Vol.2014-DPS-158 No.13
Vol.2014-CSEC-64 No.13
2014/3/6

検索可能秘密分散方式の提案

伊藤 孝一[1]　牛田 芽生恵[1]　山岡 裕司[1]　及川 孝徳[1]　菊池 浩明[2,a)]

概要：クラウドに委託したデータの漏えいを防止してそのサービスを活用する為に，様々な検索可能暗号方式が提案されている．キーワードを暗号化してデータと組みにして登録し，検索を許可されたユーザが検索用のタグを送信することで，該当するデータのみを抽出する．一方，データの暗号化の代わりに，秘密分散により複数のサーバにデータを分散管理する仕組みも注目されている．そこで，本研究では，キーワードにより該当するデータのみを抽出する検索可能秘密分散方式を提案する．

Abstract: Confidential data should be encrypted in out-sourcing services in cloud computing environment in order to minimize the risk of data revealing. There have been many schemes, classified as searchable encryption, which provides capabilities to securely search over encrypted data through keywords without decryption key. In this paper, we try to combine the technique of searchable encryption with a secret sharing scheme that allows us to retrieve the portion of confidential data without recovering data.

Keywords: cloud privacy, keyword search, secret sharing

1. はじめに

クラウドに委託したデータの漏えいを防止してそのサービスを活用する為に，様々な検索可能暗号方式が提案されている．Boneh らは，データ所有者が公開鍵を用いてデータを暗号化してクラウドに保管し，秘密鍵を持つ利用者が検索用のタグを生成して，クラウドに検索を実行させる公開鍵検索可能暗号 (Public Key Encryption with keyword Search: PEKS) を提案した [1]．これを元にして，キーの連言検索や範囲検索を可能とする方法 [2] などいくつもの改良が試みられている．松田らは，階層型 ID ベース暗号を用いてマルチユーザへの対応を可能とする方式を提案し，ブラウザと Web サーバ間で SQL 文による検索を可能とするシステムを実装している [8]．

Boneh の方式 [1] の概要は次のとおりである．双線形写像 $e : G_1 \times G_1 \to G_2$ を用いて，秘密鍵 x と公開鍵 $h = g^x$ を定め，キー k に対するタグ $t = e(H_1(k), h^r)$ を求め，暗号文 $g^r, H_2(t)$ と共にクラウドに格納する．検索者は，キー k' と秘密鍵を用いてトラップドア $T = H_1(k')^x$ をクラウドに送り，$H_2(e(T, g^r)) = H_2(t)$ を満たす暗号文 $g^r, H_2(t)$ を探す．ここで，G_1, G_2 は有限群，H_1, H_2 はハッシュ関数である．この方式を注意深く見ると，次の特徴がある．

(1) 暗号化は確率的に行われ，公開鍵を持つ誰でもデータを登録できる．
(2) トラップドアを計算できるのは秘密鍵を持つ限られた利用者である．トラップドアは確定的に計算される（キーが同じならば同じ結果）．
(3) クラウドは，タグとトラップドアが同一のキーについて計算されているかを復号することなく実施できる．ただし，登録されているデータ数 n に比例する回数検査する必要がある．
(4) キーに対応するデータを保存するには，クラウドは別の暗号化方式を用いる必要がある．

ここで，(1) の安全性は Boneh らによって，選択タグ攻撃に対する暗号文が識別不能である (IND-CCA) であることが証明されているが，(2) のトラップドアが確定的であるので，検索者のクエリーからデータが識別できてしまう．(3) の検索効率が $O(n)$ であるという点はデータベースの運用規模に関わり，応用範囲を狭めてしまう．

一方，Bellare らは，OAEP のパディングに疑似乱数のシードとしてキーを用いることで，確定的な暗号化による PEKS を提案している [3]．またその安全性モデル (PRIV セキュリティ) を提案している．彼らの方式では，暗号文が確定的であるので，クラウド側で二分木やハッシュ表を用いることで sublinear な検索効率を可能とする．ただし，

[1] 株式会社富士通研究所
FUJITSU LABORATORIES LTD., 4-1-1 Kamikodanaka, Nakahara, Kawasaki
[2] 明治大学
Meiji University, 4-21-1 Nakano, Nakano, Tokyo 164-8525 Japan
[a)] kikn@meiji.ac.jp

キー空間には十分なエントロピーがあり，暗号文から平文を総当たりで検索出来ないことを仮定している．

この代表的な二種類の PEKS 方式に基づき，タグそのものは Bellare らの方式の様に確定的に求めて，検索効率の問題点を解決し，データそのものはタグとは異なる秘密分散を用いることで，Bellare の方式の課題であったタグとデータが直接比較できない方式を提案する．提案方式は公開鍵暗号は用いず，秘密分散を基本として用いるために，シェアは確定的でなく，クラウドのしきい値までの信頼の元で情報理論的な安全性を保証している．Boneh らの方式と同様に，検索クエリーを行う利用者はある秘密情報を有した権限を持つものにアクセス制御が可能である．トラップドア（タグ）の作成には秘匿多項式評価プロトコルを応用することで，データ所有者に検索キーを情報を秘匿することが可能である．更に，トラップドア（タグ）からクラウドサーバに検索キーが漏れないように，疑似データ（チャフ）を混入させることで，完全守秘性を満たすことを保証する．

2. 要素技術

2.1 秘密分散

f を秘密 s を $f(0) = s$ とする t 次の多項式とする．互いに異なる $t+1$ 個の要素 $a_1, \ldots, a_{t+1}$ のシェア $f(a_1), \ldots, f(a_{t+1})$ から秘密を復元するスキームを，(m, t) しきい値法と呼ぶ．t 以下のシェアからは秘密に関する情報は漏れない．本稿では，代表的なしきい値法として，Shamir の秘密分散を用いる．

2.2 秘匿多項式評価 OPE

秘匿多項式評価 Oblivious Polynomial Evaluation (OPE) 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 $P(\alpha)$ を受信するプロトコルである．

Naor は，二変数の多項式と Oblivious Transfer(OT) を用いて最初のプロトコルを提案している [5]. OT の安全性の仮定のもと，プロトコルの view が理想的なものと計算量的に識別不能であることを証明している．

Lindel らは，セキュア内積プロトコルを用いて次の様に OPE を実現している [7]．送信者は多項式 $P(x) = a_0 + a_1 x + \cdots + a_t x^t$ の係数から成るベクトル $(a_0, \ldots, a_t)^{-1}$ を持ち，受信者は $(1, \alpha, \ldots, \alpha^t)^{-1}$ を持ち，セキュア内積プロトコルを用いてそれらの内積，すなわち，$P(\alpha)$ を得る．安全性は，セキュア内積プロトコルを構成する加法準同型性を満たした暗号の安全性に帰着している．

花岡らは，3 者間で行う次の様な OPE を提案している [6]．第三者がランダムな多項式 S と値 d を選び，S を送信者に，d と $g = S(d)$ を受信者に安全に送信する．受信者は $t = \alpha - d$ を送信者に送る．送信者は，$f(x) = P(x+t) + S(x)$ となる多項式を作り，受信者に送り返す．受信者は，$f(d) - g = P(d+t) + S(d) - S(d) = P(\alpha)$ を入手する．このプロトコルの安全性は，情報理論的に証明されている．

2.3 Polling-Hellman 暗号

Polling-Hellman (PH) 暗号 [4] は，$2q+1 = p$ となる素数 p, q と秘密鍵 s について，平文 x の暗号文 $E(x) = x^s \bmod p$ と定める共通鍵暗号である．暗号文 c は，s の $\bmod q$ の逆元 $1/s$ を用いて，$D(c) = c^{1/s} \bmod p$ で与えられる．この可換群をなす暗号化関数をハッシュ関数

$$H(x) = x^s \bmod p$$

に利用することで，秘匿関数計算が可能である．

2.4 ブラインド署名

ブラインド署名は，署名者に平文 x を知られることなく，署名を得るプロトコルである．RSA 暗号に基づいた代表的な方式では，次の様にして構成できる．

送信者は，乱数 r を選び，署名者の公開鍵 e, N を用いて $xr^e \pmod N$ を送信する．署名者は秘密鍵 d を用いて，$(xr^e)^d \pmod N$ を求めて送り返す．送信者は r の逆元を用いて，$(x^d r^{ed})/r = x^d \pmod N$ により x の署名を得る．

3. 検索可能秘密分散

3.1 モデル

図 1 に検索可能秘密分散の全体図を示す．本システムは，Data Owner (Owner), Data User (User), Cloud の 3 者から成る．

Owner は，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 $\{k_1, \ldots, k_n\}$ 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 $v_1, \ldots, v_n$ は重複を許す．Owner 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．

User は，あるキー k' に対応する値 v' を検索したい．ただし，Owner 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．

Cloud は，独立した m 台のサーバから構成されている．m 台の内，高々 t 台までしか不正を行わ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，Owner により分散された値である．

3.2 要求条件

Boneh[1] らにより定式化された検索可能公開鍵暗号方式で満たされている条件を元にして，検索可能秘密分散方式が満たすべき次の要求条件を定める．

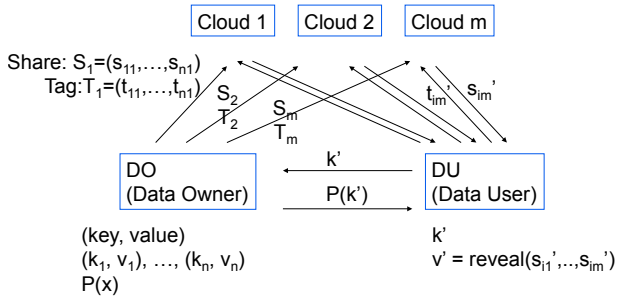

図 1 検索可能秘密分散の全体構成

(1) Cloud サーバは，格納しているタグとシェアの組から，対応するキーと値が分からない．
(2) Owner は User の検索したいキー k' が分からない．
(3) User は，Owner により許可された時だけ Cloud から検索できる．
(4) 検索は sublinear に効率よく実施する．
(5) 検索タグが与えられた時の，Cloud サーバにキーと値が漏れない．

3.3 基本方式

$P(x)$ を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 個のサーバへ対応する m 個のタグを，一方向性関数 $H : \{0,1\}^* \to R_H$ を用いて，$j = 1, \ldots, m$ について，

$$t_j = H(j \,||\, P(k))$$

と定める．

値 v は (m, t) しきい値秘密分散法により，$s_1, \ldots, s_m$ のシェアに分散される．m 個の内，$t+1$ 個が集まると秘密 v が復元する．

基本プロトコルを Algorithm 1 に示す．Owner と User は多項式 $P(x)$ を共有しており，それによりデータベースに格納した n 個のシェアから適切なものを検索することが出来る．$P(x)$ は確定的な関数であり，$k = k'$ ならば $P(k) = P(k')$ により，正しい検索結果を保証している．

n 個のキーとシェアの組は対応しているが，各 Cloud で独立に管理されていることに注意せよ．例えば，あるキー k_i の Cloud 1 に格納されているタグ $t_{i,1}$ の順番は Cloud 2 の $t_{i,2}$ と同じとは限らない．従って，仮に Cloud 1 と 2 が結託したとしても，それぞれのシェアはシャッフルされているようなものであり，秘密を復元するのは困難である．

Owner が保有するキー集合は互いに異なる n 個の要素であっても，Cloud j で格納する n 個のタグ $t_{1,j}, \ldots, t_{n,j}$ が一意である保証はなく，$t_{i,j} = t_{i',j}$ となる $i \neq i'$ が生じる可能性がある．そこで，H の値域 R_H は衝突が起きない程十分に大きいことを仮定する．

Algorithm 1 基本プロトコル

Input: Owner: $P(x)$, $(k_1, v_1), \ldots, (k_n, v_n)$
User: k', $P(x)$
Cloud: none

1. **(share)** Owner は $i = 1, \ldots, n$ 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 $t_{i,1}, \ldots, t_{i,m}$ 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 $s_{i,1}, \ldots, s_{i,m}$ の m 個のシェアに分散する．
2. **(add)** Owner は $j = 1, \ldots, m$ について，n 個のタグ $t_{1,j}, \ldots, t_{n,j}$ と対応する n 個のシェア $s_{1,j}, \ldots, s_{n,j}$ を j 番目の Cloud のデータベースへ安全に登録する．
3. **(search)** User はキー k' について $P(k')$ を求め，m 台の Cloud のそれぞれにタグ $t'_1, \ldots, t'_m$ を送信し，データベースを検索する．ここで，$t'_j = H(j \,||\, P(k'))$ とする．
4. **(recover)** User は Cloud から対応するシェア $s'_1, \ldots, s'_m$ を送り返してもらい，秘密分散法に従って値 v' を復元する．

3.4 User のアクセス制御

基本方式は，Owner と User が共通の多項式を共有しており，それゆえ，User が無制限にデータベースに検索要求を送ることを許してしまう．しかし，多項式 $P(x)$ を Owner だけが管理するようにすると，利用者が検索したいキー k' を Owner が知ってしまう．ある種のアプリケーション[*1]においては，キー k' を Owner に対して秘匿しなくてはならない．そこで，Owner が User にアクセス権限を承認することの可能なプロトコルを Algorithm 2 に示す．

Algorithm 2 検索権限を制御するプロトコル

Input: Owner: $P(x)$, $(k_1, v_1), \ldots, (k_n, v_n)$
User: k'
Cloud: none

1. **(share)**
2. **(add)** Algorithm 1 と同じ．
3. **(search)** User は Owner にアクセス権限を要求し，許可されたら，Owner との間で秘匿多項式評価 OPE を実行して，Owner にキー k' を知られることなく，$P(k')$ を得る．m 台の Cloud のそれぞれにタグ $t'_1, \ldots, t'_m$ を送信し，データベースを検索する．
4. **(recover)** Algorithm 1 と同じ．

Step 3 の OPE には，2.2 であげた Naor らによるプロトコルの他にも，加法準同型性暗号を用いたセキュア内積プロトコルを用いたものがある．前者が情報理論的な安全性であるのに対して，後者は計算量的な困難性の仮定を必要とする．秘密分散は計算量的な安全性を保証しているので，安全性の観点では Naor らのスキームを用いるのが望ましい．

ただし，計算効率などを考慮して，セキュア内積プロトコルを用いるのであれば，User が Polling-Hellman (PH) 暗号 [4] を用いて，$k'^r \bmod p$ を Owner に送り，$k'^{rs} \bmod p$ から $t' = (k'^{rs})^{1/r} \bmod p$ により秘匿してタグを求めるこ

[*1] Owner が有料のゲノムデータベースを保持し，User が製薬会社で極秘にある薬品の影響を調べたいとする例などではこの仮定が必要である．

とが出来る．PH 暗号の代わりに，ブラインド RSA 署名を用いる変種も可能である．

3.5 クラウドに対する秘匿性

基本方式も検索権限を制御するプロトコルも，User が Cloud に送信するタグは確定的である．従って，重複を含んだ複数回の検索により，Cloud 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，User 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 Cloud からもキーの統計情報が算出可能である[*2]．

そこで，秘密分散のしきい値を利用して，このリスクを確率的に下げるプロトコルを考える．User は m 個の中からランダムに $t+1$ 個の Cloud を選び，それ以外の Cloud には偽のタグ（チャフ）を送信する．本プロトコルを Algorithm 3 に示す．

Algorithm 3 キー秘匿検索プロトコル

Input: Owner: $P(x)$, $(k_1, v_1), \ldots, (k_n, v_n)$
User: k'
Cloud: none

1. (share)

2. (add) Algorithm 1 と同じ．

3. (search) Algorithm 2 と同様にして，$P(k')$ を得る．User は，m 台の Cloud の中からランダムに $t+1$ 個を選び，それらを $D \subset \{1, \ldots, m\}$ と置く．m 台の Cloud のそれぞれにタグ $\hat{t}_1, \ldots, \hat{t}_m$ を送信し，データベースを検索する．ここで，

$$\hat{t}_j = \begin{cases} H(j \,||\, P(k')) & \text{if } j \in D, \\ H(j \,||\, P(k_r)), \in & \text{otherwise.} \end{cases}$$

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

4. (recover) User は Cloud から対応するシェア $s'_1, \ldots, s'_m$ を送り返してもらい，$\{s'_j \mid j \in D\}$ となる $t+1$ 個の真のシェアから値 v' を復元する．

4. 評価

4.1 安全性

要請条件 (1) の格納されている値の Cloud サーバに対する秘匿性について考える．まず，秘密分散法の情報理論的な安全性から次の秘匿性が明らかである．

命題 4.1 高々しきい値 t 台のサーバが悪意を持つ Cloud において，格納されたシェア $s_{i,1}, \ldots, s_{i,t}$ から値 v_i は漏れない．

しきい値を超えた不正があっても，次の秘匿性が言える．ここでは，Algorithm 1, 2, 3 に依らず，Step 2 までの状態における安全性を考えている．

命題 4.2 しきい値以上のサーバが不正をする Cloud においても，タグを構成するハッシュ関数の一方向性の仮定のもと，シェアを復元出来る確率は無視できる．

(証明) $t < t'$ となる t' 台の Cloud サーバが結託をしたと仮定する．Algorithm 1 に従って t' 組のシェアが入手できるが，ハッシュ関数の一方向性により n 個のタグが n 個の値のどれに対応しているか識別ができない．あるキー k_i に対応するタグを t' 個の正しく選ぶ確率は，$1/n^{t'-1}$ であり，n を大きくするに従い 0 に収束する． (証明終)

問題点は，むしろ Step 3 において User から送信されるタグ $t'_1, \ldots, t'_m$ が与えられると Cloud がシェアの対応を知ってしまうところにある．Algorithm 1, 2 では確定的にタグを送るため，$t+1$ 台の Cloud サーバが結託すると値が露見する．一方，Algorithm 3 では，次が言える．

命題 4.3 User からあるキーに対応したタグ $\hat{t}_1, \ldots, \hat{t}_m$ が与えられた時，m 台の不正な Cloud サーバが対応する値 v' を復元できる確率は無視できる．

(証明) Algorithm 3 において，値を復元するためには，m 個のタグの中から正しいタグを $t+1$ 個選ぶ必要がある．その場合の数は，$\binom{m}{(t+1)}$ である．復元に成功する確率はその逆数であり，Staring の近似公式を用いた下限により，

$$\frac{1}{\binom{m}{t+1}} < \left(\frac{t+1}{m}\right)^{t+1}$$

である．サーバ台数 m を増加することで右辺は 0 に収束する． (証明終)

更に強い安全性として，複数のタグから元のキーが同一かどうかを判定する識別可能性を検討する．キー k' と k'' について，Algorithm 3 に従って計算されたタグ $\hat{t'_j}$ と $\hat{t''_j}$ が与えられている時，j 番目の Cloud サーバが $k' = k''$ を識別する確率は次の式で与えられる．

補題 4.1 高々 t 台しか不正でない m 台のサーバから成る Cloud において，ある Cloud サーバに送信された二つのタグが $\hat{t'} = \hat{t''}$ である時，対応するキー k', k'' が等しい確率は

$$Pr(k' = k'' \,|\, t' = t'') = \frac{p^2 + 2p/n + 1/n}{p^2 + 2p/n + 1}$$

である．ここで，$p = (t+1)/m$ はタグが真である確率である．

(証明) $k' = k''$ であるキーについて，プロトコルに従って作ったタグが $\hat{t'}, \hat{t''}$ が共に偽の乱数である時，n 個の一様乱数が一致する確率は $1/n$ で与えられる．従って，真のタグである場合を考慮したタグ一致する確率は，

$$Pr(\hat{t'} = \hat{t''} \,|\, k' = k'') = p^2 + 2p(1-p)/n + (1-p)^2/n$$

である．ベイズの定理により，タグが一致した時のキーが同じであった確率は，

[*2] この検索タグが確定的である性質は，Boneh らの PEKS[1] においても同様である．

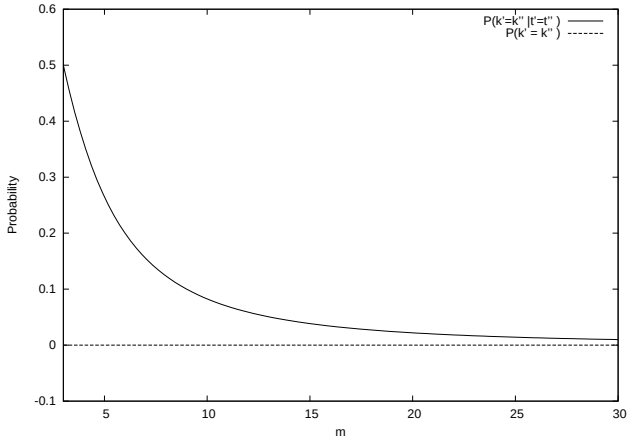

図 2 識別不能確率と Cloud サーバ数 m $(t = 3, n = 10^6)$

$$
\begin{aligned}
&Pr(k' = k'' \,|\, \hat{t}' = \hat{t}'') \\
&= \frac{Pr(\hat{t}' = \hat{t}'' \,|\, k' = k'')Pr(k' = k'')}{Pr(\hat{t}' = \hat{t}'')} \\
&= \frac{Pr(\hat{t}' = \hat{t}'' \,|\, k' = k'')Pr(k' = k'')}{Pr(\hat{t}' = \hat{t}'' \,|\, k' = k'') + Pr(\hat{t}' = \hat{t}'' \,|\, k' \neq k'')} \\
&= \frac{(p^2 + 2p(1-p)/n + (1-p)^2/n)(1/n)}{(p^2 + 2p(1-p)/n + (1-p)^2/n)(1/n) + (1/n)(1-1/n)} \\
&= \frac{(p^2 + 2p(1-p)/n + (1-p)^2/n)}{(p^2 + 2p(1-p)/n + (1-p)^2/n) + (1-1/n)}.
\end{aligned}
$$

補題を得る. (証明終)

命題 4.4 二つのタグ $\hat{t}', \hat{t}''$ が与えられた時，キーの識別可能性は無視できるほど小さい.

(証明) 条件がない時，二つのキーが一致する確率は，キーが n 個から一様に選ばれている時，$1/n$ である．一方，$\hat{t}' = \hat{t}''$ が与えられた時, 補題より $k' = k''$ である確率が与えられる．$p = (t+1)/m$ は m が大きくなるにつれて 0 に収束する．よって，$Pr(k' = k'' \,|\, \hat{t}' = \hat{t}'') = 1/n = Pr(k' = k'')$. 従って，$H(k' = k'' \,|\, \hat{t}', \hat{t}'') - H(k' = k'') = 0$ となり，情報理論的な秘匿性が示された. (証明終)

Cloud サーバ数 m に対するキー識別確率 $Pr(k' = k'' \,|\, \hat{t}' = \hat{t}'')$ の変化を図 2 に示す．m が大きくなるにつれて，$Pr(k' = k'')$ に漸近していることが示されている．決められたしきい値に対してサーバ数は制約がないので，通信コストをかけて安全性をあげることが出来る．なお，ここでは単一のサーバにおける識別可能性を議論したが，しきい値までの不正なサーバの結託に対する安全性も同様である.

4.2 性能

提案プロトコルが満たす性能を表 1 に整理する．(1) の安全性については，命題 4.1,4.2 に示す通り，提案する 3 方式とも満たしている．一方，(5) は，User がタグを Cloud に検索のタグを送信した時の安全性である．[1] は検索タグに合致したタグに該当する平文を別の方式で暗号しておく条件の元でこれを満たすが，提案方式では秘密分散されているので不正な Cloud が結託すれば復元されてしまうリスクがある．(4) の検索効率は，Boneh らの方式がサーバに登録されている全暗号文と検査をする必要があり，$O(n)$ のコストが掛かるのに対して，提案 3 方式は確定的に計算されるタグについてローカルにデータベースを構築することで，木構造で $O(\log n)$ のハッシュ表を用いると $O(1)$ のコストで検索が可能である.

4.3 動的なデータベースの更新の課題

提案方式では，n 個のキーと値を Owner が一元的に管理して一括してタグを計算する必要がある．すなわち，キーと値が静的であることを仮定している．キーが動的に変化する時，逐次的に対応するシェアを送受信すると Cloud にシェアの対応が分かってしまう危険性があり，何らかの措置を施す必要性がある.

5. おわりに

しきい値秘密分散法を用いて，キーによる検索を可能とする安全な分散データベースのプロトコルを提案した．提案方式は，確定的なタグ生成アルゴリズムを用いることにより，n 個のデータを $\log n$ のオーダーで検索を実現している．検索の際に送信するタグが与えられた時にデータが復元されてしまう確率は，分散するサーバの数の増加することで無視できるほど小さくできる.

参考文献

[1] Boneh, D., Crescenzo, G.D., Ostrovsky, R. and Persiano, G., “Public key encryption with keyword search”, *EUROCRYPT 2004*, LNCS, vol.3027, pp. 506-522 (2004).
[2] Boneh, D. and Waters, B., “Conjunctive, subset, and range queries on encrypted data”, *TCC 2007*, LNCS, vol.4392, pp. 535-554 (2007).
[3] Bellare, M., Boldyreva, A. and O’Neill, A. , “Deterministic and Efficiently Searchable Encryption”, *CRYPTO 2007*, LNCS, vol.4622, pp. 535-552 (2007).
[4] S. Pohlig and M. Hellman, “An Improved Algorithm for Computing Logarithms over $GF(p)$ and its Cryptographic Significance”, IEEE Transactions on Information Theory (24), pp. 106-110, 1078.
[5] Naor, M. and Pinkas, B., Oblivious Polynomial Evaluation SIAM Journal on Computing, 2006, Vol. 35, No. 5, pp. 1254-1281.
[6] 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.
[7] Lindell, Y., Pinkas, B, “ Privacy preserving data mining”, In *Advances in Cryptology CRYPTO 2000*, pp. 36-54, Springer, 2000.
[8] 松田 規, 伊藤 隆, 柴田 秀哉, 服部 充洋, 平野 貴人, “検索可能暗号の高速化と Web アプリケーションへの適用方式に関する提案”, マルチメディア，分散，協調とモバイル

表 1 プロトコルの性能

条件 要素技術	Boneh[1] 双線形写像	Alg. 1 基本方式 秘密分散	Alg. 2 検索権限制御 OPE	Alg. 3 キー秘匿検索 チャフ
(1) Cloud への秘匿	√	√	√	√
(2) Owner への秘匿	√	√	√	√
(3) User のアクセス制御	√	–	√	√
(4) 検索効率	$O(n)$	$O(\log n)$	$O(\log n)$	$O(\log n)$
(5) タグが与えられた時の秘匿	√	–	–	√

(DICOMO2013) シンポジウム, pp. 2067 - 2074, 2013.