Vol.2015-CSEC-69 No.10
Vol.2015-IOT-29 No.10
2015/5/21

サーバー台数を限定しない秘密分散法を用いた秘匿検索法の提案

中原将貴[†1]　　岩村惠市[†1]

データを秘匿したまま検索を行う方式に検索可能暗号がある.一方,秘密分散を用いた検索可能秘密分散[1]も第 64 回 CSEC 研究会で発表されたが,これはタグを用いて検索を行うため,秘密分散された値を直接検索することはできない.また,秘密分散された値を直接検索できる方式として SCIS2011,ISEC2014 で発表された方式がある[2][3].しかし,いずれの方式も,3 台のサーバーに分散し,2 台のサーバーで復元する場合に限定されている.本論文ではサーバー台数や復元の閾値を限定せず分散,検索が行え,秘密分散された値を直接検索する方式を提案する.すなわち,n 台のサーバーに分散し,k 台(k≦n)のサーバーで直接秘密情報の検索および復元ができる.

Proposal of confidential search method using a secret sharing scheme that does not limit the number of servers

MASAKI NAKAHARA[†1]　KEIICHI IWAMURA[†1]

There is searchable encryption in the method of performing search data while concealing. On the other hand, the searchable secret sharing using secret sharing [1] was presented at 64th CSEC, but this can not search for secret sharing values directly because it performs a search using the tag. On the other hand, the method that allows you to search the secret sharing values directly was presented at SCIS2011, ISEC2014 [2] [3]. However, both method have been limited number of servers to use to three in the case of dispersion and to two in the case of restoration. In this paper, I propose the method that does not limit the number of servers to be used for dispersion and restore and can search for the secret sharing values directly. In other words, it was dispersed in n servers and you can search and restore direct secret information on k servers (k<n).

1　はじめに

近年クラウドコンピューティングを利用するサービスが数多く見受けられる.クラウドコンピューティングでは遠隔地のサーバーにデータを保存し,データの処理も遠隔地で行う.そのため,データが漏洩せず安全にデータの通信や処理を行えることが求められている.その方法の一つとして検索可能暗号が存在する.

検索可能暗号とは,情報を暗号化した状態で検索を可能とする暗号である.最初の実用的な検索可能暗号は,2000 年に Song ら[4]によって提案された.検索可能暗号には共通鍵暗号を用いた方式[5]と公開鍵暗号を用いた方式[6]があり,Song らの方式は共通鍵暗号を用いたものである.共通鍵暗号系では,秘密鍵を持つユーザーのみが暗号化と検索を行える.しかし,秘密鍵がないと検索が行えないため,データの所有者であるオーナーと検索者であるユーザーが同一であるかユーザーに秘密鍵を与えなければならない.一方,公開鍵暗号系では,公開鍵で暗号化を行い,秘密鍵で検索を行うため,データの暗号化はだれでも行えるが,検索は特定のユーザーしか行えない.また検索における計算量が多い.

このように暗号化することでデータを秘匿化したまま検索を行えるが,クラウドコンピューティングでは秘密分散法を用いることも提案されており,それを用いた秘匿演算なども提案されている.秘密分散法は軽い計算量で秘匿演算ができることが知られており,秘密分散法を用いて秘匿検索が行えれば,秘密情報を秘匿したまま秘匿検索と秘匿演算を同時に実現できる広い応用が期待できる.秘密分散法とは,1 つの情報を複数の異なる情報に変換し,そのうちの一定数以上を集めることで元の情報を復元することが出来る.一定数未満では元の情報を復元することができないため,データの一部が漏洩してしまっても情報が漏洩することがない.これにより,災害等でデータの一部が失われても一定数のデータが無事ならば元の情報が復元でき,一定数未満の漏洩では情報流出が起こらない安全なデータの運用を行える.これは,クラウドコンピューティングに必要な要素である.

そこで本論文では,秘密分散を用いて保存したデータを,秘匿したままで検索できる方式を提案する.提案方式では,秘密分散したデータから直接検索を行う.ただし,秘密分散を用いた検索可能秘密分散[1]も伊藤らによって発表されたが,これはタグを用いて検索を行うため,秘密分散された値を直接検索することはできない.一方,秘密分散された値を直接検索できる方式として只木らによって発表された方式がある[3].しかし,この方式は,3 台のサーバーに分散し,2 台のサーバーで復元する場合に限定されている.本論文ではサーバー台数や復元の閾値を限定せず分散,検索が行え,秘密分散された値を直接検索する方式を提案する.すなわち,n 台のサーバーに分散し,k 台(k≦n)のサーバーで直接秘密情報の検索および復元ができる.

†1 東京理科大学
Tokyo University of Science

Vol.2015-CSEC-69 No.10
Vol.2015-IOT-29 No.10
2015/5/21

本論文の構成を以下に示す.第 2 章では,伊藤らが提案した方式[1],只木らが提案した方式[3]について説明する.その後第 3 章で,今回の提案方式について説明し,第 4 章で,それぞれの比較を記述する.最後に第 5 章で本論文のまとめを行う.

2 従来方式

本章では,タグを用いた検索可能秘密分散法として伊藤らの方式,サーバー台数が限定された方式として只木らの方式について示す.なお,これ以降,検索対象のデータの所有者をオーナー,検索者をユーザーと呼ぶ.

2.1 伊藤方式[1]

タグを用いた検索可能な秘密分散の方式として伊藤らの方式について示す.

2.1.1 Shamir の(m,t)閾値秘密分散法

(m,t)閾値秘密分散法は,次の 2 つの条件を満たす.

・任意の t 個の分散情報から,元の秘密情報 s を復元することができる.

・t−1 個以下の分散情報からは,秘密情報 s に関する情報は一切得ることができない.

Shamir の提案した多項式補間による方法では,以下のようにして(m,t)閾値秘密分散法を実現する.

[分散]

①s＜p かつ m＜p である素数 p を選ぶ.

②GF(p)の元から,異なる m 個の x_i(i=1,･･･,m)を選び出し,ユーザーID とする.

③GF(p)の元から,t−1 個の乱数 a_l(l=1,2,･･･,t−1)を選んで,以下の式を生成する.

$$W_i = s + a_1 x_i + a_2 x_i^2 + \cdots + a_{t-1} x_i^{t-1} \quad (1)$$

④上記式(1)の x_i に各ユーザーID を代入して,分散情報 W_i を計算し,各サーバーにこれらのサーバーID x_i と生成した分散情報 W_i を配布する.

[復号]

①復号に用いるサーバーを t 台選び,そのサーバーの持つ分散情報を W_j(j=1,2,･･･,t)とする.また,その分散情報に対応するユーザーID を x_j とする.

②分散式に x_j と W_j を代入し,t 個の連立方程式を解いて,秘密情報 s を得る.

s の復号の際には,Lagrange の補間公式を用いると便利である.

2.1.2 モデル

全体構成の概略図を図 1 に示す.

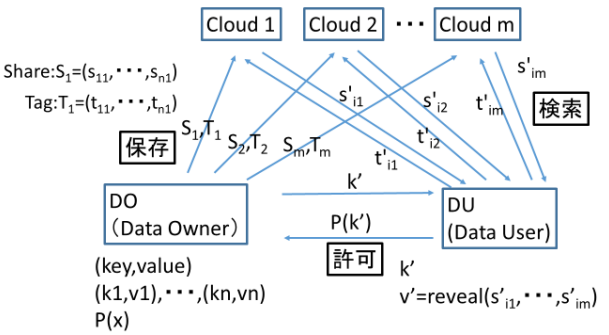

図 1 伊藤方式の全体構成

伊藤方式ではオーナー,ユーザー,クラウドの 3 者からシステムは構成されている.

・オーナー：データの所有者であり,キーk_i と対応する値 v_i の組を n 個保有している.
キーの集合{k_1,･･･,k_n}は一意であり,n 個の互いに異なる文字列から成る.一方,値 v_1,･･･,v_n は重複を許す.オーナーは検索を許可する利用者をアクセス制御するための秘密の多項式 P(x)を有する.

・ユーザー：あるキーk'に対応する値 v'を検索したい.ただし,オーナーからの検索許可の元で,クラウドに格納されたデータを入手する.

・クラウド：独立した m 台のサーバーから構成されている.m 台の内,高々t−1 台までしか盗聴されないことを仮定する.各クラウドには,独立したデータベースがあり,タグ t'に対応したシェア s'を格納している.シェアは,オーナーにより分散された値である.

2.1.3 アルゴリズム

P(x)は検索を許可する為の秘密の多項式とし,キーワード k に対応する P(k)の値から,m 個のサーバーへの対応する m 個のタグを,一方向性関数 H：{0,1}*→R_H を用いて,j＝1,･･･,m について,t_j＝H(j||P(k))と定める.

値 v は(m,t)閾値秘密分散法により,m 個の内,t 個が集まると秘密 v が復元する.

検索可能秘密分散法のアルゴリズムは以下のとおりである.

[分散]

オーナーは i=1,･･･,n について,キーk_i を入力する多項式値 P(k_i)を求め,m 個のタグ $t_{i,1}$,･･･,$t_{i,m}$ を算出する.ここで,$t_{i,j}$＝H(j||P(k_i))とする.秘密分散法により,値 v_i を $s_{i,1}$,･･･,$s_{i,m}$ の m 個のシェアに分散する.その後,j＝1,･･･,m について,n 個のタグ $t_{1,j}$,･･･,$t_{n,j}$ と対応する n 個のシェア $s_{1,j}$,･･･,$s_{n,j}$ を j 番目のクラウドのデータベースへ安全に登録する.

[許可]

ユーザーはオーナーにアクセス権限を要求し,許可されたら,オーナーとの間で秘匿多項式評価 OPE を実行して,オーナーにキーk'を知られることなく,P(k')を得る.秘匿多項式評価 Oblivious Polynomial Evaluation (OPE)とは,多項式 P を持つ送信者と値 α を持つ受信者が,送信者に α を知らせることなく P(α)を受信するプロトコルである [7][8][9].

[検索]

ユーザーは,m 台のクラウドの中からランダムに t 台を選び,それらを D⊂{1,･･･,m}と置く.m 台のクラウドのそれぞれにタグ t'1,･･･,t'm を送信し,データベースを検索する.ここで,

$$t_j' = \begin{cases} H(j||P(k')) & \text{if } j \in D, \\ H(j||P(k_r)) & \text{otherwise} \end{cases}$$

とする.k_r はキー空間{k_1,･･･,k_n}からランダムに選んだキーである.

Vol.2015-CSEC-69 No.10
Vol.2015-IOT-29 No.10
2015/5/21

ユーザーはクラウドから対応するシェア $t'_1,\cdots,t'_m$ を送り返してもらい,$\{s'_j \mid j\in D\}$となる t 個の真のシェアから値 v'を復元する.

2.1.4 クラウドに対する秘匿性

ユーザーがクラウドに送信するタグは確定的であるため,重複を含んだ複数回の検索により,クラウドはキーや値は分からないがキーの検索頻度などの統計情報が漏洩してしまう.特に,ユーザーが複数存在する場合には,検索キーの重複は避けられず,しかも単一のクラウドからもキーの統計情報が算出可能である.そこで秘密分散の閾値を利用して,このリスクを確率的に下げる.具体的には,ユーザーは m 個の中からランダムに t 個のクラウドを選び,それ以外のクライドには偽のタグ(チャフ)を送信する.これにより,統計情報の精度を下げることができる.

2.1.5 考察

伊藤方式では分散情報ごとにタグを生成し,タグとの一致判断によって検索を行っている.よって,この方式は秘密情報を直接検索することはできない.ただし,Shamir の秘密分散を用いているため,サーバー台数に制限はなく,閾値の台数のサーバーのみで検索を行える.しかし一致判断のため,全件検索を行う必要がある.また,オーナーにキーk'を知られることなく P(k')を得るために,OPE を用いるが,比較的大きな計算量が必要である.また,オーナーはキーk'を知らないとしても,キーが同じであればタグは同じであるので,偽のタグにより精度が下がるとはいえ,どのタグの情報が検索されたかクラウドに段階的に漏洩する.

2.2 只木方式[3]

この方式は[14]に示された方式を千田らが提案した高速乗算プロトコル[2]を採用して大幅な高速化を行った方式である.

2.2.1 基本設定

全体構成の概略図を図 2 に示す.

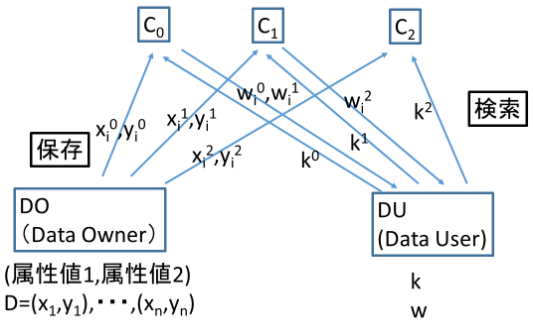

図 2　只木方式の全体構成

只木方式でのオーナー,ユーザー,クラウドは以下の通りである.

・オーナー : データベース D の所有者.データベース D は n 人の個人情報に関するリストからなるものとし,それぞれ 1 から n までの ID で区別されているものとする.個人情報は 2 種類とする.すなわちデータベース D は,次の形式が n 個集まり構成されているものとする :

ID,属性値 1,属性値 2

ここで,属性値 1 は$\{0,1\}^A$の元で表現され,属性値 2 は$\{0,1\}^B \setminus \{1^B\}$の元として表現されているものとする.以下では,$X:=\{0,1\}^A$,$Y:=\{0,1\}^B$ および $Y' := \{0,1\}^B \setminus \{1B\}$と表す.

・ユーザー : 情報の利用者（以下 U）.属性値 1 がある検索条件 P を満たすデータの ID と属性値 2 を検索したい.

・クラウド : (C_0,C_1,C_2)という 3 台のサーバーから構成されている.2 台に保存されたデータが欠損した場合,復元は不可能になる.

2.2.2 プロトコルの概要

データベース D は,ID を 1~n とする n 人の個人情報 $(x_1,y_1),\ldots,(x_n,y_n)\in X\times Y'$を完全に秘匿状態を維持しながら保持しているものとする.ここで x_1 および y_1 は,ID を i(i=1~n)とする個人の属性値 1 及び属性値 2 を,それぞれ表している.この方式では,データベース D から,属性値 1 が検索条件 P を満たす個人の ID のリスト,及び,このリストに ID を持つ個人の属性値 2 の情報を検索者 U(のみ)に開示するものである.その際,これらの情報以外の個人情報は,完全に保護される.特に,クラウドを構成する C_0,C_1,C_2 には,D に関する情報は一切漏洩しない.なお,Y'の定義から $1^B\in Y'$であるが, 1^B は U に対して個人の属性値 2 が非公開(即ち"黒塗り")であることを意味するものである.只木方式は次のプロトコルにより与えられる.

2.2.3 プロトコル

例として,属性値 1 が k を超える個人の ID をリストアップし,それらの ID の属性値 2 を開示する流れを説明する.

このプロトコルの入力は,D 上のデータ

$(x_1,y_1),\ldots,(x_n,y_n)\in X\times Y'$

である.

・Phase1

D は,各 i＝1,…,n について,独立な 4 つの乱数 $x_i^0,x_i^1\in X$ と $y_i^0,y_i^1\in X$ を選び,

$$x_i^2 := x_i \oplus x_i^0 \oplus x_i^1$$

および

$$y_i^2 := y_i \oplus y_i^0 \oplus y_i^1$$

を計算する.ここで⊕はビット毎の XOR を表す.そして,D は$((x_i^0, x_i^1),(y_i^0, y_i^1))$を C_0 に送信し,$((x_i^1, x_i^2),(y_i^1, y_i^2))$を C_1 に送信し,$((x_i^2, x_i^0),(y_i^2, y_i^0))$を C_2 に送信する.

C_0 は受信データに基づいて,

$((x_1^0, x_1^1),(y_1^0, y_1^1)),\ldots, ((x_n^0, x_n^1),(y_n^0, y_n^1)) \in X^2\times Y^2$

を保持する.C_1 は受信データに基づいて,

$((x_1^1, x_1^2),(y_1^1, y_1^2)),\ldots, ((x_n^1, x_n^2),(y_n^1, y_n^2)) \in X^2\times Y^2$

を保持する.C_2 は受信データに基づいて,

$((x_1^2, x_1^0),(y_1^2, y_1^0)),\ldots, ((x_n^2, x_n^0),(y_n^2, y_n^0)) \in X^2\times Y^2$

を保持する.

・Phase2

U は検索条件である任意の $k\in X$ を選び,これを C0,C1,C2 のそれぞれに送信する.

・Phase3

C_0,C_1,C_2 の間で,各 i＝1,…,n について,以下を行う :

$$z_i^0 \oplus z_i^1 \oplus z_i^2 = \begin{cases} 1 & \text{if } x_i^0 \oplus x_i^1 \oplus x_i^2 > k, \\ 0 & \text{otherwise} \end{cases}$$

を満たす（z_i^0, z_i^1, z_i^2）$\in\{0,1\}\times\{0,1\}\times\{0,1\}$をランダムに生成し，（$z_i^0$, z_i^1）は C_0 が保持し，（z_i^1, z_i^2）は C_1 が保持し，（z_i^2, z_i^0）は C_2 が保持する．

・Phase4

C_0,C_1,C_2 の間で,各 i＝1,…,n について,以下を行う：

$$w_i^0 \oplus w_i^1 \oplus w_i^2 = \begin{cases} y_i^0 \oplus y_i^1 \oplus y_i^2 & \text{if } z_i^0 \oplus z_i^1 \oplus z_i^2 = 1, \\ 1^B & \text{otherwise} \end{cases}$$

を満たす（w_i^0, w_i^1, w_i^2）$\in Y\times Y\times Y$ をランダムに生成し，（w_i^0, w_i^1）は C_0 が保持し，（w_i^1, w_i^2）は C_1 が保持し，（w_i^2, w_i^0）は C_2 が保持する．

・Phase5

C_0は((w_1^0, w_1^1),…, (w_n^0, w_n^1))を U に送信し,C_1は(w_1^2,…, w_n^2)を U に送信する．

・Phase6

U は,各 i＝1,…,n に対して$w_i^0 \oplus w_i^1 \oplus w_i^2$を計算し,集合

$$\{i \in \{1,\dots,n\} \mid w_i^0 \oplus w_i^1 \oplus w_i^2 \neq 1^B\}$$

を求める.この集合が,属性値 1 が条件 P を満たす個人の ID のリストになる.そしてこのリスト中に ID i を持つ個人の属性値 2 は,$w_i^0 \oplus w_i^1 \oplus w_i^2$で与えられる．

2.2.4 プライベート情報検索機能の付加

検索条件を秘匿するプライベート情報検索機能を追加することが可能である.2.2.3 のプロトコルの Phase2 では,U から C_0,C_1,C_2 に,属性値 1 についての検索条件 P の内容を表す k が渡される.しかし,この k を属性値 1,2 と同様に分散した状態で送信することで,C_0にも C_1にも C_2にも秘匿したまま,プロトコルを実行することは可能である．

2.2.5 考察

只木方式は,全件に対して大小比較を行い,検索条件を満たさないデータ ID の属性値 2 を 1^B という形で塗りつぶすことで検索を行っている.この方式は,サーバー台数 3,復元の閾値 2 のみに限定されている.また,検索の計算は 3 台で行っている．

3 提案方式

本章では,サーバー台数を限定しない秘密分散法を用いた秘匿検索法を提案する．

3.1 モデル

提案方式はオーナー,ユーザー,サーバーから構成される．

・オーナー：データの所有者,n 個のデータを持つ．

・ユーザー：検索条件を満たすデータを検索したい．

・サーバー：n 台あり,オーナーがデータを秘密分散し保存している．

3.2 基本方式

［分散］

オーナーは n 個の秘密情報を k_j(j=1,･･･,n)とし,共通乱数 r と秘密情報ごとの乱数 r_j を生成する.その後,下記の分散値 $F_j(x_i)$,$R_j(x_i)$を算出し,サーバーx_i(i=1,･･･,m)に分散保管する．

$$F_j(x_i) = r \cdot r_j(k_j + a_1x_i + a_2x_i^2 + \cdots + a_{t-1}x_i^{t-1})$$
$$R_j(x_i) = r_j + b_1x_i + b_2x_i^2 + \cdots + b_{t-1}x_i^{t-1}$$

［許可］

①ユーザーは検索キーワード k に乱数 q を乗じた q･k をオーナーに送る．

②オーナーは q･k に r を乗じた r･q･k を秘密分散し,各サーバーに送る．

$$K(x_i) = r \cdot q \cdot k + c_1x_i + c_2x_i^2 + \cdots + c_{t-1}x_i^{t-1}$$

［検索］

①ユーザーは t 台のサーバーから $F(x_i)$受け取り,乱数 q を乗じる.その後,$g(0)=0$ となる多項式 $g(x)$を生成し,以下の計算をして,各サーバーに送る．

$$F'(x_i) = q \cdot F_j(x_i) + g(x_i)$$

②サーバーは, $R_j(x_i)$を復号する.その後,乱数 q_j を生成し,以下の計算をし,ユーザーに送信する．

$$D_j(x_i) = q_j\left\{F_j'^{(x_i)} - r_j \cdot K(x_i)\right\}$$

$$= r \cdot r_j \cdot q \cdot q_j \cdot \{(k_j - k) + d_1x_i + d_2x_i^2 + \cdots + d_{t-1}x_i^{t-1}\}$$

③ユーザーは, $D_j(x_i)$を復元し， r･r_j･q･q_j･（k_j－k）より,k と k_j の大小関係をもとめる．

④ユーザーは, 上記差分が 0 である場合には,検索 ID が一致したと判断する.差分が 0 でない場合,①②③を繰り返し, 検索結果を得る．

3.3 考察

基本方式では,ユーザーが示す検索キーワードは同じ値であっても q を変えることで毎回違う値をオーナーに送ることができる.また,検索一致の判断をユーザーが行うため,ユーザーが検索結果を得た後もダミーの検索を行えば,クラウドはどれが検索の最終結果か分からないため,ユーザーは自分の検索キーワードを完全に秘匿することができる.また,基本方式は[2]と異なり秘密情報を直接検索しているため, タグとなる値も提案方式によって秘密分散すればタグと秘密情報の双方向検索が可能になる.ただし,提案方式（基本方式）はオーナーとユーザー間で頻繁に通信が必要となるため,以下にサーバー内に計算部を設け,オーナーとユーザーからの通信は 1 度だけですむ拡張方式を以下に示す．

3.4 拡張方式

［分散］

オーナーは n 個の秘密情報に対応する検索 ID(キーワード)を k_j(j=1,･･･,n)とし,共通乱数 r と検索 ID ごとの乱数 r_j を生成する.その後,下記の分散値 $F_j(x_i)$,$R_j(x_i)$を算出し,サーバーx_i(i=1,･･･,m)に分散保管する．

$$F_j(x_i) = r_j(k_j + a_1x_i + a_2x_i^2 + \cdots + a_{t-1}x_i^{t-1})$$
$$R_j(x_i) = r \cdot r_j + b_1x_i + b_2x_i^2 + \cdots + b_{t-1}x_i^{t-1}$$

［許可］

①ユーザーは検索条件(キーワード)k に乱数 q を乗じた q･k をオーナーに送る．

②オーナーは乱数 q’を生成し,q･k に q’を乗じた q･q’･k を秘密分散し,各サーバーに送る．

$$K(x_i) = q \cdot q' \cdot k + c_1x_i + c_2x_i^2 + \cdots + c_{t-1}x_i^{t-1}$$

③オーナーは q’･r をユーザーに送る．

［検索］

①ユーザーは受け取った $q'\cdot r$ に q をかけて秘密分散し,各サーバーに送る.

$$Q(x_i) = r\cdot q\cdot q' + d_1 x_i + d_2 x_i^2 + \cdots + d_{t-1} x_i^{t-1}$$

②計算部は,t 台のサーバーから $F_j(x_i)$,$R_j(x_i)$を受け取り復号し,$rj\cdot k_j$ と $r\cdot rj$ を求める.

③計算部は, $D_j(x_i)=r_j\cdot k_j\cdot Q(x_i)-r\cdot r_j\cdot K(x_i)$を計算して復元し，$r\cdot r_j\cdot q\cdot q'$（$k_j-k$）より,k と k_j の大小関係をもとめる.

④計算部は, 上記差分が 0 である場合には,検索 ID が一致したと判断する.差分が 0 でない場合,②③を繰り返し, 検索結果をユーザーに送る.

3.5 検索順序について

提案方式では検索を効率化するために全件探索ではなく,二分探索による検索を行う.これは分散の際の素数 p を十分に大きな値にすれば可能である.また,乱数 r,rj,q,q' は合成数とすることで,因数分解によって差分の値を特定することを困難にする.

二分探索は次のように行う.分散の際,あらかじめ検索 ID の昇順で保存し,検索では中央値 $k_j(j=m/2)$から比較を行う.すなわち,④において差分が正の値であれば k_j が k より大きいと判断し,上方向の中央値を検索し,差分が負であれば下方向の中央値を検索する.これにより,全件探索では n 回繰り返す計算を最大 $\log_2 n$ 回にまで減らすことが出来る.

また,検索 ID を追加する場合も,追加 ID を検索したい ID として秘匿検索を行い,最終位置の隣接 ID（追加 ID は検索 ID 中にはないので最後まで一致はしないが,最終位置では隣接 ID の検索結果の方向が逆になり,追加 ID がその中間位置に入るべきであることがわかる）の中間位置に追加 ID の分散値を追加する.ただし,データのオーナーが 1 名ではなく,異なるオーナーがデータの追加を行う場合,データは昇順に並んでも,オーナーの順番がバラバラとなるため,検索のたびにオーナーの許可が必要となる.ゆえにこの場合,オーナーはシステムに検索を許可するユーザーの条件を示し,検索の許諾をシステムに委託するなどすれば効率的となる.

4 比較

本章では,従来方式と提案方式との比較を行う.伊藤方式と只木方式それぞれとの違いを示し,考察を行う.なお,提案方式は 3.2 で説明した計算部なしの方式として比較する.

4.1 伊藤方式との違い

検索の際の計算量と通信量について比較する.計算量は検索を行うときに行う計算について算出する.通信量はユーザーとサーバーとの間で送信するデータ量,サーバー間で通信されるデータ量を算出する.なお,サーバー1 台に保存されているデータ数を n 個,検索に使用するサーバーの台数を t 台とする.

伊藤方式の検索で行われる計算は,ユーザーのタグ生成とサーバーでのユーザーから送信されたタグとデータごとのタグが一致しているかの検証なので,t 回のハッシュ関数と n 回の減算である.また,ユーザーからサーバーへの通信は,タグの送信であり,サーバー間での通信はない.

提案方式の場合,検索で行われる計算は,ユーザーによる $F'_j(x_i)$を求めるための加算,乗算,$D_j(x_i)$の復号と,サーバーでの $R_j(x_i)$の復号,$D_j(x_i)$を求める減算,乗算である.二分木検索のため,計算は最大 $\log_2 n$ 回繰り返す.ユーザー・サーバー間で通信されるのは,$F_j(x_i)$ $F'_j(x_i)$ $D_j(x_i)$であり,サーバー間で通信されるのは $R_j(x_i)$である.

これらを表 1 にまとめる.$n>>\log_2 n$ のため,計算量は提案方式の方が少ない.しかし提案方式ではサーバー間での通信を必要とするため,通信量は伊藤方式の方が少ない.

4.2 只木方式との比較

サーバー台数 3 台,復元の閾値 2,k より大きい値を検索していると仮定し,ユーザーが検索条件をサーバーに送る時から条件を満たす ID の特定までの計算量と通信量を比較する.なお,サーバー1 台に保存されているデータ数を n 個とし,検索条件は k よりも大きいデータの ID を特定することとする.只木方式の場合,ユーザーの行う計算は,復元を行う際の XOR である.一方,サーバー1 台が検索の際,行う計算は Phase3 での大小比較である.大小比較では ID ひとつにつき,12 回の乗算と 28 回の XOR が必要であり,さらに全 ID に対して行わなければならない.通信量は,ユーザー・サーバー間では,ユーザーからの条件を送った時と,サーバーが結果を返した時を考える.サーバー間の通信は,AND,OR のアルゴリズムを利用する時発生する.

一方,提案方式の場合,4.1 で求めた計算量,通信量を元に t=2 を考慮して求めている.検索条件は k よりも大きいことであるが,これは検索 ID を追加する場合の手順を利用することで実現できる.

これらを表 2 にまとめる.表 2 からもわかるように二分探索を行える提案方式に比べて只木方式は明らかに計算量,通信量が多い.また只木方式では,復元は 2 台の持つデータで行えるが,計算はサーバー3 台を必要とする.一方提案方式では,サーバー3 台に分散し,復元の閾値を 2 にした場合,計算,復元に必要なサーバーは 2 台である.

5 今後の課題

提案方式では,Shamir の秘密分散を用いているが,計算量が多く,計算を繰り返す検索に対して有効とは言えない.そこで XOR 方式において秘密情報を数値として扱う多値化方式[15]を用いて高速化を行う予定である.さらに,提案方式の安全性についてより厳密に検討していく.

参考文献

[1] 伊藤孝一,牛田芽生恵,山岡裕司,及川孝徳,菊池浩明. “検索可能秘密分散方式の提案” 2014-CSEC-64.

[2]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” SCIS2011

[3] 只木孝太郎,辻井重男.“プライバシー保護条件付き情報開示 II” ISEC2014-13.

[4]D. song, D. wagner, and A. Perrig, “Practival techniques for searches on encrypted data,” Proc. of the 2000 IEEE Symposium on Security and Privacy, pp.44-55, 2000.

[5]小岩敬太,金岡晃,岡本栄司.“動的かつ並列化された検索可能対称暗号の改良方式” SCIS2014.

[6]服部充洋,松田規,平野貴人,森拓海,伊藤隆,川合豊,坂井祐介,太田和夫． 安全性と高速性の両立を目指した検索可能暗号(2)

SCIS2012.
[7]Naor, M. and Pinkas, B., Oblivious Polynomial Evaluation SIAM Journal on Computing, 2006, Vol. 35, No. 5, pp. 1254-1281.
[8]Hanaoka, G., Imai, H., Mueller-Quade, J., Nascimento, A. C., Otsuka, A., Winter, A. “Information theoretically secure oblivious polynomial evaluation: Model, bounds, and constructions”, In Information Security and Privacy, ACISP 2004, pp. 62-73, Springer, 2004.
[9]Lindell, Y., Pinkas, B, “Privacy preserving data mining”, In Advances in Cryptology CRYPTO 2000, pp.36-54, Springer, 2000.
[10]千田浩司,五十嵐大,高橋克巳. ”効率的な 3 パーティ秘匿関数計算の提案とその運用モデルの考察” CSEC48,2010.
[11]五十嵐大,千田浩司,高橋克巳. “高効率 3 パーティ秘匿関数計算の情報理論的安全性” CSEC50,2010.
[12]千田浩司,濱田浩気,五十嵐大,高橋克巳. “軽量検証可能 3 パーティ秘匿関数計算の再考” CSS2010,2010.
[13]濱田浩気,五十嵐大,千田浩司,高橋克巳. “3 パーティ秘匿関数計算のランダム置換プロトコル” CSS2010,2010.
[14]只木孝太郎,土井範久,辻井重男.”プライバシー保護条件付き情報開示”SCIS2012
[15]高橋加寿子,須賀祐治,岩村惠市. “XOR を用いる秘密分散法の多値化とそれを用いた秘匿計算法”CSEC65,2014

表 1　伊藤方式との比較

	伊藤方式	提案方式
ユーザーの計算量	t 回のハッシュ関数	$\log_2 n$ 回の加算 $\log_2 n$ 回の乗算 $\log_2 n$ 回の復号
サーバー1 台の計算量	n 回の減算	$\log_2 n$ 回の減算 $2\log_2 n$ 回の乗算 $\log_2 n$ 回の復号
ユーザーとサーバーとの通信量	$t \cdot \lvert H(j \lvert\rvert P(k))\rvert$	$t \cdot \log_2 n \cdot (\lvert F(x)\rvert + \lvert F'(x)\rvert + \lvert D(x)\rvert)$
サーバー間での通信量	0	$(t-1) \cdot \log_2 n \cdot \lvert R(x)\rvert$

表 2　只木方式との比較

	只木方式	提案方式
ユーザーの計算量	2(n+1)回の XOR	$\log_2 n$ 回の加算 $\log_2 n$ 回の乗算 $\log_2 n$ 回の復号
サーバー1 台の計算量	12n 回の乗算 28n 回の XOR	$\log_2 n$ 回の減算 $\log_2 n$ 回の乗算 $\log_2 n$ 回の復号
ユーザーとサーバーとの通信量	$3n \cdot (\lvert k\rvert + \lvert z\rvert)$	$2\log_2 n \cdot (\lvert F(x)\rvert + \lvert F'(x)\rvert + \lvert D(x)\rvert)$
サーバー間での通信量	$3n \cdot (\lvert c\rvert + \lvert r\rvert)$	$\log_2 n \cdot \lvert R(x)\rvert$