情報処理学会論文誌 Vol.54 No.9 2137–2145 (Sep. 2013)

加法準同型ElGamal暗号を用いたビット分解プロトコル

千田 浩司[1,a)] 五十嵐 大[1] 高橋 克巳[1]

受付日 2012年12月3日, 採録日 2013年6月14日

概要：セキュアマルチパーティ計算（MPC: Secure Multi-Party Computation）は Privacy Preserving Data Mining（PPDM）の主要技術の 1 つとして近年注目されているが，通常の計算と比べ膨大な処理をともなうことが実用の障壁となっている．MPC の処理を軽減させる手法の 1 つに，数値からビット列への変換を暗号化したまま行う「ビット分解プロトコル」がある．Schoenmakers と Tuyls は Eurocrypt 2006 で Paillier 暗号を用いたビット分解プロトコルを提案しているが，MPC の高速処理に適した暗号として期待される加法準同型 ElGamal 暗号を用いた方式は我々が知る限り実現できていない．本論文では，従来のビット分解プロトコルでは加法準同型 ElGamal 暗号への適用が困難であることを述べ，従来とはまったく異なるアプローチにより，*semi-honest* モデルにおける，加法準同型 ElGamal 暗号を用いたビット分解プロトコルをいくつか構成する．特に事前処理を許す 2 パーティ限定の提案方式は，従来と比べ大きく処理削減できることを示す．またビット分解プロトコルが特に有効となるいくつかの具体的な MPC の応用方式を提案する．

キーワード：ビット分解プロトコル，加法準同型 ElGamal 暗号，セキュアマルチパーティ計算，Paillier 暗号

Bit-decomposition Protocol Using the Additive Homomorphic ElGamal Encryption

Koji Chida[1,a)] Dai Ikarashi[1] Katsumi Takahashi[1]

Received: December 3, 2012, Accepted: June 14, 2013

Abstract: Secure Multi-Party Computation (MPC) has attracted attention recently as a core technology for the privacy preserving data mining (PPDM), however, MPC remains a tremendous challenge to reduce the computation and communication complexities from a practical view point. As a potential solution to the challenge, there exists a cryptographic protocol called *bit-decomposition protocol* that converts an encrypted integer into each encrypted bit of the integer without leaking the integer. In this paper, we propose some bit-decomposition protocols in the *semi-honest* model for the ElGamal encryption with additive homomorphism, which is expected as a building block for efficient MPC, unlike that for the Paillier encryption proposed by Schoenmakers and Tuyls at Eurocrypt 2006. In particular, the proposed two-party bit-decomposition protocol with pre-computation is much more efficient than the Paillier-based method. We also propose some efficient MPC applications using the bit-decomposition protocol.

Keywords: bit-decomposition protocol, additive homomorphic ElGamal encryption, secure multi-party computation, Paillier encryption

1. はじめに

セキュア計算 [1] は，数値等のデータを秘匿しつつ各種計算を可能とする暗号応用技術として知られる．近年ではデータマイニング等においてデータの保護と利活用の両立を目指す Privacy Preserving Data Mining（PPDM）[2], [3] の研究の進展が目覚ましく，セキュア計算も PPDM の主要技術として注目を集めている．

一般にセキュア計算は，本来の計算に必要な入力データを秘匿処理して後述の計算主体に提供する主体（提供主体）

[1] 日本電信電話株式会社 NTT セキュアプラットフォーム研究所
NTT Secure Platform Laboratories, NTT Corporation, Musashino, Tokyo 180–8585, Japan
[a)] chida.koji@lab.ntt.co.jp

と，入力データの秘匿性を損ねずに計算を行う主体（計算主体）の少なくとも 2 者が存在する．用途や技術的な制約によって，提供主体や計算主体を複数としたり，一部の提供主体および計算主体を同一主体とする場合もある．特に計算主体を複数として一定数以上の計算主体が不正に結託しないことを前提とするセキュア計算は**セキュアマルチパーティ計算**（MPC: Secure Multi-Party Computation）とも呼ばれる（2 者に特化した方式は区別してセキュア 2 パーティ計算と呼ぶ場合もある）．秘匿処理は公開鍵暗号や秘密分散法等が用いられる．なお本論文では秘匿処理を総称して暗号化，そして秘匿処理されたデータを暗号文と呼ぶ．

外部への開示が望ましくない機密性/機微性の高いデータについて，クラウド等の外部の主体にデータ処理させる場合や，複数の主体が保有するデータを統合的に扱う場合に，セキュア計算は有効な手段と考えられる．特に後者は，2 以上のすべての主体が提供主体と計算主体を兼ね，各提供主体のデータの秘匿性を損ねずにデータを統合的に扱い分析結果等を出力する，文献 [3] をはじめ PPDM の代表的なモデルとして知られる．

しかしながらセキュア計算は通常の計算と比べ膨大な処理をともなうことが実用の障壁となっており，MPC も例外ではない．そのため MPC の処理を軽減させる様々な手法が研究されており，有望な手法の 1 つとして，数値からビット列への変換を暗号化したまま行う**ビット分解プロトコル**が提案されている．ビット分解プロトコルは，秘密分散法を用いた方式 [4], [5], [6] と，Paillier 暗号を用いた方式 [7] が知られている．しかし MPC の高速処理に適した暗号として期待される加法準同型 ElGamal 暗号を用いた方式は，我々が知る限り実現できていない．

本論文では，従来のビット分解プロトコルでは加法準同型 ElGamal 暗号への適用が困難であることを述べ，従来とはまったく異なるアプローチにより，*semi-honest* モデルにおける，加法準同型 ElGamal 暗号を用いたビット分解プロトコルをいくつか構成する．特に事前処理を許す 2 者限定の提案方式は，従来と比べ大きく処理削減できることを示す．またビット分解プロトコルが特に有効となるいくつかの具体的なセキュアマルチパーティ計算の応用方式を提案する．

2. 関連研究

Yao [8] は組合せ論理回路を計算可能とするセキュア 2 パーティ計算を実現し，Goldreich ら [9] は MPC に拡張した．Ben-Or ら [10] と Chaum ら [11] は独立に，秘密分散法を用いて 3 以上の計算主体について有限体上の四則演算を可能とする MPC を構成した．ただし文献 [10], [11] では計算主体の数を n としたとき，$n/2$ 以上の計算主体が不正に結託しないことを前提としている．

Cramer ら [12] は準同型暗号を用いて加減乗算を可能とする MPC を構成し，秘密分散法を用いた方式 [10], [11] では計算量および通信量が $O(n^3)$ となるのに対し，ともに $O(n^2)$ とした．文献 [12] では Paillier 暗号を用いて環 $\mathbb{Z}/N\mathbb{Z}$（N は RSA 合成数）上の加減乗算を可能とする MPC の具体例が与えられている．Schoenmakers ら [13] は加法準同型性を持つ ElGamal 暗号を用いて $\mathbb{Z}/q\mathbb{Z}$（q は大きな素数）上の加減乗算を可能とする MPC を構成した．ただし乗算は片方の乗数が 1 ビットに制限される．ElGamal 暗号や加法準同型 ElGamal 暗号は Paillier 暗号と異なり楕円曲線暗号に拡張した高速実装が可能であり，Paillier 暗号と比べおよそ 200 倍高速との報告がある [14]．なお文献 [12] の方式では加法準同型 ElGamal 暗号を用いた MPC は実現できていない．

加減乗算可能な MPC は，入力データをビット列として扱いビットごとに暗号化すれば組合せ論理回路が計算可能となることはよく知られる．しかし一般の数値を暗号化すると，組合せ論理回路の計算方法は自明でない．Algesheimer ら [4] は秘密分散法を用いた方式 [10], [11] に適用可能なビット分解プロトコルを実現することでこの問題を解決した．文献 [4] の方式は各主体が正しく処理を行うことを前提とする *semi-honest* モデルだが，Damgård ら [5] はその制限を不要とした *malicious* モデルを構成した．また Schoenmakers ら [7] は，Paillier 暗号を用いた方式 [12] に適用可能なビット分解プロトコルを構成した．しかし 1 章で述べたとおり，加法準同型 ElGamal 暗号を用いた方式 [13] に適用可能なビット分解プロトコルは，我々が知る限り実現できていない．

3. 準備

次章以降でビット分解プロトコルの詳細な説明を行うために必要となる前提知識，用語，および記号について説明する．

3.1 加法準同型 ElGamal 暗号

q を大きな素数，g を位数 q の巡回群 $\mathcal{G}$ の生成元，$\mathcal{M}$ を $\mathbb{Z}/q\mathbb{Z}$ の部分集合とし，秘密鍵を $x \in \mathbb{Z}/q\mathbb{Z}$，公開鍵を $(q, \mathcal{G}, \mathcal{M}, g, y := g^x)$ とする．$\mathbb{Z}/q\mathbb{Z}$ から選んだ乱数，すなわち $r \in_R \mathbb{Z}/q\mathbb{Z}$ から，$a \in \mathcal{M}$ の暗号文を $E(a) := (g^r, y^{a+r}) := (G, Y)$ とする．復号は，$y^a = Y/G^x$ を求め（ステップ 1），y^a から a を求めれば（ステップ 2）よいが，ステップ 2 は離散対数を解く問題であり一般には計算困難であるため，実際には $\mathcal{M}$ の要素数を q よりも十分小さくする必要がある．

暗号文 $E(a)$ および $E(b) := (g^s, y^{b+s})$ の要素ごとの乗算および除算はそれぞれ

$$E(a)E(b) = (g^{r+s}, y^{(a+b)+(r+s)}) = E(a+b),$$
$$E(a)/E(b) = (g^{r-s}, y^{(a-b)+(r-s)}) = E(a-b)$$

となるため，$E(a)$，$E(b)$ から $E(a \pm b)$ が求まる．文献 [13] の MPC を実行すれば，b を 1 ビットとして a，b の秘匿性を損ねずに $E(ab)$ が求まる．また $c \in \mathbb{Z}/q\mathbb{Z}$ を既知の定数とすれば，

$$(G, Yy^c) = (g^r, y^{(a+c)+r}) = E(a+c),$$
$$(G, Y/y^c) = (g^r, y^{(a-c)+r}) = E(a-c),$$
$$(G^c, Y^c) = (g^{rc}, y^{ac+rc}) = E(ac)$$

より，$E(a)$ と c から $E(a \pm c)$ および $E(ac)$ が求まる．

3.2 Baby-Step Giant-Step 法

Baby-Step Giant-Step（BSGS）法 [15] は，離散対数を解く手法の 1 つとして知られる．m を整数として 3.1 節で定義した $\mathcal{M}$ を $\mathcal{M} := \{0, 1, \ldots, m-1\}$ としたとき，y^a から a を求めるために以下の手続きを行う．

入力： (m, y, y^a)
出力： a

(1) $A_i := y^a/y^i$，$B_j := y^{j\lceil\sqrt{m}\rceil}$ $(i, j = 0, \ldots, \lceil\sqrt{m}\rceil - 1)$ を求める．
(2) $A_{i^*} = B_{j^*}$ となる $0 \le i^*$，$j^* < \lceil\sqrt{m}\rceil$ を求める．
(3) $a^* := j^*\lceil\sqrt{m}\rceil + i^*$ を求め出力する．

a は 0 以上 m $(\le q)$ 未満の整数であるから $a = j^*\lceil\sqrt{m}\rceil + i^*$ を満たす 0 以上 $\lceil\sqrt{m}\rceil$ 未満の整数 i^*，j^* は必ず存在し，$a = j^*\lceil\sqrt{m}\rceil + i^* \Longrightarrow y^{a-i^*} = y^{j^*\lceil\sqrt{m}\rceil} \Longleftrightarrow A_{i^*} = B_{j^*}$ より，ステップ (2) を満たす i^*，j^* も必ず存在する．

3.3 Paillier 暗号の加法準同型性

N を RSA 合成数として，$a, b, c \in \mathbb{Z}/N\mathbb{Z}$ について Paillier 暗号の暗号文 $E(a)$，$E(b)$ と平文 c から，加法準同型 ElGamal 暗号と同様に $E(a \pm b)$，$E(a \pm c)$，$E(ac)$ が求まる．文献 [12] の MPC を実行すれば，a，b の秘匿性を損ねずに $E(ab)$ が求まる．

3.4 閾値暗号

単一の公開鍵に対して複数（n 個）の秘密鍵が存在し，一定数（t 個）以上の秘密鍵を用いた場合に限り復号できる暗号は (t, n)-閾値暗号と呼ばれる．加法準同型 ElGamal 暗号を (n, n)-閾値暗号に拡張する簡単な方法は，秘密鍵 $x \in \mathbb{Z}/q\mathbb{Z}$ を $x \equiv x_0 + \cdots + x_{n-1} \pmod q$ を満たす n 個の秘密鍵 $x_0, \ldots, x_{n-1}$ に置き換え，3.1 節で述べた復号のステップ 1 で $y^a = Y/\prod_{h=0}^{n-1} G^{x_h}$ を求めればよい．計算主体 P_h が x_h を保持する場合は，P_h は G^{x_h} を求めブロードキャストすればすべての計算主体が y^a を求められるようになり，特定の計算主体にのみ G^{x_h} を送信すれば特定の計算主体のみが y^a を求められる．また前記の例では秘密鍵は直接開示せず G を底とした G^{x_h} の離散対数として開示するため，離散対数問題に基づき秘密鍵の秘匿性を損ねずに各計算主体が合意した特定の暗号文のみ復号できる．

3.5 ビット分解プロトコル

MPC の一種であるビット分解プロトコルは，2 以上の整数 ℓ について，a を ℓ ビットの整数，a_i を a の下位 $i+1$ 番目のビット，すなわち $a = \sum_{i=0}^{\ell-1} a_i 2^i$ としたとき，$E(a)$ を入力として a の秘匿性を損ねずに $E(a_i)$（$i = 0, \ldots, \ell-1$）を求め出力する．

3.6 乱数ビット生成プロトコル

どの主体も分からない乱数ビットの暗号文を出力する．基本的な構成方法として，各計算主体 P_h（$h = 0, \ldots, n-1$）が乱数ビット $d^{(h)}$ を生成し，$d = \bigoplus_{h=0}^{n-1} d^{(h)} \in \{0, 1\}$ の暗号文 $E(d)$ を出力する MPC を実行する．文献 [7] では 3 通りのプロトコルが例示されている．

4. 従来手法

semi-honest モデルにおける，Paillier 暗号を用いたビット分解プロトコル [7] について説明する．なお文献 [7] では既存のゼロ知識証明を組合せ malicious モデルとしている．

ℓ を 2 以上の整数とし，n を計算主体の数，N を RSA 合成数，k を $\ell + k + \log_2 n < \log_2 N$ を満たすセキュリティパラメータとする（文献 [7] では $k = 100$ が例示されている）．このとき，ℓ ビットの整数 $a = \sum_{i=0}^{\ell-1} a_i 2^i$ $(a_i \in \{0, 1\})$ について，以下の LSB Gate または LSBs Gate により Paillier 暗号の (t, n)-閾値暗号の暗号文 $E(a)$ を入力として $E(a_{\ell-1}), \ldots, E(a_0)$ を出力する．

[LSB Gate]
(1) 乱数ビット生成プロトコルを 1 回実行し，乱数ビット d の暗号文 $E(d)$ を求める．
(2) 計算主体 P_h（$h = 0, \ldots, n-1$）は 0 以上 $2^{\ell+k-1}$ 未満の乱数 e_h を選び，暗号文 $E(e_h)$ を求めてブロードキャストする．
(3) Paillier 暗号の加法準同型性を利用して，$E(a)$，$E(d)$，$E(e_h)$ から $C := E(a + d + 2\sum_{h=0}^{n-1} e_h)$ を求める．
(4) C を復号し，$\alpha := a + d + 2\sum_{h=0}^{n-1} e_h$ の最下位ビット α_0 を得る．
(5) $E(d)$ および α_0 から $E(d \oplus \alpha_0)$ を計算して出力する（$d \oplus \alpha_0$ は a の最下位ビット a_0 となる）．

$E(a)$ および $E(a_0)$ から，a を 1 ビット右シフトした値の暗号文 $E(2^{-1}(a - a_0))$ を計算すれば，再帰的に $E(a_1), \ldots, E(a_{\ell-1})$ を求めることができる．

[LSBs Gate]
(1) 乱数ビット生成プロトコルを ℓ 回実行し，ℓ 個の乱数ビットの暗号文 $E(d_i)$（$i = 0, \ldots, \ell-1$）を求める．
(2) 計算主体 P_h（$h = 0, \ldots, n-1$）は 0 以上 $2^{\ell+k-1}$ 未満の乱数 e_h を選び，暗号文 $E(e_h)$ を求めてブロードキャストする．
(3) Paillier 暗号の加法準同型性を利用して，$E(a)$，$E(d_i)$，$E(e_h)$ から $C := E(a - \sum_{i=0}^{\ell-1} 2^i d_i - 2^\ell \sum_{h=0}^{n-1} e_h)$ を求

める．

(4) C を復号し，$-N/2$ 以上 $N/2$ 以下の整数 $\alpha := a - \sum_{i=0}^{\ell-1} 2^i d_i - 2^\ell \sum_{h=0}^{n-1} e_h$ を得る．

(5) $\alpha + \sum_{i=0}^{\ell-1} 2^i d_i$ の下位 ℓ ビットが a のビット列と等しいことを利用し，α のビット列および $E(d_i)$ を入力として加算回路の MPC を実行し，$E(a_{\ell-1}), \ldots, E(a_0)$ を出力する．

LSB Gate および LSBs Gate は，a のビット長 ℓ について $\ell + k + \log_2 n < \log_2 N$ の制限がある．これに対して文献 [7] では，a を $\mathbb{Z}/N\mathbb{Z}$ から任意に選べる BITREP Gate が提案されている．

[BITREP Gate]

(1) ℓ を N のビット長とし，乱数ビット生成プロトコルを ℓ 回実行し，ℓ 個の乱数ビットの暗号文 $E(d_i)$ $(i = 0, \ldots, \ell-1)$ を求める．

(2) $d := \sum_{i=0}^{\ell-1} d_i 2^i < N$ を満たすかどうか d_i の秘匿性を損ねずに判定する．満たさない場合は (1) に戻る．

(3) $E(a)$ および $E(d_i)$ から $C := E(a+d)$ を求める．

(4) C を復号し，$\alpha := a + d \bmod N$ を得る．

(5) α のビット列および $E(d_i)$ を入力として減算回路の MPC を実行し，$z := \alpha - d$ の各ビットの暗号文 $E(z_\ell), E(z_{\ell-1}), \ldots, E(z_0)$ を求める（z_ℓ は z の符号ビット）．

(6) $z_\ell = 0$ ならば $z = a$，$z_\ell = 1$ ならば $z = a - N$ より，加算回路の MPC を実行し $a = z + Nz_\ell$ の各ビットの暗号文 $E(a_{\ell-1}), \ldots, E(a_0)$ を出力する．

LSB Gate, LSBs Gate, BITREP Gate はいずれも，加法準同型 ElGamal 暗号の適用が自明でない．3.1 節から分かるように，加法準同型 ElGamal 暗号では復号結果の値域が小さい必要がある．しかしステップ (4) の C の復号結果の値域は，BITREP Gate では $\mathbb{Z}/N\mathbb{Z}$ となり，LSB Gate および LSBs Gate では少なくとも $2^{\ell+k-1}$ 通り存在するため，C の復号が一般に困難となる．なおセキュリティパラメータ k の値は，統計的識別不可能性に基づき 2^{-k} が無視できるほど小さくなるようにし，文献 [7] では $k = 100$ を例示している．

5. 提案方式

加法準同型 ElGamal 暗号を用いたビット分解プロトコルを提案する．すなわち 2 以上の整数 ℓ について a を ℓ ビットの整数，a_i を a の下位 $i+1$ 番目のビットとしたとき，加法準同型 ElGamal 暗号の (t, n)-閾値暗号の暗号文 $E(a)$ を入力として a の秘匿性を損ねずに $E(a_i)$ $(i = 0, \ldots, \ell-1)$ を求め出力する．提案方式は semi-honest モデルにおいて，計算主体に対して a の識別を計算量的に困難とする．

前節でみたように，文献 [7] の方式は，暗号化したまま平文に乱数を足し合わせてから復号し，その復号結果から平文をビット分解した暗号文を生成することを特徴とする．しかし復号結果から平文の秘匿性が損なわれないよう，乱数をある程度大きくとる必要があり，加法準同型 ElGamal 暗号を用いた場合，復号が困難になってしまう．すなわち平文の定義域が小さくても乱数を足すことで定義域が拡大してしまうことが加法準同型 ElGamal 暗号への適用を困難としている．提案方式は，平文の定義域を拡大せず，ランダムな置換を用いて，平文の秘匿性を損ねずに平文をビット分解した暗号文を生成する．基本的なアイデアは，平文の定義域のすべての値をランダムに置換した組と，平文との一致検索を平文の秘匿性を損ねずに行い，一致した箇所と逆置換を用いて平文をビット分解した暗号文を生成する．平文の定義域が拡大しないため，加法準同型 ElGamal 暗号が適用できるようになる．

まず計算主体を P_0 と P_1 の 2 者に限定したセキュア 2 パーティ計算のビット分解プロトコルを 2 方式提案する．次に計算主体を任意数に拡張した MPC のビット分解プロトコルを提案する．E は加法準同型 ElGamal 暗号の (t, n)-閾値暗号とするが，セキュア 2 パーティ計算の場合は $n = t = 2$ となる．

5.1 提案方式 I（2 パーティビット分解プロトコル）

5.1.1 プロトコル

計算主体 P_0 は $E(a)$ および $u, v \in_R \mathbb{Z}/q\mathbb{Z}$ から $E(ua+v)$ を求め計算主体 P_1 に送信し，P_1 は $E(ua+v)$ から y^{ua+v} を復元する．また P_0 は $w \in_R \{0,1\}^\ell$ を選び，y^{ua+v} の 2^ℓ 個の候補 y^{uj+v} $(j = 0, \ldots, 2^\ell - 1)$ のハッシュ値 $\mathcal{H}(y^{uj+v})$ を求め，順序を撹乱するため $(j \oplus w, \mathcal{H}(y^{uj+v}))$ を適当な順序で P_1 に送信する．P_1 は $\mathcal{H}(y^{ua+v}) = \mathcal{H}(y^{uj+v})$ となる $j \oplus w$ $(= a \oplus w)$ を検索し，$a \oplus w$ のビットごとの暗号文 $E(a_i \oplus w_i)$ $(i = 0, \ldots, \ell-1)$ を求め P_0 に送信し，最後に P_0 は $E(a_i \oplus w_i)$ および w_i から $E(a_i)$ を求める．手続きを以下に示す．

入力： $E(a) := (g^r, y^{a+r})$

出力： $E(a_{\ell-1}), \ldots, E(a_0)$

(1) P_0 は以下を行う．

(a) $u, v \in_R \mathbb{Z}/q\mathbb{Z}$，$w \in_R \{0,1\}^\ell$ を選ぶ．

(b) $E(a)$，u，v から $E(ua+v) = (g^{ur}, y^{(ua+v)+ur})$ を求める．

(c) $C_j := \mathcal{H}(y^{uj+v})$ $(j = 0, \ldots, 2^\ell - 1)$ を求める．$\mathcal{H}$ は任意の値を ρ ビットの値に写す衝突困難な一方向性ハッシュ関数とする（ρ はセキュリティパラメータ）．

(d) $E(ua+v)$，$(j, C_{j\oplus w})$ $(j = 0, \ldots, 2^\ell - 1)$ を P_1 に送信する．

(2) P_0 および P_1 は協力して $E(ua+v)$ を復号し，P_1 のみが復号結果 y^{ua+v} を得る[*1]．

[*1] 実際には最終的な復号結果は $ua+v$ だが，本手続きではその途中結果の y^{ua+v} を求めればよく，y^{ua+v} から $ua+v$ を求めることは 3.1 節で述べたとおり離散対数問題に基づき困難である．

(3) P_1 は $C_{j^*\oplus w}=\mathcal{H}(y^{ua+v})$ となる j^* のビットごとの暗号文 $E(j^*_{\ell-1}),\ldots,E(j^*_0)$ を求め P_0 に送信する．

(4) P_0 は w の各ビット $w_{\ell-1},\ldots,w_0$ を用いて，$E(j^*_{\ell-1}),\ldots,E(j^*_0)$ から

$$E(j^*_i\oplus w_i):=\begin{cases}E(j^*_i)E(0) & (w_i=0)\\ E(1-j^*_i)E(0) & (w_i=1)\end{cases}$$

$(i=\ell-1,\ldots,0)$ を求め出力する．

（補足）ステップ (3) で $j^*\oplus w=a$ が成り立ち，ステップ (4) で出力される $E(j^*_i\oplus w_i)$ は a_i の暗号文となる．なおステップ (4) の $E(1)$ は固定の暗号文 $E(1):=(g,y^2)$ でよいが，$E(0)$ は各々独立の乱数を用いて別々の暗号文とする必要がある．E は加法準同型 ElGamal 暗号の $(2,2)$-閾値暗号としたが，単に加法準同型 ElGamal 暗号として秘密鍵を P_1 が保持してもよい．ステップ (3) で j^* を一意に識別するため $\mathcal{H}$ は衝突困難なハッシュ関数である必要がある．

5.1.2 秘匿性

ランダムオラクルを仮定して P_0 および P_1 が a を識別することが計算量的に困難であることを示す．P_0 については，P_0 が P_1 から受信する情報はステップ (3) の暗号文 $\mathsf{View}_0:=(E(j^*_{\ell-1}),\ldots,E(j^*_0))$ のみであるから，加法準同型 ElGamal 暗号の $(2,2)$-閾値暗号の計算量的識別困難性に基づき，ℓ 個のランダムな暗号文の組 $\mathsf{Sim}_0:=(C^{(\ell-1)},\ldots,C^{(0)})$ と View_0 は計算量的に識別困難であり a の秘匿性を損ねない．一方 P_1 については，P_1 が P_0 から受信する情報はステップ (1d) の $E(ua+v)=(g^{ur},y^{(ua+v)+ur})$，$y^{ua+v}$，$(j,\mathcal{H}(y^{u(j\oplus w)+v}))$ となる．$\mathsf{View}_1:=(E(ua+v),y^{ua+v},(j,\mathcal{H}(y^{u(j\oplus w)+v})))$ とする．このとき，以下の手続きを考える．

(1) 平文 $a^*\in\mathcal{M}$ をランダムに選び，a^* の暗号文 $E(a^*)$ を計算する．

(2) $2^{\ell-1}$ 個の ρ ビットの値 C^*_j $(j=0,\ldots,a^*-1,a^*+1,\ldots,2^{\ell-1})$ をランダムに選ぶ．

(3) $C^*_{a^*}:=\mathcal{H}(y^{a^*})$ を計算する．

(4) ℓ ビットの値 w^* をランダムに選び，$\mathsf{Sim}_1:=(E(a^*),y^{a^*},(j,C^*_{j\oplus w^*}))$ とする．

$\mathcal{H}$ をランダムオラクルと仮定すれば，加法準同型 ElGamal 暗号の $(2,2)$-閾値暗号の計算量的識別困難性に基づき，Sim_1 と View_1 は計算量的に識別困難であることを示す．まず $E(a^*)$ と $E(ua+v)$ は明らかに加法準同型 ElGamal 暗号の $(2,2)$-閾値暗号の計算量的識別困難性に基づき計算量的に識別困難である．y^{ua+v} および y^{a^*} はいずれも $\mathcal{G}$ を一様に分布するため完全識別困難である．$a\neq j\oplus w$ となる $\mathcal{H}(y^{u(j\oplus w)+v})$ は，ランダムオラクル仮定から $\{0,1\}^\rho$ を一様に分布し，$a^*\neq j\oplus w^*$ となる $C^*_{j\oplus w^*}$ も仮定から $\{0,1\}^\rho$ を一様に分布するため，完全識別困難である．$a=j\oplus w$ となる $\mathcal{H}(y^{u(j\oplus w)+v})$ は $E(ua+v)$ の復号結果 y^{ua+v} のハッシュ値となり，$a^*=j\oplus w^*$ となる $C^*_{j\oplus w^*}$ も同様に，$E(a^*)$ の復号結果 y^{a^*} のハッシュ値となるため，$\mathcal{H}(y^{ua+v})$ と $C^*_{a^*}$ の分布はランダムオラクル仮定の下で一致し完全識別困難である．

最後に w が識別困難であることを示す．$a=j^*\oplus w$ より，w が識別困難であることを示す必要がある．View_1 の $C_{j\oplus w}=\mathcal{H}(y^{u(j\oplus w)+v})$ について，$\mathcal{H}$ の一方向性の仮定より，$y^{u(j\oplus w)+v}$ を得ることはできない．したがって，$a\neq j\oplus w$ となる $j\oplus w$ から $C_{j\oplus w}$ を識別するためには，View_1 の y^{ua+v} から $y^{u(j\oplus w)+v}$ を求める必要があるが，これは u，v が独立の乱数であることから情報量的に困難である．

5.2 提案方式 II（2 パーティビット分解プロトコル）

BSGS 法を利用して，事前処理ができない場合に提案方式 I よりも計算量および通信量を削減する．なお ℓ は偶数とする．

5.2.1 プロトコル

入力： $E(a):=(g^r,y^{a+r})$

出力： $E(a_{\ell-1}),\ldots,E(a_0)$

(1) P_0 は以下を行う．

(a) $u,v\in_R\mathbb{Z}/q\mathbb{Z}$，$w\in_R\{0,1\}^{\frac{\ell}{2}}$ を選ぶ．

(b) $E(a)$，v から $E(a+v)=(g^r,y^{(a+v)+r})$ を求める．

(c) $B_j:=\mathcal{H}(y^{j\sqrt{2^\ell}+v})^u$ $(j=0,\ldots,\sqrt{2^\ell}-1)$ を求める．$\mathcal{H}$ は任意の値を $\mathcal{G}$ に写す衝突困難な一方向性ハッシュ関数とする．

(d) $E(a+v)$，$(j,B_{j\oplus w})$ $(j=0,\ldots,\sqrt{2^\ell}-1)$ を P_1 に送信する．

(2) P_0 および P_1 は協力して $E(a+v)$ を復号し，P_1 のみが復号結果 y^{a+v} を得る．

(3) P_1 は以下を行う．

(a) $\mu\in_R\mathbb{Z}/q\mathbb{Z}$，$\lambda,\nu\in_R\{0,1\}^{\frac{\ell}{2}}$ を選ぶ．

(b) $B^\mu_{j\oplus w}$，$A_i:=\mathcal{H}(y^{a+v-i})^\mu$ $(i,j=0,\ldots,\sqrt{2^\ell}-1)$ を求める．

(c) $(j,B^\mu_{j\oplus w\oplus\lambda})$，$(i,A_{i\oplus\nu})$ $(i,j=0,\ldots,\sqrt{2^\ell}-1)$ を P_0 に送信する[*2]．

(4) P_0 は以下を行う．

(a) $A^u_{i\oplus\nu}$ $(i=0,\ldots,\sqrt{2^\ell}-1)$ を求める．

(b) $A^u_{i^*\oplus\nu}=B^\mu_{j^*\oplus w\oplus\lambda}$ となる i^*，$j^*\oplus w$ のビットごとの暗号文 $E(i^*_{\frac{\ell}{2}-1}),\ldots,E(i^*_0)$，$E(j^*_{\frac{\ell}{2}-1}\oplus w_{\frac{\ell}{2}-1}),\ldots,E(j^*_0\oplus w_0)$ を求め P_1 に送信する．

(5) P_1 は λ の各ビット $\lambda_{\frac{\ell}{2}-1},\ldots,\lambda_0$，および ν の各ビット $\nu_{\frac{\ell}{2}-1},\ldots,\nu_0$ を用いて，$E(j^*_{\frac{\ell}{2}-1}\oplus w_{\frac{\ell}{2}-1}),\ldots,E(j^*_0\oplus w_0),E(i^*_{\frac{\ell}{2}-1}),\ldots,E(i^*_0)$ から

$$E(j^*_i\oplus w_i\oplus\lambda_i)=\begin{cases}E(j^*_i\oplus w_i)E(0) & (\lambda_i=0)\\ E(1-(j^*_i\oplus w_i)E(0) & (\lambda_i=1)\end{cases}$$

[*2] $(j,B^\mu_{j\oplus w\oplus\lambda})$ は $(j\oplus\lambda,B^\mu_{j\oplus w})$ を $j\oplus\lambda$ の昇順に並べればよい．

$$E(i_i^* \oplus \nu_i) = \begin{cases} E(i_i^*)E(0) & (\nu_i = 0) \\ E(1-i_i^*)E(0) & (\nu_i = 1) \end{cases}$$

$(i = \frac{\ell}{2}-1, \ldots, 0)$ を求め出力する．

（補足）上記の手続きについて，$a = (j^* \oplus w \oplus \lambda)\sqrt{2^\ell} + (i^* \oplus \nu) \Longrightarrow a + v - (i^* \oplus \nu) \equiv (j^* \oplus w \oplus \lambda)\sqrt{2^\ell} + v \pmod q \Longrightarrow (\mathcal{H}(y^{a+v-(i^* \oplus \nu)})^\mu)^u = (\mathcal{H}(y^{(j^* \oplus w \oplus \lambda)\sqrt{2^\ell}+v})^u)^\mu \Longleftrightarrow A_{i^* \oplus \nu}^{u} = B_{j^* \oplus w \oplus \lambda}^{\mu}$ より，ステップ (5) の $E(j_i^* \oplus w_i \oplus \lambda_i)$ は $a_{\frac{\ell}{2}+i}$ の暗号文，$E(i_i^* \oplus \nu_i)$ は a_i の暗号文となる．なお提案方式 I と異なり，P_1 は秘密の値 w を含む暗号文を P_0 から受信しているため（ステップ (4b)），E を単に加法準同型 ElGamal 暗号として秘密鍵を P_1 が保持できない．P_0 についても同様である．ステップ (4b) で i^*，$j^* \oplus w$ を一意に識別するため $\mathcal{H}$ は衝突困難なハッシュ関数である必要がある．

5.2.2 秘匿性

5.1.2 項同様，ランダムオラクルを仮定して P_0 および P_1 が a を識別することが計算量的に困難であることを示す．

P_0 が P_1 から受信する情報は，ステップ (3c) の $(j, B_{j \oplus w \oplus \lambda}^{\mu})$，$(i, A_{i \oplus \nu})$，およびステップ (5) の暗号文となる．ステップ (5) の暗号文の識別困難性は 5.1.2 項と同様，加法準同型 ElGamal 暗号の $(2,2)$-閾値暗号の計算量的識別困難性に基づき，ランダムな暗号文の組と計算量的に識別困難である．$\mathcal{H}$ をランダムオラクルと仮定すれば，hybrid argument より，DDH 仮定に基づき $B_{j \oplus w \oplus \lambda}^{\mu}$，$A_{i \oplus \nu}$ の列は $\mathcal{G}$ からランダムに選んだ列と識別困難となり a の秘匿性を損ねない．一方 P_1 が P_0 から受信する情報は，ステップ (1d) の $E(a+v)$，y^{a+v}，$(j, B_{j \oplus w})$，およびステップ (4b) の暗号文となる．ステップ (4b) の暗号文の識別困難性は 5.1.2 項と同様，加法準同型 ElGamal 暗号の $(2,2)$-閾値暗号の計算量的識別困難性に基づき，ランダムな暗号文の組と計算量的に識別困難である．y^{a+v} は独立な乱数 v より $|\mathcal{G}|$ のランダムな値と分布は等しく，$B_{j \oplus w}$ の列は DDH 仮定に基づき $\mathcal{G}$ からランダムに選んだ列と識別困難となる．

5.3 提案方式 III（マルチパーティビット分解プロトコル）

5.3.1 プロトコル

計算主体を任意数に拡張した MPC のビット分解プロトコルを提案する．$E(a)$ および各計算主体 P_h が選んだ乱数 $r_{h,j} \in \mathbb{Z}/q\mathbb{Z} - \{0\}$ $(j = 0, \ldots, 2^\ell - 1)$，$w^{(h)} \in \{0,1\}^\ell$ から $(j \bigoplus_{h=0}^{n-1} w^{(h)}, E((a-j)\prod_{h=0}^{n-1} r_{h,j}))$ を求め，第 2 要素の暗号文の復号結果が 0 となる第 1 要素 $j \bigoplus_{h=0}^{n-1} w^{(h)}$ $(= a \bigoplus_{h=0}^{n-1} w^{(h)})$ についてビットごとの暗号文 $E(a_i \bigoplus_{h=0}^{n-1} w_i^{(h)})$ を求め，$E(a_i)$ を得る．手続きを以下に示す．

入力： $E(a) := (g^r, y^{a+r})$

出力： $E(a_{\ell-1}), \ldots, E(a_0)$

(1) $E(a-j) := (G_{0,j}, Y_{0,j})$ $(j = 0, \ldots, 2^\ell - 1)$ を求める．

(2) P_h $(h = 0, \ldots, n-1)$ は順に以下を行う．

(a) $r_{h,j} \in_R \mathbb{Z}/q\mathbb{Z}$ $(j = 0, \ldots, 2^\ell - 1)$，$w^{(h)} \in_R \{0,1\}^\ell$ を選ぶ．

(b) $(J_{h+1,j}, G_{h+1,j}, Y_{h+1,j}) := (J_{h,j} \oplus w^{(h)}, G_{h,j}^{r_{h,j}}, Y_{h,j}^{r_{h,j}})$ を求め $J_{h+1,j}$ の昇順に並べ P_{h+1} に送信する．ただし $J_{0,j} := j$ とし，$h = n-1$ であればブロードキャストする．

(3) $(G_{n,j}, Y_{n,j})$ $(= E((a-j)\prod_{h=0}^{n-1} r_{h,j}))$ の復号結果が 0 となる $J^* := J_{n,j^*}$ のビットごとの暗号文 $E(J_{\ell-1}^*), \ldots, E(J_0^*)$ を求める．

(4) P_h $(h = 0, \ldots, n-1)$ は順に以下を行う．

(a) $w^{(h)}$ の各ビット $w_{\ell-1}^{(h)}, \ldots, w_0^{(h)}$ を用いて，$E(J_{\ell-1}^*), \ldots, E(J_0^*)$ から

$$E\left(J_i^* \bigoplus_{j=0}^{h} w_i^{(j)}\right) := \begin{cases} E(J_i^* \bigoplus_{j=0}^{h-1} w_i^{(j)})E(0) & (w_i^{(h)} = 0) \\ E(1 - (J_i^* \bigoplus_{j=0}^{h-1} w_i^{(j)}))E(0) & (w_i^{(h)} = 1) \end{cases}$$

$(i = \ell - 1, \ldots, 0)$ を求め P_{h+1} に送信する．$h = n-1$ であれば出力する．

5.3.2 秘匿性

ステップ (2b) の $G_{h+1,j}$，$Y_{h+1,j}$ の列は 5.2.2 項同様，DDH 仮定に基づき $\mathcal{G}$ からランダムに選んだ列と識別困難となる．ステップ (3) の復号結果については，$a - j \neq 0$ であればすべての主体が生成した 0 以外の独立の乱数の積 $\prod_{h=0}^{n-1} r_{h,j}$ が乗じられているため，$\mathbb{Z}/q\mathbb{Z} - \{0\}$ からとられた乱数と分布は等しく，識別困難である．ステップ (4a) の暗号文は，加法準同型 ElGamal 暗号の (t,n)-閾値暗号の計算量的識別困難性に基づき，ランダムな暗号文の組と計算量的に識別困難である．

6. 効率

本章では，提案方式の計算コストおよび通信コストを評価し，4 章で紹介した文献 [7] の LSB Gate，LSBs Gate，BITREP Gate のうち，計算コストおよび通信コストが最も小さい LSB Gate との比較を行う．また事前処理可能な部分を除いた部分についても評価・比較を行う．なお LSB Gate は通信回数が LSBs Gate および BITREP Gate よりも多くなる．

事前処理なしの全体の処理コストを**表 1** に，事前処理可能な部分を除いた処理コストを**表 2** にまとめる．$\mathsf{Exp}_{\mathsf{ElG}}$，$\mathsf{Mul}_{\mathsf{ElG}}$，$\mathsf{Div}_{\mathsf{ElG}}$ はそれぞれ $\mathcal{G}$ 上の冪乗演算，乗算，除算 1 回あたりの計算コストとする．$\mathcal{P} := \mathbb{Z}/N\mathbb{Z}$ とし，$\mathsf{Exp}_{\mathsf{Pai}}$，$\mathsf{Mul}_{\mathsf{Pai}}$，$\mathsf{Div}_{\mathsf{Pai}}$ はそれぞれ $\mathcal{P}$ 上の冪乗演算，乗算，除算 1 回あたりの計算コストとする．Hash は任意の値を ρ ビッ

表 1 効率比較（事前処理なし）
Table 1 Efficiency comparison (Total).

	提案方式 I	提案方式 II	提案方式 III	LSB Gate [7]
計算コスト	$(4\ell+6)\mathsf{Exp}_{\mathsf{ElG}}$ $+(2^\ell+2\ell+2)\mathsf{Mul}_{\mathsf{ElG}}$ $+(2\ell+1)\mathsf{Div}_{\mathsf{ElG}}$ $+2^\ell\mathsf{Hash}$ $\mathsf{Search}(2^\ell)$	$(4\sqrt{2^\ell}+2\ell+4)\mathsf{Exp}_{\mathsf{ElG}}$ $+(\sqrt{2^\ell}+2\ell+2)\mathsf{Mul}_{\mathsf{ElG}}$ $+(\sqrt{2^\ell}+2\ell)\mathsf{Div}_{\mathsf{ElG}}$ $+2\sqrt{2^\ell}\mathsf{Hash}$ $\mathsf{Search}(2\sqrt{2^\ell})$	$((n+2)2^\ell+(2n+2)\ell)\mathsf{Exp}_{\mathsf{ElG}}$ $+(2n\ell+n-1)\mathsf{Mul}_{\mathsf{ElG}}$ $+(2^{\ell+1}+2n\ell-1)\mathsf{Div}_{\mathsf{ElG}}$	$(2n+2)(\ell-1)\mathsf{Exp}_{\mathsf{Pai}}$ $+n\ell\mathsf{Mul}_{\mathsf{Pai}}$ $3(\ell-1)\mathsf{Div}_{\mathsf{Pai}}$ $+(\ell-1)\mathsf{RandBitGen}$
通信コスト	$(2\ell+3)\lvert\mathcal{G}\rvert$ $+2^\ell\lvert\mathcal{H}\rvert$	$(3\sqrt{2^\ell}+2\ell+4)\lvert\mathcal{G}\rvert$	$(3n2^\ell+2n\ell)\lvert\mathcal{G}\rvert$	$2n(\ell-1)\lvert\mathcal{P}\rvert$ $+(\ell-1)\mathsf{RandBitGen}$

表 2 効率比較（事前処理あり）
Table 2 Efficiency comparison (Excluding pre-computation).

	提案方式 I	提案方式 II	提案方式 III	LSB Gate [7]
計算コスト	$4\mathsf{Exp}_{\mathsf{ElG}}$ $+(2\ell+2)\mathsf{Mul}_{\mathsf{ElG}}$ $+(2\ell+1)\mathsf{Div}_{\mathsf{ElG}}$ $+\mathsf{PreSearch}(2^\ell)$	$(3\sqrt{2^\ell}+2)\mathsf{Exp}_{\mathsf{ElG}}$ $+(2\ell+2)\mathsf{Mul}_{\mathsf{ElG}}$ $+(\sqrt{2^\ell}+2\ell)\mathsf{Div}_{\mathsf{ElG}}$ $+\sqrt{2^\ell}\mathsf{Hash}$ $+\mathsf{PreSearch}(2\sqrt{2^\ell})$	$((n+2)2^\ell+2\ell)\mathsf{Exp}_{\mathsf{ElG}}$ $+(2n\ell+n-1)\mathsf{Mul}_{\mathsf{ElG}}$ $+(2^{\ell+1}+2n\ell-1)\mathsf{Div}_{\mathsf{ElG}}$	$2(\ell-1)\mathsf{Exp}_{\mathsf{Pai}}$ $+n(\ell-1)\mathsf{Mul}_{\mathsf{Pai}}$ $+3\mathsf{Div}_{\mathsf{Pai}}$
通信コスト	$(2\ell+3)\lvert\mathcal{G}\rvert$	$(3\sqrt{2^\ell}+2\ell+4)\lvert\mathcal{G}\rvert$	$(3n2^\ell+2n\ell)\lvert\mathcal{G}\rvert$	$n\ell\lvert\mathcal{P}\rvert$

トまたは $\mathcal{G}$ に写すハッシュ関数の演算 1 回あたりの計算コストとする．$\mathsf{Search}(x)$ は x 個の値からの一致検索のコストとし，$\mathsf{PreSearch}(x)$ は事前処理可能な部分を除いた，x 個の値からの一致検索のコストとする．RandBitGen は 3.6 節で述べた 3 通りの乱数ビット生成プロトコルのいずれかを用いたときの 1 回あたりの計算コストまたは通信コストとする．通信コストの単位はビットとする．

6.1 提案方式 I の効率

ステップ (1c) で y^u，y^v を求めた後 $2^\ell-1$ 回の乗算および 2^ℓ 回のハッシュ演算を行い $C_j=\mathcal{H}(y^{uj+v})$ $(j=0,\ldots,2^\ell-1)$ を計算し，ステップ (1d) では 2^ℓ 個の $(j,C_{j\oplus w})$ の組を送信している．これらの処理は ℓ が大きくなるにつれ指数的に増大するが，C_j は入力と独立であり，事前処理可能である．またステップ (3) では 2^ℓ 個の値からの一致検索を必要とするが，$(j,C_{j\oplus w})$ $(j=0,\ldots,2^\ell-1)$ を $C_{j\oplus w}$ について事前にソートしておけば，ハッシュテーブル等を用いて効率良く処理できる．

事前処理ありの場合，提案方式 I の計算コストおよび通信コストは特に削減できることが表 1 および 2 から分かる．事前処理ありの場合の通信コストは提案方式 II，III よりも小さく，具体的にたとえば $|\mathcal{G}|=1{,}024$，$|\mathcal{P}|=2{,}048$ とすれば，任意の ℓ について LSB Gate よりも小さくなる．事前処理ありの場合の計算コストについても，冪乗演算は定数回であり，ハッシュテーブル等を用いれば一致検索は高速処理できるため，LSB Gate よりもコストは小さいといえる．

6.2 提案方式 II の効率

表 2 から分かるように，事前処理ありの場合は提案方式 II の計算コストは $O(\sqrt{2^\ell})$，通信コストは $O(\sqrt{2^\ell}|\mathcal{G}|)$ となり，提案方式 I よりも効率が悪い．しかし事前処理なしの場合は，提案方式 I の計算コストは $O(2^\ell)$，通信コストは $O(2^\ell(|\mathcal{G}|+|\mathcal{H}|))$ であるのに対し，提案方式 II の計算コストは $O(\sqrt{2^\ell})$，通信コストは $O(\sqrt{2\ell}|\mathcal{G}|)$ となり，提案方式 I よりも効率が良い．

事前処理なしの場合の処理コストについて具体的な値を用いて提案方式 I と比較する．ElGamal 暗号を楕円曲線上で実装したとき，$\mathsf{Exp}_{\mathsf{ElG}}$，$\mathsf{Mul}_{\mathsf{ElG}}$，$\mathsf{Div}_{\mathsf{ElG}}$ はそれぞれ楕円スカラー倍，楕円加算，楕円減算に相当し，楕円加算および楕円減算の計算コストはほぼ等しくなる．また楕円加算と楕円スカラー倍演算の計算コストの比は Binary Method を仮定すればおよそ $1:1.5|\mathcal{G}|$ と見積もることができる．ハッシュ関数の計算コストおよび一致検索の計算コストは提案方式 I の方が大きく，こられを無視すると，上記の条件において，$|\mathcal{G}|=160$ のとき，少なくとも $\ell\geq 20$ であれば提案方式 II の計算コストの方が小さい．$|\mathcal{G}|=240$ のときは少なくとも $\ell\geq 21$ であれば提案方式 II の計算コストの方が小さくなる．

6.3 提案方式 III の効率

事前処理の有無にかかわらず各計算主体は $O(2^\ell)$ 回の冪乗演算を行い，通信コストも $O(2^\ell|\mathcal{G}|)$ となり効率が悪い．しかし 7.3 節で例示するように小さな ℓ について適用するような場合は，加法準同型 ElGamal 暗号の高速性の利点より，効率良く実行できることが期待される．6.2 節同様，ElGamal 暗号を楕円曲線上で実装するものとし，文献 [14]

では，OEF（Optimal Extension Fields）上の 174 ビット楕円スカラー倍演算は 2,048 ビットを法とした冪乗剰余演算に比べおよそ 200 倍高速と見積もられていることから，$\mathsf{Mul}_{\mathsf{EIG}} = \mathsf{Div}_{\mathsf{EIG}}$，$261\mathsf{Mul}_{\mathsf{EIG}} = \mathsf{Exp}_{\mathsf{EIG}}$，$\mathsf{Exp}_{\mathsf{Pai}} = 200\mathsf{Exp}_{\mathsf{EIG}}$ とし，事前処理ありの LSB Gate と計算コストを表 2 に基づき比較する．すると，LSB Gate の乗算および除算の計算コストを無視した場合でも，$n = 3$ のとき $\ell \leq 9$ であれば提案方式 III の計算コストの方が小さい．$n = 4$ のときは $\ell \leq 5$ であれば提案方式 III の計算コストの方が小さくなる．同様の条件で事前処理ありの LSB Gate と通信コストを表 2 に基づき比較すると，$n = 3$ または $n = 4$ のいずれの場合でも $\ell \leq 3$ であれば提案方式 III の通信コストの方が小さくなる．

7. 応用

本章ではビット分解プロトコルが有効となる具体的な MPC の演算について考察する．

ビット分解プロトコルの基本的な用途は，算術演算と組合せ論理回路演算の併用である．加法準同型性を用いて，ビットごとの暗号文 $E(a_i)$ から整数の暗号文 $E(\sum_{i=0}^{\ell-1} a_i)$ を求めることは容易である．すなわち，論理回路演算結果の暗号文を中間出力として，算術演算の MPC の入力に利用できる．しかし逆に算術演算結果の暗号文を中間出力として，論理回路演算の MPC の実行は一般にできないため，算術演算も論理回路演算として実行する必要があり，処理時間がかかってしまう．ビット分解プロトコルを用いれば，算術演算結果の暗号文を論理回路演算の MPC の入力に変換できる．また文献 [7] では，ビット分解プロトコルを用いて整数 a，b の暗号文 $E(a)$，$E(b)$ から $E(a^b)$ を求める応用が述べられている．以下では，その他の簡単な応用を与える．

7.1 一括 XOR

大きな整数 m について，ビット b_i（$i = 0, \ldots, m-1$）の暗号文 $E(b_i)$ から $E(\bigoplus_{i=0}^{m-1} b_i)$ を求める．

$$\bigoplus_{i=0}^{m-1} b_i = 1 \Leftrightarrow \mathrm{LSB}\left(\sum_{i=0}^{m-1} b_i\right) = 1$$

より（$\mathrm{LSB}(x)$ は x の最下位ビット），加法準同型性を用いて $E(b_i)$ から $E(\sum_{i=0}^{m-1} b_i)$ を求め，ビット分解プロトコルを実行し $\sum_{i=0}^{m-1} b_i$ の最下位ビットの暗号文を求める．

MPC を単純に実行して $E(\bigoplus_{i=0}^{m-1} b_i)$ を求める場合，$m-1$ 回の XOR の MPC を実行する必要がある．しかし加法準同型性を用いて $E(\sum_{i=0}^{m-1} b_i)$ を求めれば，$\sum_{i=0}^{m-1} b_i$ はたかだか $1 + \lfloor \log_2 m \rfloor$ ビットであるため，ビット分解プロトコルの適用は十分効果的と考えられる．

7.2 一括 OR，一括 AND

大きな整数 m について，ビット b_i（$i = 0, \ldots, m-1$）の暗号文 $E(b_i)$ から $E(\bigvee_{i=0}^{m-1} b_i)$ または $E(\bigwedge_{i=0}^{m-1} b_i)$ を求める．

$$\bigvee_{i=0}^{m-1} b_i = 0 \Leftrightarrow \sum_{i=0}^{m-1} b_i = 0,$$
$$\bigwedge_{i=0}^{m-1} b_i = 1 \Leftrightarrow \sum_{i=0}^{m-1} b_i = m$$

より，加法準同型性を用いて $E(b_i)$ から $E(\sum_{i=0}^{m-1} b_i)$ または $E(m - \sum_{i=0}^{m-1} b_i)$ を求め，ビット分解プロトコルを実行し $\sum_{i=0}^{m-1} b_i$ または $m - \sum_{i=0}^{m-1} b_i$ のビットごとの暗号文を求め，すべての暗号文の復号結果が 0 かどうか等号判定回路の MPC を実行し判定する．または当該ビットごとの暗号文を入力として，上記の処理を繰り返してすべてのビットの OR を求めてもよい．

MPC を単純に実行して $E(\bigvee_{i=0}^{m-1} b_i)$ または $E(\bigwedge_{i=0}^{m-1} b_i)$ を求める場合，$m-1$ 回の OR または AND の MPC を実行する必要がある．しかし加法準同型性を用いて $E(\sum_{i=0}^{m-1} b_i)$ または $E(m - \sum_{i=0}^{m-1} b_i)$ を求めれば，$\sum_{i=0}^{m-1} b_i$ および $m - \sum_{i=0}^{m-1} b_i$ はたかだか $1 + \lfloor \log_2 m \rfloor$ ビットであるため，7.1 節の方法同様の効果が期待できる．

7.3 等号判定

大きな整数 x，y の暗号文 $E(x)$，$E(y)$ から x，y の等号判定結果

$$b = \begin{cases} 1 & (x = y) \\ 0 & (x \neq y) \end{cases}$$

の暗号文 $E(b)$ を求める．$x = y \Leftrightarrow x - y = 0$ より，加法準同型性を用いて $E(x)$, $E(y)$ および乱数 r_i $(i = 0, \ldots, \kappa-1)$ から $E(r_0(x-y)), \ldots, E(r_{\kappa-1}(x-y))$ を求め（κ はパラメータ），$r_i(x-y)$ を ℓ ビットの整数と見なしてビット分解プロトコルを実行し，ビットごとの暗号文を求める（ℓ はパラメータ）．そしてすべてのビットの暗号文の復号結果が 0 かどうか等号判定回路の MPC を実行し判定する．すると $x = y$ であれば必ずすべての復号結果が 0 となるが，$x \neq y$ であれば，$r_i(x-y)$ が ℓ ビットを超えて等号判定回路の MPC がエラーを返すか，$2^{-\kappa\ell}$ 以下の確率ですべての復号結果が 0 となる．すなわち偽陽性の確率を十分小さくできる．

本手法は，κ を大きくとり ℓ を小さくすることができるため，ℓ について指数的に処理が増大する 5.1 節や 5.3 節の提案方式を適用する際に特に有効と考えられる．

8. まとめ

本論文では，従来手法では実現が困難であった，加法準同型 ElGamal 暗号を用いて暗号化された整数を明かすこ

となくビットごとの暗号文に変換する 3 通りのビット分解プロトコルを構成した．特に事前処理を許す 2 パーティ限定の提案方式は，従来と比べ大きく処理削減できることを示した．またビット分解プロトコルの応用として，大量のビットの暗号文から，XOR，OR，AND の基本論理演算結果の暗号文を効率良く求める方法，および大きな整数の等号判定結果の暗号文を高い確率で効率良く求める方法を提案した．

参考文献

[1] Yao, A.C.: Protocols for secure computations, *Proc. FOCS '82*, pp.160–164, IEEE Press (1982).
[2] Agrawal, R. and Srikant, S.: Privacy-Preserving Data Mining, *Proc. SIGMOD 2000*, pp.439–450, ACM (2000).
[3] Lindell, Y. and Pinkas, B.: Privacy Preserving Data Mining, *Crypto 2000*, LNCS 1880, pp.20–24, Springer-Verlag (2000).
[4] Algesheimer, J., Camenisch, J. and Shoup, V.: Efficient computation modulo a shared secret with application to the generation of shared safe-prime products, *CRYPTO 2002*, LNCS 2442, pp.417–432, Springer-Verlag (2002).
[5] Damgård, I., Fitzi, M., Kiltz, E., Nielsen, J.B. and Toft, T.: Unconditionally secure constant-rounds multi-party computation for equality, comparison bits and exponentiation, *TCC 2006*, LNCS 3876, pp.285–304, Springer-Verlag (2006).
[6] Nishide, T. and Ohta, K.: Multiparty computation for interval, equality, and comparison without bit-decomposition protocol, *PKC 2007*, LNCS 4450, pp.343–360 (2007).
[7] Schoenmakers, B. and Tuyls, P.: Efficient binary conversion for Paillier encrypted values, *EUROCRYPT 2006*, LNCS 4004, pp.522–537, Springer-Verlag (2006).
[8] Yao, A.C.: How to generate and exchange secrets, *Proc. FOCS '86*, pp.162–167 (1986).
[9] Goldreich, O., Micali, S. and Wigderson, A.: How to Play any Mental Game or a Completeness Theorem for Protocols with Honest Majority, *Proc. STOC '87*, pp.218–229 (1987).
[10] Ben-Or, M., Goldwasser, S. and Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation, *Proc. STOC '88*, pp.1–10 (1988).
[11] Chaum, D., Crepeau, C. and Damgård, I.: Multiparty unconditionally secure protocols, *Proc. STOC '88*, pp.11–19 (1988).
[12] Cramer, R., Damgård, I. and Nielsen, J.B.: Multiparty computation from threshold homomorphic encryption, *EUROCRYPT 2001*, LNCS 2045, pp.280–300, Springer-Verlag (2001).
[13] Schoenmakers, B. and Tuyls, P.: Practical two-party computation based on the conditional gate, *ASIACRYPT 2004*, LNCS 3329, pp.119–136, Springer-Verlag (2004).
[14] Yamamoto, G., Chida, K., Nasciment, A., Suzuki, K. and Uchiyama, S.: Efficient, non-optimistic secure circuit evaluation based on the ElGamal encryption, *WISA 2005*, LNCS 3786, pp.328–343, Springer-Verlag (2005).
[15] Shanks, D.: Class number, A Theory of Factorization and Genera, *Symp. Pure Math.*, Vol.20, pp.415–440, AMS (1971).
[16] Jakobsson, M. and Juels, A.: Mix and match: Secure function evaluation via ciphertexts, *ASIACRYPT 2000*, LNCS 1976, pp.162–177, Springer-Verlag (2000).
[17] Cramer, R., Damgård, I. and Schoenmakers, B.: Proofs of partial knowledge and simplified design of witness hiding protocols, *CRYPTO '94*, LNCS 839, pp.171–181, Springer-Verlag (1994).

千田 浩司 （正会員）

2000 年早稲田大学大学院理工学研究科数理科学専攻修士課程修了．同年日本電信電話（株）入社．博士（工学）．暗号応用技術，プライバシ保護技術の研究開発に従事．SCIS2000 論文賞，2011 年度本会論文賞．電子情報通信学会会員．

五十嵐 大 （正会員）

2008 年東京大学大学院情報理工学系研究科コンピュータ科学専攻修士課程修了．同年日本電信電話（株）入社．プライバシ保護技術の研究に従事．PPL2008 論文奨励賞，CSS2009 論文賞，SCIS2011 論文賞，2011 年度本会論文賞，CSS2012 論文賞，2012 年度本会山下記念研究賞．ソフトウェア科学会会員．

高橋 克巳 （正会員）

1988 年東京工業大学理学部卒業．同年日本電信電話（株）入社．2006 年東京大学大学院情報理工学系研究科博士課程修了．博士（情報理工学）．NTT 研究所にて情報検索，データマイニング，位置情報処理，および情報セキュリティ，暗号，プライバシ，セキュリティ社会科学の研究に従事．本会理事．