Vol.2015-CSEC-69 No.11
Vol.2015-IOT-29 No.11
2015/5/21

XOR演算ベースの閾値秘密分散法から秘密計算法を構成する試み

須賀 祐治[1,a)]

概要：排他的論理和演算を用いた高速な (k,n)-閾値秘密分散法は栗原ら，藤井らによって独立に提案されている．彼らの方式はともにシェアのサイズが分散対象データのサイズに等しい理想的な方式であり，分散・復元時に XOR 演算のみを用いるため非常に高速に処理できるメリットを持つ．筆者はこれらを拡張し任意の 2 以上の整数 m に対して分散対象データを m 個に等分割して $(2,2^m)$-閾値秘密分散法の構成方法を提案し，小さい次数についてその存在性について報告した．本稿にて，この秘密分散法を秘密計算に適用する方法について検討した．

キーワード：閾値秘密分散法, 排他的論理和演算, 理想的な秘密分散法, 秘密計算法,

A trail about construction of secure multiparty computation from exclusive-OR operations based threshold secret sharing schemes

Yuji Suga[1,a)]

Abstract: Fast (k, n)-threshold secret sharing schemes with exclusive-OR operations have proposed by Kurihara et al. and Fujii et al. independently. Their method are ideal that share size is equal to the size of the data to be distributed with the benefits that can be handled very fast for using only XOR operation at distribution and restoration processes. Moreover an author extended these schemes and reported existences of $(2,2^m)$-threshold secret sharing schemes with small order. This paper attempts to construct secure multiparty computations by using XOR-based secret sharing schemes.

Keywords: (k, n)-threshold secret sharing scheme, exclusive-OR operation, ideal secret sharing scheme, secure multiparty computation

1. はじめに

東日本大震災以降，ディザスタリカバリや事業継続計画に対する関心が高まっている．具体的には安定的な電力供給が見込めない，もしくは地震の余波によるネットワーク遮断の可能性からクラウドリソースを利用できない状況が鑑みられている．またデータセンターへの物理的な被害により，データ紛失という問題も考えられる．この可用性の課題に対して秘密分散共有法のクラウド適用の検討が試みられている．

[1] 株式会社インターネットイニシアティブ
Internet Initiative Japan Inc., Iidabashi Grand Bloom, 2-10-2 Fujimi, Chiyoda-ku, Tokyo 102-0071, Japan
[a)] suga@iij.ad.jp

秘密分散共有法

秘密分散共有法 (Secret Sharing Scheme; SSS) は，CIA モデルによれば秘匿性と可用性の要件をバランスよく持つ技術として認識されており文献 [1] などにより提案された概念である．例えば最もシンプルな例の一つとして (k,n)-しきい値秘密分散法が存在する．秘密情報 S を n 個の分散情報（シェア）に符号化し配布した状態で，任意の k 個の分散情報からは S を復号可能であるが，任意の $k-1$ 個の分散情報からは S に関する情報は全く得られないという性質を持つ．本技術の導入効果として 1）漏洩リスクの分散（一部漏洩しても暴露されない），2）紛失リスクの分

散（一部紛失しても復元可能）の 2 つがあり，アプリケーション，ユースケースに応じて上記パラメータ n, k を選択できる自由度を持つ．

秘密分散技術適用におけるビジネス上の課題

実際に秘密分散共有技術をクラウド環境でサービスインする動きも見受けられる．しかし公開情報からはサービスを構成するための具体的な技術要素が不明瞭である．そのため，クラウドサービス事業者が不正する，構成方法自体が脆弱であるなどの事由により，顧客から預かった情報が復元できてしまう可能性がある．このビジネス上の課題に対して，何がしかの技術適用とその技術の確からしさを顧客に提示することが必要である．しかし「なんとなく」安心＆安全に使えるクラウドサービスではあってはならない．顧客への説明責任を踏まえ，シェアの一部を利用者側でプライベートクラウドとして運用するというビジネスモデルを好む顧客も潜在的には存在するとも考えられる．可用性要件の節で前述したように「お墨付き制度」だけでクラウド事業者を選択するのではなく，透明性のあるセキュリティ技術を提供することで，顧客への安心感を提供するべきである．

排他的論理和演算で構成される秘密分散法

排他的論理和演算で構成される (k, n)-しきい値秘密分散方式 (XOR-(k, n)-SSS) は藤井, 多田ら [2], [3], [4]，栗原ら [5], [6] によって独立に提案されている．シンプルな具体例として [2] に記載の XOR-$(2, 3)$-SSS について説明する．秘密情報 $M = M_1||M_2$（M_i のサイズは d ビット）に対して d ビットデータ R_0, R_1 を生成し，シェア $W_i (i = 0, 1, 2)$ を

W_0	$(M_0 \oplus R_0) \mid\mid (M_2 \oplus R_1)$
W_1	$(M_1 \oplus R_0) \mid\mid (M_0 \oplus R_1)$
W_2	$(M_2 \oplus R_0) \mid\mid (M_1 \oplus R_1)$

とおくことで構成可能である．ただし $||$ はデータの連結を，$\oplus$ はビットごとの排他的論理和を示し M_0 は各ビットが全て 0 で構成されているものとする．本方式は，各シェアサイズが分散対象のデータサイズと同じとなる理想的な秘密分散法式となっている．

2. XOR-(k, n)-SSS 構成方法の再考

ターゲットデータ M の分割数を $n^{'}$ とすると XOR-$(2, n)$-SSS のシェア $W_i (i = 0, \ldots, n)$ を次のマトリクスで表現することとする．ここで $n^{''} := n^{'} - 1$, $W_i = W_{i0} \mid\mid \ldots \mid\mid W_{in^{''}}$ とする．

W_0	W_{00}	$\ldots$	$W_{0n^{''}}$
W_1	W_{10}	$\ldots$	$W_{1n^{''}}$
$\ldots$	$\ldots$	$\ldots$	$\ldots$
W_i	W_{i0}	$\ldots$	$W_{in^{''}}$
$\ldots$	$\ldots$	$\ldots$	$\ldots$
W_n	W_{n0}	$\ldots$	$W_{nn^{''}}$

2.1 栗原らの方式 [6]

まず巡回置換行列を用いた XOR-$(2, n)$-SSS の構成方式について説明する．素数 n_p に対して $n = n_p$ となる XOR-$(2, n)$-SSS with $n^{'} = n_p - 1$ をシェア $W_i (i = 0, \ldots, n_p - 1)$ は $n^{'}$ 個 $W_{ij} (j = 0, \ldots, n^{'} - 1)$ のパーツ の連結と考える．ターゲットデータ M は $M_1, \ldots, M_{n^{'}}$ の連結で各データ長を d ビット，$M_0 \in \{0\}^d$ とする．また M_0 と同じサイズのダミーデータ $R_i (i = 0, \ldots, n_p)$ をランダムに選択する．このとき W_{ij} を $M_j \oplus R_{(j-i) \mod n_p}$ とおく．

上記構成方式に基づいて構成される XOR-$(2, 3)$-SSS with $n^{'} = 2$ は以下の通りである．

Example1 (XOR-$(2, 3)$-SSS [6]) $M = M_1 || M_2$ $(n^{'} = 2), M_0 \in \{0\}^d$

W_0	$M_0 \oplus R_0$	$M_1 \oplus R_1$
W_1	$M_2 \oplus R_0$	$M_0 \oplus R_1$
W_2	$M_1 \oplus R_0$	$M_2 \oplus R_1$

さらに $n_p = 5$ のときに構成される XOR-$(2, 5)$-SSS with $n^{'} = 4$ は以下の通りである．

Example2 (XOR-$(2, 5)$-SSS [6])

W_0	$M_0 \oplus R_0$	$M_1 \oplus R_1$	$M_2 \oplus R_2$	$M_3 \oplus R_3$
W_1	$M_4 \oplus R_0$	$M_0 \oplus R_1$	$M_1 \oplus R_2$	$M_2 \oplus R_3$
W_2	$M_3 \oplus R_0$	$M_4 \oplus R_1$	$M_0 \oplus R_2$	$M_1 \oplus R_3$
W_3	$M_2 \oplus R_0$	$M_3 \oplus R_1$	$M_4 \oplus R_2$	$M_0 \oplus R_3$
W_4	$M_1 \oplus R_0$	$M_2 \oplus R_1$	$M_3 \oplus R_2$	$M_4 \oplus R_3$

2.2 CSS2012 方式 [7]

栗原らの方式 [6] では素数 n_p に対して $n^{'} = n_p - 1 = n - 1$ でしか構成できないという制約があった．これを解消するために以下のように CSS2012 方式 [7] が提案されている．

素数 n_p に対して $n = n_p + 1$ となる XOR-$(2, n)$-SSS with $n^{'} = n_p - 1$ をシェア $W_i (i = 0, \ldots, n_p)$ は $n^{'}$ 個 $W_{ij} (j = 0, \ldots, n^{'} - 1)$ のパーツ の連結と考える．ターゲットデータ M は $M_1, \ldots, M_{n^{'}}$ の連結で各データ長を d ビット，$M_0 \in \{0\}^d$ とする．また M_0 と同じサイズのダミーデータ $R_i (i = 0, \ldots, n_p)$ をランダムに選択する．このとき W_{ij} を以下のようにおく．

- $W_{00} = R_0$
- $W_{0j} = M_1 \oplus M_{j+1} \oplus R_j \quad (j = 1, \ldots, n^{'} - 1)$
- $W_{10} = M_1 \oplus R_0$

- $W_{1j} = W_{0,j-1} \oplus R_{j-1} \oplus R_j$ $(j = 1, \ldots, n^{'} - 1)$
- $W_{ij} = W_{i-1,j-1} \oplus R_{j-1} \oplus R_j$ $(i = 2, \ldots, n^{'} - 1, j = 1, \ldots, n^{'} - 1)$
- $W_{n^{'},j} = M_2 \ \oplus, \ldots, \oplus \ M_{n^{'}} \oplus R_j$ $(j = 1, \ldots, n^{'} - 1)$

上記構成方式に基づいて構成される XOR-$(2,4)$-SSS with $n^{'} = 2 \neq n - 1 = 3$ は以下の通りである．

Example3 (XOR-$(2,4)$-SSS [7]) $M = M_1 || \ M_2$ $(n^{'} = 2), M_0 \in \{0\}^d$

W_0	$M_0 \oplus R_0$	$M_1 \oplus M_2 \oplus R_1$
W_1	$M_1 \oplus M_2 \oplus R_0$	$M_1 \oplus R_1$
W_2	$M_1 \oplus R_0$	$M_0 \oplus R_1$
W_3	$M_2 \oplus R_0$	$M_2 \oplus R_1$

$F()$ を複数のシェアを入力すると，各パートで復元されるデータを出力する関数と考える．例えば $F(\{W_0, W_1\}) = \{M_1 \oplus M_2$ (左パート) $, M_2$ (右パート) $\}$ となり結果的に M_1, M_2 の両方を復元することが可能となる．以下他のケースについても

- $F(\{W_0, W_2\}) = \{M_1, M_1 \oplus M_2\}$,
- $F(\{W_0, W_3\}) = \{M_2, M_1\}$,
- $F(\{W_1, W_2\}) = \{M_2, M_1\}$,
- $F(\{W_1, W_3\}) = \{M_1, M_1 \oplus M_2\}$,
- $F(\{W_2, W_3\}) = \{M_1 \oplus M_2, M_2\}$.

となり M_1, M_2 の両方を復元することがわかる．

さらに $n_p = 5$ のときに構成される XOR-$(2,6)$-SSS with $n^{'} = 4$ は以下の通りである．ただし $M_{234} := M_2 \oplus M_3 \oplus M_4$ とおく．

Example4 (XOR-$(2,6)$-SSS [7])

W_0	$M_0 \oplus R_0$	$M_1 \oplus M_2 \oplus R_1$	$M_1 \oplus M_3 \oplus R_2$	$M_1 \oplus M_4 \oplus R_3$
W_1	$M_1 \oplus R_0$	$M_0 \oplus R_1$	$M_1 \oplus M_2 \oplus R_2$	$M_1 \oplus M_3 \oplus R_3$
W_2	$M_1 \oplus M_4 \oplus R_0$	$M_1 \oplus R_1$	$M_0 \oplus R_2$	$M_1 \oplus M_2 \oplus R_3$
W_3	$M_1 \oplus M_3 \oplus R_0$	$M_1 \oplus M_4 \oplus R_1$	$M_1 \oplus R_2$	$M_0 \oplus R_3$
W_4	$M_1 \oplus M_2 \oplus R_0$	$M_1 \oplus M_3 \oplus R_1$	$M_1 \oplus M_4 \oplus R_2$	$M_1 \oplus R_3$
W_5	$M_{234} \oplus R_0$	$M_{234} \oplus R_1$	$M_{234} \oplus R_2$	$M_{234} \oplus R_3$

2.3 XOR-$(2,n)$-SSS における同型性の導入

上記例のようにシェアを表現するマトリクス $\{W_{ij}\}$ に対して，次のように同型性を定義する．

Definition5 (XOR-$(2,n)$-SSS における同型性) XOR-$(2,n)$-SSS Ψ の W_{ij} 成分がマトリクス表現されているとき，以下の操作に基づいて変形されたマトリクスから生成される XOR-$(2,n)$-SSS は Ψ と同型であるとする．

(1) ある行を他の行と入れ替える
(2) ある列を他の列と入れ替える
(3) ある列の全てのサブシェアのそれぞれに対して同じデータを XOR で足し込む

(1) については配布されるシェアの index が変更したのみであり (2) についてはシェアの各パートの順番が変更されたに過ぎず配布されるデータとしては同一である．(3) については足し込むランダムデータ R_i に変化が生じたに過ぎず，選択されたランダムデータが異なるだけであると解釈できる．つまり，上記操作を行ったとしても安全性を確保しつつ，別の異なる XOR-$(2,n)$-SSS を構成することが可能である．ただしシェア生成時の効率性については増減する場合があることは自明である．

次に Example1 をもとに実際の変形例を見ることにする．

Example6 (Example1 の変形例)

W_0	$M_0 \oplus R_0$	$M_0 \oplus R_1^{'}$
W_1	$M_2 \oplus R_0$	$M_1 \oplus R_1^{'}$
W_2	$M_1 \oplus R_0$	$M_1 \oplus M_2 \oplus R_1^{'}$

上記例はサブシェア $W_{t1}(t = 0, \ldots, 2)$ に M_1 を足しこんだデータと同一である．ランダムデータとして R_1 ではなく $R_1^{'}$ と記載しているがランダムデータの選択には依存しないため以降は R_i と記載しても問題ない．次に Example3 を変形して Example1 との拡張性を見い出す．下記例は Example3 においてサブシェア $W_{t1}(t = 0, \ldots, 2)$ に $M_1 \oplus M_2$ を足しこんだデータである．

Example7 (Example3 の変形例)

W_0	$M_0 \oplus R_0$	$M_0 \oplus R_1$
W_1	$M_1 \oplus M_2 \oplus R_0$	$M_2 \oplus R_1$
W_2	$M_1 \oplus R_0$	$M_1 \oplus M_2 \oplus R_1$
W_3	$M_2 \oplus R_0$	$M_1 \oplus R_1$

これを Example6 と比較すると Example7 では W_1 が新たに追加された拡張方式であることがわかる．

XOR-SSS をシェアを表現する方式として上記例のマトリクス表現ではなく，各サブシェア W_{ij} を次のように $\mathbb{Z}_2^{n^{'}}$ の元として表現することとする．$W_{ij} = \bigoplus_{t=1}^{n^{'}} \alpha_t M_t$ のとき $w_{ij} = (\alpha_1, \ldots, \alpha_{n'}) \in \mathbb{Z}_2^{n^{'}}$ とベクトル表現を行う．このとき各列に同じように出現するランダムデータのパート R_i およびゼロデータ M_0 は無視しても一般性を失わない．

以下例は Example6 をベクトル表現に変換したものである．

Example8 (Example7 のベクトル表現)

W_0	$(0,0)$	$(0,0)$
W_1	$(1,1)$	$(0,1)$
W_2	$(1,0)$	$(1,1)$
W_3	$(0,1)$	$(1,0)$

3. 2-伝播基底集合の定義と XOR-$(2, 2^m)$-SSS への展開

前章で取り上げた XOR-(2,4)-SSS のトイケースを Example8 のようにベクトル表現したとき，各列のベクトルが $n^{'}$ 次元ベクトル空間を構成していることが分かる．具体的には Example8 において $w_{10} = w_{20} + w_{30}, w_{11} = w_{21} + w_{31}$ を満たしている．ここで + は $\mathbb{Z}_2^m$ 上の加算を意味する．

この例に見られるように m 次ベクトル空間から XOR-$(2, 2^m)$-SSS を構成する可能性について本章にて議論していく．その準備のためはじめにいくつかの定義を行う．

$\mathbb{Z}_2^m$ を張る基底 $b = b^{(1)}, b^{(2)}, \ldots, b^{(m)}$ を考える．ここで $b^{(i)}$ は m 次元ベクトルである．このとき $\mathbb{Z}_2^m$ の元は $\sum_{i=1}^{m} \alpha_i b^{(i)}$ for $\alpha_i \in \mathbb{Z}_2$ と表現できる．

Definition9 (基底の和) 2 つの基底 b_i, b_j に対して基底の和を次のように定義する．$b_i + b_j := \{b_i^{(1)} + b_j^{(1)}, b_i^{(2)} + b_j^{(2)}, \ldots, b_i^{(m)} + b_j^{(m)}\}$．ここで + は $\mathbb{Z}_2^m$ 上の加算を意味する．

Remark10 (XOR 演算と $\mathbb{Z}_2^m$ 上の加算の関係) 2 つの m 次元ベクトル $b^{(i)}, b^{(j)} \in \mathbb{Z}_2^m$ の $\mathbb{Z}_2^m$ 上の加算はビットごとの排他的論理和と同一視できる．例：$(0,0,1,1) + (0,1,0,1) = (0,1,1,0) \in \mathbb{Z}_2^4$.

3.1 2-伝播基底集合の定義とその性質

Definition11 (2-伝播基底集合) $\mathbb{Z}_2^m$ を張る基底 $\{b_i\}(i = 1, \ldots, l)$ の集合が以下の条件を満たすとき $\{b_i\}$ を 2-伝播基底集合という：b_1 は全てゼロベクトルで構成される（便宜上 b_1 も基底と同一視する）．任意の異なる 2 つの基底 b_i, b_j に対して $b_i + b_j$ が $\mathbb{Z}_2^m$ の基底となる．

Theorem12 (2-伝播基底集合の位数) $\mathbb{Z}_2^m$ 上の 2-伝播基底集合 $\{b_i\}$ の位数は最大 2^m である．

Proof. 2-伝播基底集合 $\{b_i\}$ に $2^m + 1$ 以上の元が存在した場合，$b_i^{(0)} = b_j^{(0)}$ となるベクトルが存在する．このとき基底の和 $b_i + b_j$ を考えると $(b_i + b_j)^{(0)}$ はゼロベクトルであり定義である「$(b_i + b_j)$ は $\mathbb{Z}_2^m$ の基底である」という事実に反する．ゆえに $\mathbb{Z}_2^m$ 上の 2-伝播基底集合 $\{b_i\}$ の位数は高々 2^m である． ∎

Definition13 (基底 b と基底行列 B) $\mathbb{Z}_2^m$ 上のベクトル集合 $b = b^{(1)}, b^{(2)}, \ldots, b^{(m)}$ に対して行ベクトルで構成された $m \times m$ 行列 B について，実数上の行列と同様に行列の階数 $(rank)$ を定義可能である．つまり線形独立な行ベクトルの個数を $rank(B)$ と記載する．

Remark14 2-伝播基底集合 $\{b_i\}$ の各基底に対する基底行列を $\{B_i\}$ とすると，$rank(B_1) = 0$ （b_1 はゼロベクトルの集合のため），$rank(B_i) = m (i = 2, \ldots, l)$, $rank(B_i + B_j) = m (i, j = 1, \ldots, l)$（ただし $i \neq j$）を満たす．

Theorem15 2-伝播基底集合 $\{b_i\}$ が存在すれば，任意の $\alpha_i \in \mathbb{Z}_2$ に対する $\sum_{i=1}^{m} \alpha_i b_i$ も 2-伝播基底集合 $\{b_i\}$ に含むことができる．

Proof. 2-伝播基底集合 $\{b_i\}$ の各基底に対する基底行列を $\{B_i\}$ とする．このとき，任意の $\alpha_i \in \mathbb{Z}_2$ に対して $rank(\sum_{i=1}^{m} \alpha_i Bi) = m$ であることを示せばよい．基底行列 B_i に対して 基底行列 B_j を足す操作は行列の $rank$ が変わらないことから行列への基本操作が行われている，すなわち B_i に対して両側から行列が掛けられていることを意味する．つまり $B_i + B_j = L_j B_i R_j$ となる 2 行列 L_j, R_j が存在する．一方で $B_j + B_i = L_i B_j R_i$ を満たす． $B_i + B_j = B_j + B_i$ であることから式変形を行うと $B_i = (L_j^{-1} L_i) B_j (R_i R_j^{-1})$ となり $B_i = L^{'} B_j R^{'}$ となる 2 行列 $L^{'}, R^{'}$ が存在する．よって $\sum_{i=1}^{m} \alpha_i B_i = B_{i_1} + B_{i_2} + \ldots, B_{i_t} = L_{i_2} B_{i_1} R_{i_2} + B_{i_3} + \ldots + B_{i_t} = \ldots = L B_{i_1} R$ となる 2 行列 L, R が存在し $rank(B_{i_1}) = m$ であることが分かる． ∎

Lemma16 $\mathbb{Z}_2^m$ 上の 2-伝播基底集合 $\{b_i\}$ の位数は 2^t という形となる．2-伝播基底集合 $\{b_i\}$ の中に t 個の生成基底 $\{c_i\}(i = 1, \ldots, t)$ を持ち各基底 b_i はこれらで張られるベクトル空間の元，つまり $b_i = \sum_{j=1}^{t} \alpha_j c_j$ を満たす．

Remark17 $\mathbb{Z}_2^m$ 上の 2-伝播基底集合 $\{b_i\}$ の位数が 2^m の optimal な場合，任意の j に対して $\{b_i^{(j)}\}$ $(i = 1, \ldots, 2^m)$ はすべて異なるベクトルで構成される．

3.2 XOR-$(2, 2^m)$-SSS の構成

前節で定義した $\mathbb{Z}_2^m$ 上の 2-伝播基底集合 $\{b_i\}$ から XOR-$(2, 2^m)$-SSS を構成することを示す．

Theorem18 (Main Theorem) optimal な $\mathbb{Z}_2^m$ 上の 2-伝播基底集合 $\{b_i\}$ $(i = 1, \ldots, 2^m)$ が存在するとき，ベクトル表現 $\{w_{ij} = b_i^j\}$ $(i = 1, \ldots, 2^m, i = 1, \ldots, m)$ を持つ XOR-$(2, 2^m)$-SSS が存在する．

Proof. 2-伝播基底集合の定義から任意の異なる u, v に対して $b_u + b_v$ もまた基底となるため $w_1^* = w_{u1} + w_{v1}, \ldots, w_m^* = w_{um} + w_{vm}$ は $\mathbb{Z}_2^m$ 上の基底となる．相異なる 2 つのサブシェアの XOR 和 $W_u \oplus W_v$ の 第 l 成分は $\bigoplus_{s=1}^{m} w_l^{*(s)} M_s$ となる．ここで M_s に対する独立した連立方程式が m 個存在することになり $M_s (s = 1, \ldots, m)$ の全てを復元できることを意味する． ∎

3.3 2-伝播基底集合の存在性について

Theorem18 に示したように optimal な $\mathbb{Z}_2^m$ 上の 2-伝播基底集合 $\{b_i\}$ $(i = 1, \ldots, 2^m)$ の存在性を示せば XOR-$(2, 2^m)$-SSS が存在することがわかる．文献 [8] では Theorem12 に示されるように最大位数 2^m となる 2-伝播基底集合の存在性について，$m \leq 6$ に対する存在性確認シミュレーションを行っている．

具体的構成例

以下小さい位数における具体的構成例について示す．W_0

は ゼロベクトル基底に呼応し，$W_1, \ldots, W_m$ は Lemma16 の生成基底 c_i に対応する．全シェアは Theorem18 に記載の $\bigoplus_{s=1}^{m} w_l^{*(s)} M_s$ で構成される．

Example19 ($m = 3$: **XOR-**$(2, 2^3)$**-SSS**)

W_0	$(0,0,0)$	$(0,0,0)$	$(0,0,0)$
W_1	$(1,0,0)$	$(0,1,0)$	$(0,0,1)$
W_2	$(0,1,1)$	$(1,0,0)$	$(0,1,0)$
W_3	$(1,1,0)$	$(0,1,1)$	$(1,0,0)$

Example20 ($m = 4$: **XOR-**$(2, 2^4)$**-SSS**)

W_0	$(0,0,0,0)$	$(0,0,0,0)$	$(0,0,0,0)$	$(0,0,0,0)$
W_1	$(1,0,0,0)$	$(0,1,0,0)$	$(0,0,1,0)$	$(0,0,0,1)$
W_2	$(1,1,0,0)$	$(1,0,0,0)$	$(0,0,1,1)$	$(0,0,1,0)$
W_3	$(0,0,1,1)$	$(1,0,0,1)$	$(0,1,1,0)$	$(0,1,0,0)$
W_4	$(0,1,0,1)$	$(0,1,1,0)$	$(1,1,0,0)$	$(1,0,0,0)$

文献 [8] において，以下の予測をオープンプロブレムとして提示した．

予測. 任意の素数 p に対して XOR-$(2,p)$-SSS の存在性が示されていることから，optimal な $\mathbb{Z}_2^p$ 上の 2-伝播基底集合 $\{b_i\}$ $(i = 1, \ldots, p^2)$ の一般的構成方式が存在すると考えられる．

XOR-$(2, 2^m)$-SSS の利用用途を考えると XOR-$(2, 2^6)$-SSS まで存在性が示されていれば十分とも考えられる．一方で，本予測に関して，リードソロモン符号などで用いられる $\mathbb{Z}_2$ 上の既約多項式との関連性について指摘 [9] がなされた： $\mathbb{Z}_2$ の元を係数とする多項式を $\mathbb{Z}_2$ 上の既約多項式で除算した余りの集合はガロア拡大体 $GF(2^m)$ を構成し，拡大体の元が GF(2) 上のベクトル空間の基底（正規基底）は常に存在することが知られている．例えば $GF(2^4)$ の場合には $x^4 + x + 1$ が既約多項式に該当し前述の例もここから構成できる．

3.4 $k \geq 3$ への拡張

前節まで $k = 2$ のみのケースを扱ってきたが本節にて $k \geq 3$ への拡張について触れる．XOR-$(2, 2^m)$-SSS では各シェアは同じパートであれば同じランダムデータを利用していた．これは同じデータを用いることで R_* を打ち消すことにより秘密データを復元しているためである．このとき，秘密データ部の各シェアの配備は 2-伝播基底集合，もしくは既約多項式から構成される正規基底から構成されるため，ランダムデータをうまく調整して $k \geq 3$ への拡張を行う必要がある．そこで文献 [6] で用いられている巡回行列を適用することで解決できる．秘密デー秘密データ部に対して $k-1$ 個のランダムデータを配備することで同様に構成可能である．しかしこのナイーブな方式ではランダムデータを多用することから復元時の計算量を削減するなどの改善の余地があると考えられる．

4. 秘密計算法への適用の初期検討

クラウドにおける秘匿性要件は，完全性・可用性の 2 要件とは異なり多くの検討が行われている．これは，個人向けクラウド（個人情報：電子メール，画像，住所録，家計簿），企業向けクラウド（社内機密情報，顧客情報），官公庁向けクラウド（国家機密情報，住民情報）において「商用のクラウドサービス業者に安心してデータを預けられるか？」という疑問が明瞭であることに起因すると考えられる．

この不安に対する解決策として次の技術について列挙しておく．分散データを秘密にしたままデータマイニングする Privacy-preserving Data Mining [10], 検索キーを秘匿したまま（暗号化された）検索結果を取得する Searchable Encryption [11] のほか，暗号化データを（信頼していない）クラウドに計算作業を委託して計算結果を入手可能にする Gentry による Fully Homomorphic Encryption [12] はその後改良 [13] が続けられている．また岡本-高島方式 [14] は復号条件が AND，OR，NOT 演算，閾値ゲートにより構成される関係式で表現可能であり，より柔軟性の高い復号形態を持つためクラウド環境に適した暗号方式として注目されている．このように実用的なツールとして近づきつつある．

さらに近年多発する漏洩事件・事故に起因して，プライバシドリブンの考え方が定着しつつある．入力データを秘匿して秘密裏に演算を行って出力を行う秘密計算法は，CPU・クラウドのマシンパワーも向上していることから現実的な計算量で解決する方式が提案され続けている．特に，前述の Fully Homomorphic Encryption よりも軽量で，制約はあるものの複数のエンティティ（例えばクライド，サーバ，場合によりユーザ自身の PC）間の協調計算により秘密計算を行うマルチパーティプロトコルが注目されている．医療データなどよりセンシティブなデータに対しても正しくかつ安全に活用できる実験結果も常に公表され続けている．

軽量な提案方式としては例えば，千田らによる (2,3)-SSS を用いた方式 [15] [16] [17] や XOR-SSS を用いる方式 [18], 実数演算上のナイーブな提案 [19] などが挙げられる．これらに共通するのは，数値を事前に数値を分解することによる秘密分散処理を行っており，計算代行者 U_∞ に各エンティティから出された結果を集約して，利用者が計算結果を得るというトポロジーモデルである．

このとき，秘密分散後のシェアデータを持つエンティティ U_* から 計算代行者 U_∞ に送信されるデータを第三者が収集することによって，もしくは計算代行者 U_∞ の計算結果を第三者が不正に取得，さらに計算代行者 U_∞ 自身の不正などによって安全に利用できない問題が生じる．本提案はこの問題に対する解決方法についても触れる．

4.1 提案方式のアウトライン

提案方式は計算対象の入力に対して計算結果を得るために，分散エンティティ $U_i(0 \leq i \leq 2^m - 1)$ と，計算結果を集約する 計算代行者 U_∞ の協調計算を行う方式である．依頼者は入力に対して XOR-$(k, 2^m)$-SSS で秘密分散を行い，各分散エンティティに事前に配布しておくことが前提となる．計算時には，各分散エンティティが持つ複数のシェアデータから演算を行い計算代行者 U_∞ と共有する．計算代行者 U_∞ は結果を依頼者に返却するものとする．

4.2 XOR 演算（ナイーブな方式）

XOR 演算は可換であることからほぼ自明な方式となる．依頼者は入力 A と B を事前に各エンティティに分散しておき，計算時には $A \oplus B$ を計算代行者から得る方式である．具体的な例として以下の XOR-(3,4)-SSS を用いて説明する．

Example21 (XOR-(3,4)-SSS)

W_0	$M_0 \oplus R_0 \oplus R_{01}$	$M_0 \oplus R_1 \oplus R_{11}$
W_1	$M_1 \oplus M_2 \oplus R_0 \oplus R_{11}$	$M_2 \oplus R_1 \oplus R_{21}$
W_2	$M_1 \oplus R_0 \oplus R_{21}$	$M_1 \oplus M_2 \oplus R_1 \oplus R_{31}$
W_3	$M_2 \oplus R_0 \oplus R_{31}$	$M_1 \oplus R_1 \oplus R_{01}$

エンティティ U_i に対して上記の分散規則に基づいて A, B それぞれの W_i を配布する．例えば U_0 に対しては $A_0 \oplus R_0^A \oplus R_{01}^A$, $A_0 \oplus R_1^A \oplus R_{11}^A$ と $B_0 \oplus R_0^B \oplus R_{01}^B$, $B_0 \oplus R_1^B \oplus R_{11}^B$ が配布されることになる．

計算依頼が行われた際には，各エンティティにおいて A, B それぞれから生成されたシェアを XOR 演算を行い 計算代行者 U_∞ に送信する．例えば U_0 では，$(A_0 \oplus R_0^A \oplus R_{01}^A) \oplus (B_0 \oplus R_0^B \oplus R_{01}^B)$, $(A_0 \oplus R_1^A \oplus R_{11}^A) \oplus (B_0 \oplus R_1^B \oplus R_{11}^B)$ を計算することになる．これは整理すると以下のように書き換えることができる：$(A_0 \oplus B_0) \oplus R_0^{AB} \oplus R_{01}^{AB}$, $(A_0 \oplus B_0) \oplus R_1^{AB} \oplus R_{11}^{AB}$

これは $A_i \oplus B_i$ を秘密分散したときのシェア分配と同じ構造であることは明らかであり，計算代行者 U_∞ は，通常の XOR-SSS 復元処理ルーチンを用いることで $A_i \oplus B_i$ を復元することができる．

第三者および計算代行者の不正を防止する方法

同じ秘密データ部に A や B を直接利用せず，あらかじめ鍵ストリーム K を XOR 演算でマスクしておくことで本来のデータが復元できないようにすることができる．このとき A, B 両者にマスクする必要はなく，片方のみで OK である．例えばシェア生成時に $A \oplus K$, B に対するシェアを計算しておくと，計算代行者 U_∞ から返却されてくるデータは $A \oplus B \oplus K$ であり鍵ストリームで暗号化しておく効果が得られる．なお，全てのエンティティに A, B のデータに対する漏洩もなく，一様ランダムデータとして見えるため安全に処理可能である．

依頼者は最後に $A \oplus B \oplus K$ に対して K を足し合わせて結果を得る必要があるため K の鍵管理を行うデメリットは発生することになる．また，本方式は改ざんに対しては効果がないため計算処理は全てのエンティティが正しく行う semi-honest モデルでの議論と考えることができる．

4.3 加減算

入力 A, B に対して $A \pm B$ を出力することを考える．演算に用いられる元は適当な $L = 2^l$ に対して Z/LZ 上で行うことを想定する．

4.3.1 加算時の入力と出力の関係の考察

分散処理においては，秘密データ部とランダムデータ部に分類できることから簡略化のためにシェアは $A \oplus R^A$ と $B \oplus R^B$ であり，これらのデータからうまく $A \pm B$ を計算することを考える必要がある．以下入力ビットと出力をうまく対象付けた表を説明する．

Example22 (入出力対応表（加算）)

A	B	R^A	R^B	XOR	δ	OUT
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	1	0	1
0	1	0	1	1	0	1
0	0	1	0	0	1	0
0	0	1	1	0	1	0
0	1	1	0	1	0	1
0	1	1	1	1	1	1
1	0	0	0	1	1	1
1	0	0	1	1	0	1
1	1	0	0	0	1	2
1	1	0	1	0	1	2
1	0	1	0	1	0	1
1	0	1	1	1	0	1
1	1	1	0	0	0	2
1	1	1	1	0	0	2

第 1 及び 2 列は入力，第 3 から 4 列はランダム値で依頼者が持つ秘密情報と考えることができる．第 5 列は前節で計算代行者が演算可能な $A \oplus B$ を意味しており，最終列 OUT は出力で加算結果を意味する．この表から，第 3 から 4 列も計算代行者が知りえる入力値と拡大解釈を行うと全部で 16 通りの入力があり，出力を正しく得るために第 6 列のように 0,1 の値を振り分ける必要があることがわかる．この表の第 6 列を完成させるためには，以下の演算を行うことが必要となる．

$\delta := (A \oplus R^A) \vee \neg(A \oplus B) \wedge (A \oplus B) \vee \neg(B \oplus R^B)$.

ここで，この演算は計算代行者が行うことができるため，演算の正しさが保証されることに注意する．

5. まとめ

$\mathbb{Z}_2^m$ 上の基底集合に対して 2-伝播基底集合を定義し，排他的論理和を用いた $(2, 2^m)$-閾値秘密分散法の新しい構成について再考し，XOR-$(2, 2^m)$-SSS の存在性について明らかにした．また，この方式を用いて秘密計算法の自然な適用を行った．

今後の課題としてはそのほかの演算子への適用のほか，クラウド等に置かれたデータを秘匿したまま演算処理を行う他の技術として Private Set Intersection などへの適用を試みる予定である．

参考文献

[1] Shamir Adi, "How to share a secret", Communications of the ACM 22 (11): 612-613, 1979
[2] 藤井, 多田, 保坂, 栃窪, 加藤, 高速な (2, n) 閾値法の構成法とシステムへの応用, 8C-2, CSS2005.
[3] 多田, 藤井, 保坂, 栃窪, 加藤, 閾値 3 の秘密分散法の構成法, 8C-3, CSS2005.
[4] 藤井, 栃窪, 保坂, 多田, 加藤, 排他的論理和を用いた (k,n) しきい値法の構成法, ISEC2007-05.
[5] 栗原, 清本, 福島, 田中, 排他的論理和を用いた高速な (4,n) 閾値秘密分散法と (k,n) 閾値法への拡張, ISEC2007-04.
[6] J. Kurihara, S. Kiyomoto, K. Fukushima, and T. Tanaka, On a fast (k, n)-threshold secret sharing scheme, IEICE Trans. Fundamentals, vol.91-A, no.9, Sep. 2008.
[7] 須賀, 排他的論理和を用いた (k,n) 閾値秘密分散法の新しい構成とその優位性について, 1C2-4, CSS2012.
[8] 須賀, "巡回置換行列を用いず m 次元数ベクトル空間を用いて XOR 演算だけで構成可能な $(2, 2^m)$-閾値秘密分散法", 第 62 回 CSEC 研究会, 2013.
[9] 國廣, $(2, 2^m)$ 秘密分散の構成法, 2013-07-19.
[10] C.C.Aggarwal and P.S.Yu "Privacy-Preserving Data Mining: Models and Algorithms", Advances in Database Systems Series, Springer-Verlag, 2008.
[11] Hakan Hacigumus, Hakan Hacg Um Us, Bala Iyer, Chen Li,Sharad Mehrotra, "Executing SQL over Encrypted Data in the Database-Service-Provider Model", SIGMOD02, pp.216-227
[12] Craig Gentry, "Fully homomorphic encryption using ideal lattices", Annual ACM Symposium on Theory of Computing, 2009.
[13] Damien Stehle, Ron Steinfeld, "Faster Fully Homomorphic Encryption", ASIACRYPT2010.
[14] Okamoto, Takashima, "Fully Secure Functional Encryption with General Relations from the Decisional Linear Assumption", CRYPTO 2010, pp.191-208.
[15] 千田, 五十嵐, 高橋, "効率的な 3 パーティ秘匿関数計算の提案とその運用モデルの考察", 第 48 回 CSEC 研究会, 2010.
[16] 千田, 五十嵐, 高橋, "マルチパーティ計算による効率的なビット分解プロトコル", 第 50 回 CSEC 研究会, 2010.
[17] 五十嵐, 千田, 高橋, "高効率 3 パーティ秘匿関数計算の情報理論的安全性", 第 50 回 CSEC 研究会, 2010.
[18] 橋, 須賀, 岩村, "XOR を用いる秘密分散法の多値化とそれを用いた秘匿計算法", 第 65 回 CSEC 研究会, 2014.
[19] 金岡, 宮西, 韓, 北上, 佐藤, 浦野, 白鳥, "実数演算可能な軽量秘密計算法の一考察", コンピュータセキュリティシンポジウム 2014, 2E3-4, 2014.