

(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
2022年6月16日(16.06.2022)



(10) 国際公開番号

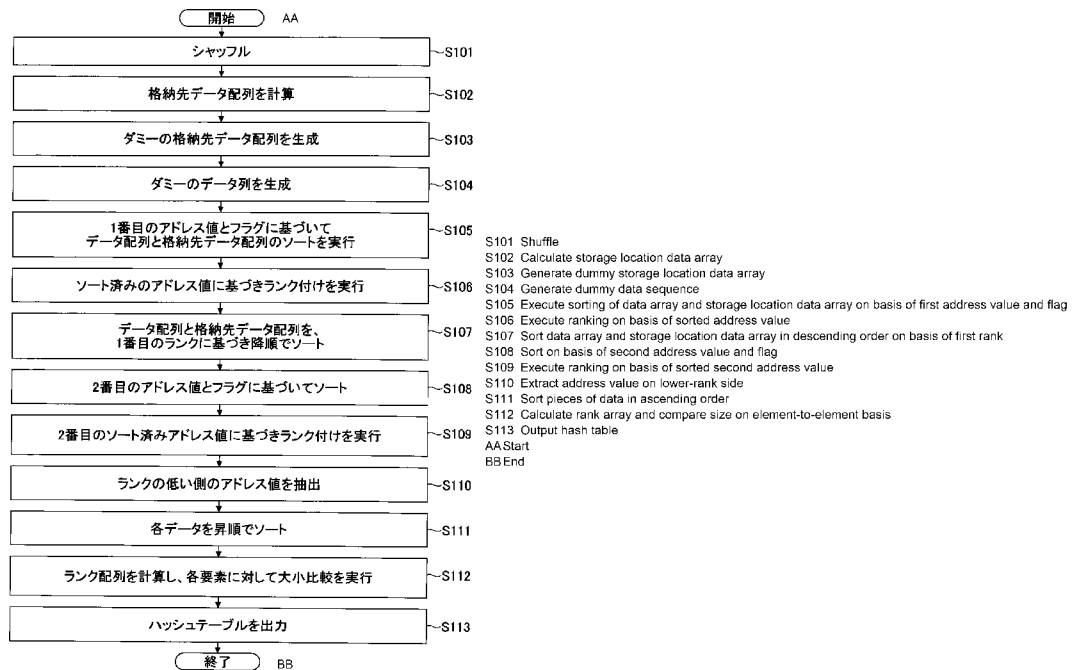
WO 2022/123744 A1

- (51) 国際特許分類:
G09C 1/00 (2006.01)
- (21) 国際出願番号: PCT/JP2020/046125
- (22) 国際出願日: 2020年12月10日(10.12.2020)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (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: 市川 敦謙 (ICHIKAWA, Atsunori); 〒1808585 東京都武蔵野市緑町3丁目9-1 NTT知的財産センタ内 Tokyo (JP).
- (74) 代理人: 伊東 忠重, 外(ITO, Tadashige et al.); 〒1000005 東京都千代田区丸の内二丁目1番1号 丸の内 M Y P L A Z A (明治安田生命ビル) 16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, HN, HR, HU, ID, IL, IN, IR, IS, IT, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY,

(54) Title: SECRET HASH TABLE CONSTRUCTION DEVICE, SECRET HASH TABLE CONSTRUCTION SYSTEM, SECRET HASH TABLE CONSTRUCTION METHOD, AND PROGRAM

(54) 発明の名称: 秘匿ハッシュテーブル構築装置、秘匿ハッシュテーブル構築システム、秘匿ハッシュテーブル構築方法、及びプログラム

[図4]



(57) Abstract: A secret hash table construction device for constructing a secret hash table capable of storing therein Z pieces of data at most at each of B different address values from an actual data sequence including a plurality of pieces of data through secret calculation, the secret hash table construction device comprising an arithmetic operation unit which:

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,
ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG,
US, UZ, VC, VN, WS, 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))

generates a first array obtained by concatenating a storage location data array including a first address value, a second address value, a flag, and a rank for each address value with respect to each piece of data in the actual data sequence and a storage location data array including a first address value, a second address value, a flag, and a rank for each address value serving as a dummy; generates a second array obtained by concatenating the actual data sequence and a dummy data sequence; performs a ranking operation on the same piece of data for the address values; and constructs the secret hash table from the first array and the second array on the basis of the ranking.

(57) 要約：データを複数個含む実データ列から、秘密計算により、B通りのアドレス値にそれぞれ最大Z個までのデータを格納できる秘匿ハッシュテーブルを構築する秘匿ハッシュテーブル構築装置であって、前記実データ列の各データに対する、第1アドレス値及び第2アドレス値と、フラグと、アドレス値毎のランクとを含む格納先データ配列と、ダミーとしての、第1アドレス値及び第2アドレス値と、フラグと、アドレス値毎のランクとを含む格納先データ配列とを連結した第1配列を生成し、前記実データ列とダミーのデータ列とを連結した第2配列を生成し、アドレス値に対して同一データに対するランク付け操作を行って、当該ランク付けに基づいて、前記第1配列と前記第2配列から、前記秘匿ハッシュテーブルを構築する演算部を備える。

明 細 書

発明の名称：

秘匿ハッシュテーブル構築装置、秘匿ハッシュテーブル構築システム、秘匿ハッシュテーブル構築方法、及びプログラム

技術分野

[0001] 本発明は、秘密計算に関するものであり、特に配列の順序を秘匿した操作に関するものである。

背景技術

[0002] ハッシュテーブルは、データのキー値をハッシュ関数等により符号化しデータ配列におけるアドレス値との紐付けを行うことで、キー値に基づくデータ検索を高速で行うためのデータ構造である。これは例えば、外部サーバ等にデータを預け、随時キー値による検索を行いたい場合などに有用である。

[0003] しかし、通常のハッシュテーブルはデータ秘匿を想定するものではなく、例えばユーザからデータ（ないしハッシュテーブル）を委託されたサーバがテーブル構造を識別でき、それによりユーザのアクセス傾向を観測しうるという問題がある。こうした問題を解決する既存の方法として、秘匿ハッシュテーブル（例えば、参考文献[4, 1]）がある。なお、参考文献名については、明細書の最後にまとめて記載した。秘匿ハッシュテーブルは、サーバに対してテーブル構造を秘匿したままでハッシュテーブルを構築させる技術であり、サーバに対してユーザのアクセス傾向を秘匿できる。

[0004] 秘匿ハッシュテーブル構築方法として自明に挙げられるのは、ユーザが全てのデータを手元で暗号化しハッシュテーブルを構築、サーバへ委託する方法であるが、この場合データ数 n に対してユーザの記憶領域が $O(n)$ 必要となる。上記に挙げた参考文献[4]及び[1]は、全ての暗号化データをサーバが持っている前提で、ユーザ・サーバの協調プロトコルによってサーバ上に秘匿ハッシュテーブルを構築する方法であり、この時ユーザの記憶領域は $O(1)$ で十分であるという利点を持つ。

[0005] しかしながら、参考文献[4, 1]の技術は秘匿ハッシュテーブル構築の際、ユーザ・サーバ間で非常に多くの通信を要するという課題がある。現実には、ユーザの通信環境や計算性能が必ずしも好条件であるとは限らないことから、上記コストは可能な限り低減されることが望ましい。これを解決した方法の1つに、非特許文献1に開示された技術が挙げられる。この文献には、複数台のサーバを利用した秘密計算により、ユーザ・サーバ間での通信を一切要さずに秘匿ハッシュテーブルを構築する手法が示されており、そのためユーザ端末の通信環境や計算性能を問わず安定した性能を実現できる。

[0006] 特に、非特許文献1にて示される手法のうち”Oblivious Greedy Hashing”と呼ばれる秘匿ハッシュ法は、格納されるべき各データに対してそれぞれ2通りのアドレス値（すなわち、テーブルのどの位置に格納されるべきかを示す値）を与えることで、各データが2つの格納先のうちより空いている方へと格納されるようなテーブル構築アルゴリズムを実現している。これにより、他の既存手法に比較してテーブルサイズを小さく抑えられるといった利点がある。

先行技術文献

非特許文献

[0007] 非特許文献1：市川敦謙，濱田浩気，菊池亮，五十嵐大．3 パーティ計算上の最適秘匿ハッシュと劣対数効率のOblivious RAM．暗号と情報セキュリティシンポジウム(SCIS)2020 予稿集(2020)．

発明の概要

発明が解決しようとする課題

[0008] 非特許文献1に開示されているOblivious Greedy Hashingはテーブルサイズが小さく、参考文献[4, 1]に開示の技術と違いユーザ・サーバ間での通信が不要であるといったメリットの一方で、テーブル構築の際、単純にシステム全体の通信量で比較した場合に非常に多くの通信を要する欠点もある。具体的には、 n をテーブルに格納するデータの

総数とすると、参考文献[4]や[1]の手法で秘匿ハッシュテーブルを構築する際には $O(n \log^2 n)$ ビットの通信を要するのに対し、Oblivious Greedy Hashingは $O(Z \times n \log^2 n)$ ビットの通信が必要となる。ここで、 Z とは方式の安全性にまつわるパラメータであり、非特許文献1では $Z = 2 \log^2 n$ が例示されている。

[0009] 本発明は上記の点に鑑みてなされたものであり、テーブルサイズを小さくしつつ、通信量を増大させないで秘匿ハッシュテーブルを構築する技術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

[0010] 開示の技術によれば、キーと、データがダミーか否かを示すフラグとを有するデータを複数個含む実データ列から、秘密計算により、 B 通りのアドレス値にそれぞれ最大 Z 個までのデータを格納できる秘匿ハッシュテーブルを構築する秘匿ハッシュテーブル構築装置であって、

前記実データ列の各データに対する、第1アドレス値及び第2アドレス値と、フラグと、アドレス値毎のランクとを含む格納先データ配列と、ダミーとしての、第1アドレス値及び第2アドレス値と、フラグと、アドレス値毎のランクとを含む格納先データ配列とを連結した第1配列を生成し、

前記実データ列とダミーのデータ列とを連結した第2配列を生成し、

前記第1配列における第1アドレス値とフラグに基づいて、前記第1配列と前記第2配列のそれぞれをソートし、ソートした前記第1配列の第1アドレス値に対して同一データに対するランク付け操作を行い、当該ランク付けに基づいて前記第1配列と前記第2配列のそれぞれをソートし、

前記第1配列における第2アドレス値とフラグに基づいて、前記第1配列と前記第2配列のそれぞれをソートし、ソートした前記第1配列の第2アドレス値に対して同一データに対するランク付け操作を行い、

前記第1配列において、第1アドレス値と第2アドレス値のうち、ランクの低い側のアドレス値を抽出し、抽出したアドレス値とフラグを有する第3配列を生成し、

前記第3配列におけるアドレス値から計算されたランク配列の各要素とZとの比較により得られた第4配列を用いて、前記第2配列をソートし、ソートした前記第2配列におけるBZ個の要素を前記秘匿ハッシュテーブルとして出力する

演算部を備える秘匿ハッシュテーブル構築装置が提供される。

発明の効果

[0011] 開示の技術によれば、テーブルサイズを小さくしつつ、通信量を増大させないで秘匿ハッシュテーブルを構築する技術が提供される。

図面の簡単な説明

[0012] [図1]秘匿ハッシュテーブル構築システムの構成図である。

[図2]サーバの機能構成図である。

[図3]装置のハードウェア構成例を示す図である。

[図4]実施例1の処理手順を示すフローチャートである。

[図5]実施例2の処理手順を示すフローチャートである。

[図6]実施例3の処理手順を示すフローチャートである。

[図7]実施例4の処理手順を示すフローチャートである。

[図8]実施例5の処理手順を示すフローチャートである。

発明を実施するための形態

[0013] 以下、図面を参照して本発明の実施の形態（本実施の形態）を説明する。

以下で説明する実施の形態は一例に過ぎず、本発明が適用される実施の形態は、以下の実施の形態に限られるわけではない。

[0014] （実施の形態の概要）

本実施の形態では、複数台のサーバを用いてユーザ・サーバ間通信コストを削減した効率的な秘匿ハッシュテーブル構築方法、及び操作方法について説明する。本実施の形態に係る技術により、各データに2通りの格納先を付与することでOblivious Greedy Hashingと同等にテーブルサイズを小さく抑えつつも、サーバ間通信量 $O(n \log^2 n)$ ビットのみでテーブル構築が可能である。以下、本実施の形態のシステム構成、

及び処理手順について詳細に説明する。

[0015] (システム構成)

図1に、本実施の形態における秘匿ハッシュテーブル構築システムの構成例を示す。図1に示すように、本実施の形態の秘匿ハッシュテーブル構築システムは、複数のサーバ100-1～100-Nが通信ネットワーク上200に備えられた構成を有する。各サーバはコンピュータである。当該コンピュータは物理マシンであってもよいし、クラウドにより提供される仮想マシンであってもよい。

[0016] 図1に示すユーザ端末300が、秘匿ハッシュテーブル構築システムにアクセスして、例えばキーに対応するデータを取得する。

[0017] 本実施の形態では、これら複数のサーバ100-1～100-N間の協調プロトコルにより秘密計算で秘匿ハッシュテーブル構築に係る処理を実行する。ただし、本発明の実施にあたって、複数サーバの協調プロトコルを使用することは例であり、これに限定されるわけではない。本発明は、秘密計算を行うことができるどのような技術にも適用可能である。例えば、1つのサーバで秘密計算を行うことができる場合に、そのサーバで本発明に係る技術を実行することも可能である。

[0018] 図2に、サーバ100の構成例を示す。図2に示すサーバ100の構成例は、複数のサーバ100-1～100-Nを仮想的に1つのサーバとみなした場合のその構成である。この場合、図2に示すサーバ100を秘匿ハッシュテーブル構築システムと呼んでもよい。また、当該サーバ100を、秘匿ハッシュテーブル構築装置と呼んでもよい。また、1つのサーバで秘密計算を行う場合におけるそのサーバが図2に示すサーバ100であってもよい。

[0019] 図2に示すように、サーバ100は、入力部110、演算部120、出力部130、データ格納部140を有する。演算部120は、秘匿ハッシュテーブル構築及び操作に係る処理を実行する。データ格納部140には、計算にあたって予め用意しておくデータ、計算途中の秘匿した配列等のデータ、テーブルが格納される。入力部110は、例えば、ユーザ端末300からの

要求を受信し、入力する。出力部１４０は、例えば、ユーザ端末３００からの要求に対する演算結果をユーザ端末３００に返す。

[0020] (ハードウェア構成例)

本実施の形態におけるサーバ１００（秘匿ハッシュテーブル構築装置）は、例えば、コンピュータに、本実施の形態で説明する処理内容を記述したプログラムを実行させることにより実現可能である。なお、この「コンピュータ」は、物理マシンであってもよいし、クラウド上の仮想マシンであってもよい。仮想マシンを使用する場合、ここで説明する「ハードウェア」は仮想的なハードウェアである。

[0021] 上記プログラムは、コンピュータが読み取り可能な記録媒体（可搬メモリ等）に記録して、保存したり、配布したりすることが可能である。また、上記プログラムをインターネットや電子メール等、ネットワークを通して提供することも可能である。

[0022] 図３は、上記コンピュータのハードウェア構成例を示す図である。図３のコンピュータは、それぞれバスＢで相互に接続されているドライブ装置１０００、補助記憶装置１００２、メモリ装置１００３、ＣＰＵ１００４、インタフェース装置１００５、表示装置１００６、入力装置１００７、出力装置１００８等を有する。

[0023] 当該コンピュータでの処理を実現するプログラムは、例えば、ＣＤ－ＲＯＭ又はメモリカード等の記録媒体１００１によって提供される。プログラムを記憶した記録媒体１００１がドライブ装置１０００にセットされると、プログラムが記録媒体１００１からドライブ装置１０００を介して補助記憶装置１００２にインストールされる。但し、プログラムのインストールは必ずしも記録媒体１００１より行う必要はなく、ネットワークを介して他のコンピュータよりダウンロードするようにしてもよい。補助記憶装置１００２は、インストールされたプログラムを格納すると共に、必要なファイルやデータ等を格納する。

[0024] メモリ装置１００３は、プログラムの起動指示があった場合に、補助記憶

装置1002からプログラムを読み出して格納する。CPU1004は、メモリ装置1003に格納されたプログラムに従って、当該サーバ100に係る機能を実現する。インタフェース装置1005は、ネットワークに接続するためのインタフェースとして用いられる。表示装置1006はプログラムによるGUI (Graphical User Interface) 等を表示する。入力装置1007はキーボード及びマウス、ボタン、又はタッチパネル等で構成され、様々な操作指示を入力させるために用いられる。出力装置1008は演算結果を出力する。

[0025] (秘密計算に係る基本的な処理について)

まず、サーバ100の演算部120により実行される秘匿ハッシュテーブル構築や操作の処理において前提となる、秘密計算に係る基本的な処理について説明する。これら基本的な処理自体は既存技術である。

[0026] <秘匿化・復元>

以下では、値 x の秘匿値を $[[x]]$ と表し、処理 $x \rightarrow [[x]]$ を x の秘匿化、 $[[x]] \rightarrow x$ を x の復元と呼ぶ。本実施の形態では、秘匿化・復元処理を行うための技術として秘密分散法（参考文献[6, 5]）を使用することを想定しているが、同等の機能及び安全性を満たす方法ならばどのような秘匿化・復元処理技術を用いてもよい。

[0027] <基本演算>

秘匿値同士の加算・減算・乗算をそれぞれ以下のように表す。

[0028] $[[a + b]] \leftarrow [[a]] + [[b]]$

$[[a - b]] \leftarrow [[a]] - [[b]]$

$[[a \times b]] \leftarrow [[a]] \times [[b]]$

a 、 b が len ビットで表現される空間にあるとすると、参考文献[7]の技術では加算と減算を通信コスト0、乗算を通信コスト $O(len)$ ビットで実行できる。

[0029] <比較>

秘匿値同士の比較を以下のように表す。

[0030] $[[c]] \leftarrow [[a]] = ? [[b]]$

上記の比較は、 a と b が等しいかどうかの判定結果が c （1ビット数）であることを示す。正しければ1、そうでなければ0になる。

[0031] $[[d]] \leftarrow [[a]] \leq ? [[b]]$

上記の比較は、 a が b 以下であるか否かの判定結果が d （1ビット数）であることを示す。正しければ1、そうでなければ0になる。

[0032] 比較の演算は、 a 、 b が len ビットで表現される空間にあるとすると、参考文献[7]の技術では通信コスト $O(len)$ ビットで実行できる。

[0033] <内積>

秘匿値のベクトル又は配列 $[[a]] = ([a_0], \dots, [a_{n-1}])$, $[[b]] = ([b_0], \dots, [b_{n-1}])$ の内積を以下のように表す。

[0034] $[[c]] \leftarrow \langle [[a]], [[b]] \rangle$

ベクトル a 、 b の各要素がそれぞれ len ビットで表現される空間にあるとすると、参考文献[7]の技術では内積を通信コスト $O(len)$ ビットで実行できる。

[0035] <配列のシャッフル>

秘密の配列 $[[A]]$ をシャッフルする処理を以下のように表す。

[0036] $[[A']] \leftarrow \text{Shuffle}([A])$

A が len ビットの要素を n 個持つ配列であるとき、参考文献[3]の技術ではこの処理を通信コスト $O(len \cdot n)$ ビットで実行できる。

[0037] <安定ソート>

秘密のキー配列 $[[K]]$ 及びソート対象配列 $[[A]]$ に対し、 $[[A]]$ を $[[K]]$ の各値に基づきソートする処理を以下のように表す。

[0038] $[[A']] \leftarrow \text{Sort}([K], [A])$

この処理は K に関して昇順・降順のいずれも可能である。 K が len ビットの要素を n 個持つ配列、 A が m ビットの要素を n 個持つ配列であるとき、参考文献[3]の技術ではこの処理を通信コスト $O(len \cdot n \log n +$

$m n$) ビットで実行できる。

[0039] <同一データのランク付け>

配列 $[[A]] = ([[a_0]], \dots, [[a_{n-1}]])$ を、ソート済みの配列の配列とする。このとき、同一データに対するランク付け操作を下記のように表す。

[0040] $[[B]] \leftarrow \text{Rank}([A])$

ただし、 $[[B]] = ([[b_0]], \dots, [[b_{n-1}]])$ であり、各 b_i は以下を満たす：

- ・ $b_0 = 1$
- ・ 全ての $i \in \{1, \dots, n-1\}$ において、 $a_{i-1} \neq a_i$ ならば $b_i = 1$
- ・ 全ての $i \in \{1, \dots, n-1\}$ において、 $a_{i-1} = a_i$ ならば $b_i = b_{i-1} + 1$

具体的に、例えば $A = (0, 0, 0, 0, 1, 1, 2, 2, 2)$ のとき、 $B = (1, 2, 3, 4, 1, 2, 1, 2, 3)$ となる。参考文献 [1] に開示されている Range Prefix Sum を用いれば、この操作を通信コスト $O(n \log n)$ で実行できる。

[0041] <擬似ランダム関数>

値 a, s をそれぞれ、擬似ランダム関数の入力及び秘密鍵であるとする。秘匿値 $[[a]]$, $[[s]]$ に対し、秘密の擬似ランダム値を計算する操作を以下のように表す。

[0042] $[[r]] \leftarrow \text{PRF}([a], [s])$

a のビット長を len とすると、この秘密計算処理は例えば参考文献 [2] の技術を用いて通信コスト $O(len)$ で実現できる。

[0043] <秘密計算による暗号化>

値 a, s をそれぞれ、ブロック暗号の平文及び秘密鍵であるとする。秘匿値 $[[a]]$, $[[s]]$ に対し、秘密計算により暗号文を生成する操作を以下のように表す。

[0044] $[[c]] \leftarrow \text{Enc}([a], [s])$

a のビット長を len とすると、この秘密計算処理は参考文献 [2] の技術を用いて通信コスト $O(len)$ で実現できる。

[0045] 以下、本実施の形態に係るサーバ 100 が実行する処理の具体例について、実施例 1 ～ 5 を用いて説明する。以下の処理はサーバ 100 の演算部 120 により実行される。また、予め用意しておくデータ、計算により得られる配列、テーブル等はデータ格納部 140 に格納される。演算部 120 は、データ格納部 140 から読み出したデータに対して演算を行い、演算結果をデータ格納部 140 に格納することを繰り返しながら処理を進める。

[0046] なお、下記の各実施例における処理は一例である。例えば、1 と 0 の意味や、昇順／降順などが、下記の各実施例におけるものと逆であってもよい。

[0047] (実施例 1)

まず、実施例 1 を説明する。本実施例 1 では、サーバ 100 が、データ格納部 140 に格納されているキー k_i とフラグ e_i からなるデータ $a_i = (k_i, e_i)$ を用いて秘匿ハッシュテーブルを構築する方法を示す。キー k_i はデータ毎に固有の識別子であり、アクセスの際などにデータを一意に特定する目的に用いる。

[0048] フラグ e_i はデータがダミーか否かを判定する 1 bit の値であり、ここでは $e_i = 0$ なら実際のデータ、 $e_i = 1$ ならダミーデータであるとする。なお、データの総数が n の場合において、 k_i のビット長は最小で $ceil(\log n)$ となるが、冗長性があっても構わないとする。また、各データにはキーとフラグ以外にも、任意の情報 v_i を付与し、 $a_i = (k_i, e_i, v_i)$ のようにしても良いとする。

[0049] 前提として、サーバ 100 が、データ格納部 140 の中に、秘匿化されたデータ列 $[[A]] = ([[a_0]], \dots, [[a_{n-1}]])$; $[[a_i]] = ([[k_i]], [[e_i]])$ を保持しているものとする。

[0050] また、以下では事前に秘匿化された秘密鍵 $[[s_0]]$, $[[s_1]]$ を生成し、データ格納部 140 に保持しておくものとするが、プロトコル中に逐次生成しても構わない。

[0051] 本実施例1で構築する秘匿ハッシュテーブルは、サイズ $B \times Z$ のデータ配列であり、別の表現をするならば B 通りのアドレス値にそれぞれ最大 Z 個までのデータを格納できるデータ構造である、とする。また、本実施例1で用いる擬似ランダム関数は、 $PRF([k], [s]) \in \{0, \dots, B-1\}$ を満たすとする。

[0052] 上記の前提の下、サーバ100は、以下で説明する処理により、秘匿ハッシュテーブルを構築する。図4のフローチャートの手順に沿って、サーバ100が実行する処理を説明する。

[0053] <S101 (ステップ101)>

S101において、サーバ100は、 $[A]$ に対してシャッフルを行い $[A] \leftarrow \text{Shuffle}([A])$ とする。ただしこの操作は、入力となる $[A]$ の状態や満たすべき安全性によっては実行しなくてもよいものとする。

[0054] <S102>

S102において、サーバ100は、 $[A]$ に対して、格納先データ配列 $[ADDR] = (([addr^0_0], [addr^1_0], [e_0], [rank^0_0], [rank^1_0]), \dots, ([addr^0_{n-1}], [addr^1_{n-1}], [e_{n-1}], [rank^0_{n-1}], [rank^1_{n-1}]))$ を計算する。

[0055] ただし、 $[addr^p_i] \leftarrow PRF([k_i], [s_p])$; $i \in \{0, \dots, n-1\}$, $p \in \{0, 1\}$ とし、また $[rank^p_i] \leftarrow [0]$ と初期化する。ここで $addr^0_i$, $addr^1_i$ はそれぞれ A の i 番目のデータ $a_i = (k_i, e_i)$ に対応する2通りのアドレス値に相当し、また $rank^p_i$ は後にランク付けをする際のランク保持用領域である。

[0056] なお、データのイメージを説明する便宜上、秘匿の記号 $[[]]$ なしで説明を行うこととすると、この時点での $ADDR$ は、例えば、 $ADDR = ((3, 1, 0, 0, 0), (18, 95, 0, 0, 0), \dots, (8, 78, 0, 0, 0))$ といった配列(n 個のデータの配列)になる。以下、適宜

、 $[[[]]]$ なしで説明を行う場合がある。

[0057] <S 1 0 3>

S 1 0 3 において、サーバ 1 0 0 は、ダミーの格納先データ配列 $[[ADDR_d]] = ([[d_0]], \dots, [[d_{BZ-1}]]); d_i = (\text{floor}(i/Z), \text{floor}(i/Z), 1, 0, 0)$ を生成する。

[0058] これは 2 通りのアドレス値 $addr^0_i, addr^1_i$ が同じ値 $\text{floor}(i/Z) \in \{0, \dots, B-1\}$ を指しているダミーの格納先データを、アドレス値 $0, \dots, B-1$ ごとに Z 個ずつ保持する配列である。

[0059] サーバ 1 0 0 は、このダミーの $[[ADDR_d]]$ を、実データに関する格納先データ配列 $[[ADDR]]$ の末尾に連結し、長さ $n+BZ$ の配列 $[[ADDR']] = [[ADDR]] \parallel [[ADDR_d]]$ とする。

[0060] この時点での $ADDR'$ は、例えば、 $ADDR' = ((3, 1, 0, 0, 0), (18, 95, 0, 0, 0), \dots, (8, 78, 0, 0, 0), (0, 0, 1, 0, 0), (0, 0, 1, 0, 0), \dots, (8, 8, 1, 0, 0))$ といった配列 ($n+BZ$ 個のデータの配列) になる。

[0061] なお、以下では格納先データ配列 $[[ADDR']]$ に対して、記法 $[[ADDR'.addr_0]]$ を、 $[[ADDR']]$ から 1 番目のアドレス値 $[[addr^0_i]]$ のみを取り出した配列 ($[[addr^0_0]], \dots, [[addr^0_{n+BZ-1}]]$) であると定義する。同様に、記法 $[[ADDR'.addr_1]], [[ADDR'.rank_0]], [[ADDR'.rank_1]]$ を、それぞれ $[[ADDR']]$ から $[[addr^1_i]], [[rank^0_i]], [[rank^1_i]]$ を取り出した配列と定義する。

[0062] さらに、記法 $[[ADDR'.pos_0]], [[ADDR'.pos_1]]$ によって、それぞれ $[[ADDR']]$ からペア ($[[addr^0_i]], [[e_i]]$)、($[[addr^1_i]], [[e_i]]$) を抽出した配列であるとする。

[0063] 例えば、 $ADDR' = ((3, 1, 0, 0, 0), (18, 95, 0, 0, 0), \dots, (8, 78, 0, 0, 0), (0, 0, 1, 0, 0), ($

0, 0, 1, 0, 0), ..., (8, 8, 1, 0, 0)) であるとする
 、 $ADDR' . pos_0 = ((3, 0), (18, 0), \dots, (8, 0), (0, 1), (0, 1), \dots, (8, 1))$ となる。

[0064] <S104>

S104において、サーバ100は、ダミーのデータ列 $[[A_d]] = ([a^d_0], \dots, [a^d_{BZ-1}])$; $a^d_i = (dummy, 1)$ を生成する。ただし、ここで $dummy$ は予め定めておいた、いずれのキー k_i とも異なる値であるとする。この $[[A_d]]$ を実データ列 $[[A]]$ の末尾に連結し、長さ $n + BZ$ の配列 $[[A']] = [[A]] \parallel [[A_d]]$ とする。

[0065] この時点での A' は、例えば、 $A' = ((11, 0), (101, 0), \dots, (3, 0), (dummy, 1), (dummy, 1), \dots, (dummy, 1))$ といった配列($n + BZ$ 個のデータの配列)になる。

[0066] <S105>

S105において、サーバ100はまず、1番目のアドレス値 $addr^0_i$ とフラグ e_i に基づいてデータ配列と格納先データ配列のそれぞれのソートを行う。すなわち、 $[[A']] \leftarrow Sort([ADDR' . pos_0], [[A']])$ 、及び $[[ADDR']] \leftarrow Sort([ADDR' . pos_0], [[ADDR']])$ を計算する。

[0067] この時、データ配列及び格納先データ配列はアドレス値による昇順かつ同アドレス値においては実データ>ダミーデータの順で並べ替えられるものとするが、以後のソートやランク付けの昇順・降順を調整することでソート順序を変更することもできる。

[0068] ソート後の A' は、例えば、 $A' = ((3, 0), (dummy, 1), (dummy, 1), \dots, (11, 0), (dummy, 1), \dots,)$ といった配列($n + BZ$ 個のデータの配列)になる。

[0069] <S106>

S106において、サーバ100は、ソート済みのアドレス値に基づきランク付けを行う。すなわち、 $[[ADDR' . rank_0]] \leftarrow Rank([$

$[ADDR'.addr_0]]$) を計算する。

[0070] $ADDR'.rank_0$ は、例えば、 $ADDR'.rank_0 = (1, 2, 3, 4, \dots, 1, 2, 3, \dots, 1, 2, 3, 4, 5)$ といった配列 ($n + BZ$ 個のデータの配列) になる。

[0071] <S107>

S107において、サーバ100は、データ配列及び格納先データ配列を、1番目のランクに基づき降順でソートする。すなわち、 $[[A']] \leftarrow \text{Sort}([[ADDR'.rank_0]], [[A']])$ 及び $[[ADDR']] \leftarrow \text{Sort}([[ADDR'.rank_0]], [[ADDR']])$ を、 $rank^0_i$ が大きい順に並ぶよう計算する。なお、この操作は単に各配列を昇順でソートしたうえ、逆順に並べ替えることでも同様に実現できる。

[0072] <S108>

S108において、サーバ100は、S105での処理と同様に、今度は2番目のアドレス値 $addr^1_i$ とフラグに基づいてソートを行う。すなわち、 $[[A']] \leftarrow \text{Sort}([[ADDR'.pos_1]], [[A']])$ 及び $[[ADDR']] \leftarrow \text{Sort}([[ADDR'.pos_1]], [[ADDR']])$ を計算する。この時、安定ソートを用いれば、同アドレス値においては実データ>ダミーデータかつ高ランク>低ランクの順に並べ替えることができている。

[0073] <S109>

S109において、サーバ100は、S106での処理と同様に、2番目のソート済みアドレス値に基づきランク付けを行う。すなわち、 $[[ADDR'.rank_1]] \leftarrow \text{Rank}([[ADDR'.addr_1]])$ を計算する。

[0074] <S110>

S110において、サーバ100は、格納先データ配列 $[[ADDR']]$ の各要素 $[[b_i]] = ([[addr^0_i]], [[addr^1_i]], [[e_i]], [[rank^0_i]], [[rank^1_i]]))$ について、それぞれラ

ソートの低い側のアドレス値を抽出する。すなわち、大小比較 $[[z_i]] \leftarrow [[rank^0_i]] \leq [[rank^1_i]]$ に基づき、 $[[addr_i]] \leftarrow [[addr^1_i]] + [[z_i]] \times ([[addr^0_i]] - [[addr^1_i]])$ を計算する。その後、選ばれたアドレス値とフラグを併せた配列 $[[ADDR^{fin}]] = ([[addr_0]], [[e_0]]), \dots, ([[addr_{n+BZ-1}]], [[e_{n+BZ-1}]])$ を計算する。

[0075] <S 1 1 1>

S 1 1 1において、サーバ100は、新たに得た配列 $[[ADDR^{fin}]]$ に基づき、各データを昇順でソートする。すなわち、 $[[A']] \leftarrow \text{Sort}([[ADDR^{fin}]], [[A']])$ 及び $[[ADDR^{fin}]] \leftarrow \text{Sort}([[ADDR^{fin}]], [[ADDR^{fin}]])$ を計算する。

[0076] <S 1 1 2>

S 1 1 2において、サーバ100は、 $[[ADDR^{fin}]]$ のアドレス値部分のみを抽出した配列 $[[ADDR^{fin}, addr]]$ を用いて、ランク配列 $[[R]] = \text{Rank}([[ADDR^{fin}, addr]])$ を計算し、さらに $[[R]]$ の各要素 $[[r_i]]$ に対して大小比較 $[[y_i]] \leftarrow [[[r_i]] \leq Z]$ を計算した結果の配列 $[[Y]] = ([[y_0]], \dots, [[y_{n+BZ-1}]])$ を得る。このとき、 $y_i = 1$ のデータはテーブルに格納され、 $y_i = 0$ のデータは削除されることを示している。

[0077] <S 1 1 3>

S 1 1 3において、サーバ100は、 $[[Y]]$ を用いてデータ配列を昇順でソートする。すなわち、 $[[A']] \leftarrow \text{Sort}([[Y]], [[A']])$ を計算する。その後、末尾のBZ個の要素のみをハッシュテーブルとして出力し、その他は削除する。また、ハッシュテーブルに付随するアクセス用情報として2つの秘密鍵 $[[s_0]], [[s_1]]$ も出力する。

[0078] 本実施例1において、 $[[k_i]]$ を $O(\log N)$ ビットとし、 $B \times Z = O(n)$ とすれば、サーバ間通信量はソート、ランク付け及び比較により $O(n \log^2 n)$ ビットとなる。

[0079] (実施例2)

次に、実施例2を説明する。本実施例2では、実施例1と同等のテーブル構築を、より効率的に行う方法を示す。実施例1では、常に格納先データ配列と共にデータ配列 $[[A]]$ （ないし $[[A']]$ ）がソートされており、単にソート回数が増していた他、仮にキーとフラグに対して大きなデータ v_i が付与されていた場合、データ配列のソートのために効率悪化を招く可能性もあった。

[0080] 本実施例2では原則として実施例1のアルゴリズムを踏襲しつつも、秘密計算上での暗号化を利用してソート回数を削減し、以下のように秘匿ハッシュテーブルを構築する。以下、図5のフローチャートの手順に沿って説明する。

[0081] <S201>

S201において、サーバ100は、実施例1のS101と同様に、 $[[A]]$ に対してシャッフルを行い $[[A]] \leftarrow \text{Shuffle}([A])$ とする。ただしこの操作は、入力となる $[[A]]$ の状態や満たすべき安全性によっては実行しなくてもよいものとする。

[0082] <S202>

S202において、サーバ100は、秘匿化されたブロック暗号の秘密鍵 $[[s]]$ を生成し、 $[[A]]$ に対応する秘密のタグ配列 $[[Tag]] = ([tag_0], \dots, [tag_{n-1}])$; $[tag_i] \leftarrow \text{Enc}([i], [s])$ を計算する。なお、秘密鍵 $[[s]]$ はプロトコルの開始以前に生成していても構わないとする。

[0083] <S203>

S203において、サーバ100は、実施例1のS102と同様に、格納先データ配列 $[[ADDR]] = (([addr^0_0], [addr^1_0], [e_0], [rank^0_0], [rank^1_0], [tag_0]), \dots, ([addr^0_{n-1}], [addr^1_{n-1}], [e_{n-1}], [rank^0_{n-1}], [rank^1_{n-1}], [tag_{n-1}])$

）を計算する。実施例1とは、タグ $[[tag_i]]$ が含まれていることのみが異なる。

[0084] <S204>

S204において、サーバ100は、実施例1のS103と同様に、ダミーの格納先データ配列 $[[ADDR_d]] = ([[d_0]], \dots, [[d_{BZ-1}]])$; $[[d_i]] = ([[floor(i/Z)]] , [[floor(i/Z)]] , [[1]] , [[0]] , [[0]] , [[tag^d_i]])$ を生成し、 $[[ADDR']] = [[ADDR]] \parallel [[ADDR_d]]$ とする。ただし、 $[[tag^d_i]] \leftarrow PRF([[n+i]] , [[s]])$ であるとする。

[0085] <S205>

S205において、サーバ100は、実施例1のS104と同様に、ダミーのデータ列 $[[A_d]] = ([[a^d_0]] , \dots, [[a^d_{BZ-1}]])$; $a^d_i = (dummy, 1)$ を生成し、 $[[A']] = [[A]] \parallel [[A_d]]$ とする。また、本実施例2のS204で計算したタグの列 $[[Tag_d]] = ([[tag^d_0]] , \dots, [[tag^d_{BZ-1}]])$ を用いて、 $[[Tag']] \leftarrow [[Tag]] \parallel [[Tag_d]]$ とする。

[0086] <S206>

S206において、サーバ100は、配列 $[[A']]$ 及び $[[Tag']]$ の並び順を秘匿・ランダム化するために、 $([[A']] , [[Tag']]) \leftarrow Shuffle([A'] , [Tag'])$ を計算する。この操作では、2つの配列の対応関係を保ったままランダム化するため、並列に同じシャッフル処理を実行している。

[0087] <S207>

S207において、サーバ100は、実施例1のS105～S113と同様の処理を行う。ただし、実施例2において、実施例1のS105～S112に対応する各手順においては配列 $[[A']]$ には一切手を触れず格納先データ配列のみに操作を行うこととし、また、実施例1のS110に対応す

る手順で新たな格納先データ配列を得る際には $[[ADDR^{fin}]]$ の各要素を $([[addr_i]], [[e_i]], [[tag_i]])$ として、タグ情報を引き継ぐようにする。

[0088] <S208>

S208において、サーバ100は、実施例1のS112に対応する手順で得た $[[Y]]$ を用いて $[[ADDR^{fin}]]$ を昇順でソートする。すなわち、 $[[ADDR^{fin}]] \leftarrow \text{Sort}([Y], [[ADDR^{fin}]])$ を計算する。その後、末尾のBZ個の要素のみ残し、その他は削除する。

[0089] <S209>

S209において、サーバ100は、タグ配列 $[[Tag']]$ の全ての要素を復元し、平文に戻す。同時に、 $[[ADDR^{fin}]]$ に含まれる全てのタグを復元し、平文に戻す。データ配列 $[[A']]$ の中から、対応するタグが $[[ADDR^{fin}]]$ に含まれるような要素BZ個を取り出し、 $[[ADDR^{fin}]]$ の並び順の通りに並べることでハッシュテーブルとする。最後にハッシュテーブルと秘密鍵 $[[s_0]]$, $[[s_1]]$ を出力する。

[0090] 本実施例2において、 $[[k_i]]$ を $O(\log n)$ ビットとし、 $B \times Z = O(n)$ とすれば、サーバ間通信量はソート、ランク付け及び比較により $O(n \log^2 n)$ ビットとなる。実施例1では任意のデータ v_i を含む組 (k_i, e_i, v_i) のデータサイズが $\omega(\log n)$ ビットとなる場合にサーバ間通信量が $\omega(n \log^2 n)$ ビットとなってしまうのに対し、本実施例は v_i のサイズに関わらず $O(n \log^2 n)$ ビットで済む利点がある。

[0091] (実施例3)

次に、実施例3を説明する。実施例3では、実施例1、2で構築した秘匿ハッシュテーブルへのデータ参照方法について説明する。前提として、サーバ100は、実施例1ないし2で構築された秘匿ハッシュテーブル（サイズ $B \times Z$ ）と秘密鍵 $[[s_0]]$, $[[s_1]]$ をデータ格納部140に保持しているとする。また、ユーザ端末300はアクセスしたいキー k を持っているとするが、これはサーバ同士の合意に基づいてサーバ自身で選定しても構

わない。以下、図6のフローチャートに示す手順に沿って説明する。

[0092] <S301>

S301において、ユーザ端末300は、サーバ100に対し、アクセスしたいデータに対応するキーの秘匿値 $[[k]]$ を送る。ただし、この秘匿キー値はユーザ端末300からサーバ100への要求だけでなく、サーバ間合意に基づくサーバ自身による生成も可能である。

[0093] <S302>

S302において、サーバ100は、疑似ランダム関数を用いて、2つのアドレス値 $[[addr^0]] \leftarrow PRF([k], [s_0])$, $[[addr^1]] \leftarrow PRF([k], [s_1])$ を計算し、これを復元して $addr^0$, $addr^1$ を得る。

[0094] <S303>

S303において、サーバ100は、データ格納部140に格納されているハッシュテーブルから $addr^0$, $addr^1$ に対応する $2Z$ 個の要素を取り出す。すなわち、ハッシュテーブルを配列 $[[Table]] = ([[a_0]], \dots, [[a_{BZ-1}]])$ と読み替えた場合に、 $[[a]] = ([[a_{Z \times addr^0}], \dots, [[a_{Z \times addr^0 + Z - 1}]], [[a_{Z \times addr^1}], \dots, [[a_{Z \times addr^1 + Z - 1}]])$ を取得する。

[0095] <S304>

S304において、サーバ100は、 $[[a]]$ の各データ $[[a_j]]$ について、キーの比較を行い $[[c]] = ([[c_0]], \dots, [[c_{2Z-1}]])$; $[[c_j]] = [[k_j]] = ?[[k]]$ を計算する。

[0096] <S305>

S305において、サーバ100は、内積 $[[a]] \leftarrow <[[a]], [[c]]>$ を計算し、これをユーザ端末300へ返送するか、又はサーバ間合意に基づき復元する。あるいは、これを復元せず、全く別の秘密計算処理に用いてもよい。

[0097] 本実施例3において、ユーザ・サーバ間の通信量 $O(\log n)$ ビットで

あり、サーバ間通信量は $O(Z \log n)$ ビットである。

[0098] (実施例4)

本実施例4では、実施例1、2で構築した秘匿ハッシュテーブルでのデータ削除方法を説明する。前提として、サーバ100は、実施例1ないし2で構築されたハッシュテーブル(サイズ $B \times Z$)と秘密鍵 $[[s_0]]$, $[[s_1]]$ をデータ記憶部140に保持しているとする。また、ユーザは削除したいデータのキー k を持っているとするが、これはサーバ同士の合意に基づいてサーバ自身で選定しても構わない。以下、図7のフローチャートに示す手順に沿って説明する。

[0099] <S401>

S401において、ユーザ端末300は、サーバ100に対し、削除したいキーの秘匿値 $[[k]]$ を送る。ただし、この秘匿キー値はユーザ端末300からサーバ100への要求だけでなく、サーバ間合意に基づくサーバ自身による生成も可能である。

[0100] <S402>

S402において、サーバ100は、疑似ランダム関数を用いて、2つのアドレス値 $[[addr^0]] \leftarrow PRF([k], [s_0])$, $[[addr^1]] \leftarrow PRF([k], [s_1])$ を計算し、これを復元して $addr^0$, $addr^1$ を得る。

[0101] <S403>

S403において、サーバ100は、データ格納部140に保持しているハッシュテーブルから $addr^0$, $addr^1$ に対応する $2Z$ 個の要素を取り出す。すなわち、ハッシュテーブルを配列 $[[Table]] = ([[a_0]], \dots, [[a_{BZ-1}]])$ と読み替えた場合に、 $[[a]] = ([[a_{Z \times addr^0}], \dots, [[a_{Z \times addr^0 + Z - 1}]], [[a_{Z \times addr^1}], \dots, [[a_{Z \times addr^1 + Z - 1}]])$ を取得する。

[0102] <S404>

S404において、サーバ100は、 $[[a]]$ の各データ $[[a_j]]$ に

ついて、キーの比較に基づきデータ（ $[[k]]$ ， $[[e]]$ ）の削除処理を行う。すなわち、全ての j に関して $[[k_j]] = [[k_j]] + [[k_j]] = ? [[k]] \times (dummy - [[k_j]])$ ， $[[e_j]] = [[e_j]] + [[k_j]] = ? [[k]]$ を計算する。

[0103] <S 4 0 5>

最後に、サーバ100は $[[a]]$ の各要素をハッシュテーブルの元の位置に上書きする。

本実施例4の通信量は実施例3と等価である。

[0104] （実施例5）

本実施例5では、実施例1、2で構築したハッシュテーブルの解体・全データの抽出方法について説明する。前提として、サーバ100は実施例1ないし2で構築されたハッシュテーブル（サイズ $B \times Z$ ）をデータ格納部140に保持しているとする。以下、図8のフローチャートに示す手順に沿って説明する。

[0105] <S 5 0 1>

S 5 0 1において、サーバ100は、ハッシュテーブルの全データを、そのフラグに基づき昇順でソートする。すなわち、テーブルを配列 $[[Table]] = ([[a_0]], \dots, [[a_{BZ-1}]])$ と読み替え、またテーブルからフラグのみを抽出した配列を $[[E]] = ([[e_0]], \dots, [[e_{BZ-1}]])$ とした場合に、 $[[Table]] = \text{Sort}([[[E]], [[Table]])$ とする。

[0106] <S 5 0 2>

S 5 0 2において、サーバ100は、 $[[Table]]$ の先頭 n データを配列 $[[A]]$ とし、残りを削除する。

[0107] 本実施例では、1ビットの情報に基づくソート1回のみ行うため、サーバ間通信量は $O(n \log n)$ となる。

[0108] （実施の形態の効果）

本実施の形態によれば、秘密計算を用いることでユーザとの通信を介さず

秘匿ハッシュテーブルを構築できる。その際、ランク付けという秘密計算処理を利用することで、テーブル構築に必要な秘匿ソートの回数を大きく減らし、従来法よりも通信コストを大きく削減できる。

[0109] すなわち、秘匿ハッシュテーブルの構築において、各データに2通りのアドレス値を割り振ることでテーブルサイズを小さくするという従来法のメリットをそのままに、サーバ間通信量を $\omega(n \log^{2.5} n)$ から $O(n \log^2 n)$ へと削減できる。構築済みのテーブルへの操作も可能である。

[0110] (付記)

本明細書には、少なくとも下記各項の秘匿ハッシュテーブル構築装置、秘匿ハッシュテーブル構築システム、秘匿ハッシュテーブル構築方法、及びプログラムが開示されている。

(第1項)

キーと、データがダミーか否かを示すフラグとを有するデータを複数個含む実データ列から、秘密計算により、B通りのアドレス値にそれぞれ最大Z個までのデータを格納できる秘匿ハッシュテーブルを構築する秘匿ハッシュテーブル構築装置であって、

前記実データ列の各データに対する、第1アドレス値及び第2アドレス値と、フラグと、アドレス値毎のランクとを含む格納先データ配列と、ダミーとしての、第1アドレス値及び第2アドレス値と、フラグと、アドレス値毎のランクとを含む格納先データ配列とを連結した第1配列を生成し、

前記実データ列とダミーのデータ列とを連結した第2配列を生成し、

前記第1配列における第1アドレス値とフラグに基づいて、前記第1配列と前記第2配列のそれぞれをソートし、ソートした前記第1配列の第1アドレス値に対して同一データに対するランク付け操作を行い、当該ランク付けに基づいて前記第1配列と前記第2配列のそれぞれをソートし、

前記第1配列における第2アドレス値とフラグに基づいて、前記第1配列と前記第2配列のそれぞれをソートし、ソートした前記第1配列の第2アドレス値に対して同一データに対するランク付け操作を行い、

前記第 1 配列において、第 1 アドレス値と第 2 アドレス値のうち、ランクの低い側のアドレス値を抽出し、抽出したアドレス値とフラグを有する第 3 配列を生成し、

前記第 3 配列におけるアドレス値から計算されたランク配列の各要素と Z との比較により得られた第 4 配列を用いて、前記第 2 配列をソートし、ソートした前記第 2 配列における B Z 個の要素を前記秘匿ハッシュテーブルとして出力する

演算部を備える秘匿ハッシュテーブル構築装置。

(第 2 項)

前記演算部は、前記実データ列における各データと前記ダミーのデータ列における各データとを示すタグを有するタグの列を使用することにより、前記実データ列と前記ダミーのデータ列とを連結した前記第 2 配列に対して、前記第 4 配列に基づくソート以外のソートを行うことなく、前記秘匿ハッシュテーブルを生成する

第 1 項に記載の秘匿ハッシュテーブル構築装置。

(第 3 項)

前記演算部は、アクセス対象のキー値から 2 つのアドレス値を計算し、前記秘匿ハッシュテーブルから、前記 2 つのアドレス値に対応する 2 Z 個のデータを取得し、2 Z 個のデータのうち、前記キー値と同一のキー値を有するデータを返す

第 1 項又は第 2 項に記載の秘匿ハッシュテーブル構築装置。

(第 4 項)

前記演算部は、削除対象のキー値から 2 つのアドレス値を計算し、前記秘匿ハッシュテーブルから、前記 2 つのアドレス値に対応する 2 Z 個のデータを取得し、2 Z 個のデータのデータのうち、前記キー値と同一のキー値を有するデータを削除する

第 1 項ないし第 3 項のうちいずれか 1 項に記載の秘匿ハッシュテーブル構築装置。

(第5項)

前記演算部は、前記秘匿ハッシュテーブルにおける全データを、そのフラグに基づいてソートし、先頭の所定個数のデータを前記実データ列として取得する

第1項ないし第4項のうちいずれか1項に記載の秘匿ハッシュテーブル構築装置。

(第6項)

キーと、データがダミーか否かを示すフラグとを有するデータを複数個含む実データ列から、秘密計算により、B通りのアドレス値にそれぞれ最大Z個までのデータを格納できる秘匿ハッシュテーブルを構築する秘匿ハッシュテーブル構築システムであって、

前記実データ列の各データに対する、第1アドレス値及び第2アドレス値と、フラグと、アドレス値毎のランクとを含む格納先データ配列と、ダミーとしての、第1アドレス値及び第2アドレス値と、フラグと、アドレス値毎のランクとを含む格納先データ配列とを連結した第1配列を生成し、

前記実データ列とダミーのデータ列とを連結した第2配列を生成し、

前記第1配列における第1アドレス値とフラグに基づいて、前記第1配列と前記第2配列のそれぞれをソートし、ソートした前記第1配列の第1アドレス値に対して同一データに対するランク付け操作を行い、当該ランク付けに基づいて前記第1配列と前記第2配列のそれぞれをソートし、

前記第1配列における第2アドレス値とフラグに基づいて、前記第1配列と前記第2配列のそれぞれをソートし、ソートした前記第1配列の第2アドレス値に対して同一データに対するランク付け操作を行い、

前記第1配列において、第1アドレス値と第2アドレス値のうち、ランクの低い側のアドレス値を抽出し、抽出したアドレス値とフラグを有する第3配列を生成し、

前記第3配列におけるアドレス値から計算されたランク配列の各要素とZとの比較により得られた第4配列を用いて、前記第2配列をソートし、ソー

トした前記第2配列におけるB Z個の要素を前記秘匿ハッシュテーブルとして出力する

演算部を備える秘匿ハッシュテーブル構築システム。

(第7項)

キーと、データがダミーか否かを示すフラグとを有するデータを複数個含む実データ列から、秘密計算により、B通りのアドレス値にそれぞれ最大Z個までのデータを格納できる秘匿ハッシュテーブルを構築する秘匿ハッシュテーブル構築システムが実行する秘匿ハッシュテーブル構築方法であって、

前記実データ列の各データに対する、第1アドレス値及び第2アドレス値と、フラグと、アドレス値毎のランクとを含む格納先データ配列と、ダミーとしての、第1アドレス値及び第2アドレス値と、フラグと、アドレス値毎のランクとを含む格納先データ配列とを連結した第1配列を生成するステップと、

前記実データ列とダミーのデータ列とを連結した第2配列を生成するステップと、

前記第1配列における第1アドレス値とフラグに基づいて、前記第1配列と前記第2配列のそれぞれをソートし、ソートした前記第1配列の第1アドレス値に対して同一データに対するランク付け操作を行い、当該ランク付けに基づいて前記第1配列と前記第2配列のそれぞれをソートするステップと、

前記第1配列における第2アドレス値とフラグに基づいて、前記第1配列と前記第2配列のそれぞれをソートし、ソートした前記第1配列の第2アドレス値に対して同一データに対するランク付け操作を行うステップと、

前記第1配列において、第1アドレス値と第2アドレス値のうち、ランクの低い側のアドレス値を抽出し、抽出したアドレス値とフラグを有する第3配列を生成するステップと、

前記第3配列におけるアドレス値から計算されたランク配列の各要素とZとの比較により得られた第4配列を用いて、前記第2配列をソートし、ソー

トした前記第2配列におけるB Z個の要素を前記秘匿ハッシュテーブルとして出力するステップと

を備える秘匿ハッシュテーブル構築方法。

(第8項)

コンピュータを、第1項ないし第5項のうちいずれか1項に記載の秘匿ハッシュテーブル構築装置における演算部として機能させるためのプログラム。

[0111] 以上、本実施の形態について説明したが、本発明はかかる特定の実施形態に限定されるものではなく、特許請求の範囲に記載された本発明の要旨の範囲内において、種々の変形・変更が可能である。

[0112] [参考文献]

[1] T-H. H. Chan, Y. Guo, W-K. Lin, and E. Shi. Oblivious hashing revisited, and applications to asymptotically efficient ORAM and OPRAM. Cryptology ePrint Archive, Report 2017/924, 2017.

[2] K. Chida, K. Hamada, D. Ikarashi, R. Kikuchi, and B. Pinkas. High-throughput secure AES computation. In WAHC@CCS 2018, pages 13-24, 2018.

[3] K. Chida, K. Hamada, D. Ikarashi, R. Kikuchi, N. Kiribuchi, B. Pinkas. Anefficient secure threeparty sorting protocol with an honest majority. Cryptology ePrint Archive, Report 2019/695 (2019), <https://eprint.iacr.org/2019/695>

[4] O. Goldreich and R. Ostrovsky. Software protection and simulation on oblivious RAMs. J. ACM, 43(3):431-473, May 1996. 8

[5] M. Ito, A. Saito, and T. Nishizeki. Secret sharing schemes realizing general access structures. Proceedings of the IEEE Global Telecommunication Conference, Globecom 87, pp. 99-102, 1987.

[6] A. Shamir. How to share a secret. Commun. ACM, Vol. 22, No. 11, pp. 612-613, 1979.

[7] 桐淵直人, 五十嵐大, 濱田浩気, 菊池亮. プログラマブルな秘密計算ライブラリMEVAL3. 暗号と情報セキュリティシンポジウム(SCIS)2018 予稿集(2018).

符号の説明

- [0113] 1 0 0 サーバ
1 1 0 入力部
1 2 0 演算部
1 3 0 出力部
1 4 0 データ格納部
2 0 0 通信ネットワーク
3 0 0 ユーザ端末
1 0 0 0 ドライブ装置
1 0 0 1 記録媒体
1 0 0 2 補助記憶装置
1 0 0 3 メモリ装置
1 0 0 4 CPU
1 0 0 5 インタフェース装置
1 0 0 6 表示装置
1 0 0 7 入力装置

請求の範囲

[請求項1]

キーと、データがダミーか否かを示すフラグとを有するデータを複数個含む実データ列から、秘密計算により、 B 通りのアドレス値にそれぞれ最大 Z 個までのデータを格納できる秘匿ハッシュテーブルを構築する秘匿ハッシュテーブル構築装置であって、

前記実データ列の各データに対する、第1アドレス値及び第2アドレス値と、フラグと、アドレス値毎のランクとを含む格納先データ配列と、ダミーとしての、第1アドレス値及び第2アドレス値と、フラグと、アドレス値毎のランクとを含む格納先データ配列とを連結した第1配列を生成し、

前記実データ列とダミーのデータ列とを連結した第2配列を生成し、

前記第1配列における第1アドレス値とフラグに基づいて、前記第1配列と前記第2配列のそれぞれをソートし、ソートした前記第1配列の第1アドレス値に対して同一データに対するランク付け操作を行い、当該ランク付けに基づいて前記第1配列と前記第2配列のそれぞれをソートし、

前記第1配列における第2アドレス値とフラグに基づいて、前記第1配列と前記第2配列のそれぞれをソートし、ソートした前記第1配列の第2アドレス値に対して同一データに対するランク付け操作を行い、

前記第1配列において、第1アドレス値と第2アドレス値のうち、ランクの低い側のアドレス値を抽出し、抽出したアドレス値とフラグを有する第3配列を生成し、

前記第3配列におけるアドレス値から計算されたランク配列の各要素と Z との比較により得られた第4配列を用いて、前記第2配列をソートし、ソートした前記第2配列における BZ 個の要素を前記秘匿ハッシュテーブルとして出力する

演算部を備える秘匿ハッシュテーブル構築装置。

[請求項2] 前記演算部は、前記実データ列における各データと前記ダミーのデータ列における各データとを示すタグを有するタグの列を使用することにより、前記実データ列と前記ダミーのデータ列とを連結した前記第2配列に対して、前記第4配列に基づくソート以外のソートを行うことなく、前記秘匿ハッシュテーブルを生成する

請求項1に記載の秘匿ハッシュテーブル構築装置。

[請求項3] 前記演算部は、アクセス対象のキー値から2つのアドレス値を計算し、前記秘匿ハッシュテーブルから、前記2つのアドレス値に対応する2Z個のデータを取得し、2Z個のデータのうち、前記キー値と同一のキー値を有するデータを返す

請求項1又は2に記載の秘匿ハッシュテーブル構築装置。

[請求項4] 前記演算部は、削除対象のキー値から2つのアドレス値を計算し、前記秘匿ハッシュテーブルから、前記2つのアドレス値に対応する2Z個のデータを取得し、2Z個のデータのデータのうち、前記キー値と同一のキー値を有するデータを削除する

請求項1ないし3のうちいずれか1項に記載の秘匿ハッシュテーブル構築装置。

[請求項5] 前記演算部は、前記秘匿ハッシュテーブルにおける全データを、そのフラグに基づいてソートし、先頭の所定個数のデータを前記実データ列として取得する

請求項1ないし4のうちいずれか1項に記載の秘匿ハッシュテーブル構築装置。

[請求項6] キーと、データがダミーか否かを示すフラグとを有するデータを複数個含む実データ列から、秘密計算により、B通りのアドレス値にそれぞれ最大Z個までのデータを格納できる秘匿ハッシュテーブルを構築する秘匿ハッシュテーブル構築システムであって、

前記実データ列の各データに対する、第1アドレス値及び第2アド

レス値と、フラグと、アドレス値毎のランクとを含む格納先データ配列と、ダミーとしての、第1アドレス値及び第2アドレス値と、フラグと、アドレス値毎のランクとを含む格納先データ配列とを連結した第1配列を生成し、

前記実データ列とダミーのデータ列とを連結した第2配列を生成し、

前記第1配列における第1アドレス値とフラグに基づいて、前記第1配列と前記第2配列のそれぞれをソートし、ソートした前記第1配列の第1アドレス値に対して同一データに対するランク付け操作を行い、当該ランク付けに基づいて前記第1配列と前記第2配列のそれぞれをソートし、

前記第1配列における第2アドレス値とフラグに基づいて、前記第1配列と前記第2配列のそれぞれをソートし、ソートした前記第1配列の第2アドレス値に対して同一データに対するランク付け操作を行い、

前記第1配列において、第1アドレス値と第2アドレス値のうち、ランクの低い側のアドレス値を抽出し、抽出したアドレス値とフラグを有する第3配列を生成し、

前記第3配列におけるアドレス値から計算されたランク配列の各要素とZとの比較により得られた第4配列を用いて、前記第2配列をソートし、ソートした前記第2配列におけるB Z個の要素を前記秘匿ハッシュテーブルとして出力する

演算部を備える秘匿ハッシュテーブル構築システム。

[請求項7]

キーと、データがダミーか否かを示すフラグとを有するデータを複数個含む実データ列から、秘密計算により、B通りのアドレス値にそれぞれ最大Z個までのデータを格納できる秘匿ハッシュテーブルを構築する秘匿ハッシュテーブル構築システムが実行する秘匿ハッシュテーブル構築方法であって、

前記実データ列の各データに対する、第1アドレス値及び第2アドレス値と、フラグと、アドレス値毎のランクとを含む格納先データ配列と、ダミーとしての、第1アドレス値及び第2アドレス値と、フラグと、アドレス値毎のランクとを含む格納先データ配列とを連結した第1配列を生成するステップと、

前記実データ列とダミーのデータ列とを連結した第2配列を生成するステップと、

前記第1配列における第1アドレス値とフラグに基づいて、前記第1配列と前記第2配列のそれぞれをソートし、ソートした前記第1配列の第1アドレス値に対して同一データに対するランク付け操作を行い、当該ランク付けに基づいて前記第1配列と前記第2配列のそれぞれをソートするステップと、

前記第1配列における第2アドレス値とフラグに基づいて、前記第1配列と前記第2配列のそれぞれをソートし、ソートした前記第1配列の第2アドレス値に対して同一データに対するランク付け操作を行うステップと、

前記第1配列において、第1アドレス値と第2アドレス値のうち、ランクの低い側のアドレス値を抽出し、抽出したアドレス値とフラグを有する第3配列を生成するステップと、

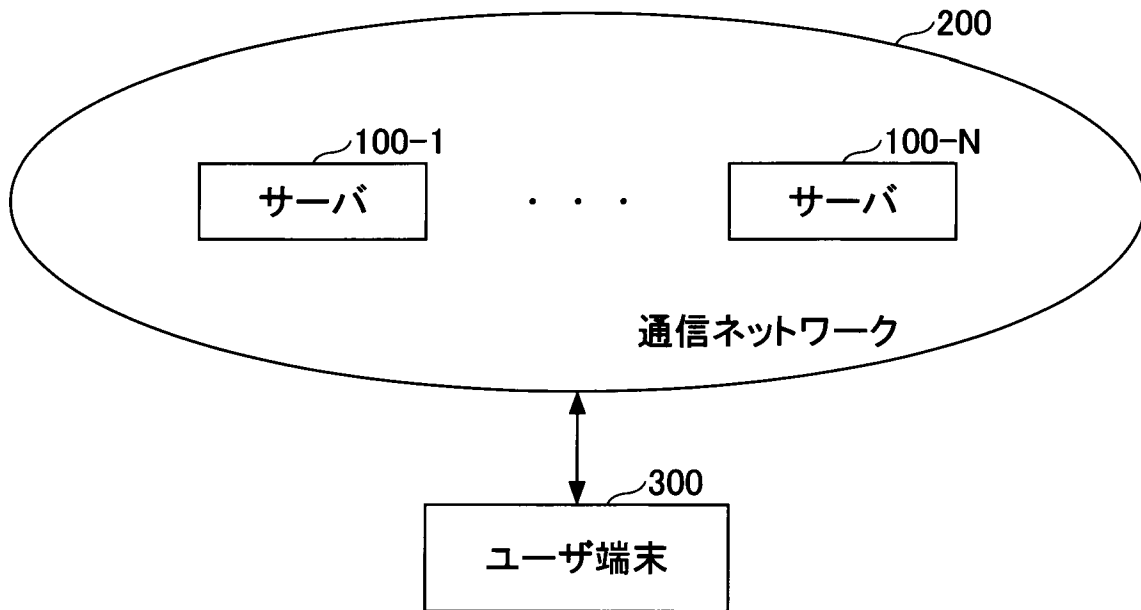
前記第3配列におけるアドレス値から計算されたランク配列の各要素とZとの比較により得られた第4配列を用いて、前記第2配列をソートし、ソートした前記第2配列におけるB Z個の要素を前記秘匿ハッシュテーブルとして出力するステップと

を備える秘匿ハッシュテーブル構築方法。

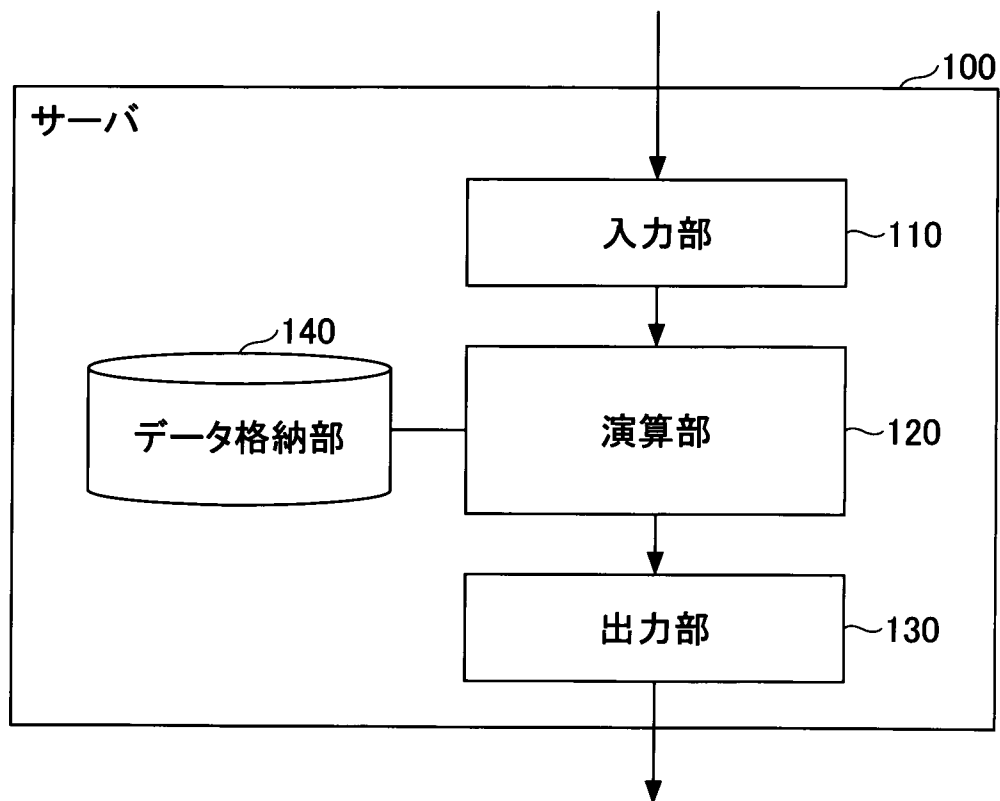
[請求項8]

コンピュータを、請求項1ないし5のうちいずれか1項に記載の秘匿ハッシュテーブル構築装置における演算部として機能させるためのプログラム。

[図1]



[図2]

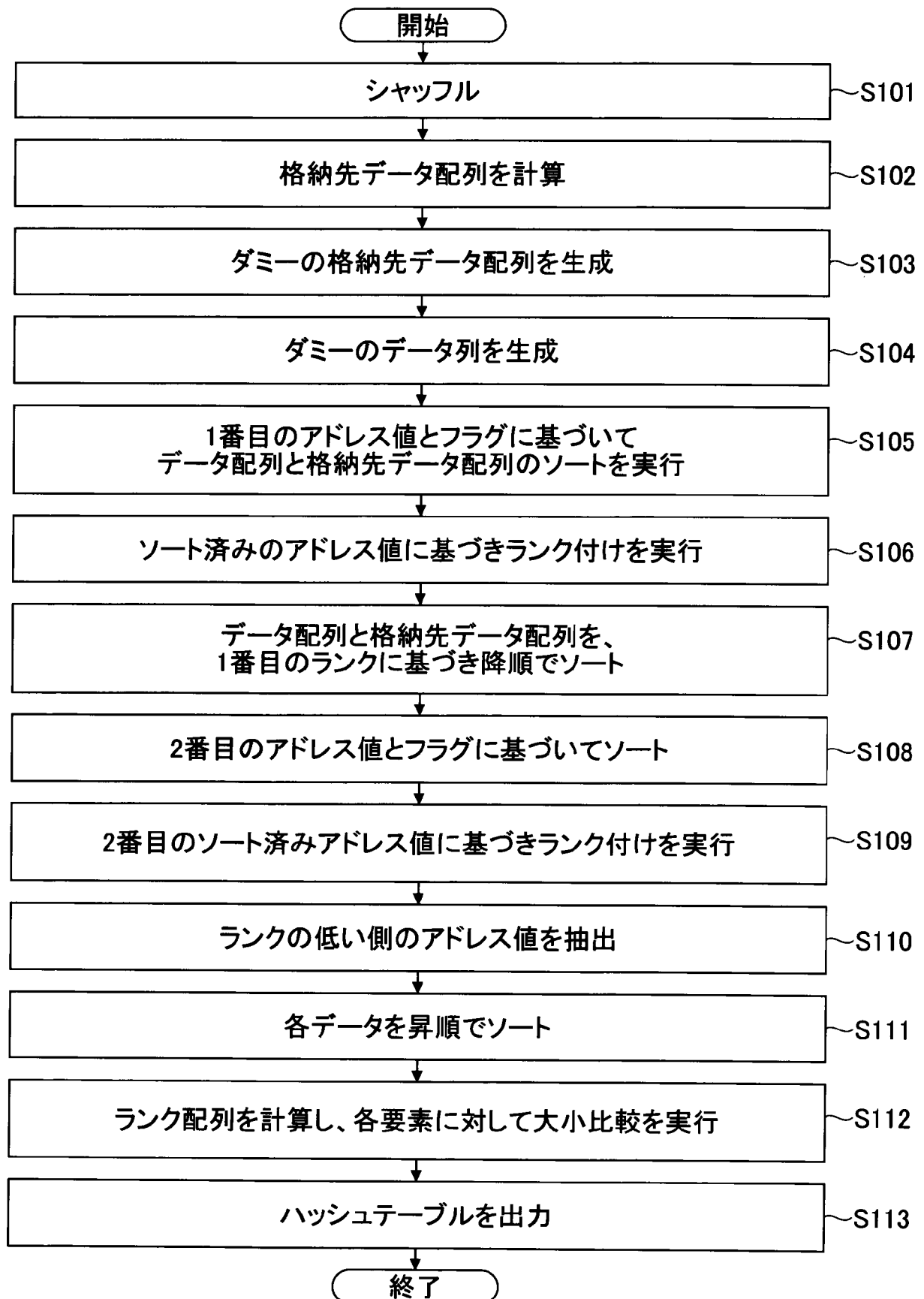


The diagram illustrates a computer system 1000 and its connection to a recording medium 1001. The computer system 1000 is connected to the recording medium 1001 via a bus B. The components of the computer system 1000 are:

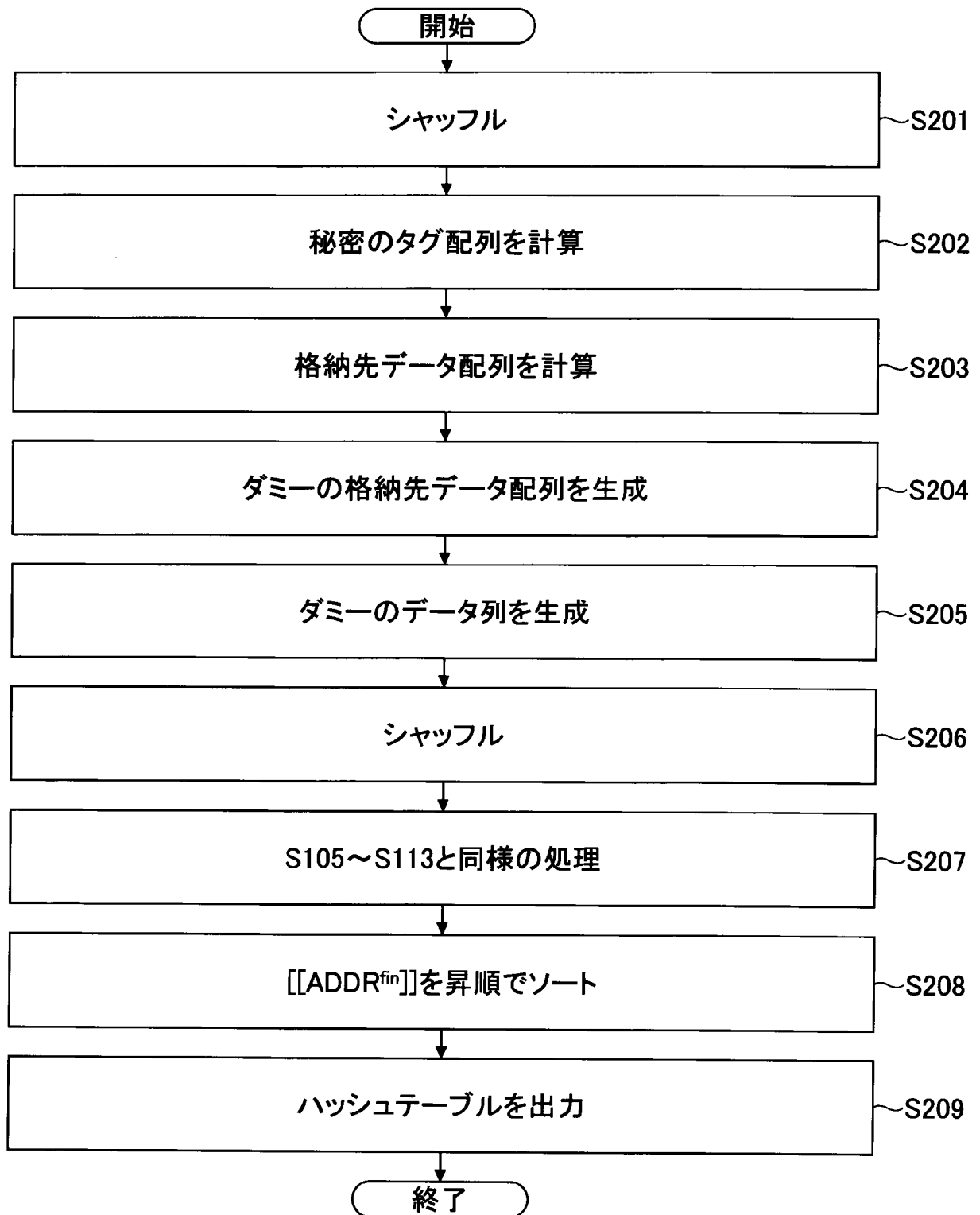
- 1004 CPU
- 1005 インタフェース装置 (Interface Device)
- 1006 表示装置 (Display Device)
- 1007 入力装置 (Input Device)
- 1000 ドライブ装置 (Drive Device)
- 1002 補助記憶装置 (Auxiliary Storage Device)
- 1003 メモリ装置 (Memory Device)
- 1008 出力装置 (Output Device)

The recording medium 1001 is labeled 記録媒体 (Recording Medium).

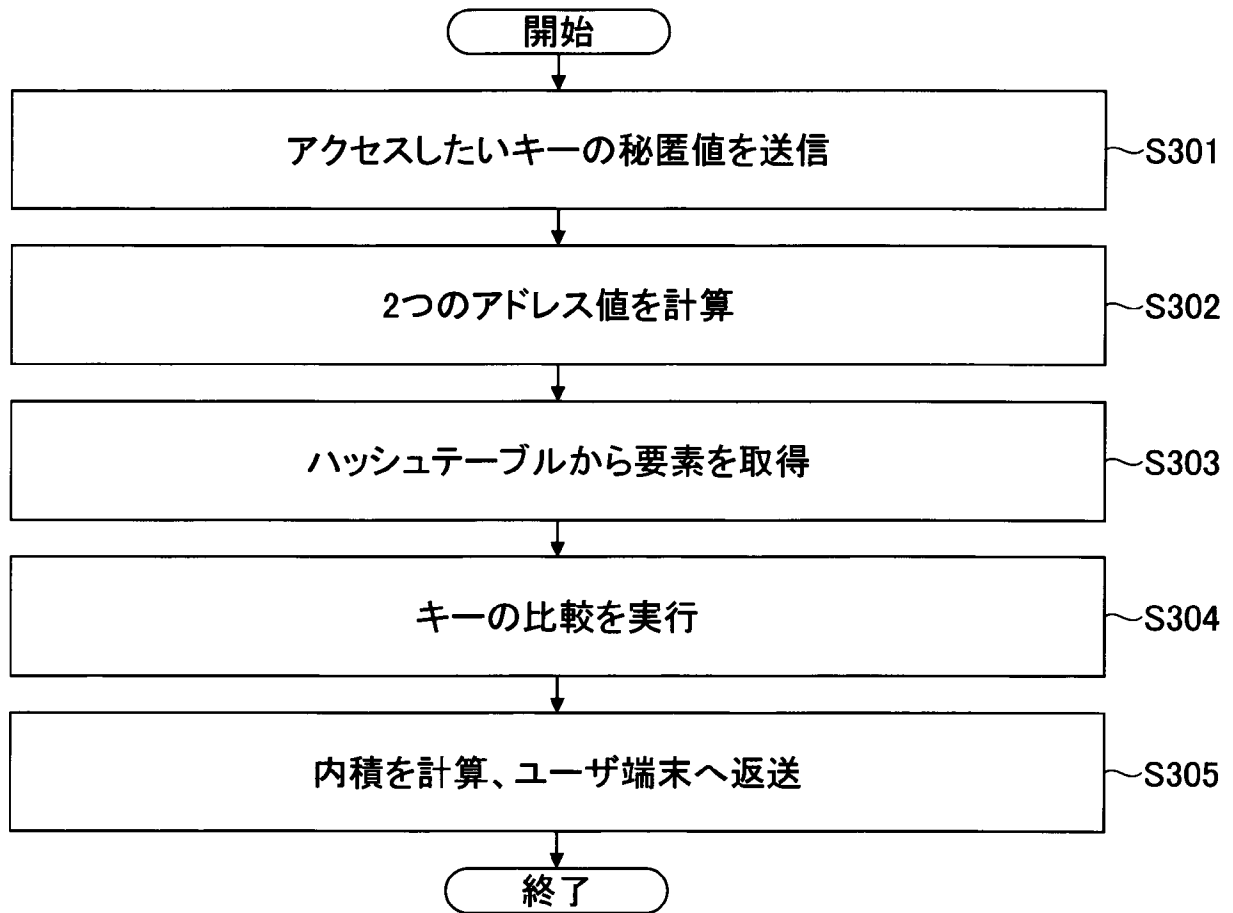
[図4]



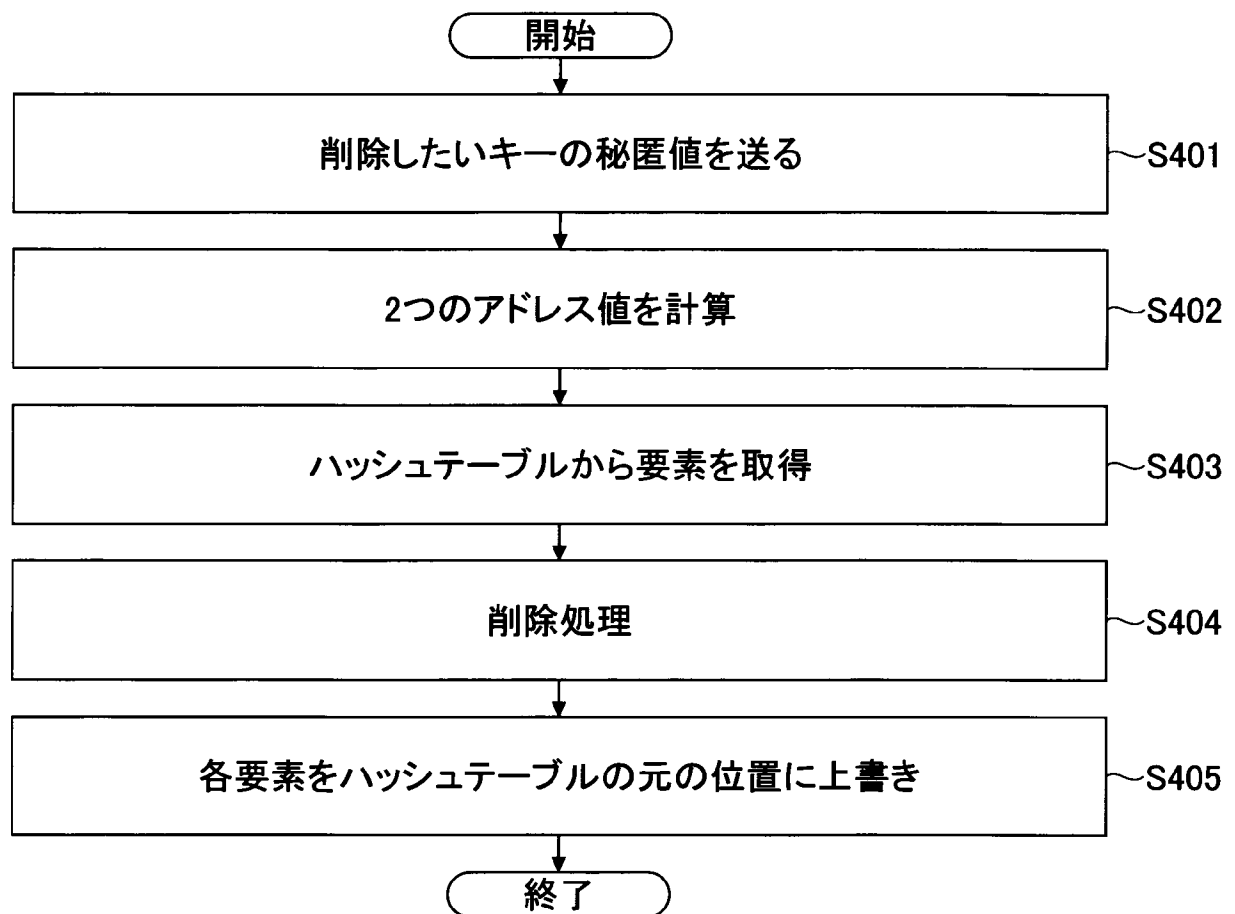
[図5]



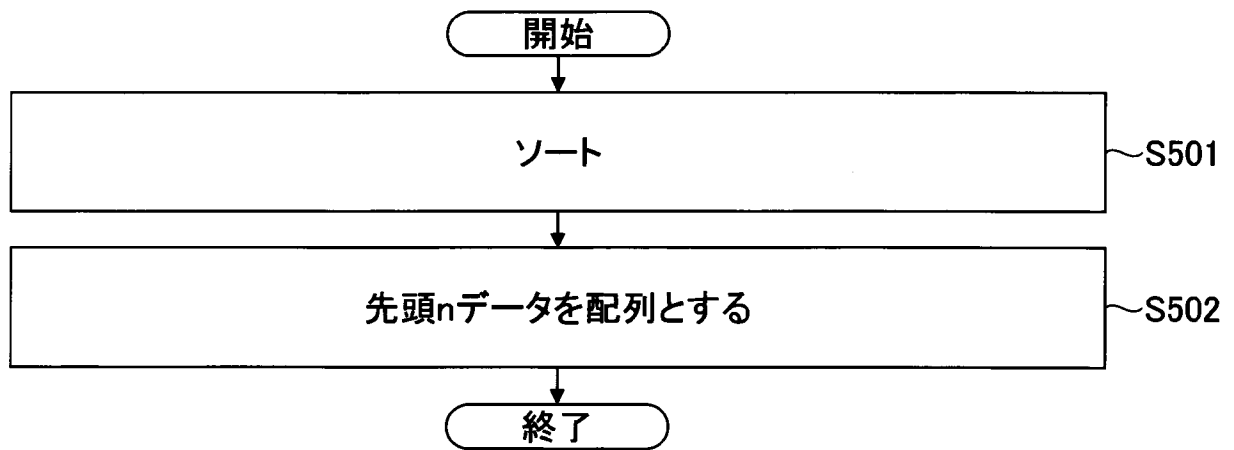
[図6]



[図7]



[図8]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2020/046125

A. CLASSIFICATION OF SUBJECT MATTER

Int. Cl. G09C1/00 (2006.01) i

FI: G09C1/00 650Z

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 1971-2021

Registered utility model specifications of Japan 1996-2021

Published registered utility model applications of Japan 1994-2021

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	CHAN, T-H. H. et al. Perfectly Secure Oblivious Parallel RAM, Cryptology ePrint Archive: Report 2018/364, Ver. 20181002:155559, 02 October 2018, pp. 1-54, [online], [retrieved on 26 March 2021], Internet: URL:https://eprint.iacr.org/2018/364, in particular, sections 4-7	1-8
A	KUSHILEVITZ, E. et al. Sub-logarithmic Distributed Oblivious RAM with Small Block Size, v3, 17 November 2018, pp. 1-42, [online], [retrieved on 31 January 2020], Internet: URL:https://arxiv.org/abs/1802.05145, in particular, section 5	1-8

☒ 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
26.03.2021

Date of mailing of the international search report
06.04.2021

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/JP2020/046125

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	ASHAROV, C. et al. OptORAMa: Optimal Oblivious RAM, Cryptology ePrint Archive: Report 2018/892, Ver.20191012:174809, 12 October 2019, pp. 1-71, [online], [retrieved on 31 January 2020], Internet: URL: https://eprint.iacr.org/2018/892 , in particular, sections 5-9	1-8

A. 発明の属する分野の分類（国際特許分類（IPC）） G09C 1/00(2006.01)i FI: G09C1/00 650Z														
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）） 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 <table border="0"> <tr> <td>日本国実用新案公報</td> <td>1922 - 1996年</td> </tr> <tr> <td>日本国公開実用新案公報</td> <td>1971 - 2021年</td> </tr> <tr> <td>日本国実用新案登録公報</td> <td>1996 - 2021年</td> </tr> <tr> <td>日本国登録実用新案公報</td> <td>1994 - 2021年</td> </tr> </table> 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）			日本国実用新案公報	1922 - 1996年	日本国公開実用新案公報	1971 - 2021年	日本国実用新案登録公報	1996 - 2021年	日本国登録実用新案公報	1994 - 2021年				
日本国実用新案公報	1922 - 1996年													
日本国公開実用新案公報	1971 - 2021年													
日本国実用新案登録公報	1996 - 2021年													
日本国登録実用新案公報	1994 - 2021年													
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	CHAN, T-H. H. et al., Perfectly Secure Oblivious Parallel RAM, Cryptology ePrint Archive: Report 2018/364, Ver. 20181002:155559, 2018.10.02, pp.1-54, [online], [令和3年3月26日検索], インターネット<URL:https://eprint.iacr.org/2018/364> 特に、4-7節	1-8												
A	KUSHILEVITZ, E. et al., Sub-logarithmic Distributed Oblivious RAM with Small Block Size, v3, 2018.11.17, pp.1-42, [online], [令和2年1月31日検索], インターネット<URL:https://arxiv.org/abs/1802.05145> 特に、5節	1-8												
A	ASHAROV, G. et al., OptORAMA: Optimal Oblivious RAM, Cryptology ePrint Archive: Report 2018/892, Ver. 20191012:174809, 2019.10.12, pp.1-71, [online], [令和2年1月31日検索], インターネット<URL:https://eprint.iacr.org/2018/892> 特に、5-9節	1-8												
☐ 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。														
<table border="0"> <tr> <td>* 引用文献のカテゴリー</td> <td>“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抵触するものではなく、発明の原理又は理論の理解のために引用するもの</td> </tr> <tr> <td>“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</td> <td>“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</td> </tr> <tr> <td>“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</td> <td>“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</td> </tr> <tr> <td>“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）</td> <td>“&” 同一パテントファミリー文献</td> </tr> <tr> <td>“O” 口頭による開示、使用、展示等に言及する文献</td> <td></td> </tr> <tr> <td>“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</td> <td></td> </tr> </table>			* 引用文献のカテゴリー	“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抵触するものではなく、発明の原理又は理論の理解のために引用するもの	“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	“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	“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	“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	“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）	“&” 同一パテントファミリー文献	“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	
* 引用文献のカテゴリー	“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抵触するものではなく、発明の原理又は理論の理解のために引用するもの													
“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	“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													
“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	“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													
“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）	“&” 同一パテントファミリー文献													
“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														
国際調査を完了した日 26.03.2021	国際調査報告の発送日 06.04.2021													
名称及びあて先 日本国特許庁(ISA/JP) 〒100-8915 日本国 東京都千代田区霞が関三丁目4番3号	権限のある職員（特許庁審査官） 行田 悦資 5S 6304 電話番号 03-3581-1101 内線 3546													