検索可能秘密分散システムの実装

金子 曜大†

明治大学先端メディアサイエンス学科

1　はじめに

2014 年に起きたベネッセコーポレーションによる個人情報流出では，クラウドに保管されたデータがクラウドの管理者によって約 2895 万件の個人情報が漏洩した．このような内部犯行はクラウドを利用の大きな脅威である．

この課題に対して，伊藤らは，サーバ管理者がサーバに保管したデータを見ることのできないような検索可能秘密分散を提案している[1]．サーバ管理者が安易に保管されたデータを閲覧することができないのであれば，データの漏えいを防止することができる．本研究では[1]に基づき，検索可能秘密分散システムの実装を試みる．このシステムが実用されていれば，ベネッセの様な個人情報流出は防止できると期待する．

2　検索可能秘密分散

2.1 秘密分散

秘密 s を f(0) = s とする k-1 次の多項式をランダムで選ぶ．シェアを $v_i = f(i)$ と計算する．ここで i は $1 \leqq i \leqq n$ とする．$v_1, \ldots, v_n$ の内，任意の k 個から元の秘密 s を復元できる．本稿では、代表的なしきい値法として，Shamir の秘密分散[2]を用いる．

2.2 検索可能秘密分散

データオーナーは秘密分散の分散過程で生成したシェア v_i をタグ t とペアにして，サーバへ格納する(図１)．このときサーバ管理者はシェアを１つしか保持していないため，秘密についての情報は何一つわからない．オーナーは検索を許可するユーザへタグを送り，ユーザはタグを用いてサーバへ検索する(図２)．各サーバは検索に用いられたタグに一致するデータベース内のタグとペアのシェアをユーザへ送り返す．閾値 k 以上のシェア v_i を集めた時，秘密 s を復号できる．

表１：share, tag の例

Share	tag
5112269157363805079097791337877451	2601851102171492
1184679855936912875391425159784376527	161454209006801905
2456000997641148553668787913279307742	102351301251358507
1053572712126966018[1]367601424452391	201731012121481791
4763955583063717345137280972429759090	422141164409808502
9025158824252	017401201
(637 ビット)	(256 ビット)

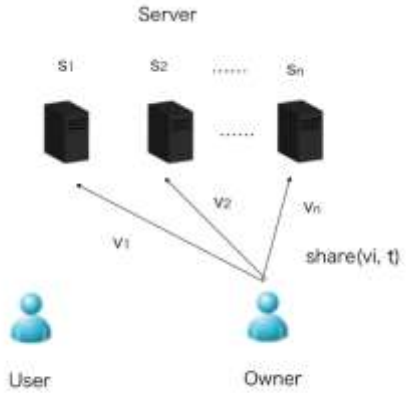

図１：分散過程

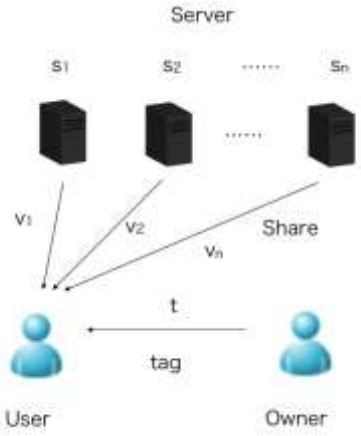

図２：復号過程

2.4 システム実装方法

クエリの検索は SQL に準拠したコマンドで行う．SQL 文を正規表現にて解析し，それぞれサーバの登録，秘密の分散等の処理を行っている．また，タグは「秘密」「サーバホスト名」「サーバユーザ名」の３つを文字列結合し，SHA-256 を用いてハッシュした値とする．実装したコマンドを表２に記載する．

2.5 秘密登録と復元

サーバ管理者から秘密を守りつつ，ユーザに情報を提供するシ

[1]Implementation of Keyword-Search Secret Sharing System

†KANEKO YODAI

Department of Frontier Media Science, Faculty of Interdisciplinary Mathematic Science, Meiji University

ステムを構築した.

使用手順を以下に記載する.

表２：コマンド一覧

1	REGISTRATION SERVER［サーバ登録名］	ssh の接続に成功し場合にのみサーバ登録データベースに新規登録する.
2	USE SERVER［サーバ登録名］	サーバ登録データベースに入力したサーバ名が存在する場合のみ実行する. サーバ登録データベースを使用状態に変更する.
3	DISUSE SERVER ［サーバ登録名］	サーバ登録データベースに入力したサーバ名が存在する場合のみ実行する. サーバ登録データベースを不使用状態に変更する.
4	DISPERSION DATA［秘密］	［秘密］を分散し，ssh を用いて有効なサーバに接続. シェアとタグを送信する.
5	SELECT KEY =［キー］	キーからタグを生成して，有効なサーバにタグを送信して，該当するシェアを得る. 得たシェアから秘密を復号する.

分散手続き

1. 表２の REGISTRATION を使用し，サーバを登録する.
2. 表２の USE を使用し，登録しているサーバの状態を変更する. このとき，表２の USE によって使用状態となっているサーバの数が秘密分散のしきい値 k にあたる.
3. 表２の DISPERSION を使用し秘密を分散する

復元手続き

1. 分散手続きの 1. 2. に同じ.
2. 表２の SELECT を使用し秘密を復号する

また表３に実験環境を示す.

表３：開発環境

言語	Java，シェルスクリプト(sh)
p の大きさ	192(10)桁
OS	OSX Yosemite
メモリ	8GB

３　評価

開発したシステムにおいて１文字あたりの平均復元時間を図３に示す. ここではそれぞれしきい値 k について，9000 回の処理時間を測定し，回帰直線 y=0.02841k+0.04913 得た.

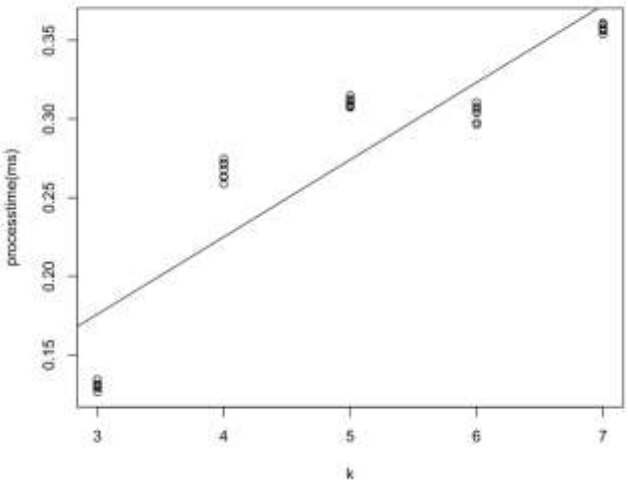

図３：しきい値 k についての復元処理時間

４　おわりに

本研究では,検索可能秘密分散システムを実装した. 各シェアから秘密について情報についての情報は漏れることはない. sql ベースのコマンドであることから慣れ親しみやすいと考えられる.また,その処理時間から実用性が高いと考える.

しかし，開発したシステムの秘密分散するデータの大きさは素数 p に依存する. データの大きさによって p を変更するなど，更なる工夫が必要である.

参考文献

[1]伊藤，牛田，山岡，及川，菊池：検索可能秘密分散方式の提案，情報処理学会研究報告，pp1-6，2012

[2] Benesse Holdings, Inc.，事故の概要｜お客様本部について｜ベネッセお客様本部，(http://www.benesse.co.jp/customer/bcinfo/01.html，2016 年 1 月参照)

[3] 黒澤馨，尾形わかは：現代暗号の基礎数理，pp116-119，2004