サーバによるキーワード推測攻撃に対して安全な検索可能公開鍵暗号の提案

齋藤 尭範[1]　中西 透[1]

概要：検索可能暗号とは、検索対象のデータとそのキーワードを暗号化したまま、ユーザにより検索されたキーワードを含むデータを検索できる暗号である。本研究では公開鍵ベースの検索可能暗号を対象とし、複数のユーザーがクラウドサーバーを介してメッセージを送受信することを考える。従来の検索可能暗号では、クラウドサーバによる検索キーワードを推測する攻撃が存在する。本研究では、サーバによるキーワード推測攻撃に対して安全な検索可能公開鍵暗号を提案する。提案方式では、暗号化ユーザーと検索ユーザーの間で放送型暗号を用いて秘密鍵を共有することにより、サーバが暗号化できないようにして、推測攻撃を防止する。そして実装に基づいてその性能を評価する。

キーワード：検索可能暗号, 放送型暗号, ペアリング

Proposal of a searchable public-key encryption secure against keyword guessing attack

Takanori Saito[1]　Toru Nakanishi[1]

Abstract: In the searchable encryption, for a ciphertext with keywords, a server can search an encrypted keyword given by a user. In this paper, we consider a public-key-based searchable encryption, where multiple users send messages to a single user through a cloud server. However, in the conventional searchable encryption, there is the server's attack to guess the searched keyword. In this paper, we propose a secure public-key-based searchable encryption against the keyword guessing attack. The proposed scheme disables the server from encrypting by sharing a secret key using a broadcast encryption between the searching user and the encrypting user. We additionally evaluate its performance on the basis of an implementation.

Keywords: searchable encryption, broadcast encryption, pairings

1. はじめに

近年、モバイル端末の普及とクラウドサービスの発展に伴い、データを外部のクラウドサーバに保存する機会が増えている。クラウド環境では、不正アクセスを防止するために、暗号化が必要となる。その際、クラウドサーバー内部での管理者によるデータ漏洩を防止するため、サーバーではなくユーザー自らで暗号化したデータをサーバーに保存することが望まれる。一方で、利便性のため、ユーザーがクラウドサーバ上のデータを検索できる必要がある。そこで、検索可能暗号が注目されている。検索可能暗号とは、検索対象のデータとそのキーワードを暗号化したまま、ユーザにより検索されたキーワードを含むデータを検索できる暗号である。このとき、キーワード自体もプライバシー保護のためサーバーに対して秘匿される。検索可能暗号は、プライバシーを保護したコンテンツ共有システム [1] などに応用されている。

本研究では公開鍵ベースの検索可能暗号 [2] を対象とし、複数のユーザーがクラウドサーバーを介してメッセージを

[1] 広島大学
Hiroshima University

送受信することを考える。このとき、メールサービスのように多くの暗号化ユーザーと、メッセージを受信する一人の検索ユーザが参加する。ここで、それぞれの暗号化ユーザはメッセージ送信時にその内容を表すキーワードを暗号化して付加し、サーバは保存する。そして、検索ユーザは、検索時に検索キーワードを暗号化してサーバに送信する。サーバは暗号化されたまま、メッセージ暗号文のキーワードとマッチするかを調べ、マッチしたメッセージのみを送信する。必要なメッセージおよび検索キーワードが暗号化されているためサーバにはキーワードが漏れずに、プライバシーが保護される。

従来の検索可能暗号では、クラウドサーバによる検索キーワードを推測する攻撃が存在する。公開鍵ベースの検索可能暗号では、キーワードの暗号化に公開鍵 pk を使用している。公開鍵は公開された情報であり、誰でも所持しているためキーワードの暗号化はサーバにも行える。こうしてキーワードのリストを用いて、サーバーは事前に可能性のあるキーワードすべてをマッチングすることができ、これにより検索ユーザが送信してきた暗号化されたキーワードを推測できる。

本研究では、サーバによるキーワード推測攻撃に対して安全な公開鍵ベースの検索可能暗号を提案する。提案方式では、暗号化ユーザーと検索ユーザーの間で秘密鍵を共有することにより、サーバが暗号化できないようにする。これにより、サーバーのキーワード推測攻撃を防止できる。秘密鍵の共有方法として放送型暗号を用いることで、多数の暗号化ユーザに対しても効率的に鍵共有する。そして実装に基づいてその性能を評価する。

本稿では、まず 2 章において、この研究で利用する暗号技術と従来方式の概要を示し、従来方式へのキーワード推測攻撃について述べる。3 章では、そのキーワード推測攻撃へに対して安全な検索可能暗号方式を提案する。第 4 章では提案方式の実装に基づいた評価を行い、第 5 章で本論文をまとめる。

2. 利用する暗号技術と従来方式

2.1 双線形写像

本研究では、双線形写像が構成できる楕円曲線上の群を利用する。G_1, G_2, G_T を素数位数 p の巡回群とする。そして、g_1, g_2 をそれぞれ G_1, G_2 の生成元とする。このとき、双線形写像 $e : G_1 \times G_2 \to G_T$ は以下の性質を満たす。

●双線形性：$\forall u \in G_1, \forall v \in G_2, \forall a, b \in Z_p$ において、

$$e(u^a, v^b) = e(u, v)^{ab}$$

●非退化性：$e(g_1, g_2) \neq 1_T$

ここで 1_T は G_T の単位元である。

このような双線形写像は、楕円曲線上に基づいた群上のペアリングにより構成できる。この論文では簡単化のため $G_1 = G_2 = G$ の場合の双線形写像を用いて方式を示す。

2.2 放送型暗号

放送型暗号では、暗号化ユーザーは、指定したユーザー集合のユーザのみが復号可能なようにコンテンツを暗号化できる。このユーザ集合は暗号化毎に指定できるため、ユーザ失効に柔軟に対応できる。[3] において、任意の不正者による結託攻撃に対して安全である効率的な放送型暗号方式が提案されている。この方式は秘密鍵および暗号文が定数サイズで、公開鍵サイズはユーザ数 n に比例する。本研究ではこの方式を利用するため、以下にアルゴリズムを示す。

鍵生成：$g \in G$、$a, r \in Z_p$ をランダムにとる。
すべての $1 \leq i \leq n$ に対して $g_i := g^{a^i}, v := g^r$ として

$$(g, g_1, g_2 \cdots g_n, g_{n+2}, ..., g_{2n}, v)$$

を公開鍵とする (g_{n+1} だけ抜けていることに注意する)。
各ユーザ $U_i (1 \leq i \leq n)$ の秘密鍵を $d_i := g_i^r = g^{ra^i}$ とする。

暗号化：$t \in Z_p$ をランダムにとり、$s := e(g_1, g_n)^t$
をセッション鍵として任意の共通鍵暗号により
コンテンツ m を暗号化する。
復号を許可したいユーザの集合 $S \subset \{1, ..., n\}$ に対して

$$H := (g^t, (v \cdot \textstyle\prod_{j \in S} g_{n+1-j})^t)$$

を計算し、$(S, H, Enc(s, m))$ を送信する。
ここで $Enc(s, m)$ はコンテンツ m に対する鍵 s での共通鍵暗号化である。

復号：ユーザ U_i は暗号文 $(S, H, Enc(s, m))$ に対して公開鍵と自身の秘密鍵 d_i を用いて、$H = (C_0, C_1)$ において、

$$s = e(g_i, C_1) / e(C_0, d_i \cdot \textstyle\prod_{j \in S \setminus \{i\}} g_{n+1-j+i})$$

を求め、$Dec(s, Enc(s, m))$ によりコンテンツ m を得る。
ここで、$Dec(s, Enc(s, m))$ とは暗号文 $Enc(s, m)$
に対する秘密鍵 s での復号である。

以下に示すように、指定されたユーザが復号時に得た s は暗号化時の s と等しい。

$$\frac{e(g_i, C_1)}{e(C_0, d_i \cdot \prod_{j \in S \setminus \{i\}} g_{n+1-j+i})}$$

$$= \frac{e(g^{a^i}, v^t) \cdot e(g^{a^i}, (\prod_{j \in S} g^{a^{n+1-j+i}})^t)}{e(g^t, g^{ra^i}) \cdot e(g^t, \prod_{j \in S \setminus \{i\}} g^{a^{n+1-j+i}})}$$

$$= \frac{e(g^{a^i}, g)^{tr} \cdot \prod_{j \in S} e(g, g^{a^{n+1-j+i}})^t}{e(g^{a^i}, g)^{tr} \cdot \prod_{j \in S \setminus \{i\}} e(g, g^{a^{n+1-j+i}})^t}$$

$$= e(g, g^{a^{n+1}})^t = e(g_1, g_n)^t = s$$

2.3 従来の検索可能暗号方式

2.3.1 検索可能暗号の概要

近年、暗号化したままキーワードをマッチングしキーワードのプライバシーを保護する公開鍵ベースの検索可能暗号 (Public-key Encryption with Keyword Search:PEKS) が提案されている [2]。

検索可能暗号では、図 2.2 に示すように、事前に検索ユーザーが公開鍵と秘密鍵を生成して、公開鍵を他の参加者に公開し、秘密鍵を安全に保持しておく。暗号化ユーザーは送信したいメッセージにおいて、そのキーワードを暗号化した $Etag$ を生成して、暗号化したメッセージとともにサーバーに送信する。一方で、検索ユーザーは検索したいキーワードをサーバーに登録するために $Trapdoor$ を生成しサーバーに送信する。

サーバーは $Etag$ と $Trapdoor$ に Test 関数を適用することにより、暗号化されたキーワードがマッチしているか、確認することができる。キーワードは暗号化されているため、サーバーがマッチングの際に、キーワードが何かを知ることはできない。Test 関数はもし、$Etag$ と $Trapdoor$ がマッチした場合は $True$ を返し、そうでなければ $false$ を返す。こうして、$True$ だった場合、サーバーは検索ユーザに、対応する暗号化メッセージを送信することができる。

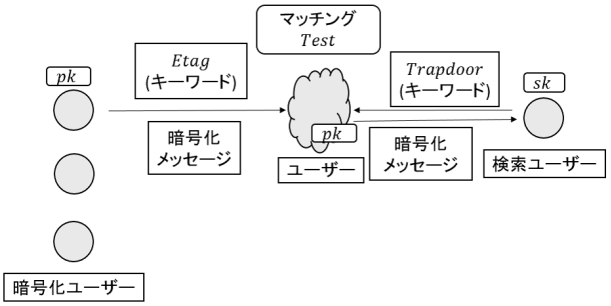

図 1 従来の検索可能暗号の概要

例えば、暗号化ユーザー 1 が送信する際に「baseball」をタグとして添付し、暗号化ユーザー 2 は「tennis」、暗号化ユーザー 3 は「soccer」をタグとして添付してメッセージをサーバに送った場合、検索ユーザーがキーワード「baseball」を検索したならば、メッセージとキーワードを明かにすることなく、検索ユーザーに「baseball」のメッセージを配信できる。

以下に [2] の方式を示す。$H(x)$ をハッシュ関数とする。秘密鍵 sk はランダムな $x \in Z_p^*$ であり、公開鍵 pk は $h = g^x$ で計算される。
このときキーワード w に対して、$Etag$ と $Trapdoor$ は乱数 $r \in Z_p$ に対して下記のように計算される。

$$Etag = (E_1, E_2) = (g^r, e(H(w), h^r))$$
$$Trapdoor = H(w)^x$$

また、$Test$ 関数は、

$$e(Trapdoor, E_1) = E_2$$

を調べ、等式が成り立つときのみ、$Etag$ と $Trapdoor$ のキーワードが同じだと判断する。
このとき $h = g^x$ 及び双線形性より、この等式は

$$e(Trapdoor, E_1) = e(H(w)^x, g^r) = e(H(w), g)^{xr}$$
$$E_2 = e(H(w), h^r) = e(H(w), g)^{xr}$$

と変形されるため、同一のキーワードなら等式が成り立つ。

2.3.2 サーバーによるキーワード推測攻撃

本節では、従来方式において、サーバーによる検索ユーザーの暗号化キーワードの推測攻撃を示す (図 2.3)。従来方式では、暗号化ユーザが $Etag$ を生成する際、公開鍵 pk を使用する。しかし、公開鍵は公開されている情報のため、$Etag$ は誰でも作ることができてしまう。こうしてサーバーは事前に可能性のあるキーワードすべての $Etag$ を計算しておき、検索ユーザが作り送ってきた $Trapdoor$ をマッチングできる。これにより、サーバーは秘密鍵 sk を知ることなくして、検索ユーザーのキーワードを推測することが可能となってしまう。

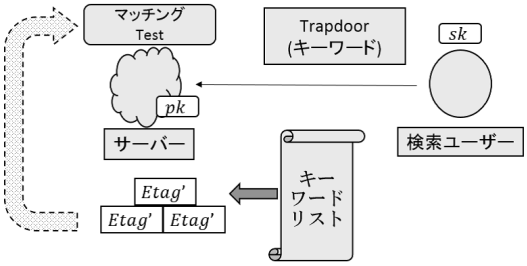

図 2 キーワード推測攻撃

3. 提案方式

3.1 構成方針

前章で示した攻撃は、$Etag$ が公開鍵を用いて暗号化されるため誰でも生成できてしまうことに起因している。

本研究では、Etag の生成において共有鍵 MK を導入することを考える (図 3)。共有鍵 MK は検索ユーザと暗号化ユーザのみ事前に共有した秘密鍵である。また、Test 関数が正しく動作するように、Trapdoor 生成時にもこの共有鍵を使用する。
共有鍵 MK を導入した場合の $Etag, Trapdoor$ は以下のようになる。ここに MK は検索ユーザ・暗号化ユーザー間の共有鍵であり、乱数 $MK \in Z_p$ の値とする。

$$Etag \quad = (E_1, E_2) = (g^r, e(H(w, MK), h^r))$$
$$Trapdoor = H(w, MK)^x$$

$Test$ 関数は従来方式と同じく

$$e(Trapdoor, E_1) = E_2$$

の検証をするものとする。
このとき Test 処理の式の左辺を考えると、e の双線形性から、

$$\begin{aligned} e(H(w, MK)^x, g^r) &= e(H(w, MK), g^r)^x \\ &= e(H(w, MK), g^{xr}) \\ &= e(H(w, MK), h^r) \end{aligned}$$

となり、右辺と等しくなるため、共有鍵 MK を付加しても Test 関数には影響ない。
次に、共有鍵 MK の効率的な共有法について考える。本研究では、公開鍵ベースの検索可能暗号 [2] のモデルと同様に、多数の暗号化ユーザと単一の検索ユーザを考えている。単純に DH 鍵共有法などの二者間鍵共有を用いた場合、各暗号化ユーザと単一ユーザ間で鍵共有するために多数の共有鍵が必要となり効率が悪い。そこで 2 章で示した放送型暗号を利用する。放送型暗号では、単一のセッション鍵を用いて限定した登録ユーザーのみに効率的に共有鍵を配布することができる。登録者数が増えても、暗号文サイズは増大しない。このことから、効率的な共有鍵配布に放送型暗号は使えるため、それを利用して、サーバに $Etag$ の生成を許さない検索可能暗号を構成する (図 4)。

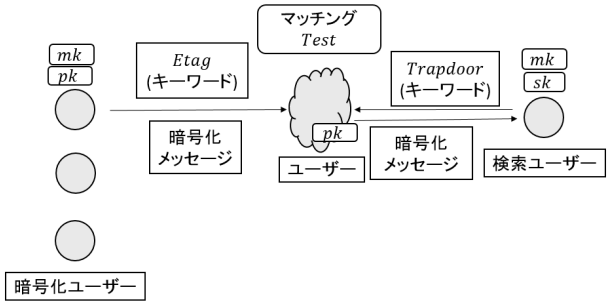

図 3 共有鍵を導入した検索可能暗号

3.2 提案方式のモデル

キーワード推測攻撃に対して安全な検索可能公開鍵暗号のモデルを示す。
-$CenterKeyGen(n)$: 鍵センターによって行われ、暗号化ユーザの数 n を入力として、暗号ユーザ U_i の秘密鍵 Csk_i と公開鍵 Cpk を出力する。
-$SUKeyGen(S, Cpk)$: 検索ユーザによって行われ、Cpk と暗号化ユーザグループ $S \subset \{1, , , n\}$ を入力として、秘密鍵 $SUsk$ と S に対応した公開鍵 $SUpk$ を出力する。
-$Trapdoor(SUsk, w)$: 検索ユーザによって行われ、キーワード w の $Trapdoor$ を出力する。
-$Etag(S, Cpk, SUpk, Csk_i, w)$: S で指定された各暗号化ユーザ U_i によって行われ、秘密鍵 Csk_i を使ってキーワード w の $Etag$ を出力する。
-$Test(Etag, Trapdoor)$: サーバーによって行われ、$Etag, Trapdoor$ を入力とし、それらのキーワードが等しいなら $True$、そうでなければ $False$ を出力する。

従来方式 [2] からは、$CenterKeyGen$ が導入されており暗号化ユーザーを特定するための、秘密鍵 Csk_i が公開鍵 Cpk とともに生成される。また $SUKeyGen$ により、暗号化ユーザ集合 S に対応した公開鍵 $SUpk$ が出力されている。
従来の安全性であるキーワード秘匿性に加えて、以下の $Etag$ の偽造不可能性を考える。S を入力したときの $SUKeyGen$ の出力 $SUpk$ と、対応する $SUsk$ から生成された $Trapdoor$ に対して、$i \in S$ となる Csk_i をもたない任意の PPT 攻撃者は、$Test(Etag, Trapdoor)$ が正しく動作するような、$Etag$ を生成できない。これにより、S で指定されていないサーバは $Etag$ を生成できず、キーワード推測攻撃を防止できる。

3.3 提案方式の構成

CenterKeyGen:
(1) g を群 G のランダムな元とし、$a, r \in Z_p$ をランダムにとる。また、$i = 1, ..., n, n+2, ...2n$ に対して、$g_i := g^{a^i}, v := g^r$ として

$$(g, g_1, g_2 \cdots g_n, g_{n+2}, ..., g_{2n}, v)$$

を公開鍵 Cpk とする.

(2) 暗号化ユーザ U_i の秘密鍵 Csk_i を $Csk_i := d_i = g_i^r = g^{ra^i}$ とし、U_i に安全に配布しておく。

SUKeyGen:
(1) $x \in Z_p$ をランダムにとり、$h = g^x$ を計算する。

(2) $t \in Z_p$ をランダムにとり、$s = e(g_1, g_n)^t$ をセッション鍵 (暗号化ユーザとの共有鍵) とする。暗号化を許可するユーザの集合 $S \subset \{1, , , n\}$ に対して

$$H := (g^t, (v \cdot \prod_{j \in S} g_{n+1-j})^t)$$

とする。

(3) 公開鍵 $SUpk$ として、(h, S, H) を公開する。秘密鍵 $SUsk$ は (x, s) となる。

Trapdoor:

(1) 秘密鍵 $SUsk = (x, s)$ を使って、キーワード w に対し、

$$Trapdoor = H(w, s)^x$$

を計算する。

Etag:

(1) 暗号化ユーザ U_i は公開鍵 $Cpk, SUpk = (h, S, H = (C_0, C_1))$ と自身の秘密鍵 $Csk_i = d_i$ を使って

$$s = e(g_i, C_1)/e(C_0, d_i \cdot \prod_{j \in S \setminus \{i\}} g_{n+1-j+i})$$

を求める。

(2) 自ら生成したランダムな $r \in Z_p$ を使って

$$Etag = (E_1, E_2) = (g^r, e(H(w, s), h^r))$$

を計算する。

Test:

(1) それぞれから受け取った $Etag = (E_1, E_2), Trapdoor$ がマッチするかテストするために、

$$e(Trapdoor, E_1) = E_2$$

が成り立つかチェックする。成り立つなら、キーワードはマッチしているとする。

3.4 安全性

3.2 節で示したように、サーバによるキーワード推測攻撃を防ぐため、$i \in S$ となる Ck_i を持たない者は S に対応した $Etag$ を生成できない必要がある。提案方式では、$i \in S$ である $Csk_i = d_i$ を持たない者は $SUpk$ 中の放送型暗号 H を復号して s を得ることができない。こうして、攻撃者は、S に対応した $SUsk$ 中の s から生成される $Trapdoor = H(w, s)^x$ とマッチされる $Etag$ を生成できない。

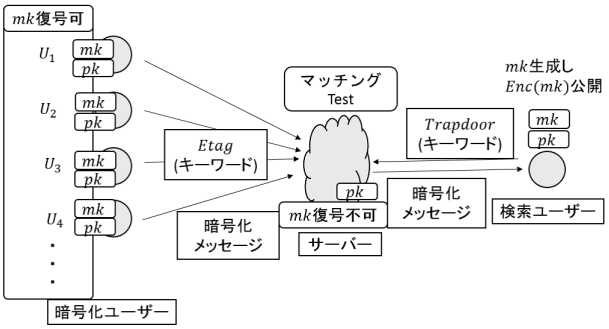

図 4 提案方式の概要

4. 実装による評価

表 1 実装環境

OS	Ubuntu 14.04
CPU	Intel Core i5-4460(3.20GHz)
メモリ	7.80GB
多倍長ライブラリ	GMP6.1.0
ペアリングライブラリ	ELIPS

4.1 実装環境

提案方式の実用性を評価するために表 1 に示す環境の PC において実装し、処理時間を測定した。実装においては、楕円曲線上の群演算とペアリング演算が必要となる。本研究では、文献 [4] の ELIPS ライブラリを利用した。ELIPS ライブラリは C 言語で実装されており、多倍長演算に GMP を使用している。

4.2 処理時間

実装した各アルゴリズムの測定結果より、$Trapdoor, Test$ のそれぞれの平均時間は 1.4[ms], 3.9[ms] であり実用的である。$Etag$ の処理は、暗号化ユーザ数に依存しているため、そのユーザ数 200 まで変化させて測定した。その処理時間は図 6 のようになっている。ユーザー数に依存して処理時間は増大するものの、ユーザー数が 200 のとき 17.1[ms] であるため、実用的であると考えられる。提案方式では、$Etag$ 生成時に放送型暗号の復号を行っているものの、ユーザ数に比例した乗算 (楕円加算) のみが必要なため、効率的に $Etag$ が生成できると考えられる。

5. おわりに

本研究では、検索サーバーの暗号化キーワードを推測する攻撃法に対して安全な検索可能暗号方式を提案した。提案方式では、放送型暗号を利用して、多数の暗号化ユーザーと単一の検索ユーザー間で効率的な鍵共有を行うことにより、サーバの推測攻撃を防止している。さらに、PC

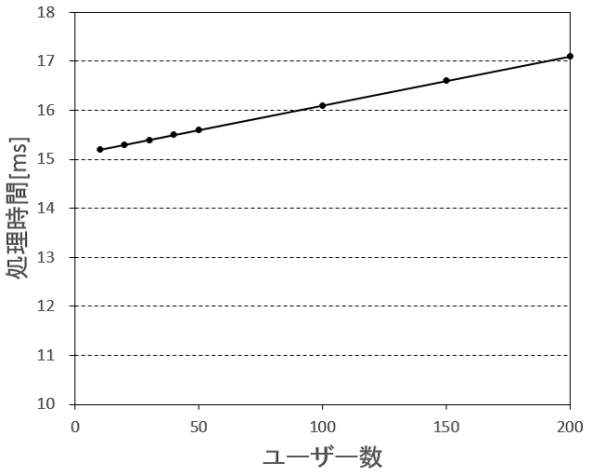

図 5 *Etag* の処理時間

上で実装を行い、各処理時間を測定した。暗号化ユーザ数に依存する *Etag* 生成時間において、ユーザ数が 200 の場合でも 17.1[ms] 程度で動作し、実用的であると考える。今後の課題として、検索ユーザーが多数存在するようなユーザグループ内でのコンテンツ共有の場合における方式の検討が考えられる。

参考文献

[1] M.R.Asghar, A.Gehani, B.Crispo, and G.Russello ”PIDGIN: Privacy-preserving Interest and Content Sharing in Opportunistic Networks, ” ACM-ASIACCS2014, pp.135-146, 2014.

[2] D.Boneh, G,Vrescenzo, R.Ostovsky, and G.Persiano, ” Public Key Encryption with Keyword Search, ” EUROCRYPT2004, pp.506-522, 2004.

[3] D.Boneh, C.Gentry and B.Warters, ”Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys ” Advances in Cryptotology -CRYPTO 2005, pp.258-275, 2005.

[4] M.Akane, Y.Nogami, and Y.Morikawa,” Fast Ate Pairing Computation of Embadding Degree 12 Using Subfield Twisted Elliptic Curve,” IEICE Trans. Fundamentals, Vol.E92-A, No.2, pp.508-516, 2009.