同種写像暗号 2
鍵交換方式SIDHとCSIDH

守谷　共起（東京大学）

2022年8月2日

目次

SIDH
1. プロトコル
2. SIDHベースの公開鍵暗号
3. SIDHへのattack

CSIDH
1. プロトコル
2. IND-CCA安全なCSIDHベースの公開鍵暗号
3. CSIDHベースの電子署名

目次

Diffie-Hellman 鍵共有（復習）

離散対数問題
素数位数の巡回群の生成元 g, g^a から a を求めるのは困難
（古典コンピュータの場合）

AliceとBobが秘密の値 g^{ab} を共有する

同種写像暗号版 Diffie-Hellman

同種写像問題
超特異楕円曲線の組 E_0, E_1 から $\phi: E_0 \to E_1$ を求めるのは困難

AliceとBobが秘密の楕円曲線 E_{AB} を共有する

SIDH [(Jao, De Feo) PQCRYPTO 2011] (1/3)
Supersingular Isogeny Diffie-Hellman

補助情報（楕円曲線の点）を使って可換図式を作る

公開パラメータ

素数 $p = 2^{e_A}3^{e_B} - 1$

超特異楕円曲線 $E_0: y^2 = x^3 + x$

$E_0[2^{e_A}] \cong (\mathbb{Z}/2^{e_A}\mathbb{Z})^2$ の生成元 P_A, Q_A

$E_0[3^{e_B}] \cong (\mathbb{Z}/3^{e_B}\mathbb{Z})^2$ の生成元 P_B, Q_B

秘密鍵

Alice: $k_A \in (\mathbb{Z}/2^{e_A}\mathbb{Z})^{\times}$ $(R_A = P_A + k_A Q_A)$

Bob : $k_B \in (\mathbb{Z}/3^{e_B}\mathbb{Z})^{\times}$ $(R_B = P_B + k_B Q_B)$

SIDH [(Jao, De Feo) PQCRYPTO 2011] (2/3)

Supersingular Isogeny Diffie-Hellman

公開鍵

Alice: $\phi_A: E_0 \to E_A \coloneqq E_0/\langle R_A \rangle,\ \ \phi_A(P_B), \phi_A(Q_B)$

Bob：$\phi_B: E_0 \to E_B \coloneqq E_0/\langle R_B \rangle,\ \ \phi_B(P_A), \phi_B(Q_A)$

共有鍵

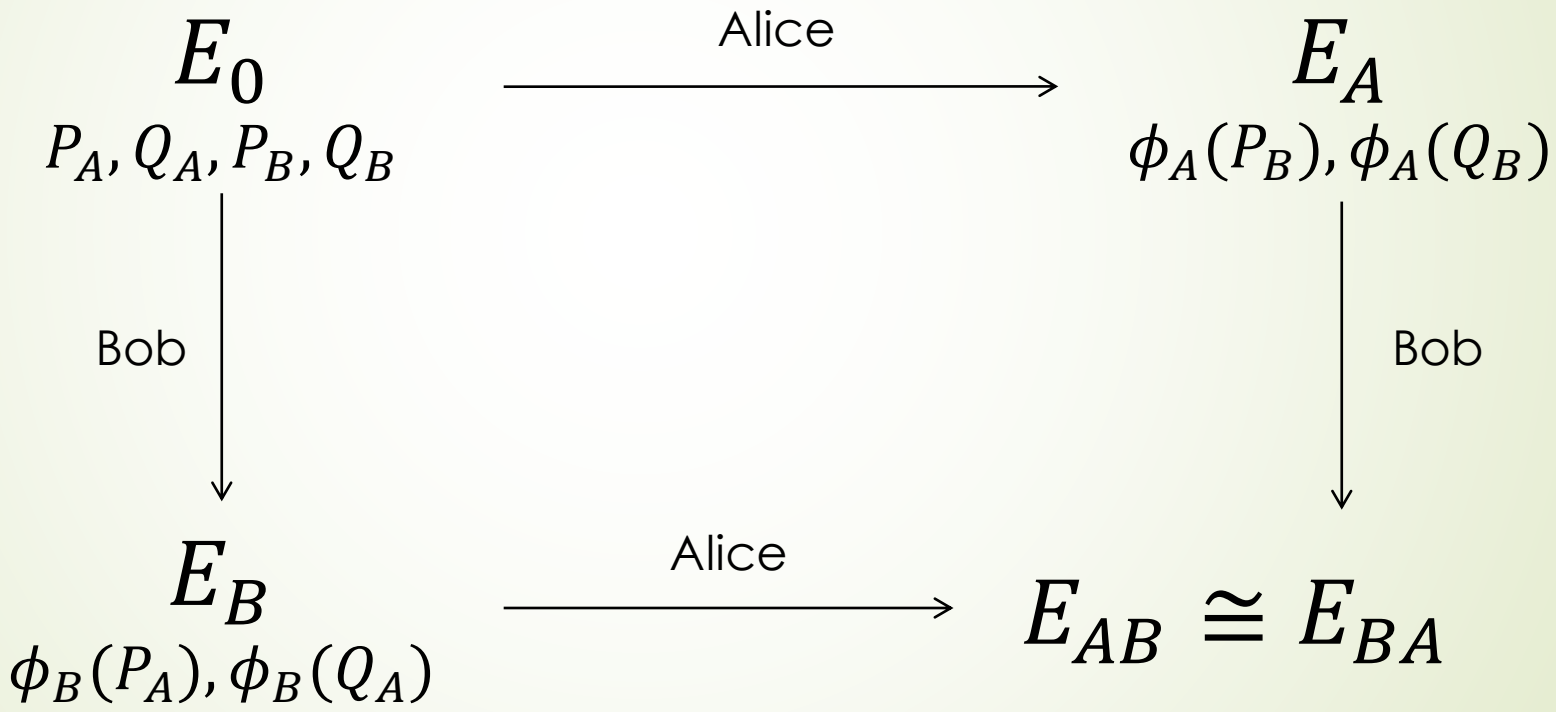
Alice: $\phi_A': E_B \to E_{BA} \coloneqq E_B/\langle \phi_B(P_A) + k_A\phi_B(Q_A) \rangle$

Bob：$\phi_B': E_A \to E_{AB} \coloneqq E_A/\langle \phi_A(P_B) + k_B\phi_A(Q_B) \rangle$

$$j(E_{BA}) = j(E_{AB})$$

SIDH [(Jao, De Feo) PQCRYPTO 2011] (3/3)

Supersingular Isogeny Diffie-Hellman



目次

SIDHベースの公開鍵暗号（SIKE.PKE）

公開鍵：$E_0, E_A, P_A, Q_A, P_B, Q_B, \phi_A(P_B), \phi_A(Q_B)$

秘密鍵：k_A

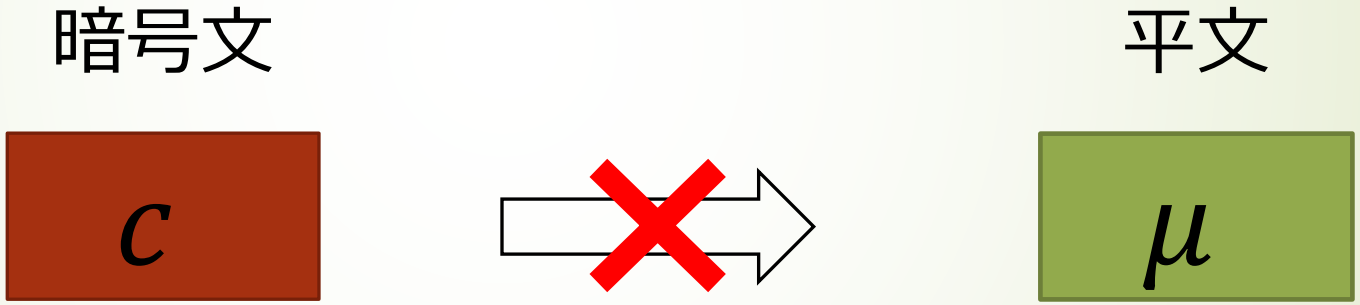
平文　：μ

暗号文：$(E_B, \phi_B(P_A), \phi_B(Q_A), \mu \oplus H(j(E_{AB})))$

復号　：k_A により $j(E_{BA})$ を計算.
$(\mu \oplus H(j(E_{AB}))) \oplus H(j(E_{BA}))$ を計算することで μ を得る.

OW-CPA安全性

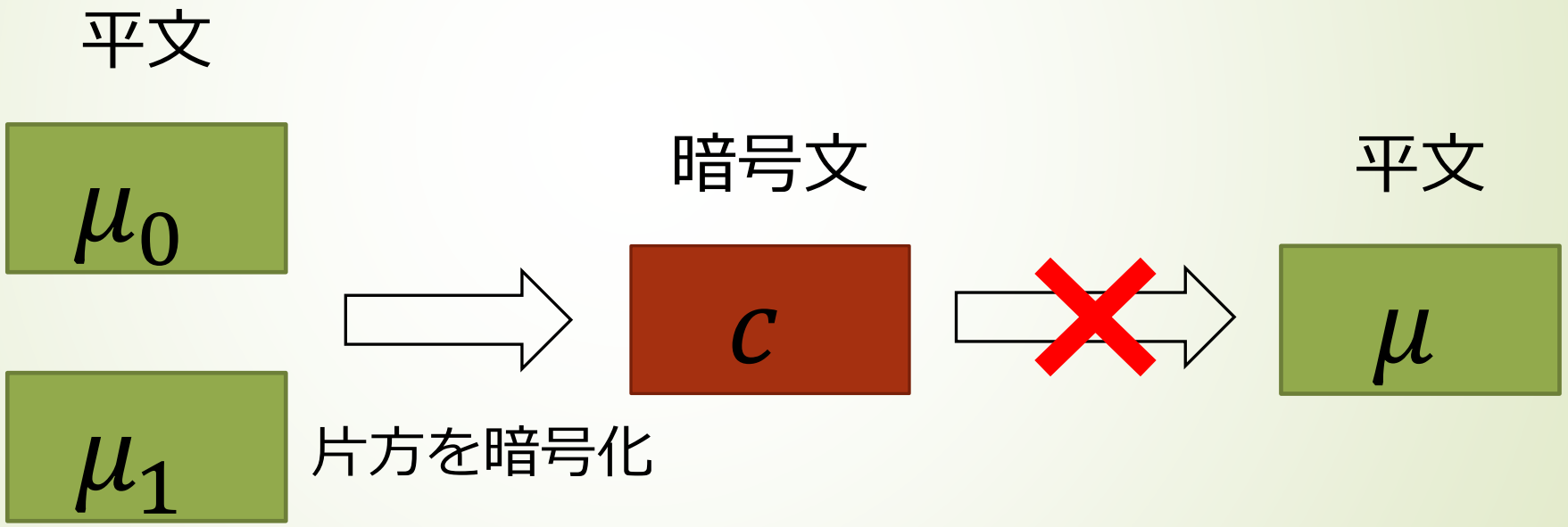
定義
（OW-CPA安全（onewayness under chosen plaintext attack））



攻撃者は暗号文から完全な平文が得られない

IND-CPA安全性 (1/2)

定義
（IND-CPA安全（indistinguishability under chosen plaintext attack））



攻撃者はどちらの平文が暗号化されたのかわからない

IND-CPA安全性 (2/2)

IND-CPA安全の気持ち

暗号文から平文の情報が漏れないことを暗に含んでいる

暗号文から平文の情報が部分的に漏れているとする

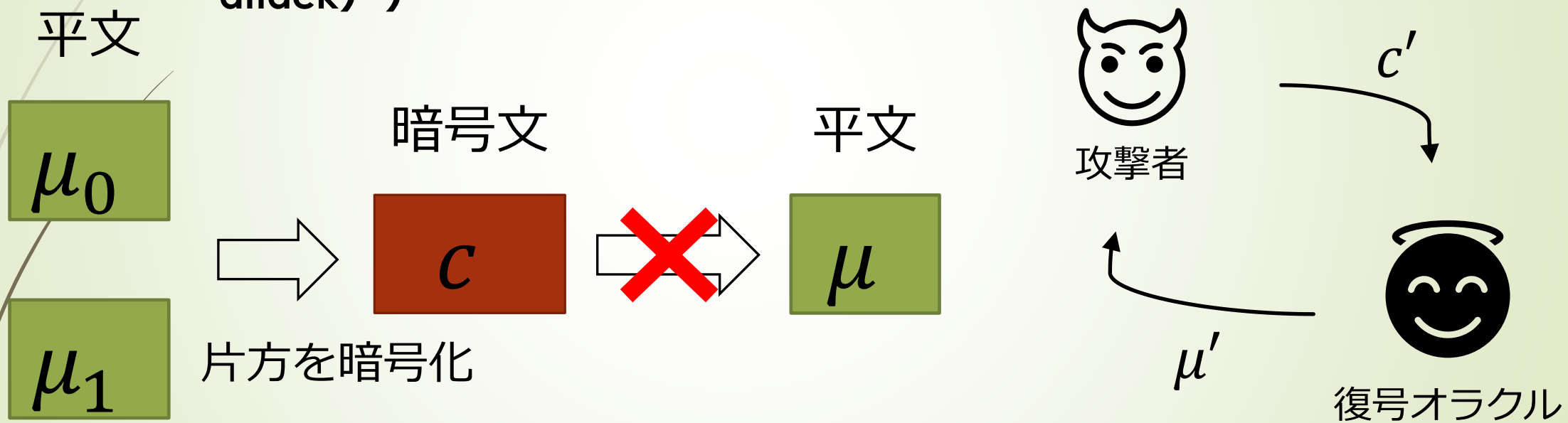
⇨ c_i から平文の情報を抜く

⇨ 抜いた情報を元の平文 (μ_0, μ_1) と比較する

⇨ どっちの平文を暗号化したのか判定できる

IND-CCA安全性 (1/2)

定義
（IND-CCA安全（indistinguishability under chosen ciphertext attack））



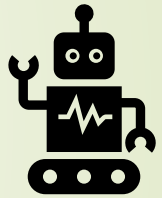
復号オラクルを有している攻撃者も
どちらの平文が暗号化されたのかわからない

IND-CCA安全性(2/2)

IND-CCA安全の気持ち

システムに組み込まれていても安全な暗号

復号された平文の内容によって挙動が変わるシステム



⇨ システムの情報から平文の情報が部分的にわかる

⇨ 元の暗号文を変形した暗号文で挙動を確認する

⇨ IND-CCAでないと元の暗号文が破られることがある

SIDHの安全性仮定

SSICDH (Supersingular Isogeny CDH)

$E_0, E_A, E_B, P_A, Q_A, P_B, Q_B, \phi_A(P_B), \phi_A(Q_B), \phi_B(P_A), \phi_B(Q_A)$ から $j(E_{AB})$ を計算する確率的多項式時間アルゴリズムは存在しない

→ SIKE.PKE （OW-CPA安全)

SSIDDH (Supersingular Isogeny DDH)

$E_0, E_A, E_B, P_A, Q_A, P_B, Q_B, \phi_A(P_B), \phi_A(Q_B), \phi_B(P_A), \phi_B(Q_A)$ から $j(E_{AB})$ とランダムな超特異楕円曲線を判別する確率的多項式時間アルゴリズムは存在しない

→ SIKE.PKE （IND-CPA安全)
→ SIKE.KEM (IND-CCA安全)

目次

SIDH

1. プロトコル
2. SIDHベースの公開鍵暗号
3. **SIDHへのattack**

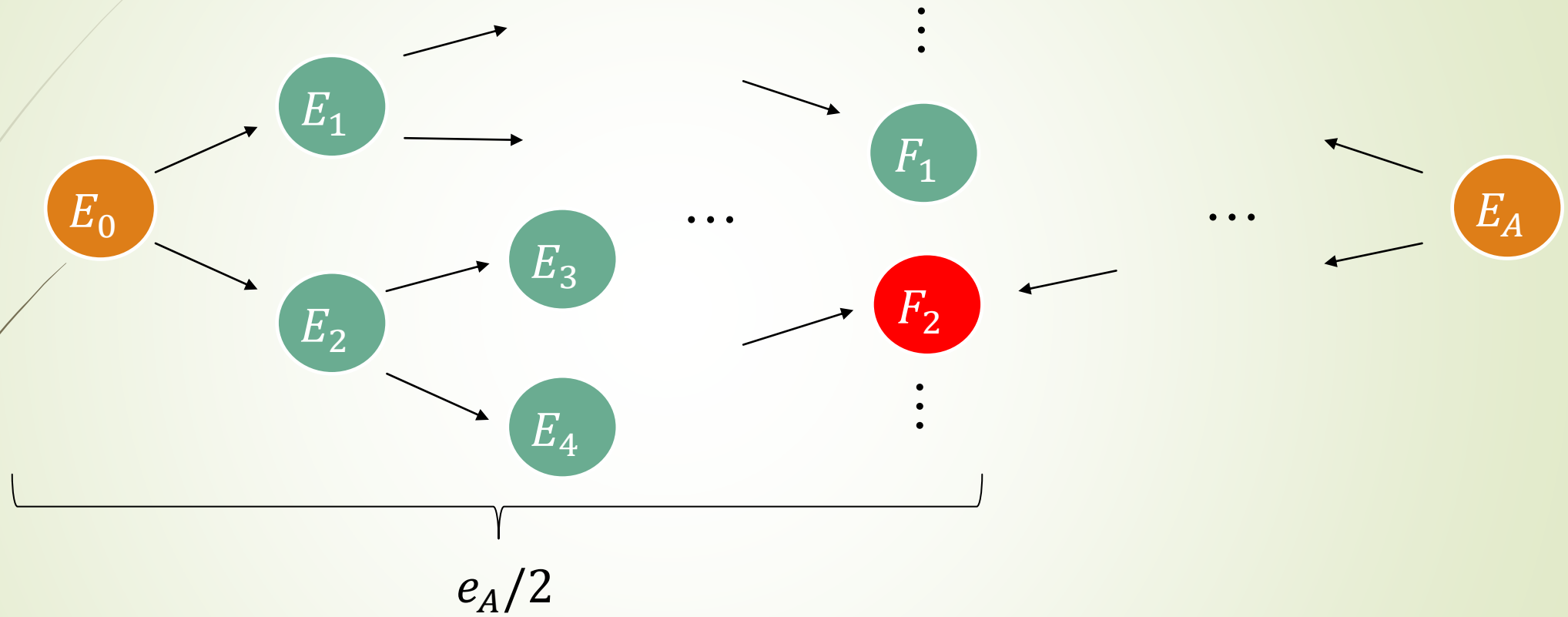
CSIDH

1. プロトコル
2. IND-CCA安全なCSIDHベースの公開鍵暗号
3. CSIDHベースの電子署名

SIDHに対するattack

- Meet in the middle (Craw finding algorithm)
- Torsion point attack
- GPST attack
- CD attack [Castryck and Dedru]
  破られました！！！！！！！！！！！！！ (2022/07/30)

Meet in the middle (Craw finding)



計算量 (古典)：$O\left(2^{e_A/2}\right) = O(p^{1/4})$

計算量 (量子)：$O(p^{1/6})$ [Tani (TCS 2009)]

Torsion point attack
[Petit (ASIACRYPT 2017)] [Quehen et al. (CRYPTO 2021)]

条件

- パラメータ p がun-balanced $(3^{e_B} \approx (2^{e_A})^4)$
- 超特異楕円曲線 E_0 の自己準同型環の構造がわかっていて小さい次数の非自明な自己準同型を持つ
  $(y^2 = x^3 + x$ など$)$
- $\phi_A(P_B), \phi_A(Q_B)$ がわかっている

この条件下で，SIDHは多項式時間で解ける

→ SIDHのパラメータは $2^{e_A} \approx 3^{e_B}$ が成り立つように定める.

GPST attack
[Galbraith, Petit, Shani, and Ti (ASIACRYPT 2016)]

前提
- BobがAliceの秘密鍵を割り出そうとしている
- Aliceは同じ秘密鍵を何度でも使いまわす
- BobはAliceと正しく鍵共有ができたかどうか判別できる

1. BobがAliceに細工した共有鍵を送る
2. Aliceと同じ秘密鍵が共有できたか否かを判別して
   Aliceの秘密鍵の情報を得る
3. これを繰り返して完全に鍵を得る

→ Aliceは鍵を使いまわしてはいけない
→ SIDHベースのIND-CCAのPKEは作れない（と思われている）

SIDHのまとめ

補助情報（楕円曲線の点）を使って可換図式を作った
Diffie-Hellman 型の鍵共有方式

いい点
- （同種写像暗号の中では）計算コストが小さい
- 小さい素数で高い安全性を実現できる

悪い点
- 補助情報を利用した攻撃がある
- IND-CCAのPKEが作れない（と思われる）
- 高機能なプロトコルが作りにくい

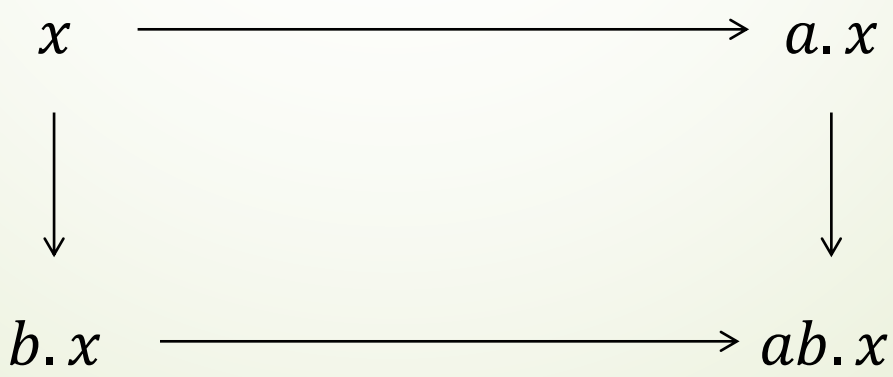
目次

Hard homogenous space [JM Couveignes (2016)]

定義

有限可換群 G が集合 X に作用していて，以下の条件を満たすとき，(G,X) を hard homogenous space と呼ぶ.

- 作用は自由かつ推移的
- ランダムな G の元を取ることができる
- 作用の計算が現実的な時間で行える
- x，$g.x$ が与えられたとき，g を計算するのが困難



CSIDH鍵共有 (1/4)

[Castryck et al. (ASIACRYPT 2018)]

可換群の超特異楕円曲線の同型類の集合への作用に基づいた鍵共有方式
群作用で可換図式を作る

AliceとBobが秘密の値 $[\mathfrak{a}][\mathfrak{b}]E_0$ を共有する

CSIDH鍵共有 (2/4)

定義（イデアル類群）

K : 虚2次体
$\mathcal{O}$: K の整環 (CSIDHの設定では $\mathcal{O} = \mathbb{Z}[\sqrt{-p}]$)

$\{0\}$ でない $\mathfrak{a} \subset K$ に対して，$\alpha\mathfrak{a}$ が $\mathcal{O}$ のイデアルとなるような $\alpha \in \mathcal{O} \setminus \{0\}$ が存在するとき，$\mathfrak{a}$ を $\mathcal{O}$ の分数イデアルと呼ぶ.

$P(\mathcal{O})$:= 可逆な分数イデアルのなす群（i.e. $\mathfrak{ab} = \mathfrak{ba} = \mathcal{O}$）
$I(\mathcal{O})$:= 単項生成分数イデアルのなす群

$P(\mathcal{O})/I(\mathcal{O})$ を $\mathcal{O}$ の**イデアル類群**と呼ぶ.
代表元に $\mathfrak{a}$ を持つイデアル類群の元を $[\mathfrak{a}]$ で表す.

CSIDH鍵共有 (3/4)

定理 [Waterhouse (Annales scientiques de l'Ecole Normale Superieure 1969)]

π_p : p -Frobenius写像　$\pi_p(X:Y:Z) = (X^p:Y^p:Z^p)$

$\mathcal{E}\ell\ell_p(\mathbb{Z}[\pi_p])$: $\mathrm{End}_p(E) \cong \mathbb{Z}[\pi_p]$ $(\cong \mathbb{Z}[\sqrt{-p}])$ をみたす$\mathbb{F}_p$上定義された超特異楕円曲線Eの$\mathbb{F}_p$同型類の集合

$\mathrm{cl}(\mathbb{Z}[\pi_p])$: $\mathbb{Z}[\pi_p]$のイデアル類群

$\mathrm{cl}(\mathbb{Z}[\pi_p])$ は下の作用で $\mathcal{E}\ell\ell_p(\mathbb{Z}[\pi_p])$ に自由かつ推移的に作用する.

$$\mathrm{cl}(\mathbb{Z}[\pi_p]) \times \mathcal{E}\ell\ell_p(\mathbb{Z}[\pi_p]) \to \mathcal{E}\ell\ell_p(\mathbb{Z}[\pi_p])$$
$$([\mathfrak{a}], E) \mapsto [\mathfrak{a}]E := E/E[\mathfrak{a}]$$

ただし，$\mathfrak{a}$ は整イデアルとし，$E[\mathfrak{a}] = \bigcap_{\alpha\in\mathfrak{a}} \ker \alpha$.

CSIDH鍵共有 (4/4)

$p : p = 4l_1 \cdots l_n - 1$ を満たす素数．ただし，$l_1, \cdots, l_n$は互いに異なる奇素数.

Heuristics

- $\mathrm{cl}(\mathbb{Z}[\pi_p])$ がほぼ巡回群になる(Cohen-Lenstra Heuristics [Cohen and Lenstra (1983)])
- $[\mathfrak{l}_1], \ldots, [\mathfrak{l}_n]$ の1つ以上が $\mathrm{cl}(\mathbb{Z}[\pi_p])$をほぼ生成する．$\mathfrak{l}_i = (l_i, \pi_p - 1)$.
- Gaussian Heuristic

◎ $[\mathfrak{l}_1]^{e_1} \cdots [\mathfrak{l}_n]^{e_n}$と表せる元により$\mathrm{cl}(\mathbb{Z}[\pi_p])$ の多くが尽くされる.
ここで，$e_1, \ldots, e_n$は$\{-m, \ldots, m\}$の元．mは$2m + 1 \geq \sqrt[2n]{p}$を満たす最小の整数.

⇨ 入力を$(e_1, \ldots, e_n)$として$[\mathfrak{l}_1]^{e_1} \cdots [\mathfrak{l}_n]^{e_n}$の作用を計算する.

CSIDHに対するattack

- Meet in the middle (古典)
- Kuperberg アルゴリズム (量子)

Meet in the middle : $O(p^{1/4})$

Kuperberg アルゴリズム : $L_N\left[\frac{1}{2}, \sqrt{2}\right]$

$$= \exp\left(\left(\sqrt{2} + o(1)\right)\sqrt{\log N \operatorname{loglog} N}\right)$$

ただし， $N \approx \sqrt{p}$

CSIDHのまとめ

群作用を使って可換図式を作った
Diffie-Hellman 型の鍵共有方式

いい点
- 補助情報がないため，特殊なattackがなさそう
- IND-CCA安全な公開鍵暗号が作れる（後述）
- 高機能なプロトコルが作りやすい（後述）

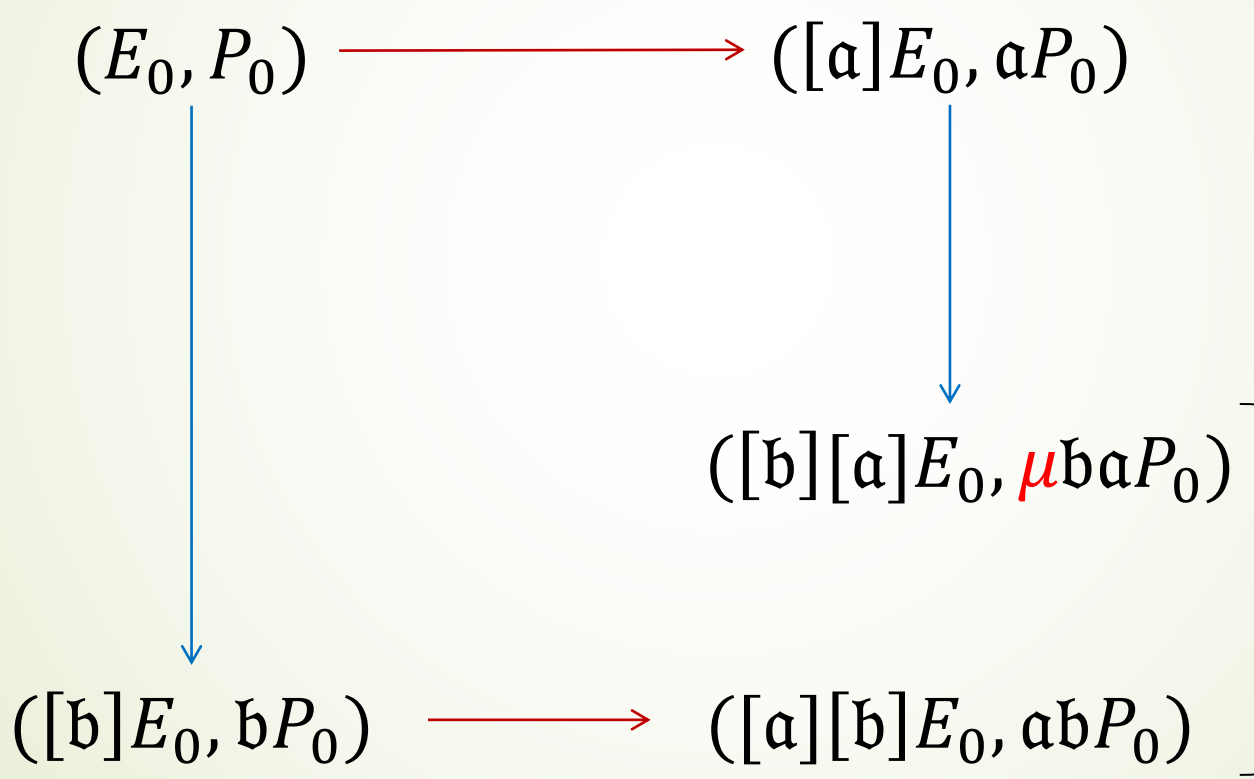
悪い点
- 素数のサイズが大きくなる
- 計算に時間がかかる

目次

SIDH
1. プロトコル
2. SIDHベースの公開鍵暗号
3. SIDHへのattack

CSIDH
1. プロトコル
2. **IND-CCA安全なCSIDHベースの公開鍵暗号**
3. CSIDHベースの電子署名

SiGamal (1/2)
[Moriya, Onuki, and Takagi (ASIACRYPT 2020)]



離散対数問題
μを得る

位数がsmoothな場合は容易に解ける
[Pohlig and Hellman (1978)]

SiGamal (2/2)

$p : p = 2^r l_1 \cdots l_n - 1$ を満たす素数.
ただし，$l_1, \cdots, l_n$は互いに異なる奇素数.
P_0 : 位数が2^rの$E_0(\mathbb{F}_p)$ の点.

公開鍵： $(E_0, P_0, [\mathfrak{a}]E_0, \mathfrak{a}P_0)$

秘密鍵： $\mathfrak{a}$

平文　： $\mu \in (\mathbb{Z}/2^r\mathbb{Z})^\times$

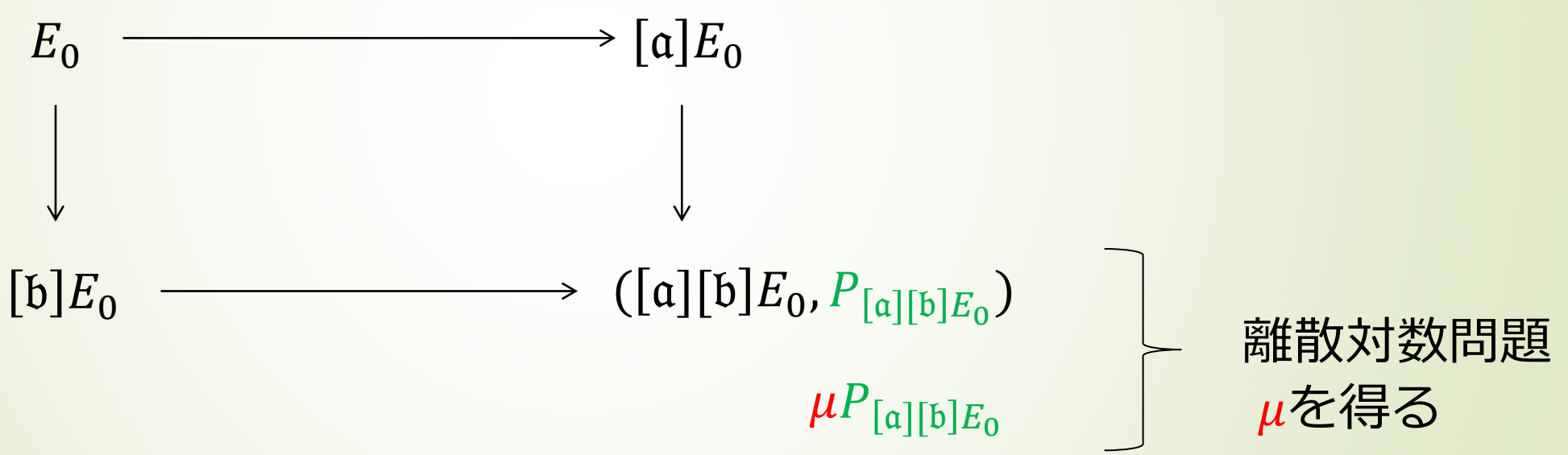
暗号文： $([\mathfrak{b}]E_0, \mathfrak{b}P_0, [\mathfrak{a}][\mathfrak{b}]E_0, \mu\mathfrak{a}\mathfrak{b}P_0)$

復号　： $\mathfrak{a}$ により $([\mathfrak{a}][\mathfrak{b}]E_0, \mathfrak{a}\mathfrak{b}P_0)$ を計算.
Pohlig-Hellmanアルゴリズムを用いてμ を得る.

P-CSSIDDH仮定の下，IND-CPA安全

SimS (1/3)
[Fouotsa and Petit (PQCRYPTO 2021)]

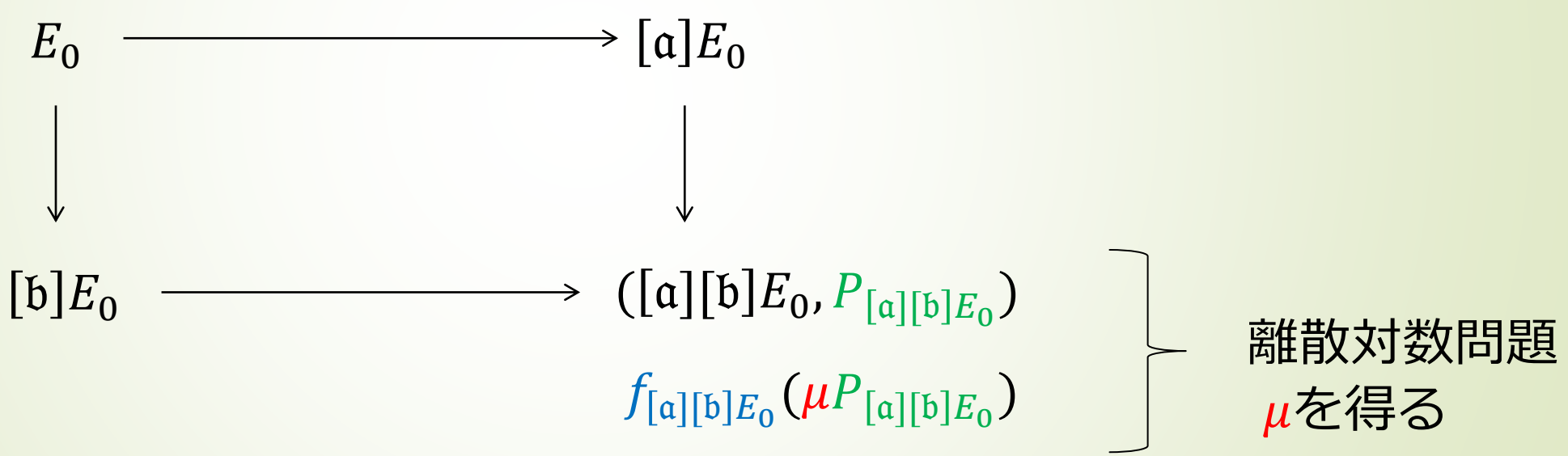
SimS (Simplification of SiGamal)



P_E : 楕円曲線 E から定まる位数 2^r の点.
構成方法は共有されている.

SimS (2/3)

SimS (Simplification of SiGamal)



$f_E : x \mapsto x \oplus A_E$　　(A_E は E の係数)

SimS (3/3)

$p : p = 2^r l_1 \cdots l_n - 1$ を満たす素数.
ただし，$l_1, \cdots, l_n$は互いに異なる奇素数.

公開鍵： $(E_0, [\mathfrak{a}]E_0)$

秘密鍵： $[\mathfrak{a}]$

平文　： $\mu \in (\mathbb{Z}/2^r\mathbb{Z})^\times$

暗号文： $([\mathfrak{b}]E_0, f_{[\mathfrak{a}][\mathfrak{b}]E}(\mu P_{[\mathfrak{a}][\mathfrak{b}]E_0}))$

復号　： $[\mathfrak{a}]$ により $[\mathfrak{a}][\mathfrak{b}]E_0$ を計算．$f^{-1}_{[\mathfrak{a}][\mathfrak{b}]E_0}$ から $\mu P_{[\mathfrak{a}][\mathfrak{b}]E_0}$ を計算.
$\mu P_{[\mathfrak{a}][\mathfrak{b}]E_0}$ の位数が 2^r でない場合，復号しない.
位数が 2^r でないなら，離散対数問題を解いて μ を得る.

CSSIDDH仮定，CSSIKoE仮定の下，IND-CCA安全

目次

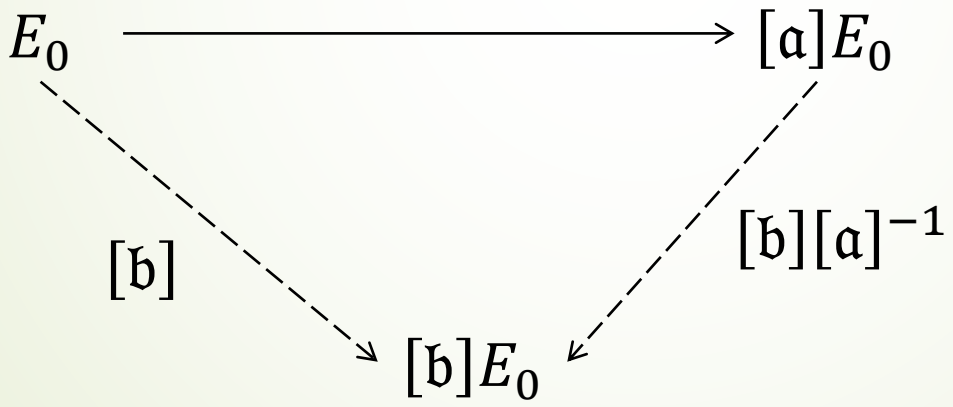
SIDH
1. プロトコル
2. SIDHベースの公開鍵暗号
3. SIDHへのattack

CSIDH
1. プロトコル
2. IND-CCA安全なCSIDHベースの公開鍵暗号
3. **CSIDHベースの電子署名**

CSIDHによる署名

$$E_0 \longrightarrow [\mathfrak{a}]E_0$$

$[\mathfrak{a}]$ を知っていることを $[\mathfrak{a}]$ を漏らさずに証明したい



ランダムな $[\mathfrak{b}]E_0$ に対して，相手が指定した方の同種を必ず出せる

$[\mathfrak{a}]$ を知らなければ，t 回全てで正解できるのは $1/2^t$ の確率

SeaSign

[De Feo and Galbraith (EUROCRYPT 2019)]

公開鍵　：$(E_0, [\mathfrak{a}]E_0)$

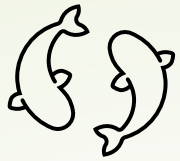
秘密鍵　：$[\mathfrak{a}]$

署名生成：$[\mathfrak{b}_1]E_0, \dots, [\mathfrak{b}_t]E_0$ をランダムに計算
$b_1 b_2 \dots b_t := H([\mathfrak{b}_1]E_0, \dots, [\mathfrak{b}_t]E_0, \mu)$（$\mu$ はメッセージ）
$b_i = 0$ なら $[\mathfrak{c}_i] := [\mathfrak{b}_i]$，$b_i = 1$ なら $[\mathfrak{c}_i] := [\mathfrak{b}_i][\mathfrak{a}]^{-1}$
$\sigma := ([\mathfrak{c}_1], \dots, [\mathfrak{c}_t], b_1 b_2 \dots b_t)$

署名検証：$b_i = 0$ なら $E_i := [\mathfrak{c}_i]E_0$
$b_i = 1$ なら $E_i := [\mathfrak{c}_i][\mathfrak{a}]E_0$ を計算
$b_1 b_2 \dots b_t = H(E_1, \dots, E_t, \mu)$ なら正しい署名

→ イデアル類群はベクトルで与えられるため，署名サイズが大きい

CSI-FiSh



[Beullens, Kleinjung, and Vercauteren (ASIACRYPT 2019)]

イデアル類群の構造を決定することで
SeaSignの署名のデータサイズを小さくし，さらに高速化した方式

→ p が512ビットのケース（CSIDHの基本的なパラメータ）
でイデアル類群の構造を決定（計算機の利用）

→ p が512ビットのケースでデータサイズが小さくできる

当初は512ビットで安全とされていたが，
Kuperberg アルゴリズムに関する研究により，基準の安全性に達していないと指摘された [Peikert (EUROCRYPT 2020)]

SeaSignとCSI-FiShまとめ

SeaSign，CSI-FiShのいい点
- 高機能なプロトコルが作りやすい
- 実装が比較的しやすい

SeaSignの問題点
- CSI-FiShよりデータサイズが大きい

CSI-FiShの問題点
- 安全性の基準を満たしていない素数でしか実装ができない

→ SQI-Sign