(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7582477号
(P7582477)

(45)発行日　令和6年11月13日(2024.11.13)

(24)登録日　令和6年11月5日(2024.11.5)

(51)国際特許分類

G06F 21/71 (2013.01)	G06F	21/71	
G06F 16/2458(2019.01)	G06F	16/2458	
G06F 21/60 (2013.01)	G06F	21/60	320
G09C 1/00 (2006.01)	G09C	1/00	650Z

請求項の数　6　（全15頁）

(21)出願番号　特願2023-532977(P2023-532977)
(86)(22)出願日　令和3年7月8日(2021.7.8)
(86)国際出願番号　PCT/JP2021/025769
(87)国際公開番号　WO2023/281693
(87)国際公開日　令和5年1月12日(2023.1.12)
審査請求日　令和5年12月14日(2023.12.14)

(73)特許権者　000004226
日本電信電話株式会社
東京都千代田区大手町一丁目５番１号
(74)代理人　100121706
弁理士　中尾 直樹
(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　菊池 亮
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
(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

最終頁に続く

(54)【発明の名称】　秘密計算システム、装置、方法及びプログラム

(57)【特許請求の範囲】

【請求項１】

複数の秘密計算装置を含む秘密計算システムであって、

mはレコード数であり１以上の整数であり、$\vec{k}=(k_1,\cdots,k_m)$であり、$\vec{f}=(f_1,\cdots,f_m)$であり、αを任意の値又は任意ベクトルとして[α]はαの暗号文であり、暗号文のままαを用いた所定の演算が可能であり、

前記複数の秘密計算装置は、

前記ベクトル$\vec{f}$の暗号文$[\vec{f}]$及び前記ベクトル$\vec{k}$の暗号文$[\vec{k}]$を用いて、前記ベクトル$\vec{f}$の否定及び前記ベクトル$\vec{k}$を結合したベクトルをキーとして、前記ベクトル$\vec{f}$及び前記ベクトル$\vec{k}$をそれぞれソートしたベクトル$\overleftrightarrow{f}$及びベクトル$\overleftrightarrow{k}$の暗号文$[\overleftrightarrow{f}]$及び暗号文$[\overleftrightarrow{k}]$を生成する複数の第一計算部と、

前記暗号文$[\overleftrightarrow{f}]$及び前記暗号文$[\overleftrightarrow{k}]$を用いて、i=1,…,m-1として、$f_i=1$かつ$k_i\neq k_{i+1}$又は$f_i=1$かつ$f_{i+1}=0$のとき$e_i=0$、それ以外のとき$e_i=1$となり、$f_m=1$のとき$e_m=0$であり、それ以外のとき$e_m=1$であるようなe_mの暗号文$[e_m]$を生成することで、e_i(i=1,…,m)を要素とするベクトル$\overleftrightarrow{e}$の暗号文$[\overleftrightarrow{e}]$を生成する複数の第二計算部と、

前記mを少なくとも用いて、前記mからフラグが0のレコード数を減じたものをm’として、m’の暗号文[m’]を生成する複数の第三計算部と、

前記暗号文$[\overleftrightarrow{e}]$を用いて、i=1,…,mとして、前記ベクトル$\vec{e}$の要素$e_i=0$である場合には値がiであり、前記ベクトル$\vec{e}$の要素$e_i=1$である場合には値がm’であるx_iの暗号文$[x_i]$を生成することで、x_i(i=1,…,m)を要素とするベクトル$\vec{x}$の暗号文$[\vec{x}]$を生成する複数

の第四計算部と、
前記暗号文[e⇀]、前記暗号文[x→]及び前記暗号文[k⇀]を用いて、前記ベクトルe⇀をキーとして、前記ベクトルx→、前記ベクトルk⇀及び前記ベクトルe⇀をそれぞれソートしたベクトルx⇀、前記ベクトルk'⇀及び前記ベクトルe'⇀の暗号文[x⇀],暗号文[k'⇀]及び暗号文[e'⇀]を求める複数の第五計算部と、
前記暗号文[x⇀]を用いて、前記ベクトルx⇀の要素x_1の暗号文[x_1]を暗号文[c_1]とし、i=2,…,mとして、前記ベクトルx⇀の要素x_iから要素x_{i-1}を減算した値であるc_iの暗号文[c_i]を生成することで、c_i(i=1,…,m)を要素とするベクトルc→の暗号文[c→]を生成する複数の第六計算部と、
前記暗号文[e'⇀]を用いて、前記ベクトルe'⇀の各要素を1から減算した値により構成されるベクトルe'→の暗号文[e'→]を計算する複数の第七計算部と、
を含む秘密計算システム。
【請求項２】
請求項１の秘密計算システムであって、
前記複数の秘密計算装置は、前記暗号文[k'⇀]、前記暗号文[c→]及び前記暗号文[e'→]'を出力する複数の出力部を更に含む、
秘密計算システム。
【請求項３】
請求項１の秘密計算システムであって、
前記複数の秘密計算装置は、前記暗号文[k'⇀]、前記暗号文[c→]及び前記暗号文[e'→]'を用いて、前記暗号文[k'⇀]及び前記暗号文[c→]から、ベクトルe'→の要素e_i'''のうちダミーレコードを示す要素に対応する要素を削除したものを出力する複数の出力部を更に含む、
秘密計算システム。
【請求項４】
請求項１から３の何れかの秘密計算システムの秘密計算装置。
【請求項５】
mはレコード数であり１以上の整数であり、k→はキーのベクトルk→=(k_1,…,k_m)であり、f→はフラグのベクトルf→=(f_1,…,f_m)であり、αを任意の値又は任意ベクトルとして[α]はαの暗号文であり、暗号文のままαを用いた所定の演算が可能であり、
複数の第一計算部が、前記ベクトルf→の暗号文[f→]及び前記ベクトルk→の暗号文[k→]を用いて、前記ベクトルf→の否定及び前記ベクトルk→を結合したベクトルをキーとして、前記ベクトルf→及び前記ベクトルk→をそれぞれソートしたベクトルf⇀及びベクトルk'→の暗号文[f⇀]及び暗号文[k⇀]を生成する第一計算ステップと、
複数の第二計算部が、前記暗号文[f⇀]及び前記暗号文[k⇀]を用いて、i=1,…,m-1として、f_i=1かつk_i≠k_{i+1}又はf_i=1かつf_{i+1}=0のときe_i=0、それ以外のときe_i=1となり、f_m=1のときe_m=0であり、それ以外のときe_m=1であるようなe_mの暗号文[e_m]を生成することで、e_i(i=1,…,m)を要素とするベクトルe⇀の暗号文[e⇀]を生成する第二計算ステップと、
複数の第三計算部が、前記mを少なくとも用いて、前記mからフラグが0のレコード数を減じたものをm'として、m'の暗号文[m']を生成する第三計算ステップと、
複数の第四計算部が、前記暗号文[e⇀]を用いて、i=1,…,mとして、前記ベクトルe→の要素e_i=0である場合には値がiであり、前記ベクトルe→の要素e_i=1である場合には値がm'であるx_iの暗号文[x_i]を生成することで、x_i(i=1,…,m)を要素とするベクトルx→の暗号文[x→]を生成する第四計算ステップと、
複数の第五計算部が、前記暗号文[e⇀]、前記暗号文[x→]及び前記暗号文[k⇀]を用いて、前記ベクトルe⇀をキーとして、前記ベクトルx→、前記ベクトルk⇀及び前記ベクトルe⇀をそれぞれソートしたベクトルx⇀、前記ベクトルk'⇀及び前記ベクトルe'⇀の暗号文[x⇀],暗号文[k'⇀]及び暗号文[e'⇀]を求める第五計算ステップと、
複数の第六計算部が、前記暗号文[x⇀]を用いて、前記ベクトルx⇀の要素x_1の暗号文[

x_1]を暗号文[c_1]とし、i=2,…,mとして、前記ベクトル$\vec{x}$の要素x_iから要素x_{i-1}を減算した値であるc_iの暗号文[c_i]を生成することで、c_i(i=1,…,m)を要素とするベクトル$\vec{c}$の暗号文[$\vec{c}$]を生成する第六計算ステップと、

複数の第七計算部が、前記暗号文[$\vec{e'}$]を用いて、前記ベクトル$\vec{e'}$の各要素を1から減算した値により構成されるベクトル$\vec{e'}$の暗号文[$\vec{e'}$]を計算する第七計算ステップと、

を含む秘密計算方法。

【請求項６】

請求項５の秘密計算方法の各ステップ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、データを秘匿したままデータベース演算を行う技術に関する。

【背景技術】

【０００２】

データを安全に扱うために、暗号化したまま分析する秘密計算という技術が研究されている。その中でも、暗号化したまま条件に合うデータの取り出しや集計値などを効率的に算出するために、暗号化データベース処理が考えられている。

【０００３】

DB処理の一種であるgroup by演算とはグループ化処理であり、テーブルを入力とし、指定したカラムの値ごとにグループ化し、場合によってそのグループごとの統計値を計算してテーブル形式で出力するものである。

【０００４】

group by演算を暗号化したまま行う方法は非特許文献１にて提案されている。ここで考えられている入出力は、通常のテーブルを、各要素ごとに暗号化したテーブルであった。

【０００５】

一方、暗号化したままデータベース処理を行う場合、その入出力は、通常のテーブルとは異なり、あるレコードが本来の出力か否かを示すフラグが付与されていることが考えられる。

【０００６】

$\vec{k}$をキーのベクトルとし、$\vec{v}$をバリューのベクトルとし、$\vec{f}$をフラグのベクトルとし、[・]を暗号化したデータとして、図７(a)に通常の暗号化されていないテーブルの例を、図７(b)に非特許文献１における暗号化されたテーブルの例を、図７(c)にフラグが追加されているテーブルの例を示す。

【０００７】

図７において、“？”は何かしらの値が入っていることを示す。フラグが０の場合、そのレコードのバリューは無視されるため、この“？”のバリューの値は任意である。

【先行技術文献】

【非特許文献】

【０００８】

【文献】菊池亮, 濱田浩気, 五十嵐大, 高橋元, 高橋克巳, 「横断的動線分析を秘密計算でやってみよう」, In SCIS,2020.

【発明の概要】

【発明が解決しようとする課題】

【０００９】

図７(c)に例示するフラグが追加されているテーブルが入力される場合、非特許文献１で提案されたアルゴリズムは機能しない。なぜなら、入力の形式が異なることに加えて、今までは全てのレコードが意味のある値であることを想定していたため、例えば、使わないレコードをスキップして処理を行うことができておらず、無視すべき“？”の値が最終結果

に影響してしまい、本来の結果を得ることができないためである。

【００１０】

本発明は、フラグが追加されているテーブルに対してgroup by count演算を行う秘密計算システム、装置、方法及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】

【００１１】

この発明の一態様による秘密計算システムは、複数の秘密計算装置を含む秘密計算システムであって、mはレコード数であり１以上の整数であり、$k^{\rightarrow}$は$k^{\rightarrow}=(k_1,\cdots,k_m)$であり、$f^{\rightarrow}$は$f^{\rightarrow}=(f_1,\cdots,f_m)$であり、αを任意の値又は任意ベクトルとして$[\alpha]$はαの暗号文であり、暗号文のままαを用いた所定の演算が可能であり、複数の秘密計算装置は、ベクトル$f^{\rightarrow}$の暗号文$[f^{\rightarrow}]$及びベクトル$k^{\rightarrow}$の暗号文$[k^{\rightarrow}]$を用いて、ベクトル$f^{\rightarrow}$の否定及びベクトル$k^{\rightarrow}$を結合したベクトルをキーとして、ベクトル$f^{\rightarrow}$及びベクトル$k^{\rightarrow}$をそれぞれソートしたベクトル$f^{\rightarrow\prime}$及びベクトル$k^{\rightarrow\prime}$の暗号文$[f^{\rightarrow\prime}]$及び暗号文$[k^{\rightarrow\prime}]$を生成する複数の第一計算部と、暗号文$[f^{\rightarrow\prime}]$及び暗号文$[k^{\rightarrow\prime}]$を用いて、i=1,…,m-1として、$f_i=1$かつ$k_i\neq k_{i+1}$又は$f_i=1$かつ$f_{i+1}=0$のとき$e_i=0$、それ以外のとき$e_i=1$となり、$f_m=1$のとき$e_m=0$であり、それ以外のとき$e_m=1$であるようなe_mの暗号文$[e_m]$を生成することで、$e_i(i=1,\cdots,m)$を要素とするベクトル$e^{\rightarrow\prime}$の暗号文$[e^{\rightarrow\prime}]$を生成する複数の第二計算部と、mを少なくとも用いて、mからフラグが0のレコード数を減じたものをm’として、m’の暗号文[m’]を生成する複数の第三計算部と、暗号文[]を用いて、i=1,…,mとして、ベクトル$e^{\rightarrow}$の要素$e_i=0$である場合には値がiであり、ベクトル$e^{\rightarrow}$の要素$e_i=1$である場合には値がm’であるiの暗号文$[x_i]$を生成することで、$x_i(i=1,\cdots,m)$を要素とするベクトル$x^{\rightarrow}$の暗号文$[x^{\rightarrow}]$を生成する複数の第四計算部と、暗号文$[e^{\rightarrow\prime}]$、暗号文$[x^{\rightarrow}]$及び暗号文$[k^{\rightarrow\prime}]$を用いて、ベクトル$e^{\rightarrow\prime}$をキーとして、ベクトル$x^{\rightarrow}$、ベクトル$k^{\rightarrow\prime}$及びベクトル$e^{\rightarrow\prime}$をそれぞれソートしたベクトル$x^{\rightarrow\prime}$、ベクトル$k^{\rightarrow\prime}$及びベクトル$e^{\rightarrow\prime}$の暗号文$[x^{\rightarrow}]$,暗号文$[k^{\rightarrow\prime}]$及び暗号文$[e^{\rightarrow\prime}]$を求める複数の第五計算部と、暗号文$[x^{\rightarrow\prime}]$を用いて、ベクトル$x^{\rightarrow\prime}$の要素x_1の暗号文$[x_1]$を暗号文$[c_1]$とし、i=2,…,mとして、ベクトル$x^{\rightarrow\prime}$の要素x_iから要素x_{i-1}を減算した値であるc_iの暗号文$[c_i]$を生成することで、$c_i(i=1,\cdots,m)$を要素とするベクトル$c^{\rightarrow}$の暗号文$[c^{\rightarrow}]$を生成する複数の第六計算部と、暗号文$[e^{\rightarrow\prime}]$を用いて、ベクトル$e^{\rightarrow\prime}$の各要素を1から減算した値により構成されるベクトル$e^{\rightarrow}$の暗号文$[e^{\rightarrow}]$を計算する複数の第七計算部と、を備えている。

【発明の効果】

【００１２】

フラグが追加されているテーブルに対してgroup by count演算を行うことができる。

【図面の簡単な説明】

【００１３】

【図１】図１は、秘密計算システムの機能構成の例を示す図である。

【図２】図２は、秘密計算装置の機能構成の例を示す図である。

【図３】図３は、アルゴリズムの例を示す図である。

【図４】図４は、アルゴリズムの例を示す図である。

【図５】図５は、入力の例と出力の例を説明するための図である。

【図６】図６は、秘密計算方法の処理手続きの例を示す図である。

【図７】図７は、背景技術を説明するための図である。

【図８】図８は、コンピュータの機能構成例を示す図である。

【発明を実施するための形態】

【００１４】

以下、本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。

【００１５】

なお、文中で使用する記号「$^{\rightarrow}$」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、当該文字の直後に記載する。

【００１６】
暗号化されたデータを[x]と書き、ベクトルを$\vec{x}=(x_1,\cdots,x_n)$と書き、$[\vec{x}]=([x_1],\cdots,[x_n])$とする。
【００１７】
暗号化は、秘密分散（例えば参考文献１）や準同型暗号（例えば参考文献２）など、暗号化したまま下記の演算が可能な方法で行われるとする。暗号文[・]の・がビット値である場合には、暗号文[・]を暗号文[[・]]と記述することがある。また、置換には〈π〉という記述を用いることがある。・は、任意の値、ベクトルである。格納する値に対して異なる暗号化を用いてもよい。すなわち、これらの暗号化は全て同じものであっても、そうでなくてもよい。
【００１８】
すなわち、αを任意の値又は任意ベクトルとして[α]はαの暗号文であり、βを任意の置換として〈β〉はβの暗号文である。
【００１９】
〈参考文献１〉Dai Ikarashi, Ryo Kikuchi, Koki Hamada, and Koji Chida. Actively private and correct MPC scheme in t < n/2 from passively secure schemes with small overhead. IACR Cryptology ePrint Archive, Vol. 2014, p. 304, 2014.
〈参考文献２〉Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. Fully homomorphic encryption without bootstrapping. Electronic Colloquium on Computational Complexity (ECCC), Vol. 18, p. 111, 2011.
加減算、定数倍に関して、秘密分散と準同型暗号はサポートされているとする。すなわち、c[a]±[b] ±d=[ca±b±d]の関係が成立しているとする。
【００２０】
乗算は、秘密分散であれば参考文献１に記載された方法で、準同型暗号であれば準同型演算で計算可能である。乗算を、[c]←Mult([a],[b])と記述する。ここで、c=abである。
【００２１】
安定ソートは、入力$[\vec{x}]=([x_1],\cdots,[x_n])$を、$i\in\{1,\cdots,n-1\}$について、$x_i' \leqq x_{i+1}'$であるような$[\vec{x}']=([x_1'],\cdots,[x_n'])$に並び替える処理である。ただし、$x_i'=x_{i+1}'$であるとき元々の$\vec{x}$の並び順が優先されるものとする。
【００２２】
安定ソートは、より具体的には２個のアルゴリズム(GENPERM,SORT)からなる。
【００２３】
GENPERMは、$\vec{x}$を並び替える置換πを暗号化したものを出力する関数である。GENPERMは、〈π〉←GENPERM($[\vec{x}]$)と記述される。
【００２４】
SORTは、πを$\vec{x}$に適用し並び替えたもの$\vec{x}'$を暗号化したまま計算する関数である。SORTは、例えば$[\vec{x}']$←SORT(〈π〉,$[\vec{x}]$)と記述される。
【００２５】
記載の簡略化のため、複数のベクトルのそれぞれを同じ置換でソートする際には、SORTは、例えば($[\vec{x}']$,$[\vec{y}']$)←SORT(〈π〉,($[\vec{x}]$,$[\vec{y}]$))と記述される。
【００２６】
SORTの自明な構成方法は、ソーティングネットワークを用いる方法である。また、秘密分散であれば、参考文献３に記載された方法により効率よくSORTを行うことができる。
【００２７】
〈参考文献３〉Koji Chida, Koki Hamada, Dai Ikarashi, Ryo Kikuchi, Naoto Kiribuchi, and Benny Pinkas. An efficient secure three-party sorting protocol with an honest majority. IACR Cryptology ePrint Archive, Vol. 2019, p. 695, 2019.
等号判定EQは、[x],[y]を入力として、x=yならば1、x≠yならば0となるようなeの暗号文[e]を出力する関数である。EQは、例えば[e]←EQ([x],[y])と記述される。ここで、eはx=yならば1、x≠yならば0である。

【００２８】
複数の要素の等号判定を行う場合、EQは、例えば[e]←EQ([a],[b]),([c],[d]))と記述される。ここで、eは、a=cかつb=dならば1、そうでないならば0である。
【００２９】
一般にビット表現でデータが暗号化されているならば、[x-y]の各ビットが0かどうかを回路計算することにより、等号判定を行うことができる。回路計算は、加減算と乗算で計算可能である。
【００３０】
整数表現で暗号化されている場合であれば、ビット分解（例えば、参考文献４参照。）を用いてビット表現に変更して同様に回路計算することにより、等号判定を行うことができる。
【００３１】
〈参考文献４〉Ryo Kikuchi, Dai Ikarashi, Takahiro Matsuda, Koki Hamada, and Koji Chida. Efficient bitdecomposition and modulus-conversion protocols with an honest majority. In ACISP 2018, pp.64-82, 2018.
他にもmod p上で暗号化されているのであれば、$[(x-y)^{p-1}]$を乗算を使って計算しても、等号判定を行うことができる。
【００３２】
IFTHENは、フラグ[f]（ただしf∈{0,1}）と[x],[y]を入力として、f=1ならば[x]を、f=0ならば[y]を出力する関数である。IFTHENは、例えば[e]←IFTHEN([f]:[x],[y])と記述される。ここで、eは、f=1ならばxであり、f=0ならばyである。
【００３３】
IFTHENは、Mult([f],[x])+Mult([1-f],[y])などで例えば実現することができる。
【００３４】
MODCONVは、ビット値の暗号化[[a]]を入力として、同じ値の暗号化ではあるが暗号文の形が違う[a]を生成する関数である。言い換えれば、MODCONVは、ビット値の暗号文[[a]]を入力として、aを整数表現した値の暗号文[a]を生成する関数である。MODCONVは、例えば[a]←MODCONV([[a]])と記述される。
【００３５】
BITDECOMPは、整数値の暗号化[a]を入力として、aをビット表現した同じ値の暗号化ではあるが、暗号文の形が違う[[a]]を生成する関数である。言い換えれば、BITDECOMPは、整数値の暗号文[a]を入力として、aをビット表現した値の暗号文[[a]]を生成する関数である。BITDECOMPは、例えば$[[\vec{k}]]$←BITDECOMP($[\vec{k}]$)と記述される。ただし、$\vec{k}=(k_1,k_2,\cdots,k_L)$としたとき，$k=\Sigma_{i=1}^{L}2^{i-1}k_i$である。
【００３６】
秘密計算システム、装置、方法及びプログラムの処理の対象となるテーブルのレコード数はmである。このテーブルは、キーのベクトル$\vec{k}$の暗号文$[\vec{k}]$、フラグのベクトル$\vec{f}$の暗号文$[\vec{f}]$から少なくとも構成されているとする。暗号文$[\vec{f}]$の要素はビットの暗号文であるとする。もし、ビット出なかった場合はビット分解プロトコルでビットに変換する。
【００３７】
秘密計算システム、装置、方法及びプログラムの処理の対象となるテーブルを、図５(a)に例示する。group by count演算は、バリューを使わないため、図５(a)に例示するテーブルでは、キー$\vec{k}$の暗号文$[\vec{k}]$とフラグ$\vec{f}$の暗号文$[\vec{f}]$のみが記載されている。
【００３８】
秘密計算システム、装置、方法及びプログラムによるgroup by count演算により、図５(a)に示すテーブルから、例えば図５(b)に示すテーブルが得られる。$[\vec{k}']$はキーのベクトル$\vec{k}$の要素を並び替えたベクトル$\vec{k}'$の暗号文である。$[\vec{c}]$はカウント数から構成されるベクトル$\vec{c}$の暗号文である。$[[\vec{e}']]$はベクトル$\vec{k}'$に対応するフラグのベクトル$\vec{e}'$の暗号文である。
【００３９】

図５(a)では、暗号文[k’⃗]の[1]のキーの数が２個であり、暗号文[k’⃗]の[2]のキーの数が１個であり、暗号文[k’⃗]の[3]のキーの数が４個である。

【００４０】

このため、図５(b)では、暗号文[k’⃗]の[1]に対応する暗号文[c⃗]の要素が[2]となっており、暗号文[k’⃗]の[2]に対応する暗号文[c⃗]の要素が[1]となっており、暗号文[k’⃗]の[4]に対応する暗号文[c⃗]の要素が[2]となっている。

【００４１】

図１を参照して、秘密計算システム及び方法の構成例を説明する。この秘密計算システム及び方法は、いわゆるgroup by count演算を秘密計算で行うものである。

【００４２】

秘密計算システムは、N（≧2）台の秘密計算装置$1_1,\cdots,1_N$を含む。本形態では、秘密計算装置$1_1,\cdots,1_N$のそれぞれは通信網２に接続されている。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）などである。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$1_1,\cdots,1_N$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$1_1,\cdots,1_N$へオフラインで入力するように構成してもよい。

【００４３】

図２を参照して、秘密計算システム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。秘密計算システムの秘密計算装置1_nは、例えば、図２に示すように、第一計算部11_n、第二計算部12_n、第三計算部13_n、第四計算部14_n、第五計算部15_n、第六計算部16_n、第七計算部17_n、出力部18_{nn}を備えている。

【００４４】

秘密計算装置1_n（$1 \leqq n \leqq N$）の各構成部が他の秘密計算装置$1_{n'}$（n'=1, …, N、ただし$n \neq n'$）の各構成部と協調しながら後述する及び図６に例示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計算が実現される。

【００４５】

なお、各ステップの処理は、秘密計算により行われる。すなわち、秘密計算装置1_nは、暗号文を復元することなく、言い換えれば暗号文の中身を知ることなく、各ステップの処理を行う。

【００４６】

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の各構成部は、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。

【００４７】

〈第一計算部$11_1,\cdots,11_N$〉

複数の第一計算部$11_1,\cdots,11_N$には、ベクトル$f^{\rightarrow}$の暗号文$[f^{\rightarrow}]$及びベクトル$k^{\rightarrow}$の暗号文$[k^{\rightarrow}]$が入力される。

【００４８】

複数の第一計算部$11_1,\cdots,11_N$は、ベクトル$f^{\rightarrow}$の暗号文$[f^{\rightarrow}]$及びベクトル$k^{\rightarrow}$の暗号文$[k^{\rightarrow}]$を用いて、ベクトル$f^{\rightarrow}$の否定及びベクトル$k^{\rightarrow}$を結合したベクトルをキーとして、ベクトル$f^{\rightarrow}$及びベクトル$k^{\rightarrow}$をそれぞれソートしたベクトル$f^{\leftrightarrows}$及びベクトル$k^{\leftrightarrows}$の暗号文$[f^{\leftrightarrows}]$及び暗号文$[k^{\leftrightarrows}]$を生成する（ステップＳ１）。

【００４９】

この複数の第一計算部$11_1,\cdots,11_N$による処理は、図３の「１：」から「４：」の処理により例えば実現される。

【００５０】

すなわち、複数の第一計算部$11_1,\cdots,11_N$は、

1:[[$k^{\dagger\rightarrow}$]]←BITDECOMP([$k^{\rightarrow}$])

2:[[$f^{*\rightarrow}$]]←1-[[$f^{\rightarrow}$]]

3:〈π〉←GENPERM([[$f^{*\rightarrow}$]],[[$k^{\dagger\rightarrow}$]])

4:([[$k^{\rightarrow\!\!\!\!\!\rightarrow}$]],[$k^{\rightarrow\!\!\!\!\!\rightarrow}$],[[$f^{\rightarrow\!\!\!\!\!\rightarrow}$]])←SORT(〈π〉,([[$k^{\dagger\rightarrow}$]],[$k^{\rightarrow}$],[[$f^{\rightarrow}$]]))

という処理を例えば行う。

【００５１】

図３の例では、暗号文[$f^{\rightarrow\!\!\!\!\!\rightarrow}$]として暗号文[[$f^{\rightarrow\!\!\!\!\!\rightarrow}$]]が生成されている。また、図３の例では、暗号文[$k^{\rightarrow\!\!\!\!\!\rightarrow}$]として暗号文[[$k^{\rightarrow\!\!\!\!\!\rightarrow}$]]及び暗号文[$k^{\rightarrow\!\!\!\!\!\rightarrow}$]が生成されている。

【００５２】

なお、GENPERM([[$f^{*\rightarrow}$]],[[$k^{\dagger\rightarrow}$]])は、暗号文[[$f^{*\rightarrow}$]]及び暗号文[[$k^{\dagger\rightarrow}$]]を用いて、ベクトル$f^{*\rightarrow}$及びベクトル$k^{\dagger\rightarrow}$を要素ごとに結合したベクトルを安定ソートする置換πの暗号文〈π〉を生成する処理を意味する。

【００５３】

〈第二計算部$12_1,\cdots,12_N$〉

複数の第二計算部$12_1,\cdots,12_N$には、暗号文[$f^{\rightarrow\!\!\!\!\!\rightarrow}$]及び暗号文[$k^{\rightarrow\!\!\!\!\!\rightarrow}$]が入力される。

【００５４】

複数の第二計算部$12_1,\cdots,12_N$は、暗号文[$f^{\rightarrow\!\!\!\!\!\rightarrow}$]及び暗号文[$k^{\rightarrow\!\!\!\!\!\rightarrow}$]を用いて、i=1,…,m-1として、$f_i$=1かつ$k_i \neq k_{i+1}$又は$f_i$=1かつ$f_{i+1}$=0のとき$e_i$=0、それ以外のとき$e_i$=1となり、$f_m$'=1のとき$e_m$'=0であり、それ以外のとき$e_m$'=1であるような$e_m$'の暗号文[$e_m$']を生成することで、$e_i$(i=1,…,m)を要素とするベクトル$e^{\rightarrow\!\!\!\!\!\rightarrow}$の暗号文[$e^{\rightarrow\!\!\!\!\!\rightarrow}$]を生成する（ステップＳ２）。

【００５５】

この複数の第二計算部$12_1,\cdots,12_N$による処理は、図３の「５：」から「１０：」の処理により例えば実現される。

【００５６】

すなわち、複数の第二計算部$12_1,\cdots,12_N$は、

5:each $1 \leqq i \leqq m-1$ do

6: [[e_i]]←IFTHEN([[f_i]]:EQ([[k_i]],[[k_{i+1}]]),[[1]])

7: [[e_i]]←IFTHEN([[f_i]]XOR[[f_{i+1}]]:[[0]],[[e_i]])

8: [e_i]←MODCONV([[e_i]])

9:[[e_m']]=1-[[f_m]]

10:[e_m']←MODCONV([[e_m']])

という処理を例えば行う。

【００５７】

〈第三計算部$13_1,\cdots,13_N$〉

複数の第三計算部$13_1,\cdots,13_N$には、mが少なくとも入力される。

【００５８】

複数の第三計算部$13_1,\cdots,13_N$は、mを少なくとも用いて、mからフラグが0のレコード数を減じたものをm'として、m'の暗号文[m']を生成する（ステップＳ３）。

【００５９】

この複数の第三計算部$13_1,\cdots,13_N$による処理は、図３の「１１：」から「１２：」の処理により例えば実現される。

【００６０】

すなわち、複数の第三計算部$13_1,\cdots,13_N$は、

11:[f^*]←MODCONV(1-[[$f^{\rightarrow\!\!\!\!\!\rightarrow}$]])

12:[m']=m - $\sum_{i=1}^{m}$[f^*_i]

という処理を例えば行う。

【００６１】

この例では、複数の第三計算部$１３_1,\cdots,１３_N$に暗号文$[[f\ ↦]]$が更に入力され、複数の第三計算部$１３_1,\cdots,１３_N$は、この暗号文$[[f\ ↦]]$を更に用いて、暗号文$[m\ ']$を生成している。

【００６２】

なお、複数の第三計算部$１３_1,\cdots,１３_N$は、暗号文$[[f\ ↦]]$に代えて暗号文$[f\ ↦]$を用いて、上記と同様にして、暗号文$[m\ ']$を生成してもよい。

【００６３】

〈第四計算部$１４_1,\cdots,１４_N$〉

複数の第四計算部$１４_1,\cdots,１４_N$には、暗号文$[e\ ↦]$が入力される。

【００６４】

複数の第四計算部$１４_1,\cdots,１４_N$は、暗号文$[e\ ↦]$を用いて、$i=1,\cdots,m$として、ベクトル$e^{→}$の要素$e_i=0$である場合には値がiであり、ベクトル$e^{→}$の要素$e_i=1$である場合には値がm'であるpの暗号文$[x_i]$を生成することで、$x_i(i=1,\cdots,m)$を要素とするベクトル$x^{→}$の暗号文$[x^{→}]$を生成する（ステップＳ４）。

【００６５】

この複数の第四計算部$１４_1,\cdots,１４_N$による処理は、図３の「１３：」から「１４：」の処理により例えば実現される。

【００６６】

すなわち、複数の第四計算部$１４_1,\cdots,１４_N$は、

13:each $1 \leqq i \leqq m$ do

14: $[x_i] \leftarrow$IFTHEN$([e_i]:[m'],[i])$

という処理を例えば行う。

【００６７】

〈第五計算部$１５_1,\cdots,１５_N$〉

複数の第五計算部$１５_1,\cdots,１５_N$には、暗号文$[e\ ↦]$、暗号文$[x^{→}]$及び暗号文$[k\ ↦]$が入力される。

【００６８】

複数の第五計算部$１５_1,\cdots,１５_N$は、暗号文$[e\ ↦]$、暗号文$[x^{→}]$及び暗号文$[k\ ↦]$を用いて、ベクトル$e\ ↦$をキーとして、ベクトル$x^{→}$、ベクトル$k\ ↦$及びベクトル$e\ ↦$をそれぞれソートしたベクトル$x\ ↦$、ベクトル$k'^{→}$及びベクトル$e'^{→}$の暗号文$[x\ ↦]$,暗号文$[k'^{→}]$及び暗号文$[e'^{→}]$を生成する（ステップＳ５）。

【００６９】

この複数の第五計算部$１５_1,\cdots,１５_N$による処理は、図３の「１５：」から「１６：」の処理により例えば実現される。

【００７０】

すなわち、複数の第五計算部$１５_1,\cdots,１５_N$は、

15:$\langle \pi' \rangle \leftarrow$GENPERM$([[\vec{e}]])$

16:$([x\ ↦],[[e'^{→}]],[k'^{→}]) \leftarrow$SORT$(\langle \pi' \rangle,([\vec{x}],[[e\ ↦]],[k\ ↦]))$

という処理を例えば行う。

【００７１】

図３の例では、暗号文$[e\ ↦]$として暗号文$[[e\ ↦]]$が用いられている。また、図３の例では、暗号文$[e'^{→}]$として暗号文$[[e'^{→}]]$が生成されている。

【００７２】

〈第六計算部$１６_1,\cdots,１６_N$〉

複数の第六計算部$１６_1,\cdots,１６_N$には、暗号文$[x\ ↦]$が入力される。

【００７３】

複数の第六計算部$１６_1,\cdots,１６_N$は、暗号文$[x\ ↦]$を用いて、ベクトル$x\ ↦$の要素x_1の暗号文$[x_1]$を暗号文$[c_1]$とし、$i=2,\cdots,m$として、ベクトル$x\ ↦$の要素x_iから要素x_{i-1}を

減算した値であるc_iの暗号文$[c_i]$を生成することで、$c_i(i=1,\cdots,m)$を要素とするベクトル$\vec{c}$の暗号文$[\vec{c}]$を生成する（ステップＳ６）。

【００７４】

この複数の第六計算部$16_1,\cdots,16_N$による処理は、図３の「１７：」から「１９：」の処理により例えば実現される。

【００７５】

すなわち、複数の第六計算部$16_1,\cdots,16_N$は、

17:$[c_1]=[x_1]$
18:each $2 \leqq i \leqq m$ do
19: $[c_i]=[x_i]-[x_{i-1}]$

という処理を例えば行う。

【００７６】

〈第七計算部$17_1,\cdots,17_N$〉

複数の第七計算部$17_1,\cdots,17_N$には、暗号文$[e'^{\rightarrow}]$が入力される。

【００７７】

複数の第七計算部$17_1,\cdots,17_N$は、暗号文$[e'^{\rightarrow}]$を用いて、ベクトル$e'^{\rightarrow}$の各要素を1から減算した値により構成されるベクトル$e'^{\rightarrow}$の暗号文$[e'^{\rightarrow}]$を計算する（ステップＳ２）。

【００７８】

この複数の第七計算部$17_1,\cdots,17_N$による処理は、図３の「２０：」の処理により例えば実現される。

【００７９】

すなわち、複数の第七計算部$17_1,\cdots,17_N$は、

20:$[[e'^{\rightarrow}]] =1-[[e'^{\rightarrow}]]$

という処理を例えば行う。

【００８０】

図３の例では、暗号文$[e'^{\rightarrow}]$として暗号文$[[e'^{\rightarrow}]]$が用いられている。また、図３の例では、暗号文$[e'^{\rightarrow}]$として暗号文$[[e'^{\rightarrow}]]$が生成されている。

【００８１】

〈出力部$18_1,\cdots,18_N$〉

複数の出力部$18_1,\cdots,18_N$には、暗号文$[k'^{\rightarrow}]$、暗号文$[\vec{c}]$及び暗号文$[e'^{\rightarrow}]$が入力される。

【００８２】

複数の出力部$18_1,\cdots,18_N$は、暗号文$[k'^{\rightarrow}]$、暗号文$[\vec{c}]$及び暗号文$[e'^{\rightarrow}]$を出力する出力を行う（ステップＳ８）。

【００８３】

なお、複数の出力部$18_1,\cdots,18_N$は、暗号文$[k'^{\rightarrow}]$、暗号文$[\vec{c}]$及び暗号文$[e'^{\rightarrow}]$を用いて、暗号文$[k'^{\rightarrow}]$及び暗号文$[\vec{c}]$から、ベクトル$e'^{\rightarrow}$の要素e_i'’’’のうちダミーレコードを示す要素に対応する要素を削除したものを出力してもよい。

【００８４】

暗号文$[k'^{\rightarrow}]$、暗号文$[\vec{c}]$及び暗号文$[e'^{\rightarrow}]$が例えば図５（b）に示されるものである場合には、複数の出力部$18_1,\cdots,18_N$は、暗号文$[e'^{\rightarrow}]$の要素[1]に対応する、暗号文$[k'^{\rightarrow}]$及び暗号文$[\vec{c}]$の要素のみを出力してもよい。なお、図５(b)の暗号文$[[e'^{\rightarrow}]]$は、暗号文$[e'^{\rightarrow}]$に対応している。すなわち、この場合、複数の出力部$18_1,\cdots,18_N$は、暗号文$[k'^{\rightarrow}]$及び暗号文$[\vec{c}]$の３番目までの要素を出力してもよい。

【００８５】

このように、ダミーフラグの否定を最上位ビットに配置してソートすることでダミーレコードを最下位レコードとし、また、グループの境界判定ビットについて、フラグが0ならば境界にならないようなif文処理を加えることで、復号することなくgroup by countを達成することができる。

【００８６】
なお、秘密計算装置$1_1, \cdots, 1_N$は、いわゆるnull処理を行ってもよい。このnull処理は、例えば、複数の第四計算部$14_1,\cdots,14_N$により行われる、図４の「１４：」の２行目の処理により例えば実現される。
【００８７】
すなわち、複数の第四計算部$14_1,\cdots,14_N$は、
13:each $1 \leqq i \leqq m$ do
14: $[x_i] \leftarrow$IFTHEN([e ¡]:[m ’],[i])
$[k'_i] \leftarrow$IFTHEN([e ¡]:[null],[k ’$_i$])
という処理を例えば行ってもよい。これにより、例えば、図５(b)に示すような暗号文[k ’ $\vec{}$]が生成される。
【００８８】
なお、図５(b)において暗号文[[e ’ $\vec{}$]]が[[0]]となっているレコード、言い換えれば暗号文[k ’$\vec{}$]が[null]となっているレコードがダミーレコードである。
【００８９】
[変形例]
以上、本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、本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、本発明に含まれることはいうまでもない。
【００９０】
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。
【００９１】
例えば、秘密計算装置の構成部間のデータのやり取りは直接行われてもよいし、図示していない記憶部を介して行われてもよい。
【００９２】
[プログラム、記録媒体]
上述した各装置の各部の処理をコンピュータにより実現してもよく、この場合は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、このプログラムを図８に示すコンピュータ１０００の記憶部１０２０に読み込ませ、演算処理部１０１０、入力部１０３０、出力部１０４０などに動作させることにより、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。
【００９３】
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は、例えば、非一時的な記録媒体であり、具体的には、磁気記録装置、光ディスク、等である。
【００９４】
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。
【００９５】
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、自己の非一時的な記憶装置である補助記録部１０５０に格納する。そして、処理の実行時、このコンピュータは、自己の非一時的な記憶装置である補助記録部１０５０に格納されたプログラムを記憶部１０２０に読み込み、読み込んだ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

プログラムを記憶部１０２０に読み込み、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。

【００９６】

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。

【００９７】

その他、この発明の趣旨を逸脱しない範囲で適宜変更が可能であることはいうまでもない。

【図面】

【図１】

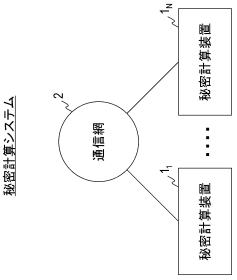

図1

【図２】

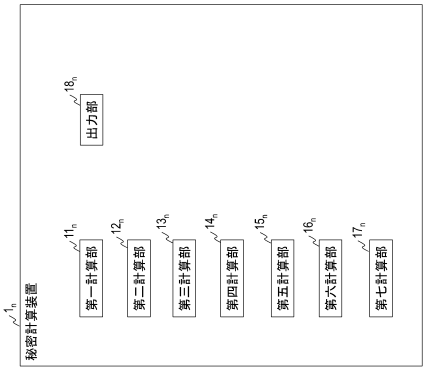

図2

【図３】

アルゴリズム(null処理なし)

```
1:[[k^t-1]]←BITDECOMP([k^-])
2:[[f^m-1]]←1-[[f^-]]
3:<π,σ>←GENPERM([[f^m-1]],[[k^t-1]])
4:([[k^-]],[k^-],[[f^-]])←SORT(<π,σ>,([[k^t-1]],[k^-],[[f^-]]))
5:each 1≦i≦m-1 do
6:  [[e_i]]←IFTHEN([[f_i']]:EQ([[k']],[[k'_i+1]]),[[1]])
7:  [[e'_i]]←IFTHEN([[f'_i]]XOR[[f'_i+1]]:[[0]],[[e_i]])
8:  [e'_i]←MODCONV([[e'_i]])
9:  [[e_m]]←1-[[f_m]]
10:[e'_m]←MODCONV([[e_m]])
11:[f^]←MODCONV(1-[[f^-]])
12:[m']=m - Σ_i=1^m[f^_i]
13:each 1≦i≦m do
14:  [x_i]←IFTHEN([e'_i]:[m'],[i])
15:<π',σ'>←GENPERM([[e'-]])
16:([x^-],[[e^-]],[k'^-])←SORT(<π',σ'>,([x^-],[[e^-]],[k'^-]))
17:[c_i]=[x'_i]
18:each 2≦i≦m do
19:  [c_i]=[x'_i]-[x'_i-1]
20:[[e'''-1]]=1-[[e^-]]
21:Output ([k'^-],[c^-],[[e'''-1]])
```

図3

【図４】

アルゴリズム(null処理あり)

```
1:[[k^t-1]]←BITDECOMP([k^-])
2:[[f^m-1]]←1-[[f^-]]
3:<π,σ>←GENPERM([[f^m-1]],[[k^t-1]])
4:([[k^-]],[k^-],[[f^-]])←SORT(<π,σ>,([[k^t-1]],[k^-],[[f^-]]))
5:each 1≦i≦m-1 do
6:  [[e_i]]←IFTHEN([[f'_i]]:EQ([[k']],[[k'_i+1]]),[[1]])
7:  [[e'_i]]←IFTHEN([[f'_i]]XOR[[f'_i+1]]:[[0]],[[e_i]])
8:  [e'_i]←MODCONV([[e'_i]])
9:  [[e_m]]=1-[[f_m]]
10:[e'_m]←MODCONV([[e_m]])
11:[f^]←MODCONV(1-[[f^-]])
12:[m']=m - Σ_i=1^m[f^_i]
13:each 1≦i≦m do
14:  [x_i]←IFTHEN([e'_i]:[m'],[i])
     [k'_i]←IFTHEN([e_i]:[null],[k'_i])
15:<π',σ'>←GENPERM([[e^-]])
16:([x^-],[[e^-]],[k'^-])←SORT(<π',σ'>,([x^-],[[e^-]],[k'^-]))
17:[c_i]=[x'_i]
18:each 2≦i≦m do
19:  [c_i]=[x'_i]-[x'_i-1]
20:[[e'''-1]]=1-[[e^-]]
24:Output ([k'^-],[c^-],[[e'''-1]])
```

図4

【図５】

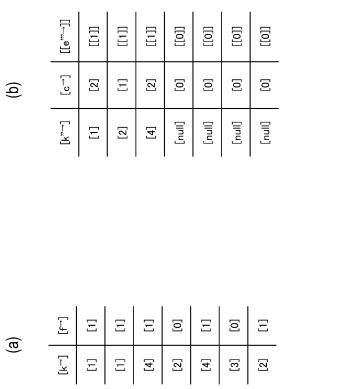

図5

【図６】

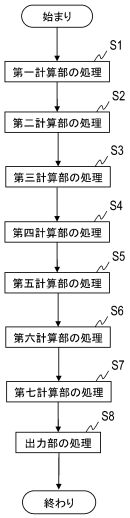

図6

【図７】

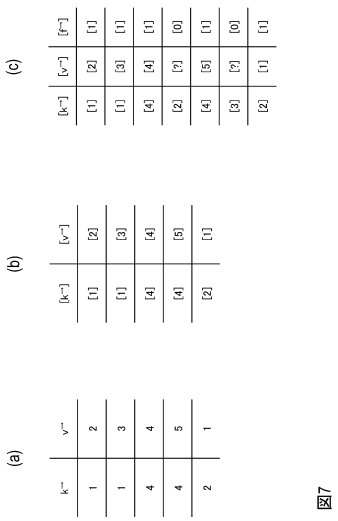

図7

【図８】

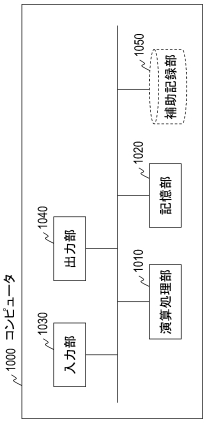

図8

フロントページの続き

(72)発明者　須藤 弘貴
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　中里 裕正

(56)参考文献　特開２０１４－１６４１４５（ＪＰ，Ａ）
国際公開第２０１６／１２０９７５（WO，Ａ１）
菊池亮 他，横断的動線分析を秘密計算でやってみよう，２０２０年 暗号と情報セキュリティシンポジウム予稿集，2020年01月，pp.1-8

(58)調査した分野　(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　２１／７１
Ｇ０６Ｆ　２１／６０
Ｇ０９Ｃ　１／００
Ｇ０６Ｆ　１６／２４５８
ＪＳＴＰｌｕｓ／ＪＭＥＤＰｌｕｓ／ＪＳＴ７５８０（ＪＤｒｅａｍＩＩＩ）
ＩＥＥＥ Ｘｐｌｏｒｅ