複数の暗号化索引を持つ共通鍵ベース秘匿検索の効率的なトラップドア生成

平野 貴人[1]　岩本 貢[2]　太田 和夫[2]

概要：Searchable Symmetric Encryption では，検索の効率化のために暗号化索引が使われる．ドキュメントを追加登録する際には，検索できるように暗号化索引に対しても処理する必要がある．既存手法として，既に生成された暗号化索引を部分的に更新する手法と，暗号化索引を新たに作成・追加する手法が知られている．後者の手法では，同じ秘密鍵を使って単純に暗号化索引を生成・追加すると追加ドキュメントの部分情報が漏れることが知られており，現在までにいくつかの安全な手法が提案されている．しかし，既存手法は検索クエリ（トラップドア）の生成コストに課題があった．本稿では，ハッシュ鎖を用いた効率的なトラップドア生成手法を提案する．

キーワード：検索可能暗号，共通鍵暗号，複数暗号化索引，トラップドア生成

Generating Trapdoors for Multiple Encrypted Indexes in Searchable Symmetric Encryption

Takato Hirano[1]　Mitsugu Iwamoto[2]　Kazuo Ohta[2]

Abstract: An encrypted index is often used in searchable symmetric encryption in order to reduce its search cost. When adding extra documents into a database, processing the encrypted index that has been already stored is required. There are two document adding approaches: One is to update a part of entries of the encrypted index, and the other is to generate a new encrypted index and to store in the database together with the encrypted indexes that have been already stored. In the latter approach, it is known that partial information on the extra documents is leaked if the new encrypted index is generated by the same secret key again. Therefore, several methods to address the above leakage problem have been proposed. However, the previous methods have a problem that computational cost on generating a search query (called trapdoor) is expensive. In this paper, we propose an efficient method using hash chains to reduce the trapdoor generation cost.

Keywords: Searchable symmetric encryption, Multiple encrypted indexes, Trapdoor generation

1. はじめに

企業内のデータを外部のクラウドサービスに保管することが現在普及しつつある．企業のデータは機微情報であるため，安全性の観点から暗号化して保管することが望ましい．一方，暗号化するとデータがかく乱されるため検索が困難になるといった問題が生じる．単純な解決策として，クラウドサーバ内で一度データを復号してから検索をする方法も考えられるが，クラウドサーバの悪意のある管理者やクラウドサーバに感染したマルウェアは，クラウドサーバ内で復号されたデータを盗み見れる可能性があるため，クラウドサーバ内で一瞬でも復号されることは安全性の観点からこの方法は好ましくない．この問題を解決する 1 つの手法として，検索可能暗号（秘匿検索とも呼ばれる）が提案され，近年活発に研究されている．

[1] 三菱電機株式会社
Mitsubishi Electric Corporation
[2] 電気通信大学
The University of Electro-Communications

秘匿検索は Song, Wagner, Perrig [27] によって提案され，現在までに様々な具体的な方式が提案されている．秘匿検索は共通鍵暗号ベースの方式 (例えば [1], [3], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33])，または公開鍵暗号ベースの方式 (例えば [4]) の 2 つに分類される．

共通鍵暗号ベースの秘匿検索（Searchable Symmetric Encryption，以降では SSE と呼ぶ）は，ドキュメント保管処理とキーワード検索処理からなり，これらの処理は同一人物によって実行される．近年は，ドキュメントの暗号化時に暗号化索引と呼ばれるデータも作成し，暗号化索引を用いて暗号化ドキュメントを効率的に検索する手法が主流である．

ドキュメント保管処理では，ユーザは秘密鍵を用いてドキュメントの集合を暗号化するとともに暗号化索引を生成し，暗号化ドキュメントの集合と暗号化索引をサーバに送信する．サーバは，受信したこれらのデータを保管する．

キーワード検索処理では，ユーザは，秘密鍵を用いて検索したいキーワードを暗号化してトラップドアと呼ばれる検索クエリを作成し，このトラップドアをサーバに送信する．サーバは，受信したトラップドアに基づき暗号化索引を検索すると，検索したいキーワードに対する検索結果（ドキュメント名）を得ることができ，この得られたドキュメント名をもつ暗号化ドキュメントをユーザに返す．このとき，ユーザは，秘密鍵を用いて暗号化ドキュメントを復号する．

SSE に対する安全性モデルも研究されている．Curtmola, Garay, Kamara, Ostrovsky が提案した安全性モデル [11] は，ドキュメント追加処理の安全性のみならず，キーワード検索処理の安全性も考慮しており，それゆえ広く利用されている．直感的には，[11] では，ドキュメント追加処理及びキーワード検索処理から漏れざるを得ない情報を抽出し，それ以上の情報が漏れない方式を安全と定義した．ここで，漏れざるを得ない具体的な情報とは，保管データのサイズ，あるトラップドアでヒットした検索結果，i 番目の検索で用いたトラップドアに含まれているキーワードと j 番目の検索で用いたトラップドアに含まれているキーワードが等しいか否か，の 3 種類の情報である．

1.1 動機

ドキュメントを追加登録することは一般的な処理であるため，そのような追加機能を持った SSE 方式が数多く提案されている．ドキュメントを追加登録する方法として，既に保管されている暗号化索引のエントリを部分的に更新して，追加ドキュメントも検索できるようにする方法（索引更新型の追加手法）と，新たに暗号化索引を生成してドキュメントのみならず暗号化索引も追加で登録する方法（索引追加型の追加手法）の 2 種類が存在する．前者は Dynamic SSE として広く知られている [1], [3], [5], [6], [7], [9], [10], [11], [12], [13], [14], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [28], [29], [30], [31], [32], [33]．本稿では，後者の暗号化索引の追加登録手法に注目し，考察する．

索引追加型の追加手法は，一般的な手法と考えられ，ドキュメント追加機能を持たない SSE 方式に対して容易に適用できると考えられる．単純な手法として，同じ秘密鍵を用いて，ドキュメント保管処理で生成した手順と全く同じように暗号化索引を生成する方法が考えられる．しかし，この方法は，追加した暗号化索引から追加ドキュメントの部分情報が漏れることが知られている．よって，この漏えい問題を回避するような手法がいくつか提案されている [8], [11], [34], [35].

[8], [11] では，ドキュメント追加時に秘密鍵も新たに生成し，その新しい秘密鍵を用いて暗号化索引を生成する手法が提案されている．ただし，本手法は，ドキュメント追加の度に秘密鍵が増えるため，秘密鍵サイズが増大するといった課題がある．加えて，トラップドア生成時に，保管されている複数の暗号化索引のそれぞれに対して独立にトラップドアを生成する必要があるため，トラップドア生成コストに課題がある．

[35] では，ドキュメント追加時に秘密鍵を毎回新たに生成する代わりに，カウンター値など公にしても問題ない情報を生成し，その情報を使って暗号化索引を構成することで，秘密鍵サイズが増えない手法を提案している．しかし，本手法も同様に，保管されている複数の暗号化索引のそれぞれに対して独立にトラップドアを生成する必要があるため，トラップドア生成コストに課題がある．

[34] では，トラップドア生成コストを下げるべく，RSA 暗号方式を使って構成している．彼らは，トラップドア生成時にユーザは唯 1 つのトラップドアを生成し，サーバはそのトラップドアから暗号化索引それぞれに対応したトラップドアを生成するといった戦略に従って既存方式よりも効率的な手法を構成した．しかし，RSA 暗号方式は公開鍵暗号プリミティブのため，擬似ランダム関数やハッシュ関数などの共通鍵暗号プリミティブよりも処理が重いため，未だにトラップドア生成コストの課題は解決できていない．

1.2 貢献

本稿は，[34], [35] の手法に注目し，1 つのトラップドア生成のみで十分な手法を提案する．具体的には，トラップドアはあるハッシュ値として生成され，ハッシュ鎖を用いて各暗号化索引のトラップドアを生成できる手法を提案する．なお，ハッシュ値を用いて暗号化索引を追加する手法が既に [30] で提案されているが，[35] のようにカウンター値が考慮されていないことや，ドキュメント追加を考慮し

た安全性証明が検討されていない．本稿では，[35] で提案されたドキュメント追加を考慮した安全性モデルの下で安全性を考察する．正確には，提案手法の安全性を示すために，本手法を [9] で提案された既存 SSE 方式に適用した方式に対して安全性証明を行う．このとき，ランダムオラクルモデルの下でその安全性が証明できる．

2. 準備

正の実数の集合を $\mathbb{R}^+$ とする．関数 $\mathtt{negl}:\mathbb{N}\to\mathbb{R}^+$ が無視できるとは，正値をとる任意の多項式 $poly$ に対して，ある整数 $n_0\in\mathbb{N}$ が存在し，任意の整数 $n\geq n_0$ に対して $\mathtt{negl}(n)<1/poly(n)$ が成立することである．A を確率的アルゴリズムとする．このとき，$y\leftarrow A(x)$ は，一様に選ばれた乱数テープと入力 x に対して A は y を出力することを意味する．また，$A^{\mathcal{O}}$ は，A は $\mathcal{O}$ にオラクルアクセス可能であることを意味する．S を有限集合とする．このとき，$s\xleftarrow{u}S$ は，s は S から一様に選ばれたことを意味する．

共通鍵暗号の *Left-Or-Right Indistinguishability against the Chosen Plaintext Attack (LOR-CPA)* と呼ばれる安全性の定義は以下の通りである [2].

定義 1 共通鍵暗号方式 $\mathtt{SKE}=(\mathtt{Gen},\mathtt{Enc},\mathtt{Dec})$ が LOR-CPA の意味で安全であるとは，任意の PPT の攻撃者 $\mathcal{A}$ に対して，$|\Pr[1\leftarrow\mathcal{A}^{\mathtt{Enc}_K(\mathcal{LR}(\cdot,\cdot,1))}(1^\lambda)\mid K\leftarrow\mathtt{Gen}(1^\lambda)]-\Pr[1\leftarrow\mathcal{A}^{\mathtt{Enc}_K(\mathcal{LR}(\cdot,\cdot,0))}(1^\lambda)\mid K\leftarrow\mathtt{Gen}(1^\lambda)]|\leq\mathtt{negl}(\lambda)$ が成立することをいう．ただし，$\mathtt{Enc}_K(\mathcal{LR}(\cdot,\cdot,b))$ は，入力 (x_0,x_1) に対して，もし $b=0$ であれば $C_0\leftarrow\mathtt{Enc}_K(x_0)$ を返し，もし $b=1$ であれば $C_1\leftarrow\mathtt{Enc}_K(x_1)$ を返す Left-or-Right オラクルである．

3. 暗号化索引の追加を考慮した SSE の定義

本章では，[35] で与えられた暗号化索引の追加してドキュメント追加することを考慮した SSE（Index Generation type SSE，以降では IG-SSE と呼ぶ）の定義を述べる．

3.1 SSE の定義

IG-SSE の定義を与えるにあたって，必要な概念をいくつか定義する．

- w をキーワードとし，$\Delta\subseteq\{0,1\}^\ell$ をキーワードの集合とする．ここで，$\#\Delta=d$ とする．
- $D\in\{0,1\}^*$ をドキュメントとし，$\mathbf{D}=(D_1,\ldots,D_n)$ をドキュメントの集合とする．$\mathbf{C}=(C_1,\ldots,C_n)$ を $\mathbf{D}$ の暗号文の集合とする．ただし，C_i は D_i $(1\leq i\leq n)$ の暗号文である．本稿では，暗号文 C_i は一意的な識別子（ファイル名のようなもの）を持つとし，$id(D_i)\in\{0,1\}^\ell$ と表わす．
- $\mathbf{D}=(D_1,\ldots,D_n)$ に対して，$\mathbf{D}(w)$ をキーワード w を含むドキュメントの識別子の集合とする．すなわち，$\mathbf{D}(w)=\{id(D_{i_1}),\ldots,id(D_{i_m})\}$ と表わされる．また，検索列 $\mathbf{w}=(w_1,\ldots,w_q)$ に対して，$\mathbf{D}(\mathbf{w})=(\mathbf{D}(w_1),\ldots,\mathbf{D}(w_q))$ と表す．

Δ 上の IG-SSE 方式 $\mathtt{IG-SSE}=(\mathtt{Gen},\mathtt{Enc},\mathtt{Trpdr},\mathtt{Search},\mathtt{Dec})$ とは，次のように定義される．

- $K\leftarrow\mathtt{Gen}(1^\lambda)$ は，秘密鍵 K を出力する確率的アルゴリズムである．ただし，λ はセキュリティパラメータである．
- $(\mathcal{I},\mathbf{C})\leftarrow\mathtt{Enc}(K,\mathbf{D})$ は，暗号化索引 $\mathcal{I}$ と暗号文の集合 $\mathbf{C}=(C_1,\ldots,C_n)$ を出力する確率的アルゴリズムである．
- $t(w)\leftarrow\mathtt{Trpdr}(K,w)$ は，キーワード w に対するトラップドア $t(w)$ を出力する確定的アルゴリズムである．
- $S(w)\leftarrow\mathtt{Search}(\mathcal{I},t(w))$ は，識別子の集合 $S(w)$ を出力する確定的アルゴリズムである．
- $D\leftarrow\mathtt{Dec}(K,C)$ は，ドキュメント D の暗号文 C を出力する確定的アルゴリズムである．

ここで，SSE 方式が correct とは，すべての $\lambda\in\mathbb{N}$，すべての $w\in\Delta$，$\mathtt{Gen}(1^\lambda)$ が出力するすべての K，すべてのドキュメント $\mathbf{D}$，$\mathtt{Enc}(K,\mathbf{D})$ が出力するすべての $(\mathcal{I},\mathbf{C})$ に対して，$\mathtt{Search}(\mathcal{I},\mathtt{Trpdr}(K,w))=\mathbf{D}(w)$ かつ $\mathtt{Dec}(K,C_i)=D_i$ $(1\leq i\leq n)$ が成立するときをいう．

3.2 SSE の安全性の定義

次に IG-SSE の安全性について述べる．

- $\mathbf{D}^{(i)}=(D_1^{(i)},\ldots,D_{n_i}^{(i)})$ を i 番目に追加したドキュメント集合とし，$\mathbf{w}^{(i)}=(w_1^{(i)},\ldots,w_{q_i}^{(i)})$ を i 番目のドキュメント追加処理直後から $i+1$ 番目のドキュメント追加処理直前までの検索列とする．
- i 番目のヒストリ $H^{(i)}$ を $(\mathbf{D}^{(i)},\mathbf{w}^{(i)})$ と定義する．言い換えると，$H^{(i)}$ は i 番目のドキュメント追加処理から $i+1$ 番目のドキュメント追加処理前までの機微情報である．ヒストリ $\mathbf{H}^{(k)}$ を $(H^{(1)},\ldots,H^{(k)})$ と定義する．ただし，$k\in\mathbb{Z}$ かつ $k\geq 0$ とする．
- ドキュメント集合 $\mathbf{D}^{(i)}=(D_1^{(i)},\ldots,D_{n_i}^{(i)})$ の暗号化ドキュメント集合 $\mathbf{C}^{(i)}=(C_1^{(i)},\ldots,C_{n_i}^{(i)})$ から，ドキュメントのビット長 $|D_1^{(i)}|,\ldots,|D_{n_i}^{(i)}|$ が推測できる．この情報は，サーバに明らかになる情報である．本情報を $\mathcal{L}_1(\mathbf{D}^{(i)})$ と表現する．
- ヒストリ $\mathbf{H}^{(k)}$ に対するアクセスパターン $\mathcal{L}_2(\mathbf{H}^{(k)})$ を $\mathbf{w}^{(1)},\ldots,\mathbf{w}^{(k)}$ の検索結果から得られた識別子の集合と定義する．すなわち，

$$\begin{array}{rccc}\mathcal{L}_2(\mathbf{H}^{(k)})= & \mathbf{D}^{(1)}(\mathbf{w}^{(1)}) & \cup\ldots\cup & \mathbf{D}^{(1)}(\mathbf{w}^{(k)})\\ \cup & \emptyset^{q_1} & \cup\ldots\cup & \mathbf{D}^{(2)}(\mathbf{w}^{(k)})\\ \cup & \emptyset^{q_1} & \cup\ldots\cup & \mathbf{D}^{(k)}(\mathbf{w}^{(k)}),\end{array}$$

ただし，$\emptyset^q$ は長さ q のベクトル $(\emptyset,\ldots,\emptyset)$ を意味する．

ここで，検索結果 $\mathbf{D}^{(2)}(\mathbf{w}^{(1)})$ は空集合であることに注意する．これは，$\mathbf{w}^{(1)}$ は $\mathbf{D}^{(2)}$ を追加する前の検索列であるためである [*1]．$\mathcal{L}_2(\mathbf{H}^{(k)})$ はサーバに明らかになる情報である．

- ヒストリ $\mathbf{H}^{(k)}$ の $\mathbf{w}^{(1)},\ldots,\mathbf{w}^{(k)}$ に対して，$\mathbf{w} = \mathbf{w}^{(1)}||\cdots||\mathbf{w}^{(k)} = (w_1,\ldots,w_Q)$ とおく．ただし，$Q = \sum_{i=1}^{k} q_i$ とする．ヒストリ $\mathbf{H}^{(k)}$ に対する検索パターン $\mathcal{L}_3(\mathbf{H}^{(k)})$ を，$1 \le i \le j \le q$ に対して，i 行 j 列の要素が，$w_i = w_j$ であれば 1，そうでなければ 0 のバイナリ対称行列と定義する．既存の多くの方式はトラップドアが確定的に生成されるため，この情報はサーバに明らかになる情報である．更に，$\mathcal{L}_3(\mathbf{H}^{(k)})$ の情報から $\mathbf{w} = (w_1,\ldots,w_Q)$ の各要素を順序付けすることができる．例えば，$\mathbf{w} = (w_1,w_2,w_3,w_4,w_5) = (a,c,a,a,b)$ とする．この時，$t(w_1) = t(w_3) = t(w_4)$ がわかるため，$(1,2,1,1,3)$ と順序付けすることができる．この順序付けされた情報を検索パターン順序と呼び，$\gamma(\mathbf{w})$ で表わす．ここで，$\gamma(\mathbf{w}) = (z_1,\ldots,z_J) \in \mathbb{Z}^Q$ に対して，$a_i \le d$ $(1 \le i \le Q)$ が成立する．ただし，z_i は正の整数で，d は Δ の要素数である．$\gamma(\mathbf{w})[i] = z_i$ とし，また $\gamma(\mathbf{w})(w)$ を $\mathbf{w}[i] = w$ となるような $\gamma(\mathbf{w})[i]$ とする．例えば，$\gamma(\mathbf{w})(a) = 1$，$\gamma(\mathbf{w})(b) = 3$，$\gamma(\mathbf{w})(c) = 2$ である．
- ヒストリ $\mathbf{H}^{(k)}$ に対する追加パターン $\mathcal{L}_4(\mathbf{H}^{(k)})$ を，$(q_1,\ldots,q_k)$ と定義する．この情報は，ドキュメント追加処理で現れるため，サーバに明らかになる情報である．
- $\mathbf{H}^{(k)}$ に対するトレース $\mathcal{L}(\mathbf{H}^{(k)})$ を，$(\mathcal{L}_1(|\mathbf{D}^{(1)}|,\ldots,|\mathbf{D}^{(k)}|),\mathcal{L}_2(\mathbf{H}^{(k)}),\mathcal{L}_3(\mathbf{H}^{(k)}),\mathcal{L}_4(\mathbf{H}^{(k)}))$ と定義する．言い換えると，$\mathcal{L}(\mathbf{H}^{(k)})$ は，ドキュメント追加可能な SSE において許容できる最大の情報漏洩である．
- ヒストリ $\mathbf{H}^{(k)}$ が非特異であるとは，(1) $\mathcal{L}(\mathbf{H}^{(k)}) = \mathcal{L}(\mathbf{H}'^{(k)})$ となるような $\mathbf{H}'^{(k)} \neq \mathbf{H}^{(k)}$ が少なくとも 1 つ存在すること，また (2) $\mathcal{L}(\mathbf{H}^{(k)})$ に対して，$\mathbf{H}'^{(k)}$ は多項式時間で見つけることが可能である．以降では，本稿で扱うすべてのヒストリは非特異と仮定する．

[35] では，[11] の安全性をベースに以下のように安全性を定義した．

定義 2 $\mathtt{IG-SSE}$ を IG-SSE 方式，λ をセキュリティパラメータとする．また，$\mathcal{A}$ と $\mathcal{S}$ をステートフルな PPT アルゴリズムとする．このとき，以下のような 2 つの確率的な試行 $\mathbf{Real}_{\mathcal{A}}(\lambda), \mathbf{Ideal}_{\mathcal{A},\mathcal{S}}(\lambda)$ を考える：

$\mathbf{Real}_{\mathcal{A}}(\lambda)$: チャレンジャー $\mathcal{C}$ は，秘密鍵 K を得るために $\mathtt{Gen}(1^\lambda)$ を動作させる．$\mathcal{A}$ はドキュメント集合 $\mathbf{D} = \{D_1,\ldots,D_n\}$ を選んで $\mathcal{C}$ に渡し，$\mathcal{C}$ は $(\mathcal{I},\mathbf{C})$ を生成するために $\mathtt{Enc}(K,\mathbf{D})$ を動作させ，$(\mathcal{I},\mathbf{C})$ を $\mathcal{A}$ に渡す．その後 $\mathcal{A}$ は，次のような多項式回 $(poly(\lambda))$ の適応的なクエリを生成する：(1) $\mathcal{A}$ は w を選び $\mathcal{C}$ に渡し，$\mathcal{C}$ は $t(w)$ を得るために $\mathtt{Trpdr}(K,w)$ を動作させ，$t(w)$ を $\mathcal{A}$ に渡す．もしくは，(2) $\mathcal{A}$ は $\mathbf{D}' = \{D'_1,\ldots,D'_{n'}\}$ を選び $\mathcal{C}$ に渡し，$\mathcal{C}$ は $(\mathcal{I}',\mathbf{C}')$ を得るために $\mathtt{Enc}(K,\mathbf{D}')$ を動作させ，$(\mathcal{I}',\mathbf{C}')$ を $\mathcal{A}$ に渡す．最後に $\mathcal{A}$ は $b \in \{0,1\}$ を出力する．

$\mathbf{Ideal}_{\mathcal{A},\mathcal{S}}(\lambda)$: $\mathcal{A}$ はドキュメント集合 $\mathbf{D} = \{D_1,\ldots,D_n\}$ を選んで $\mathcal{S}$ に渡し，$\mathcal{S}$ は $\mathcal{L}_1(\mathbf{D})$ に基づき $(\mathcal{I},\mathbf{C})$ を生成し，$(\mathcal{I},\mathbf{C})$ を $\mathcal{A}$ に渡すとともに，$\mathbb{D}$ に $\mathbf{D}$，$\mathbb{I}$ に $\mathcal{I}$ を追加する．その後 $\mathcal{A}$ は，次のような多項式回 $(poly(\lambda))$ の適応的なクエリを生成する：(1) $\mathcal{A}$ は w を選び $\mathcal{S}$ に渡し，$\mathcal{S}$ は $\mathcal{L}_2(\mathbb{D},\mathbb{I},\mathbf{w}),\mathcal{L}_3(\mathbb{D},\mathbb{I},\mathbf{w}),\mathcal{L}_4(\mathbb{D},\mathbb{I},\mathbf{w})$ に基づき $t(w)$ を生成し，$t(w)$ を $\mathcal{A}$ に渡すとともに，$\mathbf{w}$ に w を加えて更新する．もしくは，(2) $\mathcal{A}$ は $\mathbf{D}' = \{D'_1,\ldots,D'_{n'}\}$ を選び $\mathcal{S}$ に渡し，$\mathcal{S}$ は $\mathcal{L}_1(\mathcal{D}')$ に基づき $(\mathcal{I}',\mathbf{C}')$ を生成し，$(\mathcal{I}',\mathbf{C}')$ を $\mathcal{A}$ に渡すとともに，$\mathbb{D}$ に $\mathbf{D}'$，$\mathbb{I}$ に $\mathcal{I}'$ を追加する．最後に $\mathcal{A}$ は $b \in \{0,1\}$ を出力する．

$\mathtt{IG-SSE}$ が Adaptive Semantic Secure であるとは，任意のセキュリティパラメータ λ，任意の PPT アルゴリズム $\mathcal{A}$ に対して，

$$|\Pr[\mathbf{Real}_{\mathcal{A}}(\lambda) = 1] - \Pr[\mathbf{Ideal}_{\mathcal{A},\mathcal{S}}(\lambda) = 1] \le \mathtt{negl}(\lambda)$$

を満たす PPT アルゴリズム $\mathcal{S}$ が存在することをいう．

4. 構成方法

1 度に追加できるドキュメントの最大数を N とする．キーワード w に対するヒットドキュメントのバイナリ表現 $\vec{b}_w$ とは，例えば，$N = 5$，また D_1, D_2, D_3 を登録するドキュメントとし，w に対して D_1 と D_3 が引っかかる場合，$b_w = 10100$ と表わす．ハッシュ鎖の (現在の) 回数を表すパラメータを c, ハッシュチェーンの上限を M, c の初期値を 0 とする．$H_1 : \{0,1\}^* \to \{0,1\}^\lambda$ と $H_2 : \{0,1\}^* \to \{0,1\}^N$ をハッシュ関数とする．

このとき，[9] の SSE 方式に我々のドキュメント追加手法を適した方式を以下に挙げる．

- $\mathtt{Gen}(1^\lambda)$: $K_1, K_2 \xleftarrow{u} \{0,1\}^\lambda$ とし，$K_3 \leftarrow \mathtt{SKE.Gen}(1^\lambda)$ を生成し，$K = (K_1,K_2,K_3)$ を出力する．
- $\mathtt{Enc}(K,\mathbf{D}^{(c)},c)$:
  (1) $c = c+1$ とする．もし，$c > M$ ならば，これ以上追加できない旨を出力する．
  (2) $\mathbf{D}^{(c)} = \{D_1^{(c)},\ldots,D_n^{(c)}\}$ に対して，$C_i^{(c)} = \mathtt{SKE.Enc}(K_3, D_i^{(c)})$ $(1 \le i \le n)$ を計算し，$\mathbf{C}^{(c)} = (C_1^{(c)},\ldots,C_n^{(c)})$ と置く．
  (3) $\mathcal{I}^{(c)} = \emptyset$ とする
  (4) 各 $w \in \Delta$ に対して，以下のように処理する．
    (a) $\mathbf{D}^{(c)}(w) = \{id(D_{i_1}^{(c)}),\ldots,id(D_{i_m}^{(c)})\}$ に対し

[*1] このことは [28], [34] のように Forward Privacy を考慮している．

て，ヒットドキュメントのバイナリ表現 $b_w^{(c)} \in \{0,1\}^N$ を求める．

(b) $addr_w^{(c)} = H^{M-c+1}(K_1||w)$ を計算する．

(c) $val_w^{(c)} = b_w^{(c)} \oplus H^{M-c+1}(K_2||w)$ を計算する．

(d) $(addr_w^{(c)}, val_w^{(c)})$ を $\mathcal{I}^{(c)}$ へ挿入する．

(5) $(\mathcal{I}^{(c)}, \mathbf{C}^{(c)})$ を出力する．

- $\texttt{Trpdr}(K, w, c)$:

(1) $t_1 = H^{M-c+1}(K_1||w)$ を計算する．

(2) $t_2 = H^{M-c+1}(K_2||w)$ を計算する．

(3) (t_1, t_2) を出力する．

- $\texttt{Search}(\mathcal{I}, t, c)$:

(1) $\mathcal{I} = (\mathcal{I}^{(1)}, \ldots, \mathcal{I}^{(c)})$ とみなす．

(2) $List = \emptyset$ とする．

(3) $1 \leq i \leq c$ に対して，下記を行う：

(a) $t_1^{(i)} = H^{c-i}(t) (= H^{M+1-i}(K_1||t))$ を計算する．

(b) $t_2^{(i)} = H^{c-i}(t) (= H^{M+1-i}(K_2||t))$ を計算する．

(c) もし $t_1^{(i)} = addr^{(i)}$ となるような $(addr^{(i)}, val^{(i)}) \in \mathcal{I}^{(i)}$ が存在したら，$b^{(i)} = val^{(i)} \oplus t_2^{(i)}$ を計算し，$b^{(i)}$ に対応する $id(D_{j_1}^{(i)}), \ldots, id(D_{j_m}^{(i)})$ を $List$ に追加する．

(4) $List$ を出力する．

- $\texttt{Dec}(K, C)$: $D \leftarrow \texttt{SKE.Dec}(K_3, C)$ を計算し，D を出力する．

このとき次の安全性が示せる．ただし，証明は省略する．

定義 3 もし SKE が LOR-CPA 安全であれば，上記の SSE 方式はランダムオラクルモデルの下で Adaptive Semantic Secure である．

5. まとめ

本稿では，暗号化索引追加型のドキュメント追加手法に注目し，既存手法 [8], [11], [34], [35] はトラップドア生成コストが非効率であることを見出し，ハッシュ鎖を用いた追加手法を提案した．また，[9] の SSE 方式に提案手法を適用したドキュメント追加可能な方式は，ランダムオラクルモデルの下で Adaptive Semantic Secure であることを示した．

参考文献

[1] G. Asharov, M. Naor, G. Segev, and I. Shahaf. Searchable symmetric encryption: Optimal locality in linear space via two-dimensional balanced allocations. In *STOC 2016*, 2016.

[2] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A concrete security treatment of symmetric encryption. In *FOCS 1997*, pages 394–403, 1997.

[3] A. Boldyreva and N. Chenette. Efficient fuzzy search on encrypted data. In *FSE 2014*, volume 8540 of *LNCS*, pages 613–633, 2014.

[4] D. Boneh, G. di Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. In *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 506–522, 2004.

[5] D. Cash, J. Jaeger, S. Jarecki, C. Jutla, H. Krawczyk, M.C. Roşu, and M. Steiner. Dynamic searchable encryption in very-large databases: Data structures and implementation. In *NDSS 2014*, 2014.

[6] D. Cash, S. Jarecki, C. Jutla, H. Krawczyk, M.C. Roşu, and M. Steiner. Highly-scalable searchable symmetric encryption with support for boolean queries. In *CRYPTO 2013*, volume 8042 of *LNCS*, pages 353–373, 2013.

[7] D. Cash and S. Tessaro. The locality of searchable symmetric encryption. In *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 351–368, 2014.

[8] Y. Chang and M. Mitzenmacher. Privacy preserving keyword searches on remote encrypted data. In *ACNS 2005*, volume 3531 of *LNCS*, pages 442–455, 2005.

[9] M. Chase and S. Kamara. Structured encryption and controlled disclosure. In *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 577–594, 2010.

[10] M. Chase and E. Shen. Substring-searchable symmetric encryption. *PETS 2015*, 2015(2):263–281, 2015.

[11] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky. Searchable symmetric encryption: Improved definitions and efficient constructions. In *ACM CCS 2006*, pages 79–88, 2006.

[12] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky. Searchable symmetric encryption: Improved definitions and efficient constructions. *Journal of Computer Security*, 19(5):895–934, 2011.

[13] C. Dong, G. Russello, and N. Dulay. Shared and searchable encrypted data for untrusted servers. *Journal of Computer Security*, 19(3):367–397, 2011.

[14] S. Faber, S. Jarecki, H. Krawczyk, Q. Nguyen, M. Rosu, and M. Steiner. Rich queries on encrypted data: Beyond exact matches. In *ESORICS 2015*, volume 9327 of *LNCS*, pages 123–145, 2015.

[15] E-J. Goh. Secure indexes. Cryptology ePrint Archive, Report 2003/216, 2003.

[16] F. Hahn and F. Kerschbaum. Searchable encryption with secure and efficient updates. In *ACM CCS 2014*, pages 310–320, 2014.

[17] S. Kamara, C. Papamanthou, and T. Roeder. Dynamic searchable symmetric encryption. In *ACM CCS 2012*, pages 965–976, 2012.

[18] S. Kamara and C. Papamonthou. Parallel and dynamic searchable symmetric encryption. In *FC 2013*, volume 7859 of *LNCS*, pages 258–274, 2013.

[19] K. Kurosawa. Garbled searchable symmetric encryption. In *FC 2014*, volume 8437 of *LNCS*, pages 234–251, 2014.

[20] K. Kurosawa and Y. Ohtaki. UC-secure searchable symmetric encryption. In *FC 2012*, volume 7397 of *LNCS*, pages 285–298, 2012.

[21] K. Kurosawa and Y. Ohtaki. How to update documents *verifiably* in searchable symmetric encryption. In *CANS 2013*, volume 8257 of *LNCS*, pages 309–328, 2013.

[22] M. Kuzu, M. S. Islam, and M. Kantarcioglu. Efficient similarity search over encrypted data. In *IEEE ICDE 2012*, pages 1156–1167, 2012.

[23] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou. Fuzzy keyword search over encrypted data in cloud computing. In *IEEE INFOCOM 2010 (Mini-Conference)*, pages 1–5, 2010.

[24] T. Moataz and A. Shikfa. Boolean symmetric searchable

encryption. In *ASIACCS 2013*, pages 265–276, 2013.

[25] M. Naveed, M. Prabhakaran, and C. A. Gunter. Dynamic searchable encryption via blind storage. In *IEEE S&P 2014*, pages 639–654, 2014.

[26] W. Ogata, K. Koiwa, A. Kanaoka, and S. Matsuo. Toward practical searchable symmetric encryption. In *IWSEC 2013*, volume 8231 of *LNCS*, pages 151–167, 2013.

[27] D. Song, D. Wagner, and A. Perrig. Practical techniques for searching on encrypted data. In *IEEE S&P 2000*, pages 44–55, 2000.

[28] E. Stefanov, C. Papamanthou, and E. Shi. Practical dynamic searchable encryption with small leakage. In *NDSS 2014*, 2014.

[29] S. Taketani and W. Ogata. Improvement of UC secure searchable symmetric encryption scheme. In *IWSEC 2015*, volume 9241 of *LNCS*, pages 135–152, 2015.

[30] P. van Liesdonk, S. Sedghi, J. Doumen, P. Hartel, and W. Jonker. Computationally efficient searchable symmetric encryption. In *SDM 2010*, volume 6358 of *LNCS*, pages 87–100, 2010.

[31] C. Wang, K. Ren, S. Yu, and K. M. R. Urs. Achieving usable and privacy-assured similarity search over outsourced cloud data. In *IEEE INFOCOM 2012*, pages 451–459, 2012.

[32] Y. J. Yang, X. H. Ding, R. H. Deng, and F. Bao. Multi-user private queries over encrypted databases. *International Journal of Applied Cryptography*, 1(4):309–319, 2009.

[33] A. A. Yavuz and J. Guajardo. Dynamic searchable symmetric encryption with minimal leakage and efficient updates on commodity hardware. In *SAC 2015*, volume 9566, pages 241–259, 2015.

[34] 佐藤 and 大瀧. Forward privacy を考慮した動的な検索可能暗号. In *SCIS 2015 (4B2-3)*, 2015.

[35] 平野, 森, 服部, 伊藤, 松田, 川合, 坂井, and 太田. Searchable symmetric encryption のドキュメント追加後の安全性について. In *SCIS 2012 (2B3-1)*, 2012.