04

CSIDH
(Commutative – SIDH)

목차

1

CSIDH

CSIDH

Introduction

- 2006년 제안된 CRS scheme
  - Subexponential attack 존재
  - 하지만 비효율적인 속도가 가장 큰 문제
  - 비효율성의 원인은 ordinary curve의 사용

- 2018년 De Feo, Kieffer, Smith
  - CRS 기반 암호 파라미터 선택의 최적화
  - Ordinary curve 에서 효율적인 파라미터 선택 방법 제안

CSIDH

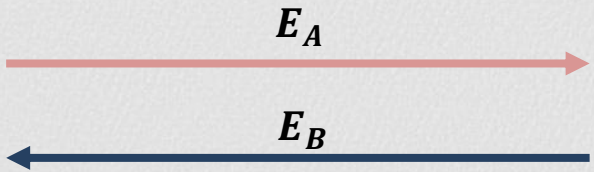
Introduction

- 2018년 CSIDH (Commutative SIDH)
  - 파라미터 선택의 문제를 F_p 에서 정의된 supersingular 곡선 사용으로 해결
  - Non-interactive key exchange 구성
  - CSI-FiSh의 효율적인 전자 서명 스킴 제안으로 CSIDH에 대한 연구 활발히 진행

CSIDH

Protocol

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

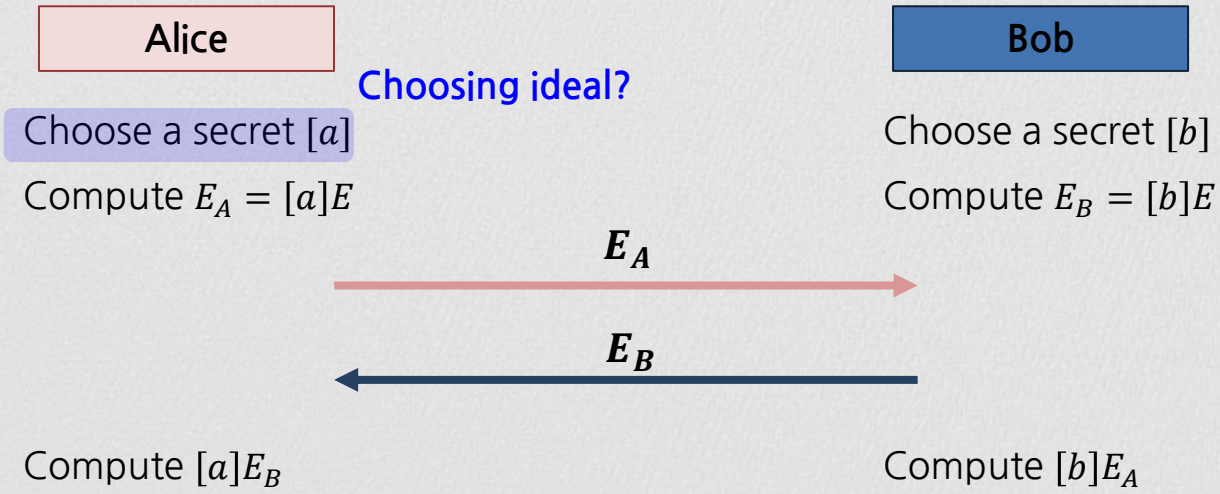


Compute $[a]E_B$	Compute $[b]E_A$

Shared Secret $[a][b]E = [a]E_B = [b]E_A$

CSIDH

Protocol

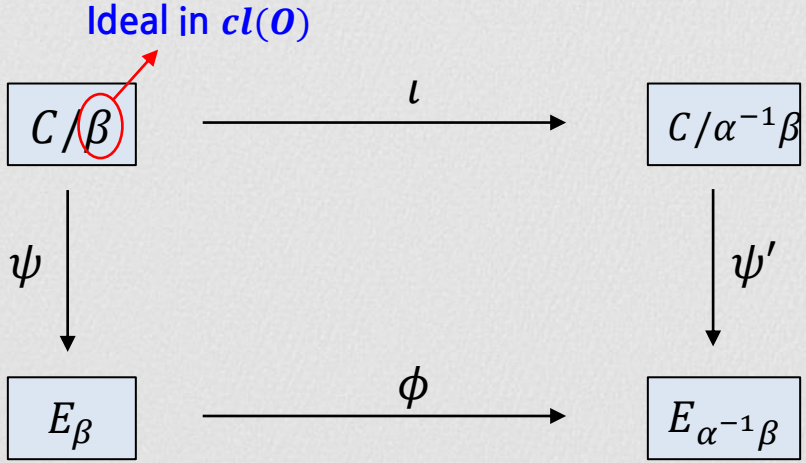


Shared Secret $[a][b]E = [a]E_B = [b]E_A$

- Computing $[\alpha]E$?
- Isogeny?

CSIDH

Computing the group action



$$cl(O) \times Ell_p(O,\pi) \to Ell_p(O,\pi)$$
$$([\alpha], E) \to [\alpha]E$$
$$\phi_\alpha : E \to \alpha E$$

$$ker\,\phi_\alpha = E[\alpha]$$
$$\deg \phi_\alpha = N(\alpha)$$

CSIDH

Computing $[\alpha]E$

- 임의의 ideal에 대한 $[\alpha]E$ 를 연산하는 것은 어려움
- $[\alpha] = [p_1]^{e_1} \cdots [p_n]^{e_n}$ 으로 표현되면 연산하는 것이 쉬움
  - 각 $[p_i]^{e_i}$ 에 대한 isogeny 연산을 합성하면 됨

$$\phi_\alpha = \phi_{p_1^{e_1}} \circ \cdots \phi_{p_n^{e_n}}$$

  - Velu formula 를 사용

CSIDH

Computing $[\alpha]E$

- Computing $[p_i]^{e_i}E$
  - Order 가 p_i 인 점 P 선택
  - Velu formula 이용
    - $\langle P \rangle$ 를 커널로 하는 아이소제니 e_i 번 연산

- Computing $[p_1]^{e_1}[p_2]^{e_2}E$
  - 위 방법을 이용해서 $[p_1]^{e_1}E$ 연산
  - $E' = [p_1]^{e_1}E$ ➔ $[p_2]^{e_2}E'$ 연산

CSIDH

Selecting the private key

- 개인키 : $[\alpha] \in cl(O)$
- 일반적으로 랜덤하게 선택하는 것은 어려움
- 사용하려는 field의 특성상
  - $[\alpha] = [\ell_1]^{e_1} \cdots [\ell_n]^{e_n}$ for all $[\alpha] \in cl(O)$
  - $[\alpha] \approx (e_1, \cdots, e_n)$
  - Select random e_i

CSIDH

SIDH vs CSIDH

	SIDH	CSIDH
Field	$F_{p^2}, p = \ell_A^{e_A} \ell_B^{e_B} f - 1$	$F_p, p = f \cdot \prod_{i=1}^{\ell} p_i - 1$
Public Key	$P \in F_p, Q,$ $R(= P - Q) \in F_{p^2}$	A (Montgomery curve coefficient)
Private Key (uncompressed)	m_A, n_A	$[a] = (e_1, \cdots, e_\ell),$ $e_i \in [-B, B]$
Shared Key	$j(E_{AB}) = j(E_{BA})$	AB

CSIDH

SIDH vs CSIDH

SIDHp503 (classical 126)	Form	Size
Field	$F_{p^2}, p = 2^{250}3^{159} - 1$	125 byte
Public Key	$P \in F_p, Q \in F_{p^2}, R(= P - Q) \in F_{p^2}$	312 byte
Private Key (uncompressed)	m_A, n_A	125 byte
Shared Key	$j(E_{AB}) = j(E_{BA})$	125 byte

CSIDH-512 (classical 128)	Form	Size
Field	$F_p, p = 2^2 \cdot 3 \cdots 587 - 1$	64 byte
Public Key	A (Montgomery curve coefficient)	64 byte
Private Key (uncompressed)	$[a] = (e_1, \cdots, e_{74}), e_i \in [-5,5]$	-
Shared Key	AB	64 byte

CSIDH

Performance

$$p_1 = 4 \cdot \underbrace{(3 \cdot \ldots \cdot 373)}_{\text{73 first odd primes}} \cdot 587 - 1 \approx 2^{510.668},$$

$$p_2 = 4 \cdot 2^2 \cdot 3^2 \cdot 11 \cdot \underbrace{(3 \cdot \ldots \cdot 373)}_{\text{73 first odd primes}} - 1 \approx 2^{510.100}.$$

	CSIDH [6]	Onuki's [13]	Hybrid [10]	CSURF [14]
p_1	32.39 ms 110,401,470 cc	30.77 ms 104,866,008 cc	28.68 ms 97,752,709 cc	- -
p_2	29.62 ms 100,945,178 cc	- -	28.01 ms 95,480,448 cc	39.38 ms 134,216,582 cc
Interval	$[-5,5]^{74}/[-5,5]^{73}$	$[-1,1] \times [-5,5]^{73}$	$[-5,5]^{74}/[-5,5]^{73}$	$[-137,137] \times [-4,4]^3 \times [-5,5]^{45} \times [-4,4]^{25}$
Security	255.998/252.536	254.123	255.998/252.536	252.535

"On the performance analysis for CSIDH-based cryptosystems"

2

CSIDH의 안전성

Security of CSIDH

CSIDH 에 대한 안전성 분석

- CSIDH에 대한 공격은 subexponential 공격으로 최근에 면밀한 양자 공격 분석이 이루어짐
  - Quantum security analysis of CSIDH (EUROCRYPT 2020)
  - He gives C-Sieves on the CSIDH (EUROCRYPT 2020)

- 분석 결과 기존 CSIDH-512를 사용할 수 없으며, 양자 환경에서 128비트 보안강도를 맞추기 위해 유한체 크기를 4096비트까지 늘려야 함

Security of CSIDH

CSIDH 에 대한 안전성 분석

- The SQALE of CSIDH - Jorge Chavez-Saab et al. (ePrint 2020)

	기존 (prime size)	Sqale-CSIDH	
		Prime size	Performance
Level 1	512	4096	23.2 Gigacycle
Level 2	1024	6144	74.8 Gigacycle
Level 3	1792	-	-
Level 5	-	9216	292.4 Gigacycle