鍵更新機能付き検索可能暗号：公開鍵更新モデルによる実現

松崎 なつめ[1]　穴田 啓晃[1]　渡邉 洋平[2]

概要：検索可能暗号は暗号化したままキーワードの検索が可能な高機能暗号技術であり，これまでに盛んに研究されてきた．松崎らは，検索可能暗号の鍵更新に着目し，(SCIS 2017, ISEC2017-5) によってモデルと要件を整理した．その特徴的な要件の 1 つは，可用性を目的として，更新前の公開鍵を用いて生成された暗号文を対象に，更新後の新しい秘密鍵を用いて検索可能とすることである．本稿では，要件に基づき具体的構成を提案する．

キーワード：検索可能暗号，公開鍵更新

Key-Updatable Public-key Encryption with Keyword Search: The Case of Public-key Update Model

Natsume Matsuzaki[1]　Hiroaki Anada[1]　Yohei Watanabe[2]

Abstract: Public-key encryption with keyword search (PEKS) enables one to search keywords stored in a server while preserving the confidentiality of the keywords. Recently, Matsuzaki et al. (SCIS 2017, ISEC 2017-5) clarified requirements for key-updatable PEKS(KU-PEKS), which is PEKS with key-updating functionality. One of its characteristic requirements for the purpose of availability is that a receiver with updated new secret key can search the ciphertexts containing the keyword generated using the public key before updating. In this paper, we propose concrete KU-PEKS scheme.

Keywords: searchable encryption, public-key update

1. はじめに

境、大岸、笠原 [8] が 2000 年に、また、Boneh と Franklin [3] が 2001 年，楕円曲線上のペアリングの双線型性を本質的に用い，任意の個人識別情報（ID）を公開鍵として利用可能な暗号方式（ID ベース暗号）を理論的に実現した．2004 年には Boneh ら [2] が，暗号化したままキーワードの検索が可能な公開鍵暗号方式（検索可能公開鍵暗号，以下，検索可能暗号）を提案する等，ペアリングを数学的構造として利用したいわゆる高機能暗号が盛んに研究されてきた．一方，これらの高機能暗号の実用面では，アメリカ国立標準技術研究所（National Institute of Standards and Technology）による標準化の動き [5] [6] がある．しかしながら，高機能暗号が普及するためには安全性評価等の課題が未だ存在するとされている [5].

我々は，高機能暗号の上記の課題に関し，公開鍵暗号のユーザ個別に配付される秘密鍵の更新（以下，鍵更新）に着眼する．松崎らの論文（SCIS2017[9]，ISEC2017-5[10]）では，高機能暗号のプリミティブの中でも検索可能暗号を採り上げ，クラウドに預託した暗号化キーワードのデータベースに対し暗号化クエリで所望の情報を検索する前提で，鍵更新のモデルと要件を検討してきた．

本稿では，上記要件に基づき具体的構成を提案する．なお，具体的構成については，以下の 1)2) の 2 つのアプローチで検討を進めており，本稿は，そのうちの 1) となる．2) のアプローチについては，[12] を参照のこと．

[1] 長崎県立大学　情報システム学部　情報セキュリティ学科
Department of Information Security, University of Nagasaki
[2] 電気通信大学 大学院情報理工学研究科
Graduate School of Informatics and Engineering, The University of Electro-Communications
日本学術振興会特別研究員 (PD)

1) 公開鍵更新モデル：ユーザの秘密鍵更新に伴い，対応する公開鍵も更新するモデルである．公開鍵の定期的な配布と管理が必要となる．
2) 鍵隔離暗号モデル：ユーザの秘密鍵は更新する一方，対応する公開鍵は更新せず固定するモデルである．公開鍵の定期的な配布と管理は不要となる．

以下では，まず 2 章で提案の準備を行い，3 章で松崎らにより整理された更新機能付き検索可能暗号のモデルと要件，および安全性の定義を示す．4 章で具体的な方式を提案し，5 章にその安全性について説明する．

2. 準備

ここでは，提案の準備として，双線型写像と mDBDH 問題について説明する．

双線型写像. q を素数，$\mathbb{G}_1$，$\mathbb{G}_2$，$\mathbb{G}_T$ を位数 q の巡回群，g_1 と g_2 をそれぞれ $\mathbb{G}_1$ と $\mathbb{G}_2$ の生成元とし，e は効率的に計算可能な双線形写像 $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ とする．双線形写像生成器 $\mathcal{G}$ を，λ を入力し，$(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$ を出力する多項式時間アルゴリズムとする．e は以下の性質を持つ: 任意の $u, u' \in \mathbb{G}_1$ 及び $v, v' \in \mathbb{G}_2$ に対し，$e(uu', v) = e(u, v)e(u', v)$ 及び $e(u, vv') = e(u, v)e(u, v')$ が成り立つ．本稿では，$\mathbb{G}_1 = \mathbb{G}_2 = \mathbb{G}$，$g_1 = g_2 = g$ とする，対称ペアリングを仮定する．

Modified Decisional Bilinear Diffie Hellman(mDBDH 問題). $\mathbb{G}, \mathbb{G}_T, q, g, e$ を上記のとおりとする．ランダムな $a, b, c \in \mathbb{Z}_q$ と，$Q \in \mathbb{G}_T$ に対して，(g, g^a, g^b, g^c) と Q が与えられたとき，Q がランダムの値か，$Q = e(g, g)^{a \times b/c}$ を満たすかを判定する問題である．この問題は，Decisional Bilinear Diffie Hellman(DBDH 問題）と等価であることが知られている [4]．

3. 鍵更新機能付き検索可能暗号

3.1 モデルと要件

本節では，松崎ら（SCIS2017[9], ISEC2017-5[10]）により整理された鍵更新機能付き検索可能暗号（Key-Updatable Public-key Encryption with Keyword search : KU-PEKS）の概念モデルと要件を概説する．

図 1 に KU-PEKS の概念モデルを示す．KU-PEKS は送信クライアント，受信クライアント，クラウドの 3 つのエンティティからなる．また，通常の検索可能暗号における (1) 預託フェーズと，(3) 検索フェーズに加え，受信クライアント側で秘密鍵を更新して，対応した更新鍵を生成する (2) 鍵更新フェーズからなる．なお，ここでは検索キーワードを含むインデックスの暗号化と検索キーワードの暗号化について焦点をあて，ドキュメント自身の暗号化と復号については省略するものとする．

(1) 預託フェーズでは，送信クライアントは，受信クライアントの公開鍵を用いて，検索キーワードを含むインデックスを暗号化してクラウドに預託する．

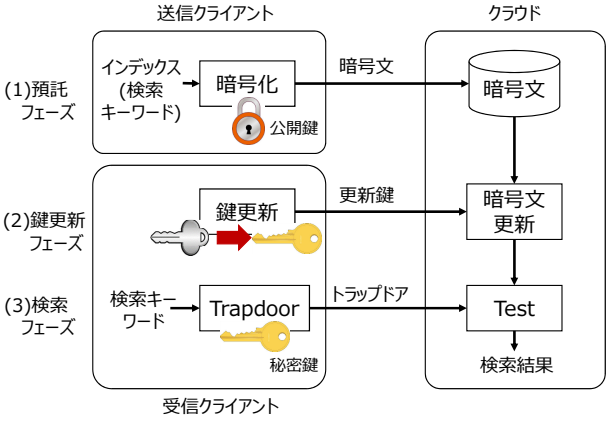

図 1 鍵更新機能付き検索可能暗号（KU-PEKS）の概念モデル

(2) 鍵更新フェーズでは，受信クライアントは，新しい秘密鍵を生成し，生成した鍵と従来持っていた秘密鍵を用いて更新鍵を生成し，クラウドに送信する．受信クライアントは，従来持っていた秘密鍵を削除する．一方，クラウドは更新鍵を用いて，預託フェーズで預かった暗号文を更新する．

(3) 検索フェーズでは，受信クライアントは，秘密鍵を用いて検索キーワードを暗号化したトラップドアを求め，クライドに送信する．クラウドは，暗号文とトラップドアを用いて検索してその結果を出力する．

KU-PEKS は，以下の 7 個のアルゴリズムからなる．

KU-PEKS= (Setup, KeyGen, ReKeyGen, Enc, ReEnc, Trapdoor, Test)

- $\mathsf{Setup}(1^\lambda) \to PP$
  セキュリティパラメータ 1^λ を入力として，公開パラメータ PP を出力する．公開パラメータ PP は以降のすべてのアルゴリズムの入力となるが，簡単のため省略する．
- $\mathsf{KeyGen}(1^\lambda) \to (sk, pk)$
  セキュリティパラメータ 1^λ を入力として，クライアントの各バージョンでの鍵ペア（秘密鍵 sk と公開鍵 pk）を出力する．
- $\mathsf{ReKeyGen}(sk_i, sk_{i+1}, state_i) \to rk_{i \to i+1}, state_{i+1}$
  受信クライアントにより実行され，隣り合ったバージョンの秘密鍵から，更新鍵を出力する．鍵の添字は，バージョン番号を示す．
- $\mathsf{Enc}(pk_i, w) \to C_{w,i}^{(0)}$
  メッセージの送信クライアントによって実行され，バージョン i の受信クライアントの公開鍵と検索キーワード w を入力して検索用の暗号文（暗号 index）を出力する．
- $\mathsf{ReEnc}(rk_{i+k \to i+k+1}, C_{w,i}^{(k)}) \to C_{w,i}^{(k+1)}$
  クラウドにより実行され，更新鍵を用いて暗号文を更新する．暗号文の上の添え字は，更新の回数を

示す．

- $\mathsf{Trapdoor}(sk_j, w', state_j) \to T_{w',j}$
  受信クライアントによって実行され，バージョン j の秘密鍵 sk_j と検索キーワード w' を入力とし，トラップドア $T_{w',j}$ を出力する．
- $\mathsf{Test}(C_{w,i}^{(j-i)}, T_{w',j}) \to 1\ or\ 0$
  クラウドによって実行され，バージョンが一致した暗号文と，トラップドアを入力とし，もし $w = w'$ であれば 1 を出力する．そうでなければ 0 を出力する．

KU-PEKS の要件は以下の 4 点である．

要件 1：受信クライアントの新しい秘密鍵は，古い鍵，および公開の情報から求められないこと．

要件 2：古い鍵は受信クライアントから削除すること．

要件 3：クラウドでは，更新前の古い公開鍵で暗号化された暗号文を対象に，更新後の新しい鍵で生成したトラップドアで検索できること．この要件は，モデルの可用性を目的としたものであり，受信クライアントが，保持する新しい鍵だけで，古い暗号文をも対象として検索可能とする．なお，この要件は，古い公開鍵で暗号化された暗号文を，復号することなく，新しい公開鍵で暗号化された暗号文に更新可能とすることで満たされる．

要件 4：クラウドでは，更新後の新しい公開鍵で暗号化された暗号文を対象に，更新前の古い鍵で生成したトラップドアで検索できないこと．この要件は，クラウドの攻撃に対する安全性を目的としたものである．本モデルでは，クラウドは正しく検索するが，漏洩の可能性のある古い鍵を入手した悪意のクライアントと結託する場合を考慮する．この場合，更新前の古い公開鍵で暗号化された暗号文を対象とした攻撃は防ぐことは難しいが，本要件により，更新後の新しい公開鍵で暗号化された暗号文を対象とした攻撃を防ぐ．なお，この要件は，新しい公開鍵で暗号化された暗号文を，古い暗号文には変換できないことで満たされる．

3.2 安全性定義

本節では，KU-PEKS 方式の安全性に関し，3 つの定義（正当性 : Correctness, 計算量的一貫性 : Computational Consistency, 選択キーワード攻撃と鍵漏洩に対する識別不可能性： IND-KU-CKA) を示す．

3.2.1 Correctness

KU-PEKS の正当性 : Correctness を，暗号レベルに対応して次のように定義する．なお，ここで暗号レベル l(l は 0 以上の整数）は，バージョン i の公開鍵を用いて生成した暗号 index を入力として ReEnc を l 回繰り返して暗号文を更新する場合を示す．

暗号レベル 0 の場合　$(l = 0)$

任意のバージョン i に対して，以下が成り立つとき Correctness が満たされる．

$$
\begin{aligned}
&\Pr[\ (sk_i, pk_i) \leftarrow \mathsf{KeyGen}(1^\lambda),\\
&w \in_R \mathrm{KeywordSpace}(1^\lambda),\\
&C_{w,i}^{(0)} \leftarrow \mathsf{Enc}(pk_i, w),\\
&T_{w,i} \leftarrow \mathsf{Trapdoor}(sk_i, w, state_i)\\
&: \mathsf{Test}(C_{w,i}^{(0)}, T_{w,i}) = 1]\\
&= 1
\end{aligned}
$$

暗号レベル l の場合 $(l \geq 1)$

任意のバージョン i に対して，以下が成り立つとき Correctness が満たされる．

$$
\begin{aligned}
&\Pr[\ ((sk_k, pk_k) \leftarrow \mathsf{KeyGen}(1^\lambda))_{k=i,\ldots,i+l},\\
&(rk_{i+k \to i+k+1} \leftarrow \mathsf{ReKeyGen}(sk_{i+k}, sk_{i+k+1}))_{k=0,\ldots,l-1},\\
&w \in_R \mathrm{KeywordSpace}(1^\lambda),\\
&C_{w,i}^{(0)} \leftarrow \mathsf{Enc}(pk_i, w),\\
&(C_{w,i}^{(k+1)} \leftarrow \mathsf{ReEnc}(rk_{i+k \to i+k+1}, C_{w,i}^{(k)}))_{k=0,\ldots,l-1},\\
&T_{w,i+l} \leftarrow \mathsf{Trapdoor}(sk_{i+l}, w, state_{i+l})\\
&: \mathsf{Test}(C_{w,i}^{(l)}, T_{w,i+l}) = 1]\\
&= 1
\end{aligned}
$$

ここで，上記の確率は KeyGen, ReKeyGen, KeywordSpace, Enc, Trapdoor, ReEnc, Test の各々で用いられるランダムネスを全て亘るものとする．

3.2.2 Computational Consistency

Correctness が真陰性が起こらないことを保証するのに対し，計算量的一貫性 : Computational Consistency は多項式アルゴリズム $\mathcal{A}$ が偽陽性を起こすような検索キーワードを見つける確率が十分小さいことを保証する．より具体的には，Abdalla ら [1] の定義にならい，暗号レベルに対応して，次の実験と優位性を用いて定義する．**暗号レベル 0 の場合**　$(l = 0)$

任意のバージョン i に対して，次の実験を定義する．

$$
\begin{aligned}
&\mathbf{Expr}_{\text{KU-PEKS},\mathcal{A}}^{\text{consistency\_0}}(\lambda)\\
&(sk_i, pk_i) \leftarrow \mathsf{KeyGen}(1^\lambda)\\
&(w, w') \leftarrow \mathcal{A}(pk)\\
&C_{w,i}^{(0)} \leftarrow \mathsf{Enc}(pk_i, w)\\
&T_{w',i} \leftarrow \mathsf{Trapdoor}(sk_i, w', state_i)\\
&\text{If } w \neq w' \text{ and } \mathsf{Test}(C_{w,i}^{(0)}, T_{w',i}) = 1\\
&\qquad\qquad \text{then return 1 else return 0}
\end{aligned}
$$

定義 1 : Computational Consistency(暗号レベル 0)

実験 $\mathbf{Expr}^{\text{consistency\_0}}_{\text{KU-PEKS},\mathcal{A}}(\lambda)$ におけるアルゴリズム $\mathcal{A}$ の KU-PEKS に対する優位度（advantage）を次のように定義する．

$$\mathbf{Adv}^{\text{consistency\_0}}_{\text{KU-PEKS},\mathcal{A}}(\lambda) \stackrel{\text{def}}{=} \Pr[1 \leftarrow \mathbf{Expr}^{\text{consistency\_0}}_{\text{KU-PEKS},\mathcal{A}}(\lambda)]. \tag{1}$$

実験 $\mathbf{Expr}^{\text{consistency\_0}}_{\text{KU-PEKS},\mathcal{A}}(1^\lambda)$ において，確率的多項式時間の任意のアルゴリズム $\mathcal{A}$ に対し，優位度 $\mathbf{Adv}^{\text{consistency\_0}}_{\text{KU-PEKS},\mathcal{A}}(\lambda)$ が λ の無視可能な関数であるとき，KU-PEKS は暗号レベル 0 において Computational consistency を持つと定義する．

暗号レベル l の場合　$(l \geq 1)$　任意のバージョン i と，暗号レベル l に対して，次の実験を定義する．

$\mathbf{Expr}^{\text{consistency\_}l}_{\text{KU-PEKS},\mathcal{A}}(\lambda)$
$((sk_k, pk_k) \leftarrow \mathsf{KeyGen}(1^\lambda))_{k=i,\ldots,i+l}$
$(rk_{i+k \to i+k+1} \leftarrow \mathsf{ReKeyGen}(sk_{i+k}, sk_{i+k+1}))_{k=0,\ldots,l-1}$
$(w, w') \leftarrow \mathcal{A}(pk)$
$C^{(0)}_{w,i} \leftarrow \mathsf{Enc}(pk_i, w)$
$(C^{(k+1)}_{w,i} \leftarrow \mathsf{ReEnc}(rk_{i+k \to i+k+1}, C^{(k)}_{w,i}))_{k=0,\ldots,l-1}$
$T_{w',i+l} \leftarrow \mathsf{Trapdoor}(sk_{i+l}, w', state_{i+l})$
If $w \neq w'$ and $\mathsf{Test}(C^{(l)}_{w,i}, T_{w',i+l}) = 1$
then return 1 else return 0

定義 2：Computational Consistency(暗号レベル $l \geq 1$)
実験 $\mathbf{Expr}^{\text{consistency\_}l}_{\text{KU-PEKS},\mathcal{A}}(\lambda)$ におけるアルゴリズム $\mathcal{A}$ の KU-PEKS に対する優位度（advantage）を次のように定義する．

$$\mathbf{Adv}^{\text{consistency\_}l}_{\text{KU-PEKS},\mathcal{A}}(\lambda) \stackrel{\text{def}}{=} \Pr[1 \leftarrow \mathbf{Expr}^{\text{consistency\_}l}_{\text{KU-PEKS},\mathcal{A}}(\lambda)]. \tag{2}$$

実験 $\mathbf{Expr}^{\text{consistency\_}l}_{\text{KU-PEKS},\mathcal{A}}(\lambda)$ において，確率的多項式時間の任意のアルゴリズム $\mathcal{A}$ に対し優位度 $\mathbf{Adv}^{\text{consistency\_}l}_{\text{KU-PEKS},\mathcal{A}}(\lambda)$ が λ の無視可能な関数であるとき，KU-PEKS は暗号レベル l において Computational consistency を持つものと定義する．

3.2.3 IND-KU-CKA 安全性

ここでは，鍵漏洩を考慮した KU-PEKS の検索キーワードに関する安全性，すなわち，鍵漏洩および選択クエリ攻撃に対する識別不可能性（Indistinguishablitiy against key exposure and chosen keyword attacks for KU-PEKS : IND-KU-CKA）を定義する．

多項式アルゴリズム $\mathcal{A}$ を仮定し，IND-KU-CKA について次の実験を定義する．

$\mathbf{Expr}^{\text{IND-KU-CKA}}_{\text{KU-PEKS},\mathcal{A}}(\lambda)$
$(sk_i, pk_i) \leftarrow \mathsf{KeyGen}(1^\lambda)$
$(w_0, w_1, state) \leftarrow \mathcal{A}^{\mathcal{O}}(find, pk_i)$
$d \leftarrow \{0,1\}$
$C^{(0)}_{w_d,i} \leftarrow \mathsf{Enc}(pk_i, w_d)$
$d' \leftarrow \mathcal{A}^{\mathcal{O}}(guess, C^{(0)}_{w_d,i}, state)$
If $d = d'$then return 1 else return 0

ただし，オラクル $\mathcal{O} = \{\mathcal{O}_{KG}, \mathcal{O}_{KL}, \mathcal{O}_{RK}, \mathcal{O}_{TD}, \mathcal{O}_{RE}, \mathcal{O}_{TE}\}$ であり，$\mathcal{A}^{\mathcal{O}}$ は $\mathcal{A}$ が各オラクルにアクセスできることを表す．なお，各オラクルは次のように定義される．

- $\mathcal{O}_{KG}$：任意のバージョン番号 i を入力として，KeyGen を実行して鍵ペアを生成し，公開鍵を出力する．
- $\mathcal{O}_{KL}$：バージョン番号 i を入力として，対応する秘密鍵を出力する．ただし，i が，最新のバージョン番号である場合は何も返さない．
- $\mathcal{O}_{RK}$：バージョン番号 i を入力とし，ReKeyGen を実行して，バージョン $i+1$ に更新するための，更新鍵 $rk_{i \to i+1}$ を出力する．
- $\mathcal{O}_{TD}$：バージョン番号 j と w を入力として，Trapdoor を実行してその出力の第 i 項であるトラップドアを出力する．なお，入力の w が，$\mathcal{A}$ からのクエリ w_0 あるいは w_1 である場合は，何も返さない．
- $\mathcal{O}_{RE}$：暗号文と更新鍵を入力として，ReEnc を実行して，更新された暗号文を出力する．
- $\mathcal{O}_{TE}$：暗号文とトラップドアを入力として，Test を実行して，その結果を出力する．

定義 3：IND-KU-CKA
実験 $\mathbf{Expr}^{\text{IND-KU-CKA}}_{\text{KU-PEKS},\mathcal{A}}(\lambda)$ における多項式アルゴリズム $\mathcal{A}$ の優位度について，次のように定義する．

$$\mathbf{Adv}^{\text{IND-KU-CKA}}_{\text{KU-PEKS},\mathcal{A}}(\lambda) \stackrel{\text{def}}{=} \Pr[1 \leftarrow \mathbf{Expr}^{\text{IND-KU-CKA}}_{\text{KU-PEKS},\mathcal{A}}(\lambda)] - 1/2. \tag{3}$$

$\mathbf{Adv}^{\text{IND-KU-CKA}}_{\text{KU-PEKS},\mathcal{A}}(\lambda)$ が λ に関して無視可能な関数であるとき，KU-PEKS は IND-KU-CKA の安全性を持つと定義する．

4. 構成

基本的なアイディアは，Boneh ら [2] の検索可能暗号に，暗号文の更新を容易とする Shao ら [7] の検索可能代理人再暗号化方式のアイディア（暗号文における公開鍵と生成元の位置を入れ替える）を組み合わせ，さらに KU-PEKS の要件 4 を実現するために，更新鍵に任意に生成した乱数 R_i を導入する．そして，乱数 R_i は鍵更新のたびに蓄積するため，打ち消すための要素を，受信クライアントの内部状態値として保持し，Test のもう一方入力である，トラップ

ドアに含める．なお，乱数要素を更新鍵に含めるアプローチは，竹谷ら [11] の検索可能代理人再暗号化方式においても，一方向の再暗号化を実現するために使われている．

KU-PEKS= (Setup, KeyGen, ReKeyGen, Enc, ReEnc, Trapdoor, Test) は以下のように構成される．

- $\mathsf{Setup}(1^\lambda) \to PP$

  $\mathbb{G}$ と $\mathbb{G}_T$ を，位数を素数 q とする（乗法）巡回群とし，g を $\mathbb{G}$ の生成元とする．双線型写像 $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ を選ぶ．また，2 つのハッシュ関数 $H_1 : \{0,1\}^* \to \mathbb{G}$，$H_2 : \mathbb{G}_T \to \{0,1\}^\lambda$ を選ぶ．公開パラメータ $PP = (q, g, \mathbb{G}, \mathbb{G}_T, e, H_1, H_2)$ を出力する．

- $\mathsf{KeyGen}(1^\lambda) \to (sk_i, pk_i)$

  $\mathbb{Z}_q$ からランダムに x を選び，バージョン i におけるクライアント鍵ペア（$sk_i = x_i, pk_i = g^{x_i}$）を出力する．

- $\mathsf{ReKeyGen}(sk_i, sk_{i+1}, state_i) \to rk_{i\to i+1}, state_{i+1}$

  $\mathbb{Z}_q$ からランダムに R_i を選び，隣り合ったバージョンの秘密鍵 sk_i と sk_{i+1} を用いて，更新鍵 $rk_{i\to i+1} = (sk_{i+1}/sk_i) \times R_i$ を求め，出力する．また，$state_1 = \emptyset$ として，$state_i = \{st_{i,1}, \ldots, st_{i,i-1}\}$ とおいたとき，$state_{i+1}$ を，$state_{i+1} = \{st_{i,1} \times R_i, \ldots, st_{i,i-1} \times R_i, R_i\} = \{st_{i+1,1}, \ldots, st_{i+1,i-1}, st_{i+1,i}\}$ と計算して求め，クライアント内に保存する．なお，$st_{i,j} = \prod_{k=j}^{i-1} R_k$ が成り立つ．

- $\mathsf{Enc}(pk_i, w) \to C_{w,i}^{(0)}$

  $\mathbb{Z}_q$ からランダムに r を選び，$A_i^{(0)} = pk_i^r, B_w = H_2(e(H_1(w), g)^r)$ を計算する．暗号文 $C_{w,i}^{(0)} = (A_i^{(0)}, B_w)$ を出力する．

- $\mathsf{ReEnc}(rk_{i+k\to i+k+1}, C_{w,i}^{(k)}) \to C_{w,i}^{(k+1)}$

  $C_{w,i}^{(k)} = (A_i^{(k)}, B_w)$ とおく．$A_i^{(k+1)} = (A_i^{(k)})^{rk_{i+k\to i+k+1}}$ を計算し，$C_{w,i}^{(k+1)} = (A_i^{(k+1)}, B_w)$ を出力する．

- $\mathsf{Trapdoor}(sk_j, w', state_j) \to T_{w',j}$

  $state_j = \{st_{j,1}, \ldots, st_{j,j-1}\}$ とおく．$t_{w',j,k} = H_1(w')^{1/(sk_j \times st_{j,k})}$ を $k = 1, \ldots, j-1$ で求め，$T_{w',j} = \{t_{w',j,1}, t_{w',j,2}, \ldots t_{w',j,j}\}$ を出力する．なお，$t_{w',j,j} = H_1(w')^{1/sk_j}$ とする．

- $\mathsf{Test}(C_{w,i}^{(j-i)}, T_{w',j})$

  $C_{w,i}^{(j-i)} = (A_i^{(j-i)}, B_w)$，および $T_{w',j} = \{t_{w',j,1}, t_{w',j,2}, \ldots t_{w',j,j}\}$ とするとき，$T_{w',j}$ から，暗号文生成に使用した公開鍵のバージョン i に対応した $t_{w',j,i}$ を選び，$B_w = H_2(e(t_{w',j,i}, A_i^{(j-i)})$ が成り立つなら，1 を出力する．成り立たなければ，0 を出力する．1 は検索が成功したことを示す．

5. 安全性

5.1 Correctness

4 章で構成した方式は，Correctness を満たす．以下，その理由を示す．

暗号レベル 0 の場合の Correctness は，
$H_2(e(t_{w,i,i}, A_i^{(0)}) = H_2(e(H_1(w)^{1/sk_i}, pk_i^r) = B_w$ より明らかである．

また，暗号レベル $l(\geq 1)$ の場合は，
$A_i^{(j-i)} = (pk_j^r)\prod_{k=i}^{j-1} R_k$
$t_{w,j,i} = H_1(w)^{1/(sk_j \times st_{j,i})} = H_1(w)^{1/(sk_j \times \prod_{k=i}^{j-1} R_k)}$
の 2 式を用いると，
$H_2(e(t_{w,j,i}, A_i^{(j-i)}))$
$= H_2(e(H_1(w)^{1/(sk_j \times \prod_{k=i}^{j-1} R_k)}, (pk_j^r)^{\prod_{k=i}^{j-1} R_k}))$
$= H_2(e(H_1(w), g)^r) = B_w$
が成り立つ．これにより，提案方式の Correctness が示される．

5.2 Computational Consistency

4 章で構成した方式は，Computational Consistency を満たす．以下，暗号レベルが 0 の場合と，1 以上の場合に分けて説明する．

暗号レベル 0 の場合

Consistency を破る多項式アルゴリズム $\mathcal{A}$ を仮定する．$\mathcal{A}$ が生成する相異なる検索キーワードを、(w, w') とする（$w \neq w'$）．$r \in \mathbb{Z}_p^*$ をランダムにとり，$X = e(H_1(w), g)^r$，$X' = e(H_1(w'), g)^r$ とおく．このとき，$H_2(X) = H_2(X')$ となれば，多項式アルゴリズム $\mathcal{A}$ の勝ちである．$w_1, w_2, ..., w_{q_1(\lambda)}$ を，H_1 に対する $\mathcal{A}$ のクエリとし，$WSET = \{w_1, w_2, ..., w_{q_1(\lambda)}\} \cup \{w, w'\}$ とおく．また，$x_1, x_2, ..., x_{q_2(\lambda)}$ を，H_2 に対する $\mathcal{A}$ のクエリとし，$XSET = \{x_1, x_2, ..., x_{q_2(\lambda)}\} \cup \{X, X'\}$ とおく．イベント $E1$ と $E2$ を次のように定義する．

- $E1$ ：$H_1(v) = H_1(v')$ を満たす，相異なる $v, v' \in WSET$ が存在する．
- $E2$ ：$H_2(x) = H_2(x')$ を満たす，相異なる $x, x' \in XSET$ が存在する．

このとき，Consistency の実験におけるアルゴリズム $\mathcal{A}$ の KU-PEKS に対する優位度は次の通りとなる．

$$\mathbf{Adv}^{\text{consistency\_0}}_{\text{KU-PEKS},\mathcal{A}}(\lambda) \leq Pr[E1] + Pr[E2] \tag{4}$$

まず，上記式 (4) の証明の準備として，$H_1(w) \neq H_1(w')$ のとき，$X \neq X'$ が成り立つことを示す．$H_1(w)$ および，$H_1(w')$ は，素数位数 q の巡回群 $\mathbb{G}$ の元であるため，相異なる α，$\alpha' \in \mathbb{Z}_p^*$ を用いて，それぞれ $H_1(w) = g^\alpha$，$H_1(w') = g^{\alpha'}$ と表すことができる．$G = e(g, g)^r$ とおくと，$X = e(H_1(w), g)^r = e(g, g)^{\alpha \times r} = G^\alpha$，$X' =$

$e(H_1(w'),g)^r = e(g,g)^{\alpha' \times r} = G^{\alpha'}$ の 2 式が成り立つ．$\mathbb{G}_T$ が素数位数の巡回群であるため，G は $\mathbb{G}_T$ の生成元となり，従って，$X \neq X'$ が成り立つ．次に，この結果を用いて，上記式 (4) の右辺以外（つまり，$\overline{E1} \cap \overline{E2}$）の場合には，$\mathbf{Expr}^{\text{consistency\_0}}_{\text{KU-PEKS},\mathcal{A}}(\lambda)$ が 1 にならないことを示す．$\overline{E1}$ の場合，相異なる w, w' に対応して，$H_1(w) \neq H_1(w')$ が成り立つ．上記の準備より，この場合，$X \neq X'$ が成り立つ．さらに，$\overline{E2}$ の場合，$H_2(X) \neq H_2(X')$ となる．そのため，$\mathbf{Expr}^{\text{consistency\_0}}_{\text{KU-PEKS},\mathcal{A}}(\lambda)$ は 0 となる．$Pr[E1]$，および $Pr[E2]$ は，$\mathbb{G}$ を全空間とした，バースディパラドックスにより，$Pr[E1] = (q_{1(\lambda)}+2)\times(q_{1(\lambda)}+1)/\{2\times|\mathbb{G}|\} < (q_{1(\lambda)}+2)^2/2^\lambda$ $Pr[E2] = (q_{2(\lambda)}+2)\times(q_{2(\lambda)}+1)/\{2\times|\mathbb{G}|\} < (q_{2(\lambda)}+2)^2/2^\lambda$ が成り立つ．従って，$\mathbf{Adv}^{\text{consistency\_0}}_{\text{KU-PEKS},\mathcal{A}}(\lambda)$ が λ の多項式で bound されるため，Computational consistency の定義を満たす．

暗号レベル l の場合　$(l \geq 1)$

暗号レベル 0 の場合と同様に，Consistency を破る多項式アルゴリズム $\mathcal{A}$ を仮定し，相異なる検索キーワードを、(w, w') とする（$w \neq w'$）．$r \in \mathbb{Z}_p^*$ をランダムにとり，$X = e(H_1(w),g)^r$，$X' = e(H_1(w'),g)^r$ とおく．$\mathsf{Test}(C^{(j-i)}_{w,i}, T_{w',i})$ における確認式 $B_w = H_2(e(t_{w',j,i}, A_i^{(j-i)}))$ の左辺は $H_2(X)$ であり，一方，右辺は $H_2(X')$ となる．従って，暗号レベル 0 の場合と同様に，$H_2(X) = H_2(X')$ となれば，多項式アルゴリズム $\mathcal{A}$ の勝ちである．暗号レベル 0 の場合と同様に $\mathbf{Adv}^{\text{consistency\_}l}_{\text{KU-PEKS},\mathcal{A}}(\lambda)$ が λ の多項式で bound され，Computational consistency の定義を満たすことが証明される．

5.3 IND-KU-CKA

4 章で構成した方式は，mDBDH 仮定が成り立つ場合，ランダムオラクルモデルにおいて IND-KU-CKA 安全性を達成する．以下，IND-KU-CKA 安全性においてアドバンテージが ϵ である多項式 $\mathcal{A}$ が存在するとき，$\mathcal{A}$ を用いて mDBDH 問題を有意に解く多項式アルゴリズム $\mathcal{B}$ を構築できることを示す．

$\mathcal{B}$ は，(g, g^a, g^b, g^c) と Q を入力とする．Q が $Q = e(g,g)^{a\times b/c}$ を満たす確率は 1/2 であり，さもなくばランダムな元である．

- ハッシュオラクル：多項式アルゴリズム $\mathcal{B}$ は，以下のようにオラクルをシミュレートする．
  - $\mathcal{O}_{H_1}$：入力 w に対して，保持する表 $\mathcal{T}_{H_1} = (w, R^{(1)}, r^{(1)}, coin)$ にあれば，$R^{(1)}$ を出力する．表になければ，確率 $1/q_{td}$ で $coin = 0$ となる，$coin \in \{0,1\}$ と乱数 $r^{(1)}$ を発生し，$coin = 0$ の場合は $R^{(1)} = (g^a)^{r^{(1)}}$ を出力する．$coin = 1$ の場合は $R^{(1)} = (g^c)^{r^{(1)}}$ を出力する．さらに，表 $\mathcal{T}_{H_1}$ に $(w, R^{(1)}, r^{(1)}, coin)$ を追加する．なお，ここで，q_{td} は，オラクル $\mathcal{O}_{TD}$ にアクセス可能な回数とする．
  - $\mathcal{O}_{H_2}$：入力 X に対して，保持する表 $\mathcal{T}_{H_2} = (X, R^{(2)})$ にあれば，$R^{(2)}$ を出力．表になければ，乱数 $R^{(2)} \in \mathbb{Z}_q$ を生成して出力し，保持する表 $\mathcal{T}_{H_3}$ に $(X, R^{(2)})$ を追加する．
- Phase 1：多項式アルゴリズム $\mathcal{B}$ は，以下のようにオラクルをシミュレートする．
  - $\mathcal{O}_{KG}$：乱数 $x_i \in \mathbb{Z}_q$ を生成し，$pk_i = (g^c)^{x_i}$ を出力する．さらに，表 $\mathcal{T}_K = (x_i, pk_i)$ に追加する．
  - $\mathcal{O}_{KL}$：公開鍵 pk_i を入力とし，それが最新バージョンのときは $\perp$ を返す．それ以外の場合は，表 $\mathcal{T}_K = (x_i, pk_i)$ から対応する x_i を出力する．もし表になければ，$\perp$ を返す．
  - $\mathcal{O}_{RK}$：2 つの公開鍵 pk_i, pk_j を入力とし，それらのうち少なくとも 1 つが最新バージョンのときは $\perp$ を返す．また，表 $\mathcal{T}_K$ に入力された公開鍵がなければ，$\perp$ を返す．それ以外の場合は，表 $\mathcal{T}_K = (x_i, pk_i)$ から対応する x_i, x_j を出力して x_j/x_i を出力する．
  - $\mathcal{O}_{TD}$：pk_i と w を入力とし，入力の公開鍵が最新バージョンのときは $\perp$ を返す．また，表 $\mathcal{T}_K$ に入力された公開鍵がなければ，$\perp$ を返す．それ以外の場合は，表 $\mathcal{T}_K$ から対応する秘密鍵 x_i を求める．そして，ハッシュオラクル $\mathcal{O}_{H_1}$ から $(w, R^{(1)}, r^{(1)}, coin)$ を獲得し $coin = 0$ の場合は，failure を出力して失敗とし，$coin = 1$ の場合は，$g^{r^{(1)}/x_i}$ を出力する．
- Challenge ：多項式アルゴリズム $\mathcal{B}$ は，$\mathcal{A}$ からのチャレンジ入力 (pk^*, w_0, w_1) に対して，もし，$pk^* \notin \mathcal{T}_K$ であれば，$\perp$ を返す．表にあれば，表から pk^* に対応した秘密鍵を得て x^* とする．また，w_0 と，w_1 をそれぞれ $\mathcal{O}_{H_1}$ に入力する．それぞれを入力した場合の，$coin$ を，$coin_0, coin_1$ とすると，$coin_0, coin_1$ の組み合わせは，$(0,0),(0,1),(1,0),(1,1)$ のいずれかとなる．このうち $(1,1)$ の場合は，$\mathcal{B}$ は，ゲームに失敗する．それ以外の場合は，$coin_d = 0$ となる，$d \in \{0,1\}$ と，対応する $r^{(1)}$ を選んで，次の暗号文を $\mathcal{A}$ に返す．
  - $A^* = (g^b)^{x^*}$
  - $B^* = H_2(Q^{r^{(1)}})$
- Phase2 ：$\mathcal{B}$ は Phase1 のときと同様に，オラクルをシミュレートする．ただし，　$\mathcal{O}_{TD}$ は，入力が w_0，あるいは w_1 の場合は　$\perp$ を返す．
- Guess ：多項式アルゴリズム $\mathcal{A}$ は，Challenge における暗号文，Phase1,2 におけるオラクルからの出力を用いて，ReEnc, Test を動作させて $d' \in \{0,1\}$ を求め，出力する．$\mathcal{B}$ は，d' が Challenge フェーズで定めた d と同じ場合，1 を出力（つまり，$Q = e(g,g)^{a\times b/c}$ が成り立つ）それ以外の場合は，0 を出力（つまり，Q は乱

数）する．

このシミュレーションにおいて，$\mathcal{B}$ は，$coin = 1$ の場合，$g^{r^{(1)}/x_i} = ((g^c)^{r^{(1)}})^{1/(cx_i)}$ が成り立つため，$\mathcal{O}_{TD}$ をシミュレートしている．一方，Challenge において，$coin_d = 0$ が成り立ち，$\mathcal{B}$ が生成した暗号文は，もし，$Q = e(g,g)^{a\times b/c}$ であれば，正当な暗号文となる．そのため，ReEnc，Test を用いて $\mathcal{A}$ は ε の確率で，正しい d' を求めることができる．従って，$d = d'$ が成り立てば，$\mathcal{B}$ は入力された Q が $Q = e(g,g)^{a\times b/c}$ であることを判別できる．一方，Q が乱数である場合は，$\mathcal{A}$ は確率 1/2 でしか，d と一致する正しい d' を判別することはできない．

次に，上記判別のアドバンテージについて求める．正しく判別できる確率は，

a) $\mathcal{B}$ が受け取る Q が，$Q = e(g,g)^{a\times b/c}$ である確率

b) 多項式アルゴリズム $\mathcal{A}$ が IND-KU-CKA を解法するアドバンテージ

c) 多項式アルゴリズム $\mathcal{B}$ が $\mathcal{A}$ を用いるに際し，失敗しない確率

の積になる．上記のうち，a) は mDBDH の仮定より 1/2，b) は ϵ となる．c) は，以下の ϵ_1 と，ϵ_2 を除く事象の確率の積より求める．

ϵ_1：オラクル $\mathcal{O}_{TD}$ において，$\mathcal{A}$ からのクエリが $coin = 0$ に対応し，失敗する場合

ϵ_2：Challenge において，$\mathcal{A}$ からのチャレンジ入力 w_0, w_1 に対応した $coin_0$, $coin_1$ が $(1,1)$ となる場合

事象 ϵ_1 が生じる確率は，$1/(q_{td}+1)$ であり，オラクル $\mathcal{O}_{TD}$ には，クエリが最大 q_{td} 回送られるため，$Pr[\bar{\epsilon_1}] \geq (1-1/(q_{td}+1))^{q_{td}}$ が成り立つ．一方，事象 ϵ_2 において，$coin = 1$ となる確率は，$1/(q_{td}+1)$ であるため，$Pr[\bar{\epsilon_2}] \geq (1-1/(q_{td}+1))^2$ が成り立つ．以上のことより，$\mathcal{B}$ は mDBDH 問題に対して持つアドバンテージは，$1/2 \times \epsilon \times Pr[\bar{\epsilon_1}] \times Pr[\bar{\epsilon_2}]$ 以上となる．

6. まとめ

本稿では，鍵更新機能付きの検索可能暗号について，具体方式を示し，その安全性をランダムオラクルモデルにおいて，Correctness，Computational Consistency, IND-KU-CKA の 3 点で確認した．今後は，非対称ペアリングを利用する方法に拡張を検討し，実装性を検討する予定である．なお，今回提案した方法は，クライアントの秘密鍵の更新に対応して，公開鍵を更新する公開鍵更新モデルである．公開鍵を一定とすることで鍵の運用を簡易にする，鍵隔離暗号モデルについては別報告を参考にされたい．

謝辞 本研究の一部は，JSPS 科研費（第一，第二著者においては JP17K00189，第三著者においては JP16J10532 と JP17K12697）の助成を受けています．

参考文献

[1] Abdalla, M., Bellare, M., Catalano, D., Kiltz, E., Kohno, T., Lange, T., Malone-Lee, J., Neven, G., Paillier, P. and Shi, H.: Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions, *J. Cryptology*, Vol. 21, No. 3, pp. 350–391 (2008).

[2] Boneh, D., Crescenzo, G. D., Ostrovsky, R. and Persiano, G.: Public Key Encryption with Keyword Search, *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, pp. 506–522 (2004).

[3] Boneh, D. and Franklin, M. K.: Identity-Based Encryption from the Weil Pairing, *Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '01, London, UK, UK, Springer-Verlag, pp. 213–229 (online), available from ⟨http://dl.acm.org/citation.cfm?id=646766.704155⟩ (2001).

[4] Canetti, R. and Hohenberger, S.: Chosen-ciphertext secure proxy re-encryption, *Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, Alexandria, Virginia, USA, October 28-31, 2007*, pp. 185–194 (2007).

[5] Moody, D., Peralta, R., Perlner, R., Regenscheid, A., Roginsky, A. and Chen, L.: Report on pairing-based cryptography, *Journal of research of the National Institute of Standards and Technology*, Vol. 120, p. 11 (2015).

[6] National Institute of Standards and Technology: Pairing-Based Cryptography (Created January 17, 2017, Updated March 01, 2017). https://beta.csrc.nist.gov/Projects/Pairing-Based-Cryptography.

[7] Shao, J., Cao, Z., Liang, X. and Lin, H.: Proxy re-encryption with keyword search, *Inf. Sci.*, Vol. 180, No. 13, pp. 2576–2587 (2010).

[8] 境隆一，大岸聖史，笠原正雄：Cryptosystems based on pairing, *Symposium on Cryptography and Information Security (SCIS2000)* (2000).

[9] 松崎なつめ，穴田啓晃：検索可能暗号の鍵更新についての調査，*Symposium on Cryptography and Information Security (SCIS2017)* (2017).

[10] 松崎なつめ，穴田啓晃，渡邉洋平：鍵更新可能な検索可能暗号の一提案：検索可能代理人再暗号化の適用について，*IEICE Technical Report ISEC2017-1* (2017).

[11] 竹谷駿佑，尾形わかは：Unidirectional な検索可能代理人再暗号化方式の構築，*Symposium on Cryptography and Information Security (SCIS2017)* (2017).

[12] 渡邉洋平，穴田啓晃，松崎なつめ：鍵更新機能付き検索可能暗号：鍵隔離暗号モデルによる実現，*Computer Security Symposium 2017* (2017).