

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7540501号
(P7540501)

(45)発行日 令和6年8月27日(2024.8.27)

(24)登録日 令和6年8月19日(2024.8.19)

(51)国際特許分類

G 0 9 C 1/00 (2006.01)

F I G 0 9 C 1/00 6 5 0 Z

請求項の数 8 (全21頁)

(21)出願番号	特願2022-556803(P2022-556803)	(73)特許権者	000004226
(86)(22)出願日	令和2年10月16日(2020.10.16)		日本電信電話株式会社
(86)国際出願番号	PCT/JP2020/039078		東京都千代田区大手町一丁目5番1号
(87)国際公開番号	WO2022/079891	(74)代理人	100121706
(87)国際公開日	令和4年4月21日(2022.4.21)		弁理士 中尾 直樹
審査請求日	令和5年4月4日(2023.4.4)	(74)代理人	100128705
			弁理士 中村 幸雄
		(74)代理人	100147773
			弁理士 義村 宗洋
		(72)発明者	五十嵐 大
			東京都千代田区大手町一丁目5番1号
			日本電信電話株式会社内
		審査官	吉田 歩

最終頁に続く

(54)【発明の名称】 秘匿MSB正規化システム、分散処理装置、秘匿MSB正規化方法、およびプログラム

(57)【特許請求の範囲】

【請求項1】

n個の分散処理装置を含む秘匿MSB正規化システムであって、
n個の前記分散処理装置は、それぞれビット分解部、論理和取得部、シフト量取得部およびシフト部を含み、
n個の前記ビット分解部は、 (k,n) -秘密分散したシェアのベクトル $[[\vec{a}]]^P$ をビット分解し、ベクトル $[[\vec{a}]]^P$ のビット表現 $[[\vec{a}]]^{2^L}$ を得、
n個の前記論理和取得部は、前記ビット表現 $[[\vec{a}]]^{2^L}$ の各ビット位置のベクトル $[[\vec{a}_i]]$ に対して、全要素の論理和 $[[A_i]]^2$ を取得し、
n個の前記シフト量取得部は、論理和 $[[A_0]]^2, \dots, [[A_{L-1}]]^2$ の最上位ビットを固定位置にシフトさせるためのシフト量 p を法 p で (k,n) -複製秘密分散したシェア $\langle \langle p \rangle \rangle^P$ を求め、
n個の前記シフト部は、前記ベクトル $[[\vec{a}]]^P$ の各要素を p ビット左シフトさせたベクトル $[[2^p \vec{a}]]^P$ を求める、
秘匿MSB正規化システム。

【請求項2】

請求項1の秘匿MSB正規化システムであって、
固定小数点ベクトル $[[\vec{a}]]^P, [[\vec{b}]]^P$ から最上位ビットを固定位置にシフトさせた後のベクトルとシフト量 $([[2^{p-a} \vec{a}]]^P, \langle \langle p_a \rangle \rangle^P), ([2^{p-b} \vec{b}]]^P, \langle \langle p_b \rangle \rangle^P)$ を求め、

n 個の前記分散処理装置は、それぞれモジュラス変換部、積和計算部、秘密分散変換部およびシフト量秘匿左右シフト部を含み、

n 個の前記モジュラス変換部は、 $\langle \langle \rho_a \rangle \rangle^P, \langle \langle \rho_b \rangle \rangle^P$ から $\text{mod } p \rightarrow \text{mod } Q$ 変換により、 $[[\rho_a]]^Q, [[\rho_b]]^Q$ を得、

n 個の前記積和計算部は、 $[[c]]^P := [[\sum_{0 \leq i < m} 2^{P-a} a_i 2^{P-b} b_i]]^P$ を計算し、

n 個の前記秘密分散変換部は、前記 $[[\rho_a]]^Q, [[\rho_b]]^Q$ から、 $[-\rho_a - \rho_b]^Q$ を計算し、秘密分散変換により、 (k, n) -複製秘密分散したシェア $\langle \langle -\rho_a - \rho_b \rangle \rangle^Q$ を求め、

n 個の前記シフト量秘匿左右シフト部は、積和のシェア $[[c]]^P$ とシフト量のシェア $\langle \langle -\rho_a - \rho_b \rangle \rangle^Q$ を受け取り、 $[[c]]^P$ を $\langle \langle -\rho_a - \rho_b \rangle \rangle^Q$ ビットシフトする、

秘匿MSB正規化システム。

10

【請求項3】

請求項1の秘匿MSB正規化システムであって、

浮動小数点ベクトル $([[\rightarrow a]]^P, [[\rightarrow \rho_a]]^Q), ([[\rightarrow b]]^P, [[\rightarrow \rho_b]]^Q)$ から最上位ビットをシフトさせた後のベクトルとシフト量 $([[\rightarrow a']]^P, \langle \langle \rho'_a \rangle \rangle^P), ([[\rightarrow b']]^P, \langle \langle \rho'_b \rangle \rangle^P)$ を求め、

n 個の前記分散処理装置は、それぞれモジュラス変換部、指数統一部および積和計算部を含み、

n 個の前記モジュラス変換部は、前記 $\langle \langle \rho'_a \rangle \rangle^P$ と前記 $\langle \langle \rho'_b \rangle \rangle^P$ とを、 $\text{mod } p \rightarrow \text{mod } Q$ 変換し $[[\rho_a']]^Q, [[\rho_b']]^Q$ を得、

n 個の前記指数統一部は、前記浮動小数点ベクトル $([[\rightarrow a]]^P, [[\rightarrow \rho_a]]^Q), ([[\rightarrow b]]^P, [[\rightarrow \rho_b]]^Q)$ の指数部 $[[\rightarrow \rho_a]]^Q, [[\rightarrow \rho_b]]^Q$ と、最上位ビットをシフトさせた後のベクトル $[[\rightarrow a']]^P, [[\rightarrow b']]^P$ と、 $\text{mod } p \rightarrow \text{mod } Q$ 変換後のシフト量 $[[\rho_a']]^Q, [[\rho_b']]^Q$ とを用いて、 $([[\rightarrow a']]^P, [[\rightarrow \rho_a - \rho_a']]^Q), ([[\rightarrow b']]^P, [[\rightarrow \rho_b - \rho_b']]^Q)$ の指数部を統一したベクトルと指数部、 $([[\rightarrow a'']]^P, [[\rho_a'']]^Q), ([[\rightarrow b'']]^P, [[\rho_b'']]^Q)$ を得、

20

n 個の前記積和計算部は、 $[[c]]^P := [[\sum_{0 \leq i < m} a''_i b''_i]]^P$ を計算し、 $([[c]]^P, [[\rho_a'' + \rho_b'']]^Q)$ を得る、

秘匿MSB正規化システム。

【請求項4】

秘匿MSB正規化システムに含まれる分散処理装置であって、

$(n-1)$ 個の分散処理装置とともに、 (k, n) -秘密分散したシェアのベクトル $[[\rightarrow a]]^P$ をビット分解し、ベクトル $[[\rightarrow a]]^P$ のビット表現 $[[\rightarrow a]]^{2^L}$ を得るビット分解部と、

30

$(n-1)$ 個の分散処理装置とともに、前記ビット表現 $[[\rightarrow a]]^{2^L}$ の各ビット位置のベクトル $[[\rightarrow a_i]]$ に対して、全要素の論理和 $[[A_i]]^2$ を取得する論理和取得部と、

$(n-1)$ 個の分散処理装置とともに、論理和 $[[A_0]]^2, \dots, [[A_{L-1}]]^2$ の最上位ビットを固定位置にシフトさせるためのシフト量 ρ を法 p で (k, n) -複製秘密分散したシェア $\langle \langle \rho \rangle \rangle^P$ を求めるシフト量取得部と、

$(n-1)$ 個の分散処理装置とともに、前記ベクトル $[[\rightarrow a]]^P$ の各要素を ρ ビット左シフトさせたベクトル $[[2^{\rho} \rightarrow a]]^P$ を求めるシフト部とを含む、

分散処理装置。

【請求項5】

40

n 個の分散処理装置を含む秘匿MSB正規化システムを用いる秘匿MSB正規化方法であって、

n 個の前記分散処理装置は、それぞれビット分解部、論理和取得部、シフト量取得部およびシフト部を含み、

n 個の前記ビット分解部が、 (k, n) -秘密分散したシェアのベクトル $[[\rightarrow a]]^P$ をビット分解し、ベクトル $[[\rightarrow a]]^P$ のビット表現 $[[\rightarrow a]]^{2^L}$ を得るビット分解ステップと、

n 個の前記論理和取得部が、前記ビット表現 $[[\rightarrow a]]^{2^L}$ の各ビット位置のベクトル $[[\rightarrow a_i]]$ に対して、全要素の論理和 $[[A_i]]^2$ を取得する論理和取得ステップと、

n 個の前記シフト量取得部が、論理和 $[[A_0]]^2, \dots, [[A_{L-1}]]^2$ の最上位ビットを固定位置にシフトさせるためのシフト量 ρ を法 p で (k, n) -複製秘密分散したシェア $\langle \langle \rho \rangle \rangle^P$ を求め

50

るシフト量取得ステップと、

n 個の前記シフト部が、前記ベクトル $[[\rightarrow a]]^P$ の各要素を p ビット左シフトさせたベクトル $[[2^P \rightarrow a]]^P$ を求めるシフトステップとを含む、

秘匿MSB正規化方法。

【請求項6】

請求項5の秘匿MSB正規化方法であって、

固定小数点ベクトル $[[\rightarrow a]]^P, [[\rightarrow b]]^P$ から最上位ビットを固定位置にシフトさせた後のベクトルとシフト量 $([[2^P - a \rightarrow a]]^P, \langle \langle p_a \rangle \rangle^P), ([[2^P - b \rightarrow b]]^P, \langle \langle p_b \rangle \rangle^P)$ を求め、

n 個の前記分散処理装置は、それぞれモジュラス変換部、積和計算部、秘密分散変換部およびシフト量秘匿左右シフト部を含み、

n 個の前記モジュラス変換部が、 $\langle \langle p_a \rangle \rangle^P, \langle \langle p_b \rangle \rangle^P$ から $\text{mod } p \rightarrow \text{mod } Q$ 変換により、 $[[p_a]]^Q, [[p_b]]^Q$ を得るモジュラス変換ステップ、

n 個の前記積和計算部が、 $[[c]]^P := [[\sum_{0 \leq i < m} 2^{P-a} a_i 2^{P-b} b_i]]^P$ を計算する積和計算ステップと、

n 個の前記秘密分散変換部が、前記 $[[p_a]]^Q, [[p_b]]^Q$ から、 $[-p_a - p_b]]^Q$ を計算し、秘密分散変換により、 (k, n) -複製秘密分散したシェア $\langle \langle -p_a - p_b \rangle \rangle^Q$ を求める秘密分散変換ステップと、

n 個の前記シフト量秘匿左右シフト部が、積和のシェア $[[c]]^P$ とシフト量のシェア $\langle \langle -p_a - p_b \rangle \rangle^Q$ とを受け取り、 $[[c]]^P$ を $\langle \langle -p_a - p_b \rangle \rangle^Q$ ビットシフトするシフト量秘匿左右シフトステップとを含む、

秘匿MSB正規化方法。

【請求項7】

請求項5の秘匿MSB正規化方法であって、

浮動小数点ベクトル $([[\rightarrow a]]^P, [[\rightarrow p_a]]^Q), ([[\rightarrow b]]^P, [[\rightarrow p_b]]^Q)$ から最上位ビットをシフトさせた後のベクトルとシフト量 $([[\rightarrow a']]^P, \langle \langle p'_a \rangle \rangle^P), ([[\rightarrow b']]^P, \langle \langle p'_b \rangle \rangle^P)$ を求め、

n 個の前記分散処理装置は、それぞれモジュラス変換部、指数統一部および積和計算部を含み、

n 個の前記モジュラス変換部が、前記 $\langle \langle p'_a \rangle \rangle^P$ と前記 $\langle \langle p'_b \rangle \rangle^P$ とを、 $\text{mod } p \rightarrow \text{mod } Q$ 変換し $[[p'_a]]^Q, [[p'_b]]^Q$ を得るモジュラス変換ステップと、

n 個の前記指数統一部が、前記浮動小数点ベクトル $([[\rightarrow a]]^P, [[\rightarrow p_a]]^Q), ([[\rightarrow b]]^P, [[\rightarrow p_b]]^Q)$ の指数部 $[[\rightarrow p_a]]^Q, [[\rightarrow p_b]]^Q$ と、最上位ビットをシフトさせた後のベクトル $[[\rightarrow a']]^P, [[\rightarrow b']]^P$ と、 $\text{mod } p \rightarrow \text{mod } Q$ 変換後のシフト量 $[[p'_a]]^Q, [[p'_b]]^Q$ とを用いて、 $([[\rightarrow a']]^P, [[\rightarrow p_a - p'_a]]^Q), ([[\rightarrow b']]^P, [[\rightarrow p_b - p'_b]]^Q)$ の指数部を統一したベクトルと指数部、 $([[\rightarrow a'']]^P, [[p_{a''}]]^Q), ([[\rightarrow b'']]^P, [[p_{b''}]]^Q)$ を得る指数統一ステップと、

n 個の前記積和計算部が、 $[[c]]^P := [[\sum_{0 \leq i < m} m a''_i b''_i]]^P$ を計算し、 $([[c]]^P, [[p_{a''} + p_{b''}]]^Q)$ を得る積和ステップとを含む、

秘匿MSB正規化方法。

【請求項8】

請求項4の分散処理装置として、コンピュータを機能させるためのプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、秘密計算において、最上位ビット(以下「MSB」(Most Significant Bit)ともいう)を所定のビット位置に合わせる技術(以下、「MSB合わせ」ともいう)に関する。

【背景技術】

【0002】

平文では積和演算は加算の繰り返しだが、秘密分散ベースの秘密計算では計算効率を上げるために並列処理できる必要がある(非特許文献1, 2参照)。また精度も考慮すると和

10

20

30

40

50

、積和については特別に演算を構成する必要がある。

【先行技術文献】

【非特許文献】

【0003】

【文献】西出隆志,天田拓磨,「通信量を削減した浮動小数点演算のためのマルチパーティ計算」, 情報処理学会論文誌, Vol. 60, No.9, pp. 1433- 1447, 2019.

【文献】Randmets, J., "Programming Languages for Secure Multiparty Computation Application Development", PhD thesis. University of Tartu. 2017.

【発明の概要】

【発明が解決しようとする課題】

【0004】

積和は出力が高ビットになるため入力時点でMSBを一定に抑えたいが、単純に右シフトすると、小さい入力が桁落ちして精度が落ちるという課題がある。

【0005】

本発明は、ベクトルに含まれる要素の中で最も絶対値の大きいデータのMSB(ベクトルMSBと呼ぶ)を所定のビット位置に合わせるようなシフトでベクトル全体を一斉にシフトすること(ベクトルMSB正規化と呼ぶ)で精度を保ったままMSB合わせを行うことができる秘匿MSB正規化システム、分散処理装置、秘匿MSB正規化方法、およびプログラムを提供することを目的とする。

【課題を解決するための手段】

【0006】

上記の課題を解決するために、本発明の一態様によれば、秘匿MSB正規化システムは、 n 個の分散処理装置を含む。 n 個の分散処理装置は、それぞれビット分解部、論理和取得部、シフト量取得部およびシフト部を含む。 n 個のビット分解部は、 (k, n) -秘密分散したシェアのベクトル $[[\rightarrow a]]^P$ をビット分解し、ベクトル $[[\rightarrow a]]^P$ のビット表現 $[[\rightarrow a]]^{2^L}$ を得、 n 個の論理和取得部は、ビット表現 $[[\rightarrow a]]^{2^L}$ の各ビット位置のベクトル $[[\rightarrow a_i]]$ に対して、全要素の論理和 $[[A_i]]^2$ を取得し、 n 個のシフト量取得部は、論理和 $[[A_0]]^2, \dots, [[A_{L-1}]]^2$ の最上位ビットを固定位置にシフトさせるためのシフト量 p を法 p で (k, n) -複製秘密分散したシェア $\langle \langle p \rangle \rangle^P$ を求め、 n 個のシフト部は、ベクトル $[[\rightarrow a]]^P$ の各要素を p ビット左シフトさせたベクトル $[[2^p \rightarrow a]]^P$ を求める。

【0007】

上記の課題を解決するために、本発明の他の態様によれば、分散処理装置は、秘匿MSB正規化システムに含まれる。分散処理装置は、 $(n-1)$ 個の分散処理装置とともに、 (k, n) -秘密分散したシェアのベクトル $[[\rightarrow a]]^P$ をビット分解し、ベクトル $[[\rightarrow a]]^P$ のビット表現 $[[\rightarrow a]]^{2^L}$ を得るビット分解部と、 $(n-1)$ 個の分散処理装置とともに、ビット表現 $[[\rightarrow a]]^{2^L}$ の各ビット位置のベクトル $[[\rightarrow a_i]]$ に対して、全要素の論理和 $[[A_i]]^2$ を取得する論理和取得部と、 $(n-1)$ 個の分散処理装置とともに、論理和 $[[A_0]]^2, \dots, [[A_{L-1}]]^2$ の最上位ビットを固定位置にシフトさせるためのシフト量 p を法 p で (k, n) -複製秘密分散したシェア $\langle \langle p \rangle \rangle^P$ を求めるシフト量取得部と、 $(n-1)$ 個の分散処理装置とともに、ベクトル $[[\rightarrow a]]^P$ の各要素を p ビット左シフトさせたベクトル $[[2^p \rightarrow a]]^P$ を求めるシフト部とを含む。

【発明の効果】

【0008】

本発明によれば、精度を保ったままMSB合わせを行うことができるという効果を奏する。

【図面の簡単な説明】

【0009】

【図1】第一、第二、第三実施形態に係る秘匿MSB正規化システムの構成例を示す図。

【図2】第一実施形態に係る秘匿MSB正規化システムの処理フローの例を示す図。

【図3】第一実施形態に係る分散処理装置の機能ブロック図。

【図4】第二実施形態に係る秘匿MSB正規化システムの処理フローの例を示す図。

【図5】第二実施形態に係る分散処理装置の機能ブロック図。

【図6】第三実施形態に係る秘匿MSB正規化システムの処理フローの例を示す図。

【図7】第三実施形態に係る分散処理装置の機能ブロック図。

【図8】実機実験の結果を示す図。

【図9】本手法を適用するコンピュータの構成例を示す図。

【発明を実施するための形態】

【0010】

以下、本発明の実施形態について、説明する。なお、以下の説明に用いる図面では、同じ機能を持つ構成部や同じ処理を行うステップには同一の符号を記し、重複説明を省略する。以下の説明において、テキスト中で使用する記号「 $\rightarrow$ 」等は、本来直後の文字の真上に記載されるべきものであるが、テキスト記法の制限により、当該文字の直前に記載する。式中においてはこれらの記号は本来の位置に記述している。また、ベクトルや行列の各要素単位で行われる処理は、特に断りが無い限り、そのベクトルやその行列の全ての要素に対して適用されるものとする。

【0011】

<第一実施形態>

まず、本実施形態における記法について説明する。

【0012】

<記法>

◎ k : 秘密分散の閾値。例えば2とする。

【0013】

◎ n : 秘密分散の分散数、言い換えると、秘密計算のパーティ数。例えば3とする。

【0014】

◎ P : 素数。本実施形態ではメルセンヌ素数 $2^{61}-1$ を想定し、処理の効率化を図る。

【0015】

◎ p : P のビット数。 P がメルセンヌ素数のとき、やはり素数であり、61である。

【0016】

◎ Q : 剰余環の位数。 P 、 p や、浮動小数点の指数部に使う位数を含む一般的な位数を意味する。浮動小数点の指数部のシェアに用いたときは特に $2^{13}-1$ を想定する。

【0017】

◎ L : 格納されるデータの最大ビット長。 p より小さいことを想定する。

【0018】

◎ λ : 格納される指数部の最大ビット長。10以下を想定する。

【0019】

◎ $[[x]]^y$: $\text{mod } y$ 要素 x を (k, n) -秘密分散したシェア。

【0020】

◎ $\langle x \rangle^y$: $\text{mod } y$ 要素 x を (k, k) -加法的秘密分散したシェア。

【0021】

◎ $\langle \langle x \rangle \rangle^y$: $\text{mod } y$ 要素 x を (k, n) -複製秘密分散したシェア。なお、 (k, n) -秘密分散であるため、 $[[x]]^y$ の形式のシェアに適用できるプロトコルはこのシェアにも適用できる。この記法の場合、特に複製秘密分散の性質を利用していることを意味する。

【0022】

◎ $[[x]]^{2^m} : [[x]]^2$ の形式のシェアを m 個並べたシェア。数値のビット表現と見なすこともある。なお、添え字における A^B は A^B を、 A_B は A_B を意味する。

【0023】

◎ $p \circ \rightarrow a$: ベクトル $\rightarrow a$ にローテーション p を施したベクトル。ローテーションは数でもあり置換でもあるため、要素ごとの乗算 $p \rightarrow a$ と区別する。

【0024】

◎ $x \sim y$: x と y は計算機上の実数として等しい。すなわち、差が一定の誤差範囲内である。

【0025】

10

20

30

40

50

◎ a/d : 小数点以下切り捨ての整数除算。特に2べき数での整数除算は右シフトと等しい。

【数 1】

◎ $\frac{a}{d}$: 実数除算。

【0026】

◎ {命題}: 命題が成り立てば1、成り立たなければ0。

【0027】

次に、本実施形態で用いる (k,n) -秘密分散、 (k,k) -加法的秘密分散の2つの秘密分散について説明する。

【0028】

〈 (k,n) -秘密分散〉

(k,n) -秘密分散とは、入力された平文を n 個の断片(シェアと呼ぶ)に分割し、それぞれ n 個の異なる主体(パーティと呼ぶ)に分散しておき、うち任意の k 個のシェアが揃えば復元でき、そして $k-1$ 個未満では平文に関する一切の情報を得られないようなセキュリティ技術である。例えば Shamir の秘密分散、複製秘密分散などがある。本実施形態では (k,n) -秘密分散で分散され、平文がある値 x であるようなシェアを全て集めた組((k,n) -秘密分散値ともいう)を $[[x]]$ のように表記する。各シェアのことは、パーティ r のシェアを $[[x]]^r$ と表記する。ここでは $r=0, \dots, n-1$ とする。秘密分散値は普段は各パーティに分散されているため、誰も所持しておらず仮想的である。また (k,n) -秘密分散値の列であって平文の列が $\rightarrow x$ となる列を、 $[[\rightarrow x]]$ のように表記する。

【0029】

〈 (k,k) -加法的秘密分散〉

(k,k) -秘密分散とは、 (k,n) -秘密分散で、 $n=k$ とした場合である。全パーティのシェアが集まらない限り復元ができない。複製秘密分散による (k,k) -秘密分散は特に、加法的秘密分散と呼ばれ、 k 個のシェアを加算するだけで平文が復元される最もシンプルな方法である。本実施形態では法 y のもとで (k,k) -加法的秘密分散で分散され、平文がある値 x であるようなシェアを全て集めた組((k,k) -加法的秘密分散値ともいう)を $\langle x \rangle^y$ のように表記し、パーティ r のシェアを $\langle x \rangle^y_r$ と表記する。また (k,k) -加法的秘密分散値の列であって平文の列が $\rightarrow x$ となる列を、 $\langle \rightarrow x \rangle^y$ のように表記する。

【0030】

まず、第一実施形態に係る秘匿 MSB 正規化システムで使われるいくつかのプロトコルについて説明する。

【0031】

〈乗法的ローテーションプロトコル〉

入力: (k,n) -秘密分散した数値シェア $[[a]]^P$ 、ローテーション量 p を複製秘密分散したシェア $\langle \langle p \rangle \rangle^P$

出力: (k,n) -秘密分散したシェア $[[2^p a]]^P$

処理: 数値シェア $[[a]]^P$ とシェア $\langle \langle p \rangle \rangle^P$ とを用いて、数値 a を p ビットローテーションさせた値 $2^p a$ を (k,n) -秘密分散したシェア $[[2^p a]]^P$ を得る。この例では、 $k=2, n=3$ とする。以下、処理の詳細を説明する。

【0032】

1: ラウンド1

2: 数値シェア $[[a]]^P$ を、 (k,k) -加法的秘密分散したシェア $\langle a \rangle^P$ に変換する。この例では、パーティ0,1がシェア $\langle a \rangle^P$ を持つ。 (k,n) -秘密分散から (k,k) -加法的秘密分散への変換は、公知の技術により行うことができる。例えば、参考文献1を利用する。

【0033】

(参考文献1) Kikuchi, R., Ikarashi, D., Matsuda, T., Hamada, K. and Chida, K., "Efficient Bit-Decomposition and Modulus-Conversion Protocols with an Honest Majority", Information Security and Privacy - 23rd Australasian Conferenc

10

20

30

40

50

e, ACISP 2018, Wollongong, NSW, Australia, July 11-13, 2018, Proceedings (Susilo, W. and Yang, G., eds.), Lecture Notes in Computer Science, Vol. 10946, Springer, pp. 64-82 (online).

3:パーティ0,1は、乱数 r_{01} 、パーティ1,2は r_{12} を共有しておく。なお、乱数は、乱数を共有するパーティの一方が生成し、他方に渡してもよいし、第三者が生成して、対応するパーティに渡してもよいし、トークン等を利用して共有してもよい。

【0034】

4:パーティ0は

【数2】

$$b_0 := 2^{<<\rho>>_{01}^p} <a>_0^p - r_{01}$$

10

を計算してパーティ2に送る。

【0035】

5:パーティ1は

【数3】

$$b_1 := 2^{<<\rho>>_{12}^p} \left(2^{<<\rho>>_{01}^p} <a>_1^p + r_{01} \right) - r_{12}$$

を計算してパーティ0に送る。

20

【0036】

6:ラウンド2

7:パーティ0は

【数4】

$$<c>_0^p := 2^{<<\rho>>_{20}^p} b_1$$

を計算する。

【0037】

8:パーティ2は

【数5】

$$<c>_2^p := 2^{<<\rho>>_{20}^p} \left(2^{<<\rho>>_{12}^p} b_0 + r_{12} \right)$$

30

を計算する。

【0038】

9:ラウンド3

10:(k,k)-加法的秘密分散したシェア<c>^pを(k,n)-秘密分散のシェア[[c]]^pに変換して出力する。ここで、 $c=2^p a$ が成り立っている。(k,k)-加法的秘密分散から(k,n)-秘密分散への変換は、公知の技術により行うことができる。例えば、参考文献1を利用する。

40

【0039】

<フラグ列→数値シェア変換プロトコル>

入力:長さpのビットシェアベクトル[[$\rightarrow f$]]²。ただし $\rightarrow f$ の中にはただ一つだけ1が存在する。

【0040】

出力: $\rightarrow f$ の中に存在する1の位置のmod pシェア[[b]]^p

処理:ビットシェアベクトル[[$\rightarrow f$]]²を用いて、 $\rightarrow f$ の中に存在する1の位置bを(k,n)-秘密分散したシェア[[$2^p a$]]^pを得る。以下、処理の詳細を説明する。

【0041】

1:一様乱数pをmod pで(k,n)-複製秘密分散したシェア<<p>>^pを生成する。

50

【0042】

2:公開値出力ローテーションプロトコルにより、公開値 $p \circ \rightarrow f$ を計算する。なお、 p が一様乱数で、 $\rightarrow f$ は1カ所だけ1と決まっているため、 $p \circ \rightarrow f$ は数値をビット位置として表現したローテーション上の一様乱数であり、公開しても安全である。公開値出力ローテーションプロトコルとは、 (k,n) -秘密分散したシェアのベクトル $[[\rightarrow a]]$ と (k,n) -複製秘密分散したローテーション量 p のシェア $\langle \langle p \rangle \rangle^Q$ とを入力とし、ベクトル $\rightarrow a$ を p だけローテーションさせて得られる $p \circ \rightarrow a$ (公開値) を得るプロトコルであり、公知の技術により実現することができる。例えば、参考文献2の公開値出力ランダム置換プロトコルの、ランダム置換の空間をランダムローテーションに制限することができる。

(参考文献2) 五十嵐大、濱田浩気、菊池亮、千田浩司、「インターネット環境レスポンス1秒の統計処理を目指した、秘密計算基数ソートの改良」、SCIS 2014 The 31st Symposium on Cryptography and Information Security.

【0043】

3: $p \circ \rightarrow f$ の1の位置を得て b' とおく。 b' は元の1の位置 b に対して、 $b' = b + p$ となっている。

【0044】

4: $\langle \langle b \rangle \rangle^P = b' - \langle \langle p \rangle \rangle^P$ を計算して出力する。

なお、入力のビットシェアベクトル $[[\rightarrow f]]^2$ の長さが p より短い場合も、上位ビットに $[[0]]^2$ をパディングすることで、本プロトコルを適用することができる。

【0045】

以下、本実施形態で実現するベクトルMSB正規化について説明する。

【0046】

〈ベクトルMSB正規化プロトコル〉

入力:シェアのベクトル $[[\rightarrow a]]^P$

パラメータ:入力の最大ビット数 L 、ベクトル長 m

出力: $[[2^P \rightarrow a]]^P$ 、 $\langle \langle p \rangle \rangle^P$ 、ただし $2^P \rightarrow a$ のベクトルMSBは $L-1$ ビット目とする。

【0047】

処理:ベクトル $[[\rightarrow a]]^P$ に含まれる要素の中で最も絶対値の大きいデータのMSB(ベクトルMSB)を固定位置(ここでは $L-1$ ビット目)に合わせるようにベクトル $[[\rightarrow a]]^P$ 全体をシフトし、シフト後のベクトル $[[2^P \rightarrow a]]^P$ とシフト量 $\langle \langle p \rangle \rangle^P$ とを求める。以下、処理の詳細を説明する。

【0048】

1:ビット分解により $[[\rightarrow a]]^P$ のビット表現 $[[\rightarrow a]]^{2^L}$ を得る。ビット分解は、公知の技術により行うことができる。例えば、参考文献1を利用する。

【0049】

2: $[[\rightarrow a]]^{2^L}$ の各ビット位置 $0 \leq i < L$ のベクトル $[[\rightarrow a_i]]^2$ に対して、全要素のORをとる。 $\rightarrow a$ の s 番目の要素を a_s とし、 $s=0,1,\dots,m-1$ 、 $[[a_s]]$ のビット表現を $[[a_s]]^{2^L}$ とし、 $[[a_s]]^{2^L}$ の i 番目のビット位置のシェアを $[(a_i)_s]^2$ とすると、ベクトル $[[\rightarrow a_i]] = ([[(a_i)_0]]^2, \dots, [[(a_i)_{m-1}]]^2)$ であり、求める論理和は、 $[[A_i]]^2 := [[(a_i)_0]]^2 \text{ OR } \dots \text{ OR } [[(a_i)_{m-1}]]^2$ である。

【0050】

3-1: $0 \leq i < L-1$ で帰納的に、 $[[f_i]]^2 := [[f_{i+1} \vee A_i]]^2$ とする。ただし $[[f_{L-1}]]^2 := [[A_{L-1}]]^2$ とする。ここまでで、ビット表現 $f = (f_{L-1}, f_{L-2}, \dots, f_0)$ は、 $0, 0, 0, 1, 1, \dots, 1$ のような、MSBを境に01が並ぶ形になっている。

【0051】

3-2:最大 $p-L$ 個の $[[1]]^2$ を下位ビットに挿入し、 $([[f'_0]]^2, \dots, [[f'_{L'-1}]]^2) := ([f_0]]^2, \dots, [f_{L-1}]]^2, [[f_0]]^2, \dots, [f_{L-1}]]^2)$ とする。 L' は L に挿入した $[[1]]^2$ の個数を足した数である。この処理により、 $\rightarrow a$ の全ての要素が0のときにもMSB位置が定義される。

3-3: $0 \leq i < L'-1$ で、 $[[x_i]]^2 := [[f'_i \text{ xor } f'_{i+1}]]^2$ とする。ただし $[[x_{L'-1}]]^2 := [[A_{L-1}]]^2$ である。ここまでで、ビット表現 $x = (x_{L'-1}, x_{L'-2}, \dots, x_0)$ は、 $0, 0, 0, 1, 0, \dots, 0$ のように、MSB位置のみ1となるフラグになっている。

【0052】

10

20

30

40

50

4: 上述の〈フラグ列→数値シェア変換プロトコル〉により、 $[[x_{L'-1}]]^2, [[x_{L'-2}]]^2, \dots, [[x_0]]^2$ を $\langle \langle \rho \rangle \rangle^P$ に変換する。ただし、降順であることに注意する。

【0053】

5: 上述の〈乗法的ローテーション〉により、シェアのベクトル $[[\vec{a}]]^P$ とローテーション量の複製秘密分散したシェア $\langle \langle \rho \rangle \rangle^P$ から、 (k, n) -秘密分散したシェア $[[2^{\rho} \vec{a}]]^P$ を求め、出力する。ただし、乗法的ローテーションは、他の公知の技術により行ってもよい。

【0054】

以下、上述のベクトルMSB正規化を実現するベクトルMSB正規化システムについて説明する。

【0055】

<第一実施形態に係るベクトルMSB正規化システム>

図1は第一実施形態に係るベクトルMSB正規化システム1の構成例を、図2はベクトルMSB正規化システム1の処理フローの例を示す。

【0056】

ベクトルMSB正規化システム1は、 n 個の分散処理装置100-rを含む。ただし、 n は3以上の整数の何れかであり、 $r=0, 1, \dots, n-1$ である。 n 個の分散処理装置100-rは、通信回線2を介して互いに通信可能である。

【0057】

ベクトルMSB正規化システム1は、法 P でベクトル $\vec{a}$ の各要素 a_s を (k, n) -秘密分散したシェア $[[a_s]]^P$ のベクトル $[[\vec{a}]]^P$ を入力とし、ベクトルMSB正規化し、ベクトルMSB正規化後のベクトル $[[2^{\rho} \vec{a}]]^P$ とシフト量 $\langle \langle \rho \rangle \rangle^P$ を求め、出力する。シェア $[[a_s]]^P$ の最大ビット長 L 、ベクトル $[[\vec{a}]]^P$ のベクトル長 m をパラメータとする。

【0058】

分散処理装置は、例えば、中央演算処理装置（CPU: Central Processing Unit）、主記憶装置（RAM: Random Access Memory）などを有する公知又は専用のコンピュータに特別なプログラムが読み込まれて構成された特別な装置である。分散処理装置は、例えば、中央演算処理装置の制御のもとで各処理を実行する。分散処理装置に入力されたデータや各処理で得られたデータは、例えば、主記憶装置に格納され、主記憶装置に格納されたデータは必要に応じて中央演算処理装置へ読み出されて他の処理に利用される。分散処理装置の各処理部は、少なくとも一部が集積回路等のハードウェアによって構成されていてもよい。分散処理装置が備える各記憶部は、例えば、RAM（Random Access Memory）などの主記憶装置、またはリレーショナルデータベースやキーバリューストアなどのミドルウェアにより構成することができる。ただし、各記憶部は、必ずしも分散処理装置がその内部に備える必要はなく、ハードディスクや光ディスクもしくはフラッシュメモリ（Flash Memory）のような半導体メモリ素子により構成される補助記憶装置により構成し、分散処理装置の外部に備える構成としてもよい。

【0059】

<分散処理装置100-r>

図3は、分散処理装置100-rの機能ブロック図の例を示す。

【0060】

分散処理装置100-rは、ビット分解部101、論理和取得部103、シフト量取得部105およびシフト部107を含む。

【0061】

以下、図2を用いて各部の処理について説明する。

【0062】

<ビット分解部101>

n 個のビット分解部101は、 (k, n) -秘密分散したシェアのベクトル $[[\vec{a}]]^P$ を受け取り、ビット分解により、ベクトル $[[\vec{a}]]^P$ のビット表現 $[[\vec{a}]]^{2^L}$ を得る（S101）。

【0063】

10

20

30

40

50

<論理和取得部 1 0 3>

n 個の論理和取得部 1 0 3 は、ビット表現 $[[\rightarrow a]]^{2^L}$ を受け取り、各ビット位置 $0 \leq i < L$ のベクトル $[[\rightarrow a_i]]$ に対して、全要素の論理和 $[[A_i]]^2$ を取得する(S 1 0 3)。ただし、ベクトル $\rightarrow a$ の s 番目の要素を a_s とし、 $s=0, 1, \dots, m-1$ 、 $[[a_s]]$ のビット表現を $[[a_s]]^{2^L}$ とし、 $[[a_s]]^{2^L}$ の i 番目のビット位置のシェアを $[[a_i]_s]^2$ とすると、ベクトル $[[\rightarrow a_i]] = ([[a_i]_0]]^2, \dots, [[a_i]_{m-1}]^2)$ であり、求める論理和は、 $[[A_i]]^2 := [[a_i]_0]^2 \text{ OR } \dots \text{ OR } [[a_i]_{m-1}]^2$ である。

【0 0 6 4】

<シフト量取得部 1 0 5>

n 個のシフト量取得部 1 0 5 は、論理和 $[[A_i]]^2$ ($0 \leq i < L$)を受け取り、最大ビット数 $L \leq p-1$ をパラメータとして、ベクトル $[[\rightarrow A]]^2 = ([[A_0]]^2, \dots, [[A_{L-1}]^2)$ のMSBを固定位置にシフトさせるためのシフト量 $\langle \langle p \rangle \rangle^p$ を求める(S 1 0 5)。

【0 0 6 5】

例えば、以下のようにして、シフト量 $\langle \langle p \rangle \rangle^p$ を求める。

【0 0 6 6】

まず、 n 個のシフト量取得部 1 0 5 は、 $[[f_{L-1}]]^2 := [[A_{L-1}]]^2$ とし、 $0 \leq i < L-1$ で帰納的に、 $[[f_i]]^2 := [[f_{i+1} \vee A_i]]^2$ とする。この処理により、ビット表現 $f = (f_{L-1}, f_{L-2}, \dots, f_0)$ は、 $0, 0, 1, 1, \dots, 1$ のような、MSBを境に01が並ぶ形になる。

【0 0 6 7】

次に、 n 個のシフト量取得部 1 0 5 は、 $[[x_{L-1}]]^2 := [[A_{L-1}]]^2$ とし、 $0 \leq i < L-1$ で、 $[[x_i]]^2 := [[f_i \text{ xor } f_{i+1}]]^2$ とする。この処理により、ビット表現 $x = (x_{L-1}, x_{L-2}, \dots, x_0)$ は、 $0, 0, 0, 1, 0, \dots, 0$ のように、MSB位置のみ1となるフラグになる。

【0 0 6 8】

最後に、 n 個のシフト量取得部 1 0 5 は、上述の〈フラグ列→数値シェア変換プロトコル〉により、長さ p の列 $[[x_{L-1}]]^2, [[x_{L-2}]]^2, \dots, [[x_0]]^2, [[1]]^2, \dots, [[1]]^2$ を $\langle \langle p \rangle \rangle^p$ に変換する。

【0 0 6 9】

<シフト部 1 0 7>

n 個のシフト部 1 0 7 は、 $[[\rightarrow a]]^p$ と $\langle \langle p \rangle \rangle^p$ を受け取り、 $[[\rightarrow a]]^p$ の各要素を p ビット左シフトさせたベクトル $[[2^p \rightarrow a]]^p$ を求め(S 1 0 7)、出力する。例えば、上述の〈乗法的ローテーション〉により、シェアのベクトル $[[\rightarrow a]]^p$ とローテーション量の複製秘密分散したシェア $\langle \langle p \rangle \rangle^p$ とから、 (k, n) -秘密分散したシェア $[[2^p a]]^p$ を求める。

【0 0 7 0】

<効果>

このような構成により、精度を保ったままMSB合わせを行うことができる。

【0 0 7 1】

<第二実施形態>

第一実施形態とは異なる部分を中心に説明する。

【0 0 7 2】

第二実施形態では、第一実施形態のベクトルMSB正規化を利用した固定小数点ベクトル積和について説明する。まず、第二実施形態に係る固定小数点ベクトル積和で使われるいくつかのプロトコルについて説明する。

【0 0 7 3】

<シフト量秘匿左右シフトプロトコル>

入力: 数値シェア $[[a]]^p$ 、正負ありうる左シフト量のシェア $\langle \langle p \rangle \rangle^q$ 、

パラメータ: 入力のMSB位置のとりうる上限 $M_{\max}$ 、シェアが許容する最大のMSB位置 $M_{\lim}$

出力: p ビットシフトした値 $[[s]]^p$

1: まず $u := M_{\lim} - M_{\max} + 1$ とおき、

【数 6】

10

20

30

40

50

$$d := \left\lceil \frac{M_{\max} - 1}{u} \right\rceil$$

とおく。 u は、一回のシフト量秘匿右シフトでカバーできる右シフト量の範囲の大きさ($0 \sim (M_{\lim} - M_{\max})$ をカバーする)であり、 d は $1 \sim (M_{\max} - 1)$ ビットの範囲の右シフトをするのに必要なシフト量秘匿右シフトの回数である。右シフト量が0以下の場合は左シフトでよく、右シフト量が $M_{\max}$ 以上の場合、出力は常に0である。

【0074】

2:商転移を使うモジュラス変換により $\langle \langle \rho \rangle \rangle^P$ を計算する。商転移を使うモジュラス変換は、公知の技術により行うことができる。例えば、参考文献1を利用する。

10

【0075】

3:大小比較により、

$$\begin{aligned} [[f_0]]^2 &:= [[\{\rho \geq -M_{\max} + 1\}]]^2, \\ [[f_1]]^2 &:= [[\{\rho \geq -M_{\max} + 1 + u\}]]^2, \dots, \\ [[f_{d-1}]]^2 &:= [[\{\rho \geq -M_{\max} + 1 + (d-1)u\}]]^2, \\ [[f_L]]^2 &:= [[\{\rho \geq 0\}]]^2 \end{aligned}$$

を計算する。ここで、 f_L 、 f_{d-1} 、 f_{d-2} 、 $\dots$ は、推移的なフラグであることに注意する。

【0076】

4: $\text{mod } 2 \rightarrow \text{mod } p$ 変換により、 $[[f_1]]^2$ 、 $[[f_2]]^2$ 、 $\dots$ 、 $[[f_{d-1}]]^2$ 、 $[[f_L]]$ から $\langle \langle f_1 \rangle \rangle^P$ 、 $\langle \langle f_2 \rangle \rangle^P$ 、 $\dots$ 、 $\langle \langle f_{d-1} \rangle \rangle^P$ 、 $\langle \langle f_L \rangle \rangle^P$ を計算する。ただし、 $\langle \langle f_0 \rangle \rangle^P$ は不要である。なお、 $\text{mod } 2 \rightarrow \text{mod } P$ 変換は、公知の技術により行うことができる。例えば、参考文献1を利用する。

20

【0077】

5: $\langle \langle \rho' \rangle \rangle^P := \langle \langle \rho \rangle \rangle^P + M_{\max} - 1 - u \sum_{1 \leq i < d} \langle \langle f_i \rangle \rangle^P + ((d-1)u - M_{\max} + 1) \langle \langle f_L \rangle \rangle^P$ を計算する。

【0078】

6: $[[a]]^P$ と $\langle \langle \rho' \rangle \rangle^P$ とを用いて〈乗法的ローテーションプロトコル〉により $[[b]]^P := [[2\rho'a]]^P$ を計算する。ただし、乗法的ローテーションプロトコルとして、公知の技術を用いてもよい。

30

【0079】

7:一括シフト量公開右シフトにより、

$$\begin{aligned} [[c_0]]^P &:= [[2\rho'a / 2^{M_{\max} - 1}]]^P, \\ [[c_1]]^P &:= [[2\rho'a / (2^{M_{\max} - 1 - u})]]^P, \dots, \\ [[c_{d-1}]]^P &:= [[2\rho'a / (2^{M_{\max} - 1 - (d-1)u})]]^P \end{aligned}$$

を計算する。

【0080】

8: $\text{mod } 2 \rightarrow \text{mod } P$ 変換により $[[f_0]]^P$ 、 $[[f_1]]^P$ 、 $\dots$ 、 $[[f_{d-1}]]^P$ 、 $[[f_L]]^P$ を計算する。ここでは $[[f_0]]^P$ が必要である。

【0081】

9: 積和により $[[s]] := [[c_0]]^P [[f_0]]^P + ([c_1] - [c_0])^P [[f_1]]^P + \dots + ([c_{d-1}] - [c_{d-2}])^P [[f_{d-1}]]^P + ([b]]^P - [c_{d-1}])^P [[f_L]]^P$

40

を計算して出力する。この式は推移的なフラグに対する選択ゲートになっていることに注意する。

【0082】

以下、本実施形態で実現する固定小数点ベクトル積和について説明する。

【0083】

〈固定小数点ベクトル積和プロトコル〉

入力: 固定小数点ベクトル $[[\vec{a}]]^P$ 、 $[[\vec{b}]]^P$

パラメータ: ベクトル長 m

50

出力: $[[c]]^P$ 、ただし $\sum_{0 \leq i < m} a_i b_i = c$

1: 第一実施形態のベクトルMSB正規化プロトコルにより、固定小数点数ベクトル $[[\rightarrow a]]^P$ 、 $[[\rightarrow b]]^P$ をそれぞれベクトルMSB正規化し、MSB位置を調整したベクトルとシフト量、 $([[\rightarrow 2^{P-a} \rightarrow a]]^P, \langle \rho_a \rangle^P)$ 、 $([[\rightarrow 2^{P-b} \rightarrow b]]^P, \langle \rho_b \rangle^P)$ を得る。

【0084】

2: $\langle \rho_a \rangle^P, \langle \rho_b \rangle^P$ から $\text{mod } p \rightarrow \text{mod } Q$ 変換により、 $[[\rho_a]]^Q, [[\rho_b]]^Q$ を得る。
 なお、 $\text{mod } p \rightarrow \text{mod } Q$ 変換は、公知の技術により行うことができる。例えば、参考文献1を利用する。また、参考文献1以外のモジュラス変換を用いてもよい。例えば、参考文献1の技術は、空きビットが所定のビット数あること(以下、商転移の条件ともいう)を満たす必要があるが、商転移の条件を満たさないモジュラス変換を利用してもよい。以下、商転移の条件を満たさないモジュラス変換について説明する。

10

〈非商転移モジュラス変換プロトコル〉

入力: (k, n) -秘密分散したシェア $[[a]]^P$

パラメータ: p のビット数 $|p|$

出力: 異なる法 Q で (k, n) -秘密分散したシェア $[[a]]^Q$

2-1: シェア $[[a]]^P$ を (k, k) -加法的秘密分散したシェア $\langle a \rangle^P$ に変換する。 $k=2$ とし、パーティ p_0, p_1 がシェア $\langle a \rangle^P$ を持つ。 (k, n) -秘密分散から (k, k) -加法的秘密分散への変換は、公知の技術により行うことができる。例えば、参考文献1を利用する。

【0085】

2-2: パーティ p_0 は $a'_0 := \langle a \rangle^{p_0} + (2^{|p|-1})$ を Z 上加算により $\text{mod } p$ せずに計算し、 a'_0 の各ビットを (k, n) -秘密分散し、ビット表現のシェア $[[a'_0]]^{2^{|p|-1}}$ を得る。ビット分解は、公知の技術により行うことができる。例えば、参考文献1を利用する。

20

【0086】

2-3: パーティ p_1 は $\langle a \rangle^{p_1}$ の各ビットを (k, n) -秘密分散し、ビット表現のシェア $[[a_1]]^{2^{|p|-1}}$ を得る。

【0087】

2-4: 加算回路により $a'_0 + a_1$ のビット表現のシェア $[[a'_0 + a_1]]^{2^{|p|-1}}$ を得る。ここで、加算回路計算後はビット長が $|p|$ から $|p| + 1$ に1増える。

【0088】

2-5: $[[a'_0 + a_1]]^{2^{|p|-1}}$ の最上位ビットを $[[q]]^2$ とおく。 q はシェア $\langle a \rangle^P$ の商、すなわち $\langle a \rangle_0 + \langle a \rangle_1 = a + qp$ と表したときの q である。

30

【0089】

2-6: $\text{mod } 2 \rightarrow \text{mod } Q$ 変換により、 $[[q]]^2$ から $[[q]]^Q$ を得る。例えば、 $\text{mod } 2 \rightarrow \text{mod } Q$ 変換は、公知の技術により行うことができる。例えば、参考文献1を利用する。

【0090】

2-7: パーティ p_0, p_1 はそれぞれ $\langle a \rangle^{p_0}, \langle a \rangle^{p_1}$ から $\langle a \rangle^{p_0} \text{ mod } Q, \langle a \rangle^{p_1} \text{ mod } Q$ を得、 $\langle a' \rangle^Q$ とする。ここで、 $a' = a + qp \text{ mod } Q$ が成り立っている。

【0091】

2-8: (k, k) -秘密分散したシェア $\langle a' \rangle^Q$ を (k, n) -秘密分散に変換し、 (k, n) -秘密分散したシェア $[[a']]^Q$ を得る。 (k, k) -加法的秘密分散から (k, n) -秘密分散への変換は、公知の技術により行うことができる。例えば、参考文献1を利用する。

40

【0092】

2-9: $[[a]]^Q = [[a']]^Q - p[[q]]^Q$ を計算して出力する。

例えば、 $p=61$ の場合、空きビットを残すためには値が31までしかとれないため、ここでの、 $\text{mod } p \rightarrow \text{mod } Q$ 変換は、商転移を使える条件を満たさないことが多いと想定される。そのため、非商転移モジュラス変換プロトコルを利用するとよい。

【0093】

3: $[[c]]^P := [[\sum_{0 \leq i < m} 2^{P-a} a_i 2^{P-b} b_i]]^P$ を計算する。

【0094】

4: $[[(-\rho_a - \rho_b)]]^Q$ を計算し、変換により $\langle -\rho_a - \rho_b \rangle^Q$ を得る。

50

【0095】

5: 上述の〈シフト量秘匿左右シフトプロトコル〉により、 $[[c]]^P$ を $\langle \langle -p_a - p_b \rangle \rangle^Q$ でシフトし、 $[[c]]^P$ を $(-p_a - p_b)$ ビットシフトした値を出力する。

【0096】

以下、上述の〈固定小数点ベクトル積和プロトコル〉を実現するベクトルMSB正規化システムについて説明する。

【0097】

＜第二実施形態に係るベクトルMSB正規化システム＞

図1は第二実施形態に係るベクトルMSB正規化システム1の構成例を、図4はベクトルMSB正規化システム1の処理フローの例を示す。

【0098】

ベクトルMSB正規化システム1は、二つの固定小数点ベクトル $[[\rightarrow a]]^P, [[\rightarrow b]]^P$ を入力とし、要素の積和 $[[c]]^P$ を求め、出力する。ただし $\sum_{0 \leq i < m} a_i b_i = c$ である。ベクトル $[[\rightarrow a]]^P, [[\rightarrow b]]^P$ のベクトル長 m をパラメータとする。

【0099】

＜分散処理装置100-r＞

図5は、分散処理装置100-rの機能ブロック図の例を示す。

【0100】

分散処理装置100-rは、ビット分解部101、論理和取得部103、シフト量取得部105およびシフト部107に加え、モジュラス変換部109、積和計算部111、秘密分散変換部113およびシフト量秘匿左右シフト部115を含む。

【0101】

以下、図4を用いて各部の処理について説明する。

【0102】

S101～S107については第一実施形態で説明したとおりである。ベクトルMSB正規化システム1は、固定小数点ベクトル $[[\rightarrow a]]^P, [[\rightarrow b]]^P$ を入力とし、ベクトルMSB正規化し、ベクトルMSB正規化後のベクトルとシフト量 $([[\rightarrow 2^{P-a} \rightarrow a]]^P, \langle \langle p_a \rangle \rangle^P), ([[\rightarrow 2^{P-b} \rightarrow b]]^P, \langle \langle p_b \rangle \rangle^P)$ を求める。S109以降の処理について説明する。

【0103】

＜モジュラス変換部109＞

n 個のモジュラス変換部109は、 $([[\rightarrow 2^{P-a} \rightarrow a]]^P, \langle \langle p_a \rangle \rangle^P), ([[\rightarrow 2^{P-b} \rightarrow b]]^P, \langle \langle p_b \rangle \rangle^P)$ を受け取り、 $\langle \langle p_a \rangle \rangle^P, \langle \langle p_b \rangle \rangle^P$ から $\text{mod } p \rightarrow \text{mod } Q$ 変換により、 $[[p_a]]^Q, [[p_b]]^Q$ を得る(S109)。

【0104】

＜積和計算部111＞

n 個の積和計算部111は、シェア $[[\rightarrow 2^{P-a} \rightarrow a]]^P$ および $([[\rightarrow 2^{P-b} \rightarrow b]]^P)$ を受け取り、積和 $[[c]]^P := [[\sum_{0 \leq i < m} 2^{P-a} a_i 2^{P-b} b_i]]^P$ を計算する(S111)。

【0105】

＜秘密分散変換部113＞

n 個の秘密分散変換部113は、 $[[p_a]]^Q, [[p_b]]^Q$ を受け取り、 $[-p_a - p_b]^Q$ を計算し、秘密分散変換により $\langle \langle -p_a - p_b \rangle \rangle^Q$ を得る(S113)。

【0106】

＜シフト量秘匿左右シフト部115＞

シフト量秘匿左右シフト部115は、積和のシェア $[[c]]^P$ とシフト量のシェア $\langle \langle -p_a - p_b \rangle \rangle^Q$ を受け取り、上述の〈シフト量秘匿左右シフトプロトコル〉により、 $[[c]]^P$ を $\langle \langle -p_a - p_b \rangle \rangle^Q$ ビットシフトし(S115)、シフト後の値を出力する。なお、上述の〈シフト量秘匿左右シフトプロトコル〉を使わずに、積和のシェア $[[c]]^P$ とシフト量のシェア $\langle \langle -p_a - p_b \rangle \rangle^Q$ を用いて公知の技術により、 $[[c]]^P$ を $\langle \langle -p_a - p_b \rangle \rangle^Q$ ビットシフトした値を求めてもよい。

【0107】

10

20

30

40

50

＜第三実施形態＞

第一実施形態と異なる部分を中心に説明する。

【0108】

第三実施形態では、第一実施形態のベクトルMSB正規化を利用した浮動小数点ベクトル積和について説明する。まず、第三実施形態に係る浮動小数点ベクトル積和で使われるいくつかのプロトコルについて説明する。

【0109】

＜浮動小数点ベクトルの指数部統一プロトコル＞

入力:浮動小数点ベクトル $([[\rightarrow a]]^P, [[\rightarrow \rho_a]]^Q)$ 、ただし、本実施形態では、仮数部を a とし、指数部を ρ_a とし、実数 x を、 $x = 2^{\rho_a} a$ のように表す。 $[[\rightarrow a]]^P = ([[a_0]]^P, \dots, [[a_{m-1}]]^P)$ 、 $[[\rightarrow \rho_a]]^Q = ([[\rho_{a_0}]]^Q, \dots, [[\rho_{a_{m-1}}]]^Q)$ であり、浮動小数点ベクトル $([[\rightarrow a]]^P, [[\rightarrow \rho_a]]^Q)$ は、 $i (0 \leq i < m-1)$ 番目の実数を $2^{\rho_a - (a-i)} a_i$ のように表す。

10

【0110】

出力: $([[\rightarrow b]]^P, [[\rho_{\max}]]^Q)$ 、ただし各 i 番目の要素に対して $2^{\rho_a - (a-i)} a_i \cdot 2^{\rho_{\max} - b_i}$
処理:浮動小数点ベクトル $([[\rightarrow a]]^P, [[\rightarrow \rho_a]]^Q)$ の指数部 $[[\rightarrow \rho_a]]^Q$ を最も大きい値 $[[\rho_{\max}]]^Q$ に統一し、仮数部 $[[\rightarrow a]]^P$ を差分 $[[\rightarrow \rho_{\text{dif}}]]^Q := [[\rightarrow \rho_a]]^Q - [[\rho_{\max}]]^Q$ だけ右シフトし、指数部を統一した浮動小数点ベクトルを求める。

【0111】

1:最大値計算により、 $[[\rightarrow \rho_a]]^Q$ に含まれる全ての要素の中から最も大きい値を $[[\rho_{\max}]]^Q$ として得る。

20

【0112】

2: $[[\rightarrow \rho_{\text{dif}}]]^Q := [[\rightarrow \rho_a]]^Q - [[\rho_{\max}]]^Q$ を計算する。 $[[\rightarrow \rho_a]]^Q$ の各要素から $[[\rho_{\max}]]^Q$ を減算する。

【0113】

3:上述の＜シフト量秘匿左右シフトプロトコル＞により、 $[[\rightarrow a]]^P$ の各要素を $[[\rightarrow \rho_{\text{dif}}]]^Q$ の各要素でシフトし、 $[[\rightarrow b]]^P$ とする。ただし ρ_{dif} の各要素が非負なので右シフトとなるため、左シフトの分岐は省いてよい。

【0114】

4: $([[\rightarrow b]]^P, [[\rho_{\max}]]^Q)$ を出力する。

【0115】

以下、本実施形態で実現する浮動小数点ベクトル積和について説明する。

30

【0116】

＜浮動小数点ベクトル積和プロトコル＞

入力:浮動小数点ベクトル $([[\rightarrow a]]^P, [[\rightarrow \rho_a]]^Q)$ 、 $([[\rightarrow b]]^P, [[\rightarrow \rho_b]]^Q)$

パラメータ:ベクトル長 m

出力: $([[c]]^P, \langle \rho_b \rangle^Q)$ 、ただし $\sum_{0 \leq i < m} 2^{(\rho_a - i) - (\rho_b - i)} a_i b_i \cdot 2^{\rho_b - b}$ とする。

【0117】

1:上述の＜ベクトルMSB正規化プロトコル＞により $[[\rightarrow a]]^P$ 、 $[[\rightarrow b]]^P$ のMSB位置を調整したベクトルとシフト量、 $([[\rightarrow a']]]^P, \langle \rho'_a \rangle^P)$ 、 $([[\rightarrow b']]]^P, \langle \rho'_b \rangle^P)$ を得る。

【0118】

2: $\text{mod } p \rightarrow \text{mod } Q$ 変換により $[[\rho_{a'}]]^Q$ 、 $[[\rho_{b'}]]^Q$ を得る。

40

【0119】

3:上述の＜浮動小数点ベクトルの指数部統一プロトコル＞により、 $([[\rightarrow a'']]^P, [[\rho_{a' - \rho_{a'}}]]^Q)$ 、 $([[\rightarrow b'']]^P, [[\rho_{b' - \rho_{b'}}]]^Q)$ の指数部を統一したベクトルと指数部、 $([[\rightarrow a'']]^P, [[\rho_{a''}]]^Q)$ 、 $([[\rightarrow b'']]^P, [[\rho_{b''}]]^Q)$ を得る。

【0120】

4: $[[c]]^P := [\sum_{0 \leq i < m} a''_i b''_i]^P$ を計算し、 $([[c]]^P, [[\rho_{a''} + \rho_{b''}]]^Q)$ を得る。

【0121】

さらに、入力のビット数がある程度既知である場合や、MSBが調整してあるなどの理由で a 、 b のビット数がある程度高いことが既知である場合には、公知のシフト量公開右シフ

50

トにより所定のビット数 σ で右シフトする。

【0122】

一方、ビット数が不明な場合には、第一実施例と同様の方法によりMSBを固定位置に合わせ、その後、公知のシフト量公開右シフトにより適切なビット位置に右シフトする。右シフト量を $[[\sigma]]^Q$ とする。

【0123】

以下、上述の〈浮動小数点ベクトル積和プロトコル〉を実現するベクトルMSB正規化システムについて説明する。

【0124】

＜第三実施形態に係るベクトルMSB正規化システム＞

10

図1は第二実施形態に係るベクトルMSB正規化システム1の構成例を、図6はベクトルMSB正規化システム1の処理フローの例を示す。

【0125】

ベクトルMSB正規化システム1は、二つの浮動小数点ベクトル $([[\rightarrow a]]^P, [[\rightarrow p_a]]^Q), ([[\rightarrow b]]^P, [[\rightarrow p_b]]^Q)$ を入力とし、要素の積和 $([[c]]^P, \langle \langle p_c \rangle \rangle^Q)$ を求め、出力する。ただし $\sum_{0 \leq i < m} 2^{(p-a)-i(p-b)-i} a_i b_i = 2^{p-c}$ である。ベクトル $[[\rightarrow a]]^P, [[\rightarrow b]]^P$ のベクトル長 m をパラメータとする。

【0126】

＜分散処理装置100-r＞

図7は、分散処理装置100-rの機能ブロック図の例を示す。

20

【0127】

分散処理装置100-rは、ビット分解部101、論理和取得部103、シフト量取得部105およびシフト部107に加え、モジュラス変換部117、指数統一部119および積和部121を含む。

【0128】

以下、図6を用いて各部の処理について説明する。

【0129】

S101～S107については第一実施形態で説明したとおりである。ベクトルMSB正規化システム1は、二つの浮動小数点ベクトル $([[\rightarrow a]]^P, [[\rightarrow p_a]]^Q), ([[\rightarrow b]]^P, [[\rightarrow p_b]]^Q)$ を入力とし、 $[[\rightarrow a]]^P, [[\rightarrow b]]^P$ をベクトルMSB正規化し、ベクトルMSB正規化後のベクトルとシフト量 $([[\rightarrow a']]^P, \langle \langle p'_a \rangle \rangle^P), ([[\rightarrow b']]^P, \langle \langle p'_b \rangle \rangle^P)$ を求める。S117以降の処理について説明する。

30

【0130】

＜モジュラス変換部117＞

n 個のモジュラス変換部117は、 $\langle \langle p'_a \rangle \rangle^P$ と $\langle \langle p'_b \rangle \rangle^P$ を受け取り、 $\text{mod } p \rightarrow \text{mod } Q$ 変換により $[[p_a']]^Q, [[p_b']]^Q$ を得る(S117)。

【0131】

＜指数統一部119＞

n 個の指数統一部119は、二つの浮動小数点ベクトル $([[\rightarrow a]]^P, [[\rightarrow p_a]]^Q), ([[\rightarrow b]]^P, [[\rightarrow p_b]]^Q)$ の指数部 $[[\rightarrow p_a]]^Q, [[\rightarrow p_b]]^Q$ と、ベクトルMSB正規化後のベクトル $[[\rightarrow a']]^P, [[\rightarrow b']]^P$ と、 $\text{mod } p \rightarrow \text{mod } Q$ 変換後のシフト量 $[[p_a']]^Q, [[p_b']]^Q$ を受け取り、上述の〈浮動小数点ベクトルの指数部統一プロトコル〉により、 $([[\rightarrow a']]^P, [[\rightarrow p_a - p_a']]^Q), ([[\rightarrow b']]^P, [[\rightarrow p_b - p_b']]^Q)$ の指数部を統一したベクトルと指数部、 $([[\rightarrow a'']]^P, [[p_a'']]^Q), ([[\rightarrow b'']]^P, [[p_b'']]^Q)$ を得る(S119)。

40

【0132】

＜積和部121＞

n 個の積和部121は、 $[[c]]^P := [[\sum_{0 \leq i < m} a''_i b''_i]]^P$ を計算し、 $([[c]]^P, [[p_{a''} + p_{b''}]]^Q)$ を得る(S121)。

【0133】

(処理効率)

50

アルゴリズムの処理効率に関して、要素演算である乗法的ローテーション、フラグ列→数値変換、シフト量秘匿左右シフトプロトコルおよび、比較用に浮動小数点加算・乗算について評価する。

【0134】

(1)乗法的ローテーション:通信量 $(4/3) \lceil P \rceil$ ビット、2ラウンド

(2)フラグ列→数値変換:通信量 $(4/3) \lceil L \rceil$ ビット、2ラウンド

(4)シフト量秘匿左右シフトプロトコル-その2-:通信量 $((5/3)d + (10/3)) \lceil P \rceil + (2d+1) \lceil p \rceil$ 、ラウンド数 $\lambda + 4$

(5)浮動小数点加算:通信量 $((5/3)d + (19/3)) \lceil P \rceil + 3 \lceil Q \rceil + 2\lambda + (4d+1) \lceil p \rceil$ 、ラウンド数 $2\lambda + 7$

10

(6)浮動小数点乗算:通信量 $(8/3) \lceil P \rceil$ 、3ラウンド

d はシフト量秘匿左右シフトプロトコル中の分割数 d である。

【0135】

比較対象として、参考文献2は、加算については2つ方式があり、通信量は対数以下のコストを丸めれば、 $22 \lceil P \rceil + 5 \lceil Q \rceil + O(\log \lceil P \rceil + \log \lceil Q \rceil)$ と表せる。

【0136】

(参考文献2)西出隆志、天田拓磨、「通信量を削減した浮動小数点演算のためのマルチパーティ計算」、情報処理学会論文誌、Vol.60、No.9, pp. 1433-1447 (2019).

ラウンド数については良い方で、定数の42である。乗算に関しては通信量 $12 \lceil P \rceil + O(1)$ 、

ラウンド数は定数の23である。加算は、 d が典型的には1であることを考えると、 $6 \lceil P \rceil + 3 \lceil Q \rceil + 2\lambda + 5 \lceil p \rceil$ である。

20

$\lceil P \rceil = 61$ 、 $\lceil Q \rceil = 13$ 、 $\lambda = 10$ 、 $\lceil p \rceil = 6$ であることを考えると、3倍程度、本方式が効率的である。加算は複雑なため、要素であるシフトを高速化してもまだ桁違いにはならないようである。乗算に関しては5倍程度高速である。

【0137】

<実機性能評価>

実機実験の結果を報告する。下記のマシン3台の、マルチパーティ計算である。

【0138】

◎CPU: Xeon Gold 6144 3.5GHz, 6cores x 2 sockets

◎memory: 768GB

◎NW: 10Gbps リングトポロジ

30

◎OS: CentOS 7.3

図8は各演算の性能である。

【0139】

パラメータとしてMSB位置の上限が重要になるが、これは28bit(0スタート表記のため、29bit数)とした。符号付きでmod P で商転移を用いることができる最大のMSB位置が57であり、その半分以内という条件である。28bitは単精度を超えており、多くのアプリケーションで十分と考える。

【0140】

積和演算としては実際には、行列乗算を選択した。行列乗算は積和から成り、かつ機械学習などで極めて重要だからである。具体的には、左の行列を行数100とし、行数×列数=“件数”となるようにし、右の行列は左の列数を長さとするベクトルとした。処理量は、サイズを列数とする積和を、行数分だけ繰り返す処理量に等しい。

40

【0141】

スケールとして、1000件、100万件、1000万件の3つと、遅延を100msと極大にすることで実ラウンド数を計測したものを記した。また、passiveモデルの他にactiveモデルの場合の性能も示した(プロトコルはpassive版から拡張したものである)。activeモデルのセキュリティパラメータは8bitであり、攻撃検知率は約99%である。計算量的安全性と異なりオフライン攻撃が不可能なため、この確率は攻撃を抑止するには十分である。

【0142】

<その他の変形例>

50

本発明は上記の実施形態及び変形例に限定されるものではない。例えば、上述の各種の処理は、記載に従って時系列に実行されるのみならず、処理を実行する装置の処理能力あるいは必要に応じて並列的にあるいは個別に実行されてもよい。その他、本発明の趣旨を逸脱しない範囲で適宜変更が可能である。

【0143】

＜プログラム及び記録媒体＞

上述の各種の処理は、図9に示すコンピュータの記憶部2020に、上記方法の各ステップを実行させるプログラムを読み込ませ、制御部2010、入力部2030、出力部2040などに動作させることで実施できる。

【0144】

この処理内容を記述したプログラムは、コンピュータで読み取り可能な記録媒体に記録しておくことができる。コンピュータで読み取り可能な記録媒体としては、例えば、磁気記録装置、光ディスク、光磁気記録媒体、半導体メモリ等のようなものでもよい。

【0145】

また、このプログラムの流通は、例えば、そのプログラムを記録したDVD、CD-ROM等の可搬型記録媒体を販売、譲渡、貸与等することによって行う。さらに、このプログラムをサーバコンピュータの記憶装置に格納しておき、ネットワークを介して、サーバコンピュータから他のコンピュータにそのプログラムを転送することにより、このプログラムを流通させる構成としてもよい。

【0146】

このようなプログラムを実行するコンピュータは、例えば、まず、可搬型記録媒体に記録されたプログラムもしくはサーバコンピュータから転送されたプログラムを、一旦、自己の記憶装置に格納する。そして、処理の実行時、このコンピュータは、自己の記録媒体に格納されたプログラムを読み取り、読み取ったプログラムに従った処理を実行する。また、このプログラムの別の実行形態として、コンピュータが可搬型記録媒体から直接プログラムを読み取り、そのプログラムに従った処理を実行することとしてもよく、さらに、このコンピュータにサーバコンピュータからプログラムが転送されるたびに、逐次、受け取ったプログラムに従った処理を実行することとしてもよい。また、サーバコンピュータから、このコンピュータへのプログラムの転送は行わず、その実行指示と結果取得のみによって処理機能を実現する、いわゆるASP（Application Service Provider）型のサービスによって、上述の処理を実行する構成としてもよい。なお、本形態におけるプログラムには、電子計算機による処理の用に供する情報であってプログラムに準ずるもの（コンピュータに対する直接の指令ではないがコンピュータの処理を規定する性質を有するデータ等）を含むものとする。

【0147】

また、この形態では、コンピュータ上で所定のプログラムを実行させることにより、本装置を構成することとしたが、これらの処理内容の少なくとも一部をハードウェア的に実現することとしてもよい。

10

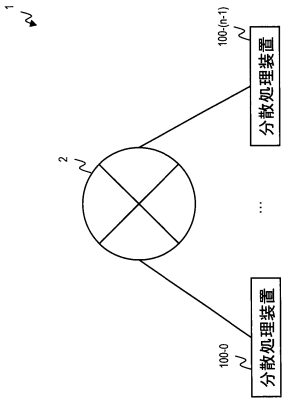
20

30

40

50

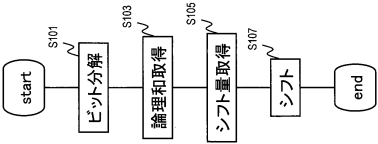
【図面】
【図 1】



【図 2】

図1

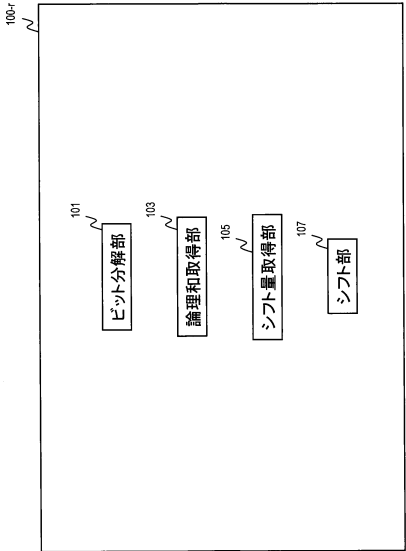
図2



10

20

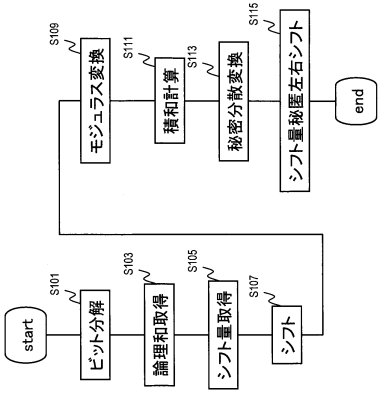
【図 3】



【図 4】

図3

図4

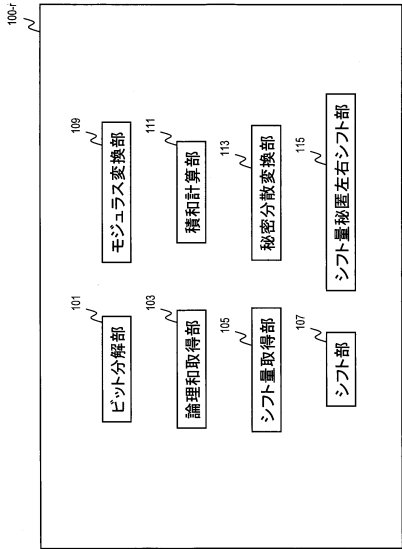


30

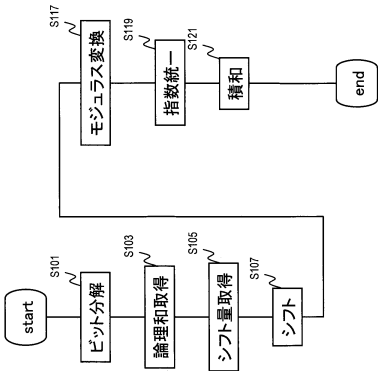
40

50

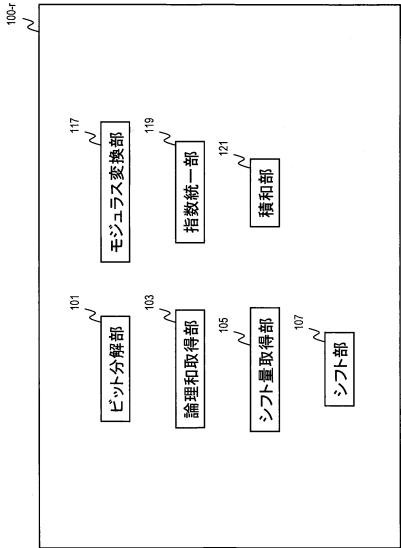
【図 5】



【図 6】



【図 7】



【図 8】

件数	passive				active			
	1000	100万	1000万	10000万	1000	100万	1000万	10000万
ベクトルMSB正規化	0.23	19.03	17.53	14	0.07	5.56	5.41	24
固定小数点行列乗算	0.03	9.58	10.33	102	0.002	0.93	1.59	123
浮動小数点行列乗算	0.03	7.01	7.05	123	0.003	0.63	1.35	152

図 8

図 7

図 5

図 6

10

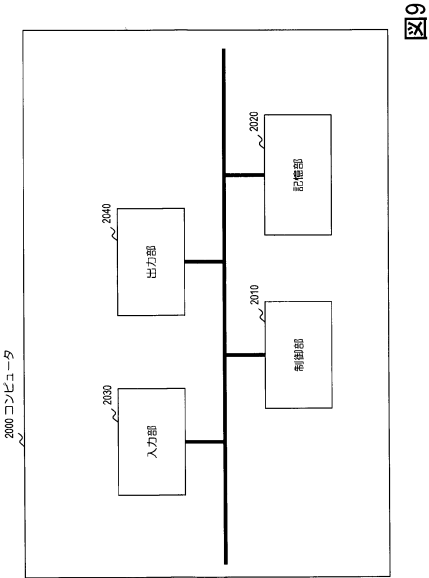
20

30

40

50

【図 9】



10

20

30

40

50

フロントページの続き

(56)参考文献 国際公開第 2 0 1 8 / 0 3 4 0 7 9 (WO, A 1)
KAMM, Liina et al., Secure Floating-Point Arithmetic and Private Satellite Collision Analysis
, Version:20131217:154023, 2013年12月16日, <<https://eprint.iacr.org/2013/850.pdf>>
五十嵐 大, 秘密計算 A I の実装に向けた秘密実数演算群の設計と実装, コンピュータセキュリティシンポジウム2019論文集, 日本, 情報処理学会, 2019年10月14日, No.2019, p
p.1557-1564, ISSN 1881-0840
(58)調査した分野 (Int.Cl., D B 名)
G 0 9 C 1 / 0 0