DAS モデルにおける安全な類似文字列検索方式の提案

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

清水 將吾♥ 權 娟大♦

Shogo SHIMIZU Yeondae KWON

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経済的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．本論文では，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 と生体認証等に応用されている 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．

Currently, a database-as-a-service (DAS) 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 an 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from database administrators while preserving the functionality of similarity search, because the gene array may be an unpublished gene that has an economic value.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.

1. はじめに

従来より，データベースの構築・運用等の技術的な理由やデータベースの維持管理費の削減等の経済的な理由から，Database as a Service（DAS）を利用してデータベース管理業務を外部に委託することが行われている[1]．DASモデルにおいては，データベースに格納される情報が個人情報や機密情報である場合，これらの情報が通信路上の第三者だけではなく，外部委託先であるデータベース管理者からも秘匿されることが望ましい．

本論文では，文字列データベースを対象とする．文字列データベースの例として，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を考える（図1）．遺伝子解析では，遺伝子配列データベースに対して，指定した問合せ配列と類似した配列を検索する作業が日常的に行われる．ここで，遺伝子配列データベースには疾患に関わる可能性がある候補遺伝子等，配列情報自体に経済的・実用的な用途や価値が含まれることがあるため，DASモデルにおいては，これらの情報が機密情報としてサーバ管理者から秘匿されることが望ましい（図1の暗号化された外部データベース）．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．例えば，検索者が知り得た遺伝子情報が業界内で既知であるか否かを判別するために，その遺伝子情報を問合せとして類似検索を行うといった状況が考えられる（図1の研究機関Aと研究機関B）．そこで，本論文では，データの管理者からの秘匿と類似の情報をもつ利用者間での共有を同時に実現し，かつ，類似文字列検索を効率的に行える問合せ処理方式を提案する．

上記の要件を満たす方式として，データの登録者が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，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式が考えられる．しかし，この方式は復号化のための鍵が必要になり，登録者や検索者の人数に比例して鍵の数が増えるため，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運用上難しくなる．このため，本論文では，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．編集距離に基づく類似文字列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，効率的な解候補のフィルタリングを行う方式である．本論文では，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，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，秘密情報を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にのみ秘密情報を開示する手法であり，生体認証等に応用されている．提案方式では，暗号化鍵を使用せずに，検索者が知り得た情報との類似性に基づいて各情報の開示を行うか否かを決定する．

本論文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．

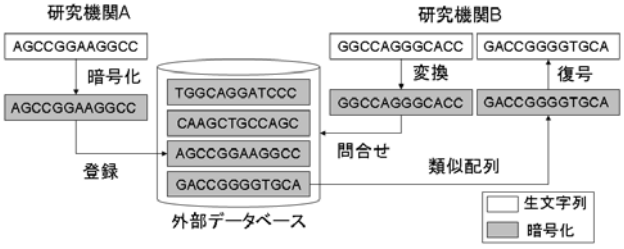

図1 DASモデルにおける文字列検索の例

Fig.1 An example of string search in a DAS model

♥ 正会員 公立大学法人首都大学東京産業技術大学院大学産業技術研究科 shimizu-syogo@aiit.ac.jp

♦ 非会員 国立遺伝学研究所生命情報・DDBJ研究センター yekwon@lab.nig.ac.jp

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いてもよい．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．本論文では，

(1) 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．
(2) 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．上記(1)の設定は，問合せの作成自体に知識を必要とするような応用を想定している．(2)の設定はDASモデルにおいて管理者を部外者と仮定しているためである．但し，管理者が適切な問合せを作成できる場合には管理者であっても正当な利用者であるとみなされる．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．

DASモデルにおける類似検索処理の概念図を図2に示す．データを格納するときは，元の文字列を秘匿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各々の秘匿化された情報を用いて行い，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秘匿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経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．

本論文では，文字列間の類似度として編集距離を採用する．

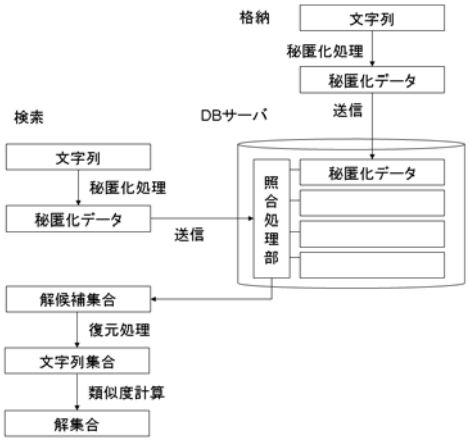

図 2 DAS モデルにおける類似検索処理

Fig. 2 Similarity search in a DAS model

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応用されている．例えば，fuzzy vaultによる指紋認証では，指紋画像の特徴点集合を秘匿化してICカードやデータベースに格納し，認証時に取得した指紋画像の特徴点集合と暗号化したままの状態で照合を行う．特徴点集合は生体や装置の条件によって揺らぎがあるため，照合時はいくつかの誤りを許容する必要がある．

fuzzy vaultの機能は次の通りである．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$s \in F^k$を集合$A \in F^t$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$R \in F^t$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$B \in F^t$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．

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に，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．q-gramとfuzzy vaultを組み合わせることにより，fuzzy vaultのみでは実現できない開錠処理のフィルタリング条件の設定についての問題を解決するとともに，q-gramのみでは実現できない情報の秘匿化の問題を解決する．

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 $d(\leqq \hat{d})$ 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

データ登録時の処理手順を以下に示す．

(1) 体を 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．この方法は復元時にも使用するため利用者間で共有する必要がある．対応付けの方法としては，例えば，s を重なりを許した k 個の部分文字列 $s_0,\ldots,s_{k-1}$ に分解し，各部分文字列を 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．
$f_g=s_{k-1}x^{k-1}+\ldots+s_1x+s_0$
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 $A=\{a_1,\ldots,a_n\}$ $(a_i\in F)$ 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．集合中に同じ q-gramが含まれる場合は，それらに元の文字列中での出現順を示す番号を付けた後にハッシュ化を行うことで異なる要素を生成できる．

(3) $X, R\leftarrow\varphi$ 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．
$x_i\in_R X$;
$y_i\in_R\{f(x_i)\}$;
$R\leftarrow R\cup(x_i,y_i)$;

(5) R 中の要素を x 値の昇順に並び替えた後に，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．R の要素の並び替えは生成順の情報を消去するために行う．

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 $B=\{b_1,\ldots,b_m\}$ $(b_i\in F)$ 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．また，登録時と同様に，同じ q-gram 文字列に対しては文字列中での出現順を追加してハッシュ化することで，これらを区別できる．

(2) B の中から任意に l（l はパラメータ）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．この間引きは，後の照合処理によって，サーバ側で問合せ結果の復元処理が確実に行えるようになることを防ぐために行う．

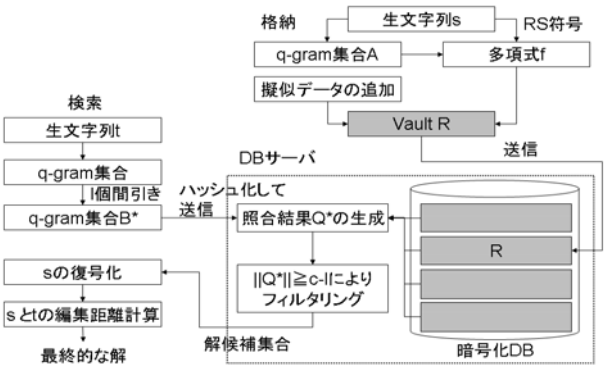

図 3　登録および問合せ処理の手順

Fig. 3　A procedure for enrollment and queries

(3) B^* を問合せ内容としてサーバに送信する．

(4) サーバ側で以下の方法により $B^*=\{b_1,\ldots,b_{m-l}\}$ と R との照合を行う．R の x 座標 b_i への射影を $(x_i,y_i)\xleftarrow{(b_i,\circ)}R$ と書く．任意の y に対して，$(b_i,y)\in R$ となる対 (b_i,y) 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．$Q^*\leftarrow\varphi$ とし，各 $i\in[1,m-l]$ に対して，以下を行う．
$(x_i,y_i)\xleftarrow{(b_i,\circ)}R$
$Q^*\leftarrow Q^*\cup(x_i,y_i)$

(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 より小さければ下記の(7)において $\|Q\|$ が c 以上になることはあり得ないため，Q^*，すなわち R に対応する文字列 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) クライアント側で上記(4)と同様の方法で B と R との照合処理を行う．照合処理は(2)で間引いた要素に対してのみ行えばよい．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-gram の定理に基づき，Q を安全に破棄できる．

(8) Reed-Solomon 符号の誤り訂正アルゴリズムにより Q を復元する．復号化が成功すれば，多項式 f_g が得られ，f_g の各係数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) 復元された文字列に対して，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 3 に示す．サーバ側で復元処理部を耐タンパ装置上に実装可能である場合は，(2)の間引き処理を行わずに q-gram から生成されたすべてのハッシュ値をサーバに送信し，サーバ側で照合結果から直接復号処理を行い，結果をクライアントに返すようにすることも可能である．この場合，間引きを行うときに比べ，誤判定の数が減るために処理効率が上がることが期待される．

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，例として，実装が容易でかつ効率的な Berlekamp-Massey 法を考える．Berlekamp-Massey 法では，Q 中に少なくとも $\left\lceil \frac{n+k}{2} \right\rceil$ 個の点があれば，復号に成功する[4]．

ここで，二つの文字列の編集距離が d である場合にそれらが共通にもつ 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．

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

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

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 $d(\leqq \hat{d})$ のとき，次式が成り立つ．

$n \geqq m$ の場合，

$$\|Q\| \geq n-(d+1)q+1$$

$$\geq n-(\hat{d}+1)q+1$$

$$= \frac{n+k}{2}$$

$n < m$ の場合，

$$\|Q\| \geq m-(d+1)q+1$$

$$> n-(d+1)q+1$$

$$\geq n-(\hat{d}+1)q+1$$

$$= \frac{n+k}{2}$$

従って，$\|Q\|$ は q-gram のフィルタリング条件と fuzzy vault の開錠条件をともに満たすため，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，かつ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，Q は $O(m)$時間で生成できるため，フィルタリング段階の計算量は q-gram と同じある．フィルタリングを通過した各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 $e = \left\lfloor \frac{n-k}{2} \right\rfloor$ に対して $O(e^2)$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度に依存する．フィルタリング効果が高ければ，復号化を行う必要がないデータ数が増えるため，処理効率は上がる．表 1 に q-gram，提案方式，暗号化鍵により暗号化されたデータを復元した後に類似度を計算する方式のそれぞれについて，問合せ処理全体の計算量を示す．ここで，n は文字列の長さ，u はデータベース中の総文字列数であり，v，$v'(\leqq u)$をフィルタリング段階を通過した文字列数とする．

表 1　計算量
Table 1　Complexity

処理	q-gram	提案方式	暗号化
フィルタリング段階	$u*O(n)$	$u*O(n)$	-
復元処理	-	$v'*O(t^2)$	$u*$復号化計算量
洗練化段階	$v*O(n^2)$	$v'*O(n^2)$	$u*O(n^2)$

3.3　パラメータの選択

まず，管理者がサーバに格納されている vault から直接元データの復元を試みようとするオフライン攻撃に対して，vault の安全性を考える．vault R から多項式 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．攻撃が成功するためには，これらのうち s に対応する一つの多項式を特定しなければならない．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．

問合せ文字列から生成した q-gram 集合 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．例えば，管理者が間引いた l 個の q-gram を総当りで推測して，B^* から B を復元して解読を試みる可能性がある．l を大きくすれば，多項式復元のための情報が減るため，安全性は高まる．しかし，フィルタリング処理における $\|Q^*\|$ の判別値 $c-l$ の値が小さくなるため，実際には類似情報を復元できる可能性が低いにも関わらず，解候補として検出される多項式点集合の個数が多くなる可能性が高くなる．この結果，類似検索処理としての処理効率が低下する．極端な例では，$l=0$ のとき，誤検出数は最も少ないが，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ため安全性は得られない．一方，$l=c$ のとき，すべてのデータがサーバから解候補として返されてしまうためフィルタリングの効果が失われる．このトレードオフを考慮して，q-gram の間引き数 l を調整する．

多項式 f の生成において，例えば，$(n+k)/2$ 個の点が判明しても s を復元できないような多項式を選択することも可能である．この場合，選択可能な多項式の総数を大きくすることができ利便性や安全性は高くなるが，s が復元可能か否かを判別するフィルタリング処理の精度が低減されるため，誤検出が多くなり，全体として処理効率が低下する可能性がある．このため，f における復元可能な点の個数を，利便性，安全性，フィルタリング処理の精度に基づいてシステムの許容範囲に応じて調節する必要がある．

誤り訂正符号の復号方法として，$(n+k)/2$ 個以下の点で多項式の復元が可能なアルゴリズムを利用することも可能である．この場合，解読するために知得する必要がある多項式の点の数が少なくなるため，vault の安全性は低下するが，r の値を大きくして多項式点の総数を増加させることにより，安全性を確保することができる．

3.4 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様でない場合，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を減らせる可能性がある．同様に，q-gram の分布が一様分布でないために，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，分布が一様に近付くように係数や x 座標への符号化規則を改良する必要がある．

fuzzy vault の無効化能力を用いて，vault から直接データを復元しようとする攻撃に対する安全性を高めることができる．fuzzy vault では，システムから生体情報を格納したテンプレートが漏洩した場合に，データベースに登録してあるテンプレートを無効化し，登録情報を生成する際に追加するチャフを変更することでほぼ無限に（同一生体情報を秘匿化した）新たなテンプレートを再登録することができる．この機能を用いて，データの登録者が定期的に vault の内容を更新してサーバに登録し直すことで管理者からの辞書攻撃に対して時間的な対策を取ることができる．この更新作業はデータの登録者のみが行えばよく，他のクライアントは自身の情報を用いて以前と同様に問い合わせることができるため，利便性を損なうことはない．

検索者が知り得た情報の正確性を情報開示の条件としている問題の設定上，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り，特にデータベースの密度がデータ空間の大きさに対して疎である場合には，十分な安全性を確保できると考えられる．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 $\bar{n}$ は既知であるものに限れば約 1200bp（塩基対）である．塩基の種類は 4 種類であるため，データ空間の大きさは 4^{1200} となる．編集距離の最大値を $\hat{d}$ とすると，一遺伝子についてその遺伝子と編集距離が $\hat{d}$ 以下であるような配列のパターンの数の最大値は，挿入・削除・置換操作の位置と塩基の種類を考慮して

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

以下である．従って，一遺伝子あたりの平均文字列空間の大きさは

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

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 として上式を計算すれば 1700 ビット程度の鍵の安全性に相当する．従って，管理者に糖鎖遺伝子配列パターン等の前提知識がない場合は攻撃は困難である．逆に，一データあたりのドメイン空間が小さい場合には，特別な前提知識なしで作成した問合せであっても何らかの出力結果が得られる可能性が高くなるため，本方式で安全性を確保することは難しい．また，問合せを用いた検索者からの辞書攻撃に対しては，一定時間内で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．

3.5 他のデータ構造への適用

本節では，同様の問題設定のもとで，本方式の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．これらの方式はいずれも，検索対象となるデータの構造情報を要約したヒストグラムを作成し，第一段階で編集距離とヒストグラム間の L_1 距離の上限との関係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 $q-1$ の任意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$(b_1,b_2,\ldots,b_{|\Gamma|})$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のサーバへの登録時に以下の式を満たすように多項式の次元 k を決定する．

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

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．以下の手順は文字列の場合と同様である．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 $\|Q\|\geqq c$ と設定すれば，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，かつ元の情報に復元できることが保証される．

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，クライアントが鍵を使用して登録データを暗号化し，値の範囲に応じた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問合せ処理を行う．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．更に，文献[10]等では，上記方式において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問題に対し，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，値の範囲に基づくハッシュ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，特にデータ登録者とデータ検索者が多数存在するような環境では厳密な鍵管理が要求される．

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，類似情報をもつデータ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，データの保有者であるサーバ管理者からの情報秘匿の目的では利用できない．

5. まとめと今後の課題

本論文では，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 と組み合わせて使用できる．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．本方式によれば，検索者毎の鍵を使用しなくても，検索情報を秘匿化できる．この結果，暗号化データベースにおける鍵管理の管理コストの問題を避けることができる．また，暗号化された登録情報や検索情報を復号化した後に照合を行う方式に比べ，類似検索を効率的に処理できる．

本方式は問合せ文字列を情報開示の条件として使用するため，問合せ文字列の作成に高度な知識を必要とするような応用に対して適用できる．他の例としては，SNP（Single Nucleotide Polymorphism）等の疾患情報，非公開特許の情報管理等が挙げられる．例えば，非公開特許の情報管理の場合，特許情報は秘匿化されてサーバに登録されるため，サーバ管理者がその内容を閲覧することはできないが，登録されている情報と類似した特許情報を作成できる検索者であればサーバに登録されている情報を検索できる．これは類似特許の検索や発明性の確認に有効である．

今後は，統計的偏りを考慮した符号化方式の開発や実データを対象としたフィルタリング効果，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. Sch¨onauer 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).

清水 將吾 Shogo SHIMIZU
公立大学法人首都大学東京産業技術大学院大学産業技術研究科助教．2001 奈良先端科学技術大学院大学情報科学研究科博士後期課程修了，工学博士．データベース理論の研究に従事．情報処理学会正会員．電子情報通信学会正会員．日本データベース学会正会員．

權 娟大 Yeondae KWON
国立遺伝学研究所生命情報・DDBJ 研究センター研究員．2000 奈良先端科学技術大学院大学情報科学研究科博士後期課程修了，工学博士．バイオインフォマティクスの研究・開発に従事．日本糖質学会正会員．日本バイオインフォマティクス学会正会員．