(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2019年11月28日(28.11.2019)

WIPO | PCT

(10) 国際公開番号
WO 2019/225401 A1

(51) 国際特許分類:
G09C 1/00 (2006.01)

(21) 国際出願番号： PCT/JP2019/019093

(22) 国際出願日： 2019年5月14日(14.05.2019)

(25) 国際出願の言語： 日本語

(26) 国際公開の言語： 日本語

(30) 優先権データ：
特願 2018-100626 2018年5月25日(25.05.2018) JP

(71) 出願人: 日本電信電話株式会社 (**NIPPON TELEGRAPH AND TELEPHONE CORPORATION**) [JP/JP]; 〒1008116 東京都千代田区大手町一丁目５番１号 Tokyo (JP).

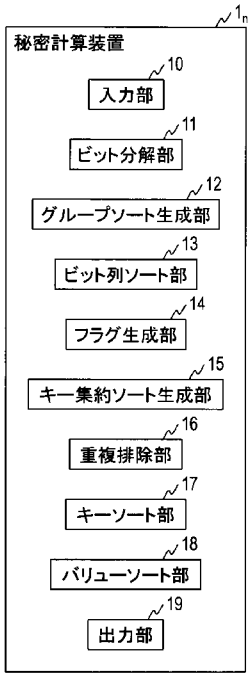
(72) 発明者: 五十嵐 大(**IKARASHI, Dai**); 〒1808585 東京都武蔵野市緑町三丁目９番１１号 ＮＴＴ 知的財産センタ内 Tokyo (JP). 濱田 浩気(**HAMADA, Koki**); 〒1808585 東京都武蔵野市緑町三丁目９番１１号 ＮＴＴ 知的財産センタ内 Tokyo (JP).

(74) 代理人: 中尾 直樹, 外 (**NAKAO, Naoki** et al.); 〒1600022 東京都新宿区新宿三丁目１番２２号 新宿ＮＳＯビル６階 Tokyo (JP).

(81) 指定国(表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,

(54) **Title:** SECRET AGGREGATE FUNCTION CALCULATION SYSTEM, SECRET CALCULATION DEVICE, SECRET AGGREGATE FUNCTION CALCULATION METHOD, AND PROGRAM

(54) 発明の名称： 秘密集約関数計算システム、秘密計算装置、秘密集約関数計算方法、およびプログラム



1n Secret calculation device
10 Input unit
11 Bit decomposition unit
12 Group sort generation unit
13 Bit stream sorting unit
14 Flag generation unit
15 Key aggregate sort generation unit
16 Overlap exclusion unit
17 Key sort unit
18 Value sort unit
19 Output unit

図2

(57) **Abstract:** In the present invention, intermediate data to be used in an aggregate function is efficiently obtained while secrecy is maintained. A bit decomposition unit (11) generates a shear of a bit stream obtained by decomposing a key attribute into bits and combining the bits together. A group sort generation unit (12) generates a shear of first substitution for performing stable sorting of bit streams in ascending order. A bit stream sort unit (13) generates a shear of a sorted bit

WO 2019/225401 A1

[続葉有]

HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) 指定国(表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, RU, TJ, TM), ヨーロッパ (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

添付公開書類:

一 国際調査報告 (条約第21条(3))

stream obtained by performing sorting of the bit streams with the first substitution. A flag generation unit (14) generates a shear of a flag indicating a group boundary. A key aggregate sort generation unit (15) generates a shear of second substitution for performing stable sorting of negative flags in ascending order. An overlap exclusion unit (16) generates a shear of an overlap excluded key attribute. A key sort unit (17) generates a shear of a sorted key attribute obtained by performing sorting of the overlap excluded key attributes with the first substitution and the second substitution in order. A value sort unit (18) generates a shear of a sorted value attribute obtained by performing sorting of value attributes with the first substitution.

(57) 要約:秘匿性を保ったまま集約関数で用いる中間データを効率的に求める。ビット分解部(１１)は、キー属性をビット分解して結合したビット列のシェアを生成する。グループソート生成部(１２)は、ビット列を昇順に安定ソートする第一の置換のシェアを生成する。ビット列ソート部(１３)は、ビット列を第一の置換でソートしたソート済みビット列のシェアを生成する。フラグ生成部(１４)は、グループの境界を示すフラグのシェアを生成する。キー集約ソート生成部(１５)は、フラグの否定を昇順に安定ソートする第二の置換のシェアを生成する。重複排除部(１６)は、重複排除済みキー属性のシェアを生成する。キーソート部(１７)は、重複排除済みキー属性を第一の置換と第二の置換とで順にソートしたソート済みキー属性のシェアを生成する。バリューソート部(１８)は、バリュー属性を第一の置換でソートしたソート済みバリュー属性のシェアを生成する。

明　細　書

発明の名称 ：

秘密集約関数計算システム、秘密計算装置、秘密集約関数計算方法、およびプログラム

技術分野

[0001] この発明は秘密計算技術に関し、特に、秘匿性を保ったまま集約関数を計算する技術に関する。

背景技術

[0002] 集約関数は、テーブルにキー属性とバリュー属性があるときに、キー属性の値に基づいてグループ分けした統計値を得る演算である。集約関数は、group-by演算とも呼ばれる。キー属性は、テーブルのレコードをグループ分けするために用いる属性であり、例えば、役職や性別などが挙げられる。バリュー属性は、統計値を計算するために用いる属性であり、例えば、給料や身長などが挙げられる。group-by演算は、例えば、キー属性が性別のときに、男女別の平均身長を求める演算などである。キー属性は複数の属性による複合キーであってもよく、例えば、キー属性が性別と年齢のときに、10代男性の平均身長、20代男性の平均身長、・・・を得るような演算であってもよい。非特許文献１には、group-by演算を秘密計算で行う方法が記載されている。

[0003] group-by演算は、具体的には、group-byカウント、group-by総和、group-by最大値／最小値、group-by中央値、グループ内の順位などがある。group-byカウントは、クロス集計のことであり、テーブルをキー属性の値に基づいてグループ分けしたときに、各グループのレコード数を集計する演算である。group-by総和は、各グループにおける所望のバリュー属性の総和である。group-by最大値／最小値は、各グループにおける所望のバリュー属性の最大値／最小値である。group-by中央値は、各グループにおける所望のバリュー属性の中央値である。グループ内の順位は、各レコードのバリュー属性の値がグループ内で何番目の値であるかを取得する関数である。

先行技術文献

非特許文献

[0004] 非特許文献1：五十嵐大，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”，２０１１年暗号と情報セキュリティシンポジウム

発明の概要

発明が解決しようとする課題

[0005] group-by演算では計算の過程で中間データを求めることがある。この中間データの中には、異なる種類のgroup-by演算の間で共通に求めるものがある。秘匿性を保ったまま複数のgroup-by演算を同時にまたは連続して計算すると、共通の中間データを求める処理が重複することで、計算量が増大する場合がある。

[0006] この発明の目的は、上記のような技術的課題に鑑みて、秘匿性を保ったまま複数のgroup-by演算を同時にまたは連続して計算するときに、group-by演算で用いる中間データを効率的に求めることができる技術を提供することである。

課題を解決するための手段

[0007] 上記の課題を解決するために、この発明の一態様の秘密集約関数計算システムは、複数の秘密計算装置を含む秘密集約関数計算システムであって、Fは任意の環であり、mは２以上の整数であり、n_kは１以上の整数であり、$[k_0]$, …, $[k_{nk-1}]$はキー属性k_0, …, $k_{nk-1} \in F^m$を秘密分散したシェアであり、秘密計算装置は、シェア$[k_0]$, …, $[k_{nk-1}]$を用いて、復元するとキー属性k_0, …, k_{nk-1}をビット分解して結合したビット列$b:=b_0$, …, b_{m-1}となるシェア$\{b\}$から、復元するとビット列bを昇順に安定ソートする置換σ_0となるシェア$\{\{\sigma_0\}\}$を生成するグループソート生成部と、シェア$\{b\}$とシェア$\{\{\sigma_0\}\}$とを用いて、復元するとビット列bを置換σ_0でソートしたソート済みビット列$b':=b'_0$, …, b'_{m-1}となるシェア$\{b'\}$を生成するビット列ソート部と、シェア$\{b'\}$を用いて、0以上

m-2以下の各整数iについて{e_i}:={$b'_i \neq b'_{i+1}$}を設定し、かつ、{e_{m-1}}:={1}を設定して、復元するとフラグe:=e_0, …, e_{m-1}となるシェア{e}を生成するフラグ生成部と、シェア{e}を用いて、復元するとフラグeの否定¬eを昇順に安定ソートする置換σとなるシェア{{σ}}を生成するキー集約ソート生成部と、を含む。

発明の効果

[0008] この発明の秘密集約関数技術によれば、秘匿性を保ったままgroup-by演算で用いる中間データを効率的に求めることができる。その中間データを用いることで、複数のgroup-by演算を同時にまたは連続して計算するときに、全体の計算量を削減することができる。

図面の簡単な説明

[0009] [図1]図１は、秘密集約関数計算システムの機能構成を例示する図である。

[図2]図２は、秘密計算装置の機能構成を例示する図である。

[図3]図３は、秘密集約関数計算方法の処理手続きを例示する図である。

発明を実施するための形態

[0010] 以下、この発明の実施の形態について詳細に説明する。なお、図面中において同じ機能を有する構成部には同じ番号を付し、重複説明を省略する。

[0011] [x]∈[F]は、ある値xが任意の環F上の秘密分散等により秘匿されていることを表す。{b}∈{B}は、１ビットのある値bが１ビットを表せる環B上の秘密分散等により秘匿されていることを表す。{{s}}∈{{S_m}}は、m個の要素の置換の集合S_mに属するある置換sが秘密分散等により秘匿されていることを表す。以下、秘密分散された値を「シェア」とも呼ぶ。

[0012] 実施形態中で用いる秘密計算におけるソート（安定ソートを含む）は、例えば、下記参考文献１に記載されたソートを用いることができる。置換sのシェア{{s}}については下記参考文献１に記載されたハイブリッド置換{{π}}を用いればよい。

[0013] 〔参考文献１〕五十嵐大，濱田浩気，菊池亮，千田浩司，“超高速秘密計算ソートの設計と実装：秘密計算がスクリプト言語に並ぶ日”，ＣＳＳ２０

１７

[0014] ＜実施形態＞

図１を参照して、実施形態の秘密集約関数計算システム１００の構成例を説明する。秘密集約関数計算システム１００は、N（≧2）台の秘密計算装置1_1，…，1_Nを含む。本形態では、秘密計算装置1_1，…，1_Nはそれぞれ通信網２へ接続される。通信網２は、接続される各装置が相互に通信可能なように構成された回線交換方式もしくはパケット交換方式の通信網であり、例えばインターネットやLAN（Local Area Network）、WAN（Wide Area Network）などを用いることができる。なお、各装置は必ずしも通信網２を介してオンラインで通信可能である必要はない。例えば、秘密計算装置1_1，…，1_Nへ入力する情報を磁気テープやUSBメモリなどの可搬型記録媒体に記憶し、その可搬型記録媒体から秘密計算装置1_1，…，1_Nへオフラインで入力するように構成してもよい。

[0015] 図２を参照して、秘密集約関数計算システム１００に含まれる秘密計算装置1_n（n=1，…，N）の構成例を説明する。秘密計算装置1_nは、例えば、図２に示すように、入力部１０、ビット分解部１１、グループソート生成部１２、ビット列ソート部１３、フラグ生成部１４、キー集約ソート生成部１５、重複排除部１６、キーソート部１７、バリューソート部１８、および出力部１９を含む。この秘密計算装置1_n（$1 \leqq n \leqq N$）が他の秘密計算装置$1_{n'}$（n'=1，…，N、ただし$n \neq n'$）と協調しながら後述する各ステップの処理を行うことにより実施形態の秘密集約関数計算方法が実現される。

[0016] 秘密計算装置1_n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。秘密計算装置1_n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。秘密計算装置1_n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。秘密

計算装置1_n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。

[0017] 図３を参照して、実施形態の秘密集約関数計算システム１００が実行する秘密集約関数計算方法の処理手続きを説明する。

[0018] ステップＳ１０において、各秘密計算装置1_nの入力部１０は、n_k個のキー属性k_0，…，$k_{nk-1}\in F^m$それぞれを秘密分散により秘匿したシェア$[k_0]$，…，$[k_{nk-1}]\in[F]^m$と、n_a個のバリュー属性v_0，…，$v_{na-1}\in F^m$それぞれを秘密分散により秘匿したシェア$[v_0]$，…，$[v_{na-1}]\in[F]^m$とを入力として受け取る。ただし、n_k，n_aは１以上の整数であり、mは２以上の整数である。以下、$[k_j]\in[F]^m$（j=0，…，n_k-1）の各要素は、$[k_{j,i}]\in[F]$（i=0，…，m-1）で参照することもある。また、$[v_h]\in[F]^m$（h=0，…，n_a-1）の各要素は、$[v_{h,i}]\in[F]$（i=0，…，m-1）で参照することもある。入力部１０は、キー属性k_0，…，k_{nk-1}のシェア$[k_0]$，…，$[k_{nk-1}]$をビット分解部１１と重複排除部１６へ出力する。また、入力部１０は、バリュー属性v_0，…，v_{na-1}のシェア$[v_0]$，…，$[v_{na-1}]$をバリューソート部１８へ出力する。

[0019] ステップＳ１１において、各秘密計算装置1_nのビット分解部１１は、キー属性k_0，…，k_{nk-1}のシェア$[k_0]$，…，$[k_{nk-1}]$をビット分解して結合し、復元するとキー属性k_0，…，k_{nk-1}のビット表現を結合したビット列$b:=b_0$，…，$b_{m-1}\in B^\lambda$となるシェア$\{b\}\in\{B\}^\lambda$を得る。ただし、λはビット列bのビット長であり、各b_i（i=0，…，m-1）のビット長の総和である。言い替えると、$\{b_i\}$は、キー属性k_0，…，k_{nk-1}のシェア$[k_0]$，…，$[k_{nk-1}]$それぞれのi番目の要素$[k_{0,i}]$，…，$[k_{nk-1,i}]$のビット表現を結合したビット列である。ビット分解部１１は、ビット列bのシェア$\{b\}$をグループソート生成部１２へ出力する。

[0020] ステップＳ１２において、各秘密計算装置1_nのグループソート生成部１２は、ビット列bのシェア$\{b\}$を用いて、復元するとビット列bを昇順で安定ソートするための置換σ_0となるシェア$\{\{\sigma_0\}\}\in\{\{S_m\}\}$を生成する。安定ソートとは、ソート演算のうち、同じ値の要素が存在した場合に、同じ値の要素同士の順序を保存する演算である。例えば、社員番号順でソートされたテーブル

に対して性別で安定ソートすると、各性別の中で社員番号順が保たれているソート結果が得られる。ビット列bはキー属性k_0，…，k_{nk-1}のビット表現を結合したものであるため、置換σ_0はキー属性k_0，…，k_{nk-1}の値が等しいレコードを連続するように並び替えてグループ分けする操作であるとも言える。グループソート生成部１２は、ビット列bのシェア{b}と置換σ_0のシェア{{σ_0}}とをビット列ソート部１３へ出力する。また、グループソート生成部１２は、置換σ_0のシェア{{σ_0}}をキーソート部１７とバリューソート部１８へ出力する。

[0021] ステップＳ１３において、各秘密計算装置1_nのビット列ソート部１３は、ビット列bのシェア{b}と置換σ_0のシェア{{σ_0}}とを用いて、復元するとビット列bを置換σ_0でソートしたソート済みビット列b':=b'$_0$，…，b'$_{m-1}$∈B^λとなるシェア{b'}∈{B}$^\lambda$を得る。ビット列ソート部１３は、ソート済みビット列b'のシェア{b'}をフラグ生成部１４へ出力する。

[0022] ステップＳ１４において、各秘密計算装置1_nのフラグ生成部１４は、ソート済みビット列b'のシェア{b'}を用いて、0以上m-2以下の各整数iについて{e_i}:={b'$_i$≠b'$_{i+1}$}を設定し、かつ、{e_{m-1}}:={1}を設定して、復元するとフラグe:=e_0，…，e_{m-1}∈B^mとなるシェア{e}∈{B}^{m}を生成する。フラグe_iはソート済みビット列b'のi番目の要素b'$_i$がi+1番目の要素b'$_{i+1}$と異なる場合に真が設定されるため、各グループの最後の要素（すなわち、グループ間の境界の直前の要素）を示すフラグとなる。フラグ生成部１４は、フラグeのシェア{e}をキー集約ソート生成部１５と重複排除部１６と出力部１９へ出力する。

[0023] ステップＳ１５において、各秘密計算装置1_nのキー集約ソート生成部１５は、まず、フラグeのシェア{e}を用いて、復元するとフラグeの否定¬eであるフラグe'となるシェア{e'}∈{B}^{m}を生成する。すなわち、0以上m-1以下の各整数iについて{e'$_i$}:={¬e_i}を設定する。次に、キー集約ソート生成部１５は、フラグe'のシェア{e'}を用いて、復元するとフラグe'を昇順に安定ソートするための置換σとなるシェア{{σ}}∈{{S_m}}を生成する。キー集約ソート生成部１５は、置換σのシェア{{σ}}をキーソート部１７と出力部１９へ出力

する。

[0024] ステップＳ１６において、各秘密計算装置１$_n$の重複排除部１６は、フラグeのシェア{e}とキー属性k_0，…，k_{nk-1}のシェア$[k_0]$，…，$[k_{nk-1}]$とを用いて、$[k''_{j,i}]:=[e_i?k_{j,i}:null]$を設定し、復元すると重複排除済みキー属性$k''_0$，…，$k''_{nk-1}$となるシェア$[k''_0]$，…，$[k''_{nk-1}]$を生成する。ここで、「?」は条件演算子（または三項演算子）である。すなわち、$\{e_i\}$が真（例えば、$\{e_i\}=\{1\}$）のときは$[k''_{j,i}]:=[k_{j,i}]$を設定し、$\{e_i\}$が偽（例えば、$\{e_i\}=\{0\}$）のときは$[k''_{j,i}]:=null$を設定する。$\{e_i\}=\{0\}$のときに設定する値はnullでなくともよく、キー属性k_0，…，k_{nk-1}が取り得ない値であればどのような値でもよい。フラグeは各グループの最後の要素のみに真が設定されたフラグであるため、重複排除済みキー属性k''_0，…，k''_{nk-1}は、各グループの最後の要素に対応する要素のみキー属性の値が設定され、それ以外の要素はキー属性が取り得ない所定の値が設定されたベクトルとなる。重複排除部１６は、重複排除済みキー属性k''_0，…，k''_{nk-1}のシェア$[k''_0]$，…，$[k''_{nk-1}]$をキーソート部１７へ出力する。

[0025] ステップＳ１７において、各秘密計算装置１$_n$のキーソート部１７は、重複排除済みキー属性k''_0，…，k''_{nk-1}のシェア$[k''_0]$，…，$[k''_{nk-1}]$と置換σ_0のシェア$\{\{\sigma_0\}\}$と置換σのシェア$\{\{\sigma\}\}$とを用いて、復元すると重複排除済みキー属性k''_0，…，k''_{nk-1}を置換σ_0と置換σとで順にソートしたソート済みキー属性k'_0，…，k'_{nk-1}となるシェア$[k'_0]$，…，$[k'_{nk-1}]$を生成する。キーソート部１７は、ソート済みキー属性k'_0，…，k'_{nk-1}のシェア$[k'_0]$，…，$[k'_{nk-1}]$を出力部１９へ出力する。

[0026] ステップＳ１８において、各秘密計算装置１$_n$のバリューソート部１８は、バリュー属性v_0，…，v_{na-1}のシェア$[v_0]$，…，$[v_{na-1}]$と置換σ_0のシェア$\{\{\sigma_0\}\}$とを用いて、復元するとバリュー属性v_0，…，v_{na-1}を置換σ_0でソートしたソート済みバリュー属性v'_0，…，v'_{na-1}となるシェア$[v'_0]$，…，$[v'_{na-1}]$を生成する。バリューソート部１８は、ソート済みバリュー属性v'_0，…，v'_{na-1}のシェア$[v'_0]$，…，$[v'_{na-1}]$を出力部１９へ出力する。

[0027] ステップＳ１９において、各秘密計算装置１$_n$の出力部１９は、ソート済み

キー属性k'_0, …, k'_{nk-1}のシェア$[k'_0]$, …, $[k'_{nk-1}]$、ソート済みバリュー属性v'_0, …, v'_{na-1}のシェア$[v'_0]$, …, $[v'_{na-1}]$、フラグeのシェア{e}、および置換σのシェア{{σ}}の少なくとも１つを出力する。出力部１９が出力すべき情報は、後に続いて計算する１つ以上のgroup-by演算で必要とされる中間データを満たすように選択する。

[0028] 以下、秘密集約関数計算システム１００が出力する中間データを用いて、各種の集約関数を計算する具体的な手順を説明する。

[0029] ≪group-byカウント≫

group-byカウントは、テーブルをキー属性の値に基づいてグループ分けしたときに、各グループのレコード数を集計する演算である。group-byカウントは、秘密集約関数計算システム１００が出力するフラグeのシェア{e}と置換σのシェア{{σ}}とを用いて、以下のように求めることができる。なお、gは最大グループ数であり、キー属性が取り得る値の組み合わせの数、すなわち、キー属性が取り得る値の種類の数である。

[0030] 第一に、フラグeのシェア$\{e\}\in\{B\}^m$を任意の環F上の秘密分散によるシェア$[e]\in[F]^m$に変換する。

[0031] 第二に、フラグeのシェア[e]を用いて、0以上m-1以下の各整数iについて$[x_i]:=[e_i?i+1:m]$を設定し、復元するとベクトル$x:=x_0, \cdots, x_{m-1}\in F$となるシェア$[x]\in[F]^m$を生成する。ベクトルxは、テーブルをキー属性で安定ソートしたときに同じキー属性の値をもつレコードを同じグループとして、各グループの最後の要素には次の要素の先頭からの位置が設定され、その他の要素にはテーブル全体のレコード数が設定されたベクトルとなる。言い替えると、各グループの最後の要素には、先頭のグループからそのグループまでの各グループのレコード数を積み上げた合計値が設定されることになる。

[0032] 第三に、ベクトルxのシェア[x]と置換σのシェア{{σ}}とを用いて、復元するとベクトルxを置換σでソートしたソート済みベクトルσ(x)となるシェア$[\sigma(x)]\in[F]^m$を生成する。以下、$[\sigma(x)]\in[F]^m$の各要素は、$[\sigma(x)_i]\in[F]$（i=0, …, m-1）で参照することもある。

[0033] 最後に、ソート済みベクトルσ(x)のシェア[σ(x)]を用いて、1以上min(g,m)-1以下の各整数iについて[c_i]:=[$\sigma(x)_i-\sigma(x)_{i-1}$]を設定し、かつ、[$c_0$]:=[$\sigma(x)_0$]を設定して、復元すると各グループのレコード数を表すベクトルc:=c_0, …, $c_{min(g,m)-1}$∈Fとなるシェア[c]∈[F]$^{min(g,m)}$を生成する。ソート済みベクトルσ(x)のi番目の要素$\sigma(x)_i$は、0番目からi番目までの各グループのレコード数を積み上げた合計値が設定されているため、ベクトルcのi番目の要素c_iには、i番目のグループのレコード数が設定されることになる。なお、キー属性は秘匿されているため、min(g,m)はグループ数が取り得る最大値であり、実際のグループ数はmin(g,m)以下の各秘密計算装置1_nには知り得ない値（以下、実際のグループ数をg'とする）となる。したがって、min(g,m)個のシェア[c_i]の中で実際のグループ数を超えるもの（すなわち、i≧g'）には、復元後に有効な値と識別可能となる無効な値を設定しておく必要がある。本形態では、[e_i]が偽のシェア[x_i]もしくは[e_i]が真のうち最後のシェア[x_i]には[x_i]=mを設定している。これにより、$c_{g'}$, …, $c_{min(g,m)-1}$には$\sigma(x)_i-\sigma(x)_{i-1}$=m-m=0が設定される。レコードが存在するグループのカウント数は１以上であるため、０は有効な値と識別可能となる無効な値として成立している。

[0034] ≪group-by総和≫

group-by総和は、テーブルをキー属性の値に基づいてグループ分けしたときに、グループごとに所望のバリュー属性の総和を集計する演算である。group-by総和を用いれば、グループごとに乗算の和を求めるgroup-by積和や、グループごとに二乗の和を求めるgroup-by二乗和も計算することができる。group-by積和であれば、各レコードのバリュー属性に乗算を施した結果に対してgroup-by総和を求めればよい。また、group-by二乗和であれば、同様に、各レコードのバリュー属性に二乗を施した結果に対してgroup-by総和を求めればよい。group-by総和は、秘密集約関数計算システム１００が出力するソート済みバリュー属性v'_0, …, v'_{na-1}のシェア[v'_0], …, [v'_{na-1}]とフラグeのシェア{e}と置換σのシェア{{σ}}とを用いて、以下のように求めることができる。なお、vはソート済みバリュー属性v'_0, …, v'_{na-1}のうちgroup-by総和を求

めたい所望のバリュー属性である。

[0035] 第一に、バリュー属性vのシェア[v]を用いて、[v']:=prefix-sum([v])を計算し、復元するとベクトルv':=v'$_0$, …, v'$_{m-1}$∈Fとなるシェア[v']∈[F]^{m}を生成する。prefix-sumは、mを入力ベクトルvの長さとして、0以上m-1以下の各整数iについて、出力ベクトルv'のi番目の要素v'$_i$には入力ベクトルvの0番目の要素v$_0$からi番目の要素v$_i$までの値の総和を設定する演算である。

[0036] 第二に、フラグeのシェア{e}∈{B}^{m}を任意の環F上の秘密分散によるシェア[e]∈[F]^{m}に変換する。

[0037] 第三に、ベクトルv'のシェア[v']とフラグeのシェア[e]とを用いて、0以上m-1以下の各整数iについて[t$_i$]:=[e$_i$?v'$_i$:v'$_{m-1}$]を設定し、復元するとベクトルt:=t$_0$, …, t$_{m-1}$∈Fとなるシェア[t]∈[F]^{m}を生成する。ベクトルtは、テーブルをキー属性で安定ソートしたときに同じキー属性の値をもつレコードを同じグループとして、各グループの最後の要素にはその要素以前のバリュー属性の値の総和が設定され、その他の要素にはテーブル全体のバリュー属性の値の総和が設定されたベクトルとなる。

[0038] 第四に、ベクトルtのシェア[t]と置換σのシェア{{σ}}とを用いて、復元するとベクトルtを置換σでソートしたソート済みベクトルσ(t)となるシェア[σ(t)]∈[F]^{m}を生成する。以下、[σ(t)]∈[F]^{m}の各要素は、[σ(t)$_i$]∈[F]（i=0, …, m-1）で参照することもある。

[0039] 最後に、ソート済みベクトルσ(t)のシェア[σ(t)]を用いて、1以上min(g, m)-1以下の各整数iについて[s$_i$]:=[σ(t)$_i$-σ(t)$_{i-1}$]を設定し、かつ、[t$_0$]:=[σ(t)$_0$]を設定して、復元するとグループ毎のバリュー属性vの総和s:=s$_0$, …, s$_{min(g,m)-1}$∈Fとなるシェア[s]∈[F]$^{min(g,m)}$を生成する。ソート済みベクトルσ(t)のi番目の要素σ(t)$_i$は、0番目からi番目までの各グループに属するバリュー属性vの値の総和が設定されているため、ベクトルtのi番目の要素t$_i$には、i番目のグループに属するバリュー属性vの値の総和が設定されることになる。

[0040] ≪group-by最大値≫

group-by最大値は、テーブルをキー属性の値に基づいてグループ分けした

ときに、グループごとに所望のバリュー属性の最大値を得る演算である。group-by最大値は、秘密集約関数計算システム１００が出力するソート済みバリュー属性v'_0，…，v'_{na-1}のシェア$[v'_0]$，…，$[v'_{na-1}]$とフラグeのシェア{e}と置換σのシェア{{σ}}とを用いて、以下のように求めることができる。なお、vはソート済みバリュー属性v'_0，…，v'_{na-1}のうちgroup-by最大値を求めたい所望のバリュー属性である。

[0041] 第一に、フラグeのシェア$\{e\}\in\{B\}^m$を任意の環F上の秘密分散によるシェア$[e]\in[F]^m$に変換する。

[0042] 第二に、バリュー属性vのシェア[v]とフラグeのシェア[e]とを用いて、0以上m-1以下の各整数iについて$[f_i]:=[e_i?v_i:0]$を設定し、復元するとベクトルf:=f_0，…，$f_{m-1}\in$Fとなるシェア$[f]\in[F]^m$を生成する。ベクトルfは、テーブルをキー属性で安定ソートしたときに同じキー属性の値をもつレコードを同じグループとして、各グループの最後の要素f_iにはその要素に対応するバリュー属性の値v_iが設定され、その他の要素には０が設定されたベクトルとなる。すなわち、各グループの最大値と０とを要素としてもつベクトルとなる。

[0043] 第三に、ベクトルfのシェア[f]と置換σのシェア{{σ}}とを用いて、復元するとベクトルfを置換σでソートしたソート済みベクトルσ(f)となるシェア$[\sigma(f)]\in[F]^m$を生成する。以下、$[\sigma(f)]\in[F]^m$の各要素は、$[\sigma(f)_i]\in[F]$（i=0，…，m-1）で参照することもある。ソート済みベクトルσ(f)は、先頭からグループ数の要素にグループ毎にソートしたときの最後の要素の値（すなわち、各グループの最大値）が設定され、それ以降の要素に０が設定されたベクトルとなる。

[0044] 最後に、ソート済みベクトルσ(f)のシェア$[\sigma(f)]$から、復元すると各グループの最大値を表すベクトルx:=$\sigma(f)_0$，…，$\sigma(f)_{min(g,m)-1}$となるシェア$[x]\in[F]^{min(g,m)}$を生成する。

[0045] ≪group-by最小値≫

group-by最小値は、テーブルをキー属性の値に基づいてグループ分けしたときに、グループごとに所望のバリュー属性の最小値を得る演算である。gro

up-by最小値は、秘密集約関数計算システム１００が出力するソート済みバリュー属性v'_0, …, v'_{na-1}のシェア$[v'_0]$, …, $[v'_{na-1}]$とフラグeのシェア{e}と置換σのシェア{{σ}}とを用いて、以下のように求めることができる。なお、vはソート済みバリュー属性v'_0, …, v'_{na-1}のうちgroup-by最小値を求めたい所望のバリュー属性である。

[0046] 第一に、フラグeのシェア{e}を用いて、1以上m-1以下の各整数iについて$\{e'_i\}:=\{e_{i-1}\}$を設定し、かつ、$\{e'_0\}:=\{1\}$を設定して、復元するとフラグ$e':=e'_0$, …, $e'_{m-1}\in B^m$となるシェア$\{e'\}\in\{B\}^m$を生成する。フラグe'は各グループの最後の要素を示すフラグeを一つずつ後方へシフトしたフラグであるため、各グループの最初の要素（すなわち、グループ間の境界の直後の要素）を示すフラグとなる。

[0047] 第二に、フラグe'のシェア$\{e'\}\in\{B\}^m$を任意の環F上の秘密分散によるシェア$[e']\in[F]^m$に変換する。

[0048] 第三に、バリュー属性vのシェア[v]とフラグe'のシェア[e']とを用いて、0以上m-1以下の各整数iについて$[f'_i]:=[e'_i?v_i:0]$を設定し、復元するとベクトル$f':=f'_0$, …, $f'_{m-1}\in F$となるシェア$[f']\in[F]^m$を生成する。ベクトルf'は、テーブルをキー属性で安定ソートしたときに同じキー属性の値をもつレコードを同じグループとして、各グループの最初の要素f'_iにはその要素に対応するバリュー属性の値v_iが設定され、その他の要素には０が設定されたベクトルとなる。すなわち、各グループの最小値と０とを要素としてもつベクトルとなる。

[0049] 第四に、ベクトルf'のシェア[f']と置換σのシェア{{σ}}とを用いて、復元するとベクトルf'を置換σでソートしたソート済みベクトルσ(f')となるシェア$[\sigma(f')]\in[F]^m$を生成する。以下、$[\sigma(f')]\in[F]^m$の各要素は、$[\sigma(f')_i]\in[F]$（i=0, …, m-1）で参照することもある。ソート済みベクトルσ(f')は、先頭からグループ数の要素にグループ毎にソートしたときの最初の要素の値（すなわち、各グループの最小値）が設定され、それ以降の要素に０が設定されたベクトルとなる。

[0050] 最後に、ソート済みベクトルσ(f')のシェア[σ(f')]から、復元すると各グループの最小値を表すベクトルx':=σ(f')$_0$, …, σ(f')$_{min(g,m)-1}$となるシェア[x']∈[F]$^{min(g,m)}$を生成する。

[0051] ≪グループ内の昇順順位≫

グループ内の昇順順位は、テーブルをキー属性の値に基づいてグループ分けしたときに、所望のバリュー属性を昇順でソートしたときにその値がグループ内で何番目の値であるかを得る演算である。グループ内の昇順順位は、秘密集約関数計算システム１００が出力するフラグeのシェア{e}と置換σのシェア{{σ}}とを用いて、以下のように求めることができる。なお、cはgroup-byカウントの結果（以下、「クロス集計」と呼ぶ）である。クロス集計cは、例えば、上述の≪group-byカウント≫の手順に従って、フラグeのシェア{e}と置換σのシェア{{σ}}とを用いて求めることができる。

[0052] 第一に、クロス集計cのシェア[c]と置換σのシェア{{σ}}とを用いて、復元するとクロス集計cに置換σを逆適用した逆置換済みクロス集計u:=σ^{-1}(c)となるシェア[u]:=[σ^{-1}(c)]∈[F]^{m}を生成する。クロス集計cは先頭からグループ数までの要素に各グループのレコード数が設定されたベクトルであり、置換σは各グループの最後の要素を先頭から順に並べる置換であるため、クロス集計cに置換σを逆適用した逆置換済みクロス集計uは各グループの最後の要素にそのグループのレコード数が設定されたベクトルとなる。以下、[u]∈[F]^{m}の各要素は、[u$_i$]∈[F]（i=0, …, m-1）で参照することもある。

[0053] 第二に、逆置換済みクロス集計uのシェア[u]を用いて、[s]:=prefix-sum([u])を計算し、復元するとベクトルs:=s$_0$, …, s$_{m-1}$∈Fとなるシェア[s]∈[F]^{m}を生成する。prefix-sumは、mを入力ベクトルuの長さとして、0以上m-1以下の各整数iについて、出力ベクトルsのi番目の要素s$_i$には入力ベクトルuの0番目の要素u$_0$からi番目の要素u$_i$までの値の総和を設定する演算である。

[0054] 最後に、ベクトルsのシェア[s]を用いて、1以上m-1以下の各整数iについて[a$_i$]:=[i-s$_{i-1}$]を設定し、かつ、[a$_0$]:=[0]を設定して、復元するとグループ内での昇順順位a:=a$_0$, …, a$_{m-1}$∈Fとなるシェア[a]∈[F]^{m}を生成する。なお、グ

ループ内での昇順順位は０スタートとなることに注意されたい。１スタートの順位を得たいのであれば、各順位に１を加算すればよい。すなわち、1以上m-1以下の各整数iについて$[a_i]:=[i-s_{i-1}+1]$を設定し、かつ、$[a_0]:=[1]$を設定して、昇順順位aを生成すればよい。

[0055] ≪グループ内の降順順位≫

グループ内の降順順位は、テーブルをキー属性の値に基づいてグループ分けしたときに、所望のバリュー属性を降順でソートしたときにその値がグループ内で何番目の値であるかを得る演算である。グループ内の降順順位は、秘密集約関数計算システム１００が出力するフラグeのシェア{e}と置換σのシェア{{σ}}とを用いて、以下のように求めることができる。なお、cはgroup-byカウントの結果（以下、「クロス集計」と呼ぶ）である。クロス集計cは、例えば、上述の≪group-byカウント≫の手順に従って、フラグeのシェア{e}と置換σのシェア{{σ}}とを用いて求めることができる。

[0056] 第一に、クロス集計cのシェア[c]を用いて、0以上m-2以下の各整数iについて$[c'_i]:=[c_{i+1}]$を設定し、かつ、$[c'_{m-1}]:=[0]$を設定して、復元するとシフト済みクロス集計$c':=c'_0, \cdots, c'_{m-1} \in F^m$となるシェア$[c'] \in [F]^m$を生成する。シフト済みクロス集計c'は各グループのレコード数を表すベクトルであるクロス集計cを一つずつ前方へシフトしたベクトルとなる。

[0057] 第二に、シフト済みクロス集計c'のシェア[c']と置換σのシェア{{σ}}とを用いて、復元するとシフト済みクロス集計c'に置換σを逆適用した逆置換済みクロス集計$u':=\sigma^{-1}(c')$となるシェア$[u']:=[\sigma^{-1}(c')] \in [F]^m$を生成する。シフト済みクロス集計c'は先頭からグループ数までの要素に各グループのレコード数が設定されたクロス集計cを一つずつ前方へシフトしたベクトルであり、置換σは各グループの最後の要素を先頭から順に並べる置換であるため、シフト済みクロス集計c'に置換σを逆適用した逆置換済みクロス集計u'は各グループの最後の要素に一つ後方のグループのレコード数が設定されたベクトルとなる。以下、$[u'] \in [F]^m$の各要素は、$[u'_i] \in [F]$（i=0, …, m-1）で参照することもある。

[0058] 第三に、逆置換済みクロス集計u'のシェア[u']を用いて、[s']:=postfix-sum([u'])を計算し、復元するとベクトルs':=s'$_0$, …, s'$_{m-1}$∈Fとなるシェア[s']∈[F]^{m}を生成する。postfix-sumは、mを入力ベクトルu'の長さとして、0以上m-1以下の各整数iについて、出力ベクトルs'のi番目の要素s'$_i$には入力ベクトルu'のi番目の要素u'$_i$からm-1番目の要素u'$_{m-1}$までの値の総和を設定する演算である。

[0059] 最後に、ベクトルs'のシェア[s']を用いて、0以上m-1以下の各整数iについて[d$_i$]:=[m-i-s'$_i$-1]を設定して、復元するとグループ内での降順順位d:=d$_0$, …, d$_{m-1}$∈Fとなるシェア[d]∈[F]^{m}を生成する。なお、グループ内での降順順位は０スタートとなることに注意されたい。１スタートの順位を得たいのであれば、各順位に１を加算すればよい。すなわち、0以上m-1以下の各整数iについて[d$_i$]:=[m-i-s'$_i$]を設定して、降順順位dを生成すればよい。

[0060] ≪group-by中央値≫

group-by中央値は、テーブルをキー属性の値に基づいてグループ分けしたときに、グループごとに所望のバリュー属性の中央値を得る演算である。group-by中央値は、秘密集約関数計算システム１００が出力するバリュー属性v'$_0$, …, v'$_{na-1}$のシェア[v'$_0$], …, [v'$_{na-1}$]とフラグeのシェア{e}と置換σのシェア{{σ}}とを用いて、以下のように求めることができる。なお、cはgroup-byカウントの結果（以下、「クロス集計」と呼ぶ）である。クロス集計cは、例えば、上述の≪group-byカウント≫の手順に従って、フラグeのシェア{e}と置換σのシェア{{σ}}とを用いて求めることができる。また、vはソート済みバリュー属性v'$_0$, …, v'$_{na-1}$のうちgroup-by中央値を求めたい所望のバリュー属性である。

[0061] 第一に、クロス集計cのシェア[c]と置換σのシェア{{σ}}とを用いて、復元するとグループ内での昇順順位を表すベクトルa:=a$_0$, …, a$_{m-1}$∈Fとなるシェア[a]∈[F]^{m}および復元するとグループ内での降順順位を表すベクトルd:=d$_0$, …, d$_{m-1}$∈Fとなるシェア[d]∈[F]^{m}を生成する。ここで、昇順順位および降順順位は１スタートとする。グループ内での昇順順位は、例えば、上述の≪

グループ内の昇順順位≫の手順に従って求めることができる。グループ内での降順順位は、例えば、上述の≪グループ内の降順順位≫の手順に従って求めることができる。

[0062] 第二に、昇順順位aのシェア[a]と降順順位dのシェア[d]とを用いて、$2^{\lambda}>m$を満たすλに対して、[2^{λ}+a-d], [2^{λ}+d-a]を計算し、[2^{λ}+a-d], [2^{λ}+d-a]をλビットにビット分解して、復元するとビット列a-dなるシェア$\{a-d\}\in\{B_{\lambda}\}^{m}$と、復元するとビット列d-aとなるシェア$\{d-a\}\in\{B_{\lambda}\}^{m}$とを生成する。

[0063] 第三に、a-dのシェア{a-d}とd-aのシェア{d-a}とから最下位ビットを除いて、復元するとa', d'となるシェア{a'}, {d'}$\in\{B_{\lambda-1}\}^{m}$を生成する。a'はa-dの最下位ビットを除いたビット列であり、d'はd-aの最下位ビットを除いたビット列である。

[0064] 第四に、a'のシェア{a'}とd'のシェア{d'}とを用いて、{a"}:={|a'=0|}, {d"}:={|d'=0|}を計算し、復元するとフラグa", d"$\in B^{m}$となるシェア{a"}, {d"}$\in\{B\}^{m}$を生成する。なお、|・|は等式・の真偽を返却する記号である。フラグa", d"は、a-d, d-aそれぞれが0以上1以下であるかどうかを表す。さらに、a"はそのレコードが大きい方の中央値であるかどうか、d"はそのレコードが小さい方の中央値であるかどうかを表す。

[0065] 第五に、フラグa", d"のシェア{a"}, {d"}$\in\{B\}^{m}$を任意の環F上の秘密分散によるシェア[a"], [d"]$\in[F]^{m}$に変換する。

[0066] 第六に、バリュー属性vのシェア[v]とフラグa", d"のシェア{a"}, {d"}とを用いて、[v_a]:=[va"], [v_d]:=[vd"]を計算し、復元するとベクトルv_a, $v_d\in F^{m}$となるシェア[v_a], [v_d]$\in[F]^{m}$を生成する。

[0067] 第七に、フラグa", d"のシェア{a"}, {d"}を用いて、復元するとフラグa", d"の否定¬a", ¬d"となるシェア{¬a"}, {¬d"}$\in\{B\}^{m}$を生成する。次に、フラグa", d"の否定¬a", ¬d"のシェア{¬a"}, {¬d"}を用いて、復元するとフラグa", d"の否定¬a", ¬d"をソートする置換σ_a, σ_dとなるシェア$\{\{\sigma_a\}\}$, $\{\{\sigma_d\}\}\in\{\{S_m\}\}$を生成する。

[0068] 最後に、ベクトルv_a, v_dのシェア[v_a], [v_d]と置換σ_a, σ_dのシェア$\{\{\sigma_a\}\}$,

{{σ_d}}とを用いて、[x]:=[$\sigma_a(v_a)+\sigma_d(v_d)$]を計算し、復元すると各グループの中央値を表すベクトルxとなるシェア[x]∈[F]^{m}を生成する。

[0069] ＜変形例＞

上記の実施形態では、各秘密計算装置１$_n$が、ソート済みキー属性k'$_0$，…，k'$_{nk-1}$のシェア[k'$_0$]，…，[k'$_{nk-1}$]、ソート済みバリュー属性v'$_0$，…，v'$_{na-1}$のシェア[v'$_0$]，…，[v'$_{na-1}$]、フラグeのシェア{e}、および置換σのシェア{{σ}}の少なくとも１つを出力するように構成する例を説明したが、後に続いて計算するgroup-by演算の種類によって、備えるべき処理部を選択して構成してもよい。例えば、group-byカウントやgroup-by中央値は、フラグeのシェア{e}および置換σのシェア{{σ}}を必要とするgroup-by演算である。group-by総和やgroup-by最大値／最小値は、ソート済みバリュー属性v'$_0$，…，v'$_{na-1}$のシェア[v'$_0$]，…，[v'$_{na-1}$]、フラグeのシェア{e}、および置換σのシェア{{σ}}を必要とするgroup-by演算である。グループ内の順位は、フラグeのシェア{e}および置換σのシェア{{σ}}を必要とするgroup-by演算である。すなわち、group-byカウント、group-by中央値、またはグループ内の順位を計算するが、group-by総和やgroup-by最大値／最小値を計算することがない状況であれば、秘密集約関数計算システム１００は、少なくともフラグeのシェア{e}および置換σのシェア{{σ}}が出力できればよい。このとき、各秘密計算装置１$_n$は、例えば、入力部１０、ビット分解部１１、グループソート生成部１２、ビット列ソート部１３、フラグ生成部１４、キー集約ソート生成部１５、および出力部１９を備え、重複排除部１６、キーソート部１７、およびバリューソート部１８は備えないように構成することができる。

[0070] 以上、この発明の実施の形態について説明したが、具体的な構成は、これらの実施の形態に限られるものではなく、この発明の趣旨を逸脱しない範囲で適宜設計の変更等があっても、この発明に含まれることはいうまでもない。実施の形態において説明した各種の処理は、記載の順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。

[0071] ［プログラム、記録媒体］

上記実施形態で説明した各装置における各種の処理機能をコンピュータによって実現する場合、各装置が有すべき機能の処理内容はプログラムによって記述される。そして、このプログラムをコンピュータで実行することにより、上記各装置における各種の処理機能がコンピュータ上で実現される。

[0072] 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、半導体メモリ等どのようなものでもよい。

[0073] 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。

[0074] 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、自己の記憶装置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。

[0075] 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。

請求の範囲

[請求項1] 複数の秘密計算装置を含む秘密集約関数計算システムであって、

Fは任意の環であり、mは２以上の整数であり、n_kは１以上の整数であり、$[k_0]$，…，$[k_{nk-1}]$はキー属性k_0，…，$k_{nk-1}\in F^m$を秘密分散したシェアであり、

上記秘密計算装置は、

上記シェア$[k_0]$，…，$[k_{nk-1}]$を用いて、復元すると上記キー属性k_0，…，k_{nk-1}をビット分解して結合したビット列$b:=b_0$，…，b_{m-1}となるシェア{b}から、復元すると上記ビット列bを昇順に安定ソートする置換σ_0となるシェア$\{\{\sigma_0\}\}$を生成するグループソート生成部と、

上記シェア{b}と上記シェア$\{\{\sigma_0\}\}$とを用いて、復元すると上記ビット列bを上記置換σ_0でソートしたソート済みビット列$b':=b'_0$，…，b'_{m-1}となるシェア{b'}を生成するビット列ソート部と、

上記シェア{b'}を用いて、0以上m-2以下の各整数iについて$\{e_i\}:=\{b'_i\neq b'_{i+1}\}$を設定し、かつ、$\{e_{m-1}\}:=\{1\}$を設定して、復元するとフラグ$e:=e_0$，…，$e_{m-1}$となるシェア{e}を生成するフラグ生成部と、

上記シェア{e}を用いて、復元すると上記フラグeの否定¬eを昇順に安定ソートする置換σとなるシェア$\{\{\sigma\}\}$を生成するキー集約ソート生成部と、

を含む秘密集約関数計算システム。

[請求項2] 請求項１に記載の秘密集約関数計算システムであって、

n_aは１以上の整数であり、$[v_0]$，…，$[v_{na-1}]$はバリュー属性v_0，…，$v_{na-1}\in F^m$を秘密分散したシェアであり、

上記秘密計算装置は、

上記シェア{e}を用いて、0以上m-1以下の各整数iおよび0以上n_k-1以下の各整数jについて、$\{e_i\}=\{1\}$ならば$[k''_{j,i}]$に$[k_{j,i}]$を設定し、$\{e_i\}\neq\{1\}$ならば$[k''_{j,i}]$に所定の固定値を設定して、復元すると重複排除済みキー属性k''_0，…，k''_{nk-1}となるシェア$[k''_0]$，…，$[k''_{nk-1}]$を生成す

る重複排除部と、

上記シェア$[k''_0]$, …, $[k''_{nk-1}]$と上記シェア$\{\{\sigma_0\}\}$と上記シェア$\{\{\sigma\}\}$とを用いて、

復元すると上記重複排除済みキー属性k''_0, …, k''_{nk-1}を上記置換σ_0と上記置換σとで順にソートしたソート済みキー属性k'_0, …, k'_{nk-1}となるシェア$[k'_0]$, …, $[k'_{nk-1}]$を生成するキーソート部と、

上記シェア$[v_0]$, …, $[v_{na-1}]$と上記シェア$\{\{\sigma_0\}\}$とを用いて、復元すると上記バリュー属性v_0, …, v_{na-1}を上記置換σ_0でソートしたソート済みバリュー属性v'_0, …, v'_{na-1}となるシェア$[v'_0]$, …, $[v'_{na-1}]$を生成するバリューソート部と、

を含む秘密集約関数計算システム。

[請求項3] Fは任意の環であり、mは２以上の整数であり、n_kは１以上の整数であり、$[k_0]$, …, $[k_{nk-1}]$はキー属性k_0, …, $k_{nk-1}\in F^m$を秘密分散したシェアであり、

上記シェア$[k_0]$, …, $[k_{nk-1}]$を用いて、復元すると上記キー属性k_0, …, k_{nk-1}をビット分解して結合したビット列$b:=b_0$, …, b_{m-1}となるシェア$\{b\}$から、復元すると上記ビット列bを昇順に安定ソートする置換σ_0となるシェア$\{\{\sigma_0\}\}$を生成するグループソート生成部と、

上記シェア$\{b\}$と上記シェア$\{\{\sigma_0\}\}$とを用いて、復元すると上記ビット列bを上記置換σ_0でソートしたソート済みビット列$b':=b'_0$, …, b'_{m-1}となるシェア$\{b'\}$を生成するビット列ソート部と、

上記シェア$\{b'\}$を用いて、0以上m-2以下の各整数iについて$\{e_i\}:=\{b'_i\neq b'_{i+1}\}$を設定し、かつ、$\{e_{m-1}\}:=\{1\}$を設定して、復元するとフラグ$e:=e_0$, …, e_{m-1}となるシェア$\{e\}$を生成するフラグ生成部と、

上記シェア$\{e\}$を用いて、復元すると上記フラグeの否定$\neg e$を昇順に安定ソートする置換σとなるシェア$\{\{\sigma\}\}$を生成するキー集約ソート生成部と、

を含む秘密計算装置。

[請求項4] 複数の秘密計算装置を含む秘密集約関数計算システムが実行する秘密集約関数計算方法であって、

Fは任意の環であり、mは２以上の整数であり、n_kは１以上の整数であり、$[k_0]$，…，$[k_{nk-1}]$はキー属性k_0，…，$k_{nk-1}\in F^m$を秘密分散したシェアであり、

上記秘密計算装置のグループソート生成部が、上記シェア$[k_0]$，…，$[k_{nk-1}]$を用いて、復元すると上記キー属性k_0，…，k_{nk-1}をビット分解して結合したビット列$b:=b_0$，…，b_{m-1}となるシェア$\{b\}$から、復元すると上記ビット列bを昇順に安定ソートする置換σ_0となるシェア$\{\{\sigma_0\}\}$を生成し、

上記秘密計算装置のビット列ソート部が、上記シェア$\{b\}$と上記シェア$\{\{\sigma_0\}\}$とを用いて、復元すると上記ビット列bを上記置換σ_0でソートしたソート済みビット列$b':=b'_0$，…，b'_{m-1}となるシェア$\{b'\}$を生成し、

上記秘密計算装置のフラグ生成部が、上記シェア$\{b'\}$を用いて、0以上m-2以下の各整数iについて$\{e_i\}:=\{b'_i\neq b'_{i+1}\}$を設定し、かつ、$\{e_{m-1}\}:=\{1\}$を設定して、復元するとフラグ$e:=e_0$，…，$e_{m-1}$となるシェア$\{e\}$を生成し、

上記秘密計算装置のキー集約ソート生成部が、上記シェア$\{e\}$を用いて、復元すると上記フラグeの否定$\neg e$を昇順に安定ソートする置換σとなるシェア$\{\{\sigma\}\}$を生成する、

秘密集約関数計算方法。

[請求項5] 請求項３に記載の秘密計算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。

[図1]

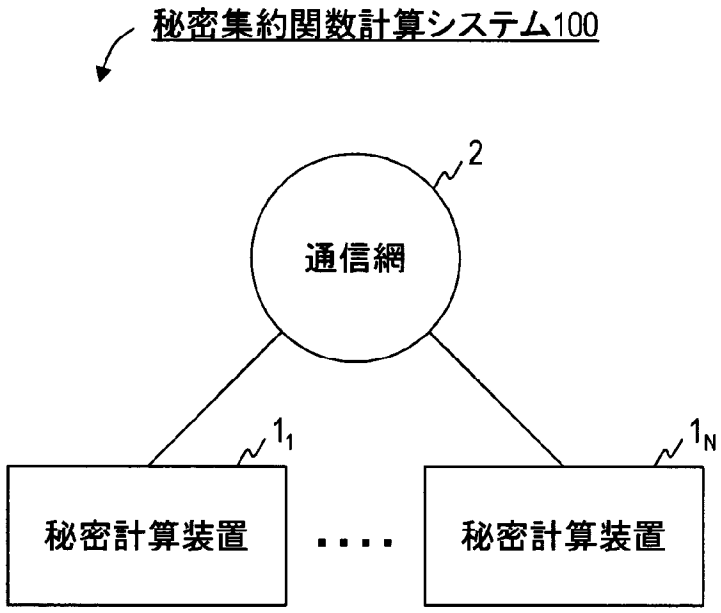


図1

[図2]

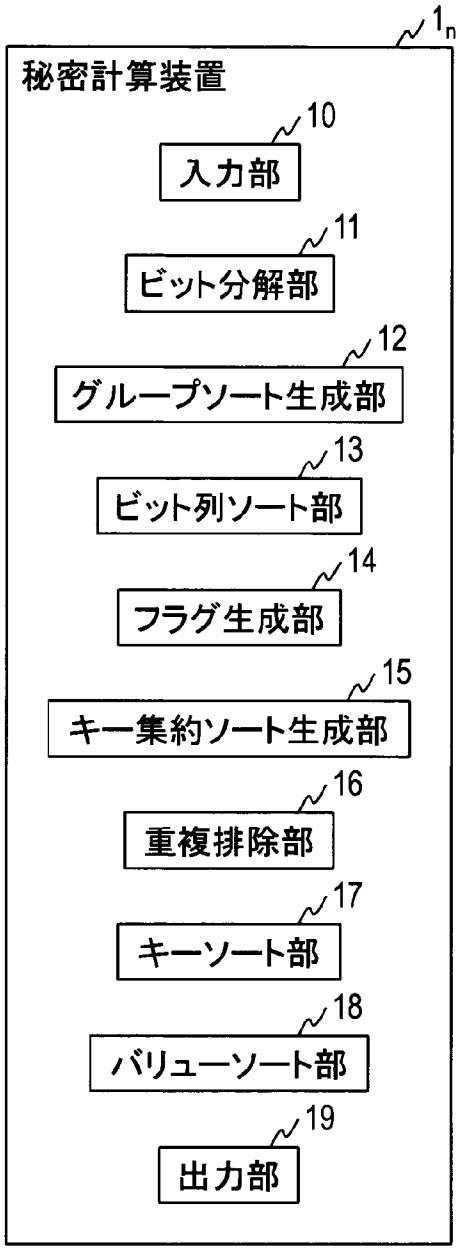

1n
秘密計算装置
10
入力部
11
ビット分解部
12
グループソート生成部
13
ビット列ソート部
14
フラグ生成部
15
キー集約ソート生成部
16
重複排除部
17
キーソート部
18
バリューソート部
19
出力部

図2

[図3]

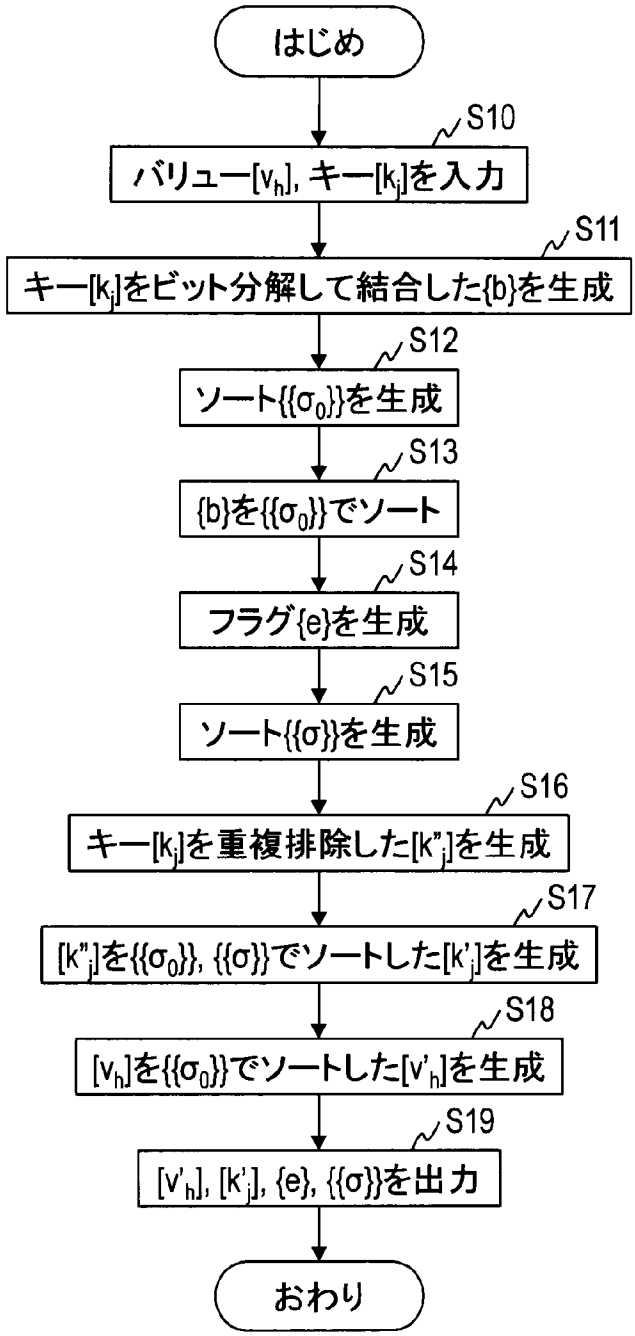

はじめ
S10
バリュー[v_h], キー[k_j]を入力
S11
キー[k_j]をビット分解して結合した{b}を生成
S12
ソート{{σ_0}}を生成
S13
{b}を{{σ_0}}でソート
S14
フラグ{e}を生成
S15
ソート{{σ}}を生成
S16
キー[k_j]を重複排除した[k''_j]を生成
S17
[k''_j]を{{σ_0}}, {{σ}}でソートした[k'_j]を生成
S18
[v_h]を{{σ_0}}でソートした[v'_h]を生成
S19
[v'_h], [k'_j], {e}, {{σ}}を出力
おわり

図3

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2019/019093

A. CLASSIFICATION OF SUBJECT MATTER
Int.Cl. G09C1/00(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
Int.Cl. G09C1/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Published examined utility model applications of Japan	1922-1996
Published unexamined utility model applications of Japan	1971-2019
Registered utility model specifications of Japan	1996-2019
Published registered utility model applications of Japan	1994-2019

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 2012-154968 A (NIPPON TELEGRAPH AND TELEPHONE CORP.) 16 August 2012, paragraphs [0013]-[0020] (Family: none)	1-5
A	WO 2015/107951 A1 (NIPPON TELEGRAPH AND TELEPHONE CORP.) 23 July 2015, paragraphs [0002]-[0009], [0017]-[0035] & US 2016/0321958 A1, paragraphs [0002]-[0009], [0025]-[0059] & EP 3096309 A1 & CN 105900164 A	1-5

☒ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
30 July 2019 (30.07.2019)	13 August 2019 (13.08.2019)
Name and mailing address of the ISA/ Japan Patent Office 3-4-3, Kasumigaseki, Chiyoda-ku, Tokyo 100-8915, Japan	Authorized officer Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2019/019093

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	桐淵直人，他，プログラマブルな秘密計算ライブラリ MEVAL3，2018年暗号と情報セキュリティシンポジウム(SCIS2018)，2A1-3，23 January 2018, pp. 1-8, in particular, table 1 (p. 5), non-official translation (KIRIBUCHI, Naoto et al., "Programmable secure computation library MEVAL3", The 2018 Symposium on Cryptography and Information Security(SCIS2018))	1-5
A	JP 5-81337 A (TOSHIBA CORP.) 02 April 1993, paragraphs [0051]-[0057] (Family: none)	1-5

国際調査報告

国際出願番号　ＰＣＴ／ＪＰ２０１９／０１９０９３

Ａ．　発明の属する分野の分類（国際特許分類（ＩＰＣ））

Int.Cl.　G09C1/00(2006.01)i

Ｂ．　調査を行った分野

調査を行った最小限資料（国際特許分類（ＩＰＣ））

Int.Cl.　G09C1/00

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	１９２２－１９９６年
日本国公開実用新案公報	１９７１－２０１９年
日本国実用新案登録公報	１９９６－２０１９年
日本国登録実用新案公報	１９９４－２０１９年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

Ｃ．　関連すると認められる文献

引用文献のカテゴリー＊	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求項の番号
A	JP 2012-154968 A（日本電信電話株式会社）2012.08.16, 段落 0013-0020（ファミリーなし）	1-5
A	WO 2015/107951 A1（日本電信電話株式会社）2015.07.23, 段落 0002-0009, 0017-0035 & US 2016/0321958 A1, paras. 0002-0009, 0025-0059 & EP 3096309 A1 & CN 105900164 A	1-5

☑　Ｃ欄の続きにも文献が列挙されている。　　☐　パテントファミリーに関する別紙を参照。

＊　引用文献のカテゴリー
「Ａ」特に関連のある文献ではなく、一般的技術水準を示すもの
「Ｅ」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
「Ｌ」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）
「Ｏ」口頭による開示、使用、展示等に言及する文献
「Ｐ」国際出願日前で、かつ優先権の主張の基礎となる出願の日の後に公表された文献
「Ｔ」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
「Ｘ」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
「Ｙ」特に関連のある文献であって、当該文献と他の１以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
「＆」同一パテントファミリー文献

国際調査を完了した日	国際調査報告の発送日
３０．０７．２０１９	１３．０８．２０１９

国際調査機関の名称及びあて先
日本国特許庁（ＩＳＡ／ＪＰ）
郵便番号１００－８９１５
東京都千代田区霞が関三丁目４番３号

特許庁審査官（権限のある職員）　５Ｓ　４５３８
金沢　史明
電話番号　０３－３５８１－１１０１　内線　３５４６

C（続き）．　関連すると認められる文献

引用文献のカテゴリー＊	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求項の番号
A	桐淵 直人，他，プログラマブルな秘密計算ライブラリ MEVAL3，2018 年暗号と情報セキュリティシンポジウム(SCIS2018)，2A1-3，2018.01.23，pp.1-8，特に表 1(p.5)	1-5
A	JP 5-81337 A（株式会社東芝）1993.04.02，段落 0051-0057（ファミリーなし）	1-5