(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7517475号
(P7517475)

(45)発行日　令和6年7月17日(2024.7.17)

(24)登録日　令和6年7月8日(2024.7.8)

(51)国際特許分類
Ｇ０６Ｆ　21/60　(2013.01)

ＦＩ
Ｇ０６Ｆ　21/60

請求項の数　10　（全27頁）

(21)出願番号　特願2022-574879(P2022-574879)
(86)(22)出願日　令和3年1月12日(2021.1.12)
(86)国際出願番号　PCT/JP2021/000661
(87)国際公開番号　WO2022/153358
(87)国際公開日　令和4年7月21日(2022.7.21)
審査請求日　令和5年7月11日(2023.7.11)

(73)特許権者　000004237
日本電気株式会社
東京都港区芝五丁目７番１号
(74)代理人　100080816
弁理士　加藤 朝道
(74)代理人　100098648
弁理士　内田 潔人
(72)発明者　土田 光
東京都港区芝五丁目７番１号 日本電気株式会社内
審査官　中里 裕正

最終頁に続く

(54)【発明の名称】　秘密計算システム、秘密計算サーバ装置、秘密計算方法および秘密計算プログラム

(57)【特許請求の範囲】

【請求項１】

相互にネットワークで接続した５台の秘密計算サーバ装置を備え、秘密分散して保持されている位数２の剰余類環上の論理シェアから位数ｎ（ｎ＝２^{m}；ｍは２以上の整数）の剰余類環上の算術シェアにビット変換する秘密計算システムであって、

前記秘密計算サーバ装置のそれぞれが、

前記論理シェアから、自己が保持していないサブシェアをゼロに設定することで他の秘密計算サーバ装置との通信を伴うことなく算術シェアを計算するローカル再分散部と、

前記論理シェアをビット変換した算術シェアを得るために、前記ローカル再分散部が得た算術シェアを用いて通信を伴った秘密計算を行う秘密計算部と、

少なくとも３台以上の秘密計算サーバ装置から受信した同一の値となるはずの受信値を比較し、少なくとも２つ以上が同一である受信値を正しい値として採用する比較検証部と、

を有し、

前記通信を伴った秘密計算における受信値を前記比較検証部が検証する、秘密計算システム。

【請求項２】

前記秘密計算サーバ装置のそれぞれが、

前記論理シェアにおけるサブシェアから計算された一時的変数を算術シェアとして分散する再分散部をさらに有し、

前記再分散部が分散した前記一時的変数の算術シェアの受信値を前記比較検証部が検証

し、
　前記再分散部が分散した前記一時的変数の算術シェアと前記ローカル再分散部が得た算術シェアとを用いて、前記論理シェアをビット変換した算術シェアを秘密計算する、請求項１に記載の秘密計算システム。
【請求項３】
　前記一時的変数は、前記秘密計算サーバ装置の５台のうち３台が共通して保持している前記論理シェアにおけるサブシェアから計算され、
　前記再分散部は、前記３台の秘密計算サーバ装置が共通して計算する前記一時的変数から決定的に生成した算術シェアを分散し、
　前記比較検証部は、前記３台の秘密計算サーバ装置から受信した算術シェアに対して、少なくとも２つ以上が同一である受信値を正しい値として採用する、請求項２に記載の秘密計算システム。
【請求項４】
　前記比較検証部は、前記受信値のハッシュ値が同一であることを判断して、前記受信値が正しい値であることを判断する、請求項１から請求項３のいずれか１項に記載の秘密計算システム。
【請求項５】
　秘密分散して保持されている位数２の剰余類環上の論理シェアから位数ｎ（ｎ＝２^{m}；ｍは２以上の整数）の剰余類環上の算術シェアにビット変換するために、相互にネットワークで接続した少なくとも５台の秘密計算サーバ装置の一つであって、
　前記論理シェアから、自己が保持していないサブシェアをゼロに設定することで他の秘密計算サーバ装置との通信を伴うことなく算術シェアを計算するローカル再分散部と、
　前記論理シェアをビット変換した算術シェアを得るために、前記ローカル再分散部が得た算術シェアを用いて通信を伴った秘密計算を行う秘密計算部と、
　少なくとも３台以上の秘密計算サーバ装置から受信した同一の値となるはずの受信値を比較し、少なくとも２つ以上が同一である受信値を正しい値として採用する比較検証部と、
　を有し、
　前記通信を伴った秘密計算における受信値を前記比較検証部が検証する、秘密計算サーバ装置。
【請求項６】
　相互にネットワークで接続した５台の秘密計算サーバ装置を用いて、秘密分散して保持されている位数２の剰余類環上の論理シェアから位数ｎ（ｎ＝２^{m}；ｍは２以上の整数）の剰余類環上の算術シェアにビット変換する秘密計算方法であって、
　前記秘密計算サーバ装置のそれぞれが、
　前記論理シェアから、自己が保持していないサブシェアをゼロに設定することで他の秘密計算サーバ装置との通信を伴うことなく算術シェアに再分散し、
　前記論理シェアをビット変換した算術シェアを得るために、前記再分散された算術シェアを用いて通信を伴った秘密計算を行い、
　少なくとも３台以上の秘密計算サーバ装置から受信した同一の値となるはずの受信値を比較し、少なくとも２つ以上が同一である受信値を正しい値として採用することで、前記通信を伴った秘密計算における受信値を検証する、秘密計算方法。
【請求項７】
　前記秘密計算サーバ装置のそれぞれが、
　前記論理シェアにおけるサブシェアから計算された一時的変数を算術シェアとして再分散し、
　少なくとも３台以上の秘密計算サーバ装置から受信した同一の値となるはずの受信値を比較し、少なくとも２つ以上が同一である受信値を正しい値として採用することで、前記再分散した前記一時的変数の算術シェアの受信値を検証し、
　前記再分散した前記一時的変数の算術シェアと前記通信を伴うことなく計算した算術シェアとを用いて、前記論理シェアをビット変換した算術シェアを秘密計算する、請求項６

に記載の秘密計算方法。
【請求項８】
前記一時的変数を、前記秘密計算サーバ装置の５台のうち３台が共通して保持している前記論理シェアにおけるサブシェアから計算し、
前記３台の秘密計算サーバ装置が共通して計算する前記一時的変数から決定的に生成した算術シェアを再分散し、
前記３台の秘密計算サーバ装置から受信した算術シェアに対して、少なくとも２つ以上が同一である受信値を正しい値として採用する、請求項７に記載の秘密計算方法。
【請求項９】
前記受信値が正しい値であることを判断する際には、前記受信値のハッシュ値が同一であることを用いて判断する、請求項６から請求項８のいずれか１項に記載の秘密計算方法。
【請求項１０】
相互にネットワークで接続した少なくとも５台以上の秘密計算サーバ装置に、秘密分散して保持されている値の秘密計算をさせる秘密計算プログラムであって、
相互にネットワークで接続した５台の秘密計算サーバ装置に、秘密分散して保持されている位数２の剰余類環上の論理シェアから位数ｎ（$n=2^m$；ｍは２以上の整数）の剰余類環上の算術シェアにビット変換をさせる秘密計算プログラムであって、
前記秘密計算サーバ装置のそれぞれが、
前記論理シェアから、自己が保持していないサブシェアをゼロに設定することで他の秘密計算サーバ装置との通信を伴うことなく算術シェアに再分散し、
前記論理シェアをビット変換した算術シェアを得るために、前記再分散された算術シェアを用いて通信を伴った秘密計算を行い、
少なくとも３台以上の秘密計算サーバ装置から受信した同一の値となるはずの受信値を比較し、少なくとも２つ以上が同一である受信値を正しい値として採用することで、前記通信を伴った秘密計算における受信値を検証する、秘密計算プログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、秘密計算システム、秘密計算サーバ装置、秘密計算方法および秘密計算プログラムに関するものである。
【背景技術】
【０００２】
近年、秘密計算と呼ばれる技術の研究開発が盛んに行われている。秘密計算は、第三者に対して計算過程とその結果を秘密にしつつ所定の処理を実行する技術の一つである。秘密計算における代表的な技術の一つとして、マルチパーティ計算技術が挙げられる。マルチパーティ計算技術では、秘密にするデータを複数のサーバ（秘密計算サーバ装置）に分散配置し、秘密にした状態を維持しながら当該データの任意の演算を実行する。なお、各秘密計算サーバ装置に分散配置したデータをシェアと呼ぶ。以下、特に断りがない限り、「秘密計算」という語を用いた場合は、マルチパーティ計算技術を意味するものとする。
【０００３】
このような秘密計算には、四則演算だけではなく、特定用途の計算プロトコルも実装することが一般的である。その特定用途の計算プロトコルの一つとして、ビット変換プロトコルがある。ビット変換は、法の変換を伴う型変換のことであり、例えば、位数２の剰余類環Z_2上のシェアから位数ｎの剰余類環Z_n上のシェアを得る変換が該当する。このようなビット変換は、例えば、算術演算と論理演算の混合回路における計算効率を向上させることができる。
【０００４】
算術演算と論理演算の混合回路の簡単な例としてハミング距離(Hamming distance)の計算がある。ハミング距離とは、２つの２進数の比較において異なっている桁の個数のことであり、例えば１１１１１１１と１０１０１０１とのハミング距離は３である。このよ

うなハミング距離の計算では、桁が異なっているか否かは排他的論理和であるので論理演算とし、異なっている桁の個数を集計するのは算術演算とすることが好ましい。ビット変換のためのプロトコルを備えた秘密計算は、このような算術演算と論理演算の混合回路における秘密計算の処理をそれぞれに適した法で計算することができるので、計算効率を向上させることが可能である。

【先行技術文献】

【非特許文献】

【０００５】

【文献】Byali, M., Chaudhari, H., Patra, A., & Suresh, A. (2020). FLASH: fast and robust framework for privacy-preserving machine learning. Proceedings on Privacy Enhancing Technologies, 2020(2), 459-480.

【発明の概要】

【発明が解決しようとする課題】

【０００６】

なお、上記先行技術文献の開示を、本書に引用をもって繰り込むものとする。以下の分析は、本発明者らによってなされたものである。

【０００７】

ところで、一般に秘密計算と呼ばれる技術の中にも、達成されている安全性の程度には高低がある。例えば、秘密計算を行うマルチパーティの参加者の中に不正者が紛れたとする。その場合に、不正者の存在を検知し処理を中断することができる秘密計算の技術と、たとえ不正者が存在しても処理を中断することなく正しい計算結果を得ることができる秘密計算の技術とでは、後者の方が前者よりも安全性が高い。そして、後者の安全性を満たす秘密計算はGuaranteed Output Delivery (GOD)と呼ばれ、これを実現する秘密計算の例も知られている（例えば、非特許文献１参照）。

【０００８】

また、秘密計算における安全性の評価には、達成できる安全性の効果だけではなく、前提条件も重要な意味を持つ。代表的な前提条件としてハッシュ関数のランダムオラクルモデルないしランダムオラクル仮定がある。

【０００９】

ハッシュ関数は、入力に対し一意の出力を返す関数であるが、出力から入力を推定することが困難であるように構成されている。ここで、出力から入力を推定することが困難であるとはいうものの、絶対に不可能であるかというとその保証はできない。そこで、安全性の評価に際し、用いられているハッシュ関数が脆弱性を持たないという前提で安全性が評価される。この前提の安全性を「ランダムオラクルモデルにおいて安全」あるいは「ランダムオラクル仮定のもとで安全」という。そして、非特許文献１における秘密計算の安全性は「ランダムオラクルモデルにおいて安全」である。

【００１０】

一方、「ランダムオラクルモデルにおいて安全」の対義語は、「標準モデルにおいて安全」である。すなわち、ハッシュ関数の出力から入力を推定することができたとしても、そのこと自体が秘密計算の脆弱性とはならないことをいう。当然のことながら、達成できる安全性が同じであれば、ランダムオラクルモデルにおける安全性よりも、標準モデルにおける安全性の方が高度な安全性が達成できることになる。したがって、ビット変換のプロトコルにおいても標準モデルにてGuaranteed Output Delivery (GOD)を達成することが望ましいことになる。

【００１１】

本発明の目的は、上述した課題を鑑み、標準モデルにおいてGuaranteed Output Delivery (GOD)を達成するビット変換に寄与する秘密計算システム、秘密計算サーバ装置、秘密計算方法および秘密計算プログラムを提供することである。

【課題を解決するための手段】

【００１２】

本発明の第１の視点では、相互にネットワークで接続した５台の秘密計算サーバ装置を備え、秘密分散して保持されている位数２の剰余類環上の論理シェアから位数ｎ（ｎ＝２^{m}；ｍは２以上の整数）の剰余類環上の算術シェアにビット変換する秘密計算システムであって、前記秘密計算サーバ装置のそれぞれが、前記論理シェアから、自己が保持していないサブシェアをゼロに設定することで他の秘密計算サーバ装置との通信を伴うことなく算術シェアを計算するローカル再分散部と、前記論理シェアをビット変換した算術シェアを得るために、前記ローカル再分散部が得た算術シェアを用いて通信を伴った秘密計算を行う秘密計算部と、少なくとも３台以上の秘密計算サーバ装置から受信した同一の値となるはずの受信値を比較し、少なくとも２つ以上が同一である受信値を正しい値として採用する比較検証部と、を有し、前記通信を伴った秘密計算における受信値を前記比較検証部が検証する、秘密計算システムが提供される。

【００１３】

本発明の第２の視点では、秘密分散して保持されている位数２の剰余類環上の論理シェアから位数ｎ（ｎ＝２^{m}；ｍは２以上の整数）の剰余類環上の算術シェアにビット変換するために、相互にネットワークで接続した少なくとも５台の秘密計算サーバ装置の一つであって、前記論理シェアから、自己が保持していないサブシェアをゼロに設定することで他の秘密計算サーバ装置との通信を伴うことなく算術シェアを計算するローカル再分散部と、前記論理シェアをビット変換した算術シェアを得るために、前記ローカル再分散部が得た算術シェアを用いて通信を伴った秘密計算を行う秘密計算部と、少なくとも３台以上の秘密計算サーバ装置から受信した同一の値となるはずの受信値を比較し、少なくとも２つ以上が同一である受信値を正しい値として採用する比較検証部と、を有し、前記通信を伴った秘密計算における受信値を前記比較検証部が検証する、秘密計算サーバ装置が提供される。

【００１４】

本発明の第３の視点では、相互にネットワークで接続した５台の秘密計算サーバ装置を用いて、秘密分散して保持されている位数２の剰余類環上の論理シェアから位数ｎ（ｎ＝２^{m}；ｍは２以上の整数）の剰余類環上の算術シェアにビット変換する秘密計算方法であって、前記秘密計算サーバ装置のそれぞれが、前記論理シェアから、自己が保持していないサブシェアをゼロに設定することで他の秘密計算サーバ装置との通信を伴うことなく算術シェアに再分散し、前記論理シェアをビット変換した算術シェアを得るために、前記再分散された算術シェアを用いて通信を伴った秘密計算を行い、少なくとも３台以上の秘密計算サーバ装置から受信した同一の値となるはずの受信値を比較し、少なくとも２つ以上が同一である受信値を正しい値として採用することで、前記通信を伴った秘密計算における受信値を検証する、秘密計算方法が提供される。

【００１５】

本発明の第４の視点では、相互にネットワークで接続した少なくとも５台以上の秘密計算サーバ装置に、秘密分散して保持されている値の秘密計算をさせる秘密計算プログラムであって、相互にネットワークで接続した５台の秘密計算サーバ装置に、秘密分散して保持されている位数２の剰余類環上の論理シェアから位数ｎ（ｎ＝２^{m}；ｍは２以上の整数）の剰余類環上の算術シェアにビット変換をさせる秘密計算プログラムであって、前記秘密計算サーバ装置のそれぞれが、前記論理シェアから、自己が保持していないサブシェアをゼロに設定することで他の秘密計算サーバ装置との通信を伴うことなく算術シェアに再分散し、前記論理シェアをビット変換した算術シェアを得るために、前記再分散された算術シェアを用いて通信を伴った秘密計算を行い、少なくとも３台以上の秘密計算サーバ装置から受信した同一の値となるはずの受信値を比較し、少なくとも２つ以上が同一である受信値を正しい値として採用することで、前記通信を伴った秘密計算における受信値を検証する、秘密計算プログラムが提供される。なお、このプログラムは、コンピュータが読み取り可能な記憶媒体に記録することができる。記憶媒体は、半導体メモリ、ハードディスク、磁気記録媒体、光記録媒体等の非トランジェント（non-transient）なものとすることができる。本発明は、コンピュータプログラム製品として具現することも可能である。

【発明の効果】
【００１６】
本発明の各視点によれば、標準モデルにおいてGuaranteed Output Delivery (GOD)を達成するビット変換に寄与する秘密計算システム、秘密計算サーバ装置、秘密計算方法および秘密計算プログラムを提供することができる。
【図面の簡単な説明】
【００１７】
【図１】図１は、第１の実施形態における秘密計算システムの機能構成例を示すブロック図である。
【図２】図２は、第１の実施形態における秘密計算サーバ装置の機能構成例を示すブロック図である。
【図３】図３は、秘密計算方法の手順の概略を示すフローチャートである。
【図４】図４は、第２の実施形態における秘密計算システムの機能構成例を示すブロック図である。
【図５】図５は、第２の実施形態における秘密計算サーバ装置の機能構成例を示すブロック図である。
【図６】図６は、第３の実施形態における秘密計算システムの機能構成例を示すブロック図である。
【図７】図７は、第３の実施形態における秘密計算サーバ装置の機能構成例を示すブロック図である。
【図８】図８は、秘密計算方法の手順の概略を示すフローチャートである。
【図９】図９は、秘密計算サーバ装置のハードウェア構成例を示す図である。
【発明を実施するための形態】
【００１８】
以下、図面を参照しながら、本発明の実施形態について説明する。ただし、以下に説明する実施形態により本発明が限定されるものではない。また、各図面において、同一または対応する要素には適宜同一の符号を付している。さらに、図面は模式的なものであり、各要素の寸法の関係、各要素の比率などは、現実のものとは異なる場合があることに留意する必要がある。図面の相互間においても、互いの寸法の関係や比率が異なる部分が含まれている場合がある。
【００１９】
［第１の実施形態］
以下、図１、図２を参照して、第１の実施形態に係る秘密計算システムおよび秘密計算サーバ装置について説明する。第１の実施形態は、本発明の基本的なコンセプトのみを説明する実施形態である。
【００２０】
図１は、第１の実施形態における秘密計算システムの機能構成例を示すブロック図である。図１に示すように、第１の実施形態による秘密計算システム１００は、第１の秘密計算サーバ装置１００_０と第２の秘密計算サーバ装置１００_１と第３の秘密計算サーバ装置１００_２と第４の秘密計算サーバ装置１００_３と第５の秘密計算サーバ装置１００_４とを備えている。第１の秘密計算サーバ装置１００_０、第２の秘密計算サーバ装置１００_１、第３の秘密計算サーバ装置１００_２、第４の秘密計算サーバ装置１００_３、および第５の秘密計算サーバ装置１００_４は、それぞれが互いにネットワーク経由で通信可能に接続されている。
【００２１】
第１～第５の秘密計算サーバ装置１００_ｉ（ｉ＝０，１，２，３，４）を備える秘密計算システム１００においては、第１～第５の秘密計算サーバ装置１００_ｉ（ｉ＝０，１，２，３，４）の内のいずれかの秘密計算サーバ装置１００_ｉが入力した値に対し、その入力や計算過程の値を知られることなく目的のシェアを計算し、その計算結果を第１～第５の秘密計算サーバ装置１００_ｉ（ｉ＝０，１，２，３，４）に分散して記憶することがで

きる。

【００２２】

また、第１～第５の秘密計算サーバ装置１００_ｉ（ｉ＝０，１，２，３，４）を備える秘密計算システム１００においては、第１～第５の秘密計算サーバ装置１００_ｉ（ｉ＝０，１，２，３，４）に分散して記憶されているシェアに対し、その計算過程の値を知られることなく目的のシェアを計算し、その計算結果を第１～第５の秘密計算サーバ装置１００_ｉ（ｉ＝０，１，２，３，４）に分散して記憶することができる。

【００２３】

なお、上記計算結果のシェアは、第１～第５の秘密計算サーバ装置１００_０～１００_４とシェアを送受信することで、復元してもよい。あるいは、第１～第５の秘密計算サーバ装置１００_０～１００_４ではない外部にシェアを送信することで、復号してもよい。

【００２４】

さらに、第１～第５の秘密計算サーバ装置１００_ｉ（ｉ＝０，１，２，３，４）を備える秘密計算システム１００においては、第１～第５の秘密計算サーバ装置１００_ｉ（ｉ＝０，１，２，３，４）のうち１つが不正者によって運営されている場合であっても、処理を停止することなく、正しい秘密計算を継続することができる。

【００２５】

例えば上記のように、第１～第５の秘密計算サーバ装置１００_ｉ（ｉ＝０，１，２，３，４）のうち１つが不正者によって運営されている場合であっても、処理を停止することなく、正しい秘密計算を継続することができるシェアの構成として以下の構成を採用することができる。

【００２６】

位数ｎの剰余類環Z_nの元$x \in Z_n$の剰余類環Z_n上のシェア（以下、これを算術シェアということがある）を以下のように定義する。ただし、ｍは２以上の整数とし、$n=2^m$であるとする。つまり、位数２の剰余類環Z_2は、位数ｎの剰余類環Z_nと区別する。

【００２７】

位数ｎの剰余類環Z_nの元$x \in Z_n$を

$$x = x_0 + x_1 + x_2 + x_3 + x_4 \bmod n$$

との関係を満たすように分解し、各参加者 P_i (i = 0, 1, 2, 3, 4)が分散保持する$[x]_i$は、以下のようにする。

$$[x]_i = (x_i, x_{i+1}, x_{i+2}, x_{i+3}), \text{ただし、} x_{4+1} = x_0$$

【００２８】

一方、位数２の剰余類環Z_2の元$x \in Z_2$の剰余類環Z_2上のシェア（以下、これを論理シェアということがある）は、上記剰余類環Z_n上のシェアにおけるn=2の場合と同様の定義であるが、記法としては位数ｎの剰余類環Z_nと区別し、$[x]^B$のように表す。すなわち、具体的には以下のように定める。

【００２９】

位数２の剰余類環Z_2の元$x \in Z_2$を以下のように分解する。なお、〇で囲まれた＋は排他的論理和を表す。

【数１】

$$x = x_0 \oplus x_1 \oplus x_2 \oplus x_3 \oplus x_4 \ \ mod\ 2$$

【００３０】

そして、各参加者 P_i (i = 0, 1, 2, 3, 4)が分散保持する$[x]^B{}_i$は、以下のようにする。

$$[x]^B{}_i = (x_i, x_{i+1}, x_{i+2}, x_{i+3}), \text{ただし、} x_{4+1} = x_0$$

【００３１】

このように各参加者P_i (i = 0, 1, 2, 3, 4)が保持するシェア$[x]_0$, $[x]_1$, $[x]_2$, $[x]_3$, $[x]_4$を定めると、各参加者P_i (i = 0, 1, 2, 3, 4)は、自己が保持するシェア$[x]_0$, $[x]_1$, $[x]_2$, $[x]_3$, $[x]_4$からはｘを復元することはできない。一方、参加者P_i (i = 0, 1, 2, 3, 4)のうち、少なくとも２名が保持しているシェアを合わせるとｘを復元することができるという

秘密分散が実現する。なお、この秘密分散方式は、2-out-of-5複製型秘密分散（Replicated Secret Sharing Scheme）と呼ばれている。

【００３２】

ところで、この秘密分散方式の秘密計算では、ｘを復元する場合に限らず、ビット変換をする場合でも、各参加者が、自己が保持していないサブシェアの値を他の参加者から直接ないし間接的に受信する状況が発生する。

【００３３】

例えば、１という数を考えた場合、(1, 0, 0, 0, 0)と(1, 1, 1, 1, 1)は、共に１を排他的論理和に分解したものである。しかしながら、1+0+0+0+0=1であるが、1+1+1+1+1=5であるので、(1, 1, 1, 1, 1)は、１という数の算術和ではない。つまり、(1, 0, 0, 0, 0)と(1, 1, 1, 1, 1)は、位数２の剰余類環Z_2の元$x \in Z_2$の剰余類環Z_2上のシェアとしては同値であっても、位数ｎの剰余類環Z_nの元$x \in Z_n$の剰余類環Z_n上のシェアとしては同値にならない。

【００３４】

そこで、ビット変換をする場合でも、各参加者が、自己が保持していないサブシェアの値を他の参加者から直接ないし間接的に受信する状況が発生するが、他の参加者の中に不正者が紛れたとすると、本来は受信したい値の代わりに別の値が送信されてくることが起こり得る。すると、誤った値に基づいて秘密計算を実行することになり、誤った計算を得ることになったり、そもそも計算自体を正常に実行することができなくなったりするのである。

【００３５】

そこで、本実施形態の秘密計算システム１００では、図２に示すように、第１～第５の秘密計算サーバ装置１００_ｉ（ｉ＝０，１，２，３，４）が、ローカル再分散部１０１_ｉと秘密計算部１０２_ｉと比較検証部１０３_ｉを備える。図２は、第１の実施形態における秘密計算サーバ装置の機能構成例を示すブロック図である。

【００３６】

ローカル再分散部１０１_ｉは、論理シェアから、自己が保持していないサブシェアをゼロに設定することで他の秘密計算サーバ装置との通信を伴うことなく算術シェアを計算し、秘密計算部１０２_ｉは、論理シェアをビット変換した算術シェアを得るために、ローカル再分散部１０１_ｉが得た算術シェアを用いて通信を伴った秘密計算を行う。そして、比較検証部１０３_ｉは、少なくとも３台以上の秘密計算サーバ装置から受信した同一の値となるはずの受信値を比較し、少なくとも２つ以上が同一である受信値を正しい値として採用することで、通信を伴った秘密計算における受信値を検証する。

【００３７】

このように、本実施形態の秘密計算システム１００は、位数２の剰余類環上の論理シェアから位数ｎの剰余類環上の算術シェアにビット変換する際に、最初に通信を伴うことなく再分散（ローカル再分散）を行い、当該ローカル再分散された算術シェアから論理シェアをビット変換した算術シェアを得るために行う通信を伴った秘密計算を行い、当該通信を伴った秘密計算における受信値を少なくとも３台以上の秘密計算サーバ装置から受信した同一の値となるはずの受信値を比較することで検証を行う。

【００３８】

なお、本実施形態を秘密計算方法として捉えると、以下のようになる。図３は、秘密計算方法の手順の概略を示すフローチャートである。

【００３９】

図３に示すように、本実施形態に係る秘密計算方法は、ローカル再分散ステップ（Ｓ１１）と通信を伴った秘密計算ステップ（Ｓ１２）と比較検証ステップ（Ｓ１３）とを有する。ローカル再分散ステップ（Ｓ１１）では、論理シェアから、自己が保持していないサブシェアをゼロに設定することで他の秘密計算サーバ装置との通信を伴うことなく算術シェアを計算し、通信を伴った秘密計算ステップ（Ｓ１２）では、論理シェアをビット変換した算術シェアを得るために、再分散された算術シェアを用いて通信を伴った秘密計算を

行う。そして、比較検証ステップ（Ｓ１３）にて、少なくとも３台以上の秘密計算サーバ装置から受信した同一の値となるはずの受信値を比較し、少なくとも２つ以上が同一である受信値を正しい値として採用することで、通信を伴った秘密計算における受信値を検証する。なお、比較検証ステップ（Ｓ１３）は、通信を伴った秘密計算ステップ（Ｓ１２）を行う度に行う。

【００４０】

このように、本実施形態の秘密計算システム１００および秘密計算方法は、少なくとも３人の他の参加者から同一となるはずの受信値を受信し、少なくとも２つ以上が同一である受信値を正しい値として採用することで、他の参加者の中に不正者が紛れたとしても正しい値を判別することができる。すなわち、不正者が紛れたとしても処理を停止することなく、正しい計算を得ることができるGuaranteed Output Delivery (GOD)が実現されている。また、上記処理では、そもそもハッシュ関数を用いていないので、標準モデルにおいてGuaranteed Output Delivery (GOD)が実現されている。

【００４１】

さらに、本実施形態の秘密計算システム１００および秘密計算方法は、最初に通信を伴うことなく再分散（ローカル再分散）を行い、その後に通信を伴った秘密計算を行うので、通信コストが低減されている。

【００４２】

以上説明した第１の実施形態は、本発明の基本的なコンセプトのみを説明する実施形態である。以下で説明する第２の実施形態は、上記説明したコンセプトを実用的な実施形態に適用したものである。

【００４３】

［第２の実施形態］

以下、図４および図５を参照して、第２の実施形態に係る秘密計算システムおよび秘密計算サーバ装置について説明する。

【００４４】

図４は、第２の実施形態における秘密計算システムの機能構成例を示すブロック図である。図４に示すように、第２の実施形態による秘密計算システム２００は、第１の秘密計算サーバ装置２００_０と第２の秘密計算サーバ装置２００_１と第３の秘密計算サーバ装置２００_２と第４の秘密計算サーバ装置２００_３と第５の秘密計算サーバ装置２００_４とを備えている。第１の秘密計算サーバ装置２００_０、第２の秘密計算サーバ装置２００_１、第３の秘密計算サーバ装置２００_２、第４の秘密計算サーバ装置２００_３、および第５の秘密計算サーバ装置２００_４は、それぞれが互いにネットワーク経由で通信可能に接続されている。

【００４５】

第１～第５の秘密計算サーバ装置２００_ｉ（ｉ＝０，１，２，３，４）を備える秘密計算システム２００においては、第１～第５の秘密計算サーバ装置２００_ｉ（ｉ＝０，１，２，３，４）の内のいずれかの秘密計算サーバ装置２００_ｉが入力した値に対し、その入力や計算過程の値を知られることなく目的のシェアを計算し、その計算結果を第１～第５の秘密計算サーバ装置２００_ｉ（ｉ＝０，１，２，３，４）に分散して記憶することができる。

【００４６】

さらに、第１～第５の秘密計算サーバ装置２００_ｉ（ｉ＝０，１，２，３，４）を備える秘密計算システム２００においては、第１～第５の秘密計算サーバ装置２００_ｉ（ｉ＝０，１，２，３，４）のうち１つが不正者によって運営されている場合であっても、処理を停止することなく、正しい秘密計算を継続することができる。

【００４７】

図５は、第２の実施形態における秘密計算サーバ装置の機能構成例を示すブロック図である。本実施形態の秘密計算システム２００は、図５に示すように、第１～第５の秘密計算サーバ装置２００_ｉ（ｉ＝０，１，２，３，４）が、ローカル再分散部２０１_ｉと秘

密計算部２０２_ｉと比較検証部２０３_ｉを備える。

【００４８】

ローカル再分散部２０１_ｉは、論理シェアから、自己が保持していないサブシェアをゼロに設定することで他の秘密計算サーバ装置との通信を伴うことなく算術シェアを計算し、秘密計算部２０２_ｉは、論理シェアをビット変換した算術シェアを得るために、ローカル再分散部２０１_ｉが得た算術シェアを用いて通信を伴った秘密計算を行う。そして、比較検証部２０３_ｉは、少なくとも３台以上の秘密計算サーバ装置から受信した同一の値となるはずの受信値を比較し、少なくとも２つ以上が同一である受信値を正しい値として採用することで、通信を伴った秘密計算における受信値を検証する。

【００４９】

以下、本実施形態におけるビット変換の実施に用いられるビルディングブロックについて説明する。なお、以下ではビット変換の実施に用いられるビルディングブロックの全てを説明することはしない。基本となる四則演算の秘密計算のうち、自明ではない乗算を中心に説明を行う。

【００５０】

［疑似乱数の生成とシードの共有］

疑似ランダム関数F_n，F_2とシードおよび識別子の関係は次の通りである。疑似ランダム関数F_n，F_2は、セキュリティパラメータκに対して定められた２項演算である。

$$F_n : \{0,1\}^\kappa \times \{0,1\}^\kappa \to \{0,1\}^n$$

$$F_2 : \{0,1\}^\kappa \times \{0,1\}^\kappa \to \{0,1\}^2$$

一方、シード$seed_i \in \{0,1\}^\kappa (i=0, 1, 2, 3, 4)$は、各秘密計算サーバ装置２００_ｉが適切に共有している値であり、識別子$vid \in \{0,1\}^\kappa$はカウンタなどの公開された値である。これらシードと識別子を入力として、疑似ランダム関数F_n，F_2は、決定的に疑似乱数を生成する。

【００５１】

５つのシード$seed_i \in \{0,1\}^\kappa (i=0, 1, 2, 3, 4)$は、各秘密計算サーバ装置２００_ｉが$(seed_i, seed_{i+1}, seed_{i+2}, seed_{i+3})$を保持する。ただし、$seed_{4+1} = seed_0$である。すなわち、各秘密計算サーバ装置２００_ｉは、すべてのシード$seed_i$を保持するのではなく、秘密計算サーバ装置２００_ｉは、シード$seed_{i+4}$のみを保持していないという関係となっている。これらシードの共有は、例えば、各秘密計算サーバ装置２００_ｉにおける事前の設定として管理者などが適切に設定することができる。

【００５２】

［マスク作成］

ここでは、参加者P_{i+4}からは乱数に見えて除去できないが、残りの参加者P_i，P_{i+1}，P_{i+2}，P_{i+3}にとっては決定的に計算可能であるような、疑似乱数(Correlated Randomness)を作成し、これを後に説明する乗算の秘密計算においてマスクとして用いる。

【００５３】

まず、参加者P_{i+4}がシード$seed_{i+3}$を保持していないことに着目すると、疑似ランダム関数F_nの入力としてシード$seed_{i+3}$を用いれば以下の疑似乱数は上記条件を満たす。すなわち、下記α_kは、参加者P_{i+4}からは乱数に見えて除去できないが、残りの参加者P_i，P_{i+1}，P_{i+2}，P_{i+3}にとっては決定的に計算可能である。

$$\alpha_k = F_n(vid_k, seed_{i+3}) - F_n(vid_{k+1}, seed_{i+3}) \bmod n$$

【００５４】

また、識別子vid_kにおけるインデックスkをk=0からk=4まで変化させると、α_kを５つ作成することが可能である。そこで、これらα_kの組を以下のように定める。このように定めたα_0，α_1，α_2，α_3，α_4が$\alpha_0+\alpha_1+\alpha_2+\alpha_3+\alpha_4=0$を満たすことは容易に確かめることができる。

$$(\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4) = CR(i+4, \{vid_k\}^4_{k=0}, seed_{i+3})$$

【００５５】

このように作成された疑似乱数α_0，α_1，α_2，α_3，α_4は、参加者P_{i+4}からは乱数に見

えて除去できないが、残りの参加者P_i, P_{i+1}, P_{i+2}, P_{i+3}にとっては決定的に計算可能である。一方、参加者P_{i+4}にとっても、各疑似乱数α_0，α_1，α_2，α_3，α_4は除去できないが、すべての疑似乱数α_0，α_1，α_2，α_3，α_4が揃うと総和は０であり除去可能になるという性質がある。

【００５６】

さらに、上記疑似乱数の作成は、すべての参加者P_{i+4}に対しても同様に行うことができる。具体的には、以下のように定めればよい。

$$(\alpha_{i,0}, \alpha_{i,1}, \alpha_{i,2}, \alpha_{i,3}, \alpha_{i,4}) = CR(i, \{vid_k\}^4_{k=0}, seed_{i+4}) \quad for\ i=0,1,2,3,4$$

$$\alpha_{i,k} = F_n(vid_k, seed_{i+4}) - F_n(vid_{k+1}, seed_{i+4}) \bmod n \quad for\ i=0,1,2,3,4$$

【００５７】

このように作成された疑似乱数の組は以下のようになる。

【表１】

$\alpha_{0,0}$	$\alpha_{1,0}$	$\alpha_{2,0}$	$\alpha_{3,0}$	$\alpha_{4,0}$
$\alpha_{0,1}$	$\alpha_{1,1}$	$\alpha_{2,1}$	$\alpha_{3,1}$	$\alpha_{4,1}$
$\alpha_{0,2}$	$\alpha_{1,2}$	$\alpha_{2,2}$	$\alpha_{3,2}$	$\alpha_{4,2}$
$\alpha_{0,3}$	$\alpha_{1,3}$	$\alpha_{2,3}$	$\alpha_{3,3}$	$\alpha_{4,3}$
$\alpha_{0,4}$	$\alpha_{1,4}$	$\alpha_{2,4}$	$\alpha_{3,4}$	$\alpha_{4,4}$

【００５８】

上記疑似乱数の表では、第１のインデックス（縦方向）に関する総和はゼロであり、第２のインデックス（横方向）に関する総和はゼロではないという性質を有する。

【００５９】

[秘密計算（乗算）]

次に、秘密計算の重要な因子である乗算について説明する。つまり、２つのシェア[x],[y]から[z]=[x・y]=[x]・[y]を計算する秘密計算の具体例を説明する。なお、x,y,zは以下のように分解されているとする。

【数２】

$$z = \sum_{i=0}^{4} z_i \ \ mod\, n$$

$$x = \sum_{i=0}^{4} x_i \ \ mod\, n$$

$$y = \sum_{i=0}^{4} y_i \ \ mod\, n$$

$$z_i = x_i \cdot \sum_{j=0}^{4} y_j \mod n$$

【００６０】
参加者P_i(i=0,1,2,3,4)は、以下のようなtmp_{zk}を計算する。このtmp_{zk}は参加者P_iがz_kを計算するためには$x_k \cdot y_{i+4}$が足りない（保持しているシェアから計算できない）ので、代わりに計算する値である。なお、$\alpha_{j,k}$は、上述の［マスク作成］の項目にて説明した疑似乱数である。
【数３】

$$tmp_{z_k} = x_k \cdot (y_i + y_{i+1} + y_{i+2} + y_{i+3}) + \sum_{j \neq i} \alpha_{j,k} \mod n$$

$$(k = i,\ i+1, i+2, i+3)$$

【００６１】
ここで、送信者集合$S_i=\{P_{i+2}, P_{i+3}, P_{i+4}\}$, $S_{i+1}=\{P_{i+3}, P_{i+4}, P_{i+1}\}$, $S_{i+2}=\{P_{i+4}, P_{i+1}, P_{i+2}\}$, $S_{i+3}=\{P_{i+1}, P_{i+2}, P_{i+3}\}$を定める。すると$S_k$に属する参加者は、$x_k y_{i+4}$を保持しているシェアから計算することができる。そこで、例えば、送信者集合$S_i=\{P_{i+2}, P_{i+3}, P_{i+4}\}$に属する参加者$P_{i+2}$, P_{i+3}, P_{i+4}は、$x_k \cdot y_{i+4}$を上記疑似乱数$\alpha_{i,k}$でマスクをした$m_{k,i+2}$, $m_{k,i+3}$, $m_{k,i+4}$を計算する。

P_{i+2}: $m_{k,i+2} = \alpha_{i,k} + x_k \cdot y_{i+4} \mod n$
P_{i+3}: $m_{k,i+3} = \alpha_{i,k} + x_k \cdot y_{i+4} \mod n$
P_{i+4}: $m_{k,i+4} = \alpha_{i,k} + x_k \cdot y_{i+4} \mod n$

【００６２】
そして、送信者集合$S_i=\{P_{i+2}, P_{i+3}, P_{i+4}\}$に属する参加者$P_{i+2}$, P_{i+3}, P_{i+4}は、例えば、参加者P_{i+2}, P_{i+3}が$m_{k,i+2}$, $m_{k,i+3}$をそのまま参加者P_iに送信し、参加者P_{i+4}が$m_{k,i+4}$のハッシュ値$h_{k,i+4}$を参加者P_iに送信する。ここで、$m_{k,i+2}$, $m_{k,i+3}$, $m_{k,i+4}$は疑似乱数$\alpha_{i,k}$でマスクをしているので$x_k y_{i+4}$が漏洩することはない。つまり、ここではハッシュ関数も用いているが、これは安全性の確保のためではなく、通信コストを削減するためである。
【００６３】
その後、$m_{k,i+2}$, $m_{k,i+3}$および$m_{k,i+4}$のハッシュ値$h_{k,i+4}$を受信した参加者P_iは、$m_{k,i+2}$, $m_{k,i+3}$および$m_{k,i+4}$のハッシュ値$h_{k,i+4}$の比較検証を行う。まず、参加者P_iは、$m_{k,i+2}$, $m_{k,i+3}$のハッシュ値$h_{k,i+2}$, $h_{k,i+3}$を計算する。そして、$h_{k,i+2}=h_{k,i+3}$または$h_{k,i+2}=h_{k,i+4}$であれば、$m_k = m_{k,i+2}$とする。一方、$h_{k,i+3}=h_{k,i+4}$であれば、$m_k = m_{k,i+2}$とする。
【００６４】
上記のように$x_k y_{i+4}$を参加者P_iへ受け渡せば、少なくとも３人の他の参加者P_jから同一となるはずのm_k（のハッシュ値）を受信し、少なくとも２つ以上が同一である受信値を正しい値として採用することで、他の参加者P_jの中に不正者が紛れたとしても正しい値を判

別することができる。

【００６５】

その後、参加者P_iは、正しい値であることが判定されたm_kを用いて、$z_k = tmp_{zk} + m_k \bmod n$ (k=i,i+1,i+2,i+3)を計算する。

【数４】

$$z_k = tmp_{z_k} + m_k$$

$$= \left(x_k \cdot (y_i + y_{i+1} + y_{i+2} + y_{i+3}) + \sum_{j \neq i} \alpha_{j,k} \right) + (\alpha_{i,k} + x_k \cdot y_{i+4})$$

$$= x_k \cdot \sum_{j=0}^{4} y_j + \sum_{j=0}^{4} \alpha_{j,k}$$

【００６６】

このように計算されたz_kは、余計な付加項が含まれているが、[z]=[xy]=[x][y]の計算結果のシェア$[z]_i=(z_i, z_{i+1}, z_{i+2}, z_{i+3})$として機能する。それは、実際に以下のように$z=z_0+z_1+z_2+z_3+z_4$を計算すると明らかとなる。

【００６７】

【数５】

$$z = z_0 + z_1 + z_2 + z_3 + z_4$$

$$= \left(x_0 \cdot \sum_{j=0}^{4} y_j + \sum_{j=0}^{4} \alpha_{j,0} \right) + \left(x_1 \cdot \sum_{j=0}^{4} y_j + \sum_{j=0}^{4} \alpha_{j,1} \right) + \left(x_2 \cdot \sum_{j=0}^{4} y_j + \sum_{j=0}^{4} \alpha_{j,2} \right) +$$

$$\left(x_3 \cdot \sum_{j=0}^{4} y_j + \sum_{j=0}^{4} \alpha_{j,3} \right) + \left(x_4 \cdot \sum_{j=0}^{4} y_j + \sum_{j=0}^{4} \alpha_{j,4} \right)$$

$$= (x_0 + x_1 + x_2 + x_3 + x_4) \cdot \sum_{j=0}^{4} y_j + \sum_{k=0}^{4} \alpha_{0,k} + \sum_{k=0}^{4} \alpha_{1,k} + \sum_{k=0}^{4} \alpha_{2,k} + \sum_{k=0}^{4} \alpha_{3,k} + \sum_{k=0}^{4} \alpha_{4,k}$$

$$= x \cdot y \bmod n$$

【００６８】
ここで、疑似乱数$\alpha_{i,k}$が消去される理由は、疑似乱数の構成から以下の関係式が成り立つからである。
【数６】

$$\sum_{k=0}^{4} \alpha_{0,k} = \sum_{k=0}^{4} \alpha_{1,k} = \sum_{k=0}^{4} \alpha_{2,k} = \sum_{k=0}^{4} \alpha_{3,k} = \sum_{k=0}^{4} \alpha_{4,k} = 0$$

【００６９】
すなわち、上述の［マスク作成］の項目にて説明したように、本構成の疑似乱数は、第１のインデックス（縦方向）に関する総和はゼロであり、第２のインデックス（横方向）に関する総和はゼロではないという性質を有する。$z_k = tmp_{zk} + m_k \bmod n$ (k=i,i+1,i+2,i+3)の計算結果に表れていた付加項は、第２のインデックス（横方向）に関する総和であり、ゼロとはならないが、[z]=[x・y]=[x]・[y]の計算結果を復元する際には、第１のインデックス（縦方向）に関する総和がゼロになるという性質を用いて、結果的に付加項（マスク）の影響を除去することが可能になる。つまり、上記計算されたz_kは、余計な付加項が含まれているが、[z]=[x・y]=[x]・[y]の計算結果のシェア$[z]_i=(z_i, z_{i+1}, z_{i+2}, z_{i+3})$として機能する。
【００７０】
以上、上記のような[z]=[x・y]=[x]・[y]の計算結果のシェア$[z]_i=(z_i, z_{i+1}, z_{i+2}, z_{i+3})$は、少なくとも３人の他の参加者$P_j$から同一となるはずの$m_k$（のハッシュ値）を受信し、少なくとも２つ以上が同一である受信値を正しい値として採用することで、他の参加者P_jの中に不正者が紛れたとしても正しい値を判別することができる。つまり、参加者の中に不正者が紛れたとしても処理を停止することなく、正しい計算を得ることができるGuaranteed Output Delivery (GOD)が実現されている。また、上記処理では、ハッシュ関数を用いているが、通信量の削減を目的として利用しているのであり、出力から入力が推定されたとしても安全性に影響を与えないので、標準モデルにおいてGuaranteed Output Delivery (GOD)が実現されている。
【００７１】
［ビット変換］
ビット変換とは、位数２の剰余類環Z_2上の論理シェア$[x]^B$から位数ｎの剰余類環Z_n上の算術シェア[x]を得るビット変換：$[x] \leftarrow BC([x]^B)$である。まず、ローカル再分散として、論理シェアから、自己が保持していないサブシェアをゼロに設定することで他の秘密計算サーバ装置との通信を伴うことなく算術シェアに再分散する。具体的には以下のように行う。
【００７２】
各参加者P_i(i=0,1,2,3,4)は以下のように、$[x_0]_i$を設定する。
P_0: $[x_0]_0 = (x_0, 0, 0, 0)$

P_1: $[x_0]_1 = (0, 0, 0, 0)$
P_2: $[x_0]_2 = (0, 0, 0, x_0)$
P_3: $[x_0]_3 = (0, 0, x_0, 0)$
P_4: $[x_0]_4 = (0, x_0, 0, 0)$

【００７３】

各参加者P_i(i=0,1,2,3,4)は以下のように、$[x_1]_i$を設定する。

P_0: $[x_1]_0 = (0, x_1, 0, 0)$
P_1: $[x_1]_1 = (x_1, 0, 0, 0)$
P_2: $[x_1]_2 = (0, 0, 0, 0)$
P_3: $[x_1]_3 = (0, 0, 0, x_1)$
P_4: $[x_1]_4 = (0, 0, x_1, 0)$

【００７４】

各参加者P_i(i=0,1,2,3,4)は以下のように、$[x_2]_i$を設定する。

P_0: $[x_2]_0 = (0, 0, x_2, 0)$
P_1: $[x_2]_1 = (0, x_2, 0, 0)$
P_2: $[x_2]_2 = (x_2, 0, 0, 0)$
P_3: $[x_2]_3 = (0, 0, 0, 0)$
P_4: $[x_2]_4 = (0, 0, 0, x_2)$

【００７５】

各参加者P_i(i=0,1,2,3,4)は以下のように、$[x_3]_i$を設定する。

P_0: $[x_3]_0 = (0, 0, 0, x_3)$
P_1: $[x_3]_1 = (0, 0, x_3, 0)$
P_2: $[x_3]_2 = (0, x_3, 0, 0)$
P_3: $[x_3]_3 = (x_3, 0, 0, 0)$
P_4: $[x_3]_4 = (0, 0, 0, 0)$

【００７６】

各参加者P_i(i=0,1,2,3,4)は以下のように、$[x_4]_i$を設定する。

P_0: $[x_4]_0 = (0, 0, 0, 0)$
P_1: $[x_4]_1 = (0, 0, 0, x_4)$
P_2: $[x_4]_2 = (0, 0, x_4, 0)$
P_3: $[x_4]_3 = (0, x_4, 0, 0)$
P_4: $[x_4]_4 = (x_4, 0, 0, 0)$

【００７７】

なお、上記ローカル再分散は、通信を伴った計算ではないので、参加者の中に不正者が紛れたとしても計算の実行に影響を及ぼさない。

【００７８】

次に、上記のように再分散された算術シェアを用いて通信を伴った秘密計算を行い、論理シェアをビット変換した算術シェアを得る。なお、ここで再度の注意を述べるが、例えば、(1, 0, 0, 0, 0)と(1, 1, 1, 1, 1)は、位数２の剰余類環Z_2の元$x \in Z_2$の剰余類環Z_2上のシェアとしては同値であっても、位数ｎの剰余類環Z_nの元$x \in Z_n$の剰余類環Z_n上のシェアとしては同値にならない。これは、排他的論理和を用いた分解は、算術和の分解に対応しないからである。そこで、排他的論理和と算術和との違いを相殺するために以下のような秘密計算を行う。

【００７９】

【数７】

$$[x_0 \oplus x_1] = ([x_0] - [x_1])^2$$

$$[x_2 \oplus x_3] = ([x_2] - [x_3])^2$$

$$[(x_0 \oplus x_1) \oplus (x_2 \oplus x_3)] = ([x_0 \oplus x_1] - [x_2 \oplus x_3])^2$$

$$[x] = [(x_0 \oplus x_1) \oplus (x_2 \oplus x_3) \oplus x_4] = ([(x_0 \oplus x_1) \oplus (x_2 \oplus x_3)] - [x_4])^2$$

【００８０】
ここで、上記秘密計算は、２乗が含まれているので乗算の秘密計算を含む。そして、乗算の秘密計算では、他の秘密計算サーバ装置との間で通信が必要となる。そこで、上述の［秘密計算（乗算）］を用い、少なくとも３台以上の秘密計算サーバ装置から受信した同一の値となるはずの受信値を比較し、少なくとも２つ以上が同一である受信値を正しい値として採用することで、通信を伴った秘密計算における受信値を検証する。
【００８１】
以上のように、第２の実施形態の秘密計算システム２００および秘密計算方法は、少なくとも３人の他の参加者から同一となるはずの受信値を受信し、少なくとも２つ以上が同一である受信値を正しい値として採用することで、他の参加者の中に不正者が紛れたとしても正しい値を判別することができる。すなわち、不正者が紛れたとしても処理を停止することなく、正しい計算を得ることができるGuaranteed Output Delivery (GOD)が実現されている。
【００８２】
また、上記処理では、ハッシュ関数を用いているが、通信量の削減を目的として利用しているのであり、出力から入力が推定されたとしても安全性に影響を与えないので、標準モデルにおいてGuaranteed Output Delivery (GOD)が実現されている。さらに、本実施形態の秘密計算システム２００および秘密計算方法は、最初に通信を伴うことなく再分散（ローカル再分散）を行い、その後に通信を伴った秘密計算を行うので、通信コストが低減されている。
【００８３】
［第３の実施形態］
以下、図６および図７を参照して、第３の実施形態に係る秘密計算システムおよび秘密計算サーバ装置について説明する。
【００８４】
図６は、第３の実施形態における秘密計算システムの機能構成例を示すブロック図である。図６に示すように、第３の実施形態による秘密計算システム３００は、第１の秘密計算サーバ装置３００_０と第２の秘密計算サーバ装置３００_１と第３の秘密計算サーバ装置３００_２と第４の秘密計算サーバ装置３００_３と第５の秘密計算サーバ装置３００_４とを備えている。第１の秘密計算サーバ装置３００_０、第２の秘密計算サーバ装置３００_１、第３の秘密計算サーバ装置３００_２、第４の秘密計算サーバ装置３００_３、および第５の秘密計算サーバ装置３００_４は、それぞれが互いにネットワーク経由で通信可能に接続されている。
【００８５】

第１～第５の秘密計算サーバ装置３００_ｉ（ｉ＝０，１，２，３，４）を備える秘密計算システム３００においては、第１～第５の秘密計算サーバ装置３００_ｉ（ｉ＝０，１，２，３，４）の内のいずれかの秘密計算サーバ装置３００_ｉが入力した値に対し、その入力や計算過程の値を知られることなく目的のシェアを計算し、その計算結果を第１～第５の秘密計算サーバ装置３００_ｉ（ｉ＝０，１，２，３，４）に分散して記憶することができる。

【００８６】

さらに、第１～第５の秘密計算サーバ装置３００_ｉ（ｉ＝０，１，２，３，４）を備える秘密計算システム３００においては、第１～第５の秘密計算サーバ装置３００_ｉ（ｉ＝０，１，２，３，４）のうち１つが不正者によって運営されている場合であっても、処理を停止することなく、正しい秘密計算を継続することができる。

【００８７】

図７は、第３の実施形態における秘密計算サーバ装置の機能構成例を示すブロック図である。本実施形態秘密計算システム３００は、図７に示すように、第１～第５の秘密計算サーバ装置３００_ｉ（ｉ＝０，１，２，３，４）が、ローカル再分散部３０１_ｉと秘密計算部３０２_ｉと比較検証部３０３_ｉと再分散部３０４_ｉを備える。

【００８８】

ローカル再分散部３０１_ｉは、論理シェアから、自己が保持していないサブシェアをゼロに設定することで他の秘密計算サーバ装置との通信を伴うことなく算術シェアを計算し、秘密計算部３０２_ｉは、論理シェアをビット変換した算術シェアを得るために、ローカル再分散部３０１_ｉが得た算術シェアを用いて通信を伴った秘密計算を行う。そして、比較検証部３０３_ｉは、少なくとも３台以上の秘密計算サーバ装置から受信した同一の値となるはずの受信値を比較し、少なくとも２つ以上が同一である受信値を正しい値として採用することで、通信を伴った秘密計算における受信値を検証する。再分散部３０４_ｉは、論理シェアにおけるサブシェアから計算された一時的変数を算術シェアとして再分散する。

【００８９】

このように、第３の実施形態における秘密計算サーバ装置３００_ｉは、第２の実施形態における秘密計算サーバ装置２００_ｉの構成に加えて再分散部３０４_ｉを備えている。この再分散部３０４_ｉの役割を秘密計算方法の中で説明すると、以下のようになる。図８は、秘密計算方法の手順の概略を示すフローチャートである。

【００９０】

図８に示すように、本実施形態に係る秘密計算方法は、再分散ステップ（Ｓ２１）とローカル再分散ステップ（Ｓ２２）と通信を伴った秘密計算ステップ（Ｓ２３）と比較検証ステップ（Ｓ２４）とを有する。再分散ステップ（Ｓ２１）では、論理シェアにおけるサブシェアから計算された一時的変数を算術シェアとして再分散し、ローカル再分散ステップ（Ｓ２２）では、論理シェアから、自己が保持していないサブシェアをゼロに設定することで他の秘密計算サーバ装置との通信を伴うことなく算術シェアを計算する。なお、再分散ステップ（Ｓ２１）とローカル再分散ステップ（Ｓ２２）は順序が逆になってもよい。

【００９１】

通信を伴った秘密計算ステップ（Ｓ２３）では、論理シェアをビット変換した算術シェアを得るために、再分散された算術シェアを用いて通信を伴った秘密計算を行う。そして、比較検証ステップ（Ｓ２４）にて、少なくとも３台以上の秘密計算サーバ装置から受信した同一の値となるはずの受信値を比較し、少なくとも２つ以上が同一である受信値を正しい値として採用することで、通信を伴った秘密計算における受信値を検証する。なお、比較検証ステップ（Ｓ２４）は、通信を伴った秘密計算を行う度に行う。したがって、通信を伴った秘密計算ステップ（Ｓ２３）の受信値に対してだけではなく、再分散ステップ（Ｓ２１）の受信値に対しても比較検証ステップ（Ｓ２４）が行われる。

【００９２】

［再分散］

まず、本実施形態において追加された再分散部３０４_ｉの機能について説明する。本実

施形態で用いられる再分散は、以下のように定義される。すなわち、参加者P_i,P_{i+1},P_{i+2}が値cを保持していた場合に、シードと識別子から決定的に定められる再分散である。

【００９３】

【数８】

$$[c] \leftarrow Reshare(P_i, P_{i+1}, P_{i+2}, c, \{vid_j\}_{j=1}^{4}, seed_{i+2}, seed_{i+3})$$

$$c_i = \begin{cases} c - r_1 - r_2 - r_3 - r_4 - r_1' - r_2' - r_3' - r_4' & (i = 0) \\ r_i + r_i' & (else) \end{cases}$$

$$where\ c = c_0 + c_1 + c_2 + c_3 + c_4\ mod\ n$$

【００９４】

ただし、$r_j = F_n(vid_k, seed_{i+2})$かつ$r'_j = F_n(vid_{k+1}, seed_{i+3})$であり、シード$seed_i \in \{0,1\}^\kappa (i=0, 1, 2, 3, 4)$は、先述の［疑似乱数の生成とシードの共有］の項で説明した性質のシードである。したがって、参加者P_{i+3}は、$seed_{i+2}$を知らず、また、参加者P_{i+4}は、$seed_{i+3}$を知らない。つまり、参加者P_{i+3}, P_{i+4}は、自分でc_{i+3},c_{i+4}を計算することができず、参加者P_i,P_{i+1},P_{i+2}からc_{i+3},c_{i+4}を受信する必要がある。

【００９５】

ここで、通信を伴った秘密計算が発生するので、参加者P_{i+3}, P_{i+4}は、参加者P_i,P_{i+1},P_{i+2}から受信した同一の値となるはずの受信値c_{i+3},c_{i+4}を比較し、少なくとも２つ以上が同一である受信値を正しい値として採用する。具体的には、以下のように行うことができる。

【００９６】

参加者P_i,P_{i+1}は、$c_{j+1},c_{j+2},c_{j+3}(j=i+3)$を参加者$P_{i+3}$に送信する。一方、参加者$P_{i+2}$は、$c_{j+1},c_{j+2},c_{j+3}(j=i+3)$のハッシュ値を各参加者に送信する。さらに、参加者$P_i,P_{i+1}$は、$c_{j'+1},c_{j'+2},c_{j'+3}(j'=i+3)$を参加者$P_{i+4}$に送信する。一方、参加者$P_{i+2}$は、$c_{j'+1},c_{j'+2},c_{j'+3}(j'=i+3)$を各参加者に送信する。そして、参加者$P_{i+3}$, P_{i+4}は、参加者P_i,P_{i+1},P_{i+2}から受信した受信値のうち少なくとも２つ以上が同一である受信値を正しい値として採用する。

【００９７】

次に、上記説明した再分散をビット変換の中でどのように用いるかを説明する。

【００９８】

［ビット変換］

ビット変換とは、位数２の剰余類環Z_2上の論理シェア$[x]^B$から位数ｎの剰余類環Z_n上の算術シェア$[x]$を得るビット変換：$[x] \leftarrow BC([x]^B)$であった。まず、参加者P_3,P_4,P_0および参加者P_0,P_1,P_2は、それぞれ以下のように、論理シェア$[x]^B$におけるサブシェアx_iから計算された一時的変数y_0,y_1を計算する。

【００９９】

【数９】

$$y_0 = x_0 \oplus x_1$$

$$y_1 = x_2 \oplus x_3$$

【０１００】
次に、参加者P_3,P_4,P_0および参加者P_0,P_1,P_2は、一時的変数y_0,y_1を再分散する。
【数１０】

$$[y_0] \leftarrow Reshare\left(P_3, P_4, P_0, y_0, \left\{vid_{0,k}\right\}_{k=1}^{4}, seed_0, seed_1\right)$$

$$[y_1] \leftarrow Reshare(P_0, P_1, P_2, y_1, \left\{vid_{1,k}\right\}_{k=1}^{4}, seed_2, seed_3)$$

【０１０１】
なお、上記再分散は、通信を伴った秘密計算であるので、既に説明したように、参加者P_{i+3}, P_{i+4}は、参加者P_i,P_{i+1},P_{i+2}から受信した受信値のうち少なくとも２つ以上が同一である受信値を正しい値として採用する。
【０１０２】
一方、各参加者P_i(i=0,1,2,3,4)は以下のように、$[x_4]_i$を設定する。なお、この処理は通信を伴った秘密計算ではないので、検証は必要ない。
P_0: $[x_4]_0 = (0, 0, 0, 0)$
P_1: $[x_4]_1 = (0, 0, 0, x_4)$
P_2: $[x_4]_2 = (0, 0, x_4, 0)$
P_3: $[x_4]_3 = (0, x_4, 0, 0)$
P_4: $[x_4]_4 = (x_4, 0, 0, 0)$
【０１０３】
そして、最終的に、一時的変数y_0,y_1の算術シェアと算術シェア$[x_4]_i$(i=0,1,2,3,4)とを用いて、前記論理シェアをビット変換した算術シェアを以下のように秘密計算する。
【０１０４】
【数１１】

$$[y_1 \oplus x_4] = ([y_1] - [x_4])^2$$

$$[x] = [y_1 \oplus x_4 \oplus y_0] = ([y_1 \oplus x_4] - [y_0])^2$$

【０１０５】
なお、ここでも、上記秘密計算は、乗算の秘密計算を含むので、上述の［秘密計算（乗

算）］を用い、少なくとも３台以上の秘密計算サーバ装置から受信した同一の値となるはずの受信値を比較し、少なくとも２つ以上が同一である受信値を正しい値として採用することで、通信を伴った秘密計算における受信値を検証する。

【０１０６】

以上のように、第３の実施形態の秘密計算システム３００および秘密計算方法は、少なくとも３人の他の参加者から同一となるはずの受信値を受信し、少なくとも２つ以上が同一である受信値を正しい値として採用することで、他の参加者の中に不正者が紛れたとしても正しい値を判別することができる。すなわち、不正者が紛れたとしても処理を停止することなく、正しい計算を得ることができるGuaranteed Output Delivery (GOD)が実現されている。

【０１０７】

また、上記処理では、ハッシュ関数を用いているが、通信量の削減を目的として利用しているのであり、出力から入力が推定されたとしても安全性に影響を与えないので、標準モデルにおいてGuaranteed Output Delivery (GOD)が実現されている。さらに、本実施形態の秘密計算システム３００および秘密計算方法は、最初に通信を伴うことなく再分散（ローカル再分散）を行い、その後に通信を伴った秘密計算を行うので、通信コストが低減されている。

【０１０８】

特に、第３の実施形態の秘密計算システム３００および秘密計算方法は、通信を伴った再分散と通信を伴わない再分散とを組み合わせて用いているので、第２の実施形態よりも通信コストが低減されている。具体的には、第２の実施形態の通信コストは、ラウンド数が３ラウンドであり、通信量が160kビットである。一方、第３の実施形態の通信コストは、ラウンド数が３ラウンドであり、通信量が112kビットである。つまり、第３の実施形態の秘密計算システム３００および秘密計算方法は、第２の実施形態との比較において、ラウンド数が同じでありながら、通信量48kビット削減することができている。

【０１０９】

［ハードウェア構成例］

図９は、秘密計算サーバ装置のハードウェア構成例を示す図である。すなわち、図９に示すハードウェア構成例は、秘密計算サーバ装置１００_ｉ，２００_ｉ，３００_ｉ（ｉ＝０，１，２，３，４）のハードウェア構成例である。図９に示すハードウェア構成を採用した情報処理装置（コンピュータ）は、上記説明した秘密計算方法をプログラムとして実行することで、秘密計算サーバ装置１００_ｉ，２００_ｉ，３００_ｉ（ｉ＝０，１，２，３，４）の各機能を実現することを可能にする。

【０１１０】

ただし、図９に示すハードウェア構成例は、秘密計算サーバ装置１００_ｉ，２００_ｉ，３００_ｉ（ｉ＝０，１，２，３，４）の各機能を実現するハードウェア構成の一例であり、秘密計算サーバ装置１００_ｉ，２００_ｉ，３００_ｉ（ｉ＝０，１，２，３，４）のハードウェア構成を限定する趣旨ではない。秘密計算サーバ装置１００_ｉ，２００_ｉ，３００_ｉ（ｉ＝０，１，２，３，４）は、図９に示さないハードウェアを含むことができる。

【０１１１】

図９に示すように、秘密計算サーバ装置１００_ｉ，２００_ｉ，３００_ｉ（ｉ＝０，１，２，３，４）が採用し得るハードウェア構成１０は、例えば内部バスにより相互に接続される、ＣＰＵ（Central Processing Unit）１１、主記憶装置１２、補助記憶装置１３、およびＩＦ（Interface）部１４を備える。

【０１１２】

ＣＰＵ１１は、秘密計算サーバ装置１００_ｉ，２００_ｉ，３００_ｉ（ｉ＝０，１，２，３，４）が実行する秘密計算プログラムに含まれる各指令を実行する。主記憶装置１２は、例えばＲＡＭ（Random Access Memory）であり、秘密計算サーバ装置１００_ｉ，２００_ｉ，３００_ｉ（ｉ＝０，１，２，３，４）が実行する秘密計算プログラムな

どの各種プログラムなどをＣＰＵ１１が処理するために一時記憶する。
【０１１３】
補助記憶装置１３は、例えば、ＨＤＤ（Hard Disk Drive）であり、秘密計算サーバ装置１００_ｉ，２００_ｉ，３００_ｉ（ｉ＝０，１，２，３，４）が実行する秘密計算プログラムなどの各種プログラムなどを中長期的に記憶しておくが可能である。秘密計算プログラムなどの各種プログラムは、非一時的なコンピュータ可読記録媒体（non-transitory computer-readable storage medium）に記録されたプログラム製品として提供することができる。補助記憶装置１３は、非一時的なコンピュータ可読記録媒体に記録された秘密計算プログラムなどの各種プログラムを中長期的に記憶することに利用することが可能である。ＩＦ部１４は、秘密計算サーバ装置１００_ｉ，２００_ｉ，３００_ｉ（ｉ＝０，１，２，３，４）間の入出力に関するインターフェイスを提供する。
【０１１４】
上記のようなハードウェア構成１０を採用した情報処理装置は、先述した秘密計算方法をプログラムとして実行することで、秘密計算サーバ装置１００_ｉ，２００_ｉ，３００_ｉ（ｉ＝０，１，２，３，４）の各機能を実現する。
【０１１５】
上記の実施形態の一部又は全部は、以下の付記のようにも記載され得るが、以下には限られない。
［付記１］
相互にネットワークで接続した５台の秘密計算サーバ装置を備え、秘密分散して保持されている位数２の剰余類環上の論理シェアから位数ｎ（ｎ＝2^m；ｍは２以上の整数）の剰余類環上の算術シェアにビット変換する秘密計算システムであって、
前記秘密計算サーバ装置のそれぞれが、
前記論理シェアから、自己が保持していないサブシェアをゼロに設定することで他の秘密計算サーバ装置との通信を伴うことなく算術シェアを計算するローカル再分散部と、
前記論理シェアをビット変換した算術シェアを得るために、前記ローカル再分散部が得た算術シェアを用いて通信を伴った秘密計算を行う秘密計算部と、
少なくとも３台以上の秘密計算サーバ装置から受信した同一の値となるはずの受信値を比較し、少なくとも２つ以上が同一である受信値を正しい値として採用する比較検証部と、
を有し、
前記通信を伴った秘密計算における受信値を前記比較検証部が検証する、秘密計算システム。
［付記２］
前記秘密計算サーバ装置のそれぞれが、
前記論理シェアにおけるサブシェアから計算された一時的変数を算術シェアとして分散する再分散部をさらに有し、
前記再分散部が分散した前記一時的変数の算術シェアの受信値を前記比較検証部が検証し、
前記再分散部が分散した前記一時的変数の算術シェアと前記ローカル再分散部が得た算術シェアとを用いて、前記論理シェアをビット変換した算術シェアを秘密計算する、付記１に記載の秘密計算システム。
［付記３］
前記一時的変数は、前記秘密計算サーバ装置の５台のうち３台が共通して保持している前記論理シェアにおけるサブシェアから計算され、
前記再分散部は、前記３台の秘密計算サーバ装置が共通して計算する前記一時的変数から決定的に生成した算術シェアを分散し、
前記比較検証部は、前記３台の秘密計算サーバ装置から受信した算術シェアに対して、少なくとも２つ以上が同一である受信値を正しい値として採用する、付記２に記載の秘密計算システム。
［付記４］

前記比較検証部は、前記受信値のハッシュ値が同一であることを判断して、前記受信値が正しい値であることを判断する、付記１から付記３のいずれか１つに記載の秘密計算システム。

［付記５］

秘密分散して保持されている位数２の剰余類環上の論理シェアから位数ｎ（ｎ＝２^{m}；ｍは２以上の整数）の剰余類環上の算術シェアにビット変換するために、相互にネットワークで接続した少なくとも５台の秘密計算サーバ装置の一つであって、

前記論理シェアから、自己が保持していないサブシェアをゼロに設定することで他の秘密計算サーバ装置との通信を伴うことなく算術シェアを計算するローカル再分散部と、

前記論理シェアをビット変換した算術シェアを得るために、前記ローカル再分散部が得た算術シェアを用いて通信を伴った秘密計算を行う秘密計算部と、

少なくとも３台以上の秘密計算サーバ装置から受信した同一の値となるはずの受信値を比較し、少なくとも２つ以上が同一である受信値を正しい値として採用する比較検証部と、

を有し、

前記通信を伴った秘密計算における受信値を前記比較検証部が検証する、秘密計算サーバ装置。

［付記６］

相互にネットワークで接続した５台の秘密計算サーバ装置を用いて、秘密分散して保持されている位数２の剰余類環上の論理シェアから位数ｎ（ｎ＝２^{m}；ｍは２以上の整数）の剰余類環上の算術シェアにビット変換する秘密計算方法であって、

前記秘密計算サーバ装置のそれぞれが、

前記論理シェアから、自己が保持していないサブシェアをゼロに設定することで他の秘密計算サーバ装置との通信を伴うことなく算術シェアに再分散し、

前記論理シェアをビット変換した算術シェアを得るために、前記再分散された算術シェアを用いて通信を伴った秘密計算を行い、

少なくとも３台以上の秘密計算サーバ装置から受信した同一の値となるはずの受信値を比較し、少なくとも２つ以上が同一である受信値を正しい値として採用することで、前記通信を伴った秘密計算における受信値を検証する、秘密計算方法。

［付記７］

前記秘密計算サーバ装置のそれぞれが、

前記論理シェアにおけるサブシェアから計算された一時的変数を算術シェアとして再分散し、

少なくとも３台以上の秘密計算サーバ装置から受信した同一の値となるはずの受信値を比較し、少なくとも２つ以上が同一である受信値を正しい値として採用することで、前記再分散した前記一時的変数の算術シェアの受信値を検証し、

前記再分散した前記一時的変数の算術シェアと前記通信を伴うことなく計算した算術シェアとを用いて、前記論理シェアをビット変換した算術シェアを秘密計算する、付記６に記載の秘密計算方法。

［付記８］

前記一時的変数を、前記秘密計算サーバ装置の５台のうち３台が共通して保持している前記論理シェアにおけるサブシェアから計算し、

前記３台の秘密計算サーバ装置が共通して計算する前記一時的変数から決定的に生成した算術シェアを再分散し、

前記３台の秘密計算サーバ装置から受信した算術シェアに対して、少なくとも２つ以上が同一である受信値を正しい値として採用する、付記７に記載の秘密計算方法。

［付記９］

前記受信値が正しい値であることを判断する際には、前記受信値のハッシュ値が同一であることを用いて判断する、付記６から付記８のいずれか１つに記載の秘密計算方法。

［付記１０］

相互にネットワークで接続した少なくとも５台以上の秘密計算サーバ装置に、秘密分散

して保持されている値の秘密計算をさせる秘密計算プログラムであって、
相互にネットワークで接続した５台の秘密計算サーバ装置に、秘密分散して保持されている位数２の剰余類環上の論理シェアから位数ｎ（ｎ＝２^{m}；ｍは２以上の整数）の剰余類環上の算術シェアにビット変換をさせる秘密計算プログラムであって、
前記秘密計算サーバ装置のそれぞれが、
前記論理シェアから、自己が保持していないサブシェアをゼロに設定することで他の秘密計算サーバ装置との通信を伴うことなく算術シェアに再分散し、
前記論理シェアをビット変換した算術シェアを得るために、前記再分散された算術シェアを用いて通信を伴った秘密計算を行い、
少なくとも３台以上の秘密計算サーバ装置から受信した同一の値となるはずの受信値を比較し、少なくとも２つ以上が同一である受信値を正しい値として採用することで、前記通信を伴った秘密計算における受信値を検証する、秘密計算プログラム。

【０１１６】

なお、引用した上記の特許文献の開示は、本書に引用をもって繰り込むものとする。本発明の全開示（請求の範囲を含む）の枠内において、さらにその基本的技術思想に基づいて、実施形態ないし実施例の変更・調整が可能である。また、本発明の全開示の枠内において種々の開示要素（各請求項の各要素、各実施形態ないし実施例の各要素、各図面の各要素等を含む）の多様な組み合わせ、ないし、選択（部分的削除を含む）が可能である。すなわち、本発明は、請求の範囲を含む全開示、技術的思想にしたがって当業者であればなし得るであろう各種変形、修正を含むことは勿論である。特に、本書に記載した数値範囲については、当該範囲内に含まれる任意の数値ないし小範囲が、別段の記載のない場合でも具体的に記載されているものと解釈されるべきである。さらに、上記引用した文献の各開示事項は、必要に応じ、本発明の趣旨に則り、本発明の開示の一部として、その一部又は全部を、本書の記載事項と組み合わせて用いることも、本願の開示事項に含まれるものと、みなされる。

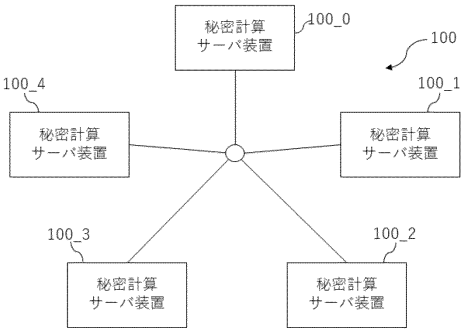
【符号の説明】

【０１１７】

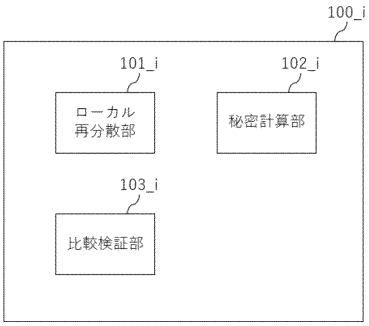
１００，２００，３００　秘密計算システム
１００_ｉ，２００_ｉ，３００_ｉ　秘密計算サーバ装置
１０１_ｉ，２０１_ｉ，３０１_ｉ　ローカル再分散部
１０２_ｉ，２０２_ｉ，３０２_ｉ　秘密計算部
１０３_ｉ，２０３_ｉ，３０３_ｉ　比較検証部
３０４_ｉ　再分散部
１０　ハードウェア構成
１１　ＣＰＵ（Central Processing Unit）
１２　主記憶装置
１３　補助記憶装置
１４　ＩＦ（Interface）部

【図面】

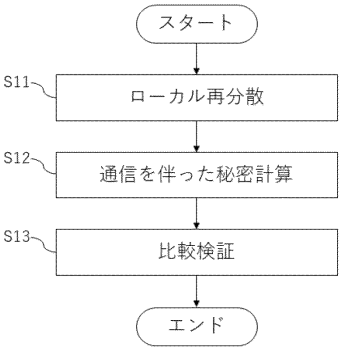
【図１】


秘密計算
サーバ装置
100_0
100
100_4
100_1
秘密計算
サーバ装置
秘密計算
サーバ装置
100_3
100_2
秘密計算
サーバ装置
秘密計算
サーバ装置

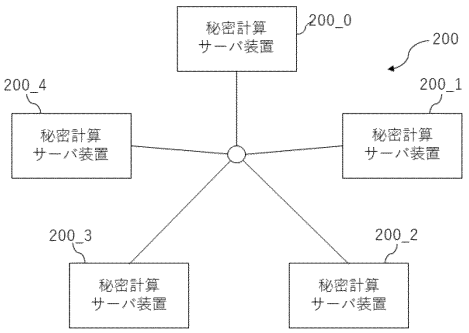
【図２】


100_i
101_i
102_i
ローカル
再分散部
秘密計算部
103_i
比較検証部

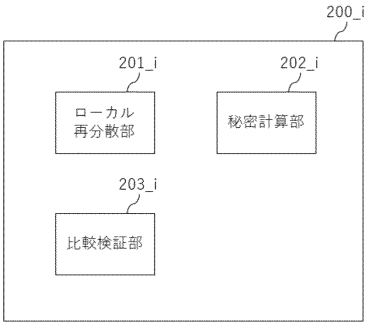
【図３】


スタート
S11
ローカル再分散
S12
通信を伴った秘密計算
S13
比較検証
エンド

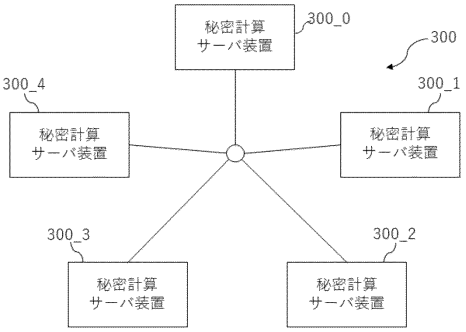
【図４】


秘密計算
サーバ装置
200_0
200
200_4
200_1
秘密計算
サーバ装置
秘密計算
サーバ装置
200_3
200_2
秘密計算
サーバ装置
秘密計算
サーバ装置

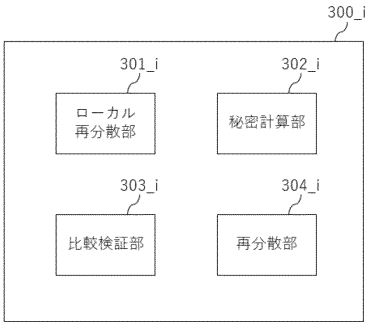
【図５】


200_i
201_i
ローカル
再分散部
202_i
秘密計算部
203_i
比較検証部

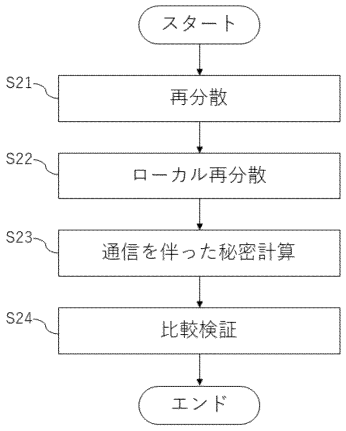
【図６】


秘密計算
サーバ装置
300_0
300
300_4
秘密計算
サーバ装置
300_1
秘密計算
サーバ装置
300_3
秘密計算
サーバ装置
300_2
秘密計算
サーバ装置

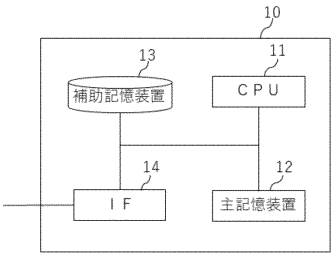
【図７】


300_i
301_i
ローカル
再分散部
302_i
秘密計算部
303_i
比較検証部
304_i
再分散部

【図８】


スタート
S21
再分散
S22
ローカル再分散
S23
通信を伴った秘密計算
S24
比較検証
エンド

【図９】



フロントページの続き

(56)参考文献 国際公開第２０１８／０５３１８５（WO，Ａ１）
加藤遼, 西出隆志, 吉浦裕 ，部分的に小さな法を用いたマルチパーティ計算のビット演算効率化，情報処理学会論文誌（ジャーナル） Ｖｏｌ．５５ Ｎｏ．９ ［ｏｎｌｉｎｅ］ ，2014年09月15日，第55巻 第9号，pp.1971-1991
KIKUCHI, R. et al.，Efficient Bit-Decomposition and Modulus-Conversion Protocols with an Honest Majority，Cryptology ePrint Archive，Report 2018/387，[online]，2018年04月，pp.1-19，URL:https://eprint.iacr.org/2018/387

(58)調査した分野 (Int.Cl.，ＤＢ名)
Ｇ０６Ｆ ２１／６０
ＪＳＴＰｌｕｓ／ＪＭＥＤＰｌｕｓ／ＪＳＴ７５８０（ＪＤｒｅａｍＩＩＩ）
ＩＥＥＥ Ｘｐｌｏｒｅ