1X-03

検証可能な機能に向けた動的検索可能暗号

小澤 響平† 山本 博章† 藤原 洋志† 三好 竜司‡

信州大学† 三菱電機株式会社‡

1. はじめに

近年クラウドサービスの普及によってデータを外部のサーバに保存する機会が増えている。このようなデータを悪意のある第三者から守るために、データを暗号化して保存することが必要であるが、暗号化した状態では検索を行うことができない。そのため暗号化したまま検索できる技術が研究されており、このような技術を検索可能暗号と呼ぶ。三好の手法 [1] によって、並列処理を用いた効率的な動的検索可能暗号方式が提案されたが、サーバが正しい検索結果を返さない場合、その不正を検出することはできない。不正を検出できる手法として Kurosawa らの手法 [2] があげられる。本論文では悪意のあるサーバが正しい検索結果を返さない場合に、その不正を検出できるように三好の手法を改良する。

2. 準備

$a||b$ を文字列 a と b を連結した文字列、集合 d において $n = |d|$ を d の要素数、さらに、$\lfloor x \rfloor$ を x 以下の最大の整数とする。本論文において、ドキュメントを d、暗号化ドキュメントを c、ドキュメントの集合を $D = \{d_0, \ldots, d_{n-1}\}$ とする。また、id はドキュメント ID であり、キーワード w を含むドキュメント ID の集合を $D(w) = \{id_0, \ldots, id_{n_w-1}\}$、キーワード w を含む暗号文の集合を $C(w) = \{c_0, \ldots, c_{n_w-1}\}$ とする。さらに、$K(D)$ は D に含まれる異なるキーワードの集合であり、$m = |K(D)|$ とする。sk を秘密鍵とすると、本手法ではトラップドアの作成に疑似ランダム関数 F_{sk}、暗号化索引の作成にランダムオラクル H、ドキュメントの暗号化に、CPA-安全性を満たす共通鍵暗号 Enc_{sk} を使用する。

3. 提案方式

提案方式は以下の 11 個の多項式時間 (確率) アルゴリズムで構成される。

- $\mathsf{KeyGen}(1^\lambda)$ はセキュリティーパラメータを入力とし、秘密鍵 $SK = (sk_1, sk_2, sk_3)$ を返す確率的アルゴリズムである。
- $\mathsf{Enc}(sk_1, d)$ は秘密鍵 sk_1 とドキュメント d を入力とし、暗号化されたドキュメント $c = E_{sk_1}(d)$ を返す確率的アルゴリズムである。
- $\mathsf{Dec}(sk_1, c)$ は秘密鍵 sk_1 と暗号化されたドキュメント c を入力とし、復号されたドキュメントを返す決定的アルゴリズムである。
- $\mathsf{BuildIndex}(K, D, \varepsilon)$ は秘密鍵 K、ドキュメントの集合 D、パラメータ ε を入力とし、暗号化索引 I_s, I_c を返す確率的アルゴリズムである。
- $\mathsf{Trpdr}(sk_2, w, SCnt)$ は秘密鍵 sk_2 とキーワード w と w の検索回数 $SCnt$ を入力とし、トラップドア T_w を返す確率的アルゴリズムである。このアルゴリズムを $\mathsf{Trpdr}_{sk_2}(w, Scnt)$ と書く。
- $\mathsf{Search}(I_s, T_w)$ は暗号化索引 I_s とトラップドア T_w を入力とし、w の含まれるドキュメントの暗号文の集合の $C(w)$ と検証用タグの集合 Tag_w を返す決定的アルゴリズムである。
- $\mathsf{Verify}(sk_3, T_w, C(w), Tag_w, I_c)$ は秘密鍵 sk_3 とキーワード w のトラップドア T_w と w の含まれるドキュメントの暗号文の集合 $C(w)$ とタグ Tag_w と暗号化索引 I_c を入力とし、`accept`/`reject` を返す決定的アルゴリズムである。
- $\mathsf{MakeUpdateToken}(SK, D(w), w, I_c, \Phi)$ は秘密鍵 sk_2、ドキュメント ID 集合 $D(w)$、キーワード w、暗号化索引 I_c、I_s の空要素のアドレスの集合 Φ を入力とし、更新トークン μ を返す確率的アルゴリズムである。
- $\mathsf{Updata}(I_s, \mu)$ は暗号化索引 I_s と、更新トークン μ を入力とし、更新された暗号化索引 I_s^* を返す決定的アルゴリズムである。
- $\mathsf{MakeAddToken}(SK, d_\aleph, I_c, \Phi)$ は秘密鍵 SK と追加するドキュメント d_α と暗号化索引 I_c と Φ を入力とし、追加トークン π を返す確率的アルゴリズムである。このアルゴリズムを $\mathsf{MakeAddToken}_{SK}(d_\alpha)$ と書く。
- $\mathsf{Add}(I_s, \pi)$ は暗号化索引 I_s と追加トークン π を入力とし、更新された暗号化索引 I_s^* を返す決定的アルゴリズムである。

[安全性] 提案方式はランダムオラクルモデルにおいて適応的安全性を満たし、追加するドキュメントに対する安全性も満たす。また、正しい検索結果 $C(w)$

Improved Searchable Symmetric Encryption for Regular Expression Search

† Shinshu University

‡ Mitsubishi Electric Corporation

に対して $\mathsf{Verify}(K, T_{w_i}, C(w)^*, tag_w) = \mathrm{accept}$ となるような偽装した検索結果 $C^*(w)$ を返すことはできない。すなわち検証は正しく行われる。

4. 暗号化索引の構成

$\mathsf{BuildIndex}(K, D, \varepsilon)$ により 2 つの暗号化索引 I_s, I_c が構成される。

I_s はサイズが $\mathcal{O}(\Sigma_w n_w)$ の整数型配列で構成され、サーバが保持する。$D(w)$ に含まれる id は ID 木と呼ばれる二分木で管理されており、各ノードには葉からの距離を表す階層番号 lev と同一階層のノードを識別するノード ID$nodeID$ が割り当てられている。w の ID 木の各ノードに対して $\mathtt{N}_{addr}$、L、R をそれぞれノード、左子ノード、右子ノードのアドレスとし、tag_w^{id} を検証用のタグ、$T_w^{SCnt} = F_{sk_2}(w||SCnt)$ とすると、$I_s[\mathtt{N}_{addr}] = (id, L, R, tag_w^{id}) \oplus H(T_w^{SCnt}||lev||nodeID)$ のようにして、I_s が構成されている。

I_c はサイズが $\mathcal{O}(m)$ でクライアントが保持する. キーワード $w \in K(D)$ に関して、これまでの検索回数 $SCnt$、ID 木の根ノードアドレス $start$、ID 木の高さ ht、w が含まれるドキュメントの数 n_w が $F_{sk_2}(w||-2)$ によって暗号化されて保存されており、$F_{sk_2}(w||-1)$ を入力することでこれらの値にアクセスすることができる。つまり、$I_c[F_{sk_2}(w||-1)] = \{SCnt, start, ht, n_w\} \oplus F_{sk_2}(w||-2)$ のようにして、I_c が構成されている。

5. 検索

検索は以下の手順で行われる。

(i) クライアントは $\mathsf{Trpdr}(sk_2, w, SCnt)$ によりトラップドア $T_w = \{T_w^{SCnt}, I_c[w].start, T_c[w].ht\}$ を作成しサーバに送信する。

(ii) サーバは $\mathsf{Search}(I_s, T_w)$ を実行し、I_s の $start$ にあるデータからドキュメント IDid_0 と検証用のタグ tag_w^0 と次のノードアドレスを取得する。その後次のノードアドレスのデータが見つからなくなるまで同様に取得し、$D(w)$ に一致する暗号化ドキュメントの集合 $C(w)$ と検証用のタグの集合 Tag_w をクライアントに送信する。

(iii) クライアントは $\mathsf{Verify}(sk_3, T_w, C(w), Tag_w, I_c)$ により、各暗号化ドキュメントについて $\mathtt{MAC}(w||c_j||I_c[w].SCnt)$ を計算し、対応する検証用のタグと比較する。すべてのタグの比較が正しく、$I_c[w].n_w$ とタグの数が等しい場合にのみ検証は成功し、それ以外の場合、検索結果は正しくない。

(iv) $\mathsf{MakeUpdateToken}(SK, D(w), w, I_c, \Phi)$ によって、クライアントは $I_c[w].SCnt$ をインクリメントし索引作成時と同様に $D(w)$ に関する ID 木を作成し、I_s の空いているアドレス $free$ と更新データ $(id, L, R, tag_w^{id}) \oplus H(T_w^{SCnt}||lev||nodeID)$ の集合 μ をサーバに送信し、$I_c[w].ht$ と $I_c[w].start$ を更新する。

(v) サーバは $\mathsf{Updata}(I_s, \mu)$ により I_s を更新する。

[命題] プロセッサ数を p とすると、任意のキーワード w の検索時間は $\mathcal{O}(\log p + \frac{n_w}{p})$ となる。

6. 追加

追加は以下の手順で行われる。

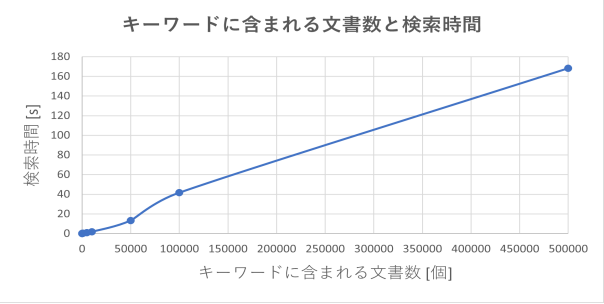
(1) クライアントは $\mathsf{MakeAddToken}(SK, d_\aleph, I_c, \Phi)$ により d_α に含まれるすべての $w \in d_\alpha$ に対し $(id_\alpha, I_c[w].start, -1, tag_w^\alpha) \oplus H(T_w^{SCnt}||I_c[w].ht + 1||0)$ を作成し、追加ドキュメント d_α を暗号化し c_α を作成する。

(2) 次にサーバから I_s の空いている要素のアドレス集合 Φ を取得し、Φ からランダムに $free$ を選択し、$free$ と (1) で作成したデータをセットで π に追加していく。そして $I_c[w].start = free$ とし、$I_c[w].ht$ をインクリメントする。最後に π をサーバに送信する。

(3) サーバは $\mathsf{Add}(I_s, \pi)$ により I_s を更新する。

7. 実験

提案方式について実験的に評価した。ドキュメントの集合として [3] を使用した。517431 個の各メールデータには高々 500 個のキーワードが含まれており、異なるキーワード数は 307830 個であった。共通鍵暗号は AES-256、疑似ランダム関数とランダムオラクルとして SHA-256 を使用した。



参考文献

[1] 三好竜司, 山本博章, 並列処理かつ動的データに向けた検索可能暗号の改良, CSS2018, pp.882-829, 2018.

[2] K. Kurosawa and Y. Ohtaki, UC-Secure Searchable Symmetric Encryption, FC 2012, LNCS 7397, pp.285-298, 2012.

[3] W. W. Cohen, The enron email dataset, http://www.cs.cmu.edu/enron/.