Vol.2018-CSEC-82 No.61
Vol.2018-SPT-29 No.61
2018/7/26

2ビット毎に大小比較を行う暗号文出力型の秘匿比較方式の提案

小林 拓美[1,a)]　伯田 恵輔[2,b)]

概要：秘匿比較プロトコルは, 秘密情報を漏えいすることなく, 入力データである実数同士（整数であることが多い）の大小比較の結果を出力するプロトコルであり, マルチパーティプロトコルなどへの応用が可能であるため, 非常に有用である. 特に, Yao のミリオネア問題に基づいた比較プロトコルは, 入力された整数同士の大小比較の結果を出力するプロトコルであり, 多くの研究者によってさまざまなプロトコルが提案されてきた. しかしながら, これらのプロトコルはプロトコル参加者に比較結果が知られてしまうため, データ（比較結果を含む）を暗号化した状態でデータ処理を行う秘匿生体認証や秘匿統計処理などへの応用には適さない場合がある. 一方, 比較結果の暗号文を出力する秘匿比較プロトコルはプロトコル参加者に比較結果が知られないため, 秘匿生体認証などへの応用が期待できる. しかしながら, 多くの既存のプロトコルは比較結果を求めるために比較対象の 2 つの整数を 1 ビット毎に比較計算するため, 効率性に問題がある. 本稿では, 効率性向上を目的として複数ビット毎に比較計算を行う秘匿比較プロトコルを実現するための第一歩として, 比較結果の暗号文を出力する秘匿比較プロトコルを提案する. 提案プロトコルは 2 ビット毎に比較計算を行うことで比較結果を求めるプロトコルである. また, 提案プロトコルの正当性の証明, 計算コストの見積もり, semi-honest model における安全性についての簡易的な議論を行う.

キーワード：準同型暗号, 秘匿比較方式, プライバシー, マルチパーティプロトコル, Semi–honest model

Secure Comparison Protocol for Computing 2 bits-by-2 bits and Outputting Encrypted Comparison Result

Kobayashi Takumi[1,a)]　Hakuta Keisuke[2,b)]

1. はじめに

秘匿比較プロトコルは, 秘密情報を漏えいすることなく, 入力された実数同士（整数であることが多い）の比較結果を計算し, 出力するプロトコルである. このプロトコルは, マルチパーティプロトコルなどへの応用が可能であり, 非常に有用である.

Yao は 1982 年に秘匿比較プロトコルを提案し, Yao のミリオネア問題を提唱した [21]. Yao のミリオネア問題は次のような問題である: “Alice と Bob は, 秘密情報である正整数 a と b をそれぞれ持っている. Alice は比較結果 $(a \leq b)$ を知りたい. 秘密情報に関する情報を漏えいすることなく, どのようにすれば, Alice は比較結果を得ることができるか?”. 多くの研究者が Yao のミリオネア問題に基づいたより高効率・高安全なプロトコルの開発に取り組んでいる. Yao のミリオネア問題に基づいたプロトコルはさまざまな応用が可能であり, 非常に有用である. その例として, 秘匿オンラインオークション [5], [14], 秘匿生体認証 [1], 秘匿統計処理, プライバシー保護データマイニング [2],

[1] 島根大学大学院総合理工学研究科
Interdisciplinary Graduate School of Science and Engineering, Shimane University, 1060 Nishikawatsu-cho, Matsue, Shimane 690–8504, Japan
[2] 島根大学学術研究院理工学系
Institute of Science and Engineering, Academic Assembly, Shimane University, 1060 Nishikawatsu-cho, Matsue, Shimane 690–8504, Japan
[a)] s179506(at)matsu.shimane-u.ac.jp
[b)] hakuta(at)cis.shimane-u.ac.jp

[18] などが挙げられる. しかしながら, このプロトコルはプロトコル参加者に比較結果が知られてしまうため, データを暗号化した状態でデータ処理を行う応用技術に適さない場合がある. その例として, 秘匿生体認証 [8], [16] や秘匿統計処理 [9] などが挙げられる.

このような背景を受け, 比較結果の暗号文を出力する暗号プロトコル（以下, 本稿では秘匿比較プロトコルと呼ぶ）に注目が集まっている. この比較プロトコルは Alice の入力 a と Bob の入力 b に対する比較結果を暗号化し, その暗号文を出力するプロトコルである. 次節では, 秘匿比較方式における関連研究について言及する.

1.1 関連研究

さまざまな暗号構成要素を用いて, 暗号文を出力する秘匿比較プロトコルが複数の研究者によって提案されている. ここで, 記法 “ $(a \leq b)$ ” は非負整数 a, b に対する命題 “$a \leq b$” の真理値を示す. 例えば, もし $a \leq b$ が真であるならば, 比較結果 $(a \leq b)$ の暗号文は “1” の暗号文と等しい.

Damgård らは, 秘密分散を用いて暗号文を出力する秘匿比較プロトコルを提案した [4], [5], [6]. さらに, Garay らは, Conditional Circuit [17] を用いて比較結果 $(a > b)$ の暗号文を出力するプロトコルを提案した [12]. 近年では, 完全準同型暗号を用いた秘匿比較プロトコル [3] が Cheon らによって提案されている. このプロトコルは比較結果 $(a < b)$ の暗号文を出力するプロトコルである. 加えて, Veugen は, 準同型暗号を用いて比較結果 $(a > b)$ の暗号文を出力するプロトコルを提案した [19], Protocol 2.

このように, さまざまな暗号構成要素を用いて秘匿比較プロトコルが提案されてきた. しかしながら, これらのプロトコルの多くは 1 ビット毎に大小比較を行うことで, 比較結果を計算するプロトコルである. そのため, 多くのプロトコルはその効率性に問題がある. 本稿では, このようなプロトコルを BRC（Binary Representation Comparison）プロトコルと呼ぶ.

一方, 入力サイズ毎に大小比較を行うプロトコルも存在する. すなわち, l ビットの非負整数 a, b が与えられたとき, このプロトコルは入力サイズ毎に大小比較を行うプロトコルである. 本稿では, このようなプロトコルを IRC（Integer Representation Comparison）プロトコルと呼ぶ. IRC プロトコルは比較結果を計算するためにほんの 2, 3 ラウンドだけ必要である [20], Protocol 4. そのため, IRC プロトコルは BRC プロトコルよりも効率的になりうる. 一方, 複数の構成要素を用いた IRC プロトコルは提案されていない. 複数の構成要素を用いることで, 高安全な比較プロトコルの実現が期待できる. そのため, 安全性の観点において, BRC プロトコルは IRC プロトコルよりも優れているといえる.

そこで, 我々は w ビット毎（$w \geq 1$）に大小比較を行う秘匿比較プロトコルに着目する. すなわち, l ビットの非負整数 a, b が与えられたとき, このプロトコルは w ビット毎に大小比較を行うプロトコルである. w ビット毎に大小比較を行うことで, より効率的なプロトコルの開発が期待できる. さらに, 入力サイズ l に対して w ビット毎に分割して計算するため, 非常に大きい入力サイズに対してもその比較結果を計算できるという特徴がある.

本研究の目標は任意の整数 w に対して w ビット毎に大小比較を行う秘匿比較プロトコルを開発することである. 加えて, 高効率性と高安全性を両立するための最適な w について議論することも目標の 1 つである.

上述の目標の第一歩として, 比較結果の暗号文を出力する 2 つの秘匿比較プロトコルを提案する. 初めに提案するプロトコルは, Veugen のプロトコル [19], Protocol 2 を改良したものである. 以下, このプロトコルを提案プロトコル 1 と呼ぶ. 提案プロトコル 1 と Veugen のプロトコルは 1 ビット毎に大小比較を行うプロトコルである. さらに, 提案プロトコル 1 を改良することで, 2 ビット毎に大小比較を行うプロトコルを提案する. 以下, このプロトコルを提案プロトコル 2 と呼ぶ.

1.2 研究成果

本稿では, 次のようなシナリオを考える. “Alice と Bob は, それぞれ秘密情報である非負整数 a, b を持っている. 彼らは, 各整数の比較結果を明らかにすることなく, 安全に比較したい. このとき, Alice は Bob の公開鍵を用いて自分の持つ平文を暗号化する. 一方, Bob は自らの公開鍵とそれに対応する復号鍵を持っている. 結果として, Alice のみが整数 a, b の比較結果の暗号文を得る.” 言い換えると, “プロトコルにおいて, Alice と Bob は, 秘密情報である非負整数 a, b をそれぞれ入力する. 互いに秘密情報に関する情報を漏えいすることなく, Alice は, 入力された整数 a, b の比較結果の暗号文を得る.” 本稿の研究成果は以下のとおりである.

(1) 比較結果の暗号文を出力する暗号プロトコルを新たに提案する. 3 章において, このプロトコルを提案プロトコル 1 として記述する. 提案プロトコル 1 は Veugen のプロトコル [19], Protocol 2 を改良したものであり, Veugen のプロトコルよりもシンプルなアルゴリズムである. 入力された整数同士の比較結果を計算するために, 提案プロトコル 1 と Veugen のプロトコルは 1 ビット毎に大小比較を行うプロトコルである. さらに, Veugen のプロトコルは比較結果 $(a > b)$ の暗号文を出力するプロトコルである. 一方, 提案プロトコル 1 は比較結果 $(a > b)$ の暗号文を出力するプロトコルである. 提案プロトコル 1 は秘匿生体認証や秘匿統計処理などのような応用技術への利用が期待できる. 例えば, 秘匿除算プロトコル [20], Protocol 1 への応用が可能である.

(2) さらに, 比較結果の暗号文を出力する暗号プロトコルを新たに提案する. 4 章において, このプロトコルを提案プロトコル 2 として記述する. 提案プロトコル 1 や関連研究における多くの秘匿比較プロトコルは 1 ビット毎に大小比較を行うプロトコルである. 一方, 提案プロトコル 2 は 2 ビット毎に大小比較を行うプロトコルである. 2 ビット毎に大小比較を行うために, 提案プロトコル 2 はサブプロトコルとして秘匿除算プロトコル [20], Protocol 1 を利用する. さらに, 秘匿除算プロトコルはサブプロトコルとして秘匿比較プロトコルを利用する. しかし, この入力サイズは非常に小さいため, サブプロトコルである秘匿比較プロトコルは非常に効率的である. 詳細は 4 章 3 節における提案プロトコル 2 の計算コストを参照されたい. 提案プロトコル 2 は本研究の目標である一般化のための第一歩である.

2 章では, 記法や Paillier 暗号方式 [15] などの数学的準備を行う. Paillier 暗号方式は広く知られている準同型暗号方式である. 3 章では, 秘匿比較プロトコルである提案プロトコル 1 を提案する. これは Veugen のプロトコル [19], Protocol 2 を改良したものである. さらに, 正当性, 計算コストや安全性について簡易的に議論する. ここで, " 正当性" とは, プロトコルが入力に対して出力を計算するために適切に動作することを意味する. 4 章では, 秘匿比較プロトコルである提案プロトコル 2 を提案する. このプロトコルは 2 ビット毎に処理を行うプロトコルである. さらに, 正当性, 計算コストや安全性について簡易的に議論する. 5 章では, 本稿のまとめと今後の課題を述べる.

2. 数学的準備

まず, 本稿で用いる記法を以下に示す. $\mathbb{Z}$ で, 整数全体の集合を表す. 非負整数全体を $\mathbb{Z}_{\geq 0}$ で示す. 正整数 $n \geq 2$ に対し, 剰余環 $\mathbb{Z}/n\mathbb{Z}$ を $\mathbb{Z}_n$ で表す. p と q を $(k/2)$ ビットの素数とし, $N = p \times q$ を k ビットの RSA モジュラスとする. $\mathbb{Z}_n$ と同様に, 剰余環 $\mathbb{Z}/N\mathbb{Z}$ と $\mathbb{Z}/N^2\mathbb{Z}$ をそれぞれ $\mathbb{Z}_N$ と $\mathbb{Z}_{N^2}$ で表す. さらに, $a \in \mathbb{Z}_{\geq 0}$ に対し, 集合 $\{a \in \mathbb{Z}/N^2\mathbb{Z} \mid \gcd(a, N^2) = 1\}$ を $\mathbb{Z}^*_{N^2}$ と表す.

次に, 公開鍵暗号方式の定義は [11], Definition 7.1 を参照されたい. κ を暗号方式のセキュリティパラメータとする. 鍵生成アルゴリズムの出力である鍵ペア (pk, sk) は, 公開鍵 pk と秘密鍵 sk のペア (pk, sk) からなる. 公開鍵空間, 秘密鍵空間, 平文空間, 暗号文空間をそれぞれ $\mathcal{PK}$, $\mathcal{SK}$, $\mathcal{P}$, $\mathcal{C}$ と表す. $\mathrm{Enc}_{pk}(m)$ は公開鍵 pk を用いた平文 $m \in \mathcal{P}$ の暗号文を示す. さらに, $\mathrm{Dec}_{sk}(c)$ は暗号文 $c \in \mathcal{C}$ を, 秘密鍵 sk を用いて復号した結果を示す. 2 つの写像 Enc と Dec は次のように定義される: $\mathrm{Enc} : \mathcal{P} \times \mathcal{PK} \to \mathcal{C}, (m, pk) \mapsto c, \mathrm{Dec} : \mathcal{C} \times \mathcal{SK} \to \mathcal{P}, (c, sk) \mapsto m$.

2.1 準同型暗号

提案プロトコルでは, semi-honest model [13], Definition7.2.2 に対して semantically secure な加法準同型暗号を必要とする. 準同型暗号, および, 加法準同型暗号は, 次のように定義される.

定義 1. **(準同型暗号 [10], Definition 23.3.1)** 平文空間 $\mathcal{P}$ と暗号文空間 $\mathcal{C}$ をもつ公開鍵暗号方式は, 次のような条件を満たすとき, 群 $\mathcal{P}$ に対して, 準同型であるという. もし, 高速に計算可能な $\mathcal{C}$ におけるバイナリ操作 $\perp_1$ と, 同様な $\mathcal{P}$ におけるバイナリ操作 $\perp_2$ が存在すると仮定する. 全ての平文 $m_1, m_2 \in \mathcal{P}$ に対して, もし, 平文 m_1 の暗号文が c_1 であり, 平文 m_2 の暗号文が c_2 であるとき (ただし, どちらも同一の公開鍵で暗号化されている), 平文 $m_1 \perp_2 m_2$ の暗号文は, $c_1 \perp_1 c_2$ である.

もし, バイナリ操作 $\perp_2$ が加算 $+$ であるとき, この暗号方式を加法準同型暗号と呼ぶ. どのような加法準同型暗号であっても, 提案プロトコルに適用可能である. 例として, Paillier 暗号 [15], Damgard-Jurik 暗号 [7], DGK 暗号 (Damgård, Geisler and Krøigård) [5], [6] などが挙げられる. 特に, 本稿の各提案プロトコルでは, Paillier 暗号方式 [15] を利用している. 準同型性から, 平文 $m \in \mathcal{P}$ に対して, $-\mathrm{Enc}_{pk}(m^{-1}) = \mathrm{Enc}_{pk}(-m)$ である. さらに, 暗号文同士の乗算を "$\cdot$" と記述する.

2.2 Paillier 暗号方式

Paillier 暗号方式は, 広く知られている加法準同型暗号である. その 3 つのアルゴリズム (鍵生成, 暗号化, および, 復号アルゴリズム) を記述する. まず, $\lambda = \mathrm{lcm}(p-1, q-1)$ であり, $a \in \mathbb{Z}_{N^2}$ に対して, $a - 1 \equiv 0 \bmod N$ [15], Theorem 9 であることに注意されたい. また, 写像 L は次のように定義される: $\mathrm{L} : \mathbb{Z}_{N^2} \to \mathbb{Z}_N, a \mapsto (a-1)/N$.

2.2.1 鍵生成アルゴリズム

RSA モジュラス $N = p \times q$ を決める. このとき, p と q は, 大きな奇素数である. 次に, $g \in \mathbb{Z}^*_{N^2}$ を一様ランダムに選ぶ. $\gcd(\mathrm{L}(g^{\lambda} \bmod N^2), N) = 1$ かどうかを確認することで, g が $\mathbb{Z}^*_{N^2}$ の生成元であることをチェックする. もし, g が $\mathbb{Z}^*_{N^2}$ の生成元であれば, 公開鍵を (N, g), 秘密鍵を (p, q) とする.

2.2.2 暗号化アルゴリズム

平文 $m \in \mathbb{Z}_N$ が与えられたとき, 乱数 $r \in \mathbb{Z}^*_N$ を選び, 平文 m の暗号文を, $c = \mathrm{Enc}_{pk}(m, r) = g^m r^N \bmod N^2$ とする.

2.2.3 復号アルゴリズム

暗号文 $c \in \mathbb{Z}^*_{N^2}$ が与えられたとき, 式 (1) を計算する:

$$m = \frac{\mathrm{L}(c^{\lambda} \bmod N^2)}{\mathrm{L}(g^{\lambda} \bmod N^2)} \bmod N. \qquad (1)$$

$c = \mathrm{Enc}_{pk}(m, r)$ であれば, 明らかに式 (1) と m は等しいので, 元の平文 m が得られる. 詳細は [15] を参照されたい.

表 1　1 回の反復処理における変数の遷移

c_i	a_i	b_i	τ_i	tb_i after step 25	tb_i after step 29	t_{i+1}
0	0	0	$\tau_i \leftarrow t_i$	$tb_i \leftarrow 0$	$tb_i \leftarrow 0$	$t_{i+1} \leftarrow t_i + b_i - tb_i = t_i$
0	0	1	$\tau_i \leftarrow t_i$	$tb_i \leftarrow t_i$	$tb_i \leftarrow t_i$	$t_{i+1} \leftarrow t_i + b_i - tb_i = 1$
0	1	0	$\tau_i \leftarrow t_i$	$tb_i \leftarrow 0$	$tb_i \leftarrow 0$	$t_{i+1} \leftarrow tb_i = 0$
0	1	1	$\tau_i \leftarrow t_i$	$tb_i \leftarrow t_i$	$tb_i \leftarrow t_i$	$t_{i+1} \leftarrow tb_i = t_i$
1	0	0	$\tau_i \leftarrow 1 - t_i$	$tb_i \leftarrow 0$	$tb_i \leftarrow b_i - tb_i = 0$	$t_{i+1} \leftarrow t_i + b_i - tb_i = t_i$
1	0	1	$\tau_i \leftarrow 1 - t_i$	$tb_i \leftarrow 1 - t_i$	$tb_i \leftarrow b_i - tb_i = t_i$	$t_{i+1} \leftarrow t_i + b_i - tb_i = 1$
1	1	0	$\tau_i \leftarrow 1 - t_i$	$tb_i \leftarrow 0$	$tb_i \leftarrow b_i - tb_i = 0$	$t_{i+1} \leftarrow tb_i = 0$
1	1	1	$\tau_i \leftarrow 1 - t_i$	$tb_i \leftarrow 1 - t_i$	$tb_i \leftarrow b_i - tb_i = t_i$	$t_{i+1} \leftarrow tb_i = t_i$

また, 以下からわかるように, Paillier 暗号方式は加法準同型暗号である.

2.2.4 準同型性

2 つの平文 $m_1, m_2 \in \mathbb{Z}_N$ が与えられたとき, 乱数 $r_1, r_2 \in \mathbb{Z}_N^*$ を選ぶ. m_1 の暗号文を $c_1 = \mathrm{Enc}_{pk}(m_1, r_1) = g^{m_1} r_1^N \bmod N^2$, m_2 の暗号文を $c_2 = \mathrm{Enc}_{pk}(m_2, r_2) = g^{m_2} r_2^N \bmod N^2$ とする. このとき, 次式が成り立つ.

$$\begin{aligned} c_1 \cdot c_2 &= \mathrm{Enc}_{pk}(m_1, r_1) \cdot \mathrm{Enc}_{pk}(m_2, r_2) \\ &= (g^{m_1} r_1^N \bmod N^2) \times (g^{m_2} r_2^N \bmod N^2) \\ &= (g^{(m_1+m_2)}) \times (r_1 r_2)^N \bmod N^2 \\ &= \mathrm{Enc}_{pk}(m_1 + m_2, r_1 r_2). \end{aligned}$$

この性質から上述の暗号文同士の乗算により得られる暗号文と, それらの平文同士の加算結果を暗号化した暗号文が一致することがわかる.

3. 提案プロトコル 1

本章では, 非負整数 $a, b \in \mathbb{Z}_{\geq 0}$ に対する比較結果 $(a < b)$ の暗号文を出力するプロトコルを提案する. 紙数の都合上, 提案プロトコル 1 全体は記述せず, Veugen のプロトコルとの差分のみを次節に記述する.

3.1 提案プロトコル 1 の記述

提案プロトコル 1 は Veugen の秘匿比較プロトコル [19], Protocol 2 を改良したものである. 提案プロトコル 1 と Veugen のプロトコルは比較結果を計算するために, 1 ビット毎に大小比較を行うプロトコルである. さらに, Veugen のプロトコルは比較結果 $(a < b)$ の暗号文を出力するプロトコルである. 一方, 提案プロトコル 1 は比較結果 $(a > b)$ の暗号文を出力するプロトコルである.

提案プロトコル 1 と Veugen のプロトコルとの差分は, 初期化ステップ, ランダム化ステップ, および, ビット反転ステップである. 初期化ステップにおいて, Alice は平文 “1” を暗号化し, その暗号文 $\mathrm{Enc}_{pk}(1)$ を得る. さらに, $\mathrm{Enc}_{pk}(1)$ を $\mathrm{Enc}_{pk}(t_0)$ に代入する. 次に, Alice と Bob は $i = 0$ から $i = l - 1$ まで反復処理を行う. この初期化ステップは Veugen のプロトコルにおけるステップ 1–6 に対応している. ランダム化ステップにおいて, Bob は $r' \in \mathbb{Z}_N^*$ をランダムに選び, $(r')^N$ を計算する. さらに, $\mathrm{Enc}_{pk}(tb) \times (r')^N$ を計算することで, Bob は $\mathrm{Enc}_{pk}(tb)$ をランダム化する. このランダム化ステップを Veugen のプロトコルにおけるステップ 22 と 23 の間に追加する. 最後に, ビット反転ステップにおいて, Alice は $\mathrm{Enc}_{pk}(1) \cdot \mathrm{Enc}_{pk}(t_i)^{-1}$ を計算する. 次に, $\mathrm{Enc}_{pk}(1 - t_i)$ を $\mathrm{Enc}_{pk}(t_i)$ に代入し, $\mathrm{Enc}_{pk}(t_i)$ を出力する. このビット反転ステップを Veugen のプロトコルにおけるステップ 38 の直後に追加する.

3.2 正当性

提案プロトコル 1 における 1 回の反復処理における変数の遷移を**表 1** に示す. 表 1 から, 明らかに提案プロトコル 1 は正当性を満たす. 詳細は [19] を参照されたい.

3.3 計算コスト

ここで, 暗号化, 乗算, 逆算, および, べき乗算の回数をそれぞれ Enc, M, I, および, Exp と表す. 提案プロトコル 1 は Veugen のプロトコルよりも平均して 1 Enc, $0.5l + 3$ M, 2.5 I, および, $0.5l$ Exp を必要とする. また, Veugen のプロトコルは $(l - 1)$ 回の反復処理を必要とする一方, 提案プロトコル 1 は l 回の反復処理を必要とする. 入力サイズ l は十分に大きいと仮定すると, 提案プロトコル 1 と Veugen のプロトコルの計算コストはほぼ等しい. 最後に, 提案プロトコル 1 の Alice, Bob, 合計の計算コストは次のように見積もることができる: Alice の Enc, M, I, および, Exp は平均して, 1, $2l + 1$, $1.5l + 1$, および, 0 である. Bob の Enc, M, I, および, Exp は平均して, $1.5l$, $0.5l$, 0, および, $0.5l$ である. さらに, 合計の Enc, M, I, および, Exp は平均して, $1.5l + 1$, $2.5l + 1$, $1.5l + 1$, および, $0.5l$ である. 詳細は [19] を参照されたい.

3.4 安全性

本節では, 提案プロトコル 1 の安全性について簡易的に議論する. 提案プロトコル 1 の安全性は Veugen のプロトコル [19], Protocol 2 と同様である. 提案プロトコル 1 と Veugen のプロトコルとの差分は, 初期化ステップ, ランダム化ステップ, および, ビット反転ステップである. 初期化ステップでは, Alice も Bob も通信を行っていない. さ

らに, ランダム化ステップでは, Bob が b_i の値に応じて τ_i の暗号文をランダム化する. 最後に, ビット反転ステップでは, Alice が単にビット反転を行う. 結果として, 上述の 3 つの改良点は Veugen のプロトコルの安全性に影響を及ぼさない. さらに, Veugen のプロトコル [19], Protocol 2 は semi-honest model に対して安全であることが知られているため（[19], Section 2.3）, 提案プロトコル 1 も同様に semi-honest model に対して安全である.

4. 提案プロトコル 2

本章では, 比較結果 $(a > b)$ の暗号文を出力するプロトコルを提案する. 本稿では, 2 つの写像 $\mathcal{F}$ と $\hat{\mathcal{F}}$ を用いる. $\mathcal{F}$ と $\hat{\mathcal{F}}$ は入力された整数に対してその比較結果を返す写像であり, 次のように定義される.

$\mathcal{F} : \mathbb{Z}_{\geq 0}^2 \to \{0,1\}$ を次のように定義する.

$$\mathcal{F}(x,y) = \begin{cases} 0 & (x > y), \\ 1 & (x \leq y). \end{cases} \tag{2}$$

また, 任意の $x, y \in \mathbb{Z}_{\geq 0}$ に対し, $\hat{\mathcal{F}}(x,y) := \mathcal{F}(x,y) \oplus 1$ で定義する. ここで, "$\oplus$" は排他的論理和を意味する.

4.1 提案プロトコル 2 の記述

提案プロトコル 2 を Algorithm 1 に示す. まず, Algorithm 1 の概要について言及する. 提案プロトコル 2 は提案プロトコル 1 を改良したものである. 提案プロトコル 1 と Veugen のプロトコルは入力された整数同士の比較結果を計算するために, 1 ビット毎に大小比較を行うプロトコルである. 一方, 提案プロトコル 2 は 2 ビット毎に大小比較を行うプロトコルである.

次に, 提案プロトコル 2 を記述するために必要な数学的準備を行う. 本稿では, Alice は l ビットの非負整数である a, Bob は l ビットの非負整数である b を持つと仮定する. このとき, 非負整数 a, b を次のように表記する.

$$a = \textstyle\sum_{i=0}^{l-1} a_i 2^i = (a_{l-1}, \ldots, a_0)_2,\ a_i \in \{0,1\},$$

$$b = \textstyle\sum_{i=0}^{l-1} b_i 2^i = (b_{l-1}, \ldots, b_0)_2,\ b_i \in \{0,1\}.$$

また, 2 つの集合 Δ と集合 $\tilde{\Delta}$ を次のように表記し, σ を $\Delta = \tilde{\Delta} \backslash \{0\}$ の元とする.

$$\tilde{\Delta} := \{0,1,2,3\},\ \sigma \in \Delta := \{1,2,3\} = \tilde{\Delta} \backslash \{0\}.$$

さらに, 整数 a を次のように表記する.

$$a = (a_{l-1}, \ldots, a_0)_2 = (A_{L-1}, \ldots, A_0),$$

$$A_i := (a_{2i-1}, a_{2i})_2,\ A_i \in \{0,1,2,3\}.$$

また, 非負整数 b も同様に表記する. もし必要であれば, ゼロパディングを行うことによって, 提案プロトコル 2 にお

Algorithm 1 提案プロトコル 2

Input: Alice : $a = (a_{l-1}, \ldots, a_0)_2 = (A_{L-1}, \ldots, A_0)$, $d := 2^w = 2^2 = 4$ and pk
Input: Bob : $b = (b_{l-1}, \ldots, b_0)_2 = (B_{L-1}, \ldots, B_0)$, $d := 2^w = 2^2 = 4$, pk and sk
Output: Alice : $\mathrm{Enc}_{pk}(t_L)$ such that $t_L = \hat{\mathcal{F}}(a,b)$
Output: Bob : N/A

1: Alice encrypts and computes $\mathrm{Enc}_{pk}(1), \mathrm{Enc}_{pk}(d)$ and $\mathrm{Enc}_{pk}(1)^{-1}$
2: Alice computes : $t_0 = 1, \mathrm{Enc}_{pk}(t_0) \leftarrow \mathrm{Enc}_{pk}(1)$
3: **for** i from 0 to $L-1$ by $+1$ **do**
4: Alice chooses c_i such that $c_i \in \{0,1\}$ at random
5: **if** $c_i = 0$ **then**
6: Alice computes: $s_i = t_i, \mathrm{Enc}_{pk}(s_i) \leftarrow \mathrm{Enc}_{pk}(t_i)$
7: **else**
8: Alice computes: $s_i = 1 - t_i, \mathrm{Enc}_{pk}(s_i) \leftarrow \mathrm{Enc}_{pk}(1 - t_i)$
9: **end if**
10: Alice sends $\mathrm{Enc}_{pk}(s_i)$ to Bob
11: Bob encrypts B_i and obtains $\mathrm{Enc}_{pk}(B_i)$
12: **if** $B_i = 0$ **then**
13: Bob computes: $u_i = 0, \mathrm{Enc}_{pk}(u_i) \leftarrow \mathrm{Enc}_{pk}(0)$
14: **else**
15: Bob computes: $u_i = s_i + B_i, \mathrm{Enc}_{pk}(u_i) \leftarrow \mathrm{Enc}_{pk}(s_i + B_i)$
16: **end if**
17: Bob sends $\mathrm{Enc}_{pk}(B_i)$ and $\mathrm{Enc}_{pk}(u_i)$ to Alice
18: **if** $c_i \neq 0$ **then**
19: Alice computes: $\mathrm{Enc}_{pk}(u_i) \leftarrow \mathrm{Enc}_{pk}(2B_i - u_i + 1)$
20: **end if**
21: **if** $A_i = 0$ **then**
22: **if** $c_i = 0$ **then**
23: Alice computes: $\mathrm{Enc}_{pk}(B_i + d - 1)$
24: Alice and Bob perform a division protocol: Alice inputs $\mathrm{Enc}_{pk}(B_i + d - 1)$ and d, Bob inputs d and sk, Alice outputs $\mathrm{Enc}_{pk}(\lfloor (B_i + d - 1)/d \rfloor)$
25: Alice computes:
26: $\mathrm{Enc}_{pk}(t_{i+1}) \leftarrow \mathrm{Enc}_{pk}(t_i + B_i - u_i + \lfloor (B_i + d - 1)/d \rfloor)$
27: **else**
28: Alice computes: $\mathrm{Enc}_{pk}(B_i + d)$
29: Alice and Bob perform a division protocol: Alice inputs $\mathrm{Enc}_{pk}(B_i + d)$ and d, Bob inputs d and sk, Alice outputs $\mathrm{Enc}_{pk}(\lfloor (B_i + d)/d \rfloor)$
30: Alice computes:
31: $\mathrm{Enc}_{pk}(t_{i+1}) \leftarrow \mathrm{Enc}_{pk}(t_i + B_i - u_i + \lfloor (B_i + d)/d \rfloor)$
32: **end if**
33: **else**
34: Alice computes: $\mathrm{Enc}_{pk}(u_i - A_i - 1 + d)$
35: Alice and Bob perform a division protocol: Alice inputs $\mathrm{Enc}_{pk}(u_i - A_i - 1 + d)$ and d, Bob inputs d and sk, Alice outputs $\mathrm{Enc}_{pk}(\lfloor (u_i - A_i - 1 + d)/d \rfloor)$
36: **end if**
37: **end for**
38: Alice computes: $\mathrm{Enc}_{pk}(t_L) \leftarrow \mathrm{Enc}_{pk}(1 - t_L)$
39: Alice outputs $\mathrm{Enc}_{pk}(t_L)$

表 2　1 回の反復処理における変数の遷移

c_i	A_i	B_i	Condition	t_{i+1}
0	0	0		$t_{i+1} \leftarrow t_i + B_i - u_i + \lfloor (d + B_i - 1)/d \rfloor = t_i$
0	0	σ		$t_{i+1} \leftarrow t_i + B_i - u_i + \lfloor (d + B_i - 1)/d \rfloor = 1$
0	σ	0		$t_{i+1} \leftarrow \lfloor (u_i + A_i - 1 + d)/d \rfloor = \lfloor (d - (A_i + 1))/d \rfloor = 0$
0	σ	σ	$(A_i < B_i)$	$t_{i+1} \leftarrow \lfloor (u_i + A_i - 1 + d)/d \rfloor = \lfloor (d + t_i + (B_i - A_i - 1))/d \rfloor = 0$
0	σ	σ	$(A_i = B_i)$	$t_{i+1} \leftarrow \lfloor (u_i + A_i - 1 + d)/d \rfloor = \lfloor (d + t_i - 1)/d \rfloor = t_i$
0	σ	σ	$(A_i > B_i)$	$t_{i+1} \leftarrow \lfloor (u_i + A_i - 1 + d)/d \rfloor = \lfloor (d + t_i + (B_i - A_i - 1))/d \rfloor = 1$
1	0	0		$t_{i+1} \leftarrow t_i + B_i - u_i + \lfloor (d + B_i)/d \rfloor = t_i$
1	0	σ		$t_{i+1} \leftarrow t_i + B_i - u_i + \lfloor (d + B_i)/d \rfloor = 1$
1	σ	0		$t_{i+1} \leftarrow \lfloor (u_i + A_i - 1 + d)/d \rfloor = \lfloor (d - A_i)/d \rfloor = 0$
1	σ	σ	$(A_i < B_i)$	$t_{i+1} \leftarrow \lfloor (u_i + A_i - 1 + d)/d \rfloor = \lfloor (d + t_i + (B_i - A_i - 1))/d \rfloor = 0$
1	σ	σ	$(A_i = B_i)$	$t_{i+1} \leftarrow \lfloor (u_i + A_i - 1 + d)/d \rfloor = \lfloor (d + t_i - 1)/d \rfloor = t_i$
1	σ	σ	$(A_i > B_i)$	$t_{i+1} \leftarrow \lfloor (u_i + A_i - 1 + d)/d \rfloor = \lfloor (d + t_i + (B_i - A_i - 1))/d \rfloor = 1$

ける整数 l を偶数と仮定する. このとき, 明らかに $L = l/2$ である.

各 i $(1 \leq i \leq L)$ に対し, $A^{(i)} := (A_{i-1}, \ldots, A_0)$, $B^{(i)} := (B_{i-1}, \ldots, B_0)$ とおく. さらに, 各 i $(1 \leq i \leq L)$ に対し, $t_i := \mathcal{F}(A^{(i)}, B^{(i)})$ とおく. このとき, 次式が成り立つことに注意されたい.

$$t_L = \mathcal{F}(a^{(L)}, b^{(L)}) = \mathcal{F}(a, b).$$

提案プロトコル 2 において, 次の命題を用いることで, 各 i $(0 \leq i \leq L-1)$ に対して値 t_{i+1} を計算する.

$$A_i < B_i \Longrightarrow t_{i+1} = 1, \tag{3}$$

$$(t_i = 1) \wedge (A_i = B_i) \Longrightarrow t_{i+1} = 1. \tag{4}$$

A_i および B_i は, i 回目の反復処理における整数 $A^{(i)}$ および $B^{(i)}$ の最上位ビットであるため, 命題 (3) は容易に証明できる. i 回目の反復処理における最上位ビット A_i および B_i が等しい場合, 値 t_{i+1} は t_i に依存する. このことから, 命題 (4) も容易に証明できる.

提案プロトコル 2 の目的は, Alice と Bob の入力 a, b に対して, Alice が $t_L = 1 - t_L = 1 - \mathcal{F}(A^{(L)}, B^{(L)}) = 1 - \mathcal{F}(a, b) = \hat{\mathcal{F}}(a, b)$ の暗号文を出力することである. これは, $t_L = \mathcal{F}(A^{(L)}, B^{(L)}) = \mathcal{F}(a, b)$ および $\hat{\mathcal{F}}(a, b) = 1 - \mathcal{F}(a, b) = 1 - t_L$ を計算することで実現できる.

ここで, 提案プロトコル 2 における各ステップについて詳細に説明する. 提案プロトコル 2 のステップ 1–2 において, Alice は初期化を行い, $t_0 = 1$ とする. ステップ 3 において, Alice と Bob は $i = 0$ から $i = L - 1$ まで反復処理を行う. ステップ 4–10 において, Alice は t_i $(= \mathcal{F}(A^{(i)}, B^{(i)}))$ に対してブラインドを行い, その結果を Bob に送る. このブラインドのおかげで, 復号したとしても, Bob は t_i と $1 - t_i$ を区別することはできない. ステップ 11–17 において, B_i に応じて Bob は $\mathrm{Enc}_{pk}(B_i)$ と $\mathrm{Enc}_{pk}(u_i)$ を計算し, その結果を Alice に送る. ステップ 18–20 において, Alice はそのブラインドを解く. ステップ 21–36 において, A_i と c_i に応じて, Alice は $\mathrm{Enc}_{pk}(t_{i+1})$ を計算する. ステップ 37 において, 反復処理を終えたとき, Alice は $\mathcal{F}(a, b)$ の暗号文を得る. これは, 式 (5) が成り立つためである.

$$t_L = \mathcal{F}(a, b). \tag{5}$$

ステップ 38–39 において, ビット反転を行うことで, 最終的に Alice は $\hat{\mathcal{F}}(a, b)$ の暗号文を得る. また, ステップ 8, 15, 19, 23, 26, 28, 31, 34, および, 38 では, 暗号方式の準同型性を利用することにより, 各暗号文を計算することができる.

さらに, 提案プロトロコル 2 はサブプロトコルとして整数除算プロトコル [20], Protocol 1 を利用する. この除算プロトコルにおいて, Alice は整数 x の暗号文と除数 d を入力し, Bob は除数 d と sk を入力する. 最後に, Alice は出力として除算結果 $\lfloor x/d \rfloor$ の暗号文を得る. Algorithm 1 のステップ 24, 29, 35 において, 除算プロトコルを実行している. 一方, Alice は除算結果の暗号文を保持することができるため, Algorithm 1 のステップ 25–26 および 30–31 において, 除算プロトコルを実行する必要はない.

加えて, この除算プロトコルはサブプロトコルとして比較プロトコルを利用する. この比較プロトコルにおいて, Alice と Bob はそれぞれサイズが $\lfloor \log_2 d \rfloor$ である整数 a, b を入力する. その結果として, Alice は比較結果 $(a > b)$ の暗号文を出力する. 提案プロトコル 2 では, 除算プロトコルにおけるサブプロトコルとして提案プロトコル 1 を利用する.

4.2　正当性

本節では, 提案プロトコル 2 の正当性を示す. 表 1 と同様に, 提案プロトコル 2 における 1 回の反復処理における変数の遷移を**表 2** に示す. 言い換えると, 表 2 は, 提案プロトコル 2 において, 値 t_{i+1} が c_i, A_i および B_i に応じてどのように計算されるかを示す. このとき, 表 2 において, $\sigma \in \Delta = \{1, 2, 3\} = \tilde{\Delta} \backslash \{0\}$ である. 提案プロトコル 2 の正当性とは, Alice の入力である a, Bob の入力である b に対して, もし $a > b$ であれば Alice は $t_L = 1$ を出力し, それ以外であれば Alice は $t_L = 0$ を出力することである. 本証

表 3　サブプロトコルを除いた提案プロトコル 2 の計算コスト

c_i	a_i	b_i	Alice				Bob				Total			
			Enc	M	I	*Exp*	*Enc*	M	I	*Exp*	*Enc*	M	I	*Exp*
0	0	0	2	$5L+1$	$L+2$	0	$2L$	0	0	0	$2L+2$	$5L+1$	$L+2$	0
0	0	σ	2	$5L+1$	$L+2$	0	$2L$	0	0	0	$2L+2$	$5L+1$	$L+2$	0
0	σ	0	2	$3L+1$	$L+2$	0	$2L$	0	0	0	$2L+2$	$3L+1$	$L+2$	0
0	σ	σ	2	$3L+1$	$L+2$	0	$2L$	0	0	0	$2L+2$	$3L+1$	$L+2$	0
σ	0	0	2	$8L+1$	$2L+2$	0	L	L	0	0	$L+2$	$9L+1$	$2L+2$	0
σ	0	σ	2	$8L+1$	$2L+2$	0	L	L	0	0	$L+2$	$9L+1$	$2L+2$	0
σ	σ	0	2	$7L+1$	$3L+2$	0	L	L	0	0	$L+2$	$8L+1$	$3L+2$	0
σ	σ	σ	2	$7L+1$	$3L+2$	0	L	L	0	0	$L+2$	$8L+1$	$3L+2$	0
AVE			2	$5.375L+1$	$1.875L+2$	0	$1.5L$	$0.5L$	0	0	$1.5L+2$	$5.875L+1$	$1.875L+2$	0

明の主な考えは, L 回目の反復処理の後, 式 (5) が成り立つことを証明することである. これを定理 1 に示す. 定理 1 より, ステップ 38–39 において $\mathrm{Enc}_{pk}(1)\cdot\mathrm{Enc}_{pk}(t_L)^{-1}$ を計算することで, Alice は $\hat{\mathcal{F}}(a,b)$ の暗号文を得ることがわかる.

定理 1. Algorithm 1 において, L 回目の反復処理の後, 式 (5) が成り立つ.

Proof.　表 2 から従う. □

4.3　計算コスト

初めに, サブプロトコルの計算コストを見積もる. サブプロトコルとして, 秘匿除算プロトコル [20], Protocol 1 を利用する. さらに, この除算プロトコルは, サブプロトコルとして秘匿比較プロトコルを利用する. Veugen は, “秘匿比較プロトコルの実行 1 回を除いて, 除算プロトコルは Alice による 2 *Enc*, 3 M と 1 I, Bob による 1 *Enc* と 1 *Dec* が必要である.” と主張している. また, Paillier 暗号方式においては事前計算を行うことで, 復号とべき乗算の計算コストがほぼ等しいと仮定できる [15].

さらに, 除算プロトコルにおける秘匿比較プロトコルとして, 提案プロトコル 1 を利用する. この秘匿比較プロトコルの入力サイズは $\lfloor\log_2 d\rfloor = \lfloor\log_2 2^w\rfloor = w = 2$ であるため, このプロトコルは非常に効率的である. 比較プロトコルの実行 1 回に対して, 平均して Alice による 1 *Enc*, 5 M と 4 I および Bob による 3 *Enc*, 1 M, 4 I と *Exp* が必要である. 結果として, サブプロトコルである比較プロトコルを含む除算プロトコルの実行 L 回に対して, その計算コストは次のように見積もることができる: Alice の *Enc*, M, I, および, *Exp* は $3L$, $8L$, $5L$, および, 0 である. Bob の *Enc*, M, I, および, *Exp* は $4L$, L, 0, および, $2L$ である. 合計の *Enc*, M, I, および, *Exp* は $7L$, $9L$, $5L$, および, $2L$ である.

サブプロトコルを除いた提案プロトコル 2 の計算コストを**表 3** に示す. 表 3 における各行は, i に対する値 c_i, A_i 及び B_i が 0 あるいは σ であるときの Alice, Bob, および, 合計の計算コストを意味する. さらに, 表 3 における最後の行は, Alice, Bob, および, 合計の計算コストの平均値を意味する. このとき, Alice と Bob はそれぞれ各ステップにおける暗号文を保持しているという仮定に基づいて, 計算コストの見積もりを行う.

最後に, サブプロトコルを含む提案プロトコル 2 の計算コストは次のように見積もることができる: Alice の *Enc*, M, I, および, *Exp* は $3L+2$, $13.375L+1$, $6.875L+2$, および, 0 である. Bob の *Enc*, M, I, および, *Exp* は $5.5L$, $1.5L$, 0, および, $2L$ である. 合計の *Enc*, M, I, および, *Exp* は $8.5L+2$, $14.875L+1$, $6.875L+2$, および, $2L$ である.

提案プロトコル 1 の計算コストと比較して, 提案プロトコル 2 の計算コストはより大きい. 1 章で述べたように, 提案プロトコルの一般化は本研究の目標である. 本稿では, その第一歩として, 2 ビット毎に大小比較を行う比較プロトコルを提案した. 一般化が実現するならば, より効率的な秘匿比較プロトコルの開発が期待できる.

4.4　安全性

本節では, 提案プロトコル 2 の安全性について議論する. 3 章 4 節と同様に, 提案プロトコル 2 の安全性は Veugen のプロトコルと同様である. Veugen のプロトコル [19], Protocol 2 は semi-honest model に対して安全であることが知られているため（[19], Section 2.3）, 提案プロトコル 2 も同様に semi-honest model に対して安全である.

次に, Alice と Bob の各ステップを確認することで, 平文に対するいかなる情報も漏えいしていないことを確認する. 提案プロトコル 2 のステップ 4–10 において, Alice はコイントス c_i により $\mathrm{Enc}_{pk}(t_i)$ をブラインドし. $\mathrm{Enc}_{pk}(s_i)$ を Bob に送る. このブラインドのおかげで, 暗号文 $\mathrm{Enc}_{pk}(s_i)$ を復元したとしても, Bob は平文 s_i が t_i あるいは $1-t_i$ であるかどうか区別することはできない. 次に, ステップ 11–17 において, Bob は B_i に応じて $\mathrm{Enc}_{pk}(u_i)$ を計算する. さらに, Bob は $\mathrm{Enc}_{pk}(B_i)$ と $\mathrm{Enc}_{pk}(u_i)$ を Alice に送る. このとき, 暗号方式は semantically secure であるため, Alice はこれらの暗号文から平文に関するいかなる情報も得ることはできない. したがって, Alice と Bob は互いに相手の平文に関するいかなる情報も得ることはできない.

5. まとめ

非負整数 a, b に対する比較結果 $(a > b)$ の暗号文を出力する 2 つのプロトコルを提案した．提案プロトコル 1 は Veugen のプロトコル [19], Protocol 2 を改良したプロトコルであり, Veugen のプロトコルよりもシンプルなアルゴリズムである．提案プロトコル 1 は秘匿生体認証や秘匿統計処理などのような応用技術への利用が期待できる．提案プロトコル 2 は提案プロトコル 1 を改良したプロトコルであり, 比較結果を計算するために 2 ビット毎に大小比較を行うプロトコルである．提案プロトコル 2 の提案は提案プロトコル 1 の一般化のための第一歩である．各提案プロトコルは semi-honest model に対して安全である．さらに, 各提案プロトコルの正当性の証明および計算コストの見積もりを行った．今後の課題として, 各提案プロトコルと他の類似の秘匿比較プロトコルとの性能比較や各提案プロトコルの厳密な安全性証明などが挙げられる．

参考文献

[1] Blanton, M., Gasti, P.: Secure and efficient protocols for iris and fingerprint identification. In: Atluri, V., Diaz, C. (eds.) European Symposium on Research in Computer Security–ESORICS 2011. Lecture Notes in Computer Science, vol. 6879, pp. 190–209. Springer, Heidelberg (2011).

[2] Bost, R., Popa, R.A., Tu, S., Goldwasser, S.: Machine learning classification over encrypted data. In: Network and Distributed System Security Symposium–NDSS 2015.

[3] Cheon, J.H., Kim, M., Kim, M.: Optimized search–and–compute circuits and their application to query evaluation on encrypted data. IEEE Trans. Inf. Forensics Secur. **11**(1), 188199 (2016). IEEE Press, New York

[4] Damgård, I., Fitzi, M., Kiltz, E., Nielsen, J.B., Toft, T.: Unconditionally Secure Constant-Rounds Multiparty Computation for Equality, Comparison, Bits and Exponentiation. In: Halevi S., Rabin T. (eds.) Theory of Cryptography Conference–TCC 2006. Lecture Notes in Computer Science, vol. 3876, pp. 285–304, Springer, Heidelberg (2006).

[5] Damgård, I., Geisler, M., Krøigård, M.: Homomorphic encryption and secure comparison. Int. J. Appl. Crypt. **1** (1), 22–31 (2008).

[6] Damgård, I., Geisler, M., Krøigård, M.: A correction to efficient and secure comparison for on-line auctions. Int. J. Appl. Crypt. **1** (4), 323–324 (2009).

[7] Damgård, I., Jurik, M.: A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System. In: Kim K. (ed.) Public-Key Cryptography–PKC 2001. Lecture Notes in Computer Science, vol. 1992, pp. 119–136. Springer, Verlag (2001).

[8] Erkin, Z., Franz, M., Katzenbeisser, S., Guajardo, J., Lagendijk, R. L., Toft, T.: Privacy-preserving face recognition. In: Goldberg I., Atallah M.J. (eds.) Privacy Enhancing Technologies Symposium–PETS 2009. Lecture Notes in Computer Science, vol. 5672, pp. 235–253. Springer, Heidelberg (2009).

[9] Erkin, Z., Veugen, T., Toft, T., Lagendijk, R.L.: Privacy-preserving user clustering in a social network. In: Workshop on Information Forensics and Security–WIFS 2009, pp. 96–100. IEEE (2009).

[10] Galbraith, S. D.: Mathematics of Public Key Cryptography. Cambridge University Press, Cambridge (2012) .

[11] Goldwasser, S., Bellare, M.: Lecture notes on cryptography. Summer course on cryptography, pp. 119–120. Massachusetts Institute of Technology, 1996–2008 (2008). http://cseweb.ucsd.edu/ mihir/papers/gb.html

[12] Garay, J., Schoenmakers, B., Villegas, J.: Practical and Secure Solutions for Integer Comparison. In: Okamoto T., Wang X. (eds.) Public-Key Cryptography–PKC 2007. Lecture Notes in Computer Science, vol. 4450, pp. 330–342. Springer, Heidelberg (2007).

[13] Goldreich, O.: Foundations of Cryptography, vol. 2. Cambridge University Press, Cambridge (2001).

[14] Kolesnikov, V., Sadeghi, A-R. and Schneider, T. Improved Garbled Circuit Building Blocks and Applications to Auctions and Computing Minima. In: Garay J.A., Miyaji A., Otsuka A. (eds.) Cryptology and Network Security–CANS 2009. Lecture Notes in Computer Science, vol. 5888, pp. 1–20. Springer, Heidelberg (2009).

[15] Paillier, P.: Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In: Stern J. (ed.) Advances in Cryptology–EUROCRYPT 1999. Lecture Notes in Computer Science, vol. 1592, pp. 223–238. Springer, Heidelberg (1999).

[16] Sadeghi, A.-R., Schneider, T., Wehrenberg, I.: Efficient privacy-preserving face recognition. In: Lee, D., Hong, S. (eds.) International Conference on Information and Communications Security–ICISC 2009. Lecture Notes in Computer Science, vol. 5984, pp. 229–244. Springer, Heidelberg (2010).

[17] Schoenmakers, B., Tuyls, P.: Practical two-party computation based on the conditional gate. In: Lee, P.J. (ed.) Advances in Cryptology–ASIACRYPT 2004. Lecture Notes in Computer Science, vol. 3329, pp. 119–136. Springer, Heidelberg (2004).

[18] Vaidya, J., Clifton, C., Zhu, Y.: Privacy Preserving Data Mining. Springer, New York (2006).

[19] Veugen, T.: Comparing encrypted data. In: Technical Report, Multimedia Signal Processing Group, Delft University of Technology, The Netherlands, and TNO Information and Communication Technology, Delft, The Netherlands (2011).

[20] Veugen, T.: Encrypted integer division and secure comparison. Int. J. Appl. Crypt. **3** (2), 166–180 (2014).

[21] Yao, A. C. C.: Protocols for secure computations. In: Proceedings of the 23rd Annual Symposium on Foundations of Computer Science–FOCS 1982, pp. 160–164 (1982).