修 士 論 文

モバイルエージェントセキュリティに関する研究

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

長谷川 亙

2007 年 3 月

修 士 論 文

モバイルエージェントセキュリティに関する研究

指導教官　双紙 正和 特任助教授

審査委員主査　双紙 正和 特任助教授
審査委員　金子 峰雄 教授
審査委員　宮地 充子 助教授

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

510080 長谷川 亙

提出年月: 2007 年 2 月

概 要

モバイルエージェントセキュリティの保護手法の１つとして，C. Cachin らによる secure function evaluation を用いた手法がある．さらに J. Algesheimer らが提案した手法では，エージェントと実行ホストの間に信頼できるホストを導入し，信頼できるホストと実行ホスト間で紛失通信と呼ばれる通信を用いる．従来の手法では，基本的な 1-out-of-2 紛失通信を用いるため，暗号回路入力の 1bit ごとに 1-out-of-2 紛失通信を繰り返し行わなければならず，効率が悪い．そこで，森らは k-out-of-n 紛失通信を適用し改良することにより，通信量と計算量の削減に成功した．しかし，そのプロトコルにおいては実行ホストが選択する秘密情報が特殊な条件下にあるとき，紛失通信に安全性の問題がある．そのため本研究では，特殊な条件下においても安全性を確保できるように紛失通信を改良し，効率の良い通信手法の提案を行った．

目 次

第1章　序論

1.1　研究背景

ネットワーク上のコンピュータを自律的に移動し，ユーザの代理として実行処理をするプログラムをモバイルエージェントという．モバイルエージェントの利点として，エージェントの自律的な移動によるコンピュータ間の通信遅延の低減，エージェントを複数生成することによる計算処理の並列化などが挙げられる．

しかしモバイルエージェントの実現にあたり，脅威となる攻撃が２つ存在する．１つは，悪意あるエージェントがウイルスなどによってホストに被害を及ぼす攻撃，もう１つはエージェントが悪意あるホスト上で実行処理をするときにおける秘密情報の盗聴やエージェントの改ざんなどの攻撃 [1] である．前者への対策は，ウイルス駆除ソフトや Java セキュリティなど確立したものが存在するが，後者への対策は，プログラムを単に暗号化するだけでは実行時に平文に復号しなければならず，暗号化の意味を成さないことから，実現は困難とされてきた．

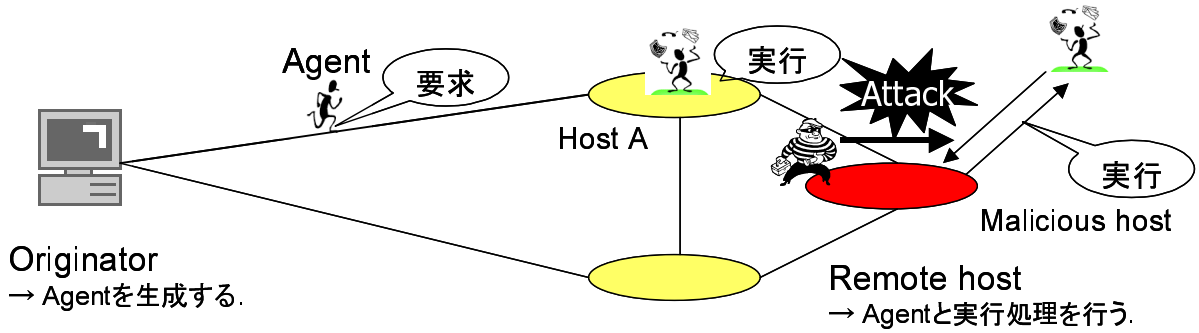

図 1.1: モバイルエージェント

その解決策として，暗号化されたプログラムを暗号文のまま実行処理できる手法が研究されており，C. Cachin らにより secure function evaluation を用いた手法 [2] が提案された．しかし，この手法では悪意ある実行ホストが暗号回路と暗号回路入力，暗号回路出力を不正に操作できるという問題が存在する．そこで，J. Algesheimer らは，エージェントと実行ホスト間に信頼できるホストを導入し，エージェントが保持する暗号回路入力を信頼できるホストの公開鍵で暗号化する手法 [3] を提案した．ここで，信頼できるホストから実行ホストへ暗号回路入力を送るために紛失通信と呼ばれる通信を用いる．しかし，このとき 1-out-of-2 紛失通信を単純に用いると，暗号回路入力の 1bit ごとに 1-out-of-2 紛失通信を繰り返し行わなければならず，効率が悪い．そのため，森らは，k-out-of-n 紛失通信をモバイルエージェントに適した形に改良し，1-out-of-2 紛失通信に代わり適用することで，通信量と計算量の削減に成功した [5]．しかし，実行ホストが選択す

る秘密情報の各 bit が全て 0 か 1 である場合など，特殊な条件下において，紛失通信に安全性の問題があることがわかった．

1.2 研究目的

本研究の目的は，エージェントと実行ホストの間で，計算効率が良く安全な通信手法の提案である．そこで，特殊な条件下でも安全性を確保するために，実行ホストが選択する秘密情報を 2 つの集合に分割せず一括して紛失通信を行った．これにより，エージェントと実行ホストの間で，効率が良く安全性が確保された通信手法を提案した．その結果として，[5] における 2 者間の通信を盗聴する攻撃者に対する安全性の問題を克服した．更に提案したプロトコルの効率性を，通信量と計算量の観点から既存方式と比較した．その結果，[2] より効率的であり，[5] と同程度の効率性を実現できることを示した．

1.3 本論文の構成

本論文は次のように構成される．

- 第 2 章：楕円曲線，双線形写像，安全性の根拠となる仮定，表記
- 第 3 章：secure function evaluation，1-out-of-2 紛失通信，k-out-of-n 紛失通信，プロトコルに求められる要件
- 第 4 章：既存研究と問題点
- 第 5 章：提案方式
- 第 6 章：安全性の考察
- 第 7 章：通信量と計算量についての評価
- 第 8 章：結論と今後の展望

第2章　準備

2.1　数学的準備

本章では，本稿に用いる暗号理論について述べる．2.1.1 節で楕円曲線，2.1.2 節では楕円曲線上の群の演算，2.1.3 節では双線形写像について述べる．

2.1.1　楕円曲線

本節では，楕円曲線の定義を与えた後，それらの判別式，j-不変量，楕円曲線の有理点について述べる．

Weierstrass 型標準形楕円曲線

Definition 1. Weierstrass 型標準形楕円曲線
$\mathbf{K}$ を体とするとき，体 $\mathbf{K}$ 上の楕円曲線とは，$(a_1, a_3, a_2, a_4, a_6 \in \mathbf{K})$ に対して，

$$\mathbf{E} : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{2.1}$$

で与えられる非特異な 3 次曲線と無限遠点 $\mathcal{O} = (\infty, \infty)$ で定義される．このような曲線を Weierstrass 型標準形という．無限遠点は，(2.1) において，$x \to \infty$ のとき $y \to \infty$ と考え，$\mathbf{E}$ の点と考える．

有限体 $\mathbb{F}_q$ 上の 3 次楕円曲線 $\mathbf{E}/\mathbb{F}_q$ $(q = p^r, r$: 素数 $, 0 < r \in \mathbb{Z})$ は，標数 p の値によって，Weierstrass 型標準形を次のように式変形できる．

Theorem 1. 標数 p による Weierstrass 型標準形の変形

1. $p = 2$ のとき，

$$y^2 + cy = x^3 + ax + b,\ (a, b, c \in \mathbb{F}_q) \tag{2.2}$$
$$y^2 + xy = x^3 + ax + b\ (a, b \in \mathbb{F}_q) \tag{2.3}$$

   のどちらかに式変形できる．

2. $p = 3$ のとき，

$$y^2 = x^3 + ax^2 + bx + c\ (a, b, c \in \mathbb{F}_q) \tag{2.4}$$

   に式変形できる．

3. $p \neq 2, 3$ のとき，

$$y^2 = x^3 + ax + b \ (a, b \in \mathbb{F}_q) \tag{2.5}$$

に式変形できる．

一般的に，標数 p が 5 以上のときの標準形として，(2.5) を用いることが多い．

判別式

標数 $p \neq 2, 3$ における楕円曲線の判別式 D は次の式で定義される．

Definition 2. 楕円曲線の判別式 $(p \neq 2, 3)$

$$D = 4a^3 + 27b^2 \tag{2.6}$$

j-不変量

Definition 3. j-不変量
楕円曲線 (2.5) に対して，

$$j = \frac{4 \cdot 1728a^3}{D} \tag{2.7}$$

を j-不変量という．

楕円曲線の有理点

Definition 4. 楕円曲線 $\mathbf{E}/\mathbb{F}$ の $\mathbb{F}-$ 有理点 $\mathbf{E}(\mathbb{F})$ は，次のように定義される．

$$\mathbf{E}(\mathbb{F}) := \{(x, y) \in \mathbb{F} \times \mathbb{F} | (x, y) \in \mathbf{E}\} \cup \{\mathcal{O}\} \tag{2.8}$$

このとき，$\mathbf{E}(\mathbb{F})$ の要素数を $\sharp\mathbf{E}(\mathbb{F})$ と表記する．

2.1.2 楕円曲線上の群

本節では，楕円曲線上の演算における法則と加算公式について述べる．

群の法則

任意の点 $P, Q \in \mathbf{E}(\mathbb{F})$ に対し，次のように $+$（加算）を定義する．

Definition 5. 楕円曲線上の群法則

1. $P + \mathcal{O} = \mathcal{O} + P = P$ ($\mathcal{O}$ は単位元)
2. $-\mathcal{O} = \mathcal{O}$

3. $P=(x_1,y_1)\neq\mathcal{O}$ とする．このとき，$-P=(x_1,-y_1-a_1x_1-a_3)$ となる．($\mathbf{E}(\mathbb{F})$ 上で x 座標が x_1 となる点は，$P,-P$ の 2 点のみで，$-P$ は一意的に存在する．)

4. $Q=-P$ のとき，$P+Q=Q+P=\mathcal{O}$ となる．(Q は P の逆元となる．)

5. $P,Q\neq\mathcal{O},Q\neq-P$ とする． $P\neq Q$ ならば，直線 PQ と楕円曲線 $\mathbf{E}/\mathbb{F}$ は第 3 の交点 R が存在する．
（ただし，直線 PQ が点 P における接線である場合は，$R=P$，直線 PQ が点 Q における接線である場合は，$R=Q$ となる．）
$P=Q$ のときは，点 P における楕円曲線の接線と曲線との交点を点 R とする．このとき，$P+Q=-R$ とする．

加算公式

$\mathbf{E}(\mathbb{F}_q)$ 上の加法は，次のような公式で表される．

Theorem 2. 加算公式

1. $P_1=(x_1,y_1)\in\mathbf{E}(\mathbb{F}_q)$ とすると，

$$-P_1=(x_1,-y_1-a_1x_1-a_3) \tag{2.9}$$

となる．

2. $P_1=(x_1,y_1),P_2=(x_2,y_2),P_3=(x_3,y_3)$ とすると，

$$P_3=P_1+P_2 \tag{2.10}$$

は，次のように求める．

(a) $P_1\neq P_2$ のとき，

$$\lambda=\frac{y_2-y_1}{x_2-x_1}\qquad\qquad \nu=\frac{y_1x_2-y_2x_1}{x_2-x_1} \tag{2.11}$$

(b) $P_1=P_2$ のとき，

$$\lambda=\frac{3x_1^2+2a_2x_1+a_4-a_1y_1}{2y_1+a_1x_1+a_3}\qquad \nu=\frac{-x_1^3+a_4x_1+2a_6-a_3y_1}{2y_1+a_1x_1+a_3} \tag{2.12}$$

とすると，

$$y=\lambda x+\nu \tag{2.13}$$

は，$P_1\neq P_2$，$P_1=P_2$ の各場合における P_1,P_2 を通る直線と楕円曲線 $\mathbf{E}(\mathbb{F}_q)$ の接線となる．このときの P_3 は，

$$x_3=\lambda^2+a_1\lambda-a_2-x_1-x_2\qquad\qquad y_3=-(\lambda+a_1)x_3-\nu-a_3 \tag{2.14}$$

で求められる．

2.1.3 双線形写像 (Bilinear map)

本節では，双線形写像の定義について述べる．

Definition 6. 双線形写像 (Bilinear map)
$\mathbb{G}_1$：素数位数 p の有限巡回加法群，$\mathbb{G}_2$：素数位数 p の有限巡回乗法群とする．このとき，次のような特性を持つ写像 $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ を，双線形写像という．

1. 双線形性:e は双方のパラメータにおいて線形性をもつ．すなわち，$e(P_1+P_2, Q) = e(P_1, Q)e(P_2, Q)$ 及び $e(P, Q_1 + Q_2) = e(P, Q_1)e(P, Q_2)$ となる．
2. 非退化：$e(P, P) \neq 1$ である．
3. 効率性：$e(P, Q)$ を効率的に計算可能なアルゴリズムが存在する．

2.2 安全性の根拠

本章では，安全性の根拠となる仮定を示す.「仮定」とは，以下に挙げる問題を無視できない確率で解く確率的多項式時間アルゴリズムが存在しないことである．以下では，$\mathbb{G}_1$ を素数位数 q の有限巡回加法群とし，$\mathbb{G}_2$ を素数位数 q の有限巡回乗法群とする．

Definition 7. 離散対数問題
$\mathbb{G}_2, g \in \mathbb{G}_2$，$a \in \mathbb{Z}_q^*$ とする．
入力：(g, g^a)
問題：a を計算する．

Definition 8. 楕円曲線上の離散対数問題 (ECDLP)
$\mathbb{G}_1, P \in \mathbb{G}_1$，$a \in \mathbb{Z}_q^*$ とする．
入力：(P, aP)
問題：a を計算する．

Definition 9. Decisional Diffie-Hellman(DDH) 問題
$\mathbb{G}_2, g \in \mathbb{G}_2$，$x, y, z \in \mathbb{Z}_q^*$ とする．
入力：(g, g^x, g^y, g^z)
問題：$g^{xy} \stackrel{?}{=} g^z$ を判定する．

Definition 10. 楕円曲線上の Decisional Diffie-Hellman(ECDDH) 問題
$\mathbb{G}_1, P \in \mathbb{G}_1$，$a, b, c \in \mathbb{Z}_q^*$ とする．
入力：(P, aP, bP, cP)
問題：$abP \stackrel{?}{=} cP$ を判定する．

Definition 11. Computational Diffie-Hellman(CDH) 問題
$\mathbb{G}_2, g \in \mathbb{G}_2$，$x, y \in \mathbb{Z}_q^*$ とする．
入力：(g, g^x, g^y)
問題：g^{xy} を計算する．

Definition 12. 楕円曲線上の Computational Diffie-Hellman(ECCDH) 問題
$\mathbb{G}_1, P \in \mathbb{G}_1$，$a, b \in \mathbb{Z}_q^*$ とする．
入力：(P, aP, bP)
問題：abP を計算する．

Definition 13. Bilinear Diffie-Hellman(BDH) 問題
$\mathbb{G}_1, \mathbb{G}_2, e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ ：双線形写像，$P \in \mathbb{G}_1, a, b, c \in \mathbb{Z}_q^*$ とする．
入力：(P, aP, bP, cP)
問題：$e(P, P)^{abc}$ を計算する．

Definition 14. Decisional Bilinear Diffie-Hellman(DBDH) 問題
$\mathbb{G}_1, \mathbb{G}_2, e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ ：双線形写像とする．次の 2 つの分布を判別する．

- $Y_1 = \{(P, aP, bP, cP, e(P, P)^{abc})\}$，ただし，$P \in \mathbb{G}_1, a, b, c \in_R \mathbb{Z}_q$
- $Y_2 = \{(P, aP, bP, cP, h)\}$，ただし，$P \in \mathbb{G}_1, a, b, c \in_R \mathbb{Z}_q, h \in_R \mathbb{G}_2$

Definition 15. Gap Diffie-Hellman(GDH) 問題
DDH 問題が解けるオラクルを与えられたときに，CDH 問題を計算する．

Definition 16. Chosen-Target Computational Diffie-Hellman (CT-CDH) 問題 [6]

$\mathbb{G}_3$:素数位数 q の Gap Diffie-Hellman 群，$P \in \mathbb{G}_3$，$s \in_R \mathbb{Z}_q^*, P_0 = sP, \mathcal{H}_1 : \{0,1\}^* \to \mathbb{G}_3$ とする．

入力：$(q, P, P_0, \mathcal{H}_1), T_{\mathbb{G}_3}(\cdot), H_{\mathbb{G}_3}(\cdot)$

$(T_{\mathbb{G}_3}(\cdot) : Q_i \in \mathbb{G}_3, H_{\mathbb{G}_3}(\cdot) : s \cdot (\cdot))$

問題：$((D_1, j_1), (D_2, j_2), \ldots, (D_k, j_k))$ の k 個のペアを計算する．

ただし，$\forall i \in \{1, 2, \ldots, k\}$ に対して，$D_i = sQ_{j_i}, (1 \leq j_i \leq q_T), q_H < k \leq q_T$ である．q_T はターゲットオラクル $T_{\mathbb{G}_3}(\cdot)$ に対するクエリ数，q_H はヘルパーオラクル $H_{\mathbb{G}_3}(\cdot)$ に対するクエリ数である．

Definition 17. New Target Computational Diffie-Hellman (NT-CDH) 問題

$\mathbb{G}_1$，$P \in \mathbb{G}_1$，$s_0, s_1 \in_R \mathbb{Z}_q^*, P_0 = s_0 P, P_1 = s_1 P, \mathcal{H}_1 : \{0,1\}^* \to \mathbb{G}_1$ とする．

入力：$(q, P, P_0, P_1, \mathcal{H}_1), (Q_1, \ldots, Q_n), (s_{b_1} Q_1, \ldots, s_{b_n} Q_n), T_{\mathbb{G}_1}(\cdot)$

$(T_{\mathbb{G}_1}(\cdot) : Q_i \in \mathbb{G}_1$を返すターゲットオラクル$, b_i \in \{0,1\})$

問題：$s_b Q_j \notin \{s_{b_1} Q_1, \ldots, s_{b_n} Q_n\}$ を計算する．　$(1 \leq j \leq n), (b \in \{0,1\})$

2.3 表記

本章では，本稿に用いる記号の定義について述べる．

- p：$2q + 1$
- q：素数
- $\mathbb{G}_1$：位数 q の有限巡回加法群
- $\mathbb{G}_2$：位数 q の有限巡回乗法群
- P：$\mathbb{G}_1$ の生成元
- g：$\mathbb{G}_2$ の生成元
- $\mathcal{H}_1$：$\{0,1\}^* \to \mathbb{G}_1$ となる一方向性衝突困難ハッシュ関数
- $\mathcal{H}_2$：$\mathbb{G}_1 \times \mathbb{Z}_q \to \{0,1\}^\ell$ となる一方向性衝突困難ハッシュ関数
- e：$\mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ となる双線形写像

第3章　モバイルエージェントセキュリティ

モバイルエージェントセキュリティとは，主に以下の 3 つの技術から構成される．

- 秘匿回路計算
- 紛失通信
- Trusted Third Party

本章では，3.1 節で秘匿回路計算，3.2 節で本研究で秘匿回路計算として用いる secure function evaluation，3.3 節で紛失通信を説明し，3.4 節でプロトコルに求められる条件について述べる．Trusted Third Party については，第 4 章の 4.2 節で説明する．

3.1　秘匿回路計算

秘匿回路計算とは Alice と Bob の 2 者間で，お互いの秘密情報を一切漏洩させることなく，演算結果を得られるような安全な関数を構築する手法である．主に，secure function evaluation[7], Mix and Match[8] や Conditional gate[9] などが挙げられ，本研究では secure function evaluation を用いる．

秘匿回路計算の概略を Algorithm 1 に示す．

Algorithm 1. 秘匿回路計算

1. Alice は自身の秘密入力 x を暗号化し，暗号化された $g(x,\cdot)$ を Bob に送る．
2. Bob は，自身の秘密入力 y を暗号化し，暗号化された $g(x,y)$ を計算し Alice に送る．
3. Alice は受け取った計算結果を復号し，$g(x,y)$ を得る．

つまり，Alice が秘密入力 x を持ち，Bob が秘密入力 y を持つとすると，互いの秘密情報に関しては一切知ることなく，双方向通信によって $g(x,y)$ を計算し，計算結果を Alice のみが入手できる．主に以下のような手法が挙げられる．

- secure function evaluation　[7]
- Mix and Match [8]
- Conditional gate [9]

3.2 secure function evaluation

secure function evaluation とは，1986 年に Yao らが提案した秘匿回路計算の手法の 1 つである [7]．主に 2 つのアルゴリズムとプロトコルから成る．暗号化アルゴリズム construct で多項式計算回路と対応する入出力を暗号化し，双方向通信プロトコル transfer で秘密情報をやり取りした後，演算アルゴリズム evaluate で暗号化した多項式計算回路と入出力の演算を暗号化したまま行う．このとき，計算回路が行う計算過程や計算結果に関する情報を一切漏洩させない．多項式計算回路や入出力を暗号化する暗号化回路は，擬似乱数関数を用いており，擬似乱数関数の安全性は DDH 仮定に帰着する．本研究では，双方向通信プロトコル transfer として紛失通信を用いる．紛失通信については 3.3 で述べる．

secure function evaluation の概要を Algorithm 2 に示す．

Algorithm 2. secure function evaluation

1. Bob がアルゴリズム construct を用いて，暗号化された計算回路を構築する．
2. 対話式プロトコル transfer で Alice と Bob の 2 者間で通信を行う．
3. アルゴリズム evaluate を用いて暗号化された計算回路と互いの暗号化された秘密情報から計算結果を Alice のみが入手する．

本研究において，2. では紛失通信を利用する．

3.3 紛失通信

紛失通信（oblivious transfer）[10, 11] とは，一般的に 1-out-of-2 紛失通信 [12, 13] を表し，送信者 Alice が 2 つの秘密情報 m_0, m_1 を保持しており，それらのどちらか 1 つの情報 m_b $(b \in \{0, 1\})$ のみが受信者 Bob に伝わるという通信である．ただし，受信者 Bob がどちらの情報を受信したか送信者 Alice は知り得ず，また受信者 Bob は受信しなかった他方の情報については一切知り得ない．1-out-of-2 紛失通信における安全性は CDH 仮定に基づく．

3.3.1 1-out-of-2 紛失通信

一般的な 1-out-of-2 紛失通信のプロトコルを Protocol 1 に示す．

Protocol 1. 1-out-of-2 紛失通信

- 送信者 Alice のメッセージ：$m_0, m_1 \in \mathbb{G}_2$
- 受信者 Bob の秘密情報：$b \in \{0, 1\}$
- 公開情報：$p, q, g \in \mathbb{G}_2$

1. 送信者 Alice は $\delta \in_R \mathbb{G}_2$ を選び，受信者 Bob に送る．

2. 受信者 Bob はランダムな $a \in_R \mathbb{Z}_q^*$ を選び，$\beta_b = g^a$ と $\beta_{b\oplus 1} = \delta/\beta_b$ を計算して，β_0, β_1 を送信者 Alice に送る．

3. 送信者 Alice は $\beta_0\beta_1 = \delta$ を計算して，正しければ $r_0, r_1 \in_R \mathbb{Z}_q^*$ を選び，$(e_0, f_0) = (g^{r_0}, m_0\beta_0^{r_0}), (e_1, f_1) = (g^{r_1}, m_1\beta_1^{r_1})$ を計算して，$(e_0, f_0), (e_1, f_1)$ を受信者 Bob に送る．

4. 受信者 Bob は f_b/e_b^a を計算して，メッセージ m_b を入手する．

1-out-of-2 紛失通信の図を図 3.1 に示す．

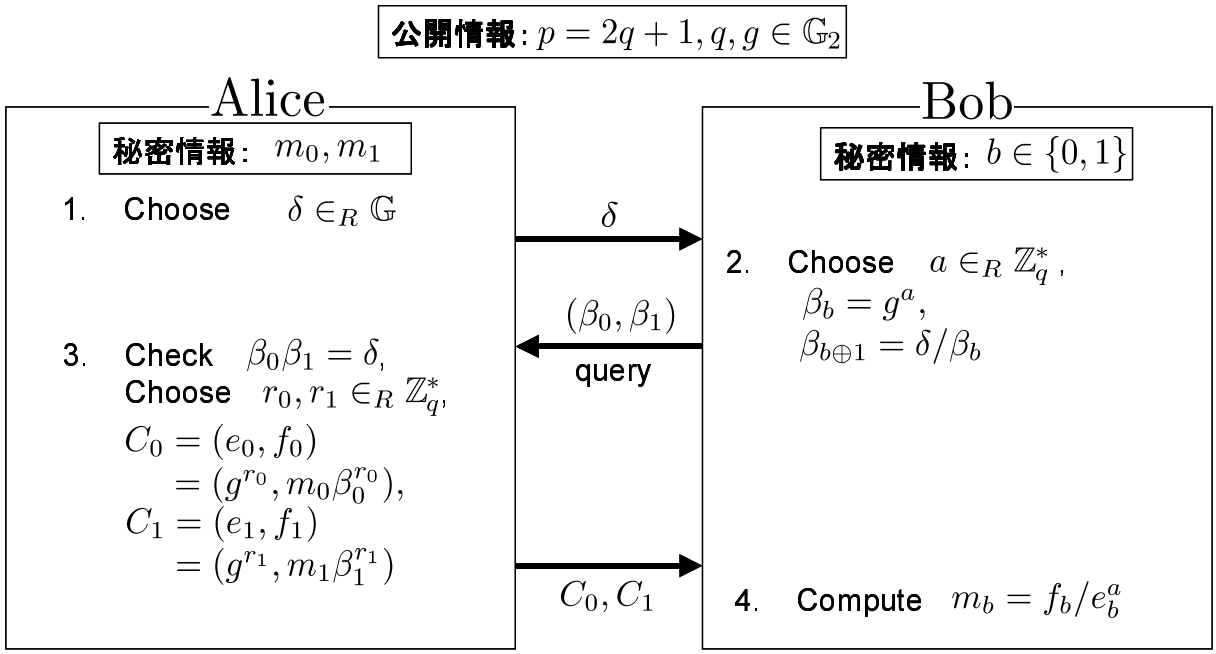

図 3.1: 1-out-of-2 紛失通信

3.3.2 k-out-of-n 紛失通信

1-out-of-2 紛失通信を単純に用いると暗号回路入力の 1bit ごとに 1-out-of-2 紛失通信を繰り返し行わなければならず，効率が悪い．そこで，より効率的な k-out-of-n 紛失通信 [4] が提案された．k-out-of-n 紛失通信とは，送信者 Alice が n 個の秘密情報 $m_1, m_2, \ldots, m_n$ を保持しており，それらのうち k 個の情報 $m_{i_1}, m_{i_2}, \ldots, m_{i_k}$ が受信者 Bob に伝わるという通信である．ただし，1-out-of-2 紛失通信と同様に受信者 Bob がどの情報を受信したか送信者 Alice は知り得ず，また受信者 Bob は受信しなかった他の情報については一切知り得ない．k-out-of-n 紛失通信における安全性は CT-CDH 仮定 [14] に基づく．

k-out-of-n 紛失通信のプロトコルを Protocol 2 示す．

Protocol 2. k-out-of-n 紛失通信

- 送信者 Alice のメッセージ： $m_1, m_2, \ldots, m_n$ ($|m_i| = \ell$ -bit)

- 受信者 Bob が選ぶインデックス：$i_1, i_2, \ldots, i_k$
- 公開情報：$p, q, P \in \mathbb{G}_1, \mathcal{H}_1 : \{0,1\}^* \to \mathbb{G}_1, \mathcal{H}_2 : \mathbb{G}_1 \times \mathbb{Z}_q \to \{0,1\}^\ell$

1. 受信者 Bob は $a_j \in_R \mathbb{Z}_q^*$ を選び，$Q_{i_j} = \mathcal{H}_1(i_j)$ と $A_j = Q_{i_j} + a_j P$ を計算する． $(j = 1, 2, \ldots, k)$
2. 受信者 Bob は，$A_1, A_2, \ldots, A_k$ を送信者 Alice に送る．
3. 送信者 Alice は $s \in_R \mathbb{Z}_q^*$ を選び，$P_0 = sP$，$D_j = sA_j$ と $c_j = m_j \oplus \mathcal{H}_2(sQ_i, i)$ を計算する．$(i = 1, 2, \ldots, n), (j = 1, 2, \ldots, k)$
4. 送信者 Alice は $P_0, D_1, D_2, \ldots, D_k, c_1, c_2, \ldots, c_n$ を受信者 Bob に送る．
5. 受信者 Bob は，$D'_j = D_j - a_j P_0$ を計算して，メッセージ $m_{i_j} = c_{i_j} \oplus \mathcal{H}_2(D'_j, i_j)$ を入手する． $(j = 1, 2, \ldots, k)$

k-out-of-n 紛失通信の図を図 3.2 に示す．

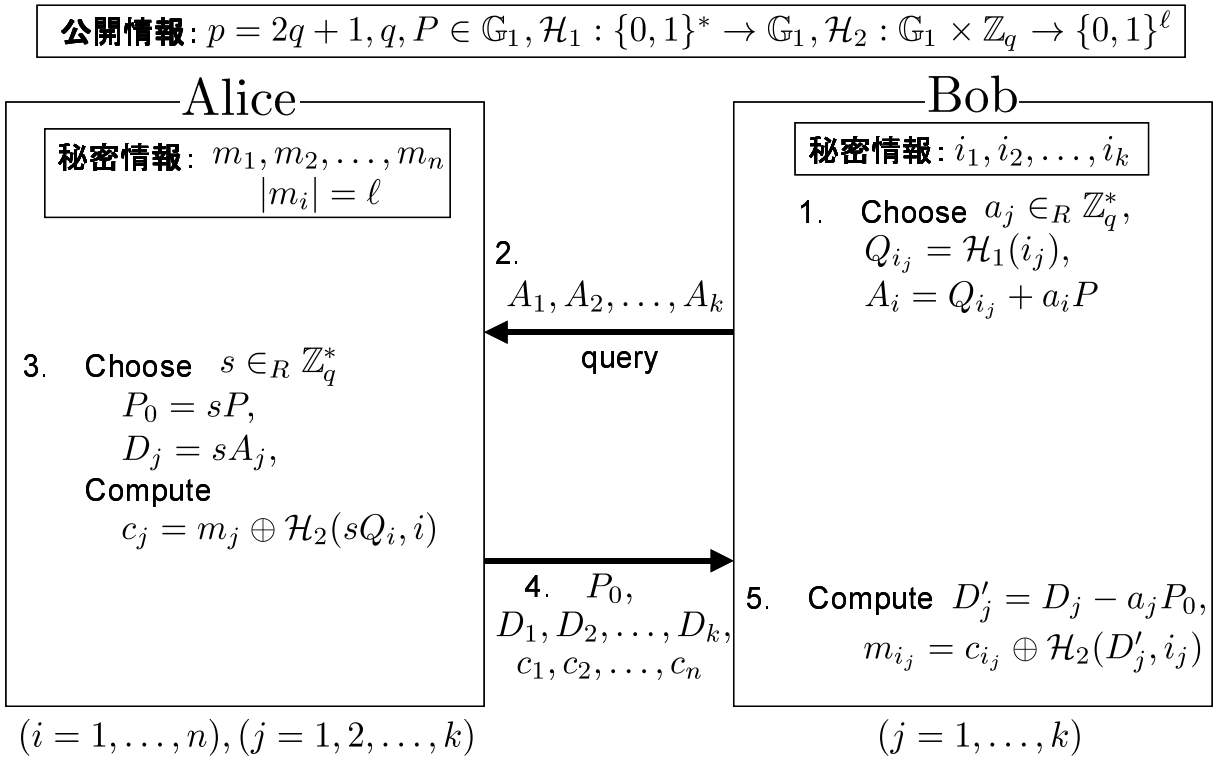

図 3.2: k-out-of-n 紛失通信

3.4 プロトコルに求められる条件

プロトコルに求められる条件を次のように定義する．ただし，送信者 S の k ペアの秘密情報を，$(m_{1,0}, m_{1,1}), (m_{2,0}, m_{2,1}), \ldots, (m_{k,0}, m_{k,1})$ とし，受信者 R の選択した情報を $m_{1,b_1}, m_{2,b_2}, \ldots, m_{k,b_k}$ $(b_i \in \{0,1\}, i = 1, 2, \ldots, k)$ とする．2 つの集合 C と C' は互いに素であるとする．

Definition 18. indistinguishability
もし，任意の確率的多項式チューリングマシン $\mathcal{D}$ に対して，次の式が成り立つとき，2 つの集合体 $\{X_i\}$ と $\{Y_i\}$ は（計算量的に）識別不能であるという．

$$|\Pr[\mathcal{D}(X_i) = 1] - \Pr[\mathcal{D}(Y_i) = 1]| \leq 1/p(i)$$

ただし，$p(i)$：多項式，i：十分大きい数とする．

Definition 19. Correcteness
プロトコルが正常に機能することを,「Correctness」という．送信者 S，受信者 R 共にプロトコルに則った動作を行ったときに，受信者 R は自身が選択した情報を正しく得られる．

Definition 20. The receiver's privacy - indistinguishability
任意の 2 つの集合 $C = \{b_1, b_2, \ldots, b_k\}$ と $C' = \{b'_1, b'_2, \ldots, b'_k\}$ に対して，C と C' に一致する transcript は送信者 S にとって識別不能であることを,「The receiver's privacy」という．$(b_i \in \{0, 1\}, i = 1, 2, \ldots, k)$
もし C と C' に対して，受け取った S のメッセージが完全に一様分布ならば，R の選択は「無条件に安全である」という．つまり，送信者 S に対して，受信者 R の選択した情報が知り得ない．

Definition 21. The sender's privacy
The sender's privacy は，受信者 R の動作により，以下の 2 つに場合分けする．

- 受信者 R が honest-but-curious behavior(semi-honest) である場合 - indistinguishability
  任意の選択の集合を $C = \{b_1, b_2, \ldots, b_k\}$ $(b_i \in \{0, 1\}, i = 1, 2, \ldots, k)$ とすると，選択しなかったメッセージはランダムなメッセージと識別不能であることを,「The sender's privacy」という．

- 受信者 R が malicious behavior である場合 - compared with the Ideal Model
  理想世界において，送信者 S は全ての秘密を，受信者 R は選択を Trusted Third Party (TTP) に送る．TTP は受信者 R に選択された秘密を送る．このとき，現実世界における任意の受信者 R に対して，受信者 R の出力と区別できないような出力を生成する確率的多項式時間チューリングマシン R^* が存在することを,「The sender's privacy」という．つまり，任意の受信者 R に対して，受信者 R は選択した秘密以外は得られない．

第4章　既存方式

本章では，4.1 節に表記を，4.2 節にセキュリティモデルを定義したのち，既存研究として，4.3 節で secure function evaluation を用いたプロトコルを，4.4 節で k-out-of-n 紛失通信を用いたプロトコルを示す．4.5 節では，既存方式の問題点について述べる．

4.1　表記

- O：モバイルエージェントを生成・派遣するホスト
- H：モバイルエージェントを実行するホスト
- T：信頼できるホスト (Trusted Third Party)
- E_T：T の公開鍵で暗号化
- D_T：T の秘密鍵で復号
- x：$x \in \mathcal{X}$ からなる集合から O がとるエージェントの初期状態（秘密入力）
- y：$y \in \mathcal{Y}$ からなる集合から H が選択する秘密入力
- f：H 上で多項式時間で計算される関数　$f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$
- z：$f(x, y)$ の出力　$(z \in \mathcal{Z})$
- $(x_1, \ldots, x_{n_x})$：x のバイナリ表記
- $(y_1, \ldots, y_{n_y})$：y のバイナリ表記
- $(z_1, \ldots, z_{n_z})$：z のバイナリ表記
- construct：暗号回路を生成するアルゴリズム
- transfer：2 者間での通信プロトコル
- evaluate：暗号回路入力を計算するアルゴリズム
- C：$f(\cdot, \cdot)$ を計算する多項式サイズ回路
- $\mathcal{C}$：多項式サイズ回路 C の暗号回路
- $\mathcal{L}$：x に相当する暗号回路入力 $(L_{i,0}, L_{i,1})$ $(1 \leq i \leq n_x)$

- $\mathcal{K}$：y に相当する暗号回路入力 $(K_{i,0}, K_{i,1})$ $(1 \leq i \leq n_y)$
- $\mathcal{U}$：z に相当する暗号回路出力 $(U_{i,0}, U_{i,1})$ $(1 \leq i \leq n_z)$

4.2 セキュリティモデル

本章では，モバイルエージェントセキュリティにおけるモデルについて述べる．

- 信頼できるホスト T(Trusted Third Party)[3] を，以下のように定義する．

  Definition 22. Trusted Third Party
  次のような性質をもつホストを「Trusted Third Party」という．

  - 耐タンパハードウェアを備えたホスト．
  - モバイルエージェントに関する秘密情報を入手しない．
  - 他のホストと結託しない．

- H については，以下の 2 つの動作モデルがある．

  - Honest-but-curious behavior (semi-honest)
    プロトコルに従うが，不正に情報を入手しようとする動作モデル．
  - Malicious behavior
    プロトコルに従わず，不正に情報を入手しようとする動作モデル．

- T と H は共に確率的多項式時間の計算能力を有するものとする．
- T と H 間の通信は，insecure channel 上で行われる．

図 4.1 には，モデルの概要を示す．

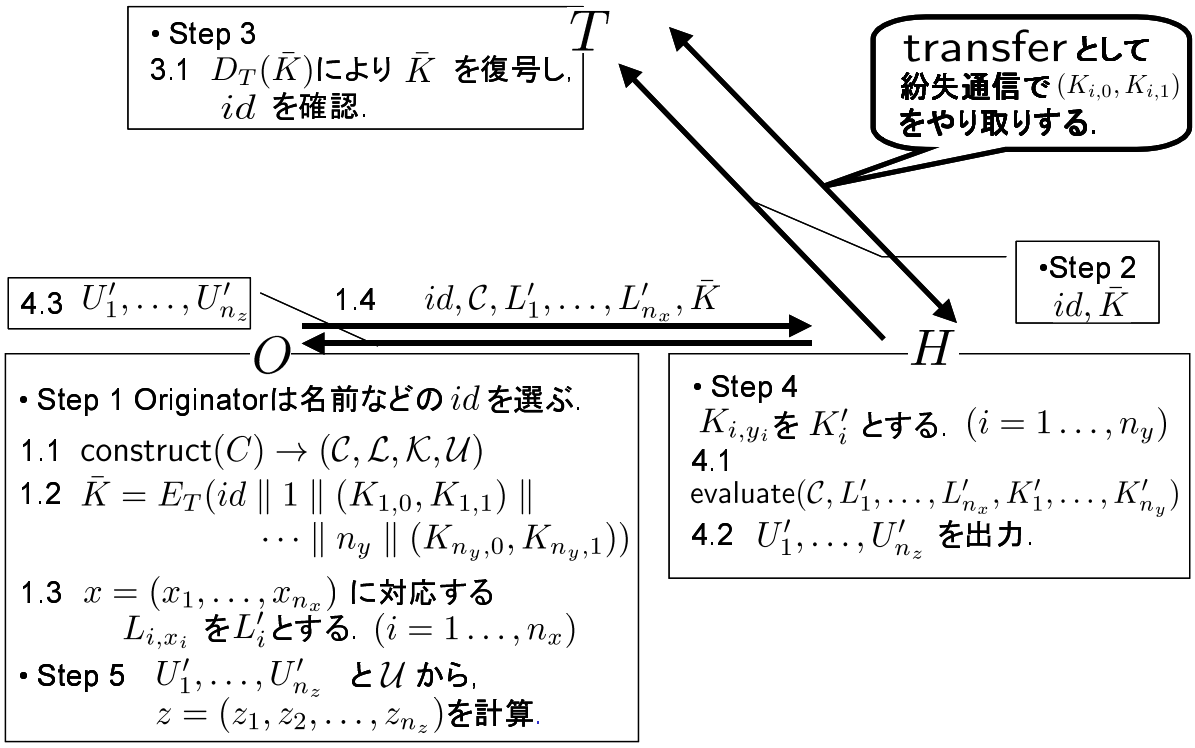

図 4.1: secure function evaluation を用いたモデル

4.3 1-out-of-2 紛失通信を用いたプロトコル

C. Cachin らは secure function evaluation を用いて，モバイルエージェントセキュリティを確保するプロトコルを提案した [2]．このプロトコルでは，モバイルエージェントが暗号回路と暗号回路入力，暗号回路出力を保持し，エージェントと実行ホスト間で 1-out-of-2 紛失通信を繰り返し用いて暗号回路入力をやり取りする．その後，実行ホストは暗号回路の計算を行う．しかし，この手法では悪意ある実行ホストが，暗号回路と暗号回路入力，暗号回路出力を不正に操作できるという問題が存在する．そこで J. Algesheimer らはエージェントと実行ホスト間に信頼できるホストを導入し，エージェントが保持する暗号回路入力を信頼できるホストの公開鍵で暗号化する手法を提案した [3]． この手法は，信頼できるホストと実行ホスト間で紛失通信を用いて暗号回路入力をやり取りする．このことにより，実行ホストは信頼できるホストに自身が選択した秘密情報に対応する暗号回路入力を復号してもらわなければならないため，実行ホストの不正な操作を防止し，エージェントが保持する秘密情報の安全性を高めた．

1-out-of-2 紛失通信を用いたプロトコルを Protocol 3 に示す．

Protocol 3. C. Cachin らが提案したプロトコル

- 公開情報：$p = 2q + 1, q, g \in \mathbb{G}_2$

Step 1. O は名前などを記入された id を選択し，
construct(C) を計算する．

(a) $\mathsf{construct}(C)$ より $(\mathcal{C}, \mathcal{L}, \mathcal{K}, \mathcal{U})$ を出力する．

(b) T の公開鍵で $\bar{K} = E_T(id \| 1 \| (K_{1,0}, K_{1,1}) \| 2 \| \ldots \| n_y \| (K_{n_y,0}, K_{n_y,1}))$ として暗号化する．

(c) 自身の秘密入力 $x = (x_1, x_2, \ldots, x_{n_x})$ に対応するように選んだ L_{i,x_i} を L'_i とする．$(i = 1, 2, \ldots, n_x)$

(d) $id, \mathcal{C}, L'_1, L'_2, \ldots, L'_{n_x}, \bar{K}$ を次のホスト H に送る．

Step 2. H は，id と $\bar{K}$ を T に送る．

Step 3. T は $D_T(\bar{K})$ により $\bar{K}$ を復号し，id を確認する．H は $\mathsf{transfer}$ に基づく通信を行う．

(a) T は，$\delta_i \in_R \mathbb{Z}_q^*$ を選択し，H に送る．

(b) H は，秘密情報 $y = (y_1, y_2, \ldots, y_{n_y})$ に対応するように a_i を選択し，$\beta_{i,y_i} = g^{a_i}, \beta_{i,y_i \oplus 1} = \delta_i / \beta_{i,y_i}$ を計算する． $(i = 1, 2, \ldots, n_y), (y_i \in \{0, 1\})$

(c) H は，$(\beta_{i,0}, \beta_{i,1})$ を T に送る． $(i = 1, 2, \ldots, n_y)$

(d) T は，$\beta_{i,0}\beta_{i,1} = \delta_i$ を確認し，正しければ $r_{i,0}, r_{i,1} \in_R \mathbb{Z}_q^*$ を選択し，$(e_{i,0}, f_{i,0}) = (g^{r_{i,0}}, K_{i,0}\beta_{i,0}^{r_{i,0}}), (e_{i,1}, f_{i,1}) = (g^{r_{i,1}}, K_{i,1}\beta_{i,1}^{r_{i,1}})$ を計算する． $(i = 1, 2, \ldots, n_y)$

(e) T は，$(e_{i,0}, f_{i,0}), (e_{i,1}, f_{i,1})$ を H に送る． $(i = 1, 2, \ldots, n_y)$

(f) H は，$D'_{i,y_i} = e_{i,y_i}^{a_i}$ を計算し，$K_{i,y_i} = f_{i,y_i} / D'_{i,y_i} = f_{i,y_i} / e_{i,y_i}^{a_i}$ を入手する． $(i = 1, 2, \ldots, n_y)$

Step 4. H は K_{i,y_i} を K'_i とし，以下の操作を行う． $(i = 1, 2, \ldots, n_y)$

(a) $\mathsf{evaluate}(\mathcal{C}, L'_1, L'_2, \ldots, L'_{n_x}, K'_1, K'_2, \ldots, K'_{n_y})$ を計算する．

(b) 計算結果の $U'_1, U'_2, \ldots, U'_{n_z}$ を出力し，O に送る．

Step 5. O は H から入手した $U'_1, U'_2, \ldots, U'_{n_z}$ と $\mathcal{U}$ から計算結果 $z = (z_1, z_2, \ldots, z_{n_z})$ を計算する．

Step 3. を抜粋した図を図 4.2 に示す．

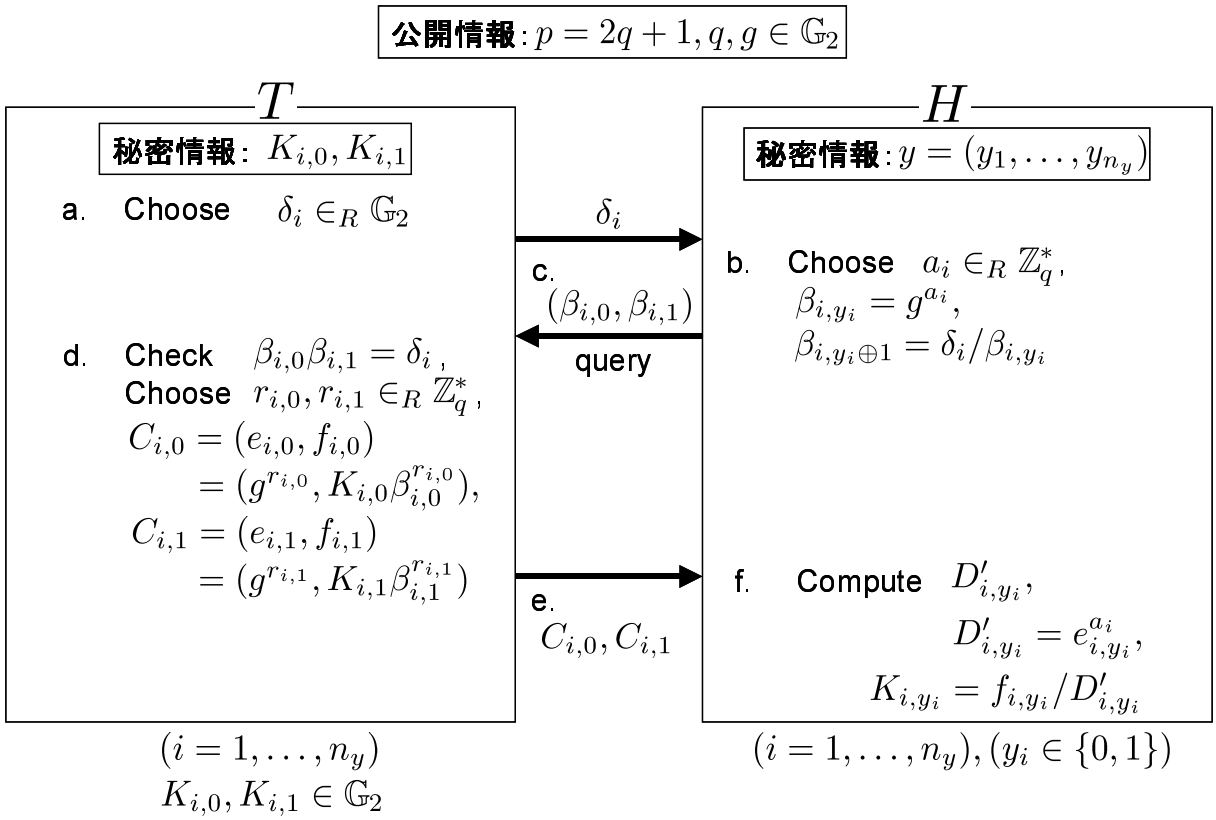

図 4.2: Cachin らのプロトコル (Step 3. のみ抜粋)

4.4 k-out-of-n 紛失通信を改良，適用したプロトコル

C. Cachin らが提案した Protocol 3 は，1-out-of-2 紛失通信を単純に用いるので，暗号回路入力の 1bit ごとに 1-out-of-2 紛失通信を繰り返し行わなければならず，効率が悪い．そこで，森らにより，Protocol 3 の Step 3. において信頼できるホストと実行ホストとの間で用いられる紛失通信を，モバイルエージェントに適した形に k-out-of-n 紛失通信 [4] を改良，適用することで，より効率的なプロトコルが提案された [5]．それにより，森らは通信量と計算量の削減に成功した．

k-out-of-n 紛失通信を改良し，適用したプロトコルを Protocol 4 に示す．

Protocol 4. 森らが提案したプロトコル

- 公開情報： $p = 2q + 1, q, P \in \mathbb{G}_1, \mathcal{H}_1 : \{0,1\}^* \to \mathbb{G}_1, \mathcal{H}_2 : \mathbb{G}_1 \times \mathbb{Z}_q \to \{0,1\}^\ell$

Step 1. O は名前などを記入された id を選択し，
construct(C) を計算する．

(a) construct(C) より $(\mathcal{C}, \mathcal{L}, \mathcal{K}, \mathcal{U})$ を出力する．

(b) T の公開鍵で $\bar{K} = E_T(id \| 1 \| (K_{1,0}, K_{1,1}) \| 2 \| \ldots \| n_y \| (K_{n_y,0}, K_{n_y,1}))$ として暗号化する．

(c) 自身の秘密入力 $x = (x_1, x_2, \ldots, x_{n_x})$ に対応するように選んだ L_{i,x_i} を L'_i とする．$(i = 1, 2, \ldots, n_x)$

(d) $id, \mathcal{C}, L'_1, L'_2, \ldots, L'_{n_x}, \bar{K}$ を次のホスト H に送る．

Step 2. H は，id と $\bar{K}$ を T に送る．

Step 3. T は $D_T(\bar{K})$ により $\bar{K}$ を復号し，id を確認する．H は transfer に基づく通信を行う．

(a) T は i を選択し，$\mathcal{H}_1(1) = Q_1, \ldots, \mathcal{H}_1(n_y) = Q_{n_y}$ を計算する． $(i = 1, 2, \ldots, n_y)$

(b) T は $s \in_R \mathbb{Z}_q^*$ を選択し，$sQ_1 = R_1, \ldots, sQ_{n_y} = R_{n_y}$ を計算し，$R_1, \ldots, R_{n_y}$ を H に送る．

(c) H は秘密情報 $y = (y_1, y_2, \ldots, y_{n_y})$ に対応するように，$I_0 = \{i | y_i = 0\}$, $I_1 = \{i | y_i = 1\}$ となるように分割する． $(i = 1, 2, \ldots, n_y)$

(d) H は，$a_i \in_R \mathbb{Z}_q^*$ を選択し，$i \in I_0$ について $A_{i,0} = R_i + a_i P$，$i \in I_1$ について $A_{i,1} = R_i + a_i P$ を計算する．$\{A_{i,0} | i \in I_0\}, \{A_{i,1} | i \in I_1\}$ を T に送る． $(i = 1, 2, \ldots, n_y)$

(e) T は H から $i \in I_0$ について $A_{i,0} = R_i + a_i P$，$i \in I_1$ について $A_{i,1} = R_i + a_i P$ を受け取り，n_y 個あることを確認する．$r_0 \neq r_1 \in_R \mathbb{Z}_q^*$ を選択し，$P_0 = r_0 P, P_1 = r_1 P, D_{i,0} = r_0 A_{i,0}, D_{i,1} = r_1 A_{i,1}$ を計算する．$\mathcal{H}_1(i) = Q_i$，$(i = 1, 2, \ldots, n_y)$ を取り，$sr_0 Q_i, sr_1 Q_i$ を計算し，$C_{i,0} = K_{i,0} \oplus \mathcal{H}_2(sr_0 Q_i, i), C_{i,1} = K_{i,1} \oplus \mathcal{H}_2(sr_1 Q_i, i)$ を計算する．$P_0, P_1, D_{i,0}, D_{i,1}, C_{i,0}, C_{i,1}$ を H に送る．

(f) H は $D'_{i,0} = D_{i,0} - a_i P_0, D'_{i,1} = D_{i,1} - a_i P_1$ を計算して，$K_{i,y_i} = C_{i,y_i} \oplus \mathcal{H}_2(D'_{i,y_i}, i)$ を入手する． $(i = 1, 2, \ldots, n_y)$ とする．

Step 4. H は K_{i,y_i} を K'_i とし，以下の操作を行う． $(i = 1, 2, \ldots, n_y)$

(a) evaluate$(\mathcal{C}, L'_1, L'_2, \ldots, L'_{n_x}, K'_1, K'_2, \ldots, K'_{n_y})$ を計算する．

(b) 計算結果の $U'_1, U'_2, \ldots, U'_{n_z}$ を出力し，O に送る．

Step 5. O は H から入手した $U'_1, U'_2, \ldots, U'_{n_z}$ と $\mathcal{U}$ から計算結果 $z = (z_1, z_2, \ldots, z_{n_z})$ を計算する．

Step 3. を抜粋した図を図 4.3 に示す.

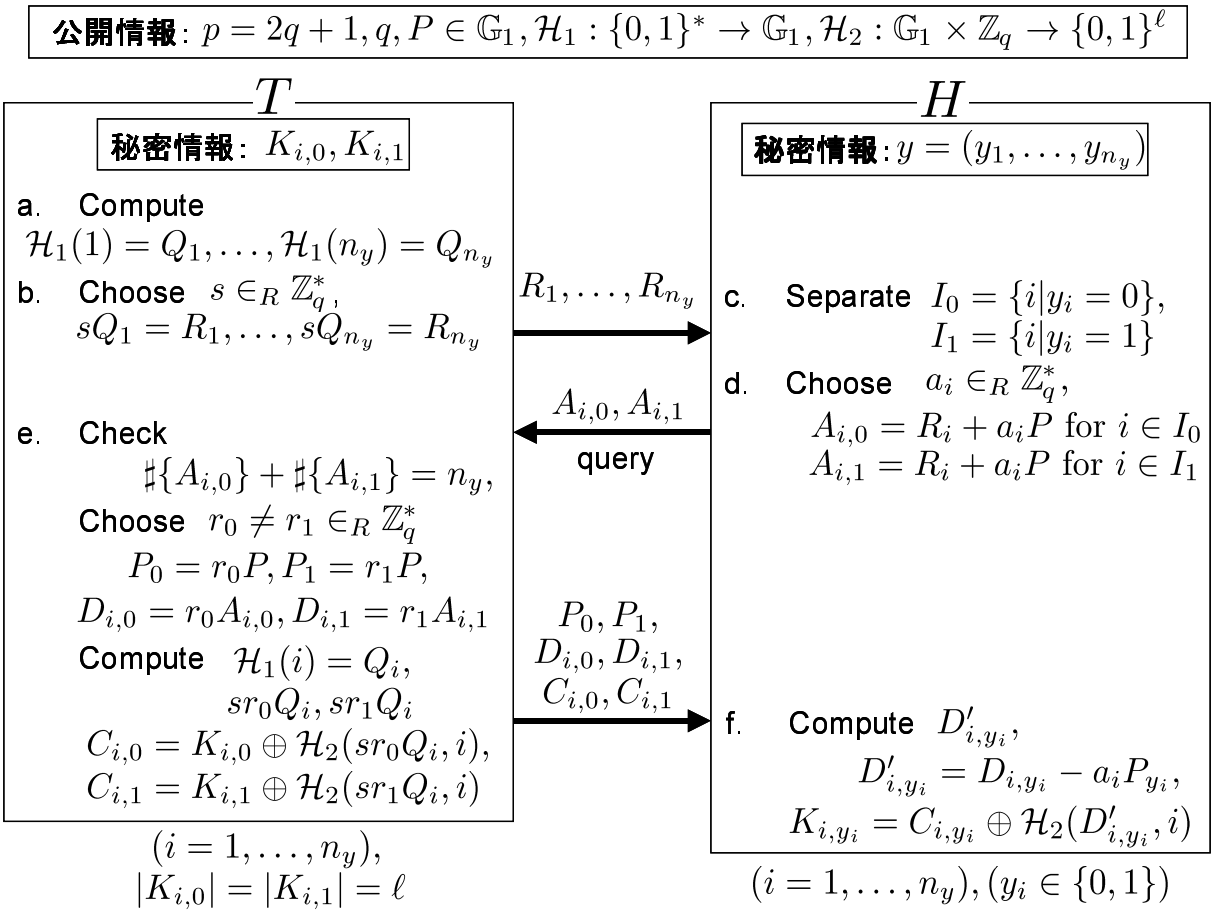

図 4.3: 森らのプロトコル (Step 3. のみ抜粋)

4.5 既存方式の問題点

森らが提案したプロトコル（Protocol 4）は，紛失通信（Step 3.）において実行ホスト H から信頼できるホスト T に，実行ホストが選択した秘密情報を送る際，秘密情報 y の各 bit を 0 と 1 の 2 つの集合に分割する．そのため，秘密情報の全ての bit が 0 や全てのビットが 1，またはどちらかの bit に偏りが生じている場合は，信頼できるホストはもちろん，ある攻撃者が紛失通信を盗聴することにより，最大 1/2 の確率で実行ホストの選択した秘密情報がわかってしまう．つまり，上記のような特殊な条件が成り立つときに，受信者である実行ホストの The receiver's privacy が確保されないという問題がある．

第5章　提案方式

本章では，4 章の 4.5 節で述べた既存方式の問題点を克服し，既存方式よりも効率が良く安全なプロトコルを 5.1 節で提案する．更に 5.2 節では，5.1 節における欠点を克服したプロトコルの提案を行う．

5.1　提案方式 1

紛失通信において，H の選択 y の各 bit を 0 と 1 の 2 つの集合に分割せずに通信を行うことで，効率的かつ The receiver's privacy を保護する手法を提案する．一般に H は複数存在するので，プロトコルを帰納的に繰り返す．

Protocol 5. 提案方式 1

- 公開情報：$p = 2q + 1, q, P \in \mathbb{G}_1, \mathcal{H}_1 : \{0,1\}^* \to \mathbb{G}_1, e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$

Step 1. O は名前などを記入された id を選択し，
$\mathsf{construct}(C)$ を計算する．

(a) $\mathsf{construct}(C)$ より $(\mathcal{C}, \mathcal{L}, \mathcal{K}, \mathcal{U})$ を出力する．

(b) T の公開鍵で $\bar{K} = E_T(id \| 1 \| (K_{1,0}, K_{1,1}) \| 2 \| \ldots \| n_y \| (K_{n_y,0}, K_{n_y,1}))$ として暗号化する．

(c) 自身の秘密入力 $x = (x_1, x_2, \ldots, x_{n_x})$ に対応するように選んだ L_{i,x_i} を L'_i とする．$(i = 1, 2, \ldots, n_x)$

(d) $id, \mathcal{C}, L'_1, L'_2, \ldots, L'_{n_x}, \bar{K}$ を次のホスト H に送る．

Step 2. H は，id と $\bar{K}$ を T に送る．

Step 3. T は $D_T(\bar{K})$ により $\bar{K}$ を復号し，id を確認する．H は transfer に基づく通信を行う．

(a) T は，$s_0, s_1 \in_R \mathbb{Z}_q^*$ を選択し，s_0P, s_1P を計算し，H に送る．

(b) H は，$a_i \in_R \mathbb{Z}_q^*$ を選択し，$A_i = \mathcal{H}_1(i) + s_{y_i}P + a_iP$ を計算する．$(i = 1, 2, \ldots, n_y), (y_i \in \{0,1\})$

(c) H は，A_i を T に送る．　$(i = 1, 2, \ldots, n_y)$

(d) T は，$D_{i,0} = s_0(A_i - s_0P), D_{i,1} = s_1(A_i - s_1P)$ を計算する．$r_{i,0}, r_{i,1} \in_R \mathbb{Z}_q^*$ を選択し，$C_{i,0} = (r_{i,0}P, K_{i,0} \times e(\mathcal{H}_1(i), s_0P)^{r_{i,0}}), C_{i,1} = (r_{i,1}P, K_{i,1} \times e(\mathcal{H}_1(i), s_1P)^{r_{i,1}})$ を計算する．　$(K_{i,0}, K_{i,1} \in \mathbb{G}_2, i = 1, 2, \ldots, n_y)$

(e) T は，$D_{i,0}, D_{i,1}, C_{i,0}, C_{i,1}$ を H に送る．　$(i = 1, 2, \ldots, n_y)$

(f) H は，$D'_{i,y_i} = D_{i,y_i} - a_i s_{y_i} P$ を計算し，$K_{i,y_i} = C_{i,y_i}[2]/e(D'_{i,y_i}, C_{i,y_i}[1]) = e(\mathcal{H}_1(i), s_{y_i}P)^{r_{i,y_i}}/e(D'_{i,y_i}, r_i$
を入手する． $(i = 1, 2, \ldots, n_y)$
($C_{i,y_i}[1]$：C_{i,y_i} の 1 番目の要素，$C_{i,y_i}[2]$：C_{i,y_i} の 2 番目の要素を表す．)

Step 4. H は K_{i,y_i} を K'_i とし，以下の操作を行う． $(i = 1, 2, \ldots, n_y)$

(a) evaluate$(\mathcal{C}, L'_1, L'_2, \ldots, L'_{n_x}, K'_1, K'_2, \ldots, K'_{n_y})$ を計算する．

(b) 計算結果の $U'_1, U'_2, \ldots, U'_{n_z}$ を出力し，O に送る．

Step 5. O は H から入手した $U'_1, U'_2, \ldots, U'_{n_z}$ と $\mathcal{U}$ から計算結果 $z = (z_1, z_2, \ldots, z_{n_z})$ を計算する．

提案方式 1(Protocol 5) を図 5.1 に示す.

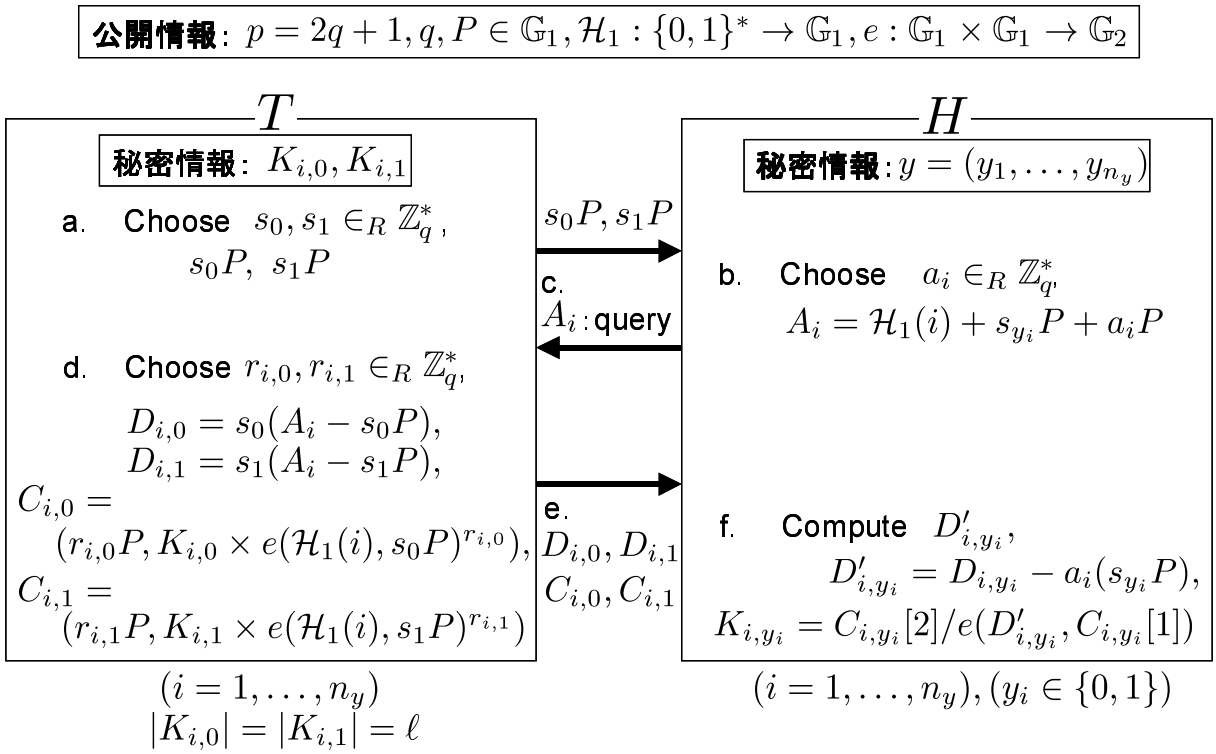

図 5.1: 提案方式 1 (Step 3. のみ抜粋)

5.2 提案方式 2

提案方式 1 では，Step3. の (c) において malicious behavior である H が特殊なクエリを送ることにより，T から余分な情報を得るという攻撃を考慮していない．そこで更に改良を加え，malicious behavior である H がプロトコルに従わない場合でも，安全に機能するように紛失通信を改良したプロトコルを提案する．

Protocol 6. 提案方式 2

- 公開情報：$p = 2q + 1, q, P \in \mathbb{G}_1, \mathcal{H}_1 : \{0,1\} \to \mathbb{G}_1, \mathcal{H}_2 : \mathbb{G}_1 \times \mathbb{Z}_q \to \{0,1\}^\ell$

Step 1. O は名前などを記入された id を選択し，
$\mathsf{construct}(C)$ を計算する．

(a) $\mathsf{construct}(C)$ より $(\mathcal{C}, \mathcal{L}, \mathcal{K}, \mathcal{U})$ を出力する．

(b) T の公開鍵で $\bar{K} = E_T(id \| 1 \| (K_{1,0}, K_{1,1}) \| 2 \| \ldots \| n_y \| (K_{n_y,0}, K_{n_y,1}))$ として暗号化する．

(c) 自身の秘密入力 $x = (x_1, x_2, \ldots, x_{n_x})$ に対応するように選んだ L_{i,x_i} を L'_i とする．$(i = 1, 2, \ldots, n_x)$

(d) $id, \mathcal{C}, L'_1, L'_2, \ldots, L'_{n_x}, \bar{K}$ を次のホスト H に送る．

Step 2. H は，id と $\bar{K}$ を T に送る．

Step 3. T は $D_T(\bar{K})$ により $\bar{K}$ を復号し，id を確認する．H は transfer に基づく通信を行う．

(a) T は，$s_0, s_1 \in_R \mathbb{Z}_q^*$ を選択し，s_0P, s_1P を計算し，H に送る．

(b) H は，$a_i \in_R \mathbb{Z}_q^*$ を選択し，$A_i = \mathcal{H}_1(i) + s_{y_i}P + a_iP$ を計算する．$(i = 1, 2, \ldots, n_y), (y_i \in \{0,1\})$

(c) H は，A_i を T に送る． $(i = 1, 2, \ldots, n_y)$

(d) T は，$D_{i,0} = s_0(A_i - s_0P), D_{i,1} = s_1(A_i - s_1P)$ を計算する．そして，$C_{i,0} = K_{i,0} \oplus \mathcal{H}_2(s_0\mathcal{H}_1(i), i)$, $C_{i,1} = K_{i,1} \oplus \mathcal{H}_2(s_1\mathcal{H}_1(i), i)$ を計算する． $(i = 1, 2, \ldots, n_y)$

(e) T は，$D_{i,0}, D_{i,1}, C_{i,0}, C_{i,1}$ を H に送る． $(i = 1, 2, \ldots, n_y)$

(f) H は，$D'_{i,y_i} = D_{i,y_i} - a_i(s_{y_i}P)$ を計算し，$K_{i,y_i} = C_{i,y_i} \oplus \mathcal{H}_2(D'_{i,y_i}, i)$ を入手する．$(i = 1, 2, \ldots, n_y)$

Step 4. H は K_{i,y_i} を K'_i とし，以下の操作を行う． $(i = 1, 2, \ldots, n_y)$

(a) $\mathsf{evaluate}(\mathcal{C}, L'_1, L'_2, \ldots, L'_{n_x}, K'_1, K'_2, \ldots, K'_{n_y})$ を計算する．

(b) 計算結果の $U'_1, U'_2, \ldots, U'_{n_z}$ を出力し，O に送る．

Step 5. O は H から入手した $U'_1, U'_2, \ldots, U'_{n_z}$ と $\mathcal{U}$ から計算結果 $z = (z_1, z_2, \ldots, z_{n_z})$ を計算する．

提案方式 2(Protocol 6) を図 5.2 に示す.

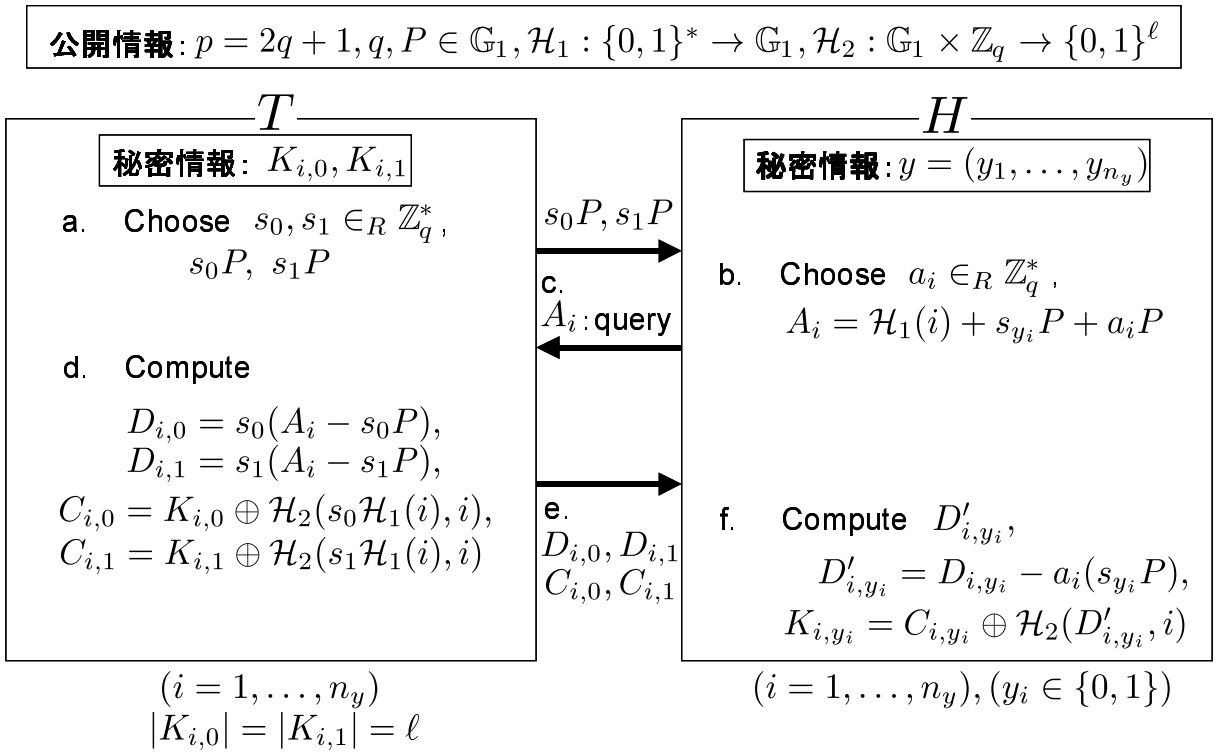

図 5.2: 提案方式 2 (Step 3. のみ抜粋)

第6章　考察

本章では，提案方式を 3.4 節に示す各項目について安全性を考察する．提案方式 1 の安全性を 6.1 節で，提案方式 2 の安全性を 6.2 節で評価する．

6.1　提案方式 1

提案方式 1 について，3.4 節の各項目について考察し，安全性を評価する．
以下では，$(i = 1, 2, \ldots, n_y), (y_i \in \{0, 1\})$ とする．

1. Correctness
   プロトコル通りに動作すると，H は選択した秘密情報 $y = (y_1, y_2, \ldots, y_{n_y})$ に対し，次の式で計算可能である．

$$\begin{aligned} C_{i,y_i}[2]/e(D'_{i,y_i}, C_{i,y_i}[1]) &= K_{i,y_i} \times e(\mathcal{H}_1(i), s_{y_i}P)^{r_{i,y_i}}/e(s_{y_i}\mathcal{H}_1(i), r_{i,y_i}P) \\ &= K_{i,y_i} \times e(\mathcal{H}_1(i), P)^{r_{i,y_i}s_{y_i}}/e(\mathcal{H}_1(i), P)^{r_{i,y_i}s_{y_i}} \\ &= K_{i,y_i}. \end{aligned} \tag{6.1}$$

   したがって，H は選択した $s_{y_i}P$ により正しく K_{i,y_i} が得られるので，Correctness を満たす．

2. The receiver's privacy

   Theorem 3. 提案方式 1 において，受信者 H の選択は無条件に安全である．よって，The receiver's privacy を満たす．

   Proof. 任意の $A_i = \mathcal{H}_1(i) + s_{y_i}P + a_iP$ と $s_{y_j}P$ $(j \neq i)$ に対して，$A_i = \mathcal{H}_1(i) + s_{y_j}P + a'_j$ を満たす a'_j が存在する．送信者 T に対して，A_i は任意のインデックスの値で隠すことが可能である．したがって，受信者 H の選択は，無条件に安全である．　□

3. The sender's privacy

   Theorem 4. 提案方式 1 において，もし受信者 H が honest-but-curious behavior (semi-honest) であれば，DBDH 仮定に基づき，選択しなかった秘密について知ることはない．つまり，The sender's privacy を満たす．

   Proof. 任意の $i \notin \{1, 2, \ldots, n_y\}$ に対して，DBDH 仮定のもとで C_{i,y_i} はランダムな値と見なせることを示す．次の 2 つの確率分布は，

   - $C_{i,y_i} = (P, s_{y_i}P, \mathcal{H}_1(i), r_{i,y_i}P, K_{i,y_i} \times e(\mathcal{H}_1(i), s_{y_i}P)^{r_{i,y_i}})$　ただし，$r_{i,y_i} \in_R \mathbb{Z}_q^*$,

- $X = (P, x_2, x_3, x_4, x_5)$ ただし，$x_2, x_3, x_4 \in_R \mathbb{G}_1, x_5 \in_R \mathbb{G}_2$

確率的多項式時間判別器 $\mathcal{D}$ によって判別可能である．このとき，DBDH 問題を解くためのサブルーチンとして，$\mathcal{D}$ を用いる他の確率的多項式時間チューリングマシン $\mathcal{D}'$ の構築を行う．

入力：(P, P_1, P_2, P_3, h) (DBDH における Y_1 か Y_2 のどちらか)

(a) $C_{i,y_i} = (U, V) = (P_3, K_{i,y_i} \times h)$; とする．
(b) $\mathcal{D}(P, P_1, P_2, U, V)$; を出力.

もし $\mathcal{D}'$ の入力が

- Y_1 であった場合,

$$(P, P_1, P_2, U, V) = (P, aP, bP, cP, K_{i,y_i} \times e(P, P)^{abc}) \tag{6.2}$$

(6.2) は C_{i,y_i} に対して，正しい形である．$(a = s_{y_i}, b = \mathcal{H}_1(i)/P, c = r_{i,y_i})$

- Y_2 であった場合,

$$(P, P_1, P_2, U, V) = (P, aP, bP, cP, K_{i,y_i} \times h) \tag{6.3}$$

(6.3) は $(\mathbb{G}_1 - \{0\}) \times \mathbb{G}_1 \times \mathbb{G}_1 \times \mathbb{G}_1 \times \mathbb{G}_2$ 上で一様分布である．

したがって，$\mathcal{D}$ の入力は正しい分布となる．もし，$\mathcal{D}$ が無視できない確率 ε で C_{i,y_i} と X が判別可能ならば，$\mathcal{D}'$ は無視できない確率 ε で DBDH 問題における Y_1 と Y_2 を判別できる．しかし，DBDH 仮定により無視できない確率 ε で DBDH 問題を解く確率的多項式チューリングマシン $\mathcal{D}$ は存在しないので，提案方式 1 は The sender's privacy を満たす. □

6.2 提案方式 2

提案方式 2 について，3.4 節の各項目にて考察し，安全性を評価する．
以下では，$(i = 1, 2, \ldots, n_y), (y_i \in \{0, 1\})$ とする．

1. Correctness
   プロトコル通りに動作すると，H は選択した秘密情報 $y = (y_1, y_2, \ldots, y_{n_y})$ に対し，次の式で計算可能である．

$$\begin{aligned}
C_{i,y_i} \oplus \mathcal{H}_2(D'_{i,y_i}, i) &= K_{i,y_i} \oplus \mathcal{H}_2(s_{y_i}\mathcal{H}_1(i), i) \oplus \mathcal{H}_2(D_{i,y_i} - a_i(s_{y_i}P), i) \\
&= K_{i,y_i} \oplus \mathcal{H}_2(s_{y_i}\mathcal{H}_1(i), i) \oplus \mathcal{H}_2(s_{y_i}(A_i - s_{y_i}P) - a_i s_{y_i}P, i) \\
&= K_{i,y_i} \oplus \mathcal{H}_2(s_{y_i}\mathcal{H}_1(i), i) \oplus \mathcal{H}_2(s_{y_i}(\mathcal{H}_1(i) + s_{y_i}P + a_iP - s_{y_i}P) - a_i s_{y_i}P, i) \\
&= K_{i,y_i} \oplus \mathcal{H}_2(s_{y_i}\mathcal{H}_1(i), i) \oplus \mathcal{H}_2(s_{y_i}(\mathcal{H}_1(i) + a_iP) - a_i s_{y_i}P, i) \\
&= K_{i,y_i} \oplus \mathcal{H}_2(s_{y_i}\mathcal{H}_1(i), i) \oplus \mathcal{H}_2(s_{y_i}\mathcal{H}_1(i) + s_{y_i}a_iP - a_i s_{y_i}P, i) \\
&= K_{i,y_i} \oplus \mathcal{H}_2(s_{y_i}\mathcal{H}_1(i), i) \oplus \mathcal{H}_2(s_{y_i}\mathcal{H}_1(i), i) \\
&= K_{i,y_i}.
\end{aligned} \tag{6.4}$$

したがって，H は選択した $s_{y_i}P$ により正しく K_{i,y_i} が得られるので，Correctness を満たす.

2. The receiver's privacy

Theorem 5. 提案方式 2 において，受信者 H の選択は無条件に安全である．よって，The receiver's privacy を満たす.

Proof. 任意の $A_i = \mathcal{H}_1(i) + s_{y_i}P + a_iP$ と $s_{y_j}P$ $(j \neq i)$ に対して，$A_i = \mathcal{H}_1(i) + s_{y_j}P + a'_jP$ を満たす a'_j が存在する．送信者 T に対して，A_i は任意のインデックスの値で隠すことが可能である．したがって，受信者 H の選択は，無条件に安全である． □

3. The sender's privacy

Theorem 6. 提案方式 2 において，受信者 H が悪意あるホストであっても，IBE(Id Based Encryption) システムとランダムオラクルモデルの安全性を仮定すると，The sender's privacy を満たす.

Proof. $\mathcal{H}_2$ をランダムオラクルとし，悪意あるホスト H は $\mathcal{H}_2(s_{y_i}Q_i, i)$ $(Q_i = \mathcal{H}_1(i))$ を得るために，クエリ $X_i = s_{y_i}Q_i$ を知らなければならない． $(i = 1, 2, \ldots, n_y)$
任意の悪意あるホスト H に対して，H と区別できない出力をするシミュレータ H^* を構築する.
H^* の動作は，次の通りである.

Step 1. $\mathcal{H}_1$ をシミュレートする.
$\mathcal{H}_1(i)$ の結果として，過去の問い合わせと異なる Q_i^* を出力する.（過去と同じ問い合わせの場合は過去と同じ値を出力する.）

Step 2. T をシミュレートする.

i. H から入力 A_i^* を得る． $(i = 1, 2, \ldots, n_y)$

ii. $D_{i,0}^* = s_0^*(A_i^* - s_0^*P), D_{i,1}^* = s_1^*(A_i^* - s_1^*P)$ を出力する． $(i = 1, 2, \ldots, n_y)$

Step 3. C^* をランダムに選択する.
$(C_{i,0}^*, C_{i,1}^*)$ を出力する． $(i = 1, 2, \ldots, n_y)$

Step 4. H をシミュレートする.

i. $(D_{i,0}^*, D_{i,1}^*), (C_{i,0}^*, C_{i,1}^*)$ を出力する． $(i = 1, 2, \ldots, n_y)$

ii. $\mathcal{H}_2$ をシミュレートする.

A. H から (x_i, i) を得る． $(i = 1, 2, \ldots, n_y)$

B. $x_i \stackrel{?}{=} s_0^*Q_i^*$ または $x_i \stackrel{?}{=} s_1^*Q_i^*$ を調べる． $(i = 1, 2, \ldots, n_y)$

- 成立する場合
  TTP から $K_{i,0}$ または $K_{i,1}$ を得て，H に $\mathcal{H}_2(x, i) = C_{i,0}^* \oplus K_{i,0}$ または $\mathcal{H}_2(x, i) = C_{i,1}^* \oplus K_{i,1}$ として送る． $(i = 1, 2, \ldots, n_y)$
- 不成立の場合
  過去の問い合わせと異なるランダムな値を H に返す.（過去と同じ問い合わせの場合は過去と同じ値を出力する.）

Step 5. $(s_0^*P, s_1^*P, A_i^*, D_{i,0}^*, D_{i,1}^*, C_{i,0}^*, C_{i,1}^*)$ を出力する． $(i = 1, 2, \ldots, n_y)$

もし，H が任意の秘密 $K_{i,0}, K_{i,1}$ $(i = 1, 2, \ldots, n_y)$ の n_y ペアに対する，$n_y + 1$ 個の復号鍵を計算できるならば，H^* は H によって選ばれた真の n_y 個のインデックスを知ることができない．このとき，シミュレーションは失敗する．したがって，H は NT-CDH 仮定の計算困難性に基づき，高々n_y 個の復号鍵しか計算できず，$K_{1,0}, K_{1,1}$ のどちらかしか入手できないことを示す．
もし，H が $\mathcal{H}_1$ にクエリを行った場合，ターゲットオラクルによりランダムな値を出力する．H^* が入力 A_i^* $(i = 1, 2, \ldots, n_y)$ を得て，T をシミュレートしたとき，クエリに一致する出力を H に返す．最終的に，もし H が任意の $1 \leq j \leq n_y + 1$ に対して，適切な (x_j, i_j) を $\mathcal{H}_2$ にクエリするならば，(x_j, i_j) の $n_y + 1$ 個の出力を行う．しかし，これは NT-CDH 仮定に矛盾する．したがって，H は高々n_y 個の復号鍵しか計算できない．

H の n_y 個の選択を $i_1, i_2, \ldots, i_{n_y}$ とする．クエリされた (x_j, i_j) $(j = 1, 2, \ldots, n_y)$ に対して，$C_{j,0}, C_{j,1}$ は返ってきたハッシュ値と一致する．$(s_{y_l}^* Q_l^*, l)$ $(l \neq i_1, i_2, \ldots, i_{n_y})$ は，ハッシュオラクル $\mathcal{H}_2$ にクエリすることができないので，$C_{l,0}, C_{l,1}$ は一様分布となる．したがって，H^* の出力の分布は H の出力の分布と区別できない． □

第7章　評価

7.1　通信量と計算量による比較

本節では，1-out-of-2 紛失通信を用いた Cachin らのプロトコル [2]（Protocol 3）と k-out-of-n 紛失通信を用いた森らのプロトコル [5]（Protocol 4），本研究の提案方式 1，2 のプロトコルを通信量と計算量に注目して比較する．表 7.1 は，T と H 間における通信量を示し，表 7.2 は T と H が行う楕円曲線上のべき倍算と双線形写像によるペアリング演算の計算量を示す．

通信量は $\mathbb{G}_1$ の生成元を基準に，1 つの生成元を 1 メッセージとし，n はメッセージ数を表す．また計算量は $\mathbb{G}_1$ の生成元を基準に，Protocol 3 を楕円曲線上の演算に置き換え，各手法の楕円曲線上のべき倍算における回数とペアリング演算の計算回数で比較する．ただし，各手法の位数 p は全て 160bit で同じ値とし，楕円曲線に用いる座標系は全て同じものとする．

表 7.1 より提案方式 1,2 は，Protocol 3 より通信量においては約 $2n$ メッセージ，計算量においては約 $n\mathbb{G}_1$ 削減に成功し，Protocol 4 と同程度の効率性が実現できた．

表 7.1: 通信量の比較

	通信量				
	通信回数	$T \to H$	$H \to T$	$T \to H$	総数
Protocol 3 [2]	3	n	$2n$	$4n$	$7n$
Protocol 4 [5]	3	n	n	$3n+2$	$5n+2$
提案方式 1	3	2	n	$4n$	$5n+2$
提案方式 2	3	2	n	$4n$	$5n+2$

表 7.2: 計算量の比較

	計算量		
	T	H	総数
Protocol 3 [2]	$5n\mathbb{G}_1$	$2n\mathbb{G}_1$	$7n\mathbb{G}_1$
Protocol 4 [5]	$(4n+2)\mathbb{G}_1$	$2n\mathbb{G}_1$	$(6n+2)\mathbb{G}_1$
提案方式 1	$(4n+2)\mathbb{G}_1 + 2ne$	$2n\mathbb{G}_1 + ne$	$(6n+2)\mathbb{G}_1 + 3ne$
提案方式 2	$(4n+2)\mathbb{G}_1$	$2n\mathbb{G}_1$	$(6n+2)\mathbb{G}_1$

$\mathbb{G}_1$:1 回の楕円べき倍算, e:1 回のペアリング演算

7.2 安全性の比較

本章では，各手法の安全性について比較する．その表を表 7.3 に示す．Cachin らのプロトコル [2]（Protocol 3）は，求められる全ての安全性を満たしている．しかし，1-out-of-2 紛失通信を利用するため，効率性に欠ける．そこで，モバイルエージェントに適するように k-out-of-n 紛失通信を改良し，1-out-of-2 紛失通信の代用として適用した森らのプロトコル [5]（Protocol 4）が提案されたが，問題点として The receiver's privacy を満たさないことがわかった．提案方式 1（Protocol 5）は，The receiver's privacy を満たし，かつ効率の良いプロトコルを提案したが，H が malicious behavior として動作するとき，The Sender's privacy を満たさないことがある．そのため，提案方式 2（Protocol 6）では，H が malicious behavior として動作しても The sender's privacy を満たすプロトコルを提案し，効率性の良さも確保した．

表 7.3: 安全性の比較

	安全性			
	Correctness	The receiver's privacy	The sender's privacy	仮定
Protocol 3 [2]	満たす	満たす	満たす	CDH 仮定
Protocol 4 [5]	満たす	満たさない	満たす †	CT-CDH 仮定
提案方式 1	満たす	満たす	満たす ‡	DBDH 仮定
提案方式 2	満たす	満たす	満たす	NT-CDH 仮定

† 具体的な証明はなし.

‡ H が honest-but-curious behavior (semi-honest) として動作した場合.

第8章　結論と今後の課題

8.1　結論

本研究では，提案方式 1,2 により H が選択する秘密情報を 2 つの集合に分割せず，一括して紛失通信を行うことで，森らのプロトコル [5] における T と通信を盗聴する攻撃者に対する安全性の問題を克服した．提案方式 1 では実行ホスト H が honest-but-curious-behavior(semi-honest) として動作するときに有効であるが，H が malicious behavior として動作するときは，クエリに特殊な操作を行い送信することで，T の保持する秘密情報から余分な情報を入手できてしまう可能性がある．そこで，提案方式 2 では H が malicious behavior として動作しても安全に実行できるように紛失通信の改良を行った．ただし，提案方式 1,2 共に T と H 間に必ず 3 回の通信が必要である．提案方式 1,2 を既存方式と通信量と計算量に注目し比較すると，Cachin らのプロトコル [2] より通信量，計算量共に削減でき，森らのプロトコル [5] と同程度の効率性を有することが示された．

8.2　今後の展望

今後は，一般的にモバイルエージェントは複数の実行ホストで実行処理をされるため，実行ホストが複数存在するときの効率的な計算手法の検討 [15] を行う．また，第三者機関の現実性を更に高めるため，第三者機関を複数のホストで構成し，閾値を用いて検証を行う紛失通信 [14] の適用やランダムオラクルを使用できるという仮定は強すぎるため，ランダムオラクルを使用しない k-out-of-n 紛失通信の適用 [16] などが考えられる．これらにより，オークションや情報検索など各々のアプリケーションに適するプロトコルの検討を行う．

謝辞

本研究の遂行に際し，熱心なご指導やご支援，ご助言をいただきました双紙 正和特任助教授に心から感謝すると共に厚くお礼申し上げます．また本研究に当たり，様々な面で貴重なご助言，ご指導していただきました宮地 充子助教授に深謝いたします．最後に公私共にご協力いただきました宮地研究室，双紙研究室の皆様に心から感謝いたします．

本研究成果は，文部科学省科学技術振興調整費の助成を受けております．

参考文献

[1] T. Sander and C. F. Tschudin: “Protecting mobile agents against malicious hosts”, Lecture Notes in Computer Science, **1419**, pp. 44–59 (1998).

[2] C. Cachin, J. Camenisch, J. Kilian and J. Muller: “One-round secure computation and secure autonomous mobile agents”, Automata, Languages and Programming, pp. 512–523 (2000).

[3] J. Algesheimer, C. Cachin, J. Camenisch and G. Karjoth: “Cryptographic security for mobile code”, Technical Report RZ 3302 (# 93348) (2000).

[4] C.-K. Chu and W.-G. Tzeng: “Efficient k-out-of-n oblivious transfer schemes with adaptive and non-adaptive queries.”, Public Key Cryptography, pp. 172–183 (2005).

[5] 森正行, 双紙正和, 宮地充子：“モバイルエージェントセキュリティに関する一考察”, 2005-CSEC-28, pp. 123–128 (2005).

[6] A. Boldyreva: “Threshold signature, multisignature and blind signatures based on the gap-diffie-hellman-group signature scheme”, Public Key Cryptography, pp. 31–46 (2003).

[7] A. C. Yao: “How to generate and exchange secrets”, in Proc. 27th FOCS, pp. 162–167 (1986).

[8] M. Jakobsson and A. Juels: “Mix and match: Secure function evaluation via ciphertexts”, Lecture Notes in Computer Science, **1976**, pp. 162+ (2000).

[9] B. Schoenmakers and P. Tuyls: “Practical two-party computation based on the conditional gate.”, ASIACRYPT, pp. 119–136 (2004).

[10] M. Naor and B. Pinkas: “Oblivious transfer and polynomial evaluation”, STOC ’99: Proceedings of the thirty-first annual ACM symposium on Theory of computing, New York, NY, USA, ACM Press, pp. 245–254 (1999).

[11] M. Naor and B. Pinkas: “Computationally secure oblivious transfer” (1999).

[12] S. Even, O. Goldreich and A. Lempel: “A randomized protocol for signing contracts”, Commun. ACM, **28**, 6, pp. 637–647 (1985).

[13] G. Brassard, C. Crépeau and J.-M. Robert: “Information theoretic reductions among disclosure problems”, FOCS, pp. 168–173 (1986).

[14] S. Zhong and Y. R. Yang: "Verifiable distributed oblivious transfer and mobile agent security", DIALM-POMC '03: Proceedings of the 2003 joint workshop on Foundations of mobile computing, New York, NY, USA, ACM Press, pp. 12–21 (2003).

[15] S. R. Tate and K. Xu: "Mobile agent security through multi-agent cryptographic protocols", Proceedings of the 4th International Conference on Internet Computing, pp. 462–468 (2003).

[16] W. OGATA and R. SASAHARA: "k out of n oblivious transfer without random oracles", IEICE Trans., E87-A, pp. 147–151 (2004).

本研究に関する発表論文

[1] W. Hasegawa, M. Soshi and A. Miyaji: “Efficient Communication on Mobile Agent Security”, The 2007 Symposium on Cryptography and Information Security (SCIS2007), 4F1-5 (Jan. 2007).