暗号化データベースにおける 統計攻撃に強いノイズ付きブルームフィルタ索引の生成

柿澤　美穂† 　金子　静花† 　渡辺知恵美†

† お茶の水女子大学理学部情報科学科　〒 112–0012 東京都文京区大塚 2 丁目 1-1
E-mail: †{g0920510,shizuka.kaneko,chiemi}@is.ocha.ac.jp

あらまし　近年, データベースの機能をネットワーク経由で利用できる DBaaS が発展してきた. DBaaS は外部の管理者がデータベースサーバを管理・運用するため, 内部で情報漏えいや悪用を行う可能性がある. この問題に対する解決策として, データを暗号化し索引を付与してデータベースに保存し, 暗号化されたまま問合せを行う暗号化データベースがある. 我々はこの暗号化データベースにおいて, ブルームフィルタを用いて複数の属性に対してまとめて一つの多属性索引を生成する手法を提案してきた. しかし, 偏った属性値を持つテーブルから生成されたブルームフィルタ索引では, 属性の頻度情報が明らかになり元の値が推測される可能性がある. そこで我々は, 統計攻撃を受けないようにするためのプライバシ保護指標を定義し, その指標を満たすためのノイズ付与戦略を提案してきた. 本研究では, その指標に基づいたノイズ付きブルームフィルタ索引の生成システムを提案する. 本システムは, 利用者が与えるテーブルの頻度情報を読み取り, 複数の戦略のうち最も適した戦略を適用し, ノイズを付与したブルームフィルタ索引を生成して利用者に返す.

キーワード　セキュリティ, プライバシ保護, データベース

Generation of a secure Bloom-filter-index with the noise to be strong in the statistic information for an Encrypted Database System

Miho KAKIZAWA†, Shizuka KANEKO†, and Chiemi WATANABE†

† 2–1–1 Otsuka, Bunkyo, Tokyo, 112–0012, Japan
E-mail: †{g0920510,shizuka.kaneko,chiemi}@is.ocha.ac.jp

1. はじめに

近年, クラウドコンピューティングサービスの発展に伴い, データベースの機能をネットワーク経由で利用できる DBaaS（DataBase as a Service）が台頭してきた. DBaaS はクラウド上の外部の管理者にデータベースサーバの管理・運用を委託するため, 個人でデータベースを管理・運用するコストを抑えられる有用なサービスであるが, 同時にセキュリティの問題が発生する. つまり, 第三者である管理者が内部での情報漏えいや悪用を行う可能性がある. 利用者は, この管理者からもプライバシを保護し機密情報を安全に保存・検索したいと要求する. この問題に対する解決策として, データを暗号化した状態でデータベースに保存し, 暗号化されたまま問合せを行う暗号化データベース [1] [4] がある. 暗号化データベースでは, クライアントから与えられたテーブルのリレーションの各タプルを暗号化し, その際復号化しなくてもサーバで問合せ処理を可能にするための検索用索引を付与し, データベースサーバに保存する. このようにして暗号化されたままのデータに問合せ処理を施すシステムである. 暗号化データベースで一般的に用いられる索引は属

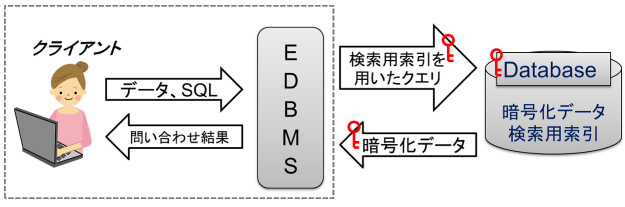


図 1　暗号化データベース

性毎に生成される単属性索引であるため, 元の値と索引値が 1 対 1 の対応関係にある場合, 索引値の統計情報を求めることによって元の値の統計情報も分かってしまう. そのため, 属性値の種類が少ないカテゴリ型の属性や, 値の頻度分布に特徴がある属性は, 索引値の統計情報をもとに元の属性値が推測されやす

いという問題がある. 特に, もし敵が特定の属性に関する統計情報を背景知識として保持していれば, 背景知識と索引値の統計情報を照らし合わせることによって, 属性値を一つに特定されてしまう可能性もある.

そこで, 我々は複数の属性に対して一つの多属性索引を生成する手法を提案してきた. 先行研究 [3] では, 多属性索引を実現する方法としてブルームフィルタを用いたプライバシ保護検索手法を提案してきた. ブルームフィルタとは, 集合にある要素が含まれるかどうかを高速に検索するための索引である. 属性名と属性値の組に対する複数のハッシュ値からビットパターンを生成し, 各タプルのビットパターンの論理和をとったものをブルームフィルタ索引として用いる. この索引を用いることで, 属性値に対するキーワード検索が可能となり, またタプルに含まれる属性値の情報を隠すため, 統計情報による攻撃に強い.

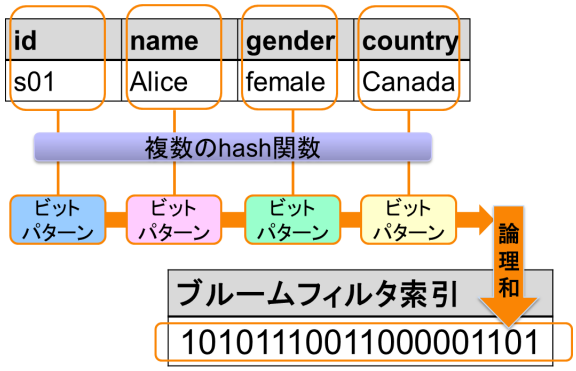


図 2　ブルームフィルタ索引の生成例

しかし, カテゴリ型の属性かつ属性値の頻度分布に偏りがあるといったある特定の場合に, ブルームフィルタ索引のビットの立っている位置から属性の頻度情報が明らかになり, 多属性索引を用いても推測されてしまう恐れがある.

そこで我々は, 統計情報を用いた攻撃を受けないようにするためのプライバシ保護指標を定義し, その指標を満たすためのノイズ付与戦略を提案してきた [3]. 本研究では, 文献 [3] に基づいたノイズ付きブルームフィルタ索引の生成システムを提案する. 利用者が与えるテーブルの頻度情報を読み取り, いくつかの戦略のうち最も適した戦略を組み合わせて適用し, ノイズを付与したブルームフィルタ索引を生成して利用者に返す. このブルームフィルタ索引を検索に用いることで, 安全かつ高速な暗号化データベースが実現すると考える.

本稿の構成は以下の通りである.2 節で先行研究であるブルームフィルタを用いた多属性索引について述べる.3 節でノイズ付きブルームフィルタ索引について,4 節でノイズ付きブルームフィルタ索引生成システムについて述べる. そして,5 節でまとめと今後の課題について述べる.

2. ブルームフィルタを用いた多属性索引

2.1 基本概念

多属性索引とは, 複数属性を対象とした安全な索引である. 複数の属性からまとめて一つの索引を作ることにより, 属性毎の頻度情報を隠すことができる. 我々は, 多属性索引を実現する方法としてブルームフィルタ索引を提案してきた. 属性名と属性値の組に対する複数のハッシュ値からビットパターンを生成し, 各タプルのビットパターンの論理和をとったものをブルームフィルタ索引として用いる. 暗号化データベースにおいて問合せが実行されるとき, このブルームフィルタ索引が利用される. 問合せ手順を以下の例を用いて示す.

図 3 は, 学生情報が入ったテーブルである.

id	name	gender	country
s01	Alice	female	Canada
s02	Bob	male	Australia
s03	Cedric	male	England
s04	Daniel	male	France
s05	Emma	female	USA

図 3　学生情報が入ったテーブル

今, このテーブルに対し, 属性 name の値が Alice であるレコードを求める問合せを行うとする. 問合せを与えられたシステムは, name と Alice をペアにして語 ”name:Alice” とし, これを複数のハッシュ関数にかけ, 得られた結果から配列の該当する箇所にビットを立てる（図 4）.

1000	0100	0000	0000	1000

図 4　問合せ検索語のブルームフィルタ

ここで, 図 3 のテーブルから生成されたブルームフィルタ索引を図 5 に示す.

	BF-index				
tuple1	1110	0110	0110	0000	1101
tuple2	0110	0101	0001	1010	0011
tuple3	0001	1000	0001	1011	0101
tuple4	1110	0101	0001	1001	0110
tuple5	0110	0110	1110	0100	0001

図 5　図 3 のテーブル内の情報のブルームフィルタ索引

検索語 ”name:Alice” から生成されたブルームフィルタに立つビットの位置に, 図 5 の 1 タプル目のブルームフィルタ索引のビットが立っているので, 1 タプル目は検索条件を満たしていると判定される.

2.2 安全性に関する問題点

多属性索引を用いたこの手法は, 単属性索引を用いる手法よりも統計情報による攻撃に強い. しかし, カテゴリ型の属性や値の頻度分布に特徴がある属性に関しては, 元の値を隠しきれず安全性に問題が出る場合がある. 例えば, 図 5 の赤と青の着色部分に注目してほしい. 赤色のビットが立つタプルと, 青色のビットが立つタプルには, 以下の特徴がある.

（1） 赤と青の領域の両方にビットが立つタプルがない

（2） 全てのタプルは必ず赤か青の領域のどちらかにビットが立つ

（3） 赤と青それぞれの領域にビットが立っているタプルの割合が 2:3 である

実は, 赤の領域のビットパターンは ”gender:female” のハッシュ値によってビットが立つ場所であり, 青の領域のビットパターンは ”gender:male” に対応する. 上記の特徴のうち, (1) と (2) は性別や国籍のようなカテゴリ型の属性値に見られる特徴である. 攻撃者がこのようなビットパターンの組を見つけ, そのビットパターンにビットが立つタプルの割合が攻撃者の持つ統計情報と一致すると, ’0011’ は ”gender:male” に対応するビットパターン, ’1100’ は ”gender:female” に対応するビットパターンと特定できてしまう.

さらにタプル数の多いテーブルでも同様のことが言える. データベース管理者が保存されているテーブルの統計情報を知っていて, 生成されたブルームフィルタ索引のビットパターン集合から属性値の頻度情報を得ることができるとき, 統計情報と頻度情報を比較することで属性値の特定が行われてしまう. すなわち, 多属性索引であるブルームフィルタ索引を用いても, 攻撃者が統計情報を持っている場合の攻撃から完全に免れることはできない. この問題を解決するため, 我々はノイズ付きブルームフィルタ索引を提案し, 統計攻撃を受けないようにするためのプライバシ保護指標を定義している.

3. ノイズ付きブルームフィルタ索引

多属性索引は, ある程度統計情報による攻撃に強いが, 完全に統計攻撃を防ぐことが出来る手法ではない. そこで我々は, 攻撃者による統計情報を用いた攻撃に強いノイズ付きブルームフィルタ索引を提案している [3]. ノイズ付きブルームフィルタ索引とは, 以下で定義されているプライバシ保護指標を利用した 4 つのノイズ付与戦略のうち, 適切な戦略を組み合わせて適用しノイズを付与したブルームフィルタ索引である. ノイズを付与することで, カテゴリ型の属性や偏った頻度情報を持つ属性が存在する場合でも, 統計情報を持つ攻撃から免れることができる.

まず我々は, 統計情報を持つ攻撃者による攻撃に対し, ビットパターン集合を利用したプライバシ保護指標を定義している [3]. 暗号化されたリレーションの多属性索引と攻撃者が持つ統計情報を用いて, 属性に対するビットパターン集合候補を求めた時, 必ず k 個以上のビットパターン集合候補が求められるようにする. 攻撃者は, 自分が持っている統計情報を用いてビットパターン集合候補を k 個以下に絞り込むことが出来ない. このプライバシ保護指標を用いて, ビットパターン集合候補を最適な個数にするための戦略を提案する. 以下の戦略のうち, 与えられたテーブルの統計情報に応じて適切な戦略を組み合わせて適用し, ノイズを付与する.

戦略 1　頻度の類似した属性集合で索引を作ることで, 攻撃者が推測するビットパターン集合候補を増やす.

戦略 2　頻度情報が等しい k -1 個の擬似属性を多属性索引に追加し, ビットパターン集合候補を増やす.

戦略 3　誤検出率を増加させ, 多属性索引から抽出される特徴の頻度を増加させる.

戦略 4　頻度の高い属性値を分割し, 属性集合の頻度を少なく見せかける.

これらの戦略を従来のブルームフィルタ索引に導入したものがノイズ付きブルームフィルタ索引である. 本研究では, このノイズ付きブルームフィルタ索引を生成するシステムを提案し, 実装した. 戦略の具体的な適用方法は, 次の節にて例を用いて説明する.

4. ノイズ付きブルームフィルタ索引生成システム

我々は, 統計攻撃に強いブルームフィルタ索引を生成するためのノイズ付与戦略を提案した. この戦略を利用して, 利用者が GUI を用いて頻度情報を隠したい属性とブルームフィルタ索引に必要なパラメータを自由に選択, 設定できるシステムを実装した. 利用者は, 元データをシステムにアップロードし,GUI にて適用する戦略を選択しパラメータを設定する. システムは, 入力された情報をもとにサーバアップロード用データを出力する. 選んだ属性と設定したパラメータによってどの程度頻度情報が隠れるのかを利用者が目で見て確認することができるツールである.

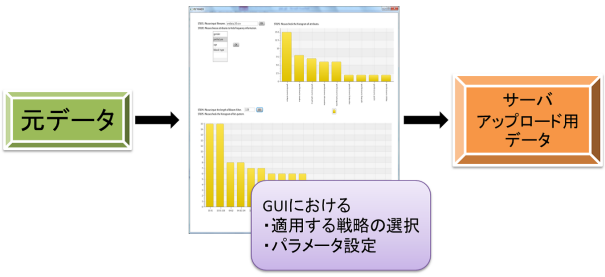


図 6　システムの流れ

ここで, 以下の例によってシステム内部での戦略適用過程を示す. 例えば, 図 7 のような学生情報の入ったテーブルがあるとする.

利用者がこのテーブルをシステムに渡すと, システムは属性毎に統計情報を作成する. id や name は全て同じ頻度であり索引から特徴をつかむことは難しいため, gender と country の 2 属性を選択して索引を生成する. この 2 属性の頻度分布は以下の図 8, 図 9 である.

仮に, 攻撃者が学生情報に関する統計情報「男女比は 3:2, USA 出身者が最も多い」という情報を持っていたとする. すると,「属性値が 2 種類かつ頻度比が 3:2」という情報を示す図 8

id	name	gender	country
s01	Alice	female	Canada
s02	Bob	male	Australia
s03	Cedric	male	England
s04	Daniel	male	France
s05	Emma	female	USA
s06	Fiona	female	USA
s07	Gabriel	male	Germany
s08	Hannah	female	USA
s09	Isaac	male	Canada
s10	Justin	male	USA

図 7　学生情報の入ったテーブル

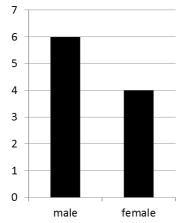


図 8　gender の頻度

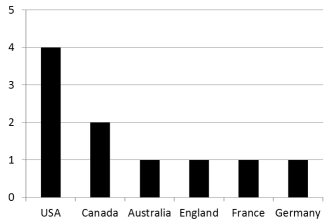


図 9　country の頻度

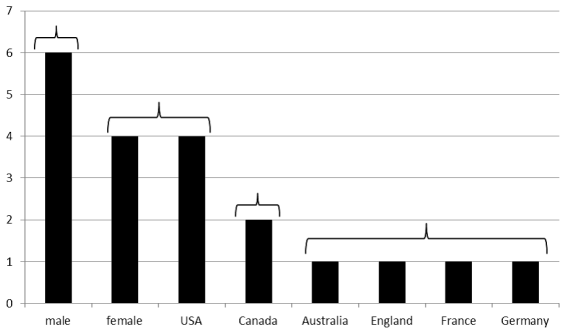


図 10　戦略 1 適用後

···	gender	ph_g1	ph_g2	···
	female	male	male	
	male	female	male	
	male	male	female	
	male	female	male	
	female	male	female	
	female	female	male	
	male	male	female	
	female	male	female	
	male	male	male	
	male	female	male	

図 11　戦略 2 適用後

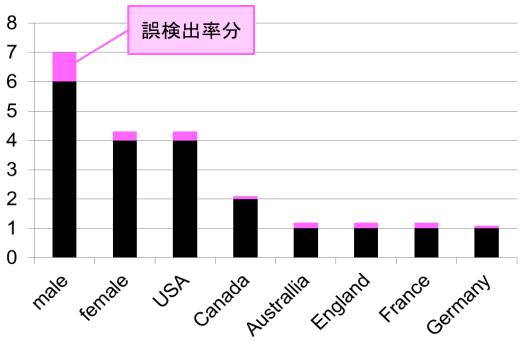


図 12　戦略 3 適用後

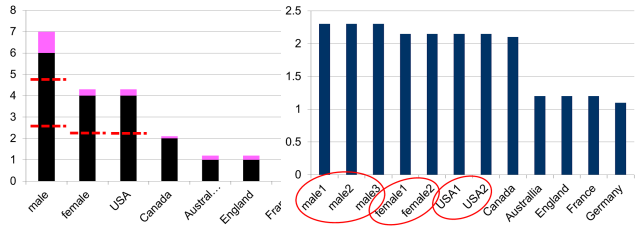


図 13　戦略 4 適用（分割前）　　図 14　戦略 4 適用（分割後）

と,「一つだけ突出している属性値がある」という情報を示す図 9 から, このままの状態でブルームフィルタ索引を生成しても攻撃者から推測されてしまう可能性があることが分かる. そこで, 2.2.2 で説明した戦略を適用する. まず戦略 1 を適用し, 選択した属性の属性値を頻度が高い順に並べ替え, 同じような頻度分布をグループ化する.

ここで, 属性 male のような突出した属性があると特定されてしまうおそれがある. そこで他の属性と同程度の頻度にするため, 戦略 2 を適用する. ここでは, k-1 個の疑似属性を索引に追加し, 必ず k 個のビットパターン集合候補が索引に含まれるようにする.

さらに戦略 3 を適用し, 誤検出率を増加させて偽陽性を持つ索引情報を追加する. 偽陽性を持つ索引情報が上乗せされることで, 敵が持つ統計情報と一致しなくなる.

そして戦略 4 を適用する. 頻度の高い属性値を分割し, 頻度の差が小さく同じ頻度の属性が複数あるヒストグラムを生成する.

図 14 のように, ブルームフィルタを作成するのに最適な頻度分布であると利用者が判断すると, システムはブルームフィルタ索引の長さの見積もり式からブルームフィルタ索引の長さを算出し, ブルームフィルタ索引を生成して利用者に渡す.

この一連の流れをシステム化したものが図 15 である. まず利用者は, 保存したいテーブルをシステムにアップロードする. システムは, 与えられたテーブルから頻度情報を作成する. そして, テーブルに含まれている属性を提示し, 利用者に索引を作るべき属性を選択させる. 利用者は, 頻度情報が明らかになると予想されるカテゴリ型の属性や偏った頻度分布を持つ属性を選ぶ. システムは, ここで選んだ属性の頻度情報を頻度の高い順にヒストグラムで表示する. さらに利用者は誤検出率を入力する. すると, システムでは戦略 1 の適用を開始し, さらに属性の頻度情報を見てどの戦略を適用するかを自動的に判断し, 戦略 2 から戦略 3 もしくは戦略 4 へと適用していく. そして戦略適用後のヒストグラムを利用者に提示する. 利用者は, 頻度情報が隠れるのに十分なヒストグラムであると判断すると, ブルームフィルタ索引生成を承認する. するとシステムは, 入力された誤検出率を用いてブルームフィルタ索引の長さを決定し, ブルームフィルタ索引を生成する. 生成されたブルームフィルタ索引は利用

者に返される.

ブルームフィルタ索引の生成を GUI を利用して実装した理由は二つある. 一つは利用者が頻度情報を隠したい属性を選ぶため, もう一つはパラメータが与える効果を利用者の目で確認しながらパラメータの数値を調節するためである. 頻度情報に特徴のない属性に対してノイズを付与する必要はなく, そのような属性を含む場合全ての属性に戦略を適用することは無駄である. 必要以上のノイズを付与して偽陽性を持つタプルを増やすこと, 反対にノイズを減らしてブルームフィルタの長さを長くすることは, 取り扱うデータの増加につながり非効率である. 本システムの利点は, データベースサーバへアップロードする情報量を減らし効率よくブルームフィルタ索引を生成できる点, 且つ頻度情報が隠れる状態になるまで利用者が目で見ながらパラメータを調節できる点である.

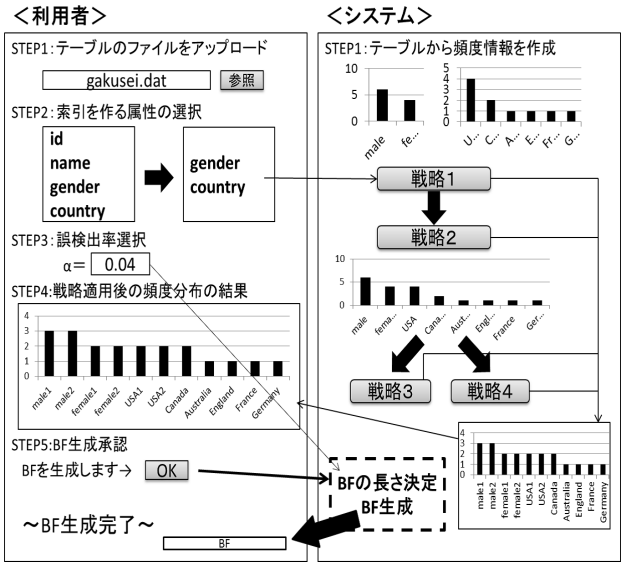


図 15　システム概略図

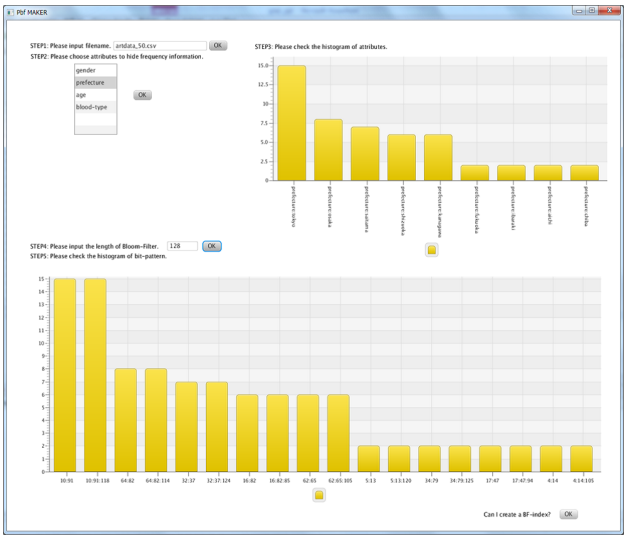


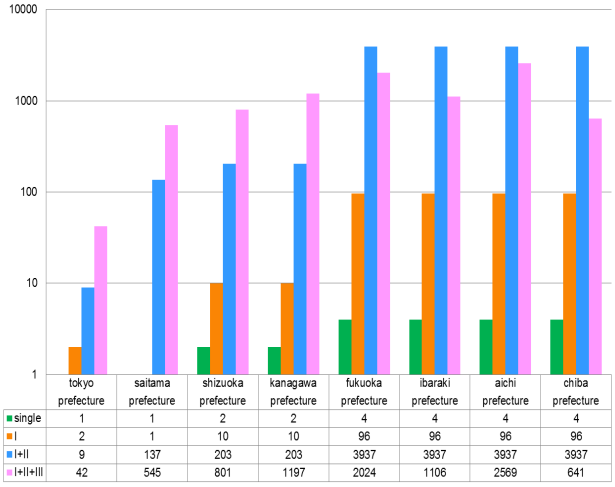
図 16　実際のシステム

5.　検 証 実 験

本研究で提案した戦略を用いた場合に, どれほどビットパターン集合候補数が増加し効果が現れるのかを検証した. 実験対象データは, 属性 gender, prefecture, age, blood-type を持つ 500 タプルの疑似データである. 属性 prefecture に対し, 以下の 4 つの場合のビットパターン集合候補数を比較した.

1　単属性索引を利用
2　属性 blood-type との多属性索引を利用, 戦略は非適用
3　属性 blood-type との多属性索引を利用, 戦略 1,2 を適用
4　属性 blood-type との多属性索引を利用, 戦略 1,2,3 を適用

結果は図 17 のようになった. 横軸は各属性値を表し, 縦軸はビットパターン集合候補数の対数をとっている. 単属性索引や, 戦略を適用しない多属性索引を用いた場合, ビットパターン集合候補は数少なく特定される可能性が高いのに対し, 戦略を適用すると, 候補数は格段に増加していることが見て取れる. この結果から, 提案した戦略が非常に有効な手法であることが分かった.



	tokyo prefecture	saitama prefecture	shizuoka prefecture	kanagawa prefecture	fukuoka prefecture	ibaraki prefecture	aichi prefecture	chiba prefecture
single	1	1	2	2	4	4	4	4
I	2	1	10	10	96	96	96	96
I+II	9	137	203	203	3937	3937	3937	3937
I+II+III	42	545	801	1197	2024	1106	2569	641

図 17　検 証 結 果

6.　まとめと今後の課題

クラウド上の DBaaS において, 利用者はデータの管理者からも機密情報を隠すことを要求する. そこで, データを暗号化してサーバに保存し, 暗号化されたまま問い合わせを行う暗号化データベースが提案されてきた. 我々は, 暗号化データベースにおいて用いられる検索用索引として, 複数の属性を対象とした多属性索引を提案し, それを実現するブルームフィルタ索引の提案を行ってきた. 本研究では, 第三者による統計情報を用いた攻撃モデルと, そのような攻撃を受けないためのプライバシ保護指標を定義した. さらに, プライバシ保護指標を取り入れたノイズ付与戦略の提案と, ノイズ付きブルームフィルタ索引の生成システムを実装した. 今後, 本システムで生成されたノイズ付きブルームフィルタ索引を実際の暗号化データベースに適用できるよう, システムの更なる改良を行う予定である.

文　　献

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

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

[3] 金子 静花, 渡辺 知恵美, 柿澤 美穂, 天笠 俊之: ”暗号化データベースのための安全かつ高速な多属性索引の諸検討”, 第 5 回データ工学と情報マネジメントに関するフォーラム (DEIM2013), F5-5, 2013

[4] R.A.Popa, C.M.S.Redfield, N.Zeldovich, and H.Balakrishman: ”CryptDB: Protecting Confidentiality with Encrypted Query Processing.”,Proceeding of the 23rd ACM Symposium on Operating Systems Principles (SOSP2011), 2011.

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