鍵更新効能付き検索可能暗号の概念と一般的構成
Key-Updatable Public-Key Encryption with Keyword Search : Concept and Generic Construction

代表研究者　松崎　なつめ　長崎県立大学情報システム学部情報セキュリティ学科　教授
共同研究者　穴田　啓晃　長崎県立大学情報システム学部情報セキュリティ学科　教授

1　はじめに

検索可能暗号は，クラウドの安全な利用を目的とした高機能暗号の一種であり，クラウドに預託した暗号文から，暗号化したクエリを用いて所望の情報を検索可能とする．検索可能暗号は，預託時と検索時に同じ鍵を用いる共通鍵型（Symmetric Searchable Encryption: SSE）と，異なる鍵を用いる公開鍵型（Public Key Encryption with Keyword Search : PEKS）に分類される．このうち公開鍵型は，境，大岸，笠原 [8] が 2000 年に，また，Boneh と Franklin[2] が 2001 年に提案した ID ベース暗号（Identity-Based Encryption : IBE）を応用し，匿名 IBE 方式から一般的に構成されることが知られている [1].

本研究では，公開鍵型の検索可能暗号における鍵の更新に着目する．検索可能暗号は，従来より，検索を行うクライアント側の安全な鍵管理を前提としたうえで，様々な方式が検討されてきた．しかし，IoT 機器のように，安全性の確保のためのコストがかけにくいクライアントでの適用を考えると，今までの研究に加え，クライアントデバイスからの鍵漏洩への対策を考慮する必要がある．本研究で着目するクライアント鍵の更新は，鍵漏洩対策の一アプローチであり，万が一，鍵が漏洩しても，鍵更新により，古い鍵が無効化されて，新しい鍵が利用される．

本稿では，鍵更新機能付き検索可能暗号の概念と要件を示し，その一般的構成を提案する．提案する一般的構成は，PEKS と公開鍵暗号を組み合わせたものである．さらに，鍵更新機能付きの検索可能暗号の安全性を，PEKS の安全性定義を拡張することにより定義し，上記一般的構成の安全性を示した．また，ベースとする PEKS の特性により，それぞれ，ランダムオラクルモデルと，スタンダードモデルの鍵更新機能付き検索可能暗号が構成される．なお，鍵更新機能付き検索可能暗号には，以下の 2 つのモデルを考えることができる．第 1 のモデルは，クライアントデバイスの（秘密）鍵更新に伴い，対応する公開鍵も更新されるモデルである．このモデルでは，構成がシンプルで実装容易である一方，更新した公開鍵を，広く公開する方法が別途必要となるため，運用において課題がある．第 2 のモデルは，公開鍵を一定としたまま，鍵更新するモデルである．公開鍵を一定とするため，運用において利点がある一方，第 1 のモデルに比べると計算が複雑になる．この構成は，秘密鍵を復号用と復号用鍵を更新する更新用の 2 種類に分けて，更新用の鍵を隔離することにより，一定の安全性を保障する鍵隔離型の暗号研究（例えば，[3]）を拡張したものとなっている．本稿では，第

1 のモデルについて説明する．なお，第 2 のモデルについては，[10] を参考にするとよい．

本稿の構成は次のとおりである．まず，第 2 章で方式や安全性定義を整理した上で，第 3 章で鍵更新機能付き検索可能暗号のモデルと要件を定義する．そして第 4 章で，第 3 章で定義した，鍵更新機能付き検索可能暗号を，既存の公開鍵型検索可能暗号をベースとし，公開鍵暗号を組み合わせることにより，一般的に構成する．第 5 章では，IoT 機器での実装を見据えた評価結果を示す．

2 準備

2.1 公開鍵暗号

公開鍵暗号 (Public Key Encryption: PKE) $\mathcal{PKE} = (\mathsf{PG}, \mathsf{G}, \mathsf{E}, \mathsf{D})$ は以下のように定義される．

- $\mathsf{PG}(1^\lambda) \to \mathsf{par}_{\mathrm{PKE}}$: 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 $\mathsf{par}_{\mathrm{PKE}}$ を出力する．
- $\mathsf{G}(\mathsf{par}_{\mathrm{PKE}}) \to (\mathsf{dk}, \mathsf{ek})$: 公開パラメータ $\mathsf{par}_{\mathrm{PKE}}$ を入力として，鍵ペア（復号鍵 dk と暗号化鍵 ek）を出力する．
- $\mathsf{E}(\mathsf{ek}, m) \to \mathsf{ct}$: 暗号化鍵 ek と平文 $m \in \mathcal{M}_{\mathrm{PKE}}$ を入力して，暗号文 ct を出力する．$\mathcal{M}$ はセキュリティパラメータによって決まる平文集合である．
- $\mathsf{D}(\mathsf{dk}, \mathsf{ct}) \to m$ or $\perp$: 復号鍵 dk と暗号文 ct を入力として，m または復号失敗を表す $\perp$ を出力する．

上記モデルは以下の正当性を要求する．全ての $\lambda \in \mathbb{N}$, $\mathsf{par}_{\mathrm{PKE}} \leftarrow \mathsf{PG}(1^\lambda)$, $(\mathsf{dk}, \mathsf{ek}) \leftarrow \mathsf{G}(\mathsf{par}_{\mathrm{PKE}})$, 任意の $m \in \mathcal{M}$ に対して，$\mathsf{D}(\mathsf{dk}, \mathsf{E}(\mathsf{ek}, m)) = m$.

本稿で扱う安全性である選択平文攻撃に対する識別不可能性 (IND-CPA) について，以下の試行 $\mathsf{Exp}^{\mathsf{CPA}}_{\mathcal{PKE},\mathcal{A}}(1^\lambda)$ を考える．

$\mathsf{Exp}^{\mathsf{CPA}}_{\mathcal{PKE},\mathcal{A}}(1^\lambda)$

$\mathsf{par}_{\mathrm{PKE}} \leftarrow \mathsf{PG}(1^\lambda), \quad (\mathsf{dk}, \mathsf{ek}) \leftarrow \mathsf{G}(\mathsf{par}_{\mathrm{PKE}})$
$(m_0^*, m_1^*, \mathit{state}) \leftarrow \mathcal{A}(\mathsf{par}_{\mathrm{PKE}}, \mathsf{ek})$ s.t. $|m_0^*| = |m_1^*|$
$b \xleftarrow{\$} \{0,1\}, \quad \mathsf{ct}_b^* \leftarrow \mathsf{E}(\mathsf{ek}, m_b^*)$
$b' \leftarrow \mathcal{A}(\mathit{state}, \mathsf{ct}_b^*)$
If $b' = b$ return 1 else return 0

定義 2.1 (IND-CPA) 任意の確率的多項式時間攻撃者 $\mathcal{A}$ に対して，$\mathcal{PKE}$ が $\mathsf{Adv}^{\mathsf{CPA}}_{\mathcal{PKE},\mathcal{A}}(1^\lambda) := |\Pr[\mathsf{Exp}^{\mathsf{CPA}}_{\mathcal{PKE},\mathcal{A}}(1^\lambda) = 1] - 1/2| < \epsilon(\lambda)$ を満たすとき，$\mathcal{PKE}$ は IND-CPA を満たすという．

2.2 検索可能暗号

検索可能暗号 (Public Key Encryption with Keyword Search: PEKS) $\mathcal{PEKS} = (\mathsf{Setup}_{\mathrm{PEKS}}, \mathsf{KeyGen}_{\mathrm{PEKS}}, \mathsf{Enc}_{\mathrm{PEKS}}, \mathsf{Trapdoor}_{\mathrm{PEKS}}, \mathsf{Test}_{\mathrm{PEKS}})$ は以下のように定義される．

- $\mathsf{Setup}_{\mathrm{PEKS}}(1^\lambda) \to \mathsf{par}_{\mathrm{PEKS}}$: 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 $\mathsf{par}_{\mathrm{PEKS}}$ を出力する．

- $\mathsf{KeyGen}_{\mathrm{PEKS}}(\mathsf{par}_{\mathrm{PEKS}}) \to (\mathsf{msk}, \mathsf{mpk})$: 公開パラメータを $\mathsf{par}_{\mathrm{PEKS}}$ を入力として，鍵ペア（秘密鍵 msk と公開鍵 mpk）を出力する．
- $\mathsf{Enc}_{\mathrm{PEKS}}(\mathsf{mpk}, w) \to \mathsf{ct}_w$: 公開鍵 mpk と検索キーワード $w \in \mathcal{W}$ を入力として，検索用の暗号文（暗号 index）ct_w を出力する．$\mathcal{W}$ はセキュリティパラメータによって決まるキーワード集合である．
- $\mathsf{Trapdoor}_{\mathrm{PEKS}}(\mathsf{msk}, w') \to \mathsf{td}_{w'}$: 秘密鍵 msk と検索キーワード $w' \in \mathcal{W}$ を入力として，トラップドア $\mathsf{td}_{w'}$ を出力する．
- $\mathsf{Test}_{\mathrm{PEKS}}(\mathsf{td}_{w'}, \mathsf{ct}_w) \to 1$ or 0: 暗号文 ct_w とトラップドア $\mathsf{t}_{w'}$ を入力として，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．

上記モデルは以下の正当性を要求する．全ての $\lambda \in \mathbb{N}$, 全ての $\mathsf{par}_{\mathrm{PEKS}} \leftarrow \mathsf{Setup}_{\mathrm{PEKS}}(1^\lambda)$, 全ての $(\mathsf{msk}, \mathsf{mpk}) \leftarrow \mathsf{KeyGen}_{\mathrm{PEKS}}(\mathsf{par}_{\mathrm{PEKS}})$, 全ての $w \in \mathcal{W}$ に対して，

$$\mathsf{Test}_{\mathrm{PEKS}}(\mathsf{Trapdoor}_{\mathrm{PEKS}}(\mathsf{msk}, w), \mathsf{Enc}_{\mathrm{PEKS}}(\mathsf{mpk}, w)) = 1.$$

本稿で扱う安全性である選択キーワード攻撃に対する識別不可能性 (IND-CKA) について，次の試行 $\mathsf{Exp}^{\mathsf{CKA}}_{\mathcal{PEKS},\mathcal{A}}(1^\lambda)$ を考える．

$\mathsf{Exp}^{\mathsf{CKA}}_{\mathcal{PEKS},\mathcal{A}}(1^\lambda)$

$\mathsf{par}_{\mathrm{PEKS}} \leftarrow \mathsf{Setup}_{\mathrm{PEKS}}(1^\lambda)$
$(\mathsf{msk}, \mathsf{mpk}) \leftarrow \mathsf{KeyGen}_{\mathrm{PEKS}}(\mathsf{par}_{\mathrm{PEKS}})$
$(w_0^*, w_1^*, state) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathrm{TD}}}(\mathsf{par}_{\mathrm{PEKS}}, \mathsf{mpk})$
$\qquad\qquad\qquad\qquad\qquad\qquad$ s.t. $|w_0^*| = |w_1^*|$
$b \xleftarrow{\$} \{0,1\}$, $\mathsf{ct}^*_{w_b^*} \leftarrow \mathsf{Enc}_{\mathrm{PEKS}}(\mathsf{mpk}, w_b^*)$
$b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathrm{TD}}}(state, \mathsf{ct}^*_{w_b^*})$
If $b' = b$ return 1 else return 0

$\mathcal{A}$ はオラクル $\mathcal{O}_{\mathrm{TD}}$ にアクセスできる．$\mathcal{O}_{\mathrm{TD}}$ は，$\mathcal{A}$ から w を受け取り，$\mathsf{td}_w \leftarrow \mathsf{Trapdoor}_{\mathrm{PEKS}}(\mathsf{msk}, w)$ を返すオラクルであり，$w \notin \{w_0^*, w_1^*\}$ でなくてはならない．これは，$\mathcal{A}$ が可能な限りトラップドアを手に入れられること，すなわち攻撃者とクラウドの結託を想定していることを意味する．

定義 2.2 (IND-CKA [1]) 任意の確率的多項式時間攻撃者 $\mathcal{A}$ に対して，$\mathcal{PEKS}$ が $\mathsf{Adv}^{\mathsf{CKA}}_{\mathcal{PEKS},\mathcal{A}}(1^\lambda) := |\Pr[\mathsf{Exp}^{\mathsf{CKA}}_{\mathcal{PEKS},\mathcal{A}}(1^\lambda) = 1] - 1/2| < \epsilon(\lambda)$ を満たすとき，$\mathcal{PEKS}$ は IND-CKA を満たすという．

また，以下の計算量的一貫性 (Computational Consistency) を考える．以下の試行 $\mathsf{Exp}^{\mathsf{Cons}}_{\mathcal{PEKS},\mathcal{A}}(1^\lambda)$ を考える．

$\mathsf{Exp}^{\mathsf{Cons}}_{\mathcal{PEKS},\mathcal{A}}(1^\lambda)$

$\mathsf{par}_{\mathrm{PEKS}} \leftarrow \mathsf{Setup}_{\mathrm{PEKS}}(1^\lambda)$
$(\mathsf{msk}, \mathsf{mpk}) \leftarrow \mathsf{KeyGen}_{\mathrm{PEKS}}(\mathsf{par}_{\mathrm{PEKS}})$
$(w_0^*, w_1^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathrm{TD}}}(\mathsf{par}_{\mathrm{PEKS}}, \mathsf{mpk})$ s.t. $|w_0^*| = |w_1^*|$
$\mathsf{ct}^*_{w_0^*} \leftarrow \mathsf{Enc}_{\mathrm{PEKS}}(\mathsf{mpk}, w_0^*)$
$\mathsf{td}_{w_1^*} \leftarrow \mathsf{Trapdoor}_{\mathrm{PEKS}}(\mathsf{msk}, w_1^*)$
If $\mathsf{Test}_{\mathrm{PEKS}}(\mathsf{td}_{w_1^*}, \mathsf{ct}_{w_0^*}) = 1$ and $w_0^* \neq w_1^*$ return 1
else return 0

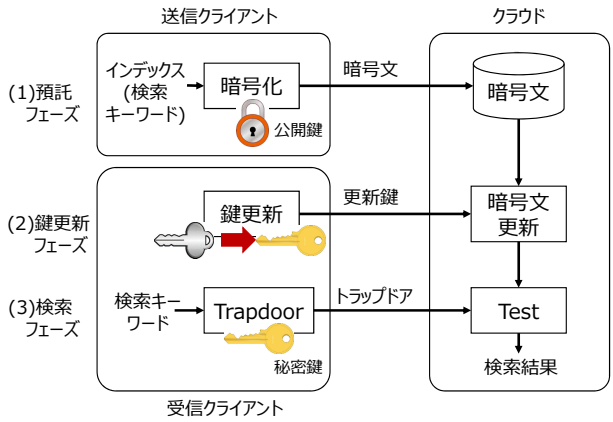

図 1　鍵更新機能付き検索可能暗号（KU-PEKS）の概念モデル

IND-CKA 同様，$\mathcal{A}$ はオラクル $\mathcal{O}_{\mathrm{TD}}$ に同様の制限の下でアクセスできる．

定義 2.3 (Computational Consistency[1]) 任意の確率的多項式時間攻撃者 $\mathcal{A}$ に対して，$\mathcal{PEKS}$ が $\mathsf{Adv}^{\mathsf{Cons}}_{\mathcal{PEKS},\mathcal{A}}(1^\lambda) := \Pr[\mathsf{Exp}^{\mathsf{Cons}}_{\mathcal{PEKS},\mathcal{A}}(1^\lambda) = 1] < \epsilon(\lambda)$ を満たすとき，$\mathcal{PEKS}$ は Computational Consistency を満たすという．

3　鍵更新機能付き検索可能暗号

3.1　概念と要件

本節では，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，(2) 鍵更新フェーズを備える．鍵更新フェーズでは，受信クライアント側で鍵ペアを更新して更新鍵を生成し，クラウド側で更新鍵を用いて，暗号文を更新する．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．

KU-PEKS $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{ReKeyGen}, \mathsf{Enc}, \mathsf{ReEnc}, \mathsf{Trapdoor}, \mathsf{Test})$ は，以下の 7 個のアルゴリズムからなる．更新可能期間を $\mathcal{T}$ とし $|\mathcal{T}| = \mathrm{poly}(\lambda)$ とする．

- $\mathsf{Setup}(1^\lambda) \to \mathsf{pp}$: 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．
- $\mathsf{KeyGen}(\mathsf{sk}_{i-1}, \mathsf{pk}_{i-1}) \to (\mathsf{sk}_i, \mathsf{pk}_i)$: 期間 $i-1 \in \mathcal{T}$ の鍵ペア $(\mathsf{sk}_{i-1}, \mathsf{pk}_{i-1})$ を入力として，期間 $i \in \mathcal{T}$ の鍵ペア（秘密鍵 sk_i と公開鍵 pk_i）を出力する．
- $\mathsf{ReKeyGen}(\mathsf{pk}_i, \mathsf{sk}_i, \mathsf{pk}_{i+1}\mathsf{sk}_{i+1}) \to \mathsf{rk}_{i\to i+1}$: 隣り合ったバージョンの鍵ペア $(\mathsf{pk}_i, \mathsf{sk}_i), (\mathsf{pk}_{i+1}, \mathsf{sk}_{i+1})$ から，更新鍵 $\mathsf{rk}_{i\to i+1}$ を出力する．鍵の添字は，バージョン番号を示す．

- $\mathsf{Enc}(\mathsf{pk}_i, w) \to \mathsf{c}_{w,i}^{(0)}$: バージョン i の公開鍵 pk_i と検索キーワード $w \in \mathcal{W}$ を入力して検索用の暗号文（暗号 index）$\mathsf{c}_{w,i}^{(0)}$ を出力する．暗号文の上の添え字は，更新の回数を示し，この時点では 0 である．
- $\mathsf{ReEnc}(\mathsf{rk}_{i\to i+1}, \mathsf{c}_{w,j}^{(k)}) \to \mathsf{c}_{w,j}^{(k+1)}$: 更新鍵 $\mathsf{rk}_{i\to i+1}$ を用いて，$j+k=i$ であるような暗号文 $\mathsf{c}_{w,j}^{(k)}$ を更新し，$\mathsf{c}_{w,j}^{(k+1)}$ を出力する．
- $\mathsf{Trapdoor}(\mathsf{sk}_i, w') \to \mathsf{t}_{w',i}$: バージョン i の秘密鍵 sk_i と検索キーワード $w' \in \mathcal{W}$ を入力として，トラップドア $\mathsf{t}_{w',i}$ を出力する．
- $\mathsf{Test}(\mathsf{t}_{w',i}, \mathsf{c}_{w,j}^{(k)}) \to 1$ or 0: $j+k=i$ となるようなトラップドア $\mathsf{t}_{w',i}$ と暗号文 $\mathsf{c}_{w,j}^{(k)}$ を入力として，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導出困難であること．
要件 2：古い鍵は受信クライアントデバイスから速やかに削除すること．
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，可用性を目的としたものであり，受信クライアントが，デバイスに保持する新しい鍵だけで，古い暗号文（更新前の古い公開鍵で暗号化された暗号文）を対象として検索可能とする．
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，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，漏洩の可能性のある古い鍵を無効化する意味を持つ．古い鍵で生成したトラップドアを用いて，古い公開鍵で暗号化された暗号文を対象とした検索を抑止することは難しい．しかし，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検索は不可能となる．

3.2 安全性定義

正当性. KU-PEKS Π は以下の復号の正当性を要求する．全ての $\lambda \in \mathbb{N}$, 全ての $i = j+k$ となるような i,j,k, $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$, $(\mathsf{sk}_i, \mathsf{pk}_i) \leftarrow \overbrace{\mathsf{KeyGen}(\mathsf{KeyGen}(\cdots \mathsf{KeyGen}(}^{i\text{ 個}}\mathsf{pp}) \cdots))$ について，$\mathsf{Test}(\mathsf{Trapdoor}(\mathsf{sk}_i, \omega), \mathsf{c}_{w,j}^{(k)}) \to 1$. ただし，$\mathsf{c}_{w,j}^{(\ell)} \leftarrow \mathsf{ReEnc}(\mathsf{ReKeyGen}(\mathsf{sk}_{j+\ell-1}, \mathsf{sk}_{j+\ell}), \mathsf{c}_{w,j}^{(\ell-1)})$ $(1 \le \ell \le k)$ であり，i,j,k は λ の多項式である．

安全性. 以下の試行 $\mathsf{Exp}_{\Pi,\mathcal{A}}^{\mathsf{KU\text{-}CKA}}(1^\lambda)$ を考える．

$\mathsf{Exp}_{\Pi,\mathcal{A}}^{\mathsf{KU\text{-}CKA}}(1^\lambda)$

$\mathsf{ctr} := 0, \quad \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$
$(\mathsf{sk}_1, \mathsf{pk}_1) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$
$(w_0^*, w_1^*, state) \leftarrow \mathcal{A}^{\mathcal{O}_1}(\mathsf{pp}, \mathsf{pk}_1)$ s.t. $|w_0^*| = |w_1^*|$
$b \xleftarrow{\$} \{0,1\}, \quad \mathsf{c}_{w_b^*}^* \leftarrow \mathsf{Enc}(\mathsf{pk}_{\mathsf{ctr}}, w_b^*)$
$b' \leftarrow \mathcal{A}^{\mathcal{O}_2}(state, \mathsf{c}_{w_b^*}^*)$
If $b' = b$ return 1 else return 0

$\mathcal{A}$ は以下のオラクル $\mathcal{O}_1 = \{\mathcal{O}_{\mathrm{KG}}, \mathcal{O}_{\mathrm{KL}}, \mathcal{O}_{\mathrm{RK}}, \mathcal{O}_{\mathrm{TD}}\}$, $\mathcal{O}_2 = \{\mathcal{O}_{\mathrm{KL}}, \mathcal{O}_{\mathrm{RK}}, \mathcal{O}_{\mathrm{TD}}\}$ にアクセスできる．各オラクルの

説明は次の通りである．

$\mathcal{O}_{\mathrm{KG}}$: $\mathcal{A}$ からクエリを受け取り，$\mathsf{ctr} := \mathsf{ctr}+1$ とし，$(\mathsf{sk}_{\mathsf{ctr}}, \mathsf{pk}_{\mathsf{ctr}}) \leftarrow \mathsf{KeyGen}(\mathsf{sk}_{\mathsf{ctr}-1}, \mathsf{pk}_{\mathsf{ctr}-1})$ を実行，$\mathsf{pk}_{\mathsf{ctr}}$ を $\mathcal{A}$ に返すオラクル．$\mathsf{sk}_{\mathsf{ctr}}$ は保持しておく．
$\mathcal{O}_{\mathrm{KL}}$: $\mathcal{A}$ から i を受け取り，sk_i を返すオラクル．
$\mathcal{O}_{\mathrm{RK}}$: $\mathcal{A}$ から i を受け取り，$\mathsf{rk}_{i-1\to i} \leftarrow \mathsf{ReKeyGen}(\mathsf{sk}_{i-1}, \mathsf{sk}_i)$ を返すオラクル．
$\mathcal{O}_{\mathrm{TD}}$: $\mathcal{A}$ から (i, w) を受け取り，$\mathsf{t}_{w,i} \leftarrow \mathsf{Trapdoor}(\mathsf{sk}_i, w)$ を返すオラクル．

$\mathcal{A}$ がチャレンジ後は $\mathcal{O}_{\mathrm{KG}}$ にアクセスできないことに留意する．また，$\mathcal{A}$ は以下の制限を除き，自由にオラクルにアクセスできる．

- $\mathcal{O}_{\mathrm{KL}}$ にクエリする i は $i < \mathsf{ctr}$ でなくてはならない．これは，現在の期間以外であれば，任意の過去の期間の秘密鍵漏洩を許していることを意味する．
- $\mathcal{O}_{\mathrm{RK}}$ にクエリする i は $i \leq \mathsf{ctr}$ でなくてはならない．これは，$\mathcal{A}$ が全ての再暗号化鍵を手に入れられること，すなわち攻撃者とクラウドの結託も想定していることを意味する．
- $\mathcal{O}_{\mathrm{TD}}$ にクエリする (i, w) は $i \leq \mathsf{ctr}$, かつ $w \notin \{w_0^*, w_1^*\}$ でなくてはならない．これは，$\mathcal{A}$ が可能な限りトラップドアを手に入れられること，すなわち攻撃者とクラウドの結託も想定していることを意味する．

定義 3.1 (IND-KU-CKA)　任意の確率的多項式時間攻撃者 $\mathcal{A}$ に対して，*KU-PEKS* Π が $\mathsf{Adv}^{\mathsf{KU\text{-}CKA}}_{\Pi,\mathcal{A}}(1^\lambda) := |\Pr[\mathsf{Exp}^{\mathsf{KU\text{-}CKA}}_{\Pi,\mathcal{A}}(1^\lambda) = 1] - 1/2| < \epsilon(\lambda)$ を満たすとき，Π は IND-KU-CKA を満たすという．

計算量的一貫性. 以下の試行 $\mathsf{Exp}^{\mathsf{KU\text{-}Cons}}_{\Pi,\mathcal{A}}(1^\lambda)$ を考える．

$\mathsf{Exp}^{\mathsf{KU\text{-}Cons}}_{\Pi,\mathcal{A}}(1^\lambda)$

$\mathsf{ctr} := 0, \quad \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$
$(\mathsf{sk}_1, \mathsf{pk}_1) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$
$(w_0^*, w_1^*) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{pp}, \mathsf{pk}_1)$ s.t. $|w_0^*| = |w_1^*|$
$\mathsf{c}^*_{w_0^*} \leftarrow \mathsf{Enc}(\mathsf{pk}_{\mathsf{ctr}}, w_0^*)$
$\mathsf{t}_{w_1^*} \leftarrow \mathsf{Trapdoor}(\mathsf{sk}_{\mathsf{ctr}}, w_1^*)$
If $\mathsf{Test}(\mathsf{td}_{w_1^*}, \mathsf{c}_{w_0^*}) = 1$ and $w_0^* \neq w_1^*$ return 1
else return 0

$|w^*| = |\tilde{w}^*|$ であり，$\mathcal{A}$ は IND-KU-CKA と同様の制限の下，同様のオラクル $\mathcal{O} = \{\mathcal{O}_{\mathrm{KG}}, \mathcal{O}_{\mathrm{KL}}, \mathcal{O}_{\mathrm{RK}}, \mathcal{O}_{\mathrm{TD}}\}$ にアクセスできる．

定義 3.2 (KU-Computational Consistency)　任意の確率的多項式時間攻撃者 $\mathcal{A}$ に対して，*KU-PEKS* Π が $\mathsf{Adv}^{\mathsf{KU\text{-}Cons}}_{\Pi,\mathcal{A}}(1^\lambda) := \Pr[\mathsf{Exp}^{\mathsf{KU\text{-}Cons}}_{\Pi,\mathcal{A}}(1^\lambda) = 1] < \epsilon(\lambda)$ を満たすとき，Π は計算量的一貫性 *(*KU-Computational Consistency*)* を満たすという．

4 構成

本節では，任意の PKE（公開鍵暗号）と PEKS（検索可能暗号）を用いた KU-PEKS の一般的構成を提案する．具体的には，PKE $\mathcal{PKE} = (\mathsf{PG}, \mathsf{G}, \mathsf{E}, \mathsf{D})$，及び PEKS $\mathcal{PEKS} = (\mathsf{Setup}_{\mathrm{PEKS}}, \mathsf{KeyGen}_{\mathrm{PEKS}}, \mathsf{Enc}_{\mathrm{PEKS}}, \mathsf{Trapdoor}_{\mathrm{PEKS}}, \mathsf{Test}_{\mathrm{PEKS}})$ を用いて，KU-PEKS $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{ReKeyGen}, \mathsf{Enc}, \mathsf{ReEnc}, \mathsf{Trapdoor}, \mathsf{Test})$ を以下のように構成する．以下では，キーワード集合 $\mathcal{W}$ と $\mathcal{PKE}$ の平文集合 $\mathcal{M}$ を同一視する．

- $\mathsf{Setup}(1^\lambda) \to \mathsf{pp}$: $\mathsf{par}_{\mathrm{PKE}} \leftarrow \mathsf{PG}(1^\lambda)$, $\mathsf{par}_{\mathrm{PEKS}} \leftarrow \mathsf{Setup}_{\mathrm{PEKS}}(1^\lambda)$ を実行し，$\mathsf{pp} := (\mathsf{par}_{\mathrm{PKE}}, \mathsf{par}_{\mathrm{PEKS}})$ を出力する．
- $\mathsf{KeyGen}(\mathsf{sk}_{i-1}, \mathsf{pk}_{i-1}) \to (\mathsf{sk}_i, \mathsf{pk}_i)$: $\mathsf{sk}_{i-1} = (\mathsf{dk}_{i-1}, \mathsf{msk}_{i-1})$, $\mathsf{pk}_{i-1} = (\mathsf{ek}_{i-1}, \mathsf{mpk}_{i-1})$ とする（$i = 1$ の時は空の文字列とする）．$(\mathsf{dk}_i, \mathsf{ek}_i) \leftarrow \mathsf{G}(\mathsf{par}_{\mathrm{PKE}})$, $(\mathsf{msk}_i, \mathsf{mpk}_i) \leftarrow \mathsf{KeyGen}_{\mathrm{PEKS}}(\mathsf{par}_{\mathrm{PEKS}})$ を実行し，$\mathsf{sk}_i := (\mathsf{dk}_i, \mathsf{msk}_i)$, $\mathsf{pk}_i := (\mathsf{ek}_i, \mathsf{mpk}_i)$ を出力する．
- $\mathsf{ReKeyGen}(\mathsf{pk}_i, \mathsf{sk}_i, \mathsf{pk}_{i+1}\mathsf{sk}_{i+1}) \to \mathsf{rk}_{i\to i+1}$: $\mathsf{sk}_i = (\mathsf{dk}_i, \mathsf{msk}_i)$, 及び $\mathsf{sk}_{i+1} = (\mathsf{dk}_{i+1}, \mathsf{msk}_{i+1})$ とする．$\mathsf{rk}_{i\to i+1} := \mathsf{dk}_i$ を出力する．
- $\mathsf{Enc}(\mathsf{pk}_i, w) \to \mathsf{c}_{w,i}^{(0)}$: $\mathsf{ct}_i \leftarrow \mathsf{E}(\mathsf{ek}_i, w)$, 及び $\mathsf{ct}_{w,i} \leftarrow \mathsf{Enc}_{\mathrm{PEKS}}(\mathsf{mpk}_i, w)$ を計算し，$\mathsf{c}_{w,i}^{(0)} := (\mathsf{ct}_i, \mathsf{ct}_{w,i})$ を出力する，
- $\mathsf{ReEnc}(\mathsf{rk}_{i\to i+1}, \mathsf{c}_{w,j}^{(k)}) \to \mathsf{c}_{w,j}^{(k+1)}$: $i \neq j + k$ であれば $\perp$ を出力する．そうでなければ，$\mathsf{rk}_{i\to i+1} = \mathsf{dk}_i$ とし，$\mathsf{c}_{w,j}^{(k)} = (\mathsf{ct}_i, \mathsf{ct}_{w,i})$ とし，$w \leftarrow \mathsf{D}(\mathsf{dk}_i, \mathsf{ct}_i)$ を計算する．$\mathsf{ct}_{i+1} \leftarrow \mathsf{E}(\mathsf{ek}_{i+1}, w)$ 及び $\mathsf{ct}_{w,i+1} \leftarrow \mathsf{Enc}_{\mathrm{PEKS}}(\mathsf{mpk}_{i+1}, w)$ を実行し，$\mathsf{c}_{w,j}^{(k+1)} := (\mathsf{ct}_{i+1}, \mathsf{ct}_{w,i+1})$ を出力する，
- $\mathsf{Trapdoor}(\mathsf{sk}_i, w') \to \mathsf{t}_{w',i}$: $\mathsf{sk}_i = (\mathsf{dk}_i, \mathsf{msk}_i)$ とし，$\mathsf{t}_{w',i} := \mathsf{td}_{w',i} \leftarrow \mathsf{Trapdoor}_{\mathrm{PEKS}}(\mathsf{msk}_i, w')$ を出力する．
- $\mathsf{Test}(\mathsf{t}_{w',i}, \mathsf{c}_{w,j}^{(k)}) \to 1$ or 0: $i \neq j + k$ であれば $\perp$ を出力する．$\mathsf{c}_{w,j}^{(k)} = (\mathsf{ct}_i, \mathsf{ct}_{w,i})$ とし，$\mathsf{Test}_{\mathrm{PEKS}}(\mathsf{t}_{w',i}, \mathsf{ct}_{w,i})$ の結果を出力する．

定理 4.1 $\mathcal{PKE}$ が IND-CPA を満たし，$\mathcal{PEKS}$ が IND-CKA を満たすならば，上記構成法による *KU-PEKS* Π は IND-KU-CKA を満たす．

証明． $\mathsf{Exp}_{\Pi,\mathcal{A}}^{\mathsf{KU\text{-}CKA}}$ を G_0 とし，$\mathsf{Exp}_{\Pi,\mathcal{A}}^{\mathsf{KU\text{-}CKA}} = 1$ となることを S_0 と書く．次の $\mathsf{G}_1, \mathsf{G}_2, \mathsf{G}_3$ を考え，同様に $\mathsf{S}_1, \mathsf{S}_2, \mathsf{S}_3$ とする．

G_1: G_0 において，チャレンジャーがランダムに $i^* \in \mathcal{T}$ を推測するゲーム．チャレンジフェーズ時点（$\mathcal{A}$ が (w_0^*, w_1^*) を出力する時点）において，$\mathsf{ctr} \neq i^*$ であるイベントを Fail とする．S_1 と Fail は独立事象であるから，$\Pr[\mathsf{S}_1 \mid \neg\mathsf{Fail}] = \Pr[\mathsf{S}_1]$ であり，この変更は $\mathcal{A}$ に何の影響も与えないので，$\Pr[\mathsf{S}_1] = \Pr[\mathsf{S}_0]$ である．

G_2: G_1 において，Fail が起きた場合に $\mathcal{A}$ の出力 b' をランダムビットに置き換えるゲーム．$\Pr[\mathsf{Fail}] = (|\mathcal{T}| - 1)/|\mathcal{T}|$, $\Pr[\mathsf{S}_2 \mid \neg\mathsf{Fail}] = \Pr[\mathsf{S}_1 \mid \neg\mathsf{Fail}] = \Pr[\mathsf{S}_1]$，及び $\Pr[\mathsf{S}_2 \mid \mathsf{Fail}] = 1/2$ であるから，

$$\begin{aligned}\left|\Pr[\mathsf{S}_2] - \frac{1}{2}\right| &= \left|\Pr[\mathsf{S}_2 \wedge \neg\mathsf{Fail}] + \Pr[\mathsf{S}_2 \wedge \mathsf{Fail}] - \frac{1}{2}\right| \\ &= \left|\frac{1}{|\mathcal{T}|} \cdot \Pr[\mathsf{S}_1 \mid \neg\mathsf{Fail}] + \frac{|\mathcal{T}| - 1}{|\mathcal{T}|} \cdot \frac{1}{2} - \frac{1}{2}\right| \\ &= \frac{1}{|\mathcal{T}|}\left|\Pr[\mathsf{S}_1] - \frac{1}{2}\right|.\end{aligned}$$

G_3: G_2 において，チャレンジ暗号文 $\mathsf{c}^{(0)}_{w_b^*,\mathsf{ctr}} = (\mathsf{ct}_i^*, \mathsf{ct}^*_{w_b^*,i})$ を作成する際に，$\mathsf{ct}_i^* \leftarrow \mathsf{E}(\mathsf{ek}_{\mathsf{ctr}}, w_b^*)$ とするところをランダムな暗号文とする (すなわち $\mathsf{ct}_i^* \xleftarrow{\$} \mathcal{C}_\lambda$) ゲーム．$\mathsf{ctr}$ $(= i^*)$ 番目の鍵ペアにおける $(\mathsf{dk}_{\mathsf{ctr}}, \mathsf{ek}_{\mathsf{ctr}})$ を $\mathsf{Exp}^{\mathsf{CPA}}_{\mathcal{PKE},\mathcal{B}}$ における鍵ペアとし，$\mathcal{PKE}$ の IND-CPA 安全性を破る攻撃者 $\mathcal{B}$ を構成することで，$|\Pr[\mathsf{S}_3] - \Pr[\mathsf{S}_2]| \leq \mathsf{Adv}^{\mathsf{CPA}}_{\mathcal{PKE},\mathcal{B}}$ であることを証明できる．

また，ctr $(= i^*)$ 番目の鍵ペアにおける $(\mathsf{msk}_{\mathsf{ctr}}, \mathsf{mpk}_{\mathsf{ctr}})$ を $\mathsf{Exp}^{\mathsf{CKA}}_{\mathcal{PEKS},\mathcal{B}}$ における鍵ペアとし，$\mathcal{PEKS}$ の IND-CKA 安全性を破る攻撃者 $\mathcal{B}$ を構成することで，$|\Pr[\mathsf{S}_3] - 1/2| \leq \mathsf{Adv}^{\mathsf{CKA}}_{\mathcal{PEKS},\mathcal{B}}$ であることを証明できる．

従って，$\mathsf{Adv}^{\mathsf{KU\text{-}CKA}}_{\Pi,\mathcal{A}} \leq |\mathcal{T}| \left(\mathsf{Adv}^{\mathsf{CPA}}_{\mathcal{PKE},\mathcal{B}} + \mathsf{Adv}^{\mathsf{CKA}}_{\mathcal{PEKS},\mathcal{B}}\right)$.

定理 4.2 $\mathcal{PKE}$ が IND-CPA を満たし，$\mathcal{PEKS}$ が Computational Consistency を満たすならば，上記構成法による *KU-PEKS* Π は KU-Computational Consistency を満たす．

証明． 定理 4.1 と同様に証明可能であるため省略．

上記構成法は一般的構成法であるため，それぞれベースとする PEKS（検索可能暗号）の特性により，ランダムオラクルモデルの KU-PEKS，スタンダードモデルの KU-PEKS を実現可能である．例えば，Boneh-Franklin (Anonymous) IBE [2] を基にした PEKS (+ PKE として ElGamal 暗号等を組み合わせる) を適用することで，ランダムオラクルモデルの KU-PEKS が得られ，Jutla-Roy Anonymous IBE [4, 6] を基にした PEKS(+ PKE として ElGamal 暗号等を組み合わせる) を適用することで，スタンダードモデルの KU-PEKS が実現できる．

5 実装性能の見積もり

本章では，4 章で説明した KU-PEKS の実装性能を見積もる参考として，Boneh-Franklin の PEKS（非対称ペアリング版）[2] をソフトウェアライブラリ TEPLA [5] を用いて実装した結果を示す．TEPLA は，筑波大学が開発を行ったオープンソースの C 言語暗号ライブラリである．楕円曲線暗号の演算や，ペアリング演算などを含む．表 1 に，PEKS の各関数ごとの処理時間を示す．計測の環境は，CPU:Intel Core i7-3770(3.40GHz)，RAM:31 GB，OS:Linux(Ubuntu 15.04, kernel 3.19.0-15-generic) である．また，性能評価には，電子メールデータでの利用を想定し，Enron データセット [7] を用いている．Enron データセットは，約 50 万個のメールを備えたデータセットであり，各メールの平均サイズは 2.68kB である．

表 1

PEKS[2] の関数ごとの処理時間

関数	処理時間 (msec)
Enc	11.20
Trapdoor	1.04
Test	4.71

TEPLA のような通常のペアリング演算ライブラリは並列処理は想定していない．そのため処理時間は直接プロセッサの周波数に依存する．例えば，IoT 組み込みデバイスとして利用される ARM の Cortex-M7 の周波数が 300MHz であることを考慮すれば，Enc と Trapdoor の処理時間は，表の数値の約 11.3 倍となる．つまり，Enc は 127msec，Trapdoor は 11.8msec と換算される．この数値は，IoT デバイスを想定している我々

のシナリオでは十分実現可能な処理時間であると考えられる．

6 まとめ

本稿では，IoT 機器を想定し，クライアントデバイスからの鍵漏洩に対策した，鍵更新機能付き検索可能暗号について概念と一般的構成を示した．一般的構成は，公開鍵が更新されるモデルにおいて，IND-CKA 安全性を持つ既存検索可能暗号と，IND-CPA 安全性を持つ公開鍵暗号を組み合わせている．また，構成の主な構成要素である Enc と Trapdoor について実装評価し，IoT デバイスにおいて十分実用的な方法であることを示した．

謝辞 本報告書は，主に文献 [9] の内容となっている．本報告書の報告者 2 名は，[9] の共著である，東邦大学大学院理学部情報科学科の金岡晃氏，および電気通信大学大学院情報理工学研究科の渡邉洋平氏に感謝の意を表する．

参考文献

[1] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi. Searchable encryption revisited: Consistency properties, relation to anonymous ibe, and extensions. *J. Cryptology*, 21(3):350–391, 2008.

[2] D. Boneh and M. K. Franklin. Identity-based encryption from the weil pairing. In *Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '01, pages 213–229, London, UK., 2001. Springer-Verlag.

[3] Y. Dodis, J. Katz, S. Xu, and M. Yung. Key-insulated public key cryptosystems. In *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques: Advances in Cryptology*, EUROCRYPT '02, pages 65–82, London, UK, UK, 2002. Springer-Verlag.

[4] C. S. Jutla and A. Roy. Shorter quasi-adaptive NIZK proofs for linear subspaces. In K. Sako and P. Sarkar, editors, *Advances in Cryptology – ASIACRYPT 2013*, volume 8269 of *Lecture Notes in Computer Science*, pages 1–20. Springer Berlin Heidelberg, 2013.

[5] U. of Tsukuba. Tepla: University of Tsukuba Elliptic, curve and Pairing Libraary, Jan.2013 Released TEPLA 1.0, Dec-2015 Released TEPLA 2.0. http://www.cipher.risk.tsukuba.ac.jp/tepla/.

[6] S. C. Ramanna and P. Sarkar. Efficient (anonymous) compact HIBE from standard assumptions. In S. Chow, J. Liu, L. Hui, and S. Yiu, editors, *Provable Security, ProvSec 2014*, volume 8782 of *Lecture Notes in Computer Science*, pages 243–258. Springer International Publishing, 2014.

[7] C. M. University. Enron email dataset, May 7, 2015. http://www.cs.cmu.edu/ enron/.

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

[9] 松崎なつめ, 穴田啓晃, 金岡晃, and 渡邉洋平. 鍵更新機能付き検索可能暗号の一般的構成. *Symposium on Cryptography and Information Security (SCIS2018)*, 2018.

[10] 渡邉洋平, 穴田啓晃, and 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.

＜発表資料＞

題名	掲載誌・学会名等	発表年月
鍵更新可能な検索可能暗号の一提案	電子情報通信学会信学技報 vol. 117, no. 25, ISEC2017-1	2017 年 5 月
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	コンピュータセキュリティシンポジウム 2017 論文集	2017 年 10 月
鍵更新機能付き検索可能暗号： 鍵隔離モデルによる実現	コンピュータセキュリティシンポジウム 2017 論文集	2017 年 10 月
鍵更新機能付き検索可能暗号の一般的構成	暗号と情報セキュリティシンポジウム 2018 論文集	2018 年 1 月