ペアリングによる内積計算

1 Weil ペアリング

有限体 $\mathbb{F}_p$ 上の楕円曲線 $E/\mathbb{F}_p$，その上の座標 $E(\mathbb{F}_p)$ について，q 等分部分群 $E[q]$ を次のように定義する．

$$E[q] := \{\forall P \in E(\mathbb{F}_p) : qP = \mathcal{O}\}. \tag{1}$$

そして，$E[p^e] \simeq \mathbb{Z}/p^e\mathbb{Z}$ のとき，この楕円曲線を**超特異 (supersingular)** と呼び，$E[p^e] \simeq \{\mathcal{O}\}$ のとき，**通常 (ordinary)** と呼ぶ．ただし，$\forall e \in \mathbb{N}^+$ である．また，$\gcd(\mathsf{ch}(\mathbb{F}_p)^{*1}, q) = 1$ のとき，$E[q] \simeq \mathbb{Z}/q\mathbb{Z} \times \mathbb{Z}/q\mathbb{Z}$ である．Weil ペアリングでは，$\gcd(\mathsf{ch}(\mathbb{F}_p), q) = 1$ を満たす素数 q $(\in \mathbb{P})$ における部分群 $E[q] \simeq \mathbb{Z}/q\mathbb{Z} \times \mathbb{Z}/q\mathbb{Z}$ を採用する．

このとき，ペアリング写像 $e_q : G_1 \times G_2 \to G_T$ が存在する．つまり，$P \in G_1, Q \in G_2$ のとき，$e_q(P, Q) \in G_T$ である．ただし，$|G_1| = |G_2| = |G_T| = q$ である．

(1) 双線型性 (bilinear)

$\forall a, \forall b \in \mathbb{Z}_q$ のとき，

$$\begin{cases} e_q(aP, Q) = e_q(P, aQ) = \{e_q(P, Q)\}^a \pmod p, \\ e_q(P, bQ) = e_q(bP, Q) = \{e_q(P, Q)\}^b \pmod p, \\ e_q(aP, bQ) = \{e_q(P, Q)\}^{ab} \pmod p. \end{cases} \tag{2}$$

(2) 同一性 (identity)

$$e_q(P, P) = 1 \pmod p. \tag{3}$$

(3) 非縮退 (non degenerate)

$\forall Q \in G_2$ $(\neq G_1)$ のとき，$e_q(P, Q) = 1$ ならば，$P = \mathcal{O}$ である．よって，

$$e_q(P, \mathcal{O}) = e_q(\mathcal{O}, Q) = 1 \pmod p. \tag{4}$$

(4) 準同型性 (homomorphic)

$\forall P, P' \in G_1, \forall Q, Q' \in G_2$ のとき，

$$\begin{aligned} e_q(P + P', Q + Q') &= e_q(P, Q + Q') \cdot e_q(P', Q + Q') \pmod p \\ &= e_q(P + P', Q) \cdot e_q(P + P', Q') \pmod p \\ &= e_q(P, Q) \cdot e_q(P, Q') \cdot e_q(P', Q) \cdot e_q(P', Q') \pmod p. \end{aligned} \tag{5}$$

(5) 交代性 (alternating)

$$e_q(P, Q) = \{e_q(Q, P)\}^{-1} \pmod p. \tag{6}$$

---

*1 有限体 $\mathbb{F}_p$ の標数は，$\underbrace{1 + 1 + \cdots + 1}_{p} = 0$ なので $\mathsf{ch}(\mathbb{F}_p) = p$ である．

2　内積の秘匿計算プロトコル

2 つのベクトル：

$$\begin{cases} \boldsymbol{x} := [x_1, x_2, \cdots, x_n] \in \mathbb{Z}_q^n \cdots \mathsf{User} \\ \boldsymbol{y} := [y_1, y_2, \cdots, y_n] \in \mathbb{Z}_q^n \ \cdots \mathsf{Owner} \end{cases} \tag{7}$$

の内積 $\boldsymbol{x} \cdot \boldsymbol{y} = \sum x_j y_j$ を Weil ペアリングを用いて秘匿計算する．ただし，User は Owner に対して，または Owner は User に対して，自分のベクトルを公開しない．このような秘匿計算を次のような手順に従って行う．

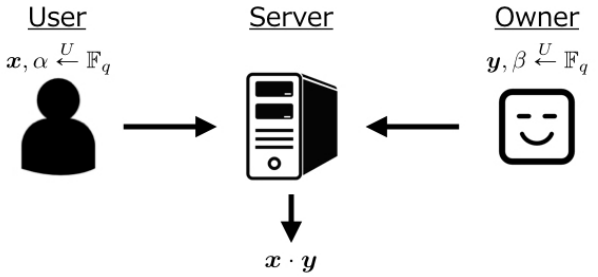

図 1　System Model

(1) User, Server, Owner の 3 者でペアリングに関する情報を共有する．具体的に，有限体 $\mathbb{F}_p$，楕円曲線 $E/\mathbb{F}_p$，q 等分部分群 $E[q] \simeq \mathbb{Z}/q\mathbb{Z} \times \mathbb{Z}/q\mathbb{Z}$ とその位数 q $(\in \mathbb{P})$，座標 $\exists P \in G_1, \exists Q \in G_2$，そしてペアリング写像 $e_q : G_1 \times G_2 \to G_T$ を 3 者で共有する．

(2) User は，乱数 $\alpha \xleftarrow{U} \mathbb{F}_q$ を生成する．ただし，User は α を公開しない．

(3) Owner は，乱数 $\beta \xleftarrow{U} \mathbb{F}_q$ を生成する．ただし，Owner は β を公開しない．

(4) User は，P に x_j と α をスカラー倍した結果：$\alpha P, \alpha x_1 P, \alpha x_2 P, \cdots, \alpha x_n P$ を Server に送信する．

(5) Owner は，Q に y_j と β をスカラー倍した結果：$\beta Q, \beta y_1 Q, \beta y_2 Q, \cdots, \beta y_n Q$ を Server に送信する．

(6) Server は，ペアリングの値を計算する．$g := e_q(P, Q)$，$\hat{g} := e_q(\alpha P, \beta Q) = g^{\alpha\beta}$ とすると，

$$\begin{cases} e_q(\alpha x_1 P, \beta y_1 Q) = \hat{g}^{x_1 y_1} \pmod p, \\ e_q(\alpha x_2 P, \beta y_2 Q) = \hat{g}^{x_2 y_2} \pmod p, \\ \qquad \vdots \\ e_q(\alpha x_n P, \beta y_n Q) = \hat{g}^{x_n y_n} \pmod p. \end{cases} \tag{8}$$

そして，$\hat{g}^{\boldsymbol{x}\cdot\boldsymbol{y}} = \hat{g}^{x_1 y_1} \times \hat{g}^{x_2 y_2} \times \cdots \times \hat{g}^{x_n y_n}$ と $\hat{g}$ の**離散対数問題 (DLP)**[*2]を解いて，$\boldsymbol{x} \cdot \boldsymbol{y}$ を得る．

[*2] 離散対数問題は p のサイズに依存するが，2012 年に 923 ビットの離散対数問題を計算機で解くことができることを NICT が発表した．ペアリングで用いられる p のビット数は，NICT によると 224 ビットが推奨なので内積を得ることは実現可能である．

3　安全性について

図 1 のような系の場合，User と Owner は Server に情報を送信するだけなので，相手のベクトルに関する部分情報を一切得ることはできない．しかしながら，Owner は User と Server との通信を，または User は Owner と Server との通信を盗聴する可能性がある．したがって，Semi-honest-Server，Malicious-Server，Malicious-Owner，Malicious-User について議論する．

(1) Semi-honest-Server

プロトコルに従うが，その途中で得られた情報から積極的に x_j, y_j を推測しようとする Server を Semi-honest であると定義する．つまり，安全性の要件として，プロトコル実行中にベクトルの成分の特定に繋がる部分情報を漏らさないことを要求する．

(4), (5) での安全性の仮定は，**楕円離散対数問題 (ECDLP)** である．つまり，$\alpha x_j P$, P や $\beta y_j Q$, Q から $\alpha x_j, \beta y_j$ を，延いては α, β, x_j, y_j を求めることはできない．したがって，Server は $\alpha x_j, \beta y_j$ を求めることはできない．しかしながら，*(6)* で離散対数問題を解くことが可能であると仮定しているので，Server はそれらの積 $\alpha x_j \cdot \beta y_j$ を求めることができる．ただし，乱数 α, β は公開されていないので，その積から x_j, y_j を特定することはできない．

(2) Malicious-Server

プロトコルを逸脱し，勝手な振る舞いによって得られた情報から x_j, y_j を推測しようとする Server を Malicious であると定義する．つまり，安全性の要件として，プロトコルなどで得られた情報からベクトルの成分の特定に繋がらないような困難さを要求する．

まず Server は，楕円離散対数問題をペアリングによって離散対数問題に帰着させることによって，α, β を特定しようとする．

$$\begin{cases} e_q(\alpha P, Q) = g^{\alpha} \pmod p, \\ e_q(P, \beta Q) = g^{\beta} \pmod p. \end{cases} \tag{9}$$

仮定より $(g^{\alpha}, g, p), (g^{\beta}, g, p)$ の離散対数問題を解くことが可能なので，乱数 α, β を求めることができる．そして，次のようにペアリングの値を計算する．

$$\begin{cases} e_q(\alpha x_j P, Q) = g^{\alpha x_j} \pmod p, \\ e_q(P, \beta y_j Q) = g^{\beta y_j} \pmod p. \end{cases} \tag{10}$$

同様に，$g^{\alpha x_j}, g^{\beta y_j}$ から離散対数問題を解き，$\alpha x_j, \beta y_j$ を得る．最後にそれらの値と α, β から Server は x_j, y_j を求めることができる．

(3) Malicious-Owner/ Malicious-User

通信を盗聴することによって，x_j および y_j を推測しようとする Server または User を Malicious であると定義する．つまり，安全性の要件として，盗聴した情報からベクトルの成分の特定に繋がらないような困難さを要求する．前述のとおり，まず乱数を特定する必要がある．

$$\begin{cases} e_q(\alpha P, Q) = g^{\alpha} \pmod p, \\ e_q(P, \beta Q) = g^{\beta} \pmod p. \end{cases} \tag{11}$$

$(g^{\alpha}, g, p), (g^{\beta}, g, p)$ の離散対数問題を解くことで容易に乱数 α, β を求めることができる．そして，次のようにペアリングの値を計算する．

$$\begin{cases} e_q(\alpha x_j P, Q) = g^{\alpha x_j} \pmod{p}, \\ e_q(P, \beta y_j Q) = g^{\beta y_j} \pmod{p}. \end{cases} \tag{12}$$

同様に，$g^{\alpha x_j}, g^{\beta y_j}$ から離散対数問題を解き，$\alpha x_j, \beta y_j$ を得る．最後にそれらの値と α, β から x_j, y_j を求めることができてしまう．そこで，SSL などのセキュア通信によって安全性を確保する．

4 効率性について

通信量と計算量は，それぞれ次の表 1,2 のとおりである．

	User	Server	Owner
(4)	$\mathcal{O}(n)$	–	–
(5)	–	–	$\mathcal{O}(n)$

表 1　Communication cost of User, Server and Owner

	User	Server	Owner
(4)	$\mathcal{O}(n)$	–	–
(5)	–	–	$\mathcal{O}(n)$
(6)	–	$\mathcal{O}(n)$	–

表 2　Computation cost of User, Server and Owner

ペアリングと述語暗号

1　正準的な双対直交基底

n 次元のベクトル空間 V の正準基底 $\mathbb{A}$ と相対する n 次元のベクトル空間 V^* の正準基底 $\mathbb{A}^*$ を次のように定義する．

$$\mathbb{A} = \{\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n\}, \quad \mathbb{A}^* = \{\boldsymbol{a}_1^*, \boldsymbol{a}_2^*, \cdots, \boldsymbol{a}_n^*\}. \tag{1}$$

そして，$\forall a_i \in \mathbb{F}_q, \forall a_i^* \in \mathbb{F}_q$ のとき，

$$\begin{cases} \boldsymbol{a}_1 = [a_1, 0, 0, \cdots, 0], \\ \boldsymbol{a}_2 = [0, a_2, 0, \cdots, 0], \\ \quad \vdots \\ \boldsymbol{a}_n = [0, 0, 0, \cdots, a_n], \end{cases} \qquad \begin{cases} \boldsymbol{a}_1^* = [a_1^*, 0, 0, \cdots, 0], \\ \boldsymbol{a}_2^* = [0, a_2^*, 0, \cdots, 0], \\ \quad \vdots \\ \boldsymbol{a}_n^* = [0, 0, 0, \cdots, a_n^*]. \end{cases} \tag{2}$$

このとき，双対直交基底なので次のような関係を満たす．

$$\boldsymbol{a}_i \cdot \boldsymbol{a}_j^* = \delta_{ij} \pmod{q}. \tag{3}$$

成分 a_i と a_i^* は，次のように選択する．

$$a_i \xleftarrow{U} \mathbb{F}_q, \quad a_i^* \leftarrow a_i^{-1} \pmod{q} \tag{4}$$

2 つのベクトル $\boldsymbol{x}, \boldsymbol{y}$ を次のように定義する．

$$\begin{cases} \boldsymbol{x} := [x_1, x_2, \cdots, x_n] \in \mathbb{Z}_q^n \\ \boldsymbol{y} := [y_1, y_2, \cdots, y_n] \in \mathbb{Z}_q^n \end{cases} \tag{5}$$

内積 $\boldsymbol{x} \cdot \boldsymbol{y}$ を次のように秘匿計算するため，x_c, y_c を次のように定義する．

$$x_c := \sum_{j=1}^{n} x_j \boldsymbol{a}_j \left(= \sum_{j=1}^{n} x_j a_j \right), \quad y_c := \sum_{j=1}^{n} y_j \boldsymbol{a}_j^* \left(= \sum_{j=1}^{n} y_j a_j^* \right). \tag{6}$$

$g := e_q(P, Q)$ $(\neq 0)$ とするとき，$\mathbb{A}, \mathbb{A}^*$ は次のような関係を満たす．

$$e_q(\boldsymbol{a}_i P, \boldsymbol{a}_j^* Q) = g^{\sum \boldsymbol{a}_i \cdot \boldsymbol{a}_j^*} = g^{\delta_{ij}} \pmod{p}. \tag{7}$$

n 次元ベクトル空間 V, V^* におけるペアリング演算 e_q は，

$$e_q(x_c P, y_c Q) := \prod_{j=1}^{n} e_q(x_j \boldsymbol{a}_j P, y_j \boldsymbol{a}_j^* Q) = e_q(P, Q)^{\sum x_j y_j \boldsymbol{a}_j \cdot \boldsymbol{a}_j^*} = g^{\boldsymbol{x} \cdot \boldsymbol{y}} \pmod{p}. \tag{8}$$

2　双対直交基底の一般線形変換

ユニタリ行列 $X := (\chi_{ij}) \xleftarrow{U} GL(n, \mathbb{F}_p)$ を用いて，正準基底 $\mathbb{A}, \mathbb{A}^*$ を $\mathbb{B}, \mathbb{B}^*$ に変換する．ユニタリ行列なので，$U = (U^T)^{-1}$ を満たす．したがって，

$$\boldsymbol{b}_i = \sum_{j=1}^{n} \chi_{ij} \boldsymbol{a}_j, \quad \boldsymbol{b}_j^* = \sum_{i=1}^{n} (\chi_{ji})^{-1} \boldsymbol{a}_i^*. \tag{9}$$

そして，$X^TX = E$ なので $\boldsymbol{b}_i \cdot \boldsymbol{b}_j^* = \boldsymbol{a}_i \cdot \boldsymbol{a}_j^* = \delta_{ij}$ である．したがって，$\mathbb{B}, \mathbb{B}^*$ は次のような関係を満たす．

$$e_q(\boldsymbol{b}_i, \boldsymbol{b}_j^*) = g^{\delta_{ij}} \pmod p. \tag{10}$$

3　1 階層内積述語暗号

$n+3$ 次元のペアリングベクトル空間 $(p, V, V^*, G_T, \boldsymbol{B}, \boldsymbol{B}^*)$ を考える．

$$\begin{cases} \mathbb{B} = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n, \boldsymbol{b}_{n+1}, \boldsymbol{b}_{n+2}, \boldsymbol{b}_{n+3}\}, \\ \mathbb{B}^* = \{\boldsymbol{b}_1^*, \boldsymbol{b}_2^*, \cdots, \boldsymbol{b}_n^*, \boldsymbol{b}_{n+1}^*, \boldsymbol{b}_{n+2}^*, \boldsymbol{b}_{n+3}^*\}, \\ \hat{\mathbb{B}} = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n, \boldsymbol{d}_{n+1}, \boldsymbol{b}_{n+3}\}. \end{cases} \tag{11}$$

(1) Key Generation

$\boldsymbol{x} = [x_1, x_2, \cdots, x_n] \in \mathbb{F}_q$ を**属性ベクトル**，$\boldsymbol{v} = [v_1, v_2, \cdots, v_n] \in \mathbb{F}_q$ を**述語ベクトル**と呼ぶ．ただし，$\boldsymbol{x} \cdot \boldsymbol{v} = 0$ を満たすように設定する．そして，$\sigma, \eta \xleftarrow{U} \mathbb{F}_q$ とするとき，

$$k^* := \sigma(v_1\boldsymbol{b}_1^* + \cdots + v_n\boldsymbol{b}_n^*) + \eta\boldsymbol{b}_{n+1}^* + (1-\eta)\boldsymbol{b}_{n+2}^*. \tag{12}$$

このとき，$\mathbb{B}^*, k^*$ を秘密鍵 sk，$\hat{\mathbb{B}}$ を公開鍵 pk とする．

(2) Encryption

$\delta_1, \delta_2, \zeta \xleftarrow{U} \mathbb{F}_q$，$\boldsymbol{d}_{n+1} := \boldsymbol{b}_{n+1} + \boldsymbol{b}_{n+2}$ のとき，次のように平文 m $(\in \mathbb{Z}_p)$ を暗号化する．

$$\begin{cases} c_1 = \delta_1(x_1\boldsymbol{b}_1 + \cdots + x_n\boldsymbol{b}_n) + \zeta\boldsymbol{d}_{n+1} + \delta_2\boldsymbol{b}_{n+3}, \\ c_2 = g^\zeta \cdot m \pmod p. \end{cases} \tag{13}$$

(3) Decryption

$$e_q(c_1, k^*) = \prod_{i=1}^{n} e_q(\delta_1 x_i\boldsymbol{b}_i, \sigma v_i\boldsymbol{b}_i^*) \cdot e_q(\zeta\boldsymbol{b}_{n+1}, \eta\boldsymbol{b}_{n+1}^*) \cdot e_q(\zeta\boldsymbol{b}_{n+2}, (1-\eta)\boldsymbol{b}_{n+2}^*) \tag{14}$$

$$= g^{\delta_1\sigma\sum x_i v_i} \cdot g^{\zeta\eta} \cdot g^{\zeta(1-\eta)} = g^\zeta \pmod p. \tag{15}$$

よって，次のように復号する[*3]．

$$\frac{c_2}{e_q(c_1, k^*)} = m \pmod p. \tag{16}$$

[*3] 秘密鍵を持っていても属性ベクトル $\boldsymbol{x}$ が $\boldsymbol{x} \cdot \boldsymbol{v} = 0$ を満たさない場合，復号することができない．

類似検索可能暗号の実装評価

1　重み付き Euclid 距離の秘匿計算プロトコル

2 つのベクトル：

$$\begin{cases} \boldsymbol{x} := [x_1, x_2, \cdots, x_n] \in \mathbb{Z}_q^n \cdots \mathsf{User} \\ \boldsymbol{y} := [y_1, y_2, \cdots, y_n] \in \mathbb{Z}_q^n \ \cdots \mathsf{Owner} \end{cases} \tag{1}$$

の重み付き Euclid 距離 $l_{xy} = \sum w_j |x_j^2 - y_j^2|$ の 2 乗 l_{xy}^2 を Weil ペアリングを用いて秘匿計算する．ただし，User は Owner に対して，または Owner は User に対して，自分のベクトルを公開しない．このような秘匿計算を次のような手順に従って行う．

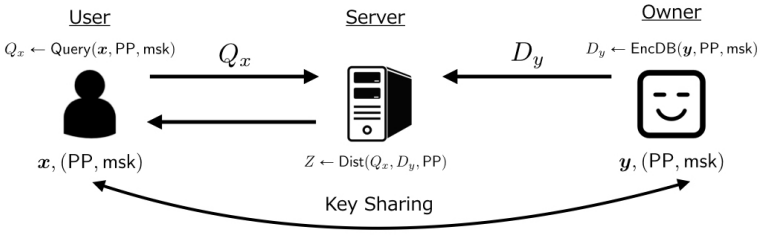

図 1　System Model

(1) Owner は，n 個の重み w_j $(\in \mathbb{Z}_q)$ を設定し公開する．

(2) User は，$\boldsymbol{x}$ を次のように符号化し $\boldsymbol{x}''$ を得る．ただし，$|\boldsymbol{x}''| = n + 4$ である．

$$\begin{cases} x_1' = \sum w_j x_j^2, \ \ x_2' \leftarrow 1, \\ x_j' = -2 w_{j-2} x_{j-2} \ \ (j = 3, 4, \cdots, n+2). \end{cases} \tag{2}$$

さらに，$(0, 1)$ を追加する．

$$\boldsymbol{x}'' = (0, 1, x_1', \cdots, x_{n+2}'). \tag{3}$$

(3) Owner は，$\boldsymbol{y}$ を次のように符号化して $\boldsymbol{y}''$ を得る．ただし，$|\boldsymbol{y}''| = n + 4$ である．

$$\begin{cases} y_1' = 1, \ \ y_2' \leftarrow \sum w_j y_j^2, \\ y_j' = y_{j-2} \ \ (j = 3, 4, \cdots, n+2). \end{cases} \tag{4}$$

さらに，$(1, 0)$ を追加する．

$$\boldsymbol{y}'' = (1, 0, y_1', \cdots, y_{n+2}'). \tag{5}$$

(4) Owner はマスター鍵 msk を生成し，User と共有する．具体的に，2 つの双対直交基底 $(\mathbb{B}, \mathbb{B}^*)$, $(\mathbb{D}, \mathbb{D}^*)$ である．ただし，$|\mathbb{B}| = |\mathbb{B}^*| = 2(n+4)$, $|\mathbb{D}| = |\mathbb{D}^*| = 2$ である．

$$\begin{cases} \mathbb{B} = [\boldsymbol{b}_1, \cdots, \boldsymbol{b}_{2n+8}], \ \mathbb{B}^* = [\boldsymbol{b}_1^*, \cdots, \boldsymbol{b}_{2n+8}^*], \\ \mathbb{D} = [\boldsymbol{d}_1, \boldsymbol{d}_2], \ \mathbb{D}^* = [\boldsymbol{d}_1^*, \boldsymbol{d}_2^*]. \end{cases} \tag{6}$$

双対直交基底なので，$\boldsymbol{b}_i \cdot \boldsymbol{b}_j^* = \delta_{ij},\ \boldsymbol{d}_i \cdot \boldsymbol{d}_j^* = \delta_{ij}$ を満たす．したがって，マスター鍵は，

$$\mathsf{msk} = [\mathbb{B}, \mathbb{B}^*, \mathbb{D}, \mathbb{D}^*]. \tag{7}$$

(5) User, Server, Owner の 3 者でペアリングに関する公開パラメタ PP を共有する．具体的に，有限体 $\mathbb{F}_p$，楕円曲線 $E/\mathbb{F}_p$，q 等分部分群 $E[q] \simeq \mathbb{Z}/q\mathbb{Z} \times \mathbb{Z}/q\mathbb{Z}$ とその位数 q $(\in \mathbb{P})$，座標 $\exists P \in G_1, \exists Q \in G_2$，そしてペアリング写像 $e_q : G_1 \times G_2 \to G_T$ を 3 者で共有する．したがって，公開パラメタは，

$$\mathsf{PP} = [G_1, G_2, G_T, P, Q, p, q]. \tag{8}$$

(6) User は，検索クエリ $Q_x = (Q_x^{(1)}, Q_x^{(2)})$ をつくり Server に送信する．$\beta, \beta^* \xleftarrow{U} \mathbb{F}_q$ を用いて，

$$\begin{cases} Q_x^{(1)} = \left(\beta \sum_{j=1}^{n+4} x_j'' \boldsymbol{b}_j + \beta^* \sum_{j=1}^{n+4} x_j'' \boldsymbol{b}_{n+4+j}\right) Q, \\ Q_x^{(2)} = (\beta \boldsymbol{d}_1 + \beta^* \boldsymbol{d}_2)\, Q. \end{cases} \tag{9}$$

(7) Owner は，参照データベース $D_y = (D_y^{(1)}, D_y^{(2)})$ をつくり Server に送信する．$\alpha, \alpha^* \xleftarrow{U} \mathbb{F}_q$ を用いて，

$$\begin{cases} D_y^{(1)} = \left(\alpha \sum_{j=1}^{n+4} y_j'' \boldsymbol{b}_j^* + \alpha^* \sum_{j=1}^{n+4} y_j'' \boldsymbol{b}_{n+4+j}^*\right) P, \\ D_y^{(2)} = (\alpha \boldsymbol{d}_1^* + \alpha^* \boldsymbol{d}_2^*)\, P. \end{cases} \tag{10}$$

(8) Server は，あらかじめ M 個の参照データベース D_y を Owner から得ている仮定とする．そして，User から検索クエリ Q_x を受け取り，次のようなペアリングの値を計算する．ここで，$g := e_q(P, Q)$ $(\neq 0)$ とすると，

$$\begin{cases} D_1 = e_q(D_y^{(1)}, Q_x^{(1)}) = g^{(\alpha\beta + \alpha^*\beta^*) \sum x_j'' y_j''} \pmod p, \\ D_2 = e_q(D_y^{(2)}, Q_x^{(2)}) = g^{\alpha\beta + \alpha^*\beta^*} \pmod p. \end{cases} \tag{11}$$

(9) Server は，(D_1, D_2, p) の離散対数問題を解き，$\sum x_j'' y_j'' = l_{xy}$ を得る．そして，Server は，l_{xy} からある基準によって何かを判断し，それに相当する何らかの情報を User に送信する．

2 実装上の注意点

(1) 簡単のため $w_j = 1$ とする．

(4) 双対直交基底 $(\mathbb{B}, \mathbb{B}^*)$ は，正準的な双対直交基底 $(\mathbb{A}, \mathbb{A}^*)$ をユニタリ行列 $X \xleftarrow{U} GL(n, q)$ によって線形変換することによってつくることが可能である．

(4-1) 正準的な直交基底の成分は，$\forall a_i \in \mathbb{F}_q, \forall a_i^* \in \mathbb{F}_q$ のとき，

$$\begin{cases} \boldsymbol{a}_1 = [a_1, 0, 0, \cdots, 0], \\ \boldsymbol{a}_2 = [0, a_2, 0, \cdots, 0], \\ \quad \vdots \\ \boldsymbol{a}_n = [0, 0, 0, \cdots, a_n], \end{cases} \qquad \begin{cases} \boldsymbol{a}_1^* = [a_1^{-1}, 0, 0, \cdots, 0], \\ \boldsymbol{a}_2^* = [0, a_2^{-1}, 0, \cdots, 0], \\ \quad \vdots \\ \boldsymbol{a}_n^* = [0, 0, 0, \cdots, a_n^{-1}]. \end{cases} \tag{12}$$

このとき，

$$\boldsymbol{a}_i \cdot \boldsymbol{a}_j^* = \delta_{ij} \pmod q. \tag{13}$$

(4-2) $r_j \xleftarrow{U} \mathbb{F}_q$ を用いて $(\mathbb{A}, \mathbb{A}^*)$ から $(\mathbb{B}, \mathbb{B}^*)$ に線形変換する．

$$\begin{cases} \boldsymbol{b}_1 = [a_1^{r_1}, 0, 0, \cdots, 0], \\ \boldsymbol{b}_2 = [0, a_2^{r_2}, 0, \cdots, 0], \\ \quad \vdots \\ \boldsymbol{b}_n = [0, 0, 0, \cdots, a_n^{r_n}], \end{cases} \quad \begin{cases} \boldsymbol{b}_1^* = [a_1^{-r_1}, 0, 0, \cdots, 0], \\ \boldsymbol{b}_2^* = [0, a_2^{-r_2}, 0, \cdots, 0], \\ \quad \vdots \\ \boldsymbol{b}_n^* = [0, 0, 0, \cdots, a_n^{-r_n}]. \end{cases} \tag{14}$$

$\mathbb{D}, \mathbb{D}^*$ も同様に，$\bar{a}_j \xleftarrow{U} \mathbb{F}_q$, $\bar{r}_j \xleftarrow{U} \mathbb{F}_q$ において，

$$\boldsymbol{d}_1 = [\bar{a}_1^{\bar{r}_1}, 0], \quad \boldsymbol{d}_2 = [0, \bar{a}_2^{\bar{r}_2}], \ \boldsymbol{d}_1^* = [(\bar{a}_1^{\bar{r}_1})^{-1}, 0], \quad \boldsymbol{d}_2^* = [0, (\bar{a}_2^{\bar{r}_2})^{-1}]. \tag{15}$$

(6) (4) に従うと，

$$Q_x^{(1)} = \begin{bmatrix} \beta x_1'' b_1 Q, & \beta^* x_1'' b_{n+5} Q, \\ \beta x_2'' b_2 Q, & \beta^* x_2'' b_{n+6} Q, \\ \vdots & \vdots \\ \beta x_{n+4}'' b_{n+4} Q, & \beta^* x_{n+4}'' b_{2n+8} Q, \end{bmatrix}, \ Q_x^{(2)} = [\beta d_1 Q, \beta^* d_2 Q]. \tag{16}$$

(7) (4) に従うと，

$$D_y^{(1)} = \begin{bmatrix} \alpha y_1'' b_1^* P, & \alpha^* y_1'' b_{n+5}^* P, \\ \alpha y_2'' b_2^* P, & \alpha^* y_2'' b_{n+6}^* P, \\ \vdots & \vdots \\ \alpha y_{n+4}'' b_{n+4}^* P, & \alpha^* y_{n+4}'' b_{2n+8}^* P, \end{bmatrix}, \ D_y^{(2)} = [\alpha d_1^* P, \alpha^* d_2^* P]. \tag{17}$$

(8) $g := e_q(P, Q)$ $(\neq 0)$ とするとき，ペアリングは次のように計算する．

$$\begin{aligned} e_q(D_y^{(1)}, Q_x^{(1)}) &= \prod_{j=1}^{n+4} e_q(\alpha y_j'' b_j^* P, \beta x_j'' b_j Q) \times \prod_{j=1}^{n+4} e_q(\alpha^* y_j'' b_{j+n+4}^* P, \beta^* x_j'' b_{j+n+4} Q) \\ &= g^{\alpha\beta \sum x_j'' y_j''} \times g^{\alpha^*\beta^* \sum x_j'' y_j''} = g^{(\alpha\beta + \alpha^*\beta^*) \sum x_j'' y_j''} \pmod p, \end{aligned} \tag{18}$$

$$e_q(D_y^{(2)}, Q_x^{(2)}) = e_q(\alpha d_1^* P, \beta d_1 Q) \times e_q(\alpha^* d_2^* P, \beta^* d_2 Q) = g^{\alpha\beta + \alpha^*\beta^*} \pmod p. \tag{19}$$

3 安全性について

図 1 のような系の場合，User と Owner は Server に情報を送信し，Server も User に情報を返信する．したがって，Owner は，相手のベクトルに関する部分情報を一切得ることはできない．しかしながら，Owner は User と Server との通信を盗聴する可能性がある．ただし，Owner が Server に参照データベースを置いてからサービスが開始するため，User が Owner と Server の通信を盗聴することを想定しない．したがって，Semi-honest-Server，Malicious-Server，Malicious-Owner について議論する．

(1) Semi-honest-Server

プロトコルに従うが，その途中で得られた情報から積極的に x_j'', y_j'' を推測しようとする Server を Semi-honest であると定義する．つまり，安全性の要件として，プロトコル実行中にベクトルの成分の特定に繋がる部分情報を漏らさないことを要求する．

(6), (7) での安全性の仮定は，楕円離散対数問題である．$\beta x''_j b_j P$ や $\alpha y''_j b^*_j Q$ から $\beta x''_j b_j, \alpha y''_j b^*_j y_j$ を，延いては $\alpha, \beta, x''_j, y''_j$ を求めることはできない．したがって，Server は $\beta x''_j b_j, \alpha y''_j b^*_j$ を求めることはできない．しかしながら，離散対数問題を解くことが可能であると仮定しているので，Server はそれらの積を求めることができる．ただし，乱数 α, β は公開されていないので，その積から x''_j, y''_j を特定することはできない．

(2) Malicious-Server

プロトコルを逸脱し，勝手な振る舞いによって得られた情報から x_j, y_j を推測しようとする Server を Malicious であると定義する．つまり，安全性の要件として，プロトコルなどで得られた情報からベクトルの成分の特定に繋がらないような困難さを要求する．

まず，Server は，ペアリングによって楕円離散対数問題を離散対数問題に還元することで係数を求めることが可能である．

$$\begin{cases} \alpha y''_j b^*_j,\ \alpha^* y''_j b^*_{j+5},\ \alpha d^*_1,\ \alpha^* d^*_2, \\ \beta x''_j b_j,\ \beta^* x''_j b^*_{j+5},\ \beta d_1,\ \beta^* d_2, \\ \alpha\beta x''_j y''_j,\ \alpha^*\beta^* x''_j y''_j. \end{cases} \tag{20}$$

しかし，マスター鍵：$(\mathbb{B}, \mathbb{B}^*, \mathbb{D}, \mathbb{D}^*)$ と乱数：$(\alpha, \alpha^*, \beta, \beta^*)$ を同時に特定することはできないので，Malicious といへどもベクトルの成分の特定は困難であるといえる．

次に，検索クエリや参照データベースの識別可能性について議論するために，次のようなゲームを考える．図 2 の Challenger は，Owner と User に相当し，Adversary は Malicious-Server に相当する．

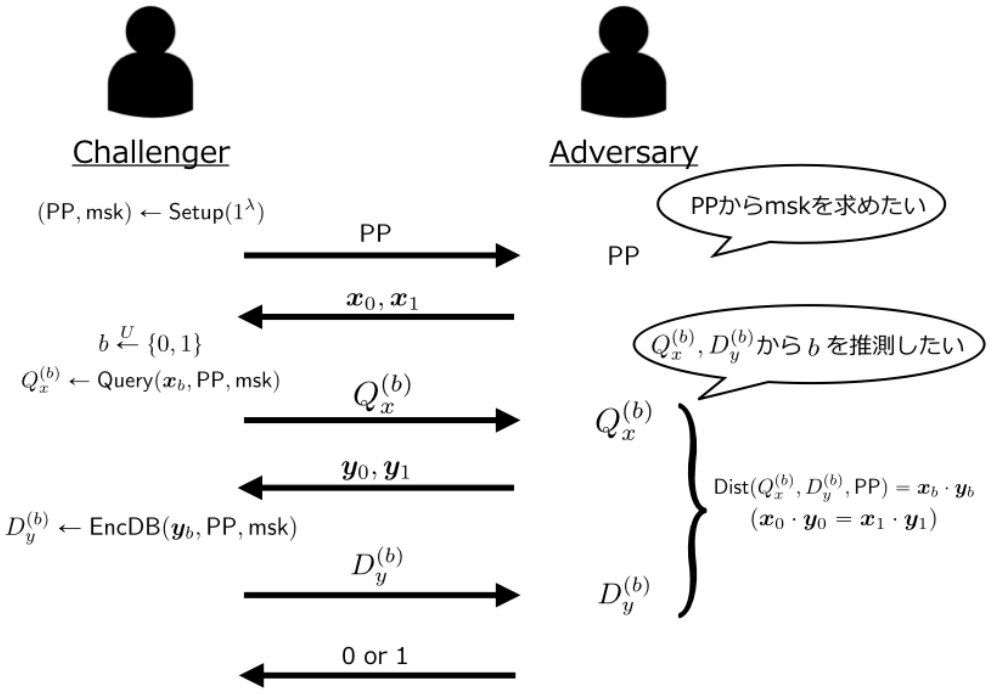

図 2　Security Game

検索クエリ Q_x や参照データベース D_y は，乱数 $\alpha, \alpha^*, \beta, \beta^*$ を用いて生成する．これらの乱数は Q_x, D_y を生成する度に生成するため，Adversary はこの乱数を Q_x, D_y から特定することができなければ，Q_x, D_y を多く集めても識別することは不可能である．しかし，前述のとおり Server は，乱数を特定することはできない．また，双対直交基底も乱数を用いて生成しているので，Owner から msk についての情報が漏れない限り，Server は msk を特定することはできない．したがって，Q_x, D_y からベクトルの成分を特定するための部分情報は一切漏れない．

(3) Malicious-Owner

User と Server の通信を盗聴することによって，x_j を推測しようとする Owner を Malicious であると定義する．つまり，安全性の要件として，盗聴した情報からベクトルの成分の特定に繋がらないような困難さを要求する．

まず，Owner は次のようにペアリングにより還元して乱数を求める．

$$\begin{cases} e_q(d_1^* P, \beta d_1 Q) = g^{\beta} \pmod p, \\ e_q(d_2 P, \beta^* d_2 Q) = g^{\beta^*} \pmod p. \end{cases} \tag{21}$$

そして Owner は，$\alpha\beta x_j'' y_j''$ から α, β, y_j を用いて x_j を特定する．このように Owner は盗聴することで簡単に User のベクトルの成分を求めることができてしまう．そこで，SSL などのセキュア通信によって安全性を確保する．

4 効率性について

通信量と計算量は，それぞれ次の表 1,2 のとおりである．

	User	Server	Owner
(6)	$\mathcal{O}(n)$	–	–
(7)	–	–	$\mathcal{O}(Mn)$

表 1　Computation cost of User, Server and Owner

	User	Server	Owner
(2)	$\mathcal{O}(n)$	–	–
(3)	–	–	$\mathcal{O}(n)$
(6)	$\mathcal{O}(n)$	–	–
(7)	–	–	$\mathcal{O}(n)$
(8)	–	$\mathcal{O}(Mn)$	–
(9)	–	–	$\mathcal{O}(M)$

表 2　Communication cost of User, Server and Owner