DAS モデルにおけるプライバシー保護に考慮した範囲検索法

Privacy-Preserving Range Query in a DAS model

新井 裕子♥ 渡辺 知恵美♦
Yuko ARAI Chiemi WATANABE

近年，データベースの管理運用をサービスとして行う DAS (Database as a service)の普及に伴い，データプライバシー保護に対する関心が高まっている．DAS モデルにおいて管理者は第 3 者であるため，不十分な管理や，意図的にデータを悪用することも考えられる．そこで，データを暗号化した後サービスプロバイダに格納することで，管理者に対して機密を保持する研究が行われてきた．その研究の一つに，2 段階暗号を用いた完全一致検索がある[3]．この手法を用いると，管理者に問合せ条件やその結果を知られることなく，サービスプロバイダ側で問合せを実行することができる．本稿では，文献[3]の手法を拡張し，範囲検索を実現する．本手法では，まず各属性のドメインを複数の領域に分割し，属性値をドメインの最小値からその値までの領域と見なす．範囲検索は，領域同士の包含関係を調べることで実現している．包含の判定にはブルームフィルタを用いている．ブルームフィルタは，キーワード検索を行う際に用いられるビット列である．

Recently, data confidentiality becomes major concern with the spread of a DAS model which entrusts database management to an outside administrator. Because an administrator of the outsourced databases is in the third party, users should consider data confidentiality in assumption that an administrator may not have enough skill for managing data or make bad use of the user's data maliciously. To resolve the problem, various mechanisms with cryptographic technologies are proposed. Yang, et al[4] proposed an approach for performing a exact match query without awareness conditions and results of queries to the database administrator by using two-phases encryption method. In this paper, we extend the approach in [4] to perform a range query. In our approach, we first divide each attribute domain into some ranges, and we assume each attribute value as regions from minimal value of the domain to the value. Next we translate the range query condition to the problem of inclusion relation between two ranges. We adopted bloomfilter for the judgment of the inclusion relation. It is a tool of searching keywords quickly.

1. はじめに

近年，個人情報保護法の制定や大規模な情報漏洩事件が次々に明るみに出たことにより，データベースセキュリティに対する意識が急速に高まっている．多くの DBMS 製品は豊富なセキュリティ機能を備えているが，原則としてデータベース管理者に全面的な信頼を置くことを前提に提供されている．そのため，管理不十分による情報漏洩や管理者自身の内部犯行に十分に対処できない．また，データベース製品の管理運用を外部に委託する DAS(Database as a service)モデルが現在普及している．この場合，管理者はデータに関して第 3 者であるため，安全性に問題がある．そこで近年，暗号化したデータと索引をサーバに置くことにより，管理者に対して機密を保持しつつ，サーバ側で問合せを実行する研究が行われている[1]．また同様のシステムを用いた，より安全な索引構成法の研究もある[2]．別の流れでは，暗号化した文章に対するキーワード検索法[4]や，それをデータベースに応用した研究がある[3]．これは，元データを 1 回暗号化したものと 2 回暗号化したものをサーバに置き，2 つの差分を用いることで，サーバ側で安全に問合せを実行するというものである．これらの実用化を考えるとき，完全一致検索に加え範囲検索や結合演算など様々な問合せに対応できなければならない．そこで本研究では，範囲検索に対応した安全な問合せ法を提案する．この手法は参考文献[3]の 2 段階暗号とブルームフィルタを用い実現する．

本稿の構成は，第 2 節：DAS と攻撃モデル，第 3 節：関連研究，第 4 節：2 段階暗号による検索，第 5 節：暗号化データに対する範囲検索法，第 6 節：提案手法の改良，第 7 節：実装，第 8 節：性能評価，第 9 節：まとめと今後の課題である．

2. DAS と攻撃モデル

ここでは，DAS（Database as a service）とその攻撃モデルを示す．DAS とは，専門的な技術を有する企業がデータベースの管理運用をデータ所有者の代わりに行うサービスのことである．サービスを利用する企業は，膨大なデータを管理運用するのに必要な人的コストや物質的なコストを削減することができる．機密データを狙う一般的な攻撃者には，侵入者（システムに対するアクセス権を不法に得て，情報を取りだそうとする者），内部者（信頼されたユーザグループに属し，そのアクセス権外の情報を得ようとする者）が考えられる．DAS モデルでは，それらに加えデータベース管理者もまた攻撃者となりうる．管理者が攻撃者となる例として，①怠惰な管理により他のユーザや侵入者に不正な権利を譲渡してしまった場合，②管理者自身が意図的にデータを第三者に売り渡した場合が挙げられる．管理者が行う攻撃には以下が考えられる．

- 直接的な攻撃：データベースの不正な閲覧・修正・削除
- 間接的な攻撃：ログやシステムカタログによる統計情報の不正な利用
- メモリへの攻撃：サーバに直接アクセス．メモリ上のデータを不正に利用

これらの攻撃は，①データの暗号化（閲覧しても内容把握ができない）②ログへの配慮（解析からの情報漏洩を防ぐ）③データ管理の徹底（サーバのメモリ上にヒントを残さない）により対処することができる．

3. 関連研究

本節では，暗号化を用いたデータプライバシー保護法の研究を紹介する．Hacigums 氏らは，DAS モデルにおける安全な問合せ実行法を提案した[1]．図 1 に問合せの流れを示す．こ

♥ 学生会員 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 yukoarai@db.is.ocha.ac.jp
♦ 正会員 お茶の水女子大学大学院人間文化創成科学研究科 chiemi@is.ocha.ac.jp

こでのクライアントとは，データ所有者およびデータ検索を許されたデータ利用者であり，サーバはデータ管理者（サービスプロバイダ）である．データは全てクライアント側で暗号化・索引付与されてから，サーバへ格納される．索引には，その属性のドメイン領域を適当に分割し割り当てられたハッシュ値を使う．暗号化されたデータに対する検索は，データに付与された索引を用いて，2 段階問合せにより実現する．問合せの一部をサーバで行うため，クライアント側の処理は少なく，様々な問合せを実行することができる．しかし，索引の分布に偏りがある場合，大まかな情報が露出する可能性がある．

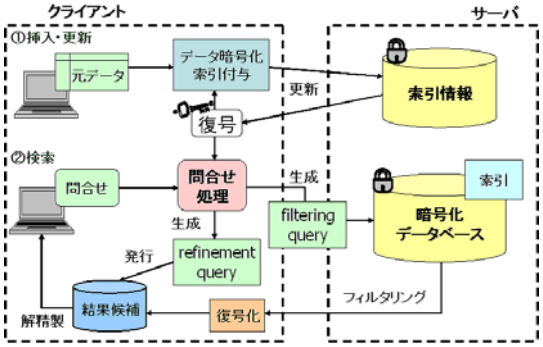


図 1　DAS モデルにおける問合せの流れ

Fig.1 Privacy-preserving query in a DAS model

そこで三浦氏らは，動的な索引構成法を提案した[2]．この研究では，MaxDiff[5]での領域分割法を用いている．これは，1 つのバケットに入るタップル数の差が閾値以下になるように分割する方法である．この方法を用いることにより，バケットによるタップル数の偏りが生じなくなるため，分布解析による情報露出を防ぐことができる．しかしデータ挿入の度に索引情報とデータの更新を行うため，時間コストが高く，更新が頻繁に行われる場合には適さない．同じ流れに，バケット数の最適化に関する研究もある[6]．Hacigums 氏らの場合と同様，更新が頻繁に行われる環境では分布に偏りを生じ，大まかな情報が露出する可能性がある．

DAS モデルにおけるデータプライバシー保護の異なるアプローチとして，Agrawal 氏ら提案した順序付き暗号スキーマ（OPES）がある[7]．これは，ユーザが指定した特有のターゲット分布に元データを変換し，元データの持つ特徴的な分布をなくすことで，情報露出を防ぐ技術である．元データと変換後のデータは 1 対 1 の関係にあり，大小関係も保存されているため様々な問合せが可能である．しかし，元データの大小関係が変換後も保存されていることや，同じ値が同じ値へと変換されることは，データ特徴が露出していると言え，安全面で十分とはいえない．

別の流れに，プライバシー保護に考慮したキーワード検索がある[4]．ここでは，あるキーワードを探索可能にする PEKS（*Public-Key Encryption with keyword Search*）という暗号技術が用いられる．図 2 に PEKS を用いた安全なメール振り分けの例を示す．まずキー生成部で公開鍵と秘密鍵を作る．送信者からのメッセージは公開鍵で暗号化される．これを *emessage* としメールサーバに格納しておく．メール受信者は探索したいキーワードを秘密鍵で暗号化した *trapdoor* を生成し．これをメールサーバに渡す．サーバはこの 2 つを比較し，受信者が必要とするキーワードを含んだメールのみを返すというものである．この研究をデータベースに応用したものとして，2 段階暗号を用いた安全な問合せ法がある[3]．この手法の詳細は 4 章にて示すが，問題点として完全一致検索にしか対応していないことが挙げられる．このように，DAS モデルを想定した研究では，管理者からデータの情報露出を防ぐことはもちろん，問合せの多様性やパフォーマンスも重要な課題となる．

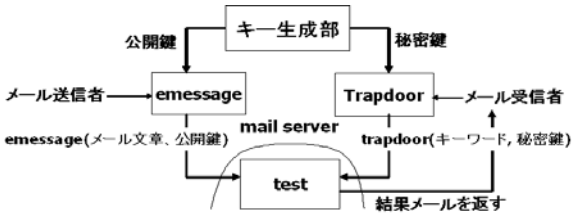


図 2　PECS を用いたメール振り分け

Fig.2 Mail distribution by a PECS

4. 2 段階暗号による暗号化データへの検索

本節では，Yang 氏らが提案した 2 段階暗号[3]を用いた問合せについて述べる．図 3 はデータの挿入および検索の流れを示している．暗号化はセル単位に行われ，その際 2 つの異なる鍵を用いる．鍵①で暗号化されたデータを data1，鍵②で暗号化されたデータを trapdoor，さらに trapdoor を鍵として data1 を暗号化したものを data2 とする．これらの処理は全てクライアント側で行うため，管理者でもデータの内容を知ることはできない．また data1 を生成する際，元の値に乱数を加えることで，同じ値が同じ文字列へと暗号化されることはない．暗号化処理を終えた data1 と data2 はサーバに格納される（図 3①）．つまり，1 つの属性に対して 2 つデータがサーバに格納される．検索は，クライアントで生成される check1 とサーバに格納されている data1 から check2 を生成し，その check2 と data2 を比較することで行われる（図 3②）．check1 とは，ユーザから発行された問合せの where 文に含まれる値を鍵②で暗号化したものである．例えば，テーブル商品（id，商品名，在庫）に対して，ユーザから『select id from 商品 where 在庫≧30』という問合せが発行されたとすると，check1 は 30 を鍵②で暗号化したものとなる．次にサーバ側で check2 を生成する．check2 とは check1 を鍵とし data1 を暗号化したものである．サーバに格納された data2 と生成した check2 を比較することで，問合せの解であるかどうか判定する．問合せの解とされたタップルはクライアントに返され，復号化の後，ユーザに送られる．

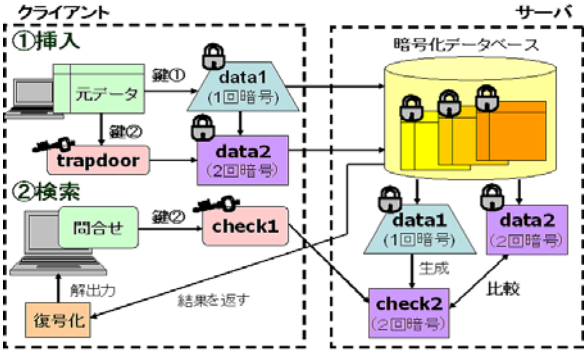


図 3　2 段階暗号を用いた検索

Fig.3 Search in 2-phases encryption schema

テーブル T の i 行 j 列目のセルを T_{ij}，暗号化関数 E( )，乱数 r_i，2 種類の鍵 k1,k2，問合せの where 文に含まれる値を x とし，各要素を式で示すと次のようになる．

$data1 = E_{k1}(T_{ij}+r_i)$
$trapdoor = E_{k2}(T_{ij})$
$data2 = E_{trapdoor}(E_{k1}(T_{ij}+r_i))$
$check1 = E_{k2}(x)$
$check2 = E_{check1}(E_{k1}(T_{ij}+r_i))$

サーバで比較される data2 と check2 の違いは，鍵にある(上式下線部)．trapdoor と check1 はそれぞれ T_{ij}，x を鍵②で暗号化したものであるから，T_{ij}=x(T_{ij} が解)であれば，trapdoor=check1 となるため，data2=check2 となり，問合せの解と判定される．一方 $T_{ij}\neq x$(T_{ij} が解ではない)のとき，trapdoor≠check1 となるため，data2≠check2 となり，問合せの解でないとわかる．

5. 暗号化データベースに対する範囲検索

本研究では 4 節で紹介した 2 段階暗号をベースとし，そこにブルームフィルタを用いることで範囲検索へと拡張した．また，問合せの一部をサーバで行わせることで，クライアントの負担を減らしている．本節では，まずブルームフィルタの概要を示し，後に提案する範囲検索について述べる．

5.1 ブルームフィルタ

ブルームフィルタとは，ある要素がある集合に含まれるかどうかをテストする際に用いられるビット列のことである．あらかじめ適当な数のハッシュ関数と全て 0 に設定された空のブルームフィルタを用意しておく．集合の各要素に対するハッシュ値を計算し，空のブルームフィルタに 1 を立てていく．次に各要素のブルームフィルタの論理和をとり，集合のブルームフィルタとする．テストは集合のブルームフィルタと検索語句のハッシュ値を用いて行われる．検索語句のハッシュ値の全ての位置に 1 が立っていれば，その要素は集合に含まれていると判断される.集合 S={東京都，文京区，大塚}，3 種類のハッシュ関数 hash1，hash2，hash3，8 ビットの空のブルームフィルタを例に説明する．図 4 に集合 S の各要素である『東京都』『文京区』『大塚』から得られたブルームフィルタを示す．集合の各要素のブルームフィルタの論理和をとり，集合 S のブルームフィルタを生成する．次に，このブルームフィルタと検索語句『豊島区』のハッシュ値を比較しテストを行う．検索語句『豊島区』から次のようなハッシュ値が得られたとする．

hash1(豊島区)=2
hash2(豊島区)=8
hash3(豊島区)=1

集合 S のブルームフィルタの 2,8,1 番目のビット列を見ると，2 ビット目が 0 となっているため豊島区は集合 S に含まれないとわかる．一方，検索語句『文京区』のハッシュ値は 1,6,8 である．集合 S のブルームフィルタの 1,6,8 番目のビット列を見ると，すべての位置に 1 が立っているので文京区は集合 S に含まれるとわかる．

	1	2	3	4	5	6	7	8
東京都	0	0	1	1	1	0	0	0
文京区	1	0	0	0	0	1	0	1
大塚	0	0	1	1	0	0	0	1
集合S	1	0	1	1	1	1	0	1

論理和

図 4 ブルームフィルタを用いた文字列検索の様子
Fig.4 String search by a bloomfilter

5.2 提案する範囲検索法

図 5 に提案する範囲検索法の流れを示す．暗号化には 4 節で紹介した手法を用いる．また属性のドメイン領域を適当に分割し，それぞれの領域に対してバケット番号を用意しておく．乱数を加えた後，元データを鍵①で暗号化したものを data1，元データの索引判定をした後，得られた各バケット番号を鍵②で暗号化したものを trapdoor，各 trapdoor を鍵とし data1 を暗号化したものを data2 とする．値によって trapdoor，data2 は複数となる．次に data2 を各要素としたブルームフィルタを生成する．ブルームフィルタと data1 をサーバのデータベースに格納する(図 5：挿入)．また trapdoor を生成する際，2 種類の索引判定を行う．2 種類の索引とは，範囲検索条件『以上』『以下』用の索引である．詳細は 5.3 節の挿入例で示すが，結果としてサーバには 2 つのブルームフィルタが格納されることになる．検索は 5.1 節の図 4 のように，check2 のハッシュ値とサーバに格納されたブルームフィルタを比較することで行われる．check1 は各バケット番号を鍵②で暗号化したものであり，check2 は check1 を鍵としサーバに格納された data1 を暗号化したものである．問合せの解であるとされたタップルはクライアントに返され，復号化・解精製の後，ユーザに送られる(図 5：検索)．

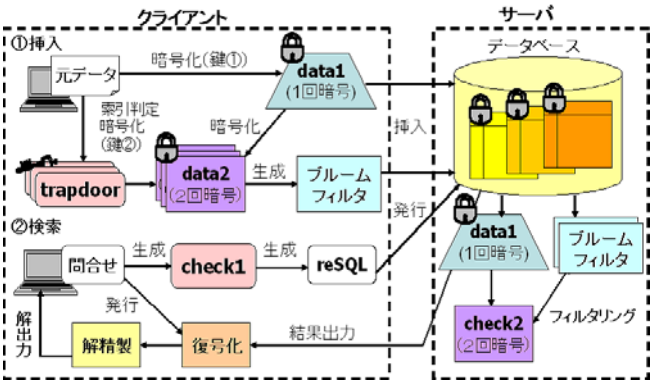


図 5 提案する範囲検索の流れ
Fig.5 A proposing range query

5.3 挿入例

図 6 に値 20，65 を持つ属性『在庫』の挿入例を示す．元データに適当な乱数 r_i を加え，鍵①で暗号化し data1 を生成しておく（図 6①）．次にあらかじめ用意しておいた索引情報から，各値に対して索引判定を行い 2 種類のバケット番号を割り出す．2 種類とは範囲検索に使われる不等号，以上(≧)以下(≦)のことである．『以上』用のバケット $B_{\geqq}$，『以下』用のバケット $B_{\leqq}$とすると，20 は $B_{\geqq}$={a}，$B_{\leqq}$={a,b,c,d}，65 は $B_{\geqq}$={a,b,c}，$B_{\leqq}$={c,d}となる．全てのバケット番号を要素に持

つ全体集合U={a,b,c,d}とすると，$B_{\leqq}$= (該当バケット) ∪($B_{\geqq}$の補集合)となる．各バケット番号を鍵②で暗号化しtrapdoorを生成する(図6②)．trapdoorを鍵としdata1を暗号化することでdata2を生成する(図6③)．data2からブルームフィルタを生成し（図6④)，サーバにそのブルームフィルタとdata1を格納する．

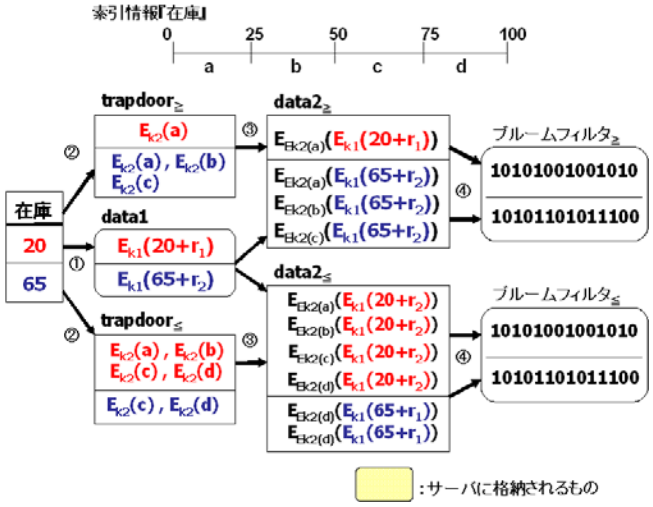


図6　挿入例
Fig.6 Example of insertion

5.3　検索例

図7に，ユーザから『select id from 商品 where 在庫≧30』が発行された場合の検索例を示す．問合せにはブルームフィルタ$_{\geqq}$を用いる．まずwhere文に含まれる30の該当バケットbからcheck1を生成する．check1を用いて問合せを書き換え，サーバに発行する．書き換え後の問合せに含まれるmatch()関数は範囲検索を行う関数である．詳細は7.3節の検索モジュールにて示す．クライアントから渡されたcheck1を鍵として各行のdata1を暗号化し，check2を生成する．1つ目のcheck2からハッシュ値6,8,10が得られたとする．1行目のブルームフィルタの6番目には1が立っておらず，解ではないと判断される．一方，2つ目のcheck2からハッシュ値3,5,6が得られたとすると，2行目のブルームフィルタには，いずれにも1が立っているのでcheck2は含まれ，この元データは30より大きいと判断される．結果はクライアントに返され，解精製の後ユーザに渡される．検索条件が『以下』の場合も，ブルームフィルタ$_{\leqq}$を用いて，同様な処理を行う．

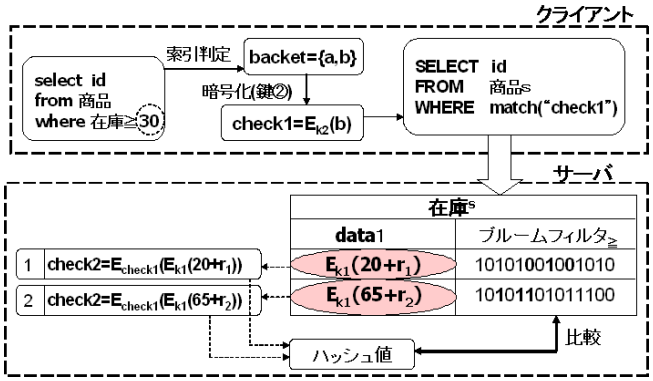


図7　検索例
Fig.7 Example of search

6.　提案手法の改良

前節で提案した範囲検索法には，ドメインの分割数を多くすると検索の精度は上がるが，バケット数が増えるため挿入時間が膨大になるという問題がある．実際，ドメイン[0,10000]，ドメイン領域の分割数2000とし，1000個のレコードを挿入したところ，1レコードあたり5秒ほどかかってしまった．この問題の解決策として我々は多段階の索引構成法を提案する．

6.1　索引構成法

1段階分の分割数をkとしたn段階の索引について述べる．なお分割は等間隔に行うものとする．図8(a)はk=3,n=3としたときの挿入時での索引判定例である．まず，ドメインをk分割し(level1)，値を含むバケット(図8(a)ではバケットb)をさらにk分割する(level2)．これをlevelnになるまで繰り返す．このようなn段階の索引に対し，該当するバケットとそれより左側にあるバケットからブルームフィルタを生成し，サーバに格納する(図8(a)ではバケット a,b,g,h,m,n,o)．これにより，バケットa及びgに該当する部分のバケット数を節約することができる．改良前の手法で，level3相当の分割数を持つ索引を用いた場合，540に対応するバケット数は15個(a:9,g:3,m,n,o)であるが，改良後の索引を用いると7個(a,b,g,h,m,n,o)となり，挿入時間の短縮につながる．n段階の索引，分割数kの平均バケット数はnk/2となる．図8(b)は，検索時の索引判定例である．検索条件は450以上としている．最大分割数を持つ索引以外(図8(b)ではlevel1,2)では，該当するバケットの右隣りにあるバケット(図8(b)ではc,i)，最大分割数を持つ索引(図8(b)ではlevel3)では，該当バケット(図8(b)ではm)からcheck1を生成し，検索を行う．改良前の手法では，検索に用いるバケット数は常に1個であったのに対し，改良手法では，n段階の索引，分割数kとした場合，検索に用いる最大バケット数はnとなり，検索コストは高くなる．挿入コスト・検索コストはs，トレードオフの関係となっているため，適切な分割数，レベル数を検証する必要がある．また，検索条件が以下(≦)の場合，以上(≧)と対称な処理を行えばよい．つまり，540を挿入する際の対応バケットは b,c,h,i,o，450以下を検索する際の対応バケットはa,g,mとなる．

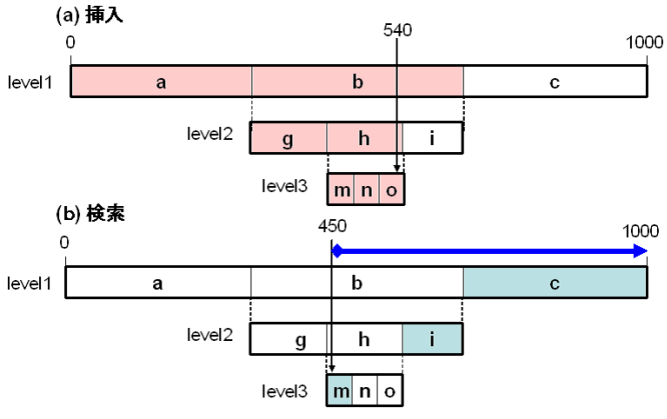


図8　改良法による索引判定
Fig.8 An index judgment by the improved method

7. 実装

範囲検索システムはクライアントで実行される暗号モジュールと reSQL モジュール，サーバで実行される検索モジュールの 3 つから構成される(図 9)．クライアント部分の実装は ruby 言語で行い，RDBMS には PostgreSQL のバージョン 8.2 を用いている．

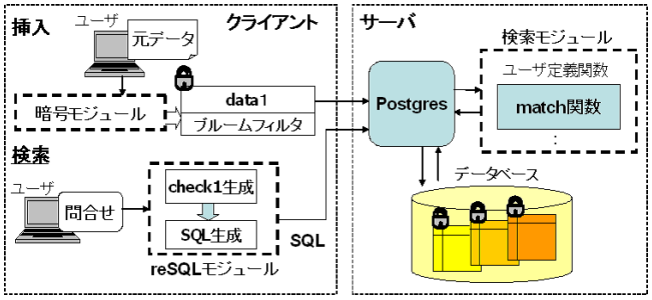


図 9　範囲検索システムの構成
Fig. 9 System configuration of range query

7.1 暗号モジュール

暗号モジュールでは，ユーザから提供された元データを 5.3 節で示した手順で暗号化し，data1 とブルームフィルタをサーバに格納する．5.3 節同様，属性『在庫』を例に，暗号モジュールの様子を示す(図 10)．暗号モジュールでの具体的な処理は，図 6 に対応している．

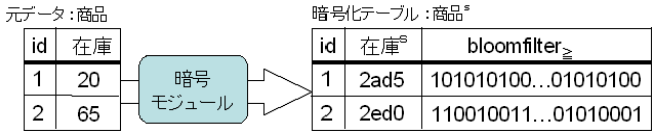


図 10　暗号モジュール
Fig. 10 A encryption module

7.2 reSQL モジュール

reSQL モジュールでは，ユーザが発行した問合せから check1 を生成し，問合せを書き換え,サーバに発行する．ユーザから以下の問合せが発行された場合を例に, reSQL モジュールの様子を図 11 に示す.reSQL モジュールでの具体的な処理は，図 7 で示したクライアントでの処理に対応している．

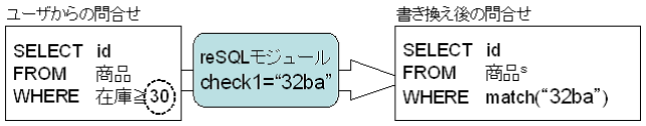


図 11　reSQL モジュール
Fig. 11 A reSQL module

7.3 検索モジュール

検索モジュールでは, reSQL モジュールで書き換えられた問合せを実行し，包含関係を調べることで範囲検索を行う．実際の処理は, 問合せに含まれる match 関数により行われる．match 関数は，PostgreSQL に定義する新たなユーザ定義関数であり，C 言語で実装した．第一引数には，暗号鍵となる check1 が入り，check2 の生成と包含テストが行われる．
前項で示したテーブル商品ˢ を例に，検索モジュールの様子とその結果を図 12 に記す. match 関数の具体的な処理は，図 7 で示したサーバでの処理に対応している．

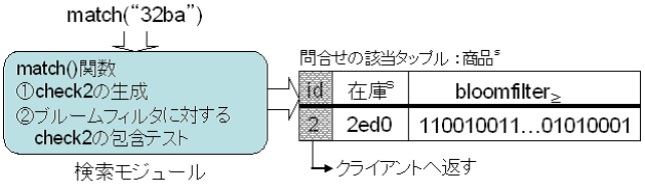


図 12　検索モジュール
Fig. 12 A search module

①check1="32ba"を鍵として，data1 に相当する『在庫ˢ』"2ad5","2ed0"をそれぞれ暗号化し，check2 を生成する
②check2 のハッシュ値と各行の bloomfilter から包含テストを行い，範囲検索を実行する

8. 性能評価

改良前後の手法で性能評価を行った．挿入用データには，[0,1000]の乱数により生成した値を用い, 1000 個のレコードを挿入した. また, 改良前のドメイン領域の分割数を 1000, 改良後では 3 段階の索引，ドメイン分割数は 10 とした．7.1 節の暗号モジュールを用いて挿入を行ったところ，改良前の手法では 2887 秒かかった．単純に 1 レコードあたりの挿入時間を測定すると，平均 2.887 秒となる．一方，6 節で示した改良法を用いて挿入を行ったところ，結果は 72 秒となり，改良前に比べて 1/40 と大幅に減少した. 次に 1000 個のレコードに対して検索を行った. 図 13 に各問合せ(Query1,2,3,4: SELECT * FROM example WHERE value ≧ α(α=800,600,400,200)) に対する改良前後の実行時間を示す．なお改良後の reSQL モジュールでは，各段階の該当バケットから check1 を生成し，各 check1 を引数とした match 関数を or でつなぐことで，問合せを書き換えている．

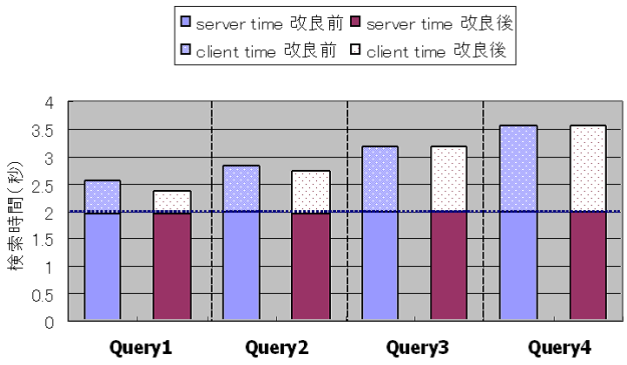


図 13　実行時間の比較
Fig. 13 The comparison of the execution time

各問合せに対する改良前後のサーバの処理時間は共に 2 秒程度となっている．これは，問合せ条件である α が切りの良い値だったため 1 段階目の索引(level1)のみで検索が完了したことが原因だと考えられる．そこで，条件を変え検索を行った(図 14)．なお各問合せは以下のように設定した．(Query1,2,3,4: SELECT * FROM example WHERE

value ≧ β(β=821,621,421,221))

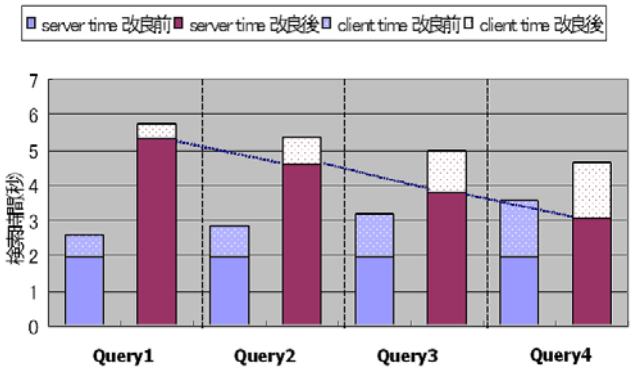
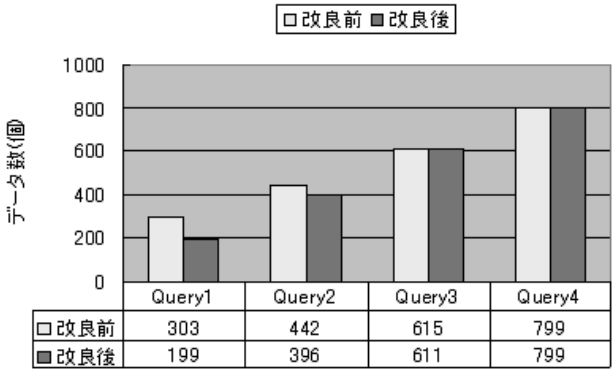


図 14　実行時間の比較
Fig.14 The comparison of the execution time

Query1 では，改良前に比べて 3 倍程度実行時間が増加してしまっている．この理由として，検索条件が 3 つに増えたこと，シーケンシャルスキャンを行っていることが考えられる．また，改良後のサーバでの処理時間が徐々に減少している理由については，221 以上のような解の多い条件の場合，1 段階目の索引(level1)で殆どのデータがフィルタリングされるため，2 段階目 3 段階目の索引を使わなくて済んでいるためだと考えられる．クライアントの処理時間はサーバから返されるタップル数に依存する．改良前後のサーバから返されるタップル数（フィルタリングの結果数）を図 15 に示した．ところで，Query1,2,3,4 における真の解の個数はそれぞれ 199,396,611,799 である．図 15 から，改良前に比べて改良後は false positive が減少していることがわかる．これは，挿入に用いるバケット数の削減により，各バケット番号に対するハッシュ値の重複が減少したためだと考えられる．



	Query1	Query2	Query3	Query4
□改良前	303	442	615	799
■改良後	199	396	611	799

図 15　フィルタリング処理後のタップル数
Fig.15 A number of tuples after filtering processing

9. まとめと今後の課題

本稿では，DAS(Database as a service)モデルにおけるプライバシー保護に考慮した範囲検索法を提案した．攻撃者には，データベースに強い権利を持つデータ管理者を想定している．本手法では，元データと暗号値は 1 対 N の関係になるため，データの閲覧・解析による情報露出に強い．暗号化の手法には，Yang 氏らにより提案された 2 段階暗号を用いた[3]．値を領域とみなし，ブルームフィルタを用いてその包含関係を調べることで範囲検索を実現した．データ挿入時の処理が多く，実行時間が膨大となったため，6 節にて索引構成の改良手法を示した．改良前後の比較・検証を行ったところ，挿入時間の大幅な減少が確認できた．課題には，検索時間の高速化，最適ビット数・バケット数の検証がある．今後は，さらに改良を行い，安全面に関しても提案手法がどの程度のセキュリティに貢献できるかなど検討・検証を行いたい．

[文献]

[1] Hakan Hacigümüş, Bala Iyer, Chen Li, Sharad Mehrotra :``Executing SQL over Encrypted Data in the Database-Service-Provider Model", Proceeding of the ACM SIGMOD International Conference on Management of Data, pp. 216--227, June 2002.

[2] 三浦志保，渡辺知恵美：``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 ",第 18 回データ工学ワークショップ(DEWS2007)，E7-8, 2007.

[3] Zhiqiang Yang, Sheng Zhong, Rebecca N. Wright: ``Privacy-Preserving Queries on Encrypted Data", Proceedings of the 11th European Symposium on Research in Computer Security (Esorics), LNCS4189, pp.479-495, 2006.

[4] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, Giuseppe Persiano: ``Public key Encryption with keyword Search ", EUROCRYPT, LNCS3027, pp.506-522, 2004.

[5] V. Poosala, P. J. Haas et al: ``Improved histograms for selectivity estimation of range predicates," Proceedings of the ACM SIGMOD, pp.294-305, 1996.

[6] Bijit Hore, Sharad Mehrotra, Gene Tsudik : ``A privacy-Preserving Index for Range Queries", Proceedings of the 30th VLDB Conference, pp.720-731 , 2004.

[7] Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, Yirong Xu :``Order Preserving Encryption for Numeric Data", Proceedings of the ACM SIGMOD, pp.563-574, 2004.

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 お茶の水女子大学理学部情報科学科卒業．データベースセキュリティの研究に従事．情報処理学会学生会員．日本データベース学会学生会員．

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 お茶の水女子大学大学院人間文化研究科修了．博士(理学)．データベースシステムの研究・開発に従事．日本データベース学会正会員．