

DAS モデルにおける安全な類似文字列検索方式の提案

清水 将吾[†] 権 娟大^{††}

[†] 産業技術大学院大学 産業技術研究科
〒140-0011 東京都品川区東大井 1-10-40

^{††} (株)ワールドフュージョン
〒103-0013 東京都中央区日本橋人形町 2-15-15 新扇堂ビル 7F
E-mail: †shimizu-syogo@aait.ac.jp, ††yd-kwon@w-fusion.co.jp

あらまし データベース管理業務をサービスとして行う Database as a Service (DAS) と呼ばれるモデルが普及している。遺伝子配列データベースの運用管理を外部委託する場合、未公開の配列等、データベースに登録する配列自体に価値がある場合があり、このような情報を管理者から秘匿したまま問合せを行えることが望ましい。本稿では、文字列データベースを対象とし、類似文字列検索の従来手法である q -gram と生体認証等で使用されている Fuzzy Vault と呼ばれる曖昧照合法を組み合わせることで、元の文字列を管理者から秘匿したまま類似検索を処理できる方式を提案する。

キーワード DAS, 類似文字列検索, データ保護, q -gram, Fuzzy Vault

A Proposal of a Secure Search Method for Similar Strings in a DAS Model

Shogo SHIMIZU[†] and Yeondae KWON^{††}

[†] Graduate School of Industrial Technology, Advanced Institute of Industrial Technology
1-10-40 Higashiooi, Shinagawa-ku, Tokyo 140-0011

^{††} World Fusion Co., Ltd.
Sinsendo Bldg. 7F, 2-15-15 Nihonbashi-ningyocho, Chuo-ku, Tokyo 103-0013
E-mail: †shimizu-syogo@aait.ac.jp, ††yd-kwon@w-fusion.co.jp

Abstract Currently, a database-as-a-service paradigm has become popular where database administration tasks are provided as a service. When outsourcing the operation and administration of a gene database, it is desirable that a gene array be protected from database administrators while preserving the functionality of similarity search, because the gene array may be an unpublished gene that has a great value. In this paper, for string databases, we propose a method for processing similarity search with hiding an original string from database administrators. The proposed method is implemented by the combination of q -gram, a classical method for similar string search, and a fuzzy matching method, called Fuzzy Vault, which is applied to biometric authentication.

Key words DAS, similarity string search, data protection, q -gram, Fuzzy Vault

1. はじめに

技術的、経済的観点から、データベース管理業務を Database as a Service (DAS) を利用して外部に委託することがある [1]。DAS モデルにおいては、データベースに格納されている情報を通信路上の第三者だけではなく、委託先であるデータベース管理者からも秘匿できることが望ましい。

本稿では、遺伝子配列等の文字列データベースを対象とする。遺伝子解析では、核酸配列やアミノ酸配列を格納した遺伝子配

列データベースに対して、問合せ配列と類似した配列を検索する作業が頻繁に行われる。遺伝子配列データベースの場合、まだ公開されていない候補遺伝子等、配列自体に価値があることが望ましい。一方で、機能予測を行う等の目的で、類似した配列に興味をもつ組織に対してはその配列情報を共有したいという要求がある。そこで、本稿では、元の文字列データを管理者から秘匿したまま、類似検索を効率的に行える問合せ処理方式を提案する。

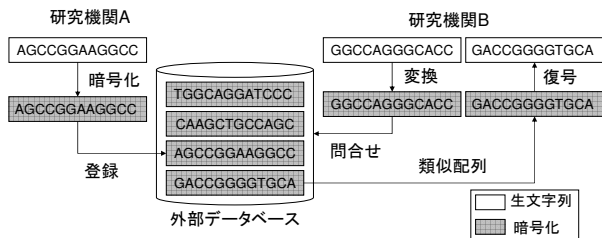


図 1 DAS モデルでの類似検索

これを実現する方式として、データベースに登録するデータを標準的な暗号化アルゴリズムで暗号化し、登録者が暗号化データとともに対応する索引の構成要素をサーバに提供する方法が考えられる。しかし、この方式は復号化のための鍵が必要になり、多数の登録者と利用者が存在するデータ共有型の環境では安全な鍵の配布や管理が難しくなる。このため、本稿では、暗号化鍵を使用せずに前述の機能を実現することを目的とする。

文字列間の類似度の定義としては、遺伝子配列の相同性検索等で使われている編集距離を採用する。編集距離に基づく文字列類似検索を効率的に処理する方法として、 q -gram [2] が知られている。 q -gram は二つの文字列間の編集距離とそれらが共通にもつ長さ q の部分文字列の数との間に成り立つ関係を利用して、解候補のフィルタリングを行う。本稿では、 q -gram を fuzzy vault [3] と呼ばれる集合間の曖昧照合法と組み合わせることで、暗号化鍵を必要とせずに、秘匿化データ上の類似検索を効率的に処理できる方式を提案する。fuzzy vault は、秘密情報のある集合を用いて施錠し、問合せとして与えられた集合が施錠時に使用した集合と十分似ている場合のみ秘密情報を開示する手法であり、生体認証等に应用されている。

本稿の構成は次の通りである。まず、2 章で準備を行う。次に、3 章で、提案方式について述べ、パラメータの選択方法や提案方式の安全性について考察する。4 章で、関連研究を紹介する。最後に、5 章で、まとめと今後の課題について述べる。

2. 準備

2.1 DAS モデル

DAS モデルにおける実体は、データ所有者（登録者）、クライアント（検索者）、データベース管理者の三者である。検索者は登録者と異なってもよい。データベース管理者はデータベース管理業務のみを委託される。本稿では、

- データベースに格納されている文字列と十分近い文字列を知っている検索者に対しては、その文字列を開示しても良い。
 - 管理者がデータベースの内容を見ることは許可されない。
- という設定のもとで、効率的な類似文字列検索処理を実現することを目的とする。但し、サーバ上での問合せ処理は正しく実装されるものと仮定する。

処理モデルの概念図を図 1 に示す。データを格納するときは、元の文字列を暗号化した状態でデータベースに登録する。問合せを行うときは、問合せ文字列を登録時と同様の方法で変換し、ハッシュ化した後にサーバに送信する。サーバはデータベース

に格納されている各文字列と問合せ文字列との照合処理をそれぞれ暗号化されたままの状態で行い、指定された類似度を満たさないことが保証されるデータを解候補から排除する。フィルタリングを通過した暗号化データをクライアント側で元の文字列に復元し、問合せ文字列との類似度を実際の定義に従って計算することで最終的な解を得る。復元処理部を耐タンパ装置上に実装できる場合には、サーバ側で復元処理を行った後に暗号化された通信路を經由してクライアントに最終結果を送信することも可能である。

2.2 q -gram フィルタリング

提案方式では、効率的な類似文字列検索を実現するための従来手法である q -gram の枠組みを利用する。以下、 q -gram の概要について述べる。

本稿では、文字列間の類似度として編集距離を採用する。編集距離は文字の挿入、削除、置換操作によって二つの文字列を同一にするために必要な編集操作の最小数として定義される。データベースを文字列の集合とする。類似文字列検索では、文字列 s と編集距離 d が与えられたとき、 s との編集距離が d 以下であるようなデータベース中のすべての文字列を解として出力する。

類似検索の処理方法として、第一段階で実際の類似度計算よりも効率の良い方法で粗い粒度のフィルタリングを行い、第二段階で類似度定義に基づく計算を行って第一段階で得られた解候補の洗練化を行う方式がある。長さ n の文字列と長さ m の文字列の編集距離の計算は動的計画法により $O(nm)$ 時間要するため、フィルタリング方式により類似文字列検索を効率良く処理するためには、第一段階でこれよりも効率的かつ効果的な手法で解になり得ない文字列を排除する必要がある。

類似文字列検索に対するフィルタリング処理の代表的な手法として、 q -gram が知られている。 q -gram とは元の文字列の長さ q の部分文字列のことである。長さ n の文字列と長さ m の文字列の編集距離が d であれば、それらは少なくとも $\max(n, m) - (d - 1)q - 1$ 個の q -gram を共通にもつことが保証されている [2]。この性質を用いて、データベース中の文字列と問合せ文字列との共通の q -gram の個数を調べることで、解になり得ない文字列を $O(n + m)$ 時間で効率的に排除できる。

2.3 Fuzzy Vault

次に、fuzzy vault の概要について述べる。fuzzy vault は誤り訂正符号を用いて集合間の曖昧照合を実現する手法であり、生体認証やパスワード復元等に应用されている。

\mathcal{F} を大きさ p の体とする。施錠時は、パラメータ k, t に対して、秘密情報 $s \in \mathcal{F}^k$ を集合 $A \in \mathcal{F}^t$ を用いて施錠し、安全性に関するあるパラメータ r に対して vault と呼ばれる $R \in \mathcal{F}^r$ を出力する。このとき、 R から s が推測できないように R を構成する。開錠時は、 R と集合 $B \in \mathcal{F}^t$ を引数とし、 B が A と十分近い場合には s 、そうでなければ空を出力する。

この開錠機能を実現するために、Reed-Solomon 符号を用いる。符号長が n で k 個の情報記号からなる (n, k) 符号を考える。各符号語を次元数が $k - 1$ 以下の \mathcal{F} 上の多項式に対応させる。 f を多項式、 $x_i \in \mathcal{F}$ (但し、すべての x_i は異なる) 、

$y_i = f(x_i)$ としたとき, 点の集合 $\{(x_i, y_i)\}_{i=1}^n$ を符号語とみなす. 復号アルゴリズムでは, これらの点集合を入力として受け取り, 大部分の点がかたがた $k-1$ 次元の一つの多項式上にあるものとみなして復号を試みる. 復号に成功すれば多項式 f を出力する. 入力と適合する正しい次元の多項式が存在しない場合や対応する符号語が壊れており多項式の計算が難しい場合は復号に失敗し, 空を出力する.

fuzzy vault は多項式復元問題と R を生成する際に追加されるチャフと呼ばれる擬似データの存在によって情報理論的に安全性が保証されている. 集合 S の大きさを $\|S\|$ と書く. $\|A\| = \|B\| = n$, $\|R\| = r$ としたとき, vault から元の多項式を復元するには, 小さい実数 $\mu > 0$ について, 少なくとも $1 - \mu$ の確率で

$$\frac{\mu}{3} p^{k-n} \left(\frac{r}{n}\right)^n$$

個の組合せ数 (多項式数) が存在することが示されている [3].

提案方式においては, 二つの文字列が一定の類似度をもつ場合, 秘匿化して登録したデータベース中のデータを元の文字列に復元する際に fuzzy vault の原理を使用する.

3. 問合せ処理

検索者が問合せ時に指定可能な編集距離の最大値を \hat{d} とする. この値は登録する文字列毎に指定できる. データベースは問合せ文字列 s と編集距離 $d (\leq \hat{d})$ が与えられたとき, s との編集距離が d 以下であるような文字列をすべて含む文字列の集合を結果として返す.

以下, 文字列 s の長さを $|s|$ と書く.

3.1 登録時

(1) 体を \mathcal{F} とする. 登録する文字列 s を任意の可逆的な方法で $k-1$ 元多項式に対応付ける. この方法は利用者間で共有する. 対応付けの方法としては, 例えば, s を $k-1$ 個の部分文字列に分解し, 各部分文字列を \mathcal{F} の元に対応付けてそれらを多項式の係数とする方法が考えられる. パラメータ k の値の選択方法については後述する. これを元に Reed-Solomon 符号により符号多項式 f を生成する.

(2) s から生成される $n = |s| - q + 1$ 個の q -gram の集合を $A = \{a_1, \dots, a_n\} (a_i \in \mathcal{F})$ に対応付ける. この対応付けは, 任意の i, j について $a_i \neq a_j$ が成り立ち, かつ a_i から元の q -gram が推測できないようにハッシュ化して行う.

(3) $X, R \leftarrow \phi$ とし, 各 $i \in [1, n]$ に対して, 以下を行う.

$$(x_i, y_i) \leftarrow (a_i, f(a_i));$$

$$X \leftarrow X \cup \{x_i\};$$

$$R \leftarrow R \cup (x_i, y_i);$$

(4) R にチャフと呼ばれる擬似データ群を追加する. 各 $i \in [n+1, r]$ に対して, 以下を行う. r は安全性に関するパラメータである.

$$x_i \in \mathcal{F} - X;$$

$$y_i \in \mathcal{F} - \{f(x_i)\};$$

$$R \leftarrow R \cup (x_i, y_i);$$

(5) R 中の要素を x 値の昇順に並び替える. R を s に対応する秘匿化情報 (vault) としてサーバに送信し, データベースに格納する.

3.2 問合せ時

データベースに格納されている各 vault R に対して, 以下の処理を行う.

(1) 問合せ文字列 t から $m = |t| - q + 1$ 個の q -gram の集合を生成し, これを登録時と同じ方法で $B = \{b_1, \dots, b_m\} (b_i \in \mathcal{F})$ に対応付ける. q -gram をハッシュ化するのは, 送信された q -gram 集合からサーバ側で問合せ文字列に復元されることを防ぐためである.

(2) B の中から任意に l 個の q -gram を間引き, B からこれらを取り除いたものを B^* とする. 間引きはサーバ側で問合せ結果の復元処理が確実に行えることを防ぐために行う.

(3) B^* をサーバに送信する.

(4) サーバ側で以下の方法により B^* と R との照合処理を行う. R の x 座標 b_i への射影を $(x_i, y_i) \xleftarrow{(b_i, \phi)} R$ と書く. 任意の y に対して, $(b_i, y) \in R$ となる対があれば, $(x_i, y_i) = (b_i, y)$ とする. そのような対がなければ, (x_i, y_i) には空が割り当てられる. $Q^* \leftarrow \phi$ とし, 各 $i \in [1, m]$ に対して, 以下を行う.

$$(x_i, y_i) \xleftarrow{(b_i, \phi)} R;$$

$$Q^* \leftarrow Q^* \cup (x_i, y_i);$$

(5) c を d, n, m, q によって定まるある整数とする. $\|Q^*\|$ が $c-l$ より小さければ, Q^* , すなわち s を解候補から排除する. $\|Q^*\|$ が $c-l$ 以上であれば, R を解候補集合に含め, 次の処理を行う. c の値の計算方法については後述する.

(6) R が解候補であれば, R をクライアントに送信する.

(7) クライアント側で (4) と同様の方法で B と R との照合処理を行う. この結果出力される集合 Q の大きさが c 以上であれば, 次の処理を行う. そうでなければ, Q を破棄する.

(8) 誤り訂正により Q を復元する. 復号化が成功すれば, 多項式 f が得られ, f に対して登録時の逆変換を適用することで元の文字列 s が得られる.

(9) 復号された各文字列に対して, t との編集距離を計算することで解候補集合の洗練化を行い, 最終的な解を得る.

登録および問合せ処理の手順を図 2 に示す.

誤り訂正符号の復号化アルゴリズムとして任意の方式が適用可能であるが, 本稿では, 実装が容易でかつ効率的な Berlekamp-Massey アルゴリズムを考える. Berlekamp-Massey では, Q 中に少なくとも $\lceil \frac{n+k}{2} \rceil$ 個の点があれば, 復号に成功する [4]. ここで, 編集距離が d である場合に二つの文字列間で共通する q -gram の数が少なくとも $\max(n, m) - (d-1)q - 1$ 個存在することを利用して, 以下の式を満たすようにパラメータ k の値を選択する.

$$\frac{n+k}{2} = n - (d-1)q - 1$$

このとき, フィルタリングの条件として使用する c の値を以下

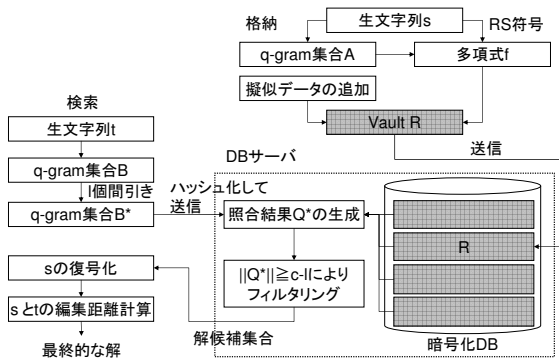


図2 登録および問合せ処理の手順

のように定める．

$$c = \max(n, m) - (d - 1)q - 1$$

実際， s と t の編集距離が $d(\leq \hat{d})$ のとき，次式が成り立つ．
 $n \geq m$ の場合，

$$\begin{aligned} \|Q\| &\geq n - (d - 1)q - 1 \\ &\geq n - (\hat{d} - 1)q - 1 \\ &= \frac{n + k}{2} \end{aligned}$$

$n < m$ の場合，

$$\begin{aligned} \|Q\| &\geq m - (d - 1)q - 1 \\ &> n - (d - 1)q - 1 \\ &\geq n - (\hat{d} - 1)q - 1 \\ &= \frac{n + k}{2} \end{aligned}$$

従って，本フィルタリング方式によって，データベース中に存在する解は必ず解候補集合に含まれ，復号化アルゴリズムによって元の文字列に復元できることが保証される．

計算時間について， A, B の要素がソート済みであれば， Q は $O(n + m)$ 時間で生成できる．従って，フィルタリング段階の計算量は q -gram と同じである．フィルタリングを通過したデータについては更に Reed-Solomon 符号の復号化処理が必要になるが，Berlekamp-Massey アルゴリズムで復号した場合，計算時間は誤り訂正可能な数 $t = \lfloor \frac{n-k}{2} \rfloor$ に対して $O(t^2)$ である [4]．このため，復号処理の計算時間はフィルタリング効果と各文字列との類似性の度合に依存する．

次に，オフライン攻撃に対する vault の安全性について述べる． R から f を推測できる可能性は，2 章で述べたように， f を攻撃者から隠す多項式の組合せ数に依存する．例えば， $r = p = 10^4, n = 22, k = 14$ とした場合， 2^{86} 個の多項式が存在する．従って， r と k ，すなわち \hat{d} の値の選択は安全性に影響を与える．

r を大きくすれば，次元が k より小さく，かつ R 中のちょうど n 個の点と一致するような多項式の数が増えることになる．攻撃者は正しい多項式 p とこれらの偽の多項式を見分けることができないため，多項式の数が増える程安全性は高まる．

\hat{d} の大きさは，利便性と安全性の間のトレードオフを決定する． \hat{d} の値はデータの登録時に決定する必要があるため，一旦登録を行った後は変更できない．このため， \hat{d} の大きさに余裕をもたせておけば，問合せにおいて指定可能な編集距離の範囲が広がり，利便性が向上する．一方， \hat{d} の値が大きくなる程 k の値は小さくなるため，多項式の数は減少し，vault の安全性は低下する．また，問合せにおいては， \hat{d} を大きくすれば一回の問合せでより多くのデータを検索できるようになる一方で，低い類似度をもつ問合せに対しても vault が開錠できるようになるため，施錠の強度は弱くなる．

B から間引く q -gram の数 l は，サーバでの問合せ処理の際に管理者が Q から元の文字列を復元できる可能性の程度を表す． l を大きくすれば，多項式復元のための情報は減るが，フィルタリングでの誤検出が増え，処理効率は低下する．極端な例では， $l = 0$ のとき，フィルタリングを通過した Q から必ず元の多項式を復元できることが保証される．一方， $l = c$ のとき，すべてのデータがサーバから返される．このトレードオフを考慮して，フィルタリング条件を調整する．

3.3 安全性に関する考察

vault から多項式に関する情報の一部を推測できる可能性について考察する．まず，秘密情報として登録する文字列中の文字の分布が一樣でない場合，多項式の係数に偏りが現れる可能性がある．この場合，文字の出現頻度や出現パターンを利用して，攻撃者は候補多項式の数減らせる可能性がある．同様に， q -gram の分布が一樣分布でないために， q -gram の統計情報から R 中のチャフの一部を推測できる可能性がある．また，多項式の生成と x 座標集合の生成に同じ情報源を用いているため，これらを組み合わせた攻撃が存在する可能性がある．従って，十分な多項式数を確保するためには，分布が一樣に近づくように係数や x 座標への符号化規則を改良する必要がある．

問題設定上，サーバ管理者であってもデータベースに格納されている文字列と十分類似した文字列を与えることができれば，問合せ経由で情報を得ることは可能である．しかし，その場合でも得たい情報から距離 \hat{d} 以下の類似した問合せを作成する必要がある．例として，糖鎖遺伝子データベースを考える [5]．糖鎖遺伝子の数は 300 程度と予測されており，その平均配列長 \bar{n} は既知のものに限れば約 1200 bp (塩基対) である．従って，編集距離を \hat{d} としたときの一遺伝子あたりの平均文字列空間の大きさは

$$\frac{4^{1200}}{300 \times \left(1 + \sum_{i=0}^{\hat{d}-j} \sum_{j=0}^{\hat{d}} (4^{\hat{d}-i-j} \binom{\hat{d}-i-j}{\bar{n}+1} C_{\hat{d}-i-j} + \bar{n} C_i + 3^j \bar{n} C_j)\right)}$$

以上であり，文字列の統計的性質に偏りがないと仮定した場合， $\hat{d} = 100$ で 1700 ビット程度の鍵の安全性に相当する．上式の分母第二項はある文字列から編集距離 \hat{d} 以下の文字列のパターン数の上限を表す．従って，管理者に前提知識がない場合，攻撃は困難である．逆に，一データあたりのドメイン空間が小さい場合には，問合せで何らかの結果が得られる可能性が高くなるため，本方式は向かない．

3.4 他のデータ構造への適用

本節では、同様の DAS 設定のもとで他のデータ構造への拡張を考える．順序付き木 [6]，順序無し木 [7]，グラフ [8] に対して，二段階方式による類似検索の効率的な処理方法が提案されている．これらはいずれも第一段階でそれぞれのデータの構造情報を要約したヒストグラム間の L_1 距離を利用して解候補のフィルタリングを行う．このときのフィルタリング条件を vault の開錠条件に関連付けることができれば，文字列と同様の方法で安全な類似検索を実現できる．以下では，順序付き木の場合について述べる．

文献 [6] の方法では，まず木を特定の変換方法によって対応する完全二分木表現へ変換する． q -level 二分岐とは二分木の高さ $q-1$ の部分木の分岐構造のことである．木 T の q -level 二分岐ベクトルとは各要素 b_i が木の i 番目の q -level 二分岐の出現回数を表すようなベクトル $(b_1, b_2, \dots, b_{|\Gamma|})$ である．ここで， $|\Gamma|$ はデータセットにおける q -level 二分岐空間の大きさである．このとき，二つの木 T と T' の編集距離が d であれば，それらの q -level 二分岐ベクトル間の L_1 距離は $(4 \times (q-1) + 1) \times d$ 以下であるという定理が成り立つ．

この定理を利用して，木 T の登録時に以下のように多項式の次元 k を選択する．

$$\frac{n+k}{2} = n - (4 \times (q-1) + 1) \times \hat{d}$$

ここで， n は T に含まれるノードの数である．多項式上の点集合 A は q -level 二分岐から生成する．問合せに使用される木を T' とし， T' に含まれるノードの数が n 以下である場合を考える．パラメータ c を

$$c = n - (4 \times (q-1) + 1) \times d$$

とし，フィルタリング条件を $\|Q\| \geq c$ とすれば， T と T' の編集距離が d であるとき，

$$\begin{aligned} \|Q\| &\geq n - (4 \times (q-1) + 1) \times d \\ &\geq n - (4 \times (q-1) + 1) \times \hat{d} \\ &= \frac{n+k}{2} \end{aligned}$$

となり，解が必ず出力結果に含まれ，元の情報に復元できることが保証される．他のデータモデルについても同様の方法で fuzzy vault と結び付けることができる．

4. 関連研究

文献 [1] では，DAS モデルにおける暗号化データへの問合せ処理方式についてまとめられている．具体的には，暗号化された関係データベースに対して比較演算や算術演算を含めた SQL 問合せを処理する方式について，暗号に基づく手法と情報ハイディングに基づく手法に分類して述べられている．しかし，対象が関係データであり，類似文字列検索については触れられていない．

文献 [9] では，DAS モデルにおいて，鍵を使用して登録データを暗号化し，値の範囲に応じてクライアントがハッシュ索引

の構成要素を暗号化データとともにサーバに送信する手法が提案されている．問合せ時は，まずサーバ側で問合せから解が含まれるバケットを決定し，このバケットに含まれるすべてのデータをクライアントに送信する．次に，クライアント側がこれらのデータを復号した後に通常の間合せ処理を行う．データベースは暗号化されており，鍵はクライアントのみが保持しているため，サーバ管理者から情報を秘匿できる．しかし，バケットの構成から部分的な機微情報が開示される危険性がある．この問題に対し，文献 [10] 等では，同様の枠組において，秘匿度を高めるためにバケット中のデータ数を均等化する改良を行っている．しかし，バケット型の索引は類似検索の高速化には適用が難しい．また，復号鍵を必要とするため，特にデータの登録を行うクライアントと検索を行うクライアントが多数存在するような環境では厳密な鍵管理が必要になる．

複数のデータ提供者とデータ検索者が存在する環境において，ある要素が与えられた集合に含まれるか否かを安全に問い合わせる方式として，暗号化 Bloom フィルタを用いた方式が提案されている [11]．この方式では，提供者は自身の鍵を使用して作成した Bloom フィルタを外部に公開し，検索者は問合せから自身の鍵を使用して Bloom フィルタを生成する．この Bloom フィルタを第三者機関がグループ暗号の原理に基づきデータ提供者の鍵を使用した形式に変換し，照合を行う．これにより，問合せ内容を該当するデータ提供者以外の第三者に知られることなく，類似データをもつ提供者を検索できる．但し，この方式は P2P 型を想定しているため，DAS モデルにおける管理者からの情報秘匿の目的では利用できない．

5. おわりに

本稿では，DAS モデルにおいて，「データベースに格納されている文字列と十分近い文字列を知っている検索者に対しては情報を開示しても良い」という設定のもとで，管理者からデータを秘匿したまま類似文字列検索を効率的に行う方式を提案した．また，文字列以外のデータモデルでも，ヒストグラムに基づくフィルタリングによって類似検索を処理する方式であれば，パラメータを適切に設定することで fuzzy vault と組み合わせで使用できる．具体的には，木構造データとラベル付きグラフの類似検索に対して，同様の方式で暗号化データ上での類似検索を処理できることを示した．

本方式は vault の解析に対しては安全であるように構成できるが，適当な問合せを作成することで容易にいくつかの元データを得ることができる可能性がある．検索者からの辞書攻撃に対しては，一定時間内の同一アカウントからの問合せ回数を制限する等の対策を取ることができる．サーバ管理者からの攻撃に対しては，新規候補遺伝子の配列や専門性の高い文章等，攻撃者がデータベースに格納されている文字列と類似した文字列を作成することが困難であると仮定できる場合には，この方式を安全に適用できる．例えば，SNP 等の疾患情報や非公開特許の情報管理にも本方式を適用できると考えられる．

今後は，安全性の評価や実データを対象としたフィルタリング効果や問合せ処理時間の評価を行う予定である．

文 献

- [1] H. Hacigumus, B. Hore, B. Iyer and S. Mehrotra: “Search on Encrypted Data”, *Secure Data Management in Decentralized Systems* (Eds. by T. Yu and S. Jajodia), Springer-Verlag (2007).
- [2] E. Ukkonen: “Approximate string matching with q-grams and maximal matches”, *Theor. Comput. Sci.*, **92**, 1, pp. 191–211 (1992).
- [3] A. Juels and M. Sudan: “A fuzzy vault scheme”, *Des. Codes Cryptography*, **38**, 2, pp. 237–257 (2006).
- [4] 今井: “符号理論”, 電子情報通信学会 (1990).
- [5] Y. Kwon, A. Togayachi and H. Narimatsu: “GGDB: A database system for glycogenes”, *The Second Symposium of Japanese Consortium for Glycobiology and Glycotechnology*, pp. 42–43 (2004).
- [6] R. Yang, P. Kalnis and A. K. H. Tung: “Similarity evaluation on tree-structured data”, *SIGMOD Conference*, pp. 754–765 (2005).
- [7] K. Kailing, H.-P. Kriegel, S. Schönauer and T. Seidl: “Efficient similarity search for hierarchical data in large databases”, *EDBT*, pp. 676–693 (2004).
- [8] A. Papadopoulos and Y. Manolopoulos: “Structure-based similarity search with graph histograms”, *DEXA Workshop*, pp. 174–178 (1999).
- [9] H. Hacigumus, B. R. Iyer, C. Li and S. Mehrotra: “Executing sql over encrypted data in the database-service-provider model”, *SIGMOD Conference* (Eds. by M. J. Franklin, B. Moon and A. Ailamaki), ACM, pp. 216–227 (2002).
- [10] 三浦, 渡辺: “管理者に対しても機密を保持できる暗号化データベースの索引構成法”, 電子情報通信学会第 18 回データ工学ワークショップ/第 5 回日本データベース学会年次大会 (DEWS2007) (2007).
- [11] S. Bellovin and W. Cheswick: “Privacy-enhanced searches using encrypted bloom filters” (2004).