DBSJ Journal, Vol.12, No.1
June 2013

暗号化データベースにおける多属性索引に対する統計攻撃モデルと攪乱戦略

A Statistical Attack Model and Perturbation Strategies for Multi-Attribute Secure Indices for Encrypted DBMS

金子 静花♥　渡辺 知恵美♥
柿澤 美穂♥　天笠 俊之♦

Shizuka KANEKO　Chiemi WATANABE
Miho KAKIZAWA　Toshiyuki AMAGASA

本研究では，ブルームフィルタを用いて作成した索引にノイズを付与することによって攻撃者を攪乱させる多属性索引を提案する．攻撃者が持つ統計情報とカテゴリ属性の特性を利用した攻撃モデルと，その攻撃に対するプライバシ指標としてビットパターン露呈率を定義する．そして，ビットパターン露呈率を下げるため 4 つのノイズ付加戦略を定義し，各属性の値の分布に基づき適切な戦略を適用することでビットパターンの候補を組み合わせ爆発的に増加させ安全性を担保できる．また，これらの戦略はノイズを加えることで実現しているため，性能の劣化なく安全な索引生成に成功していることを実験によって示した．

In this paper, we propose a statistical attack model against secure indices, and we propose perturbation strategies for multi-attribute index scheme. We first introduce an attack model using properties for categorical attributes of tables when the attacker has information about schema of the table and their statistics. We also provide four perturbation strategies for multi-attribute index to reduce the revelation rate of original attribute values. Our proposed strategies are all simple and easy to apply. In addition, they have little degradation effect to the query performance. In our experiments, we demonstrate that the perturbed multi-attribute index ensures the good resistibility against statistical attacks, and it processes SELECT operation ten times faster than the other multi-attribute index schemes.

1. はじめに

現在，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の管理運用を請け負う Database as a Service (DBaaS) が注目を集めている． DBaaSを用いることで，マシン資源の総保有コスト (TCO) を削減できる．しかしながら，DBaaS 管理者は第三者であるため，管理者に情報を悪用されるリスクも考えられる．様々なセキュリティインシデントが報告されており，DBaaS は完全なデータ機密性を保証しているとはいえない．

このようなリスクに対応するため，暗号化データを格納し，暗号化したまま問合せを施す暗号化データベースシステム (EDBMS) [1]が多く研究されてきた．EDBMSでは，元の値を暗号化値に紐づく検索用の安全な検索用索引を用いる．既存手法は列(属性)毎に暗号化した単属性索引を検索用索引として使用している．その場合，同一な元の値からは同一の単属性索引が生成されることとなるため，検索用索引は元リレーションの統計情報に関する大きな手がかりを含んでしまう．

我々は先行研究にて，ブルームフィルタを用いた DBaaS 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[2][3][4]を提案した．この手法では，タプル内の複数の属性を1つの多属性索引として生成することで安全性を担保する．しかし素の多属性索引に対しても，多属性索引の値から元の値推測の可能性が否定できない．そこで我々は二つのキーを用いたハッシュ関数を適用し多属性索引の安全性を向上させた ShuffledBF [2][3] や，多属性索引と ShuffledBF との組合せで ShuffledBF の処理速度を改善させた Semi-ShuffledBF [4]を提案したが，これらは大規模なデータに置いては処理速度が遅く実用的でない．

そこで我々は，ノイズを加えることで性能を損なわずに安全性を保障できるPerturbedBFを提案する．安全性を保障するため，まず敵の攻撃モデルを定義した．このモデルにおいて攻撃者は元テーブルの属性の統計情報を保持しており，検索用索引の特徴と攻撃者の持つ統計情報とを照合して攻撃を行う．この時攻撃者が統計的な分析を行うことで元の値のビットパターンが推測される確率をビットパターン露呈率とし，露呈率を下げるための4種類のノイズ混入戦略を定義した．

2. 事前準備

2.1 暗号化データベース管理システム

一般的な EDBMS のモデルを図 1 に示す．Alice はデータ所有者，Bob はデータベース利用者であり，Alice から閲覧権限を付与されている．Malone は DBaaS におけるデータベース管理者であり，攻撃者とする．データ所有者(Alice)はデータを，クライアント側にある EDBMS で暗号化しサーバに保存する．この時，暗号化したデータとその検索用索引を生成し，双方をサーバに送っている．検索用索引は元の値を推測できないように作られる．また，データのスキーマ情報やどの属性にたいしてどの検索用索引を付与したかというメタデータも暗号化を施された形 E(META)に変換し，サーバに格納される．もう一方の利用者 (BOB) は事前に Alice から暗号化・復号化のための key を受け取り問合せを行う．まずサーバからメタデータ E(META)を取得し復号化する．それを元に EDBMS で Bob が発行した問合せ Q を検索用索引で検索できる形 Q' に変換してサーバへ発行する．サーバでは検索用索引と Q' を用いて問合せが行われ，暗号化されたデータが EDBMS に返される．検索用索引を用いての問合せは一般に false-positive を含んでいるため，EDBMS はこれを復号化し Q で再び問合せを行った結果を Bob へ返す．

♥ 学生会員　お茶の水女子大学人間文化創成科学研究科
kaneko.shizuka@is.ocha.ac.jp，現日本オラクル株式会社
♥ 学生会員　お茶の水女子大学 理学部 情報科学科
kakizawa.miho@is.ocha.ac.jp
♥ 正会員　お茶の水女子大学　理学部　情報科学科
chiemi@acm.org，現筑波大学　システム情報系
♦ 正会員　筑波大学　システム情報系，
amagasa@cs.tsukuba.ac.jp

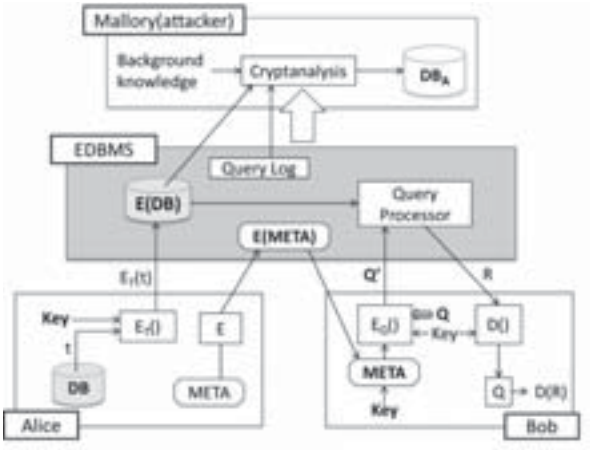

図 1　EDBMS の流れ
Fig.1 EDBMS Framework

攻撃者である Malone は暗号化されたデータベース E(DB) と暗号化されたクエリ Q’ のみを閲覧することができ，データベース管理者ゆえに知り得た外部情報も保有していると考えられる．Malone はこれらの情報を組み合わせて元データを推測しようと試みる．我々は，Malone が元データを推測することのできないような検索用索引を生成するべきである．次項にて，検索用索引の関連研究を紹介する．

2.2 関連研究

Hacigumus らによるEDBMS フレームワークの提案を先駆けとし，現在も数多くの関連研究が行われている．Hacigumus らの検索用索引生成方法では，元の値が同じであると検索用索引 の値も一致するため，検索用索引の統計から元の値が推測される可能性があった．これに対してHore らは統計による値の推測を困難にするバケット分割法を提案している[6]．Agrawal ら[1] は，数値属性の順序関係を保存できる検索用索引の生成方法(OPE) を提案している．変換した値の分散が元の値の分散と異なるように変換することによって元の値の推測を防いでおり，属性の比較演算や結合などが可能である．ただ，順序関係が保存されているため一部のデータが漏洩した場合に他の値がなし崩しに判明するという問題があり，Lee ら[5]， Hasan ら[6] などにより改良手法が対案されている．またMykletun ら[7]，Ge ら[8] による準同型性を持つ暗号化手法を利用した集約演算を可能にする検索用索引生成手法やk-近傍検索に対応した検索用索引生成手法なども提案されている．2011 年には，これらの検索用索引を組み合わせてCryptDB というEDBMS のオープンソースパッケージがPopa[11] ら (MIT) により公開された．

検索用索引は，EDBMS の問合せで用いられる検索用の安全な索引であり，単属性索引と多造成索引に分けることができる．

単属性索引はEDBMSで用いられている一般的な検索用索引生成方法で，属性ごとに検索用索引が生成される検索用索引である．属性のデータ型や属性に対して行われる演算によって適用されるアルゴリズムが変化する．CryptDBをはじめとする，2.2節で述べた研究のほとんどは単属性索引を用いて行われている．単属性索引の例を図2に示す．Id, name, age, genderで構成された元テーブルに対し，各属性に対する単属性索引としてS_name, S_age, S_gender が格納される．

図 2　単属性索引の生成
Fig.2　Single-Attribute Indices for the Original Table

単属性索引は元の値とその単属性索引との間には1対1の対応関係があるため，単属性索引の値の傾向が元リレーションの統計情報に関する大きな手がかりを含んでしまう．

多属性索引は，複数の属性の索引情報をまとめて1つの検索用索引を生成する．多属性索引の例を図3に示す．Id, name, age, genderで構成された元テーブルに対し，各属性に対する多属性索引としたmattr が各タプルに対して1つ格納される．複数の属性を1つの多属性索引としてまとめることで検索用索引の値に対する統計情報の漏えいを防ぐ．暗号化クエリQ’ の検索条件文においても，条件に使われる属性を隠蔽することができる． Bonehらの提案したPAKS[11]は暗号化された文章そのものから，あるキーワードが含まれているかどうかを検索できる公開鍵暗号を用いている．Songら[12]も同様に文章からのキーワード検索が可能な多属性索引を提案している．我々もブルームフィルタを用いて多属性索引を生成する方法を提案してきた[2][3][4]．

図 3　多属性索引の生成
Fig.3　Multi-Attribute Indices for the Original Table

3. 先行研究:Semi-ShuffledBF

我々は，多属性索引の安全性を担保するため，様々な研究を行ってきた．ShuffledBF[3]はブルームフィルタを適用して生成された多属性索引をより安全にするため，更にもう一段階ブルームフィルタによるハッシュ関数を適用（シャッフル）して生成された安全な多属性索引である．ShuffledBF の生成方法を図 4 に示す．元テーブルの属性と語の集合”ID:330”,” 名前:Alice”, “病気: fracture”に対し，各々複数のハッシュ関数を適用したのち，タプル全体を暗号化した値etupleをキーにハッシュ関数を更に適用(シャッフル)した結果を多属性索引として生成する．このように，タプル毎に生成される etuple をキーとして 2 段階目のハッシュ関数を適用しているので，安全性は完全に保たれている．

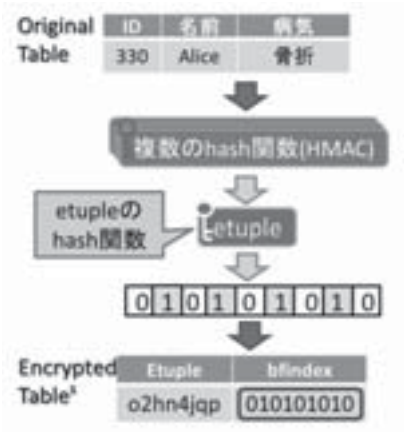

図 4　ShuffledBF の生成方法
Fig.4　Process Flow for Generating ShuffledBF

ShuffledBF の問合せ方法を図 5 に示す. 索引生成時と同様に，検索したい元データの属性と語の集合”name:Alice”に対し，1 段階目の複数のハッシュ関数を適用した結果 QMAI’を検索条件としてサーバへ送信する．サーバ側では，受け取った QMAI’にタプル全体を暗号化したものである etuple をキーにしたハッシュ関数をタプル毎に適用して QMAI”とし，多属性索引と一致するかどうかを調べる．

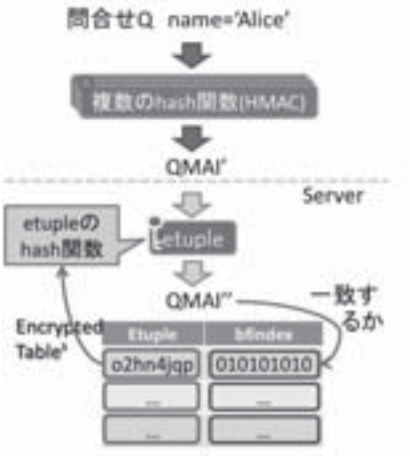

図 5　ShuffledBF の問合せ方法
Fig.5 A Query Process by using ShuffledBF

サーバ側でタプル毎にハッシュ関数を適用するため検索時間が遅いという問題がある．そこで，我々は ShuffledBF と通常ブルームフィルタとを組み合わせることで性能を改善した Semi-ShuffledBF を提案した[4]．Semi-ShuffledBF では通常の多属性索引での問合せに該当したタプルに対してのみ etuple での 2 段階目のハッシュ関数適用（シャッフル）を行う．ただしこの手法の場合問合せに該当するタプルが多いと却って負荷が大きくなる問題があった(図 11 参考)．

4. PerturbedBF

我々は，多属性索引をシャッフルせずノイズを加えることで，性能を損なわず安全性を保障する手法を提案する．敵とその攻撃モデルを定め，攻撃モデルに対する索引のビットパターン露呈率を定義し，露呈率を低下させるための攪乱戦略を複数提案する．以下の例を用いて検索用索引に対してどのように攻撃が行われるかを示す．

name	gender	blood_type	country
Alice	female	Null	Japan
Bob	male	O	Japan
Carol	female	AB	China
Dave	male	A	Japan
Eve	female	A	China
Franc	male	O	Korea
Giorgia	male	B	Japan
Hillary	female	B	Japan
Isaac	male	A	Japan

図 6　利用者が所有するテーブル例
Fig.6 An Example of Plaintext Table

4.1 攻撃モデル

攻撃者は EDBMS から得られるデータの他に，データベースに関する背景知識として，データベースのスキーマ情報と，その一部の統計情報を得ていることを想定する．たとえば図 6 のテーブルが暗号化された状態でサーバに保存されているが，そのテーブルに対して攻撃者はあらかじめ「このテーブルには性別に関する属性が含まれ，その男女比は 4:5 である」という情報を外部から取得していたとする．攻撃者はその統計情報と多属性索引に含まれるビットパターンの出現頻度を照合し，属性値に対するビットパターンを推測することができる．ビットパターンとはブルームフィルタ中で 1 の立っている位置の部分集合を表す．たとえば図 7 の左テーブルの属性 MAI が図 6 のテーブルに対する多属性索引だとする．青色で示されるビットパターン(つまり 1,2 番目に 1 が立つ)と赤色で示されるビットパターン（3,4 番目に 1 が立つ）に着目したとき，その出現頻度はちょうど 4:5 であり，それぞれ「性別:男」「性別:女」という属性値から導出されたビットパターンであると推測することができる．

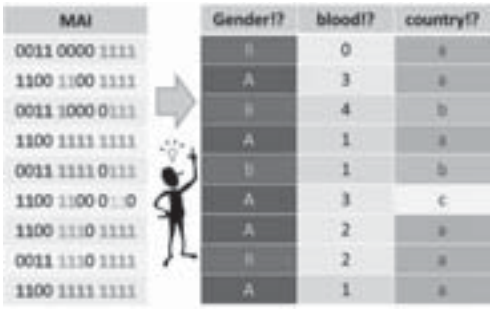

図 7　索引に含まれるビットパターンと推測される属性値[1]
Fig.7 Multi-Attribute Index and Attribute Values Which may be Inferred by an Attacker

特にカテゴリ属性（性別や出身地など，タプルと属性値には 1 つの値のみ含まれる属性）を想定したとき，多属性索引中のカテゴリ属性のビットパターン集合(BS)は以下の 2 つの性質を持つ．

[1]図 6 では簡単のため単純なビット配列を用いているが，実際のビット配列はより長く複雑なものとなっている．

(a) 排他性
各索引値は，ビットパターン集合においてたった1つの特徴を持つ
(b) 相補性
全索引値は，ビットパターン集合において少なくとも1つの特徴を持つ

この性質と統計情報を組み合わせることによって属性値の候補は大きく絞り込まれる恐れがある．

これらの考えに基づいた多属性索引に対する攻撃モデルのアルゴリズムを以下にAlorithm1に示す．すべてのビットパターンを探し出し（2行目），属性値と同じ頻度を持つビットパターンを見つけ出し（5,6行目），値のビットパターンの候補とする．それを用いて12～23行目でビットパターンの組を増やしながら排他性をチェックする．最後に24～28行目で相補性をチェックし，最終的に候補とみなす．

```
Algorithm 1: StatisticalAttack_MultiAttributeIndex (attribute)
1.  Input : M(Γ), H^A(attribute)={h^A(v_1),...,h^A(v_n)}
2.    Output : F
3.      Sort(H^A(attribute), descending)
4.      F = FindAllFeatures(M(Γ))
5.      for each h^A(v_i) in H^A(attribute) do
6.         F(vi) = SelectFeatures_withSimilarFrequency(F)
7.      end do

8.      for each f in F(v_0) do
9.         FS = {f}
10.        put FS in F
11.     end do

12.     for i = 1 to n do
13.        for each FS in F do
14.           for each f in F(vi) do
15.              if Satisfy_Exclusiveness(FS,f) then
16.                 put f in FS
17.              else
18.                 remove FS in F
19.              end
20.           end do
21.        end do
22.        if | F | > 100,000 then exit end
23.     end do

24.     for each FS in F do
25.        if Satisfy_Complementary(FS) == false then
26.           remove FS in F
27.        end
28.     end do
```

4.2 ビットパターン暴露率

前項で定義した多属性索引への攻撃モデルに対するビットパターン暴露率を定義する．

『索引に各属性に対するビットパターンの組が k 個以上含まれる時，攻撃者は統計情報を用いてこれらの特徴集合候補をさらに絞り込むことはできないため，ビットパターンが露呈する確率は1/k以下となる』

図8の例では，敵がこのテーブルにgenderという2値のみを含む属性が含まれていることを知っているとする．敵はこの情報を用いて多属性索引からビットパターン集合 を探しだし，どのタプルがmale かfemaleかを特定したい．しかしながら，この多属性索引はk=3 に設定されているため，この条件に該当するビットパターン集合は3 候補現れ，敵はこの3候補のうちどのビットパターン集合が本物か区別することはできない．

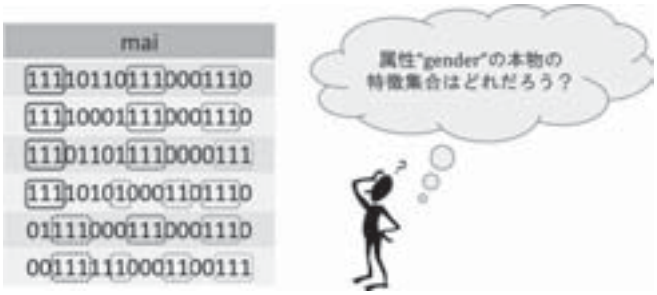

図8 属性値候補を絞り込めない多属性索引の例
Fig.8 A Multi-Attribute Index from Which Attackers cannot Infer the Original Values

攻撃者がカテゴリ属性の特性と頻度分布を利用したときのビットパターン露呈率を以下に定義した．

・単属性索引に対するビットパターン露呈率
(a) 値暴露率

$$rev_{value}(attribute) = \sum_{v \in attribute} \frac{1}{|Cand_v|}$$

ここで，$|\mathrm{Cand}_v|$は属性値vにおける索引の候補数を表す．
(b) レコード暴露率

$$rev_{record}(attribute) = \sum_{v \in attribute} \frac{1}{|Cand_v|} \times \frac{|v|}{|R|}$$

ここで，$|R|$はリレーションRにおけるレコードの数を表し，$|v|$は属性値vであるレコードの数を表す．

・多属性索引に対するビットパターン露呈率

$$rev_{value}(attribute) = \max_{1 \le i \le n}\left(\frac{1}{|BS(\Delta_i)|} \times \frac{i}{n}\right)$$

$$rev_{record}(attribute) = \max_{1 \le i \le n}\left(\frac{1}{|BS(\Delta_i)|} \times \frac{\sum_{1 \le i \le n}|v_i|}{|M(\Gamma)|}\right),$$

ここで，$\Delta_i = \{v_1,\ldots,v_i\} \subset \Gamma$, and $|v_i| > |v_j|$ if $i < j$ とする．

多属性索引においてはAlgorithm1からもわかるように，値の候補を一つずつ順番に調査しているため，候補順位の高いものは暴露率が高くなるということを考慮しなくてはならない．

4.3 ノイズ付加攪乱戦略

前項で定義した暴露率を下げるため，4つのノイズ付加戦略を定義し，リレーションの各属性の値の分布に基づき適切な戦略を適用する手法を提案した．

- **戦略1：頻度の類似した属性集合で多属性索引を生成**

頻度が類似する属性値を多数多属性索引に含むことで，攻撃者の背景知識（頻度情報）と一致するビットパターン候補を増やすことができる．

- **戦略2：偽属性の混入**

各属性の頻度情報と頻度情報が全く等しいk-1の偽属性を生成し，多属性索引に含める．この場合，偽属性がk-1個含まれているので，必ずk個のビットパターンの組が索引に含まれることとなる．

- **戦略3：誤検出率の調整**

多属性索引の誤検出を増加させることで，実際にはそのタプルに存在しない値の特徴が発生するため，ビットパターン候補を増加させることができる．しかしながら誤検出率の増加は問合せコストの増加をも招くため，注意が必要である．

- **戦略 4：属性値の分割**

頻度の高い属性値は特定しやすく，さらに類似した属性集合を見つけることが難しい．そこで，そのような属性を 2 つ以上の属性集合に分割し，属性集合の頻度を下げることとする．ただし，この場合複数の属性集合に対して問合せを行わなければならなくなるため，検索コストへの影響も考慮しなければならない．

5 実験

本稿では，前項で述べた各戦略を組み合わせ，実際のデータを用いて攪乱を行った結果を提示する．データは UCI[13] より，Adult Dataset を用いる．8 のカテゴリ属性の 32000 レコードで構成されている．図 12 にカテゴリ属性の頻度分布を示す．この Adult Dataset を用いて多属性索引を生成しビットパターン露呈率を測定する．

手法 1 (M1)：戦略 2 のみを採用
手法 2 (M2)：戦略 1，3 を採用
手法 3 (M3)：戦略 1，2，3 を採用
手法 4 (M4)：戦略 1，2，4 を採用

手法 1～4 の誤検出率とブルームフィルタの長さは表 1 の通りとする．手法 2 で混入させる偽属性数は各属性につき 2 とする．

表 1 各攪乱手法におけるパラメタ
Table 1. Parameters for Perturbation Methods

	M1	M2	M3	M4
誤検出率	0.001	0.001	0.01	0.001
ブルームフィルタ長	232	668	304	668

5.1 ビットパターン暴露率

4.2 項で定義したビットパターン暴露率の式に基づき，各暴露率を算出した．図 9，10 はそれぞれ値暴露率とレコード暴露率を示す．SINGLE は単属性索引を表す．戦略 2 を適用した手法 2～4 は暴露率 1/3 をおさえる結果となった．2 値しか含んでいない属性 Sex においては，戦略 4 の属性の分割を行わないと，候補が k より増えなかった．また，属性によっては 100,000 以上もの候補が現れる結果となった．値の暴露率・レコードの暴露率を比較すると，暴露率が極端に異なる属性が存在することがわかる．特に native-country の値の暴露率はどの手法を用いても低いが，レコード暴露率は単属性索引，手法 1，2 で大きな値となる．これは属性値の頻度が少数の属性値に大きく偏っていることに起因する．

5.2 問合せ処理速度

図 11 に問合せ処理速度を示す．NOENC は暗号化なし，SINGLE は単属性索引，PFB は PerturbedBF，SBF は ShuffledBF，SONG は多属性索引として利用可能なキーワード検索用索引 [12] である．先行研究である SBF や SONG らの手法に比べ，ビット攪乱手法を適用した手法は 10 倍の処理速度差があり，単属性索引とはほぼ同じ処理速度を実現している．これは本手法におけるサーバでの処理がブルームフィルタの照合のみで実現できることによる．

6. まとめと今後の課題

本稿では，Encrypted DBMS における安全かつ高速な多属性索 PerturbedBF の提案を行った．PerturbedBF は安全かつ高速な多属性索引を実現したが，選択演算にしか対応していない．今後は他演算に対応するため，CryptDB などの他システムとの統合を実現し，実際に EDBMS として機能するシステムを提案することが今後の課題である．

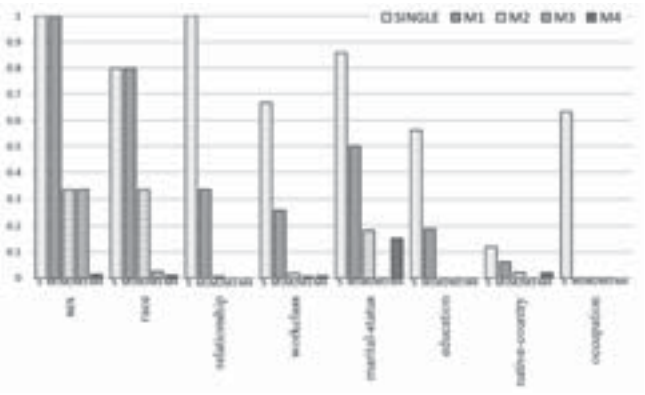

図 9 値の暴露率
Fig.9 Value Revelation Rates

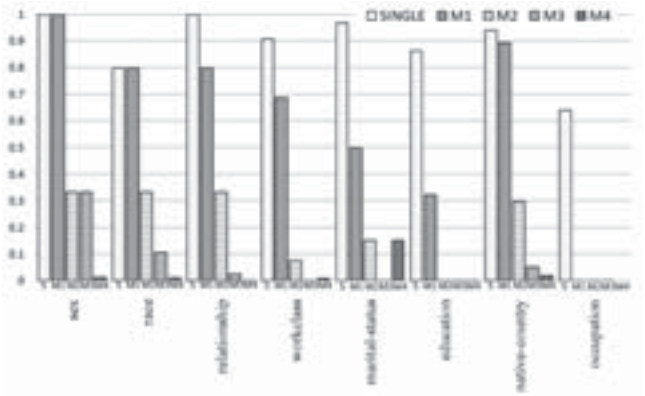

図 10 レコードの暴露率
Fig.10 Record Revelation Rates

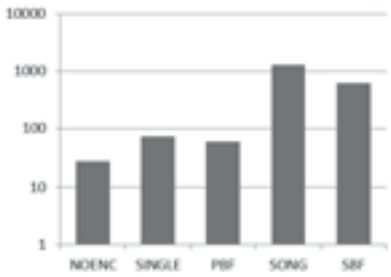

図 11 問合せ処理時間
Fig.11 Query Processing Time

[文献]

[1]H. Hacigumus, B. Iyer, C. Li, and S. Mehrotra.: “Executing SQL over Encrypted Data in the Database-Service-Provider Model,” In Proceeding of the 2002 ACM SIGMOD International Conference on Management of Data, , pp. 216-227, June 2002.

[2]渡辺知恵美，新井裕子: “DaaS におけるスキーマ情報と複合的検索条件を隠蔽したプライバシ保護検索法,” 情報

処理学会研究報告, 2008-DBS-146, Vol.2008, pp.163-168

LNCS, pp.506 522 (2004)

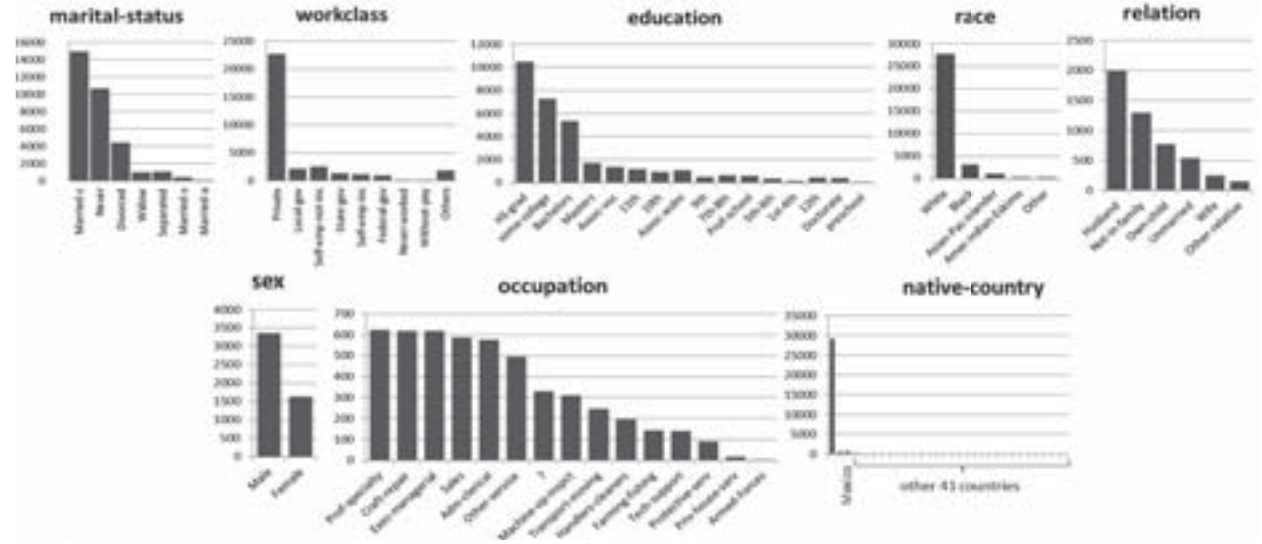

図 12:Adult データセットのカテゴリ属性における属性値の頻度分布

Fig.12 Frequent Distribution of Categorical Attributes in Adult Dataset

[3]Watanabe C. and Arai Y.: “ Privacy-Preserving Queries for a DAS model using Two-Phase Encrypted Bloomfilter,” In Proceeding of International Conference on Database Systems for Advanced Applications, pp. 491-495, 2009.

[4]Kaneko S., Amagasa T. and Watanabe C.: “ Semi-ShuffledBF: Performance Improvement of a Privacy-Preserving Query Method for a DaaS Model Using aBloom filter,” In PDPTA 2011.

[5]Hore B, Mehrotra S. and Tsudik G.: “ A privacypreserving index for range queries,” In Proceedings of the 30th International Conference on Very Large Data Bases, 2004.

[6]S. Lee, T. Paek, D. Lee, T. Nam and S. Kim: “Chaotic Order Preserving Encryption for Efficient and Secure Queries on Databases,” In IEICE Transactions on Information and Systems E92.D(11), 2207-2217(2009)

[7]Kadhem H. Amagasa T. and Hiroyuki K.: “ A Secure and Efficient Order Preserving Encryption Scheme forRelational Databases,” In Int'l Conf. on Knowledge Management and Information Sharing（KMIS2010）, Valencia,(2010)

[8]E. Mykletun, G. Tsudik: “ Aggregation queries in the database-as-a-service model,” In IFIP WG 11.3 on Data and Application Security (2006)

[9]T. Ge, S. B. Zdonik: “ Answering Aggregation Queries in a Secure System Model,” In Proceedings of VLDB 2007, pp.519-530(2007)

[10]R. A. Popa, C. M. S. Redfield, N. Zeldovich, and H. Balakrishnan: “CryptDB: Protecting Confidentiality with Encrypted Query Processing,” In In Proceedings of the 23rd ACM Symposium on Operating Systems Principles (SOSP 2011)

[11]D . Boneh, G.D. Crescenzo, R. Ostrovsky and G. Persiano: “ Public Key Encryption with Keyword Search,” In Proceedings of EUROCRYPT ’04, vol.3027

[12]D.X. Song, D.Wagner, and A. Perrig: “Practial techniques for searches on encrypted data,” In In Proceedings of the 21th IEEE Symposium on Security and Privacy (2000)

[13]C. Blake and C. Merz. UCI repository of machine learning databases (1998)Meehan, M.: “Survey of Multi-User Distributed Virtual Environment”, in Course Notes: Developing Shared Virtual Environments, ACM Press (1999).

金子静花 Shizuka KANEKO

2013 年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修了．日本データベース学会学生会員．2013 年 6 月現在，日本オラクル株式会社に勤務．

柿澤美穂 Miho KAKIZAWA

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．2013 年お茶の水女子大学理学部情報科学科卒業．日本データベース学会学生会員．

渡辺知恵美 Chiemi WATANA BE

お茶の水女子理学部情報科学科講師．データベース，データベースセキュリティ，匿名化，Web マイニング等の研究に従事．日本データベース学会，情報処理学会，ACM 各会員．2013 年 6 月現在，筑波大学システム情報系助教．

天笠俊之 Toshiyuki AMAGASA

筑波大学システム情報系准教授．データ工学，データベース，Web マイニング等の研究に従事．日本データベース学会，情報処理学会，ACM 各会員．電子情報通信学会，IEEE 各シニア会員．