

HQC暗号を応用した秘匿内積計算プロトコル（II）

中山 太雅^{1,a)} 廣友 雅徳^{2,b)} 福田 洋治³ 毛利 公美⁴ 白石 善明⁵

概要：ビックデータ解析、データマイニングでは扱うデータに個人情報などの秘密情報が含まれるため、プライバシーを保護したまま計算できる秘匿計算が注目されている。それらの秘匿計算プロトコルは、素因数分解や離散対数問題などの数論ベースの問題の困難性を利用した公開鍵暗号を応用して設計されており、実用的な量子計算機が実現した場合、秘匿計算プロトコルの安全性は保てなくなる。耐量子暗号として符号ベース暗号方式がいくつか提案されている。Gaborit らは準巡回シンドローム復号問題に基づいた公開鍵暗号方式 HQC を NIST のポスト量子暗号標準化コンペティションへ提案している。筆者らは、HQC 暗号を応用し、耐量子性を有する内積の秘匿計算プロトコルを提案している。本稿では、秘匿内積計算プロトコルの正当性と安全性について考察する。

キーワード：秘匿計算、内積、HQC 暗号、耐量子性

A Secure Computation Protocol of Inner Product Using HQC Cryptosystem (II)

TAIGA NAKAYAMA^{1,a)} MASANORI HIROTOMO^{2,b)} YOUJI FUKUTA³ MASAMI MOHRI⁴
YOSHIAKI SHIRAISHI⁵

Abstract: In big data analysis and data mining, confidential data such as personal information is included in the data to be handled, so confidential calculation that can be calculated while protecting privacy is drawing attention. These secret computation protocols are designed by applying public key cryptography that utilizes the difficulty of number theory-based problems such as prime factorization and discrete logarithm problems. The security of the protocol cannot be maintained. Several code-based cryptosystems have been proposed as quantum resistant cryptosystems. Gaborit et al. have proposed a public key cryptosystem HQC based on the quasi-cyclic syndrome decryption problem to the NIST post-quantum cryptographic standardization competition. We have applied the HQC cryptosystem and have proposed a quantum-resistant secure computation protocol of inner product. In this paper, we consider the validity and security of the quantum-resistant secure computation protocol of inner product by applying HQC cryptosystem.

Keywords: Secure Computation, Inner Product, HQC Cryptosystem, Post-Quantum

¹ 佐賀大学大学院理工学研究科
Graduate of Science and Engineering, Saga University
² 佐賀大学理工学部
Faculty of Science and Engineering, Saga University
³ 近畿大学理工学部
Faculty of Science and Engineering, Kindai University
⁴ 岐阜大学工学部
Faculty of Engineering, Gifu University
⁵ 神戸大学大学院工学研究科
Graduate School of Engineering, Kobe University
a) nakayam@ma.is.saga-u.ac.jp
b) hirotomo@cc.saga-u.ac.jp

1. はじめに

情報化社会の発展がめざましい現代では、人々が持つ情報は数値化されてインターネットと繋がっている。インターネットで流される個人情報の保護が重要な課題となっている。最近では、個人情報を社会や産業の発展のために二次利用する機運が高まってきている。しかし個人情報の扱いは、情報提供者である個人のプライバシーについて十

分配慮する必要がある。情報を安全に解析するため、数値化されたデータは暗号化されてから様々な操作が行われる。例えば、ビッグデータ解析やデータマイニングは多くのデータを分析する。しかし、分析するデータに個人情報が含まれるという問題があるため、プライバシーを保護したまま計算できる秘匿計算という技術が注目されている。現在広く用いられている秘匿計算プロトコルの大部分は、素因数分解や離散対数問題などの数論ベースの問題の困難性に基づいて安全性が保障されている。しかし、量子コンピュータが実現されると素因数分解、離散対数問題が効率よく解けてしまうため秘匿計算プロトコルの安全性は保てなくなる。そのため、量子コンピュータでも解くことができない問題に基づいた秘密計算プロトコルを開発する必要がある。そこで、符号理論の問題をベースにした秘密計算プロトコルは量子コンピュータの攻撃に耐えうると考えられている。Gaborit らが提案した準巡回シンドローム復号問題に基づいた公開鍵暗号方式 HQC[1][2] は、符号ベースの暗号であるため攻撃に耐えうることができる。文献 [3] で、HQC 暗号を二者間秘匿計算に応用し、準巡回シンドローム復号問題に基づいた線形計算の秘匿計算プロトコルを提案している。筆者らは文献 [4] で、HQC 暗号を二者間秘匿計算に応用し、準巡回シンドローム復号問題に基づいた内積の秘匿計算プロトコルを提案している。

本稿では、筆者らが文献 [4] で提案した HQC 暗号を応用した準巡回シンドローム復号問題に基づいた内積の秘匿計算プロトコルの正当性と安全性について考察する。

2. 準備

2.1 表記

\mathbb{F} を体とし、 \mathbb{F}_2 を二元体とする。 \mathbb{F}_2^n を二元体 \mathbb{F}_2 上の n 次元ベクトル空間とする。 \mathcal{R} を係数が \mathbb{F} の $X^n - 1$ を法とする多項式環 $\mathbb{F}[X]/\langle X^n - 1 \rangle$ とする。本稿では、 \mathcal{R} の元と n 次元ベクトルを同一視する。すなわち、 $n-1$ 次の多項式 $a(X) = a_0 + a_1 X + \cdots + a_{n-1} X^{n-1}$ と $\mathbf{a} = (a_0, a_1, \dots, a_{n-1})$ を同一視する。このとき、 $\mathbf{x}, \mathbf{y} \in \mathcal{R}$ の積 $\mathbf{z} = \mathbf{x} \cdot \mathbf{y} \in \mathcal{R}$ は次のようになる。

$$z_k = \sum_{i+j \equiv k+1 \pmod{n}} x_i y_j, \quad \text{for } k = 1, 2, \dots, n-1 \quad (1)$$

定義 1. $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathcal{R}$ による巡回行列を次のように定義する。

$$\text{rot}(\mathbf{x}) = \begin{bmatrix} x_1 & x_n & \cdots & x_2 \\ x_2 & x_1 & \cdots & x_3 \\ \vdots & \vdots & \ddots & \vdots \\ x_n & x_{n-1} & \cdots & x_1 \end{bmatrix} \in \mathbb{F}^{n \times n} \quad (2)$$

$\mathbf{x}, \mathbf{y} \in \mathcal{R}$ の積に次の関係が成り立つ。

$$\mathbf{x} \cdot \mathbf{y} = \mathbf{x} \times \text{rot}(\mathbf{y})^\top = (\text{rot}(\mathbf{x}) \times \mathbf{y}^\top)^\top = \mathbf{y} \cdot \mathbf{x} \quad (3)$$

ただし、 T は転置記号である。

C を符号長 n 、情報点数 k の線形符号とする ($[n, k]$ 符号と表記する)。符号 C の最小距離を d で表し、誤り訂正能力を $\delta (= \lfloor (d-1)/2 \rfloor)$ で表す。また、符号 C の生成行列を \mathbf{G} で表し、パリティ検査行列を \mathbf{H} で表す。

定義 2. \mathbb{F}_2^{sn} のベクトル $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_s)$ を s 個の連続するブロックとして表示する。 $[sn, k, d]$ 線形符号 C は、任意の $\mathbf{c} = (\mathbf{c}_1, \dots, \mathbf{c}_s) \in C$ について、すべてのブロック $\mathbf{c}_1, \dots, \mathbf{c}_s$ に巡回シフトを適用した後に得られるベクトルも符号語であるとき、符号 C をインデックス s の準巡回符号 (QC) という。より正式には、各ブロック \mathbf{c}_i を $\mathcal{R} = \mathbb{F}[X]/(X^n - 1)$ の多項式と見なすことにより、符号 C は、任意の $\mathbf{c} = (\mathbf{c}_1, \dots, \mathbf{c}_s) \in C$ が $(X \cdot \mathbf{c}_1, \dots, X \cdot \mathbf{c}_s) \in C$ である場合、インデックス s の QC である。

定義 3. 次のパリティ検査行列で定義される準巡回符号を組織的準巡回符号と呼ぶ。

$$\mathbf{H} = \begin{bmatrix} \mathbf{I}_n & 0 & \cdots & 0 & \mathbf{A}_1 \\ 0 & \mathbf{I}_n & \cdots & 0 & \mathbf{A}_2 \\ \vdots & \vdots & \ddots & 0 & \vdots \\ 0 & 0 & \cdots & \mathbf{I}_n & \mathbf{A}_{s-1} \end{bmatrix} \quad (4)$$

ただし、 $\mathbf{A}_1, \dots, \mathbf{A}_{s-1}$ は $n \times n$ 巡回行列である。

2.2 安全性仮定

定義 4 (シンドローム復号分布). ω を \mathcal{R} 上のノルムとする。 n, k および w が正の整数の場合、SD(n, k, w) 分布は、 $\omega(\mathbf{x}) = w$ となるように $\mathbf{H} \xleftarrow{\$} \mathbb{F}^{(n-k) \times n}$ および $\mathbf{x} \xleftarrow{\$} \mathbb{F}^n$ を選択し、 $(\mathbf{H}, \sigma(\mathbf{x}) = \mathbf{H}\mathbf{x}^\top)$ を出力する。

定義 5 (シンドローム復号問題). SD 分布からの入力を $(\mathbf{H}, \mathbf{y}^\top) \in \mathbb{F}^{(n-k) \times n} \times \mathbb{F}^{(n-k)}$ としたとき、シンドローム復号問題 SD(n, k, w) は、 $\mathbf{H}\mathbf{x}^\top = \mathbf{y}^\top$ および $\omega(\mathbf{x}) = w$ となる $\mathbf{x} \in \mathbb{F}^n$ を求めるという問題である。

定義 6 (シンドローム復号決定仮定). $(\mathbf{H}, \mathbf{y}^\top) \xleftarrow{\$} \mathbb{F}^{(n-k) \times n} \times \mathbb{F}^{(n-k)}$ を入力としたとき、シンドローム復号決定仮定 DSD(n, k, w) は、 (\mathbf{H}, \mathbf{y}) が SD(n, k, w) 分布から得られたものか、 $\mathbb{F}^{(n-k) \times n} \times \mathbb{F}^{(n-k)}$ 上の分布から選んだ一様乱数かを無視できない確率で区別することができないという仮定である。

定義 7 (s-準巡回シンドローム復号問題). $\mathbf{y} \xleftarrow{\$} \mathbb{F}^{sn-n}$ およびインデックス s の組織的準巡回符号 C のランダムなパリティ検査行列 \mathbf{H} の n, w, s が正の整数の場合、s-準巡回シンドローム復号問題 s-QCSD(n, w) は、 $\mathbf{x}\mathbf{H}^\top = \mathbf{y}$ および $\omega(\mathbf{x}_i) = w, (i = 1, \dots, s)$ である $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_s) \in \mathbb{F}^{sn}$ を求めるという問題である。

定義 8 (s-準巡回シンドローム復号決定仮定). $\mathbf{y} \xleftarrow{\$} \mathbb{F}^{sn}$ および組織的準巡回符号 C のランダムなパリティ検査行列 \mathbf{H} の n, w, s が正の整数で b ビットの場合、s-準巡回シンドローム復号決定仮定 s-DQCSD(n, w, b) は、 (\mathbf{H}, \mathbf{y}) が

s -QCSD(n, w) 分布からのものか, $\mathbb{F}^{(sn-n) \times sn} \times \mathbb{F}^{(sn-n)}$ のパリティ b のベクトル上の分布から選んだ一様乱数かを無視できない確率で区別することができないという仮定である。

3. HQC 暗号

本章では, Gaborit らが提案した HQC (Hamming Quasi-Cyclic) 暗号について説明する。HQC 暗号は準巡回シンドローム復号問題の困難性を利用した公開鍵暗号である。文献 [2] では, 準巡回符号として, 巡回符号である 2 元 BCH 符号と繰り返し符号を組み合わせたテンソル積符号を用いている。

HQC 暗号

セットアップ: グローバルパラメータ $\text{param} = (n, k, \delta, w, w_r, w_e)$ と符号 C の生成行列 $\mathbf{G} \in \mathbb{F}^{k \times n}$ を設定する。ただし, w, w_r, w_e は誤り訂正能力の範囲内で設定する。

鍵生成: ランダムに $\mathbf{h} \xleftarrow{\$} \mathcal{R}$ を生成する。さらに, $(\mathbf{x}, \mathbf{y}) \xleftarrow{\$} \mathcal{R}^2$ を生成する。ただし, \mathbf{x}, \mathbf{y} のハミング重みは w である。秘密鍵を $\text{sk} = (\mathbf{x}, \mathbf{y})$ とし, 公開鍵を $\text{pk} = (\mathbf{h}, \mathbf{s} = \mathbf{x} + \mathbf{h} \cdot \mathbf{y})$ とする。

暗号化: ランダムに $\mathbf{e} \xleftarrow{\$} \mathcal{R}$ と $(\mathbf{r}_1, \mathbf{r}_2) \xleftarrow{\$} \mathcal{R}^2$ を生成する。ただし, \mathbf{e} のハミング重みは w_e であり, $\mathbf{r}_1, \mathbf{r}_2$ のハミング重みは w_r である。メッセージ \mathbf{m} から $\mathbf{u} = \mathbf{r}_1 + \mathbf{h} \cdot \mathbf{r}_2$ と $\mathbf{v} = \mathbf{m} \cdot \mathbf{G} + \mathbf{s} \cdot \mathbf{r}_2 + \mathbf{e}$ を計算する。暗号文を (\mathbf{u}, \mathbf{v}) とする。

復号: 符号 C の復号法を用いて $\mathbf{v} - \mathbf{u} \cdot \mathbf{y}$ を復号し, メッセージ \mathbf{m} に復元する。

HQC 暗号では, 暗号化するときに, 符号 C で符号化したメッセージ \mathbf{m} に公開鍵 \mathbf{s} を加える。 \mathbf{s} は準巡回符号で生成したハミング重みが大きい誤りであるため, 準巡回シンドローム復号決定仮定によって安全性が保証される。また, 復号では, 秘密鍵を用いて, 暗号文に含まれる \mathbf{s} に関する大きな誤りを取り除くことができる。しかし, $\mathbf{x} \cdot \mathbf{r}_2 - \mathbf{r}_1 \cdot \mathbf{y} + \mathbf{e}$ の誤りが残る。この誤りのハミング重みが符号 C の誤り訂正能力 δ より小さければ, 正しく復号できる。文献 [2] では, ハミング重み $w, w_r, w_e = O(\sqrt{n})$ を想定して解析をし, 符号長 n が大きくなるほど $w(\mathbf{x} \cdot \mathbf{r}_2 - \mathbf{r}_1 \cdot \mathbf{y} + \mathbf{e}) \leq \delta$ となる確率が高くなると示されている。

4. 秘匿内積計算プロトコル

4.1 提案プロトコル

筆者らは文献 [4] で, HQC 暗号を応用した二者間の秘匿内積計算プロトコルを提案している。参加者は A と B で, A の入力は二元体上の m 次元ベクトル (a_1, a_2, \dots, a_m) , B の入力は二元体上の m 次元ベクトル (b_1, b_2, \dots, b_m) とし, A が内積 $a_1b_1 + a_2b_2 + \dots + a_mb_m$ を得ることができると

いうプロトコルである。本節では, そのプロトコルについて説明する。

内積秘匿計算プロトコル

入力 A: $(a_1, a_2, \dots, a_m) \in \mathbb{F}_2^m$

B: $(b_1, b_2, \dots, b_m) \in \mathbb{F}_2^m$

出力 A: $a_1b_1 + a_2b_2 + \dots + a_mb_m$

B: \perp

セットアップ: グローバルパラメータ $\text{param} = (n, k, \delta, w, w_r, w_e)$ と符号 C の生成行列 $\mathbf{G} \in \mathbb{F}^{k \times n}$ を設定する。ただし, w, w_r, w_e は誤り訂正能力の範囲内で設定する。

鍵生成: A は次のように秘密鍵と公開鍵を生成する。

(1) ランダムに $\mathbf{h} \xleftarrow{\$} \mathcal{R}$ を生成する。さらに, $(\mathbf{x}, \mathbf{y}) \xleftarrow{\$} \mathcal{R}^2$ を生成する。ただし, \mathbf{x}, \mathbf{y} のハミング重みは w である。

(2) $\mathbf{s} = \mathbf{x} + \mathbf{h} \cdot \mathbf{y}$ を計算する。そして, 秘密鍵, 公開鍵を次のようにする。

$$\text{sk} = (\mathbf{x}, \mathbf{y}) \quad (5)$$

$$\text{pk} = (\mathbf{h}, \mathbf{s}) \quad (6)$$

暗号化: A は次の処理を行う。

(1) 各入力 a_1, \dots, a_m に 0 をパディングし, k 次元ベクトル

$$\mathbf{a}_i = (a_i, 0, 0, \dots, 0), \quad i = 1, 2, \dots, m \quad (7)$$

を生成する。

(2) ランダムに $\mathbf{r}_{A_1}, \dots, \mathbf{r}_{A_m}, \mathbf{r}_{u_1}, \dots, \mathbf{r}_{u_m}, \mathbf{r}_{v_1}, \dots, \mathbf{r}_{v_m} \xleftarrow{\$} \mathcal{R}$ を生成する。ただし, $\mathbf{r}_{A_1}, \dots, \mathbf{r}_{A_m}, \mathbf{r}_{u_1}, \dots, \mathbf{r}_{u_m}, \mathbf{r}_{v_1}, \dots, \mathbf{r}_{v_m}$ のハミング重みは w_r である。

(3) 次式を計算し, 公開鍵 pk と一緒に B に送る。

$$\mathbf{u}_i = \mathbf{h} \cdot \mathbf{r}_{A_i} + \mathbf{r}_{u_i}, \quad i = 1, 2, \dots, m \quad (8)$$

$$\mathbf{v}_i = \mathbf{a}_i \cdot \mathbf{G} + \mathbf{s} \cdot \mathbf{r}_{A_i} + \mathbf{r}_{v_i}, \quad i = 1, 2, \dots, m \quad (9)$$

内積計算: B は次の処理を行う。

(1) ランダムに $\mathbf{r}_B \xleftarrow{\$} \mathcal{R}$ と $(\mathbf{e}_u, \mathbf{e}_v) \xleftarrow{\$} \mathcal{R}^2$ を生成する。ただし, \mathbf{r}_B のハミング重みは w_r であり, $\mathbf{e}_u, \mathbf{e}_v$ のハミング重みは w_e である。

(2) 入力 (b_1, b_2, \dots, b_m) と A から受け取った $\mathbf{u}_1, \dots, \mathbf{u}_m, \mathbf{v}_1, \dots, \mathbf{v}_m$ を用いて次式を計算する。

$$\mathbf{u}' = b_1\mathbf{u}_1 + \dots + b_m\mathbf{u}_m + \mathbf{h} \cdot \mathbf{r}_B + \mathbf{e}_u \quad (10)$$

$$\mathbf{v}' = b_1\mathbf{v}_1 + \dots + b_m\mathbf{v}_m + \mathbf{s} \cdot \mathbf{r}_B + \mathbf{e}_v \quad (11)$$

そして, \mathbf{u}', \mathbf{v}' を A に送り返す。

復号： 符号 C の復号法を用いて $\mathbf{v}' - \mathbf{u}' \cdot \mathbf{y}$ を復号し，その結果の最初の 1 ビットを取ることで $a_1 b_1 + a_2 b_2 + \cdots + a_m b_m$ を復元する。 ■

復号の段階で行う $\mathbf{v}' - \mathbf{u}' \cdot \mathbf{y}$ の計算は，式を整理すると，

$$\begin{aligned} & \mathbf{v}' - \mathbf{u}' \cdot \mathbf{y} \\ &= (a_1 b_1 + \cdots + a_m b_m) \mathbf{G} \\ &+ \mathbf{x} \cdot ((\mathbf{b}_1 \cdot \mathbf{r}_{A_1} + \cdots + \mathbf{b}_m \cdot \mathbf{r}_{A_m}) + \mathbf{r}_B) \\ &- \mathbf{y} \cdot ((\mathbf{b}_1 \cdot \mathbf{r}_{u_1} + \cdots + \mathbf{b}_m \cdot \mathbf{r}_{u_m}) + \mathbf{e}_u) \\ &+ ((\mathbf{b}_1 \cdot \mathbf{r}_{v_1} + \cdots + \mathbf{b}_m \cdot \mathbf{r}_{v_m}) + \mathbf{e}_v) \end{aligned} \quad (12)$$

となる。そのため，符号 C の復号法を用いて

$$\begin{aligned} & \mathbf{x} \cdot ((\mathbf{b}_1 \cdot \mathbf{r}_{A_1} + \cdots + \mathbf{b}_m \cdot \mathbf{r}_{A_m}) + \mathbf{r}_B) \\ &- \mathbf{y} \cdot ((\mathbf{b}_1 \cdot \mathbf{r}_{u_1} + \cdots + \mathbf{b}_m \cdot \mathbf{r}_{u_m}) + \mathbf{e}_u) \\ &+ ((\mathbf{b}_1 \cdot \mathbf{r}_{v_1} + \cdots + \mathbf{b}_m \cdot \mathbf{r}_{v_m}) + \mathbf{e}_v) \end{aligned} \quad (13)$$

を誤りとして取り除く必要性がある。

4.2 生成するベクトルについて

本節では $m = 2$ および $m = 3$ のときに提案プロトコルで用いるベクトル $\mathbf{r}_{A_1}, \dots, \mathbf{r}_{A_m}, \mathbf{r}_{u_1}, \dots, \mathbf{r}_{u_m}, \mathbf{r}_{v_1}, \dots, \mathbf{r}_{v_m}$ の生成手法について述べる。誤りベクトルである式 (13) は多くのベクトルが含まれているため，誤りベクトルの重みが大きくなると復号できない。そのため，ベクトルの生成の仕方を工夫する必要がある。提案手法では，生成するベクトルの重みの乗せ方を工夫することで復号を可能にする。

生成手法 ($m = 2$)： A は図 1 のような形でベクトルを生成する。○が付いている部分は重みを乗せていることを表している。

r_{A_1}	○	○	
r_{A_2}	○	○	○

図 1 例： r_{A_1}, r_{A_2} の生成手法

- (1) 生成する各ベクトルを 3 つに分割する。分割した各部分を先頭から 0 番目, 1 番目, 2 番目とする。
 $i(i = 0, 1, 2)$ 番目に重みを乗せるとき，各ベクトルは同じ位置に重みを乗せる。
 - (2) すべてのベクトルの 0 番目に重みを乗せる。
 - (3) 1 番目と 2 番目は各番目で 1 つのベクトルのみが重みを持つように重みを乗せる。
- この生成手法を用いることで式 (13) の重みを制限することができる。重みを乗せる部分にはそれぞれ $w_r/4$ の重みを乗せる。すべてのベクトルは 4 つの部分に重みを乗せるため，各ベクトルは重み w_r になる。また，各ベクトルをどの組み合わせで足しても重み w_r になる。そのため，復号を行うとき最大誤り訂正能力の範囲内で受信語を訂正できる。

とができる。重みを乗せる部分にはそれぞれ $w_r/2$ の重みを乗せる。すべてのベクトルは 2 つの部分に重みを乗せるため，各ベクトルは重み w_r になる。また，各ベクトルをどの組み合わせで足しても重み w_r になる。そのため，復号を行うとき最大誤り訂正能力の範囲内で受信語を訂正できる。

生成手法 ($m = 3$)： A は図 2 のような形でベクトルを生成する。○が付いている部分は重みを乗せていることを表している。

r_{A_1}	○	○		○	○		
r_{A_2}	○	○	○			○	
r_{A_3}	○		○	○			○

図 2 例： $r_{A_1}, r_{A_2}, r_{A_3}$ の生成手法

- (1) 生成する各ベクトルを 7 つに分割する。分割した各部分を先頭から 0 番目, 1 番目, ..., 6 番目とする。
 $i(i = 0, 1, \dots, 6)$ 番目に重みを乗せるとき，各ベクトルは同じ位置に重みを乗せる。
- (2) すべてのベクトルの 0 番目に重みを乗せる。
- (3) 1 から 3 番目までは，各番目で 2 つのベクトルが重みを持つように重みを乗せる。
- (4) 4 から 6 番目は，各番目で 1 つのベクトルのみが重みを持つように重みを乗せる。

この生成手法を用いることで式 (13) の重みを制限することができる。重みを乗せる部分にはそれぞれ $w_r/4$ の重みを乗せる。すべてのベクトルは 4 つの部分に重みを乗せるため，各ベクトルは重み w_r になる。また，各ベクトルをどの組み合わせで足しても重み w_r になる。そのため，復号を行うとき最大誤り訂正能力の範囲内で受信語を訂正できる。

5. 誤りベクトル分布の分析

提案プロトコルでは式 (13) の誤りベクトルが生まれる。式 (13) の誤りベクトルを \mathbf{e}' とおくと，提案プロトコルは次の式を満たす必要がある。

$$w(\mathbf{e}') \leq \delta \quad (14)$$

本章の目的は，式 (14) の条件が成立する \mathbf{e}' の分布を分析することである。

ベクトル $\mathbf{x}, \mathbf{y}, (\mathbf{b}_1 \cdot \mathbf{r}_{A_1} + \cdots + \mathbf{b}_m \cdot \mathbf{r}_{A_m}), (\mathbf{b}_1 \cdot \mathbf{r}_{u_1} + \cdots + \mathbf{b}_m \cdot \mathbf{r}_{u_m}), (\mathbf{b}_1 \cdot \mathbf{r}_{v_1} + \cdots + \mathbf{b}_m \cdot \mathbf{r}_{v_m}), \mathbf{r}_B, \mathbf{e}_u, \mathbf{e}_v$ は，重み w ,

w_r , または w_e のベクトルから均一かつ独立して選択されるようになっている。非常に近い確率モデルは、これらの独立したベクトルすべてが、2次元ベクトルの内積計算の場合は座標がパラメータ $p = w/n$ (または $p_{r_1} = w_r/n$, $p_{r_2} = 3w_r/4n$, $p_e = w_e/n$), 3次元ベクトルの内積計算の場合は座標がパラメータ $p = w/n$ (または $p_{r_1} = w_r/n$, $p_{r_2} = 7w_r/8n$, $p_e = w_e/n$)の独立したベルヌーイ変数であるランダムなベクトルの分布に従うように選択されたときである。分析を簡単にするために、このモデルを想定し、次のような流れで誤りベクトル e' の分布を調べる。

I: 和 $(\mathbf{b}_1 \cdot \mathbf{r}_{A_1} + \dots + \mathbf{b}_m \cdot \mathbf{r}_{A_m}) + \mathbf{r}_B$ の分布を調べる。

$\mathbf{b}_1 \cdot \mathbf{r}_{A_1} + \dots + \mathbf{b}_m \cdot \mathbf{r}_{A_m} = \mathbf{r}_A = (R_{A_1}, \dots, R_{A_m})$, $\mathbf{r}_B = (R_{B_1}, \dots, R_{B_n})$ をランダムなベクトルとする。ここで R_{A_i} (または R_{B_i}) は $P[R_{A_i} = 1] = p_{r_2}$ (または $P[R_{B_i} = 1] = p_{r_1}$) のようにパラメータ p_{r_2} (または p_{r_1}) の独立したベルヌーイ変数である。 \mathbf{r}_A と \mathbf{r}_B が独立していると仮定し、 \mathbf{r}_A と \mathbf{r}_B の和を $\mathbf{z}_1 = (\mathbf{b}_1 \cdot \mathbf{r}_{A_1} + \dots + \mathbf{b}_m \cdot \mathbf{r}_{A_m}) + \mathbf{r}_B = (z_{11}, \dots, z_{1n})$ と表すと、以下の式が求まる。

$$\begin{cases} \Pr[z_{1k} = 1] = p_{r_1}(1 - p_{r_2}) + p_{r_2}(1 - p_{r_1}) \\ \Pr[z_{1k} = 0] = p_{r_1}p_{r_2} + (1 - p_{r_1})(1 - p_{r_2}) \end{cases} \quad (15)$$

そして、 $\Pr[z_{1k} = 1]$ を p_1 とする。

II: 次に、積 $\mathbf{x} \cdot ((\mathbf{b}_1 \cdot \mathbf{r}_{A_1} + \dots + \mathbf{b}_m \cdot \mathbf{r}_{A_m}) + \mathbf{r}_B)$ の分布を調べる。ランダムなベクトル \mathbf{x} を $\mathbf{x} = (X_1, \dots, X_n)$ とする。ここで X_i はパラメータ p の独立したベルヌーイ変数とする。 \mathbf{x} と \mathbf{z}_1 の積を $\mathbf{z}_2 = \mathbf{x} \cdot \mathbf{z}_1 = (z_{21}, \dots, z_{2n})$ と表すと、以下の式が求まる。

$$\Pr[z_{2k} = 1] = \sum_{i=0, i \text{ odd}}^n \binom{n}{i} (pp_1)^i (1 - pp_1)^{n-i} \quad (16)$$

この式は簡単に次の式で表すことができる。

$$\begin{cases} \Pr[z_{2k} = 1] = \frac{1}{2} - \frac{1}{2}(1 - 2pp_1)^n \\ \Pr[z_{2k} = 0] = \frac{1}{2} + \frac{1}{2}(1 - 2pp_1)^n \end{cases} \quad (17)$$

そして、 $\Pr[z_{2k} = 1]$ を p_2 とする。

III: 次に、 $\mathbf{y} \cdot ((\mathbf{b}_1 \cdot \mathbf{r}_{u_1} + \dots + \mathbf{b}_m \cdot \mathbf{r}_{u_m}) + \mathbf{e}_u)$ の分布を調べる。ベクトル \mathbf{y} の各要素はパラメータ p , ベクトル \mathbf{e}_u の各要素はパラメータ p_e の独立したベルヌーイ変数とする。I, II と同様に進める。まず、 $(\mathbf{b}_1 \cdot \mathbf{r}_{u_1} + \dots + \mathbf{b}_m \cdot \mathbf{r}_{u_m}) + \mathbf{e}_u$ の分布を調べる。 $(\mathbf{b}_1 \cdot \mathbf{r}_{u_1} + \dots + \mathbf{b}_m \cdot \mathbf{r}_{u_m}) + \mathbf{e}_u$ を次のベクトル $\mathbf{z}_3 = (\mathbf{b}_1 \cdot \mathbf{r}_{u_1} + \dots + \mathbf{b}_m \cdot \mathbf{r}_{u_m}) + \mathbf{e}_u = (z_{31}, \dots, z_{3n})$ と表すと、次の式が求まる。

$$\begin{cases} \Pr[z_{3k} = 1] = p_e(1 - p_{r_2}) + p_{r_2}(1 - p_e) \\ \Pr[z_{3k} = 0] = p_e p_{r_2} + (1 - p_e)(1 - p_{r_2}) \end{cases} \quad (18)$$

そして、 $\Pr[z_{3k} = 1]$ を p_3 とする。次に、積 $\mathbf{y} \cdot ((\mathbf{b}_1 \cdot \mathbf{r}_{u_1} + \dots + \mathbf{b}_m \cdot \mathbf{r}_{u_m}) + \mathbf{e}_u)$ の分布を調べる。 $\mathbf{y} \cdot ((\mathbf{b}_1 \cdot \mathbf{r}_{u_1} + \dots + \mathbf{b}_m \cdot \mathbf{r}_{u_m}) + \mathbf{e}_u)$ を $\mathbf{z}_4 = \mathbf{y} \cdot ((\mathbf{b}_1 \cdot \mathbf{r}_{u_1} + \dots + \mathbf{b}_m \cdot \mathbf{r}_{u_m}) + \mathbf{e}_u) = (z_{41}, \dots, z_{4n})$ と表すと、次の式が求まる。

$$\begin{cases} \Pr[z_{4k} = 1] = \frac{1}{2} - \frac{1}{2}(1 - 2pp_3)^n \\ \Pr[z_{4k} = 0] = \frac{1}{2} + \frac{1}{2}(1 - 2pp_3)^n \end{cases} \quad (19)$$

そして、 $\Pr[z_{4k} = 1]$ を p_4 とする。

IV: 次に、差 $\mathbf{x} \cdot ((\mathbf{b}_1 \cdot \mathbf{r}_{A_1} + \dots + \mathbf{b}_m \cdot \mathbf{r}_{A_m}) + \mathbf{r}_B) - \mathbf{y} \cdot ((\mathbf{b}_1 \cdot \mathbf{r}_{u_1} + \dots + \mathbf{b}_m \cdot \mathbf{r}_{u_m}) + \mathbf{e}_u)$ の分布を調べる。 $\mathbf{x} \cdot ((\mathbf{b}_1 \cdot \mathbf{r}_{A_1} + \dots + \mathbf{b}_m \cdot \mathbf{r}_{A_m}) + \mathbf{r}_B)$ および $\mathbf{y} \cdot ((\mathbf{b}_1 \cdot \mathbf{r}_{u_1} + \dots + \mathbf{b}_m \cdot \mathbf{r}_{u_m}) + \mathbf{e}_u)$ の各要素はパラメータ p_2 および p_4 の独立したベルヌーイ変数である。 $\mathbf{x} \cdot ((\mathbf{b}_1 \cdot \mathbf{r}_{A_1} + \dots + \mathbf{b}_m \cdot \mathbf{r}_{A_m}) + \mathbf{r}_B) - \mathbf{y} \cdot ((\mathbf{b}_1 \cdot \mathbf{r}_{u_1} + \dots + \mathbf{b}_m \cdot \mathbf{r}_{u_m}) + \mathbf{e}_u)$ を次のベクトル $\mathbf{t} = \mathbf{x} \cdot ((\mathbf{b}_1 \cdot \mathbf{r}_{A_1} + \dots + \mathbf{b}_m \cdot \mathbf{r}_{A_m}) + \mathbf{r}_B) - \mathbf{y} \cdot ((\mathbf{b}_1 \cdot \mathbf{r}_{u_1} + \dots + \mathbf{b}_m \cdot \mathbf{r}_{u_m}) + \mathbf{e}_u) = (t_1, \dots, t_n)$ と表すと、以下の式が求まる。

$$\begin{cases} \Pr[t_k = 1] = p_2(1 - p_4) + (1 - p_2)p_4 \\ \Pr[t_k = 0] = p_2p_4 + (1 - p_2)(1 - p_4) \end{cases} \quad (20)$$

V: 次に、式(13)の誤りベクトルの分布を調べる。まず、 $((\mathbf{b}_1 \cdot \mathbf{r}_{v_1} + \dots + \mathbf{b}_m \cdot \mathbf{r}_{v_m}) + \mathbf{e}_v)$ の分布について調べる。 $((\mathbf{b}_1 \cdot \mathbf{r}_{v_1} + \dots + \mathbf{b}_m \cdot \mathbf{r}_{v_m}) + \mathbf{e}_v)$ の各要素のパラメータは $\mathbf{y} \cdot ((\mathbf{b}_1 \cdot \mathbf{r}_{u_1} + \dots + \mathbf{b}_m \cdot \mathbf{r}_{u_m}) + \mathbf{e}_u)$ と同じであるため、式(18)と同じ結果を得る。したがって、誤りベクトル e' を $e' = (e'_1, \dots, e'_n)$ と表すと、以下の式が求まる。

$$\begin{cases} \Pr[e'_k = 1] \\ = (1 - p_3)(p_2(1 - p_4) + (1 - p_2)p_4) \\ + p_3(p_2p_4 + (1 - p_2)(1 - p_4)) \\ \Pr[e'_k = 0] \\ = p_3(p_2p_4 + (1 - p_2)(1 - p_4)) \\ + (1 - p_3)((1 - p_2)p_4 + p_2(1 - p_4)) \end{cases} \quad (21)$$

そして、 $\Pr[e'_k = 1]$ を p_5 とする。

VI: V より、誤りベクトル e' がパラメータ p_5 の二項分布に従うことがわかった。よって、 $0 \leq d \leq \min(w_r(3w+1) + w_e(w+1), n)$ のとき、次の式が求まる。

$$\Pr[w(e') = d] = \binom{n}{d} (p_5)^d (1 - p_5)^{n-d} \quad (22)$$

6. 復号失敗確率の分析

前章では、誤りベクトルである式(13)の分布を決定する

ことができた。本章では、誤りベクトルの分布を求める式(22)を使用して、提案プロトコルの復号失敗確率の分析を行う。

テンソル積符号： C_1 を \mathbb{F} 上の $[n_1, k_1, d_1]$ 線形符号、 C_2 を \mathbb{F} 上の $[n_2, k_2, d_2]$ 線形符号とする。 $C_1 \otimes C_2$ と示された C_1 および C_2 のテンソル積符号は、行が C_1 の符号語で列が C_2 の符号語である $n_2 \times n_1$ として定義される。より正式に説明すると、 C_1 が \mathbf{G}_1 、 C_2 が \mathbf{G}_2 によって生成されたとき、次の式でテンソル積符号は定義できる。

$$C_1 \otimes C_2 = \{\mathbf{G}_2^\top \mathbf{X} \mathbf{G}_1 \text{ for } \mathbf{X} \in \mathbb{F}^{k_2 \times k_1}\} \quad (23)$$

上記の式で求められたテンソル積は $[n_1 n_2, k_1 k_2, d_1 d_2]$ 線形符号となる。

C_1 を符号長 n_1 、次元 k_1 、誤り訂正能力 δ_1 の BCH 符号、 C_2 を符号長 n_2 、次元 1、誤り訂正能力 $\delta_2 = \lfloor \frac{n_2-1}{2} \rfloor$ の繰り返し符号とする。この二つの符号を利用して提案プロトコルの復号失敗確率を強く保証するために、テンソル積符号 $C = C_1 \otimes C_2$ に限定して復号失敗確率を考える。

提案するプロトコルでの \mathbf{A} の入力を $\mu \in \mathbb{F}^{k_1}$ と考える。プロトコルでは、まず BCH 符号 (n_1, k_1, δ_1) により $\mu_1 \in \mathbb{F}^{n_1}$ に暗号化される。次に、繰り返し符号 $(n_2, 1, \delta_2)$ により μ_1 の各座標 $\mu_{1,i}$ ($i = 1, 2, \dots, n_1$) が $\mu'_{1,i}$ ($i = 1, 2, \dots, n_1$) に暗号化される。つまりベクトル $\mu' = (\mu'_{1,1}, \dots, \mu'_{1,n_1}) \in \mathbb{F}^{n_1 n_2}$ に暗号化される。より正式にしたものが次の式である。

$$\mathbb{1}_{n_2}.Decode(\mu'_{1,j}) = \begin{cases} 1 & \text{if } \sum_{i=0}^{n_2-1} \mu'_{1,j,i} \geq \lceil \frac{n_2+1}{2} \rceil, \\ 0 & \text{otherwise.} \end{cases} \quad (24)$$

復号失敗確率： BCH 符号と繰り返し符号を使用し復号失敗確率を求める。上記で定義されたテンソル積符号 $C = \text{BCH}(n_1, k_1, \delta) \otimes \mathbb{1}_{n_2}$ を使用すると、復号の失敗は、BCH 符号の復号アルゴリズムが、繰り返し符号による誤った復号後に発生したであろう誤りの訂正に成功しないときに起きる。したがって、復号失敗確率の分析は、3 つの段階に分けられる。繰り返し符号が正しく復号されない確率の評価、誤った重みが与えられた BCH 符号の復号失敗の条件付き確率、最後に、全確率の定理を使用することで復号失敗確率を求める。

I: まず、繰り返し符号の復号中に誤りが発生する確率を調べる。6 章で示すように、誤りベクトルである e' の座標が 1 となる確率は p_4 である。上記のように、 $\mathbb{1}_{n_2}$ は最大 $\delta_2 = \lfloor \frac{n_2-1}{2} \rfloor$ 個の誤りを訂正できる。したがって、誤りベクトル e' の重みを γ とすると、繰り返し符号 $\mathbb{1}_{n_2}$ の单一ブロックで復号誤りが発生する確率は、次の式で求まる。

$$p_\gamma = \sum_{i=\lfloor \frac{n_2-1}{2} \rfloor + 1}^{n_2} \binom{n_2}{i} \left(\frac{\gamma}{n_1 n_2} \right)^i \left(1 - \frac{\gamma}{n_1 n_2} \right)^{n_2-i} \quad (25)$$

II: BCH 符号 (n_1, k_1, δ_1) は、誤り訂正能力 δ_1 の範囲内の誤りを訂正できる。ここで、BCH 符号 (n_1, k_1, δ_1) 符号化された μ_1 から μ に正しく復号できない確率 \mathcal{P} は、繰り返し符号の少なくとも $\delta_1 + 1$ ブロックで誤りが起きた確率によって与えられる。したがって、確率 \mathcal{P} は次の式で求まる。

$$\mathcal{P} = \mathcal{P}(\delta_1, n_1, n_2, \gamma) = \sum_{i=\delta_1+1}^{n_1} \binom{n_1}{i} (p_\gamma)^i (1-p_\gamma)^{n_1-i} \quad (26)$$

III: 最後に、全確率の定理を使用して、復号の失敗確率を求める。復号失敗確率は、誤りが特定の重みを持つすべての確率の合計に、特定の重みの復号誤り確率 \mathcal{P} をかけたもので与えられる。したがって、復号失敗確率は次の式で求まる。

$$p_{\text{fail}} = \sum_{\gamma=0}^{\min(w_r(3w+1)+w_e(w+1), n_1 n_2)} \Pr[w(e') = \gamma] \times \mathcal{P} \quad (27)$$

文献 [2] に示されている BCH 符号 (n_1, k_1, δ_1) 、繰り返し符号 (n_2) 、重み w 、 w_r 、 w_e のパラメータを式(27)に代入して計算した結果が表 1 である。ただし、 w_r 、 w_e については式(13)の重みが HQC 暗号の $\mathbf{x} \cdot \mathbf{r}_2 - \mathbf{r}_1 \cdot \mathbf{y} + \mathbf{e}$ より 2 倍の重みになることから、 w_r 、 w_e の重みを HQC 暗号の $1/2$ に設定している。表 1 を見ると、Basic-II, Basic-III, Advanced-II, Advanced-III, Paranoiac-III, Paranoiac-IV のパラメータを使用したとき、復号失敗確率がセキュリティレベルの範囲内であることがわかる。したがって、表のパラメータを使うと復号失敗確率は無視できる確率以下になる。

7. 提案プロトコルの安全性

7.1 密鑑鍵の安全性

提案プロトコルでは公開鍵暗号である HQC 暗号を応用している。プロトコルで使用している公開鍵から秘密鍵が漏れてしまうと暗号化された入力が解読されてしまう。そのため、公開鍵から秘密鍵を求めることができないことを本節で証明する。

定理 1. 2-準巡回シンドローム復号決定仮定の下で、公開鍵 $\text{pk} = (\mathbf{h}, \mathbf{s} = \mathbf{x} + \mathbf{h} \cdot \mathbf{y})$ から秘密鍵 $\text{sk} = (\mathbf{x}, \mathbf{y})$ を求めることは困難である。

証明. 公開鍵 \mathbf{s} は次のように変形することができる。

表 1 復号失敗確率の計算結果 ($m = 3$)

Instance	n_1	n_2	$n \approx n_1 n_2$	k_1	δ_1	w	$w_r = w_e$	security	p_{fail}
Basic-I	766	29	22229	256	57	67	38	128	$< 2^{-99}$
Basic-II	766	31	23747	256	57	67	38	128	$< 2^{-146}$
Basic-III	796	31	24677	256	60	67	38	128	$< 2^{-174}$
Advanced-I	796	51	40597	256	60	101	58	192	$< 2^{-124}$
Advanced-II	766	57	43669	256	57	101	58	192	$< 2^{-196}$
Advanced- III	766	61	46747	256	57	101	58	192	$< 2^{-274}$
Paranoiac-I	766	77	59011	256	57	133	76	256	$< 2^{-114}$
Paranoiac-II	766	83	63587	256	57	133	76	256	$< 2^{-190}$
Paranoiac-III	796	85	67699	256	60	133	76	256	$< 2^{-267}$
Paranoiac- IV	796	89	70853	256	60	133	76	256	$< 2^{-335}$

$$\begin{aligned}
s &= \mathbf{x} + \mathbf{h} \cdot \mathbf{y} \\
&= \mathbf{x} + \mathbf{y} \cdot \text{rot}(\mathbf{h}) \\
&= [\mathbf{x} \ \mathbf{y}] [\mathbf{I}_n \ \text{rot}(\mathbf{h})]^\top
\end{aligned} \tag{28}$$

式(14)の $[\mathbf{I}_n \ \text{rot}(\mathbf{h})]$ は定義3で述べた組織的準巡回符号のパリティ検査行列の形になっている。すなわち、公開鍵 s は2-準巡回シンドローム復号決定仮定に帰着できるため、 $\mathbf{x} + \mathbf{h} \cdot \mathbf{y}$ と一様乱数の区別はできない。ゆえに、 (\mathbf{x}, \mathbf{y}) を求めるることはできない。したがって、公開鍵 pk から秘密鍵 sk を求めるることは困難である。□

7.2 提案プロトコルの安全性

二者間計算の安全性要件は二者の入力を相手方に漏らさず、任意の機能の計算をプロトコルで行い、計算結果のみが知られるというものである。本節では、提案プロトコルの相手方の参加者を敵対者とみなし、その相手方に対する安全性を証明する。

定理 2. 3-準巡回シンドローム復号決定仮定の下で、Aの入力 (a_1, a_2, \dots, a_m) は敵対者Bに漏れない。

証明. 提案プロトコルでは、AからBに $(\mathbf{u}_1, \dots, \mathbf{u}_m, \mathbf{v}_1, \dots, \mathbf{v}_m)$ を送る。式(8),(9)より $\mathbf{h} \cdot \mathbf{r}_{A_i} + \mathbf{r}_{u_i}, \mathbf{s} \cdot \mathbf{r}_{A_i} + \mathbf{r}_{v_i}$ ($i = 0, 1, \dots, m$)は次のように変形できる。

$$\begin{bmatrix} \mathbf{h} \cdot \mathbf{r}_{A_i} + \mathbf{r}_{u_i} \\ \mathbf{s} \cdot \mathbf{r}_{A_i} + \mathbf{r}_{v_i} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_n & 0 & \text{rot}(\mathbf{h}) \\ 0 & \mathbf{I}_n & \text{rot}(\mathbf{s}) \end{bmatrix} \begin{bmatrix} \mathbf{r}_{u_i} \\ \mathbf{r}_{v_i} \\ \mathbf{r}_{A_i} \end{bmatrix} \tag{29}$$

式(15)の行列 $\begin{bmatrix} \mathbf{I}_n & 0 & \text{rot}(\mathbf{h}) \\ 0 & \mathbf{I}_n & \text{rot}(\mathbf{s}) \end{bmatrix}$ は組織的準巡回符号のパリティ検査行列の形になっている。すなわち、 $(\mathbf{u}_1, \dots, \mathbf{u}_m, \mathbf{v}_1, \dots, \mathbf{v}_m)$ は3-準巡回シンドローム復号決定仮定に帰着できるため、敵対者Bは $\mathbf{h} \cdot \mathbf{r}_{A_i} + \mathbf{r}_{u_i}, \mathbf{s} \cdot \mathbf{r}_{A_i} + \mathbf{r}_{v_i}$ ($i = 0, 1, \dots, m$)と一様乱数の区別ができない。ゆえに、Aだけが知る $\mathbf{r}_{A_1}, \dots, \mathbf{r}_{A_m}, \mathbf{r}_{u_1}, \dots, \mathbf{r}_{u_m}, \mathbf{r}_{v_1}, \dots, \mathbf{r}_{v_m}$ を特定できないため、BはAの入力 (a_1, a_2, \dots, a_m) を求めることができない。したがって、Aの入力 (a_1, a_2, \dots, a_m) はBに

漏れない。□

定理 3. 3-準巡回シンドローム復号決定仮定の下で、Bの入力 (b_1, b_2, \dots, b_m) は敵対者Aに漏れない。

証明. 提案プロトコルでは、BからAに $(\mathbf{u}', \mathbf{v}')$ を送る。式(10), (11)より $\mathbf{h} \cdot \mathbf{r}_B + \mathbf{e}_u, \mathbf{s} \cdot \mathbf{r}_B + \mathbf{e}_v$ は次のように変形できる。

$$\begin{bmatrix} \mathbf{h} \cdot \mathbf{r}_B + \mathbf{e}_u \\ \mathbf{s} \cdot \mathbf{r}_B + \mathbf{e}_v \end{bmatrix} = \begin{bmatrix} \mathbf{I}_n & 0 & \text{rot}(\mathbf{h}) \\ 0 & \mathbf{I}_n & \text{rot}(\mathbf{s}) \end{bmatrix} \begin{bmatrix} \mathbf{e}_u \\ \mathbf{e}_v \\ \mathbf{r}_B \end{bmatrix} \tag{30}$$

式(16)の行列 $\begin{bmatrix} \mathbf{I}_n & 0 & \text{rot}(\mathbf{h}) \\ 0 & \mathbf{I}_n & \text{rot}(\mathbf{s}) \end{bmatrix}$ は組織的準巡回符号のパリティ検査行列の形になっている。すなわち、 $(\mathbf{u}', \mathbf{v}')$ は3-準巡回シンドローム復号決定仮定に帰着できるため、敵対者Aは $\mathbf{h} \cdot \mathbf{r}_B + \mathbf{e}_u, \mathbf{s} \cdot \mathbf{r}_B + \mathbf{e}_v$ と一様乱数の区別ができない。ゆえに、Bだけが知る $\mathbf{e}_u, \mathbf{e}_v, \mathbf{r}_B$ を特定できない。また、定理2より $(\mathbf{u}_1, \dots, \mathbf{u}_m, \mathbf{v}_1, \dots, \mathbf{v}_m)$ は一様乱数と区別がつかない。よって、 $\mathbf{h} \cdot \mathbf{r}_B + \mathbf{e}_u, \mathbf{s} \cdot \mathbf{r}_B + \mathbf{e}_v$ と $(\mathbf{u}_1, \dots, \mathbf{u}_m, \mathbf{v}_1, \dots, \mathbf{v}_m)$ は一様乱数の区別ができないため、 $(\mathbf{u}', \mathbf{v}')$ の分布も一様乱数に近づく。これより、Bは $(\mathbf{u}', \mathbf{v}')$ からAの入力 (b_1, b_2, \dots, b_m) を求めることができない。したがって、Bの入力 (b_1, b_2, \dots, b_m) は敵対者Aに漏れない。□

8. まとめ

本稿では、準巡回シンドローム復号問題に基づいた公開鍵暗号方式 HQC を応用した秘匿内積計算プロトコルの正当性と安全性について考察した。このプロトコルは符号ベースの暗号が耐量子性を持つという仮定のもと安全性を証明している。本稿で提案されたプロトコルは2次元および3次元ベクトルの \mathbb{F}_2 の入力による内積計算が可能である。今後の課題として、任意の長さのベクトルの内積が計算可能にするように拡張することがあげられる。

参考文献

- [1] HQC, <https://pqc-hqc.org/>.
- [2] C. Aguilar-Melchor, O. Blazy, J.-C. Deneuville, P. Gaborit, and Gilles Zémor, “Efficient encryption from random quasi-cyclic codes,” IEEE Trans. Inf. Theory, vol.64, no.5, pp.3927–4943, May 2018.
- [3] 邱儀穎, 河内亮周, 宮地充子, “準巡回シンドローム復号問題に基づく線形関数秘匿計算,” コンピュータセキュリティシンポジウム 2018 (CSS2018) 予稿集, pp.1237–1242, Oct. 2018.
- [4] 中山太雅, 廣友雅徳, 福田洋治, 毛利公美, 白石善明, “HQC 暗号を応用した秘匿内積計算プロトコル,” 信学技報, Sept. 2020.