少パーティ数の秘密分散ベース秘密計算における効率的な malicious モデル上 SIMD 計算の構成法

五十嵐　大 †　　菊池　亮 †　　濱田　浩気 †　　千田　浩司 †

† 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
180-8585 東京都武蔵野市緑町 3-9-11

ikarashi.dai@lab.ntt.co.jp

あらまし 本稿では秘密分散ベースの秘密計算において, SIMD(Single-Instruction, Multiple-Data) 計算を malicious モデル上で効率良く行う方法を提案する. SIMD とは複数のデータに対して単一の命令を並列かつ独立に実行するような計算アーキテクチャである. 統計処理, 数値計算, データマイニングなど, 多くの並列計算が SIMD で成り立っている.

提案方式は $n \geq 2k-1$ の malicous モデルで統計的秘匿性および統計的正当性を持つ. 従来の $n \geq 2k-1$ の malicious モデル上の秘密計算が体上乗算の性質を利用していたのに対して, 提案方式はランダム置換を用いて安全性を確保する. この方針により, 提案方式はデータ数 N, 回路サイズと入力サイズの和 C, セキュリティパラメータ κ として通信量 $O(\frac{\kappa}{\log N}C)$ を実現した. SCIS2013 において我々が提案した従来手法は $O(\kappa C)$ であり, $\log N$ 倍の通信量改善である.

また提案手法と従来手法の実装実験を行い, 従来手法よりも 3〜7 倍のパフォーマンスを持つことを示す.

An Efficient SIMD Protocol against Malicious Adversaries for Secure Computation Schemes Based on (k, n) Secret Sharing Schemes with Small Party Sets

Dai IKARASHI†　　Ryo KIKUCHI†　　Koki HAMADA†　　Koji CHIDA†

†NTT Secure Platform Laboratories, NTT Corporation
3-9-11, Midoricho, Musashino-shi, Tokyo 180-8585, Japan

ikarashi.dai@lab.ntt.co.jp

Abstract In this paper, we propose an efficient scheme of SIMD computation on secret-sharing-based secure computation against malicious adversaries.

Using random permutation, our scheme achieves $O(\frac{\kappa}{\log N}C)$ communication complexity, where N is the number of data, C is the size of circuit, κ is the security parameter. The best existing scheme that we showed in SCIS2013 is $O(\kappa C)$, thus, that complexity is $\log N$ times of improvement when n is small.

Furthermore, we implement and experiment our scheme and the SCIS2013 scheme, and show the performances.

1　はじめに

近年, 暗号化/秘密分散されたデータに対して処理を行う, 秘密計算の研究が盛んである. 中でも特に秘密分散ベースの秘密計算は処理効率が比較的高く, sharemind[1, 3] など大規模計算に耐えうる性能を持つ実装が現れ始めている. ビッグデータ分析の潮流の中, セキュアなデータ分

析を可能とする秘密計算の大規模対応は重要な課題である. 以降本稿で秘密計算と言えば秘密分散ベースのこととする.

1.1 背景と関連研究

既存の実装のほとんどは, 攻撃者がプロトコルに従うことを仮定する semi-honest モデルであり, 能動的な攻撃者には対抗できない. VIFF[2, 5] は能動的な攻撃者を仮定する malicious モデルにも対応するが, malicious 対応は $n \geq 3k-2$ の場合のみであり, 最小構成である $n \geq 2k-1$ には適用できないほか, 速度が数百乗算/s と適用範囲が限られる.

そのような中, 筆者らは [10] において $n \geq 2k-1$ の malicious モデル上秘密計算において, 従来の $O(\kappa n^9)$ ビットの非常に高い通信量を現実的な $O(\kappa n)$ ビットに低減する構成法を示した. (ただし κ はセキュリティパラメータ, つまり 2^κ を negligible な関数とする.)

しかし体のサイズに関して未だに課題がある. ベースとする秘密分散の体を $\mathcal{F}$, そのビット長を $\ell_{\mathcal{F}}$ とすると, semi-honest の秘密計算の通信量が最良で $O(\ell_{\mathcal{F}} n)$[6] であることから, 対象の秘密分散がもともと κ 以上のビット長の体を用いているときは $O(\ell_{\mathcal{F}} n) \fallingdotseq O(\kappa n)$ であり [10] の semi-honest に対するオーバーヘッドは小さいと言えるが, $\mathbb{Z}_2$ などの小さい体の場合には semi-honest の $O(\ell_{\mathcal{F}} n) = O(n)$ と比べ通信量が約 κ 倍となってしまうのである.

1.2 本稿の貢献

対し本稿では従来のアプローチとは異なり, ランダム置換を用いたセキュリティを実現することで, 並列計算のデータ当たり通信量を $O(\frac{\kappa n \log n}{\log N} + 2^n)$ とする. ただし N はデータ数である. n を定数と見れば $O(\frac{\kappa}{\log N})$ と, 提案方式はパーティ数が非常に小さくデータ数が大きい場合に適する方式であり, そのような場合には $\log N$ 倍の改善となる.

また実装実験で, $k=2, n=3, N=1,000,000$ で 3〜7 倍程度の改善を実現することを示す.

2 記法

平文空間を $\mathcal{R}$ とする.

- 関数 $f : \mathcal{R} \to \mathcal{R}'$ に対して, $f^N : \mathcal{R}^N \to \mathcal{R}'^N$ を f の並列実行とする. すなわち

$$f^N(a_0, \cdots, a_{N-1}) = (f(a_0), \cdots, f(a_{N-1}))$$

- 環 $\mathcal{R}$ に対して $0_{\mathcal{R}}$ を $\mathcal{R}$ の零元とする.
- X^m の元 x に対して, i 番目の要素を x_i と書く.
- $x \in X^m, y \in X^{m'}$ に対して, 結合

$$(x_0, \cdots, x_{m-1}, y_0, \cdots, y_{m'-1}) \in X^{m+m'}$$

を $x \parallel y$ と書く.

- $x \in (X^m)^N, y \in (X^{m'})^N$ に対して, 垂直結合

$$(x_0 \parallel y_0, \cdots, x_{N-1} \parallel y_{N-1})$$

を $x \parallel_{\mathrm{V}} y$ と書く.

3 準備

本節では準備として提案手法の構成要素のうち既存の要素を説明する.

3.1 ランダム置換プロトコル

提案手法の構成要素に, malicious モデルで動作するランダム置換がある.

秘密分散ベースのランダム置換には幾つかの方式がある. パーティ数が小さい場合には [13], [8] の, 再分散をベースとする方式が効率的である. [8] には malicious 版も提案されている他, [10] の構成法により semi-honest 版を malicious 対応に変換することができる.

また [8] には行列ベースの手法も提案されているが, データ並列数 N に対する通信量が少なくとも $O(N^2)$ であり大規模計算には向かない.

3.2 複製秘密分散

提案手法は一般の線形秘密分散ベースの秘密計算に適用されるが, 特に適用を想定する秘密分散は $\mathbb{Z}_2$ 上の複製秘密分散 [4] である. $\mathbb{Z}_2$ は任意の演算を行うための最小の構造であり, 通信効率が最良であるからである.

複製秘密分散はパーティ数に対して符号化効率が指数的に低下するが, 少パーティ数では大きな問題とはならない.

なお Shamir の秘密分散など符号化効率の良い秘密分散方式では $\mathbb{Z}_2$ 上線形秘密分散を行うことができない. 例えば Shamir の秘密分散はビット長 $\ell_{\mathcal{F}} \geq n+1$ なる体 $\mathcal{F}$ が必要であり $\mathbb{Z}_2$ は不適である.

3.3 改ざん模倣可能性 (お詫び含む)

文献 [10] で誤った記述をしたためここでお詫びする. 文献 [10] の適用対象は, 任意の semi-honest のプロトコルではなく, これから記述する, **改ざん模倣可能性**を持つような semi-honest プロトコルである. 本稿の提案手法も適用対象は同じであり, そのような semi-honest プロトコル群を malicious モデル上で統計的秘匿性および正当性を持つ方式に変換する.

定義 3.1 改ざん模倣可能性
秘密分散値を入力として秘密分散値を出力するプロトコル Π が改ざん模倣可能とは以下のことを言う.

任意の攻撃者 $\mathcal{A}$ と, 任意の *honest* な k パーティの組 $\mathcal{P}$ を考える. また Π の入力は不正な入力 *(*すなわち秘密分散においては, 復元するパーティの組によって復元結果が異なるような秘密分散値*)* ではないとする. このとき Π の正しい出力の平文を $a \in \mathcal{R}$ とすると, $\mathcal{P}$ による復元結果はある改ざん値 x が存在して $a+x \in \mathcal{R}$ と表される. このときある模倣者 $\mathcal{B}$ が存在して, $\mathcal{B}$ が Π の実行無しに x を出力できるならば, Π は改ざん模倣可能である.

紙面の都合上証明しないが, [7, 6, 11] 等, 知る限り全ての semi-honest 乗算プロトコルは改ざん模倣可能である.

ただしランダム置換に関しては実は [13], [8] の再分散ベースのランダム置換はそのままでは改ざん模倣可能ではない. しかし, 当該手法が “再分散” プロトコルの繰り返しであり, その再分散プロトコルが改ざん模倣可能なため, プロトコルごとに行うチェックサム保存を, 再分散プロトコルの単位で行うように変更すればよい.

4 提案手法

提案手法はランダム化フェーズ (Scheme 1), 計算フェーズ (Scheme 2), 正当性証明フェーズ (Scheme 3) の 3 フェーズに分かれて順に行われる.

4.0.1 フェーズ 1 : ランダム化フェーズ

Scheme 1 にランダム化フェーズを示す. ダミーレコード列を追加して, 正当性を持つランダム置換を行う. ランダム置換は ν 回行われる. パラメータ ν は, $\nu = \lceil \kappa / \log N \rceil$ 程度の整数である. なお公開値の分散は, 各パーティが乱数成分固定として自分のシェアを作成すればよく, オフライン処理である.

4.0.2 フェーズ 2 : 計算フェーズ

Scheme 2 に計算フェーズを示す. 計算フェーズでは, 非ランダム化系統, ランダム化系統, ダミーレコード列の 3 系統で所望の関数 F を並列に計算する. その際, 出力は全てチェックサムとして保存される.

4.0.3 フェーズ 3 : 正当性証明フェーズ

Scheme 3 に正当性証明フェーズを示す. 正当性証明フェーズでは計算フェーズで保存されたチェックサムを元に正当性を検証する. SYNC は, その時点までの全ての受信すべきデータを受信したことを他の全パーティに伝える信号を送信し, また他の全パーティからの当該信号を受信する処理である. これが確認できるまでは後続の処理を行わないことで非同期ネットワークでのセキュリティを担保する.

4.1 セキュリティ

定理 4.1 *malicious* モデルにおいて, 提案手法は統計的秘匿性を持つ.

また, *semi-honest* モデルにおいて, 提案手法は完全秘匿性を持つ.

定理 4.2 *malicious* モデルにおいて, 提案手法は統計的正当性を持つ. すなわち出力が不正となる

Scheme 1 [プロトコル] ランダム化フェーズ

パラメータ: 計算すべき m 入力 μ 出力関数 F, レコード数 N, 挿入するダミーレコード数 $|D|$, ランダム化入力数 $\nu \in \mathbb{N}$

入力: 非ランダム化入力 $[\![A]\!] \in ([\![\mathcal{R}]\!]^m)^N$

出力: ランダム置換 $[\![\pi_0]\!], \cdots, [\![\pi_{\nu-1}]\!] \in [\![\Pi^{N+|D|}]\!]$ ランダム化入力 $[\![B_0]\!] = [\![\pi_0(A \parallel D)]\!], \cdots, [\![B_{\nu-1}]\!] = [\![\pi_{\nu-1}(A \parallel D)]\!] \in ([\![\mathcal{R}]\!]^\mu)^{N+|D|}$, ダミーレコード列 $D \in (\mathcal{R}^m)^{|D|}$,

1: 公開値から成るダミーレコード列 $D \in (\mathcal{R}^m)^{|D|}$ を作成する. 内容は $F^{|D|}$ の定義域内で任意である.
2: D を正当性を持つ方法で秘匿し, $[\![D]\!] \in ([\![\mathcal{R}]\!]^m)^{|D|}$ を得る.
3: 入力にダミーレコード列を結合し $[\![A \parallel D]\!] := [\![A]\!] \parallel [\![D]\!]$ を得る.
4: **for each** $i < \nu$ **do**
5: $\quad [\![A \parallel D]\!]$ に正当性をもつランダム置換を施して $[\![B_i]\!] := [\![\pi_i(A \parallel D)]\!]$ を得, 出力する. この際利用した秘密のランダム置換 $[\![\pi_i]\!] \in [\![\Pi]\!]$ も出力する.
6: $\quad D$ を出力する.

Scheme 2 [プロトコル] 計算フェーズ

パラメータ: 計算すべき m 入力 μ 出力関数 F, レコード数 N, 挿入するダミーレコード数 $|D|$, ランダム化入力数 $\nu \in \mathbb{N}$

入力: 非ランダム化入力 $[\![A]\!] \in ([\![\mathcal{R}]\!]^m)^N$, ランダム化入力 $[\![B_0]\!], \cdots, [\![B_{\nu-1}]\!] \in ([\![\mathcal{R}]\!]^\mu)^{N+|D|}$, ダミーレコード列 $D \in (\mathcal{R}^m)^{|D|}$

出力: 出力 $[\![F^N(A)]\!] \in ([\![\mathcal{R}]\!]^\mu)^N$, $\nu + 2$ 個のチェックサム $[\![C_A]\!], [\![C_{B_0}]\!], \cdots, [\![C_{B_{\nu-1}}]\!], [\![C_D]\!]$

1: チェックサム $[\![C_A]\!] := \emptyset \in ([\![\mathcal{R}]\!]^0)^N$, $[\![C_{B_0}]\!] := \emptyset \in ([\![\mathcal{R}]\!]^0)^{N+|D|}, \cdots, [\![C_{B_{\nu-1}}]\!] := \emptyset \in ([\![\mathcal{R}]\!]^0)^{N+|D|}$, $[\![C_D]\!] := \emptyset \in ([\![\mathcal{R}]\!]^0)^{|D|}$, ただし $\emptyset$ は空ベクトルである.
2: $[\![A]\!], [\![B_0]\!], \cdots, [\![B_{\nu-1}]\!], D$ の $\nu + 2$ 系統に対して semi-honest の秘密計算 (D については平文の計算) により所望の関数 F を, サブプロトコル $f_i : [\![\mathcal{R}]\!]^{m_i} \to [\![\mathcal{R}]\!]^{\mu_i}$ ごとに以下のチェックサム更新を行いながら計算し, $[\![F^N(A)]\!] \in ([\![\mathcal{R}]\!]^\mu)^N$ を出力する. ただし m_i, μ_i はそれぞれ f_i の入力数, 出力数である.
3: [チェックサム更新]
4: $\quad f_i$ への $\nu + 2$ 系統の各出力を $[\![A']\!] \in ([\![\mathcal{R}]\!]^{\mu_i})^N$, $[\![B'_0]\!], \cdots, [\![B'_{\nu-1}]\!] \in ([\![\mathcal{R}]\!]^{\mu_i})^{N+|D|}$, $D' \in (\mathcal{R}^{\mu_i})^{|D|}$ とおく.
5: $\quad D'$ を正当性を持つ方法で秘匿し, $[\![D']\!]$ を得る.
6: $\quad [\![C_A]\!] := [\![C_A]\!] \parallel_{\mathrm{V}} [\![A']\!]$ を出力する.
7: $\quad$ **for each** $i < \nu$ **do**
8: $\quad\quad [\![C_{B_i}]\!] := [\![C_{B_i}]\!] \parallel_{\mathrm{V}} [\![B'_i]\!]$ を出力する.
9: $\quad [\![C_D]\!] := [\![C_D]\!] \parallel_{\mathrm{V}} [\![D']\!]$ を出力する.

Scheme 3 [プロトコル] 正当性証明フェーズ
パラメータ: 計算すべき m 入力 μ 出力関数 F, レコード数 N, 挿入するダミーレコード数 $|D|$, ランダム化入力数 $\nu \in \mathbb{N}$, 分割単位 $\sigma \in \mathbb{N}$
入力: 3 つのチェックサム $[\![C_A]\!], [\![C_B]\!], [\![C_D]\!]$, ランダム置換 $[\![\pi_0]\!], \cdots, [\![\pi_{\nu-1}]\!] \in [\![\Pi^{N+|D|}]\!]$
出力: 改ざんがあれば $\perp$, なければ $\top$

1: SYNC
2: **for each** $i < \nu$ **do**
3: $[\![\pi_i]\!]$ を公開し π_i を得る.
4: $[\![\zeta_i]\!] := [\![C_{B_i}]\!] - ([\![C_A]\!] \parallel [\![C_D]\!]) \in ([\![\mathcal{R}]\!]^M)^{N+|D|}$ を計算する. なお M は $[\![C_A]\!]$(または $[\![C_{B_i}]\!]$, どちらでも等しい) の 1 レコード当たりの要素数とする.
5: $[\![\zeta_i]\!]$ の各レコードを, σ 要素ごとに垂直分割する. $([\![\mathcal{R}]\!]^\sigma)^{N+|D|}$ の要素が $M' = \lceil M/\sigma \rceil$ 個できあがる. なお最後の分割に端数が現れる場合があるが, σ 要素に満たない部分は 0 パディングする. 出来上がったデータを $([\![\zeta'_i]\!] \in ([\![\mathcal{R}]\!]^\sigma)^{(N+|D|)M'}$ とおく.
6: 乱数の分散値 $[\![\rho_i]\!] \in ([\![\mathcal{R}]\!]^\sigma)^{(N+|D|)M'}$ を生成する.
7: 積和プロトコルにより $[\![\varphi]\!] = \sum_{i<\nu} [\![\rho_i]\!][\![\zeta'_i]\!]$ を計算する.
8: SYNC
9: $[\![\varphi]\!]$ を公開して φ を得る.
10: 各パーティは $\varphi = 0$ を確認し, 真ならば $\top$, 偽ならば $\perp$ となる検証結果を他の全パーティに送信する.
11: 各パーティは, 全パーティからの検証結果で $\perp$ が存在すれば $\perp$, そうでなければ $\top$ を出力する.

確率は高々, パラメータ ν と σ に対して *negligible* な以下の確率である.

$$\frac{1}{(N+|D|)^\nu} + \frac{1}{|\mathcal{R}|^\sigma} - \frac{1}{(N+|D|)^\nu |\mathcal{R}|^\sigma} \qquad (1)$$

4.1.1 計算フェーズまでの秘匿性

ランダム化フェーズ, 計算フェーズは無条件秘匿性を持つプロトコルだけから成るため, やはり無条件秘匿性を持つ.

4.1.2 正当性

ランダム化フェーズの処理は正当性を満たすシャッフルのみであるから, ランダム化フェーズは正当性を持つ. よってここでは計算フェーズの処理の正当性が, 正当性証明フェースにおけるチェックで保証されることを示す. そのために重要となるのが, 改ざん模倣可能なサブプロトコルの正当性検証可能性を述べる以下の補題である.

補題 4.1 $\mathcal{R}, \mathcal{R}'$ を環とし, f を $f : \mathcal{R} \to \mathcal{R}'$, レコード数とダミーレコード数 $N, |D|$ を正整数, π を $N + |D|$ 要素の一様ランダム置換, 非ランダム化入力ベクトル A を $\mathcal{R}^N$ 上, またランダム化入力ベクトル B を $\mathcal{R}^{N+|D|}$ 上の任意の確率変数, ダミーレコードベクトル D を任意の $\mathcal{R}^{|D|}$ の元, 改ざん値 $X \in \mathcal{R}^N, Y \in \mathcal{R}^{N+|D|}$ を π と独立な確率変数とする.

このとき, $B = \pi(A \parallel D)$ が成り立つならば, $g : A \mapsto f^N(A) + X, h : B \mapsto f^{N+|D|}(B) + Y$ として以下が成り立つ.

$$\Pr[g(A) \parallel f^{|D|}(D) = \pi^{-1}(h(B)) \mid X \neq 0_{\mathcal{R}^N}] \leq \frac{1}{N+|D|} \qquad (2)$$

証明 式 (2) 中の等式 $g(A) \parallel f^{|D|}(D) = \pi^{-1}(h(B))$ の成立確率を評価するため, まず $\pi^{-1}(h(B))$ について考える. 置換とレコード毎の演算は順序交換可能であり, 以下が成り立つ.

$$\begin{aligned} \pi^{-1}(h(B)) &= \pi^{-1}(f^{N+|D|}(B) + Y) \\ &= \pi^{-1}(f^{N+|D|}(B)) + \pi^{-1}(Y) \end{aligned} \qquad (3)$$

次に仮定より $B = \pi(A \parallel D)$ であるから以下の変形ができる.

$$\pi^{-1}(f^{N+|D|}(B)) = \pi^{-1}(f^{N+|D|}(\pi(A \parallel D))) \qquad (4)$$

$f^{N+|D|}$ は f のレコード毎の適用であるからやはり置換と順序交換可能であり, π を打ち消すこと

ができる.

$$\begin{aligned} & \pi^{-1}(f^{N+|D|}(\pi(A \parallel D))) \\ = \quad & f^{N+|D|}(A \parallel D) = f^{N}(A) \parallel f^{|D|}(D) \end{aligned} \quad (5)$$

よって式 (*3*), 式 (*4*), 式 (*5*) より $\pi^{-1}(h(B))$ は以下のように変形される.

$$\pi^{-1}(h(B)) = f^{N}(A) \parallel f^{|D|}(D) + \pi^{-1}(Y) \quad (6)$$

次に, $g(A)$ は以下のように変形されるので,

$$\begin{aligned} g(A) &= (f^{N}(A) + X) \parallel f^{|D|}(D) \\ &= f^{N}(A) \parallel f^{|D|}(D) + X \parallel 0_{\mathcal{R}^{|D|}} \end{aligned}$$

式 (*6*) と合わせ, f が打ち消されて以下の式を得る.

$$g(A) - \pi^{-1}(h(B)) = X \parallel 0_{\mathcal{R}^{|D|}} - \pi^{-1}(Y) \quad (7)$$

ここで $X \parallel 0_{\mathcal{R}^{|D|}} - \pi^{-1}(Y)$ は $\mathcal{R}$ 上 $N + |D|$ 次ベクトルだから, $X \parallel 0_{\mathcal{R}^{|D|}} - \pi^{-1}(Y) = 0_{\mathcal{R}^{N+|D|}}$ は以下の条件と等価である.

$$\begin{aligned} &[\text{すべての } i < N \text{ に対して} \\ &y_{\pi^{-1}(i)} = x_i, \text{かつ } y_{\pi^{-1}(N)} = 0] \end{aligned} \quad (8)$$

ここで X の N 個の成分のうち非零である個数を ℓ とおく. 式 (*2*) の確率は $X \neq 0_{R^N}$ の条件付き確率であるから, $1 \leq \ell \leq N$ で考えてよい. また Y の成分のうち非零である個数を ℓ' とする. このとき $\ell = \ell'$ かどうかで場合分けして式 (*2*) の確率を考える.

(i) $\ell \neq \ell'$ のとき

このとき, $X \parallel 0_{\mathcal{R}^{|D|}}$ と $\pi^{-1}(Y)$ で非零である個数がそもそも異なるため式 (*8*) を満たすことは無く, 式 (*2*) の確率は 0 である.

(ii) $\ell = \ell'$ のとき

このとき式 (*2*) を満たすためには, π^{-1} による Y の置換後, X と $\pi^{-1}(Y)$ の非零成分の位置が一致することが必要条件 (十分条件でもあるのは $\mathcal{R} = \mathbb{Z}_2$ のときに限ることに注意) であり, その確率は以下の組み合わせ数である.

$$\frac{1}{{}_{N+|D|}\mathrm{C}_{\ell}}$$

$1 \leq \ell \leq N < N + |D|$ と組み合わせ数の性質より,

$${}_{N+|D|}\mathrm{C}_{\ell} \geq {}_{N+|D|}\mathrm{C}_{N+|D|-1} = {}_{N+|D|}\mathrm{C}_{1} = N + |D|$$

であり, 式 (*2*) の確率が $1/(N + |D|)$ 以下であることが分かる. $\ell \leq N + |D|$ でなく $\ell \leq N$ であるのが, 公開値として計算され改ざんできない $|D|$ 個のダミーレコードのおかげであることに注意.

以上 (i), (ii) よりどちらの場合も式 (*2*) の確率は $1/(N + |D|)$ 以下であり, 式 (*2*) が満たされることが証明された.

□ 補題 *4.1*

補題は計算フェーズにおける改ざん模倣可能なサブプロトコル f の実行に関して, 入力が正当すなわち $B = \pi(A \parallel D)$ ならば, 非ランダム化入力に対する出力 $g(A)$, ランダム化入力に対する出力 $h(B)$ を用いて $g(A) \parallel f(D) = \pi^{-1}(h(B))$ をチェックすることで改ざんが検知されることを示している.

補題が述べているのは改ざん模倣可能なサブプロトコル 1 回の実行の正当性の検証可能性のみである. しかし全ての改ざん模倣可能なサブプロトコルについて検証を行うことで, 計算フェーズ全体の正当性も保証される. なぜなら, ランダム化フェーズで用いられるランダム置換が正当性を持つことから, 計算フェーズの最初の入力は正当である. よって攻撃者が改ざんを行ったならば, 少なくとも最初の改ざんを行ったサブプロトコルについて補題の仮定を満たす. すなわち式 (2) が満たされるようなサブプロトコルが必ず存在する.

さて, ここまでは分散値の中身の平文に関する正確な議論を行った. 正当性証明フェーズでは実際に $g(A) \parallel f(D) = \pi^{-1}(h(B))$ を全サブプロトコルに関してチェックしなくてはならない. 正当性証明フェーズの各 $[\![\rho_i]\!][\![\zeta'_i]\!]$ は, $\zeta'_i \neq 0$ のとき一様乱数となるため, その和である $[\![\varphi]\!]$ は, 0 でない ζ'_i が一つでも存在すれば一様乱数である. よって $\mathcal{R}$ が体のとき, [10] に書いたように $[\![\mathcal{R}]\!]^{\sigma}$ 上の乗算を σ 次拡大体乗算とすれば, 改ざん成功確率は高々以下である.

$$\frac{1}{(N + |D|)^{\nu}} + \frac{1}{|\mathcal{R}|^{\sigma}} - \frac{1}{(N + |D|)^{\nu}|\mathcal{R}|^{\sigma}}$$

4.1.3 全体の秘匿性

最後の $[\![\varphi]\!]$ の公開以外については無条件秘匿性を持つプロトコルの組み合わせであるから気にする必要はない. 最後の $[\![\varphi]\!]$ では, 攻撃者が改ざんを行った際に, 改ざんの成否に情報を含め

る攻撃を行う場合があるかもしれない. 例えば攻撃者があるビット x を知りたいときに, 命題 $[\zeta_i = 0_{(\mathcal{R}^M)^{N+|D|}}]$ を x と等しくするような攻撃である. このようなとき, x の値は改ざんの成否と等しい.

しかし, 計算フェーズにおけるサブプロトコルがすべて改ざん模倣可能であること, 全てチェックサムに出力が記録されていること, 計算フェーズの入力が正当であることから, $\zeta_i \neq 0_{(\mathcal{R}^M)^{N+|D|}}$ であるならば, 改ざんされた成分, すなわち ζ_i の非零である $\mathcal{R}$ 要素の中に攻撃者が模倣できる値が存在する. よって攻撃者が改ざんを行えば, negligible である式 (1) 以下の確率を除いて $\varphi \neq 0$ となるから, 提案方式は統計的秘匿性を満たす.

4.2 性能

4.2.1 通信量

ランダム化フェーズ ランダム化フェーズは正当性を持つランダム置換だけが通信を伴う. [13], [8] の再分散ベースの手法は, パーティ数が小さい場合は効率的であり $O(\ell_{\mathcal{R}} N)$ ビット, データ当たりで $O(\ell_{\mathcal{R}})$ ビットである.

正当性を持たせるためには上記ランダム置換に [10] の構成法を合わせるのが最も効率的である. なお $\mathcal{R}$ が小さい環であっても, [9] に書いたように体上の σ 個の秘密分散値はオフラインでその σ 次拡大体上の秘密分散値に変換できるため, m 個の入力を σ 次拡大体ずつに分割してランダム置換を行えば [10] の最大効率である, semi-honest と比べて 2 倍の通信量で済む.

各入力に対してこのランダム置換を ν 回行うので, $O(\ell_{\mathcal{R}} \nu m)$ ビットである.

計算フェーズ semi-honest における実行 $\nu + 1$ 回分の通信量であり, $O(\ell_{\mathcal{R}} \nu C)$ ビットである.

正当性証明フェーズ 通信はほぼ乱数生成のみであり, $O(\ell_{\mathcal{R}} \nu (C + m))$ である.

全体の通信量 以上より全体では, $O(\ell_{\mathcal{R}} \nu (C + m)) = O(\ell_{\mathcal{R}} \frac{\kappa}{\log N} (C + m))$ となる. 想定適用対象である $\mathcal{R} = \mathbb{Z}_2$ では, $O(\frac{\kappa}{\log N}(C + m))$ である.

4.2.2 ラウンド数

ラウンド数は, semi-honest と比べランダム化フェーズにおいて正当性を持つランダム置換の分および, 2 回の SYNC と積和, 乱数生成, π_i および φ の公開がオーバーヘッドである. SYNC を隔てない限りにおいて並列化可能であり, 例えば $(k, n) = (2, 3)$ では semi-honest と比べ高々 10 ラウンドのオーバーヘッドである.

5 実装実験

表 1 に提案手法と従来手法の性能比較を示す. 正当性を持つランダム置換には再分散ベースのランダム置換 [13, 8] と 従来手法 [10] の組み合わせを用いている. 入力は $\mathbb{Z}_2$ 上 $(2, 3)$-複製秘密分散値である. なお $m = 93, C = 59$ は [12] のビット分解 (データ長 29bit, シェア長 31bit) である. $m = 186, C = 147$ は, 2 つの値をビット分解して等号判定する処理である. $m = 93, C = 290$ は, ビット分解 5 回分の模擬回路である.

いずれも提案手法が従来手法の 8 次拡大 (改ざん成功率約 1/128, $\kappa \fallingdotseq 7$) と比較して 3 倍程度, 16 次拡大 (改ざん成功率約 1/32768, $\kappa \fallingdotseq 15$) と比較して 7 倍程度の高速化となっている. なお提案手法の $\sigma = 16$ であり, 改ざん成功率はほぼ従来手法の 16 次拡大の場合と同じである.

また semi-honest と比べ, 入力と比較して回路サイズが大きくなるほど semi-honest に近いパフォーマンスを発揮することが分かる. これは計算フェーズの通信量が $O(\ell_{\mathcal{R}} \nu C)$ であり 1 入力データ当たりのサイズ m に依存しないことによる. また最大効率である, “semi-honest の倍” よりも効率が良いのは, 非ランダム化系列とランダム化系列の計算の並列性が高く, 自動並列化機能を持つ本実装と相性が良いためと考えられる.

6 おわりに

本稿ではパーティ数の小さい秘密分散ベース秘密計算において, malicious モデルで大規模並列計算を行うための効率的な構成法を提案した. パーティ数を定数と見ると, 提案手法はデータ当たり通信量 $O(\lceil \frac{\kappa}{\log N} \rceil (C + m))$ ビットであり, 従来手法の $O(\kappa(C + m))$ に比べ $\log N$ の改善である. ただし κ はセキュリティパラメータ, N は

表 1: 提案手法と従来手法 [10] の性能比較 ($\mathcal{R} = \mathbb{Z}_2$, $(k, n) = (2, 3)$, $N = 1,000,000$)

	処理の所要時間 [ms]		
適用手法	$m = 93, C = 59$	$m = 186, C = 147$	$m = 93, C = 290$
semi-honest (参考) [11]	696	1,710	3,484
提案手法	1,794	3,523	3,945
従来手法 (8 次拡大) [10]	5,291	11,387	14,844
従来手法 (16 次拡大) [10]	10,266	23,799	29,831

[環境] CPU: 2.5GHz x 2 core, memory: 4GB, Network: 600Mbps

データの並列数, C は回路サイズ, m は入力サイズである.

また実装により, $N = 1,000,000$ で従来手法よりも実際に $\kappa = 7$ で 3 倍程度, $\kappa = 15$ で 7 倍程度の改善が為されることを示した.

お詫び 3.3 節において, 任意の semi-honest の秘密分散ベース秘密計算を malicious モデルで統計的秘匿性および正当性を持つ秘密計算に変換する手法 [10] における筆者ら自身の誤りを指摘し, [10] および本稿の提案手法の適用対象が, “任意の”semi-honest ではなく, 知る限り全ての乗算プロトコルが満たす, “改ざん模倣可能性”を持つ任意の semi-honest の秘密計算であることを述べた. 深くお詫び申し上げる.

参考文献

[1] sharemind news blog. `http://sharemind.cyber.ee/`.

[2] Viff, the virtual ideal functionality framework. `http://viff.dk/`.

[3] D. Bogdanov, S. Laur, and J. Willemson. Sharemind: A framework for fast privacy-preserving computations. In S. Jajodia and J. López eds., *ESORICS*, Vol. 5283 of *Lecture Notes in Computer Science*, pp. 192–206. Springer, 2008.

[4] R. Cramer, I. Damgård, and Y. Ishai. Share conversion, pseudorandom secret-sharing and applications to secure computation. In J. Kilian ed., *TCC*, Vol. 3378 of *Lecture Notes in Computer Science*, pp. 342–362. Springer, 2005.

[5] I. Damgård, M. Geisler, M. Krøigaard, and J. B. Nielsen. Asynchronous multiparty computation: Theory and implementation. *IACR Cryptology ePrint Archive*, 2008:415, 2008.

[6] I. Damgård and J. B. Nielsen. Scalable and unconditionally secure multiparty computation. In A. Menezes ed., *CRYPTO*, Vol. 4622 of *Lecture Notes in Computer Science*, pp. 572–590. Springer, 2007.

[7] R. Gennaro, M. O. Rabin, and T. Rabin. Simplified vss and fact-track multiparty computations with applications to threshold cryptography. In B. A. Coan and Y. Afek eds., *PODC*, pp. 101–111. ACM, 1998.

[8] S. Laur, J. Willemson, and B. Zhang. Round-efficient oblivious database manipulation. In X. Lai, J. Zhou, and H. Li eds., *ISC*, Vol. 7001 of *Lecture Notes in Computer Science*, pp. 262–277. Springer, 2011.

[9] 五十嵐大, 菊池亮, 濱田浩気, 千田浩司. マルチパーティ計算可能な秘密分散におけるデータ改ざん検知方法. In *SCIS2013*, 2013.

[10] 五十嵐大, 千田浩司, 濱田浩気, 菊池亮. 非常に高効率な $n \geq 2k - 1$ malicious モデル上秘密分散ベースマルチパーティ計算の構成法. In *SCIS2013*, 2013.

[11] 五十嵐大, 千田浩司, 濱田浩気, 高橋克巳. 軽量検証可能 3 パーティ秘匿関数計算の効率化及びこれを用いたセキュアなデータベース処理. In *SCIS2011*, 2011.

[12] 五十嵐大, 濱田浩気, 菊池亮, 千田浩司. 少パーティの秘密分散ベース秘密計算のための $o(\ell)$ ビット通信ビット分解および $o(p')$ ビット通信 modulus 変換法. In *CSS2013*, 2013.

[13] 濱田浩気, 五十嵐大, 千田浩司, 高橋克巳. 3 パーティ秘匿関数計算のランダム置換プロトコル. In *CSS2010*, 2010.