FIBS : Fast Isogeny-Based Signature

Feb. 24, 2023

성신여자대학교 김수리

NSHC 윤기순　**NSHC** 이영도　**고려대학교** 허동회　**고려대학교** 김현학

Contents

Isogeny

- Isogeny $\phi: E_1 \rightarrow E_2$
  - Non-constant morphism that maps the distinguished point of E_1 to the distinguished point of E_2

모든곳에서 정의되는 유리함수

Isogeny

- Standard form of ϕ

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

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

Isogeny

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

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

Isogeny

- Some facts
  - Isogeny ≠ Isomorphism
    - E_0, E_1 is isomorphic if there exists an isogeny $\phi_1 : E_0 \to E_1$ and $\phi_2 : E_1 \to E_0$ such that $\phi_1 \circ \phi_2$=identity
    - Example by Cohen and Frey

$$\boldsymbol{\phi}(x, y) = \left(\frac{\boldsymbol{x^2 + 301x + 527}}{\boldsymbol{x + 301}}, \frac{\boldsymbol{x^2 + 602x + 1942}}{\boldsymbol{x^2 + 602x\_466}} \boldsymbol{y} \right)$$

$$E_0 : y^2 = x^3 + 1132x + 278 \xrightarrow{\boldsymbol{\phi}} E_1 : y^2 = x^3 + 500x + 1005$$

Cyclic group

Not a cyclic group

Isogeny

- Separable isogeny

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

  - Separable if $\left(\frac{u(x)}{v(x)}\right)' \neq 0$

Isogeny

- Separable isogeny
  - ϕ 가 d 차인 경우
    - $d = p_0^{e_0} p_1^{e_1} \cdots p_n^{e_n}$
    - $\phi = \underbrace{\phi_{p_0} \circ \cdots \circ \phi_{p_0}}_{e_0\text{-times}} \circ \cdots \circ \underbrace{\phi_{p_n} \circ \cdots \circ \phi_{p_n}}_{e_n\text{-times}}$

Isogeny

- Separable isogeny
  - ϕ 가 d 차인 경우
    - $d = p_0^{e_0} p_1^{e_1} \cdots p_n^{e_n}$
    - $\phi = \underbrace{\phi_{p_0} \circ \cdots \circ \phi_{p_0}}_{e_0\text{-times}} \circ \cdots \circ \underbrace{\phi_{p_n} \circ \cdots \circ \phi_{p_n}}_{e_n\text{-times}}$

- Velu
  - 주어진 타원곡선 $E(\bar{K})$ 의 유한 subgroup $G \subset E(\bar{K})$ 를 kernel로 하는 isogeny ϕ 를 만들 수 있다 Order of such isogeny $\phi = ord\ G$

Velu's Formula

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

모든 커널의 원소와 연산해야 함

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

Kernel

Velu's Formula

- Algorithm
  - Input
    - Curve of Weierstrass from E

$$E: y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

    - Points of finite subgroup of $E(\bar{K})$
  - Output
    - Codomain curve
    - Coordinate map

Velu's formula

- STEP 1 : 커널 C 의 원소를 나누기
  - 무한 원점 제거
  - C_2 : C 의 2-torsion point 들의 집합 ➔ $R: C - C_2$
  - R 을 R_+ 와 R_- 로 분해
    - $P \in R_+$ then $-P \in R_-$
  - $S = R_+ \cup C_2$

Example

$$E: y^2 = x^3 + x \in F(3^2), P = (2, 2)$$

$$\langle P \rangle = \{\mathcal{O}, P, 2P, 3P\}$$

Velu's formula

- STEP 1 : 커널 C 의 원소를 나누기
  - 무한 원점 제거
  - C_2 : C 의 2-torsion point 들의 집합 ➔ $R: C - C_2$
  - R 을 R_+ 와 R_- 로 분해
    - $P \in R_+$ then $-P \in R_-$
  - $S = R_+ \cup C_2$

Example

$\boldsymbol{E: y^2 = x^3 + x \in F(3^2), P = (2,2)}$

Order 4

$\boldsymbol{\langle P \rangle = \{\mathcal{O}, P, 2P, 3P\}}$

Order 4

Order 2

Velu's formula

- STEP 1 : 커널 C 의 원소를 나누기
  - 무한 원점 제거
  - C_2 : C 의 2-torsion point 들의 집합 ➔ $R: C - C_2$
  - R 을 R_+ 와 R_- 로 분해
    - $P \in R_+$ then $-P \in R_-$
  - $S = R_+ \cup C_2$

Example

$$E: y^2 = x^3 + x \in F(3^2), P = (2, 2)$$

$$\langle P \rangle = \{\mathcal{O}, P, 2P, 3P\}$$

- $C_2 = 2P$
- $R_+ = P$
- $R_- = 3P$

Velu's formula

- STEP 2 : $Q \in S$ 에 대해서 다음을 연산

$$g_Q^x = 3x_Q^2 + 2a_2x_Q + a_4 - a_1y_Q$$

$$g_Q^y = -2y_Q - a_1x_Q - a_3$$

$$v_Q = \begin{cases} g_Q^x & \text{if } 2Q = \infty \\ 2g_Q^x - a_1g_Q^y & \text{otherwise} \end{cases}$$

$$u_Q = (g_Q^y)^2$$

$$v = \sum_{Q \in S} v_Q, \quad w = \sum_{Q \in S} (u_Q + x_Q v_Q)$$

Velu's formula

- STEP 2 : $Q \in S$ 에 대해서 다음을 연산

Example

$$E: y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

$$E: y^2 = x^3 + x \rightarrow a_1 = a_3 = a_2 = a_6 = 0, a_4 = 1$$

$$S = R_+ \cup C_2 = \{P\} \cup \{2P\}$$

$$g_Q^x = 3x_Q^2 + 2a_2x_Q + a_4 - a_1y_Q$$

$$g_Q^y = -2y_Q - a_1x_Q - a_3$$

$$v_Q = \begin{cases} g_Q^x & \text{if } 2Q = \infty \\ 2g_Q^x - a_1g_Q^y & \text{otherwise} \end{cases}$$

$$u_Q = (g_Q^y)^2$$

$$v = \sum_{Q \in S} v_Q, \quad w = \sum_{Q \in S} (u_Q + x_Q v_Q)$$

Velu's formula

- STEP 2 : $Q \in S$ 에 대해서 다음을 연산

Example

$$E: y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

$$E: y^2 = x^3 + x \rightarrow a_1 = a_3 = a_2 = a_6 = 0, a_4 = 1$$

$$S = R_+ \cup C_2 = \{P\} \cup \{2P\}$$

$P = (2,2)$

$g_P^x = 3(2)^2 + 2(0)(2) + (1) - (0)(2) = 13 \equiv 1 \; mod \; 3$
$g_P^y = -2(2) - (0)(2) - 0 = -4 \equiv 2 \; mod \; 3$
$v_P = 2g_P^x - a_1 g_P^y = 2$
$u_P = (2)^2 = 4 \equiv 1 \; mod \; 3$

$$g_Q^x = 3x_Q^2 + 2a_2x_Q + a_4 - a_1y_Q$$

$$g_Q^y = -2y_Q - a_1x_Q - a_3$$

$$v_Q = \begin{cases} g_Q^x & \text{if } 2Q = \infty \\ 2g_Q^x - a_1 g_Q^y & \text{otherwise} \end{cases}$$

$$u_Q = (g_Q^y)^2$$

$$v = \sum_{Q \in S} v_Q, \quad w = \sum_{Q \in S} (u_Q + x_Q v_Q)$$

Velu's formula

- STEP 2 : $Q \in S$ 에 대해서 다음을 연산

Example

$$E: y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

$$E: y^2 = x^3 + x \rightarrow a_1 = a_3 = a_2 = a_6 = 0, a_4 = 1$$

$$S = R_+ \cup C_2 = \{P\} \cup \{2P\}$$

$$2P = (0,0)$$

$$g^x_{[2]P} = 3(0)^2 + 2(0)(0) + (1) - (0)(2) = 1$$
$$g^y_{[2]P} = -2(0) - (0)(0) - 0 = 0$$
$$v_{[2]P} = g^x_{[2]P} = 1$$
$$u_{[2]P} = (0)^2 = 0$$

$$g^x_Q = 3x_Q^2 + 2a_2x_Q + a_4 - a_1y_Q$$

$$g^y_Q = -2y_Q - a_1x_Q - a_3$$

$$v_Q = \begin{cases} g^x_Q & \text{if } 2Q = \infty \\ 2g^x_Q - a_1g^y_Q & \text{otherwise} \end{cases}$$

$$u_Q = (g^y_Q)^2$$

$$v = \sum_{Q \in S} v_Q, \quad w = \sum_{Q \in S} (u_Q + x_Q v_Q)$$

Velu's formula

- STEP 2 : $Q \in S$ 에 대해서 다음을 연산

Example

$$E: y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

$$E: y^2 = x^3 + x \rightarrow a_1 = a_3 = a_2 = a_6 = 0, a_4 = 1$$

$$S = R_+ \cup C_2 = \{P\} \cup \{2P\}$$

P

$v_P = 2$
$u_P = 1$

2P

$v_{[2]P} = 1$
$u_{[2]P} = 0$

$$v = v_P + v_{[2]P} = 3 \equiv 0 \; mod \; 3$$

$$w = (u_P + x_P v_P) + \left(u_{[2]P} + x_{[2]P} v_{[2]P}\right) = 5 \equiv 2 \; mod \; 3$$

$$g_Q^x = 3x_Q^2 + 2a_2x_Q + a_4 - a_1y_Q$$

$$g_Q^y = -2y_Q - a_1x_Q - a_3$$

$$v_Q = \begin{cases} g_Q^x & \text{if } 2Q = \infty \\ 2g_Q^x - a_1g_Q^y & \text{otherwise} \end{cases}$$

$$u_Q = (g_Q^y)^2$$

$$v = \sum_{Q \in S} v_Q, \quad w = \sum_{Q \in S} (u_Q + x_Q v_Q)$$

Velu's formula

- STEP 3 : image 곡선의 계수 구하기

$$A = a_1, A_2 = a_2\ , A_3 = a_3$$
$$A_4 = a_4 - 5v, A_6 = a_6 - \left(a_1^2 + 4a_2\right)v - 7w$$

- $E' : y^2 + A_1xy + A_3y = x^3 + A_2x^2 + A_4x + A_6$

Example

$E : y^2 = x^3 + x$ ➔ $a_1 = a_3 = a_2 = a_6 = 0, a_4 = 1$ $\quad v = 0, w = 2$

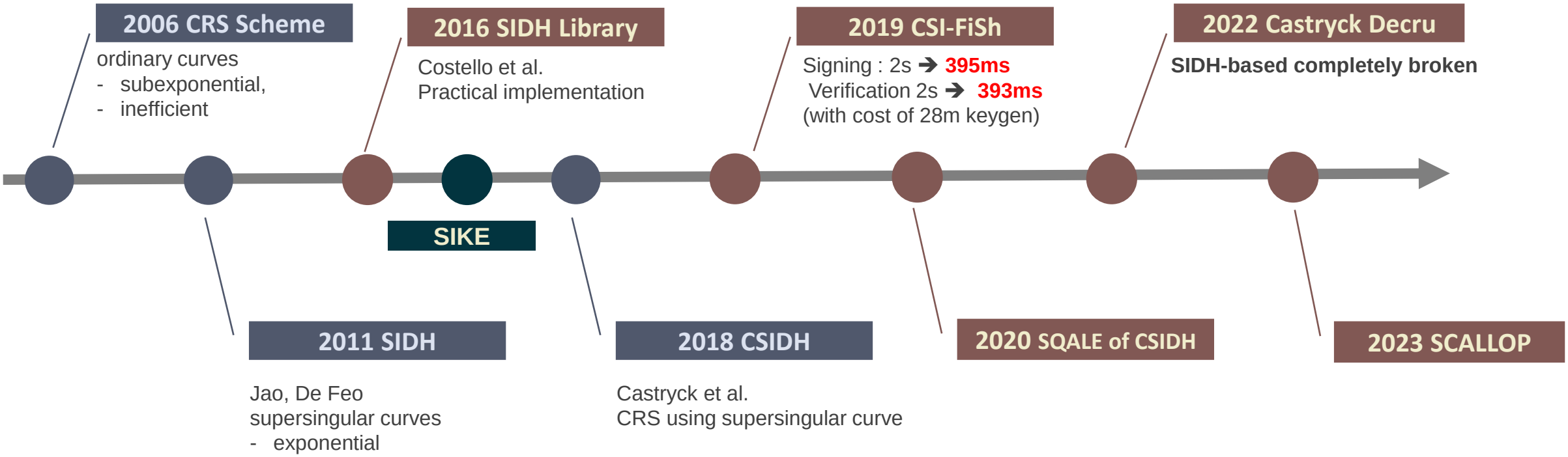
$$A = 0, A_2 = 0\ , A_3 = 0$$
$$A_4 = 1, A_6 = -14 \equiv 1\ mod\ 3$$

$$E' : y^2 = x^3 + x + 1$$

Isogeny-based Cryptography

Isogeny-based cryptography

- History



~~SIDH~~ (Supersingular Isogeny Diffie-Hellman)

- Application to Isogeny-based cryptography
  - Isogeny 기반 암호의 개인키는 isogeny를 사용
    - Isogeny ➔ Rational map
    - 함수를 키로 저장하는 것이 어려움 (large key sizes)
  - Velu's formula 로 임의의 subgroup을 커널로 하는 아이소제니 생성 가능

IDEA 1) Isogeny 대신에 커널을 저장하자! ➔ 커널을 비밀키로 사용

~~SIDH~~ (Supersingular Isogeny Diffie-Hellman)

- Application to Isogeny-based cryptography
  - Isogeny 기반 암호의 개인키는 isogeny를 사용
    - Isogeny ➔ Rational map
    - 함수를 키로 저장하는 것이 어려움 (large key sizes)
  - Velu's formula 로 임의의 subgroup을 커널로 하는 아이소제니 생성 가능

IDEA 1) Isogeny 대신에 커널을 저장하자! ➔ 커널을 개인키로 사용

  - '그룹'을 개인키로 할 경우 마찬가지로 저장의 문제
    - 구조가 명확하지 않은 그룹일 경우 집합의 원소를 다 저장해야 함
    - Cyclic group 이용!
      - Cyclic group은 generator로 모든 원소 표현 가능 ➔ generator를 저장
      - 또한 Velu의 공식은 모든 원소와 타원곡선 연산이 필요해 연산을 수행해야하는데, cyclic 그룹은 타원곡선 연산을 최적화 가능
        - 2P, 3P..

IDEA 2) 커널은 cyclic group으로 선택하자!

~~SIDH~~ (Supersingular Isogeny Diffie-Hellman)

- Application to Isogeny-based cryptography
  - 커널의 order가 증가해야 isogeny 차수도 증가하고 안전성도 증가됨
    - 하지만 큰 커널의 order은 마찬가지로 타원곡선 연산을 수행해야하기 때문에 연산량 증가
    - Isogeny 기반 암호는 separable isogeny 를 활용하기 때문에 효율적 연산 가능

IDEA 3) 커널의 order는 ℓ^e 형태로 하자 !

~~SIDH~~ (Supersingular Isogeny Diffie-Hellman)

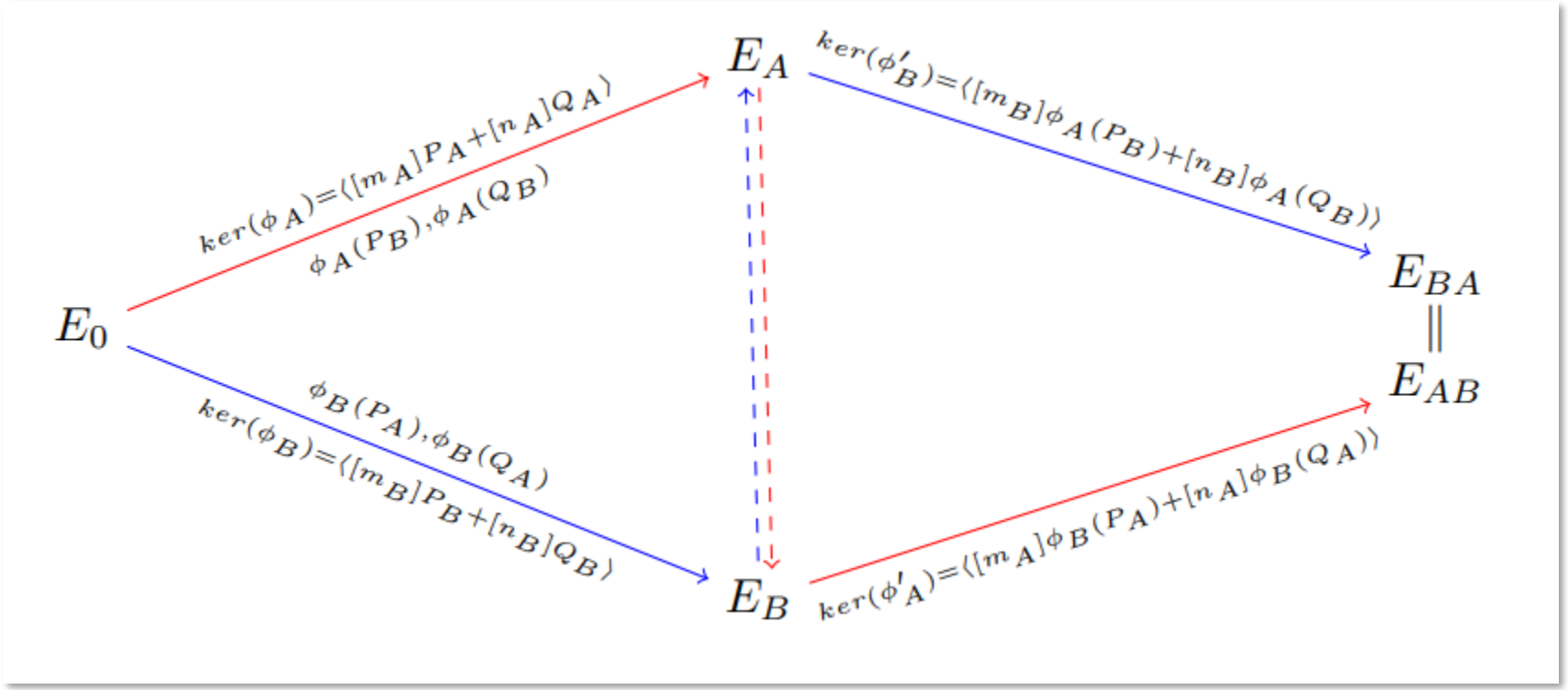
- IDEA
  - Isogeny 기반 암호는 2006년 Couveignue 에 의해 처음으로 제안
  - Ordinary curve 사용으로 비효율적일 뿐만 아니라 Childs 등의 하지수시간 복잡도 존재
  - De Feo, Jao 가 Supersingular 곡선을 사용하는 SIDH 제안
  - SIDH는 효율성을 위해서 앞서 3개의 idea를 선택

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

Kernel P w/ order 2^{372} ➔ 연산량 많음

- 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~~ (Supersingular Isogeny Diffie-Hellman)

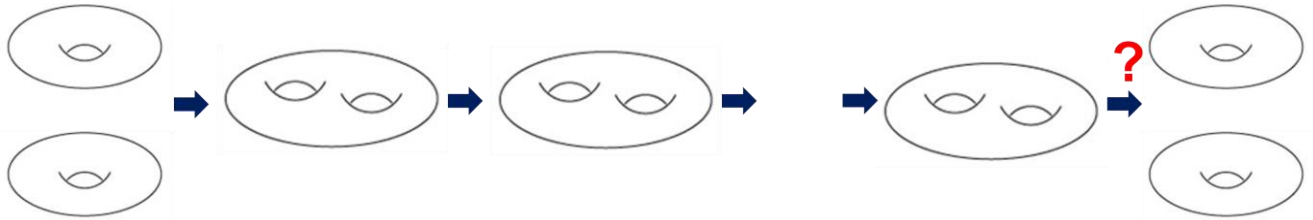


Recent attack in Isogeny-based Cryptography

- Key recovery attack (Castryck, Decru)
  - Kani의 'glue-and-split' 알고리즘 사용

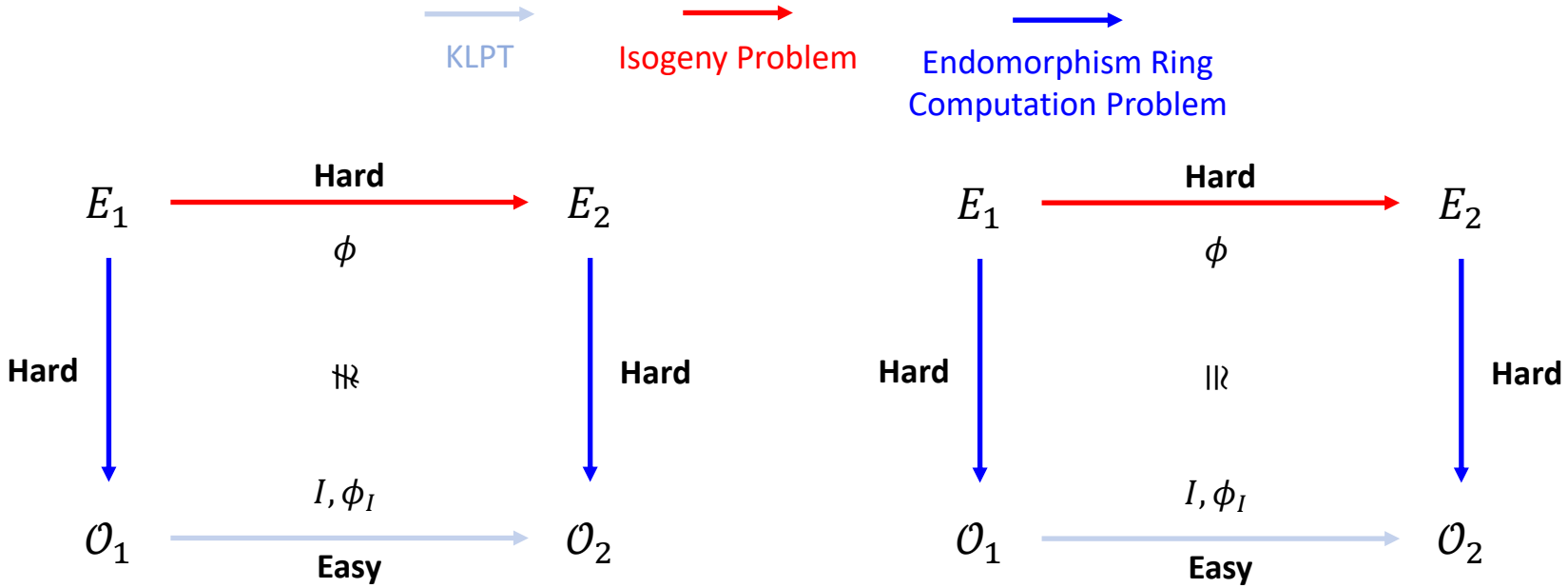
	SIKEp434	SIKEp503	SIKEp610	SIKEp751
보안강도	1	2	3	5
공격시간	62 m	2hr 19m	8h 15m	20h 37m

  - SIDH/SIKE는 완전히 깨진 것으로 간주
  - SIDH/SIKE 기반으로 설계된 암호들은 더 이상 사용할 수 없음



Recent attack in Isogeny-based Cryptography

- Key recovery attack (Castryck, Decru)
  - **Attack 에 사용된 핵심 :** Torsion-point 정보
    - 상대방에게 자신의 비밀값으로 연산된 함수값 전달
    - 이전부터 이를 활용한 공격에 관한 분석이 있음

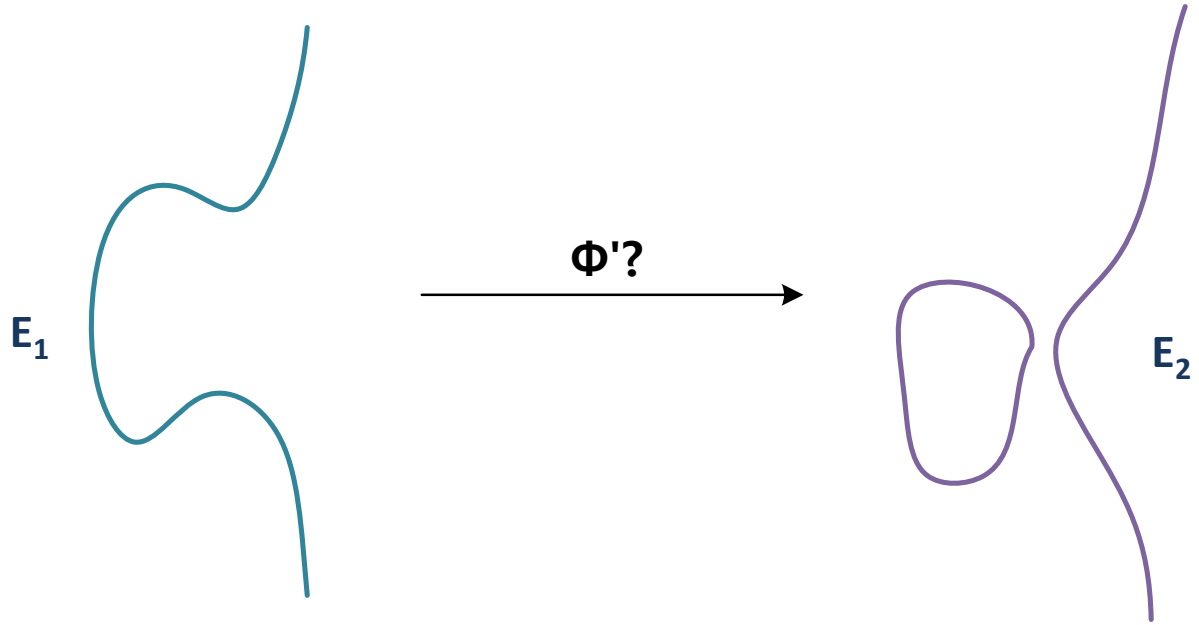


E_1, E_2 *only*

with torsion point information

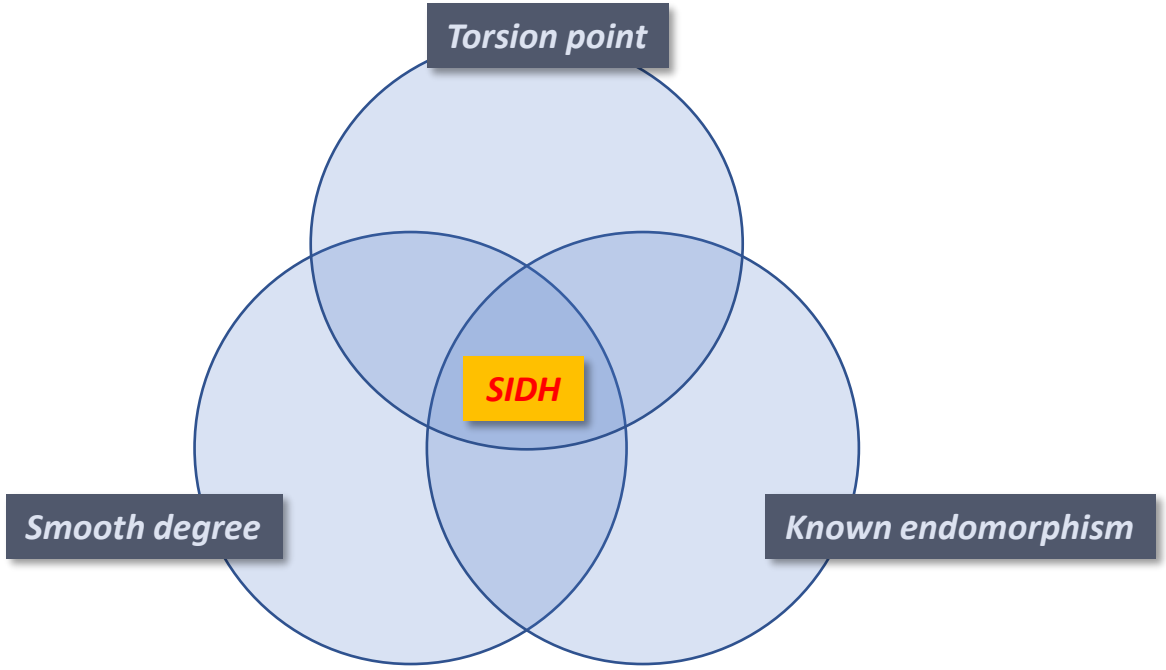
Recent attack in Isogeny-based Cryptography

- Key recovery attack (Castryck, Decru)
  - **Attack 에 사용된 핵심 :** Known smooth degree isogeny
    - Isogeny 기반 암호는 두 타원곡선 사이의 isogeny를 찾는것에 기반
    - 효율성을 위해 ℓ^n-isogeny 사용



Recent attack in Isogeny-based Cryptography

- Key recovery attack (Castryck, Decru)
  - **Attack 에 사용된 핵심**
    ① Torsion-point 정보 : 상대방에게 자신의 비밀값으로 연산된 함수값 전달
    ② Known smooth degree isogeny
    ③ Known endomorphism ring



Recent attack in Isogeny-based Cryptography

- Key recovery attack (Castryck, Decru)
  - **Attack 에 사용된 핵심**
    ① Torsion-point 정보 : 상대방에게 자신의 비밀값으로 연산된 함수값 전달
    ② Known smooth degree isogeny
    ③ Known endomorphism ring

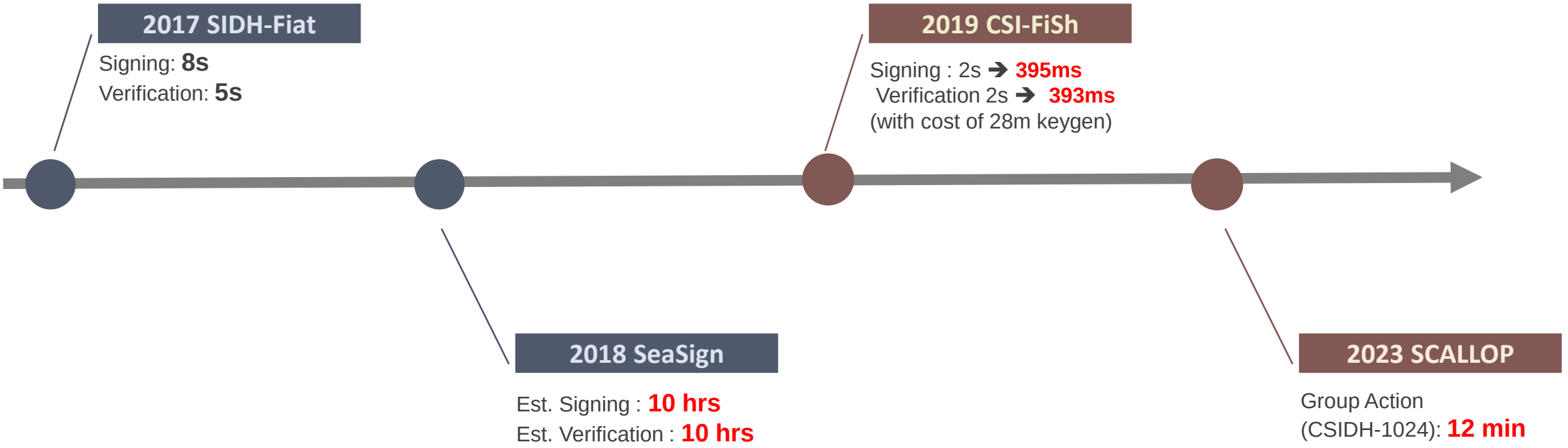
  - Impact on isogeny-based cryptography

	SIDH-based	SETA	CSIDH-based	SQISign	CGL
Algorithm	SIDH, SIKE, BSIDH, SIDH-Fiat		CSIDH, SeaSign CSI-FiSh	전자서명	**해시**
Result	**X**	**X**	**O**	**O**	**O**

CGL-Hash and FIBS

Outline

- ***Struggle*** for an efficient digital signature algorithm in isogeny-based cryptography

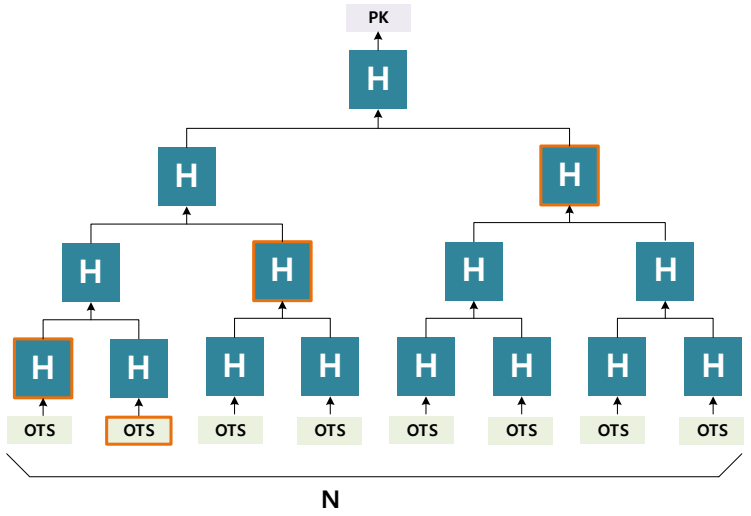


Outline

- ***Struggle*** for an efficient digital signature algorithm in isogeny-based cryptography

Hash-based Digital Signature

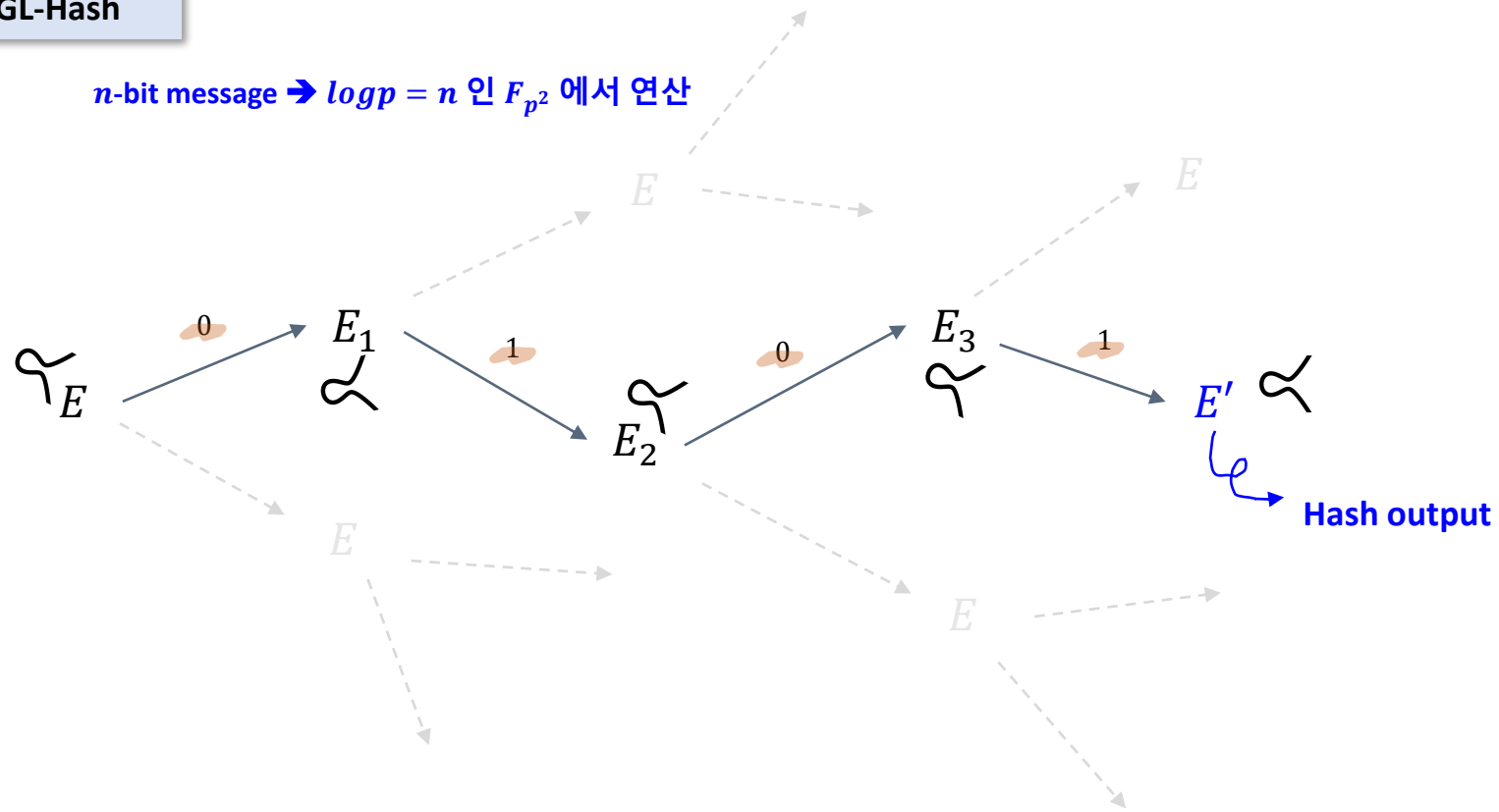
- Security base
  - 해시함수의 안전성에 의존
- ***Stateful HBS*** (i.e., LMS, XMSS) :
  - Include a 4-byte index value in their signature which represents the state.
  - The state is used when signing a message and should never be reused as that could allow for signature.
- ***Stateless HBS*** (i.e., SPHINCS+)
  - While stateless eliminates the need for proper state management.
  - Significant increase in signature size and slower performance because of the FTS(i.e., FORS) structure.



CGL-Hash

- 2009년 Charles, Goren, Lauter에 의해 제안
- ℓ-isogeny graph의 fast mixing property 이용
- 이후 충돌 가능성이 제안되어 2019년 Panny에 의해 개선됨

기존 CGL-Hash



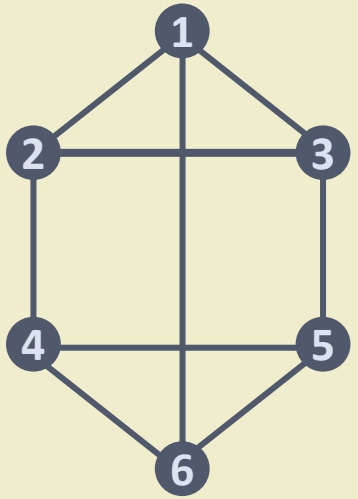
CGL-Hash

- IDEA
  - Provable hash
    - 충돌이 발생하는 것이 어느 어려운 문제를 푸는 것과 동일할 때
  - Charles, Goren, Lauter 는 expander graph를 이용한 해시 함수 제안
    - Input : Message ➔ expander graph에서 걷는 방향을 지시
    - Output : Ending Vertex
  - 제안하는 방법은 어느 expander graph를 사용해도 좋지만, 특히 ***Ramanujan graph***와 LPS graph 에 적용 가능성을 보임

CGL-Hash

- Ramanujan Graph $G = (V, E)$
  - G : k-regular graph, 꼭지점의 개수를 h 라 하자
  - G 의 adjacency matrix 를 A 라 하면 A 는 $h \times h$ 의 symmetric matrix 이다.

Example



3-regular graph

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

CGL-Hash

- Ramanujan Graph $G = (V, E)$
  - G : k-regular graph, 꼭지점의 개수를 h 라 하자
  - G 의 adjacency matrix 를 A 라 하면 A 는 $h \times h$ 의 symmetric matrix 이다.
  - A 의 eigenvalue는 $|\lambda| \leq k$ 를 만족함
  - Ramanujan graph는 특별한 형태의 expander graph로 $-k$ 가 아닌 non-trivial eigenvalue 에 대해 $|\lambda| \leq 2\sqrt{k-1}$ 를 만족시킴
  - Random walk에 대해 rapid mixing property를 가짐

CGL-Hash

- ℓ-Isogeny graph $\in F_{p^2}$
  - Nodes (Vertex)
    - $\bar{F}_q$-isomorphic 한 타원곡선
    - j-invariant로 표현
    - $\left\lfloor \frac{p}{12} \right\rfloor + \epsilon$ 개의 꼭지점이 존재 ($\epsilon \in \{0,1,2\}$)
  - Edge
    - 타원곡선 사이의 ℓ - isogeny

- ℓ-Isogeny graph 는 연결된 $\ell + 1$ – regular graph
- $|\lambda| \leq 2\sqrt{\ell}$ 을 만족시키는 Ramanujan graph

CGL-Hash

- Algorithm

Algorithm : CGL hash function

INPUT : message $m = (m_n, \cdots, m_0)_2$, starting curve $E \in F_{p^2}$
OUTPUT : $j(E) \in F_{p^2}$

1. While $m \neq 0$
2. if $m_i = 0$
3. $\phi: E \rightarrow E_A, \ker \phi = P_\alpha$
4. $E \leftarrow E_A$
5. else
6. $\phi: E \rightarrow E_A, \ker \phi = P_{1-\alpha}$
7. $E \leftarrow E_A$
8. $m = m \gg 1$

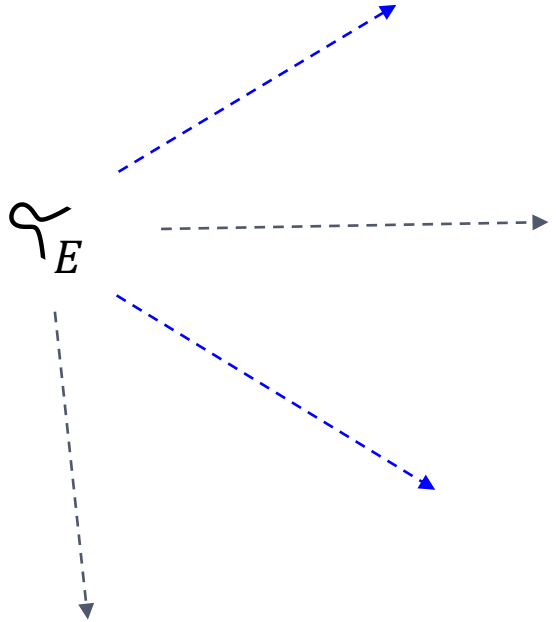
Return $j(E)$

CGL-Hash

New CGL-Hash

- 알려진 endomorphism ring 을 가지는 타원곡선을 사용해도 안전

n-bit message ➔ $logp = 2n$ 인 F_{p^2} 에서 연산



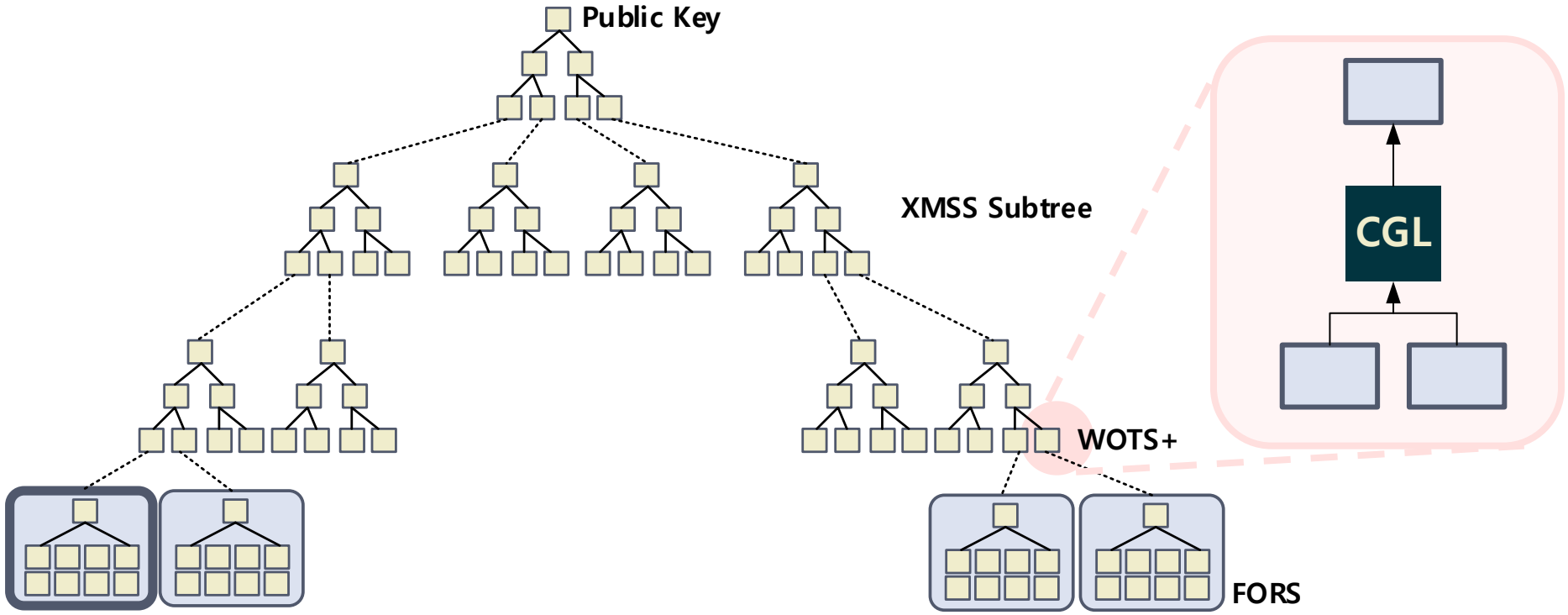
FIBS – Design Rationale

- FIBS
  - Fast Isogeny-Based Signature
  - **Isogeny-Hash**-based digital signature algorithm
  - ***Hash-based digital signature algorithm using isogeny-based hash***

CSI-FiSh (4096) << FIBS << SQI-Sign

The Algorithm

- SPHINCS + Isogeny hash



Parameters

- Parameters for SPHINCS

	n	h	d	$log\ t$	k	w
SPHINCS+_SHA256-128f-simple	16	66	22	6	33	16

- Parameters for CGL Hash
  - $p = 2^{607} - 1$
  - Generator
    - XPA : $5 + i$
    - XQA : $15 + i$
    - XRA :

Performance

- Performance
  - Intel Core i7-7700 @ 3.6 GHz, gcc v. 9.4.0

Classic 128

	SeaSign	FIBS	SQI-Sign
KeyGen	0.03 s	132.42 s	0.6 s
Sig Gen	36,372 s	3,158.44 s	2.5 s
Sig Ver	36,372 s	189.07 s	0.05 s

	SeaSign	FIBS	SQI-Sign
Public Key (byte)	64	32	64
Private Key (byte)	32	64	16
Signature (byte)	20,144	17,088	204

Conclusion and Future Work

- Conclusion
  - Isogeny 기반 암호는 (효율적인) 전자 서명을 만들기 위해 많은 연구 진행
    - ***FIBS***: 기존 알려진 공격에 대응할 수 있는 CGL-hash를 이용한 전자서명

- Security
  - 해시 기반 전자 서명의 안전성은 해시 함수의 안전성에 의존
  - 제안하는 파라미터에서 CGL-Hash 의 추가적인 안전성 분석 필요

- Future Work
  - CGL-hash 최적화 및 SPHINCS+에 최적 적용

Thank you