DBSJ Journal, Vol.8, No.1
June 2009

ブルームフィルタを用いたプライバシ保護検索における攻撃モデルとデータ撹乱法の一検討

An Attack Model and A Data Perturbation Method for Privacy Preserving Query using Bloomfilter

渡辺 知恵美♥　新井裕子♦
天笠俊之♦

Chiemi WATANABE　Yuko ARAI
Toshiyuki AMAGASA

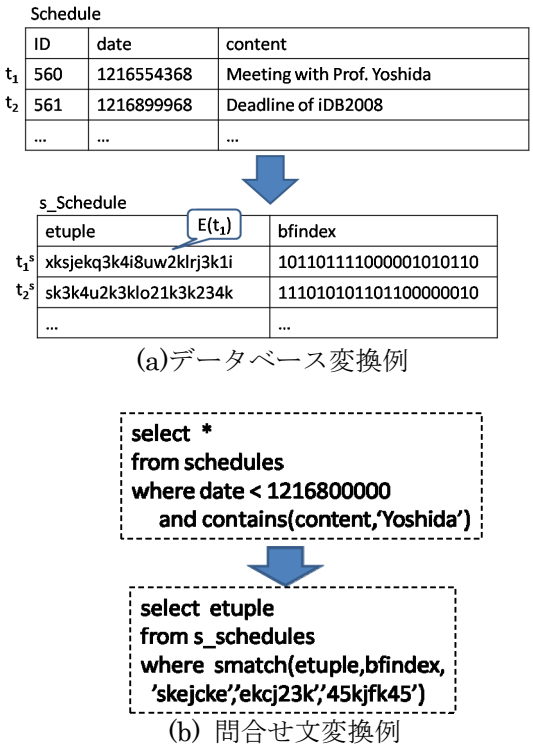
クラウドコンピューティングにおいて注目されている Database as a Service(DAS)ではデータベースを暗号化し，サーバ管理者に対してもプライバシを保護しながら問合せを行うプライバシ保護検索が注目されている．著者らはこれまで文字列検索および数値の範囲検索を扱うことのできる安全性の高い検索手法を提案してきたが，サーバでの検索処理時間がかかることが課題であった．本稿では，ブルームフィルタにノイズを混入させることにより攻撃者の推測を防ぐため，ブルームフィルタのビットパターンから攻撃者がテーブルの特徴を推測するための攻撃モデルを定義し，その攻撃モデルに対応するビットパターン撹乱の戦略を立てる．

Recently, Database-As-a-Service (DAS) has attracted considerable attention. Users require protecting sensitive data from the DAS administrators. We propose a secure query execution model for such an environment. Our approach is to represent all schemes of each tuple in a plaintext table as one Bloom filter index, and to replace queries with keyword searches of the Bloom filter index. Although this approach ensures confidence, there remains a problem about the performance of query processing.

In this paper, we define an attack model which guesses a schema of the original table by analyzing the bit patterns of the bloomfilter index, and we apply the models to artificial data and real data. From the experiences, this paper discusses a perturbation model for bloomfilter index which adds noise bits to the original index.

1. はじめに

近年，Database as a Service(DAS)がにわかに注目を集めている．DASとはクラウドコンピューティング環境においてデータベース管理を請け負うサービスである．利用者はデータベースを設置・管理するための労力をかけることなく高性能なデータベース機能を利用することができるが，管理者がデータ所有者と異なる第三者であるため，データ所有者が管理者に対して機密情報を守る必要性が生じてきた．そのための手法としてデータを暗号化した状態でデータベースに保存し，暗号化したまま問合せを施すプライバシ保護検索手法についてこれまで多くの研究がなされてきた[1-8]が，対象データのデータ型及び演算によって個別の手法が提案されており，実際に暗号化データベースに対する問合せを実現するためには，属性毎に別々の暗号化を施したり索引を作る必要があった．



(a)データベース変換例

(b) 問合せ文変換例

図 1 : 提案手法によるデータベースおよび問合せの変換例
Fig. 1. A translated table s_schedules, and query by using our proposed method.

そこで我々はブルームフィルタを用い文字列属性と数値属性を対象としたプライバシ保護検索手法を提案している[6][9][10]．本提案手法の特徴は属性毎ではなくタプル毎に索引を構成することである．また，文字列属性に対する完全一致，部分一致および数値属性に対する範囲検索をブルームフィルタによるマッチングという形で統一的に行うことにより，ANDで結ばれた複数の検索条件を一つの検索条件に変換する．図1(a)が本提案手法におけるデータベース変換の例である．IDと日付(date)と内容(content)で構成されたテーブルSchedulesが変換されて，サーバにはタプルt全体を暗号化したt(etaple)と検索用の索引t(bfindex)のみがアップロードされる．攻撃者はテーブルSchedulesがどのような属性で構成されているかを推測することができない．

図1(b)は本提案手法による問合せ変換例である．date属性に対する範囲検索条件やcontent属性に対するテキスト検索が，索引bfindexに対するマッチング関数smatchに置き換えられている．そのため攻撃者は何の属性値を用いてどのような検索条件を指定したかを読み取ることができない．

♥ 学生会員　お茶の水女子大学大学院人間文化研究科博士後期課程　chiemi@dblab.is.ocha.ac.jp
♦ 学生会員　お茶の水女子大学大学院人間文化研究科博士前期課程　{ayumi, kozue}@dblab.is.ocha.ac.jp

先行研究[6][9][10]では本提案手法の基礎となるアルゴリズムを提案し，さらにブルームフィルタのビットパターンからタプルの特徴を推測されないよう2段階でハッシュ関数を適用する手法を提案し，複数のタプルがある属性において同じ値である場合にもビットパターンが異なるよう対処していた．この手法は攻撃者に対する機密性は高いものの，サーバ側でタプル毎に数回のハッシュ関数の適用が必要となるため，検索時間がかかるという問題があった．
そこで本稿ではブルームフィルタにノイズを混入させることにより攻撃者の推測を防ぐ手法を提案する．本手法ではブルームフィルタのビットパターンから攻撃者がテーブルの特徴を推測するための攻撃モデルを定義し，その攻撃モデルに対応するノイズ付与の戦略を立てる．
ビットパターンを利用した攻撃モデルとして相関ルールマイニングによる推測攻撃を想定する．まず頻出アイテム集合の抽出によりブルームフィルタに含まれる語の発見，相関ルールマイニングによる同一属性に対する語集合の抽出，数値属性の抽出などが想定される．我々はそれらの攻撃モデルに対する防御方法として，ダミービットを戦略的に混入する等のビットパターン撹乱を行う．本稿では2節にて先行研究にて述べたのち，第3節にてビットパターンに対する攻撃モデルについて述べる．第4節ではビットパターン撹乱をする前のデータに対して攻撃モデルを実践・検証方法について，第5節ではビットパターン撹乱法について検討する．

2. 先行研究

2.1 DAS におけるプライバシ保護検索

本節では，DASにおける一般的なプライバシ保護検索手法について述べる．図2にDaaSにおける暗号化データベース検索の流れを示す．DaaSではデータベース管理者が攻撃者となりうるという点が大きな特徴となる．データベース管理者による攻撃モデルは以下の3種類があると考えられており[4]，暗号化データに対する問合せ手法は以下の攻撃に対して安全でなければならない．

(1) Direct Attack ： DB上のデータを直接盗みとる
(2) Indirect Attack ： ログから統計情報を盗み取る
(3) Memory Attack ： サーバのメモリ上のデータを盗み取る

これらの攻撃に対してデータを守るため，信頼できるサーバもしくはデータベースを利用するクライアントを通して問合せ及びデータを暗号化したうえでサーバに送信する．サーバでは暗号化したデータに対して問合せを行うため，通常の問合せと同様に正確な（不正解データが含まれない）結果を返すことは難しい．そのため，サーバで一度問合せをして仮の結果を求め，信頼できるサーバまたはクライアントにて復号化をしたのちに再度問合せを行うことによって正確な回答を得る．

これまで提案されてきたプライバシ保護検索手法は，(1)文字列属性や暗号化文書に対するキーワード検索に対する手法と(2)数値属性に対する比較演算に対する手法に分けることができる．

文字列属性に対する検索手法は[5][8]などが挙げられる．これらの手法は基本的に暗号化された文書に対してあるキーワードが含まれているかを確認することができる手法である．これはたとえば外部サーバでのメール文書のフィルタリング等では有効に働くが，データベースの文字列属性に対して適用する場合はタプル毎の認証処理が必要となるため検索時間がかかるという問題がある．

数値属性に対する比較演算に対する手法は，数値をバケットに分割しバケット番号による問合せに書き換える手法[1][3]と属性値の大小関係を維持するがその値の分散を元のデータと全く異なるものに変換する手法[2]が提案されている．

これまで提案されてきたリレーショナルデータベースに対するプライバシ保護検索では属性毎に上記のいずれかの手法でデータを暗号化する．そのため攻撃者からテーブルのスキーマ構成や問合せ条件が対象としている属性や検索のタイプを隠すことができない．一方で我々の提案手法は索引や問合せログから，スキーマ構成や問合せの傾向を読み取ることができない．

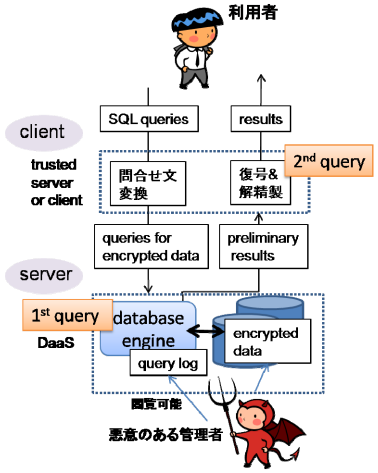


図 2　プライバシ保護検索の流れ

Fig.2　Process Flow for Privacy Preserving Query

2.2 ブルームフィルタによるプライバシ保護検索

2.2.1 基本手法

本提案手法において，タプルからブルームフィルタ索引を生成する流れを図3に示す．まずタプルtから語の集合Wt={w0, …, wn}を生成する．各語wiは属性名と属性値からなる．属性値が文字列である場合は，部分一致用に単語毎やn-gramに従って分割し属性名を合わせて語とする．数値属性から語を生成する方法については2.2.2項にて述べる．これらの語に対してHMAC（鍵付きハッシュによるメッセージ認証関数）を複数（図3ではr個としている）適用し，その値に基づいてブルームフィルタにビットを立てる．

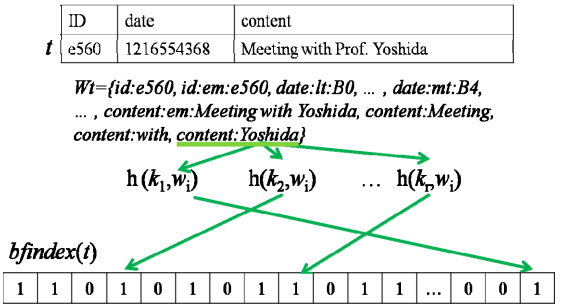


図 3　タプル t から索引を生成する流れ

Fig.3 Process flow for translating a tuple t

2.2.2 数値からの語の生成

我々は先行研究[6]にて数値から語を生成する手法を提案している．基本的には数値属性のドメインを複数のバケットに分割し，該当するバケットの名前を属性名と合わせて語とする．図 4 は 374，671，23 とそれぞれに対する語の集合を表したものである．すべてのバケット B={B1,…,Ba}に対して，バケットの上限が値 v より小さい場合「<属性名>:lt:<バケット名>」，バケットの下限が値 v より大きな場合「<属性名>:mt:<バケット名>」それ以外の場合は「<属性名>:em:<バケット名>」という語を追加する．これを用いて値の代表比較をする場合，例えば 620 より大きな値を調べたい場合は，620 が含まれるバケット(B7)の左隣(B6)に注目し，<属性名>lt:B6 という語を持つタプルを探せばよい．逆に 420 より小さな値を探す場合には，右隣のバケット(B7)に注目し，<属性名>:mt:B7 という語を持つタプルを探す．このようにして数値の比較演算を文字列のマッチングと同様に扱う．

-∞ 0 100 200 300 400 500 600 700 800 ∞

	B1	B2	B3	B4	B5	B6	B7	B8	B9	Ba
374	lt:B1	lt:B2	lt:B3	lt:B4	em:B5	mt:B6	mt:B7	mt:B8	mt:B9	mt:Ba
671	lt:B1	lt:B2	lt:B3	lt:B4	lt:B5	lt:B6	lt:B7	em:B8	mt:B9	mt:Ba
23	lt:B1	em:B2	mt:B3	mt:B4	mt:B5	mt:B6	mt:B7	mt:B8	mt:B9	mt:Ba

図 4：数値属性の変換
Fig.4. Numerical Data Translation

2.3 2 段階ハッシュによるより安全な索引

基本手法で索引を生成した場合，同じ語が含まれている複数のタプルはすべて同じ位置にビットが立っているためそこからデータの傾向などを推測される可能性がある．そこで先行研究[9][10]では基本手法のように一度ハッシュ関数を求めた後，タプルを暗号化した値（etuple）を鍵として再び HMAC を適用する．これにより，複数のタプルが同じ値を持っていた場合も 2 回目のハッシュ関数適用により異なる場所にビットが立ち，攻撃者はビットパターンから元データの特徴を推測することができない．この手法に関する問合せの手順を図 5 に示す．

まずクライアントで検索条件文から検索用の語を生成し，1 回目のハッシュ関数を適用しサーバに送る．サーバ側ではタプル毎に etuple を鍵にしてハッシュ関数を適用し，その値でブルームフィルタへのマッチングを行う．

先行研究における手法ではサーバ側でタプル毎にクライアントから渡されたハッシュ値の数だけハッシュ関数を適用しなければならず検索時間がかかるという問題がある．文献[10]における実験では Google App Engine でタプル数 1000，検索用のハッシュ値数 5，検索に対する正解率 60%の場合 0.4 秒かかることがわかった．その場合，タプル数が 20000 以上になるとタイムアウト(8 秒)せずに回答できる保証がなくなってしまう．対処法としてはサーバの分散化や複数行で一つの索引を作るなどが考えれるが，根本的な問題の解決とはならずスケーラビリティに問題があった．

3. ビットパターンを利用した攻撃モデル

先行研究における検索時間の問題は 2 段階ハッシングを採用していることにある．そこで基本手法でブルームフィルタを生成した後，攻撃者がブルームフィルタのビットパターンでテーブルの特徴を推測出来ないようにノイズビットを混入させることとした．

3.1 攻撃の目標

攻撃者にとっての目標を以下のように想定する．

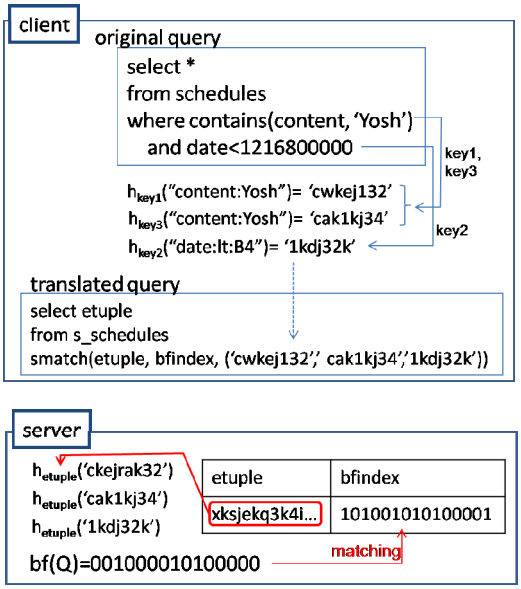


図 5：2 段階ハッシングによる問合せの手順
Fig.5. Query Processing by using Two-Phase Hashing

目標 1：同じ値（語）を持つタプルを発見する．

同じ値を持つタプルを検出することにより，そのうちあるタプルの値が漏えいした場合に連鎖的に他のタプルの値を知ることができる．

目標 2：語が属する属性の特性を発見する．

目標 1 では発見した語が何の属性によるどんな値かまでは分からない．ここでは語が属する属性の特性（データ型，分布など）を発見することを目標とする．語が属する属性の分布を発見することにより，語を持つタプルの特性が連鎖的に推測される可能性が高まる．

3.2 想定される攻撃

攻撃 1：語を構成するビットセットの推測

基本手法ではタプルを構成するそれぞれの語に対して，k 個のハッシュ関数を適用しその値に基づいてブルームフィルタにビットを立てる．そこで攻撃者はどの k 個のビットが語に対応しているかを推測すると考えられる．基本手法ではタプル内に複数の語が含まれているため単一の bfindex では推測は難しいが，テーブル内のすべての bfindex を用いて頻出アイテム集合を求めることによりビットセットを発見し，前項にのべた目標 1 が達成される恐れがある．

攻撃 2：相関ルールマイニングで語の関係を推測する

各タプルにおける bfindex を攻撃 1 で得られた語の集合に置き換えて語の相関をマイニングすることにより前項に述べた目標 2 のうち以下の特性を発見することができる．

I. 語 A,B に対して A⇒B の確信度が 0 である場合，A,B は同じタプルに同時に登場することがない．このような同時に登場しない語集合の支持度が 1 に近い場合はそれらの語は「カテゴリ型」属性の値であると推測される．ここでカテゴリ型属性とは例えば性別や出身地（都

道府県）など単一の語で表され，カテゴリ型属性において同じ値を持つタプルを発見することにより，タプル間の相関を発見しやすくなる．推測方法としては語の非共起の関係を無向グラフで表し，その中のクリークを発見する手法が考えられる．クリークを構成する語が現れるタプルの合計数がテーブルのタプル数に近いとき，あるカテゴリ型属性の値はそれらの語のいずれかを取りえると推測できる．

II. 語 A,B に対して A⇒B の確信度が 1 である場合，語 A が登場するタプルは必ず B も登場するという包含関係が生じる．その場合，語 A,B は同じ数値属性に対する語である可能性が高い. 数値属性の場合 2.2.2 項の説明からわかるように，登場するタプルに包含関係があり，例えば lt:B4 を含むタプルは必ず lt:B3 を含むという関係があるからである．そのような包含関係にある一連の語を見つけることにより，語の間の順序関係を推測することができる．推測方法としては，各語をノードとし，語 A,B が出現するタプル集合 T(A),T(B)の包含関係が T(A)⊇T(B)であるとき A から B への有向辺を張ったグラフを生成し，その中から部分連結グラフを抽出する．

III. 事前にテーブルや属性のドメインに関する情報（属性間の関係など）が攻撃者に漏れている場合，相関ルールマイニングの結果と事前情報を照らし合わせてどの語がどの属性に対応するかを推測する可能性がある．

上記のうち I,II はテーブルのドメインに特化しない共通した攻撃手法，III はドメインに特化した手法である．本稿では I,II について取り上げることとする．

4. 攻撃モデルの検証

本節では前節にて想定した攻撃モデルの検証を行う．複数のデータセット（人工データとリアルデータ）を用意し，前節に述べた攻撃 1，2，3 によってどの程度ビットパターンが推測されるかを検証する．

4.1 データセット

データセットは 2 種類の人工データと 1 種類のリアルデータを用意することとする．

TableA: 単純な構成のテーブル

属性数：3
タプル数：100,000
データ型：カテゴリ型.
全ての属性は 1 つの単語からなり，濃度は 50.
またデータの分布はすべて一様分布に従う．

TableB: 実際の構成に近いデータ

属性数：9
タプル数：100,000
データ型
・文章型（5～10 単語，単語は zipf 分布に従う）
…1 属性
・数値型（正規分布に従う．バケット分割数は 10）
…2 属性
・カテゴリ型（1 つの単語からなり，濃度は 50）
…3 属性（正規分布に従う）
＋2 属性（一様分布に従う）

TableC: リアルデータ

UCI ML リポジトリの Adult データセット[10]を使用.

属性数：14
タプル数：48842
データ型：数値型…5，カテゴリ型…9

bfindex 索引を生成するにあたり，1 語に適用するハッシュ関数の数を 5，索引のビット長を 2,048 とする．

4.2 検証 1:頻出アイテム集合検出によるビットセットの推測

Table A, B, C から生成された bfindex 索引に対して頻出アイテムセットを抽出する．適切な最小サポート値を確かめるため，ここでは頻度分布を眺める．図 6 に TableA における頻度分布を示す．

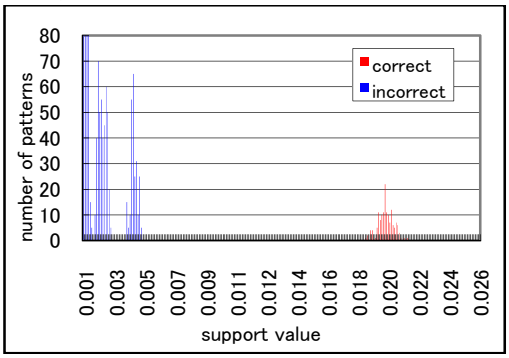


図 6：TableA（属性数 3）における頻度分布
Fig.6. Frequency Distribution of TableA

上記のうち I,II はテーブルのドメインに特化しない共通した攻撃手法，III はドメインに特化した手法である．本稿では I,II について取り上げることとする．図 6 では x 軸にサポート値を表し，そのサポート値を持つパターンの数を y 軸に示している．暗号前のデータベースから正解ビットパターンを抽出しその分布を correct としてプロットしている．また暗号化したデータベースの bfindex からビットが 1 となる位置をアイテムとしてリストアップし，1 タプルを 1 トランザクションとして 100000 トランザクションから抽出されるアイテム集合の頻度分布を調査した．暗号化の際 1 つの値に対して 5 つのハッシュ関数を適用することを想定し，かつ攻撃者もハッシュ関数の数が事前に判明していることを想定してアイテム数を 5 に固定して抽出を行った．図 6 の incorrect はそのようにして抽出された中から correct に含まれていないパターン，攻撃者にとってはノイズとなるパターンである．図 6 を見ると，ほとんどのノイズパターンはサポート値が少なく正解パターンと分離してしまっていることがわかる．つまり最小サポート値 0.019 で頻出アイテム集合を抽出すると正解パターンのみ抽出できてしまうことがわかった. TableA では属性数が少なくすべて濃度も同じ一様分布で構成されているため TableA’1, TableA’2 として以下の人工データでも検証を行った．

TableA’1：正規分布による属性数 3 のテーブル
TableA’2：属性数 40 のテーブル（一様分布）

また数値属性が含まれる TableB の人工データでも検証をいった．それぞれの頻度分布を図 7，図 8，図 9 に示す．図 7 をみると，正規分布だと頻出アイテム集合を取ることによって正解パターンは抽出されるものの，半数はノイズパターンにまぎれるため再現率は半減することがわかる．また属性数

の多い TableA’2 はノイズパターンが増えて正解パターンの分布が混合しており，容易に分離することが難しくなってくる．2 つの数値属性を含む TableB では大量のノイズパターンが正解パターン数を上回り抽出がかなり難しい状態になっている．

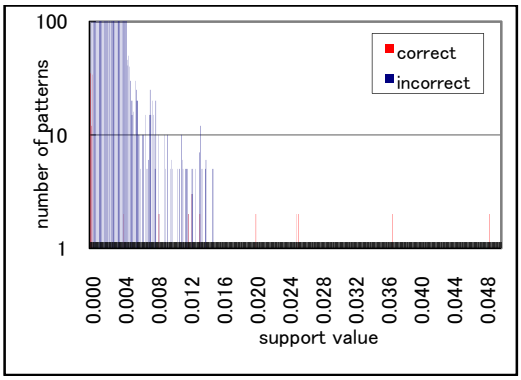


図 7：TableA’1 の頻度分布
Fig.7. Frequency Distribution of TableA’1

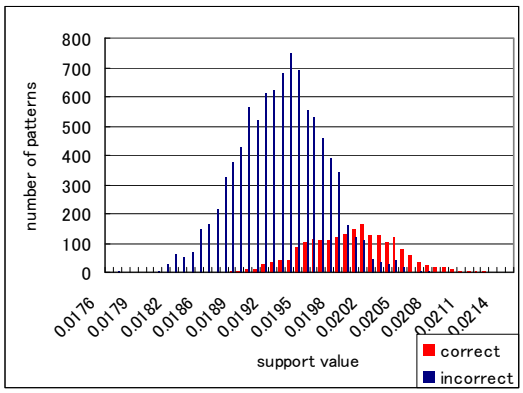


図 8：TableA’2 の頻度分布
Fig.8. Frequency Distribution of TableA’2

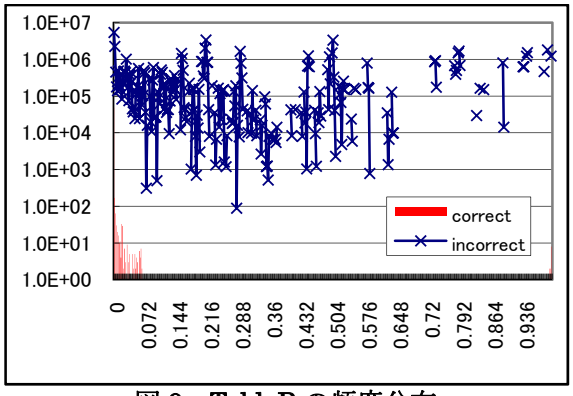


図 9：TableB の頻度分布
Fig.9. Frequency Distribution of TableB

これらの結果から以下のように考察する．

・一様分布の場合は属性数が少ないとノイズパターン数の頻度が少なく容易に分離できてしまうが，属性数が増えるにしたがってノイズパターンの数が正解パターンを上回るようになり，正解パターンを隠すことができる．
・正規分布（など一様分布以外）の場合は正解でも頻度の少ないものが多く，頻度が高いごく一部の正解パターン以外のほとんどはノイズパターンで隠蔽することができる．
・数値属性が含まれると多くのタプルに複数の語が含まれることが多くなり，複数の語からなるビット集合の組合せが爆発的に増えるためノイズパターンが圧倒的に多くなる．そのため数値属性が含まれたテーブルではノイズパターンが正解パターンを大きく上回り頻出アイテム集合の抽出では正解パターンを見つけることができなくなる．

4.3 検証 2:同一の属性に対する語の推測

前節に述べた攻撃 2 における I の成功率を検証する．本来の攻撃では抽出したビットセットを語に置き換えた上で同時に出現しない語の集合を求める．ただしここでは実験 1 の結果と実験 2 の結果を分離するため，暗号前のデータベースから求めた語を用いて実験した．

TableB では非共起の関係を求めた結果ノード数 673,エッジ数 155919 のグラフが生成された．ノード数 673 の完全グラフの辺数の約 3 割が使われており，密なグラフである．その中から抽出されたクリークはノード数が 477 であり，予測したような同一の属性の値だけではなく，多くの値を混合していた．属性数が多い場合はこのような手法で語を属性ごとに分類するのは難しいと判断される．TableC の場合は 1500 タプルを読み込んでグラフを作成した時点でクリークを検出したところ 929 のノードからなるクリークが検出された．これも全く異なる属性の値から構成されており，これを利用して属性の分類を行うことは難しいと判断できる．

4.3 検証 3:数値属性に対する語の推測

前節に述べた攻撃 2 における II の成功率を検証する．本実験では数値属性が含まれている TableB, TableC でのみ実験を行う．確信度が 1 である語のペアを発見し，その相関関係をグラフで表し，部分連結グラフを抽出したときにそこから数値属性が抽出できるか検証する．TableB から抽出した部分連結グラフを図 10 に示す．TableB に含まれる数値属性に対応する語の関係が抽出できていることがわかる．また，TableC（リアルデータ）による検証の結果 5 属性中の 3 属性が抽出された．

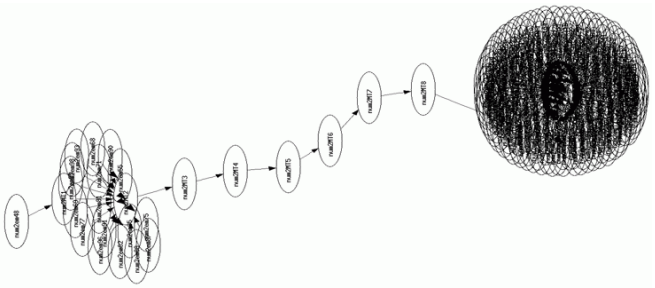


図 10　部分連結グラフで数値型の属性値を検出
Fig.10. The Connected Graph Which can Detect Relations of the Words for Numerical Attribute Values

4.4 考察

前節までの検証により以下の考察を行った．

(1) ビットパターンからの語の抽出はカテゴリ型のみから構成され属性数が少ない場合には頻出アイテム集合を使って抽出が可能である．属性数が多い場合や数値属性を含

む場合はノイズパターンが多くなり抽出が難しくなる．特に数値属性は複数のパターンが多くのタプルに同時に含まれるため非常に多くのノイズパターンが抽出されるために正解パターンの推測が非常に難しくなる．

(2) 抽出された各語が何の属性に対するものかを推測する際，攻撃 2 の I で述べた非共起関係を用いてカテゴリ属性を抽出することは容易ではないが，数値属性の抽出は非常に精度高く抽出することができる．

(3) 検証ではビットパターンからの語の抽出と，語からの属性ドメイン情報の抽出を別々に検証したが，実際はビットパターンから語を抽出できないとその次に進むことは難しい．攻撃 2 では数値属性の関係が比較的容易に抽出されてしまったが，その一方で攻撃 1 では数値属性によりノイズパターンが多く含まれるため語の抽出が難しいため攻撃 2 まで進めることが難しいと推察される．

(4) しかしながら数値属性により圧倒的にノイズパターンが増えることがわかると，今度はほとんどのタプルに登場するビットを分離して頻出アイテム集合を求める手法が考えられる．

以上のことから攻撃に対する防御法として以下の手法を提案する．

手法 1：ダミー属性の追加

数値属性がテーブルに含まれることにより頻出アイテム集合抽出によって語を推測することが難しくなる．そこで数値属性が含まれないテーブルに対して数値属性をダミーとして混入することによって語の推測を防御する．しかしながら数値属性のみを分離される可能性もあるため，文字列型・カテゴリ型のノイズ用属性も混在させることを考える．

手法 2：複数タプルに対して 1 つの bfindex を生成

攻撃モデルはタプル毎の bfindex のパターンをマイニングすることによりテーブルの特徴を推測してきた．そこで複数タプルをまとめて暗号化し，それに対して一つの bfindex 索引を作ることによりビットパターンを撹乱させる．この手法は適用が容易である上にある一定の効果が見込めるが，複数のタプルのうち一つでも問合せの条件に該当するものがあった場合全てのデータをクライアントに送らなければいけないため，いわゆる擬陽性が高くなる恐れがある．

手法 3：数値属性の語の推測を防ぐダミー語の混入

4.3 節にて検証した結果により，語の集合から数値属性を抽出するのは容易にできてしまうことがわかった．そこで手法 1 で混入させるダミー語を決定する際にタプルの包含関係の抽出を妨害するための混入戦略を立てる．

4. まとめと今後の課題

DAS モデルにおけるプライバシ保護検索として提案しているブルームフィルタによる検索手法について，先行研究で用いている 2 段階ハッシュ関数適用を使わずにビットパターン撹乱手法を用いる方法を検討することを目指し，その手始めとしてビットパターンによる攻撃モデルの方針を立て，その検証を行った．その検証の結果からダミー属性を混入させて推測を撹乱させる防御法の戦略を立てることができた．

今後はビットパターン撹乱法を提案し，その検証を行う必要がある．その際ビットパターンを混入させることによって擬陽性が生じることも考慮しなければならない．本研究ではサーバ側の問合せで擬陽性が生じることはすなわちサーバからクライアントへの結果送信，クライアントでの復号化，クライアントでの 2 回目の問合せのコストに影響が出ることになる．これらのコストとサーバ側での問合せのコストを総合的に考えた上で安全かつ高速なプライバシ保護検索を実現するための適切な撹乱法を確立していきたい．

[文献]

[1] Hacigumus H., Iyer B. R., Li C. and Mehrotra S.: Executing sql over encrypted data in the database-service-provider model, Proceedings of the 2002 SIGMOD International Conference, pp. 216–227 (2002).

[2] Agrawal R., Kiernan J., Srikant R. and Xu Y.: Order preserving encryption for numerical data, Proceedings of the 2004 SIGMOD International Conference, pp. 563 – 574 (2004).

[3] Hore B, Mehrotra S. and Tsudik G.: A privacy-preserving index for range queries, Proceedings of the 30th International Conference on Very Large Data Bases, pp.720—731 (2004).

[4] Ting Yu and Shushil Jajodia: Secure Data Management in Decentralized Systems, Springer-Verlag New York Inc, p.462 (2006).

[5] Boneh D, Crescenzo G.D., Ostrovsky R. and Persiano G, Public Key Encryption with Keyword Search, Proceedings of EUROCRYPT '04, vol.3027 LNCS, pp. 506—522 (2004)

[6] 新井裕子，渡辺知恵美: データベースアウトソーシングにおけるプライバシ保護に考慮した範囲検索法，電子情報通信学会 第 19 回データ工学ワークショップ，C1-1 (2008).

[7] Tingjian Ge, Stanley B. Zdonik: Answering Aggregation Queries in a Secure System Model. , Proceedings of VLDB 2007, pp.519-530 (2007).

[8] Bellovin S. and Cheswick W. : Privacy-enhanced searches using encrypted bloom filters" (2004).

[9] 渡辺知恵美，新井裕子: DAS におけるスキーマ情報と複合的な検索条件を隠ぺいしたプライバシ保護検索手法，情報処理学会研究会報告，No.2008-DBS-146，Vol.2008, No.88, pp.163−168 (2008).

[10] Watanabe C. and Arai Y.: Privacy-Preserving Queries for a DAS model using Two-Phase Encrypted Bloomfilter, Proc. of International Conference on Database Systems for Advanced Applications (2009) (to appear)

渡辺 知恵美 Chiemi WATANABE

お茶の水女子大学大学院人間文化創成科学研究科講師. 2003 お茶の水女子大学博士後期課程修了．プライバシ保護検索，P2P 環境における検索に興味を持つ．情報処理学会会員．日本データベース学会会員．

新井裕子 Yuko ARAI

お茶の水女子大学大学院人間文化創成科学研究科博士前期課程在学中．2008 年お茶の水女子大学理学部情報科学科卒業．プライバシ保護検索における研究・開発に従事．日本データベース学会学生会員．

天笠俊之 Toshiyuki AMAGASA

筑波大学大学院システム情報工学研究科，計算科学研究センター講師．データベースシステムの研究に従事．情報処理学会，電子情報通信学会，日本データベース学会各会員