6Y-04

複数人鍵共有プロトコルSIBDの実装

カク 海萍 † 國廣 昇 ‡
東京大学大学院 † 筑波大学 ‡

1 はじめに

Diffie-Hellman 方式は二者間鍵共有を行う基本プロトコルである．2011 年，Jao らは超特異楕円曲線間の同種を利用した耐量子鍵共有方式 (SIDH) を提案した [2]．2018 年，Furukawa らは SIDH 鍵共有方式と Burmester-Desmedt 鍵共有方式を組み合わせて，複数人鍵共有方式 (SIBD) を提案した [1]．SIBD 鍵共有方式は耐量子性を持ち，通信ラウンドの少ない n-party 2-round 鍵共有方式である．しかし，彼らは実装しておらず，鍵共有に要する計算時間は不明であった．我々は，SIBD プロトコルの実装を行い，効率的に鍵共有が可能であることを確認した．

2 準備

2.1 超特異楕円曲線

q を素数べきとし，$\mathbb{F}_q$ を位数 q の有限体とする．$a, b \in \mathbb{F}_q$ として，集合

$$E(\mathbb{F}_q) = \{(x, y) \in \mathbb{F}_q^2 | y^2 = x^3 + ax + b\} \cup \{\mathcal{O}\}$$

を考える．ここで，$\mathcal{O}$ は無限遠点である．適切に演算を定義することにより，$E(\mathbb{F}_q)$ は群になる．一般に，$E(\mathbb{F}_q)$ は楕円曲線と呼ばれる．p を素数とし，有限体 $\mathbb{F}_{p^m}$ 上で定義され，$p^m + 1 - \#E$ が p で整除される楕円曲線 E は超特異楕円曲線と呼ばれる．

2.2 同種

体 $\mathbb{K}$ 上の楕円曲線 E_1 と E_2 の間の同種 $\phi : E_1 \to E_2$ は，2 つの楕円曲線の間の代数群の射であり，全射かつ有限の核を持つものを言う．同種 ϕ の核 $\ker(\phi) := \{P \in E_1 | \phi(P) = \mathcal{O}_{E_2}\}$ は，楕円曲線 E_1 の点の部分集合である．E_1 とその部分群 E_1' に対して，E_1' が核となるの唯一の同種 $\phi : E_1 \to E_2$ が計算可能である．

2.3 SIDH 鍵共有プロトコル [2]

SIDH 方式の流れは DH 鍵共有と同一である．最初に，ネットワーク内の全員が共有できるパブリックパラメータを作る．

1. 素数 l_A, l_B と整数 e_A, e_B を適切に選ぶ.
2. $p = l_A^{e_A} l_B^{e_B} \cdot f \pm 1$ となる素数 p を計算する.
3. 有限体 $\mathbb{F}_{p^2}$ 上で超特異楕円曲線 E_0 を作る.
4. E_0 上で点 P_A, Q_A, P_B, Q_B を選ぶ.

鍵共有は以下の手順により行われる．

1. Alice と Bob は，それぞれランダムな自然数 m_A, n_A と m_B, n_B を選ぶ.
2. P_A, Q_A と P_B, Q_B を使って，共通の楕円曲線 E_0 から同種 $\phi_A : E_0 \to E_A$ と $\phi_B : E_0 \to E_B$ を計算し，楕円曲線 E_A, E_B を作る.
3. $\phi_A(P_B), \phi_A(Q_B) \in E_A, \phi_B(P_A), \phi_B(Q_A) \in E_B$ を計算する.
4. Alice は $E_A, \phi_A(P_B), \phi_A(Q_B)$ を Bob に送り，Bob から $E_B, \phi_B(P_A), \phi_B(Q_A)$ を受け取る.
5. Alice は $S_{AB} := [m_A]\phi_B(P_A) + [n_A]\phi_B(Q_A)$ を計算し，S_{AB} を核となる同種 $\phi_{AB} : E_B \to E_{AB}$ 及び E_B と同種な楕円曲線 E_{AB} を計算する．Bob も同様に E_A と同種な楕円曲線 E_{BA} を計算する.
6. Alice は $j(E_{AB})$，Bob は $j(E_{BA})$ を計算する．E_{BA} と E_{AB} は同じ j 不変量を持つため，$K = j(E_{AB}) = j(E_{BA})$ は共有鍵と

Implementation of SIBD Key Exchange Protocol
† Haiping Hao, The University of Tokyo
‡ Noboru Kunihiro, University of Tsukuba

なる．

3 SIBD 鍵共有プロトコル [1]

SIBD 鍵共有方式は，SIDH 鍵共有方式の拡張法であり，n-party 2-round 鍵共有方式である．

3.1 鍵共有 Round 1

まず，SIDH 鍵共有を用いて隣の二人と鍵共有する．これにより，例えば，i 番目の人は $j(E_{R_{i-1}R_i})$ と $j(E_{R_iR_{i+1}})$ を得る．ついで，i 番目の人は $X_i = j(E_{R_iR_{i+1}}) \cdot j(E_{R_{i-1}R_i})^{-1}$ を計算して，他の参加者全員に送る．

3.2 鍵共有 Round 2

i 番目の人は X_i を配布し，他の参加者から $X_1, X_2, \ldots, X_n$ を受け取り，

$$K_i = j(E_{R_{i-1}R_i})^n \cdot {X_i}^{n-1} \cdot {X_{i+1}}^{n-2} \ldots X_{i-2}$$

を計算する．$X_i = j(E_{R_iR_{i+1}}) \cdot j(E_{R_{i-1}R_i})^{-1}$ であるので，K_i の値は，

$$\begin{aligned} K_i &= j(E_{R_{i-1}R_i})^n \cdot (j(E_{R_iR_{i+1}}) \cdot j(E_{R_{i-1}R_i})^{-1})^{n-1} \\ &\quad \ldots (j(E_{R_{i-2}R_{i-1}}) \cdot j(E_{R_{i-3}R_{i-2}})^{-1}) \\ &= j(E_{R_1R_2}) \cdot j(E_{R_2R_3}) \ldots j(E_{R_nR_1}) \end{aligned}$$

で与えられる．全員が計算した j 不変量の積であるため，$K = K_1 = K_2 = \cdots = K_n$ が成り立つ．これにより，参加者全員は共有鍵を得る．

4 実装結果

我々は C 言語と Python 言語を用いて，SIBD 鍵共有プロトコルを実装した．

4.1 p を異なるサイズとした結果

p を異なるサイズとしたときの実装結果を図 1 に示す．ここで，$l_0 = 2, l_1 = 3, n = 6$ とし，整数 e_A, e_B は p のサイズに合わせて適切に設定する．

p が大きくなるほど，鍵共有プロトコルの安全性は高まるが，図 1 より，鍵共有の必要時間は依然短いことがわかる．6 人鍵共有の場合で，768-bit の p を使った鍵共有時間は平均約 0.66 秒で完了する．

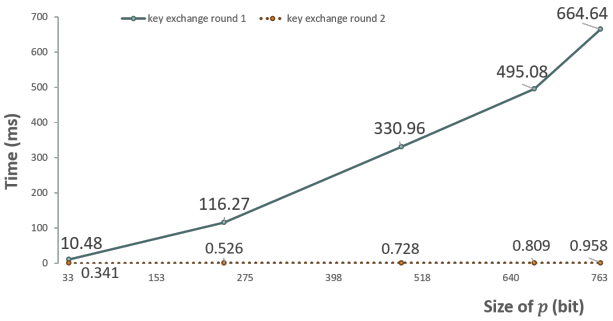

図 1 6 人鍵共有の計算時間

4.2 l_0, l_1 を異なる値とした結果

l_0, l_1 を異なる値とした時の実装結果を表 1 に示す．ここで，$n = 6$，512 bit の p とし，e_A, e_B は l_0, l_1 の値に合わせて適切に設定する．

l_0	l_1	Round 1(ms)	Round 2(ms)
2	3	330.96	0.728
3	5	506.20	0.715
5	7	691.89	0.726
11	13	758.90	0.700
17	19	911.17	0.718
23	29	1078.51	0.721

表 1 各ラウンドの計算時間

同じ安全性の下で，l_0, l_1 の値が大きくなると，鍵共有の時間が長くなる．つまり，鍵共有の時間を短くするためには，小さい l_0 と l_1 を使うべきである．

参考文献

[1] S. Furukawa, N. Kunihiro, K. Takashima: Multi-party Key Exchange Protocols from Supersingular Isogenies. In Proc. of ISITA2018, pp. 208–212, 2018.

[2] D. Jao, L. D. Feo: Towards Quantum Resistant Cryptosystems from Supersingular Elliptic Curve Isogenies. In PQCrypto2011. pp. 19–34, 2011.