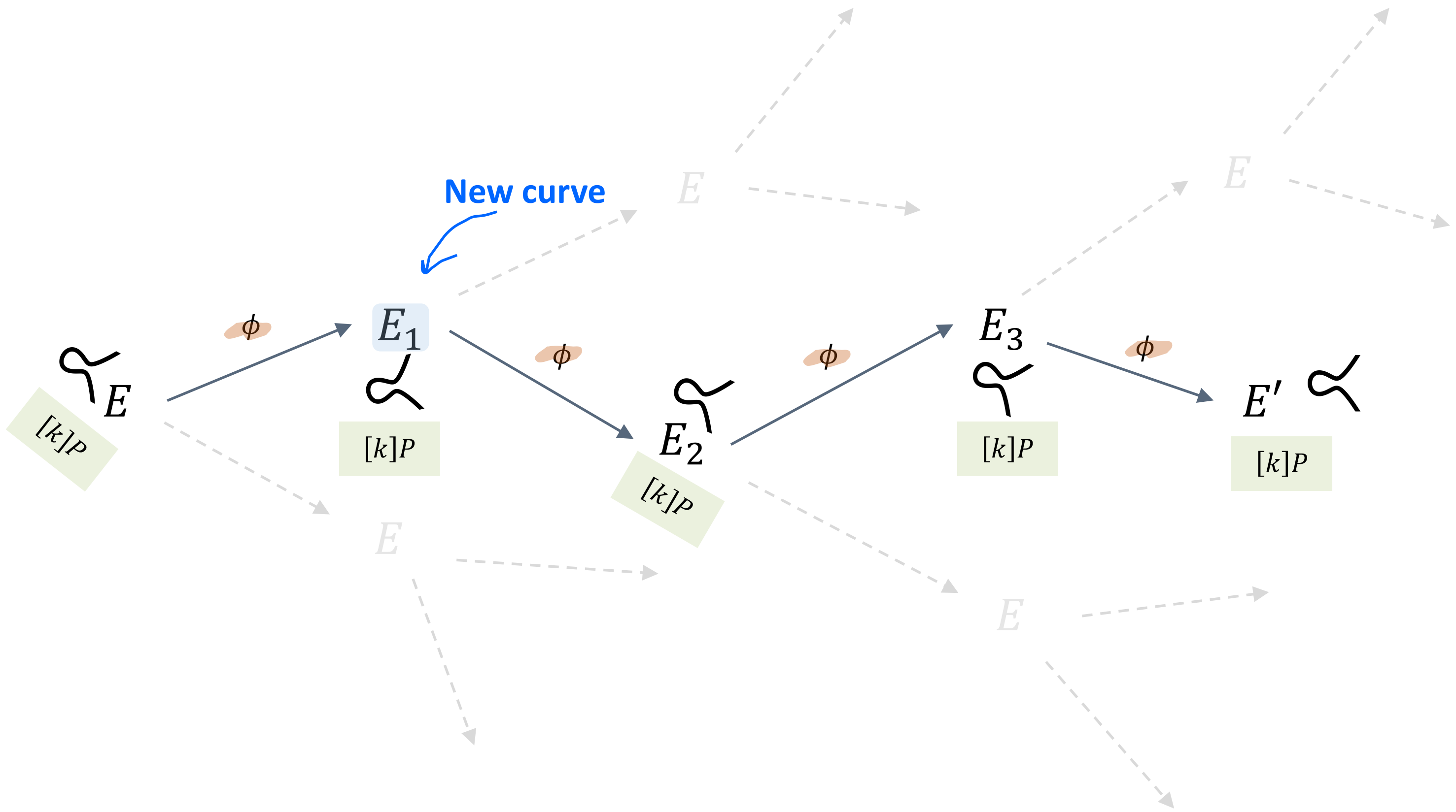
$$F = (X - Z)(X_3 + Z_3) = XX_3 + XZ_3 - ZX_3 - ZZ_3$$
$$G = (X + Z)(X_3 - Z_3) = XX_3 - XZ_3 + ZX_3 - ZZ_3$$
$$F + G = 2(XX_3 - ZZ_3)$$
$$F - G = 2(XZ_3 - ZX_3)$$

➔ **COST : 2M**

➔ **TOTAL COST : 4M+2S**

Isogeny - Coefficients

- Outline



Isogeny - Coefficients

- Image curve에서의 계수 복원
  - Example : 3-isogeny

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

Isogeny - Coefficients

- Image curve에서의 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 point 의 좌표로 표현 가능
    - A 를 x_3 이용해 표현 가능
  - **Curve coefficient 도 분수 형태로 표현**
    - 연산 효율을 위해 projective version 이용
    - 기존 **ECC 구현과 다르게 projective curve coefficient 이용**
      - 타원곡선 연산 공식도 이에 맞게 변경함

Isogeny - Coefficients

- Image curve에서의 계수 복원
  - Example : 3-isogeny

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

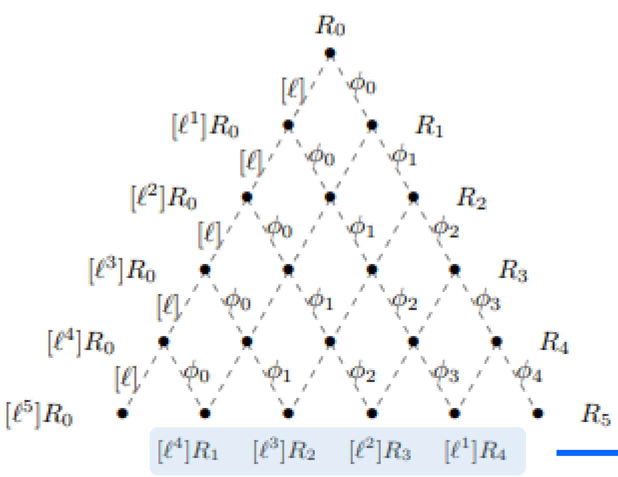
- 타원곡선의 curve coefficient
  - n 차 division polynomial 을 이용해 n-torsion point 의 좌표로 표현 가능
    - A 를 x_3 이용해 표현 가능
  - **Curve coefficient 도 분수 형태로 표현**
    - 연산 효율을 위해 projective version 이용
    - 기존 **ECC 구현과 다르게 projective curve coefficient 이용**
      - 타원곡선 연산 공식도 이에 맞게 변경함

$$\frac{A'}{C'} = \frac{Z_3^4 + 18X_3^2Z_3^2 - 27X_3^4}{4X_3Z_3^2}$$

➔ **TOTAL COST : 2M+3S**

Others

- Strategies in SIDH



연속적인 ℓ -isogeny 연산을 위해 필요

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

Isogeny – Evaluation + Coefficient

- Isogeny in SIDH

```c
// Traverse tree
index = 0;
for (row = 1; row < MAX_Bob; row++) {
    while (index < MAX_Bob-row) {
        fp2copy(R->X, pts[npts]->X);
        fp2copy(R->Z, pts[npts]->Z);
        pts_index[npts++] = index;
        m = strat_Bob[ii++];
        xTPLe(R, R, A24minus, A24plus, (int)m);
        index += m;
    }
    get_3_isog(R, A24minus, A24plus, coeff);

    for (i = 0; i < npts; i++) {
        eval_3_isog(pts[i], coeff);
    }
    eval_3_isog(phiP, coeff);
    eval_3_isog(phiQ, coeff);
    eval_3_isog(phiR, coeff);

    fp2copy(pts[npts-1]->X, R->X);
    fp2copy(pts[npts-1]->Z, R->Z);
    index = pts_index[npts-1];
    npts -= 1;
}
```

특정 위수로 만들기 위해 tripling

Image curve의 계수 연산 (한 번)

커널 point image curve 로 이동 (phiR)
상대방 공개키 연산 (phiP, phiQ)

연구 동향

- 속도 향상을 위해 연구 진행

Isogeny-based cryptography

- New scheme
  - CSURF
    - *- use of 2-isogeny*
  - B-SIDH
    - *- can use efficient primes for reduction*
- Implementation
  - Alternate curves
    - *- Edwards, Huff, ..*
  - Square-root Velu
    - - $O(\ell) \rightarrow O(\sqrt{\ell})$
  - Radical isogeny
    - *- expressing small torsion points using curve coefficients*

Thank you