情報処理学会論文誌　データベース　Vol. 4　No. 3　33–44 (Oct. 2011)

キー・バリュー型データベースにおける利用者のプライバシを考慮した範囲問合せの実現手法

川 本 淳 平[†1]　吉 川 正 俊[†1]

本論文では，キー・バリュー型データベースを対象に，利用者のプライバシを考慮した範囲問合せ手法を提案する．従来議論されてきたプライバシを考慮した情報検索（PIR; private information retrieval）の枠組みでは，単独では範囲問合せが行えず，また検索用キー属性の値に重複が認められないという制限がある．提案手法では，キー属性の値と問合せそれぞれに摂動を加え暗号化を施すことで，これらの制限を設けることなく問合せへの頻度分析攻撃を防ぐ範囲問合せを実現する．

Private Range Query in Key-Value Databases

Junpei Kawamoto[†1] and Masatoshi Yoshikawa[†1]

In this paper, we introduce a new private range query method in key-value type database. Existing PIR (Private Information Retrieval) approaches have two limitations: it supports only equal queries but range queries; the key attribute used to query processing must be unique. Our approach, on the other hands, guarantees private range queries and allows tuples having a same value of the key attribute against frequency analysis attacks. For these properties, we add perturbations to both key attributes and queries, and encrypt them.

1. はじめに

キー・バリューデータストア（KVS）は，キー属性 Key とバリュー属性 $Value$ の 2 つからなる単純なデータモデルによって表されるデータベースである．単純なデータモデルを採用しているが，キー属性に複数の属性の組を指定できるものやバリュー属性に構造化されたデータを保存できるものなど柔軟性を確保している製品もある．この KVS は，CAP 定理[1),2)] における，一貫性，可用性，ネットワーク分断耐性のうち，どれを優先するかで 2 種類に分けることができる．Dynamo[3)] や Voldemort[⋆1] などの可用性とネットワーク分断耐性を優先する種類と，Scalaris[4)] や MemcacheDB[⋆2] などの一貫性とネットワーク分断耐性を優先する種類である．どちらの種類もネットワーク分断耐性を重視しており，初めからサーバの分散化を考慮して設計されているものが多い．KVS はこうした特徴により Web サービスを中心に利用が広がっている．また，このキー属性とバリュー属性のみからなるデータモデルは汎用的で，他のモデルもこのキー・バリュー型のデータモデルとして扱うことができる．たとえば，RESTful[5)] Web サービスは，URI をキー属性としてその URI に紐付けられたリソースをバリュー属性と考えることもでき，この意味で広義 KVS の 1 つと見なすことができる．また，結合演算などを考えなければ通常の RDBMS も広義 KVS の 1 つとして考えられる．本論文では，KVS に加えこうしたキー・バリュー型のデータモデルと見なすことのできるその他のサービスも含めて**キー・バリュー型**データベースとして扱う．

一方で，データベース型サービス（DaaS; Database as a Service）においては，サーバに預けるデータの安全性や利用者のプライバシの保護はつねに議論されている問題の 1 つである[6),7)]．その中でも，データベースへの問合せに関するあらゆる情報をサーバを含む攻撃者に漏洩することなく目的のデータを取得したいという要求は古くからある．本論文では，この要求を実現する問合せ手法を**利用者のプライバシを考慮した問合せ手法**と呼ぶ．PIR (Private Information Retrieval)[8)] は，利用者のプライバシを考慮した問合せ手法に関する研究の 1 つである．PIR における仮定では，サーバは 0 から $n-1$ までの数値が割り当てられた n 個の 1 ビットデータ $\{x_0, x_1, \cdots, x_{n-1}\}$ を格納しているデータベースである．キー・バリュー型のデータモデルとして書けば，キー属性の定義域が 0 から $n-1$ までの整数集合 $\{0, 1, \cdots, n-1\}$ であり，バリュー属性の定義域が 0 または 1 の二値集合 $\{0, 1\}$ となる．したがって，PIR の仮定ではキー属性値に対して重複を許さないという制約がある．このように仮定されたデータベースに対して，利用者は i 番目のビット値を問い合わせる．PIR における問題は，問合せである i の値をサーバを含む第三者に知られることなく目的のビット値を取得することである．

†1 京都大学大学院情報学研究科
Graduate School of Informatics, Kyoto University

⋆1 Project Voldemort: http://project-voldemort.com/
⋆2 MemcacheDB: http://memcachedb.org/

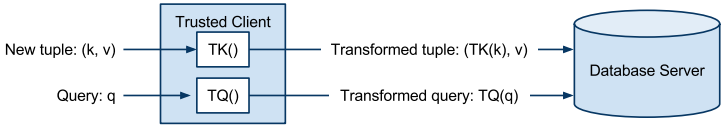

図 1　信頼できるクライアントによる書き換えスキーム
Fig. 1　Query transformation by trusted client scheme of IPP method.

この PIR は利用者のプライバシを考慮した問合せ手法を実現しているが，**範囲問合せ**と**キー属性値の重複**という実用上必要である 2 つの機能が提供されていない．本論文では，キー・バリュー型のデータベースを対象とし上記 2 つの機能を提供する，利用者のプライバシを考慮した新しい問合せ手法である内積述語法（IPP method; Inner Products Predicate method）を提案する．なお，以降では主に IPP 法における範囲問合せについて議論する．しかし，一致問合せは範囲問合せの特別な場合として考えることができるため以降の議論は一致問合せに対しても成立する．

IPP 法は，信頼できるクライアントによる問合せの書き換えスキーム（**図 1**）を用いて利用者のプライバシを考慮した範囲問合せ手法を実現する．このスキームでは，利用者がキー属性値 k とバリュー属性値 v からなるタプル (k, v) をサーバへ追加する場合，利用者の信頼できるクライアントは変換手続き TK を用いてキー属性値 k を $\mathrm{TK}(k)$ に書き換えたタプル $(\mathrm{TK}(k), v)$ を作成しサーバへ送信する．また，利用者が内容の自明な問合せ q を発行した場合，信頼できるクライアントは変換手続き TQ を用いて問合せ q を $\mathrm{TQ}(q)$ に書き換えサーバへ送信する．なお，対象となるデータベースサービスの利用者が複数人いる場合，すべての利用者にこの変換手続き TK と TQ をあらかじめ通知しておくものとする．本論文で想定する攻撃は，変換後の問合せ $TQ(q)$ とキー属性値 $TK(k)$ に対する頻度分析攻撃（Frequency analysis attack）である．すなわち，攻撃者は変換後の問合せがそれぞれ何度用いられたのか，および変換後のキー属性値を持つタプルがそれぞれいくつあるのかを取得する．それにより，変換前の問合せおよびキー属性値の頻度に関する知識を持つ攻撃者が変換後の値から元の値を計算できるという攻撃を考える．IPP 法は信頼できるクライアントによる書き換えスキームにおいて次の 3 つの機能を提供することで頻度分析攻撃を防ぐ．

(1)　内容が自明な問合せ q の書き換え候補を無数に用意できる．
(2)　元のキー属性値 k の書き換え候補を無数に用意できる．
(3)　安全な問合せおよび安全な問合せ用のキー属性値から直接元の値が計算できない．

機能 (1) により，たとえばある自明な問合せ q が 2 回行われた場合，サーバが受け取る安全な問合せは，q_1^* および q_2^* という異なる値にできる．その結果，攻撃者が問合せの頻度解析により元の問合せ内容を計算することを防ぐ．機能 (2) は，同様の機能をキー属性値に対しても提供するものである．たとえば，あるキー属性値を持つタプルが複数存在した場合でも，書き換えられたキー属性値からは同じ属性値を持つタプルがいくつ存在しているのかを攻撃者が正確に計算することを困難にする．最後の機能 (3) により，機能 (1)，(2) が簡単に破られないことを保証する．

IPP 法では，これらの特徴を問合せとキー属性値に摂動（perturbation）を加えさらに暗号化を施すことで実現する．摂動とは攻撃者による真の値の推測を困難にするために加える乱数のことである[9]．また暗号化には，サーバ上で復号することなく問合せ処理を実行するために行列演算による暗号化を用いる．そのため，キー属性値と範囲問合せをそれぞれベクトルを用いて表現し，サーバ上での問合せ処理をそれらベクトルどうしの内積演算に対応付けている．IPP 法におけるサーバの問合せ処理は，サーバが保持するタプルそれぞれに対して，キー属性値を表すベクトルと利用者からの問合せであるベクトルの内積計算を行うことになり $O(n)$ の計算量が必要である．しかしながら，サーバでの計算量が $O(n)$ より小さい手法を用いた場合，サーバはその問合せに関する何らかの情報を取得しているといえる．なぜなら，内積計算の対象外となるタプルを選別するためには，その問合せが表している範囲を幾ばくか知る必要があるからである．したがって，IPP 法が必要とする計算量 $O(n)$ は，利用者のプライバシを考慮した問合せ手法実現のために必要な下限計算量となっている．

以降の章は次のような構成になっている．2 章では，利用者のプライバシを考慮した問合せ手法に関する既存の研究について述べる．その後，3 章から 5 章にかけて IPP 法について説明する．3 章は基本的な IPP 法を，4 章では計算誤差を考慮した実用的な IPP 法について説明する．5 章は IPP 法のスキームについてまとめ必要なデータサイズや計算コストについて述べる．6 章で評価実験を行い，7 章はまとめである．

2. 関 連 研 究

PIR に関する研究は，使用するサーバの台数で 2 通り（結託しないことを仮定した複数のサーバを利用するものと単一のサーバのみを用いるもの[10]），保証する安全性の度合いで 2 通り（情報理論を基にした安全性と計算量を基にした安全性）の合計 4 種類に分けることができる．その中で，本論文が対象とするキー・バリュー型データベースに適しているのは，単一サーバのみを用い計算量を基にした安全性を保証するもので，cPIR（Computational Private Information Retreval）[11] と呼ばれる種類である．cPIR は，2 つの十分大きな秘匿

Basic cPIR Protocol

1: クライアントは，十分大きな整数 p_1, p_2 を生成し $N \leftarrow p_1 \times p_2$ とする．
2: クライアントは，ベクトル $\mathbf{y} \leftarrow (y_1, y_2, \cdots, y_s)$ をサーバに送る．
ただし，$y_g \in QNR_N, y_i \in QR_N (i \neq g)$.
3: サーバは，データベースすべて走査し，$z_i = \Pi_{j=1}^{t} w_{i,j}$ を計算する．
ただし，$w_{i,j}$ はデータベースの (i,j) 成分が 0 の時 y_j^2，(i,j) 成分が 1 の時 y_j とする．
4: サーバは，z_i からなるベクトル $\mathbf{z} \leftarrow (z_1, z_2, \cdots, z_s)$ をクライアントへ送る．
5: クライアントは，受け取った $\mathbf{z}$ の e 番目の値 z_e が，$z_e \in QR_N$ ならば 0 を，そうでなければ 1 を問合せ結果とする．

図 2　cPIR の基本的なプロトコル
Fig. 2　Basic protocol of cPIR.

された素数 p_1，p_2 と整数 $N = p_1 \times p_2$ を法とする規約剰余群 Z_N^* に対して，任意の整数 $x \in Z_N^*$ が平方剰余[⋆1]である（$x \in QR_N$）か平方非剰余である（$x \in QNR_N$）かの判定は困難であるという仮定に基づいている．

cPIR では，n 個のバイナリデータからなるデータベースを $s \times t = n$ を満たす $s \times t$ 行列として扱う．利用者は i 番目のビット値を求める代わりに，この行列の (e, g) 成分のビット値を求めることになる．問合せのプロトコルは**図 2** に示したとおりである．このプロトコルの特徴は，(1) 1 ビットの値を得るために s ビットを送信し t ビットを受信する，(2) 1 ビットではなく m ビットを保存する場合これらの仕組みを m 個用意する，(3) サーバは問合せのたびにデータベース上のすべての値を調べる必要がある，の 3 つである．特に (1) のために総データ数に応じて通信コストが大きくなり (2) のために各タプルのデータサイズに応じてサーバ計算時間が長くなる．本論文で提案する IPP 法は，データベースが保持する総データ数と通信コストは関係せず，各タプルのデータサイズとサーバにおける計算時間も関係しない手法である．

cPIR のサーバにおける計算時間問題を改善する手法として bbPIR[12] がある．bbPIR は，cPIR に k-匿名性[13] の概念を加えたもので，利用者による問合せが k 個のタプルのうちどれを求めていたのかを隠すことのみを保証する代わりに高速化を行っている．具体的には，cPIR における問合せ処理の前にデータベース行列の部分行列として $\sqrt{k} \times \sqrt{k}$ のウインドウを指定する．その後のプロトコルは，このウインドウの内部のみを対象として行われる．したがって，サーバは k 個のタプルを保存する $\sqrt{k} \times \sqrt{k}$ 行列と見なすことができ，問合せ実行時間を削減することができる．また，cPIR を位置情報データベースに用い一種の範囲問合せを実現する手法も提案されている[14]．位置情報データベースにおける範囲問合せとは，利用者が興味のある位置情報として緯度・経度の範囲を問い合わせるものをいう．ここでは 2 次元空間をあらかじめ n 個の部分領域に分割する．そして，ある領域が問い合わされるとクライアントはその領域を含む部分領域の番号を計算する．後はその番号を求める PIR プロトコルを実行することになる．したがって，あらかじめ定められた範囲に対する問合せしか行えないという問題がある．一方，IPP 法では任意の問合せ範囲を用いることができる．

PIR に関する研究以外でも，暗号化データベース[7] に関する研究成果の一部[15),16)] は利用者のプライバシを考慮した問合せを**部分的**に実現する．暗号化データベースでは，サーバには暗号化されたデータしか保存されておらずその暗号化データに対する効率的な問合せ処理の実現が目的となっている．そのため，問合せにも平文の値は含まれておらず，この意味で利用者のプライバシを考慮した問合せを実現していると考えることもできる．しかし，以下で紹介する暗号化データベースにおける問合せ手法では，同じ内容の平文問合せは同じ暗号化データベース用の問合せへ変換される．つまり，問合せの頻度分析攻撃は考慮されていない．そのため，元のドメインにおける問合せの頻度に関する情報を持つ攻撃者が，暗号化されたデータベース用の問合せから元のデータに対する問合せへの対応関係を計算できる危険性がある．一方，IPP 法では行列演算による暗号化に加えて問合せとキー属性値に摂動を追加している．その結果，同じ問合せであっても変換後の問合せは無数の値をとり，頻度分析攻撃を防ぐことができる．

文献 7), 15) では，問い合わせる属性のドメインをいくつかのラベル付きバケットに分割し，属性値をその属性値を含むバケットのラベルに書き換えるバケット化（bucketization）を提案している．バケット化では，キーの定義域を全順序が定義されている集合 K を次の 4 条件を満足する m 個の部分集合 K_i（$i = 1, \cdots, m$）に分割する，(1) $K_i \cap K_j = 0$ $(\forall i, j, i \neq j)$，(2) $\bigcup_{i=1}^{m} K_i = K$，(3) $i < j \Longrightarrow s < t$ $(\forall s \in K_i,\ t \in K_j)$，(4) $|K_i| \geq k$ （k，m はセキュリティパラメータである）．キー属性の値が c であるタプルをサーバへ保存する場合，クライアントは問合せ用のキー属性値 c^* として，そのキー属性値が属する部分集合の番号を用いる．たとえば，c が K_i に含まれる場合 $c^* = i$ となる．そしてキー属性の値が $[a, b]$ に含まれるタプルの問合せの書き換えには次の 2 通りがある．まず，$a \in K_x$，$b \in K_y$ かつ $K_x = K_y$ の場合，つまり $x = y$ の場合，問合せは $c^* = x$ となる．次に，$K_x \neq K_y$（$x < y$）の場合，問合せは $x \leq c^* \leq y$ となる．このように，実際のキー属性の

⋆1 $x \in Z_N^*$ が平方剰余であるとは，$y^2 = x \mod N$ を満たす $y \in Z_N^*$ が存在することをいう．

値ではなく k 個以上の要素を含む部分集合の番号を用いて問い合わせることで，攻撃者は k 個の値のうちどの値を求めているのかを知ることができなくなる．すなわち k 匿名性を保証する．

一方で，この方法では問合せ結果に本来求めていた値と異なるタプルが最大 $2k-2$ 個含まれることになる．したがって，部分集合 K_i の分割方法が問合せ効率に関して重要であり，あらかじめ対象となるドメインに関する統計情報が必要となる．また，ある部分集合に含まれる属性値の種類が少ない場合，元の値が計算できてしまうという問題も指摘されている．この問題に対して，文献 17) では，それぞれの部分集合には少なくとも l 種類以上の値が含まれている（l-多様性）分割方法を提案している．これらの手法を安全にかつ効率的に利用するためには，サーバへ追加されるタプルにおけるキー属性値の分布があらかじめ分かっている必要がある．そうでなければ，利用状況に応じては当初 k-匿名性や l-多様性を満たしていたのに途中で満足しなくなる恐れがある．データの追加削除により当初の分割が適さなくなった時点で，DB を再度構築しなおすという方法も考えられているが，更新前後の情報を攻撃者が突き合わせることで情報漏洩が起きてしまう場合も指摘されている[18]．

文献 16) では，ドメインを順序関係を保存し任意の分布に従う別の集合へ変換させる手法である OPES（Order Preserving Encryption Scheme）を提案している．暗号という名が表すとおり変換後の値から元の値を計算できないことが保証されている．また，順保存のため暗号化したまま範囲問合せを行うことが可能である．これら暗号化データベースにおける問合せ手法は，問合せに平文の値を含まずサーバが保持するタプルのキー属性値は k-匿名性などの匿名性を保証する．その結果キー属性値に対する頻度分析攻撃を防いでおり，この点で暗号化データベースにおける問合せ手法の一部は利用者のプライバシを考慮した問合せ手法を一部実現している．しかし，先ほど述べたとおり，問合せの頻度分析攻撃を考慮していないという問題がある．そのため，元の平文問合せの分布に関する知識を持つ攻撃者は，変換後の問合せから元の問合せを計算できる可能性があり，またそこからキー属性値の元の値を計算できる可能性がある．

最後に，行列演算による暗号化を用いて利用者のプライバシを考慮した問合せを実現している研究として SCONEDB[19] がある．SCONEDB はベクトルデータベースに対する k 近傍問合せを実現する暗号化データベースである．k 近傍問合せとは，問合せとして与えられたベクトルとの距離が最も近いものから k 個のベクトルを取り出す問合せである．SCONEDB はサーバへ保存するベクトルデータの暗号化として行列 M を，問合せとしてサーバへ送信するベクトルの暗号化として行列 M^{-1} を乗じサーバ上で内積計算をとることで内容を秘匿したまま計算を行っている．なお SCONEDB は k 近傍問合せに特化した暗号化データベースであり，我々が対象としているキー属性値に対する範囲問合せには直接応用することはできない．

3. 安全な範囲問合せの構成

本論文で，我々は利用者の問合せ内容がサーバを含む攻撃者に知られることなく範囲問合せを実行する IPP 法を提案する．IPP 法の対象は，キー属性とバリュー属性の 2 つ組からなるキー・バリュー型データベース $(Key, Value)$ で次のことを仮定する．

- キー属性の定義域 $\mathbb{D}_K$ は l_K bit の自然数とする．
- キー属性値が閉区間 $[a,b]$（$a,b \in \mathbb{D}_K$）に入るタプルを求める範囲問合せのみを考える．

1 章で述べた 3 つの目的を実現させるため，IPP 法は問合せを述語モデルで考える．すなわち，クライアントは求めているタプルか否かを判定する述語 p を問合せとしてサーバへ送る．サーバは保持しているタプルの中からキー属性の値 k が $p(k)$ を真にするものを検索しクライアントに返す．なお，IPP 法では問合せ述語 p として真偽値の代わりに実数値をとる関数を用い，$p(k)$ が 0 以下のときを真，正のときを偽として扱うことにする．キー属性の値 k が閉区間 $[a,b]$ に含まれる場合のみ 0 以下の値をとる単純な問合せ述語 $p_{a,b}$ には，

$$p_{a,b}(k) = (k-a)(k-b) \tag{1}$$

がある．IPP 法ではこの問合せ述語 $p_{a,b}$ とキー属性値 k に摂動を加え暗号化を施す．

3.1 摂動項の追加

式 (1) の基本的な問合せ述語 $p_{a,b}$ に摂動項として l_d bit の自然数 d を追加する．なお，l_d は与えられるセキュリティパラメータである．摂動項を追加した問合せ述語 $p_{a,b,d}$ は，

$$p_{a,b,d}(k) = (k-a)(k-b)(k+d) \tag{2}$$

となる．仮定より，キー属性値 k と問合せ範囲 a，b，そして摂動項 d はすべて自然数である．したがって，式 (2) の述語が真つまり 0 以下となるのは，$a \le k \le b$ のときに限られる．いい換えれば，キー属性の値が目的の閉区間 $[a,b]$ に含まれるタプルに対してのみ真となり，期待どおりのタプルを取得できる．

次に，キー属性値 k に摂動項として実数 δ を加える．ここで，δ は $|\delta| < 1/2$ を満たすとする．キー属性値 k に摂動項 δ を追加した値 $\tilde{k}_\delta$ は次のとおりである．

$$\tilde{k}_\delta = k + \delta$$

この摂動項を加えた属性値に対しても問合せ述語が正しく機能するように，式 (2) の問合せ述語 $p_{a,b,d}$ を拡張し，

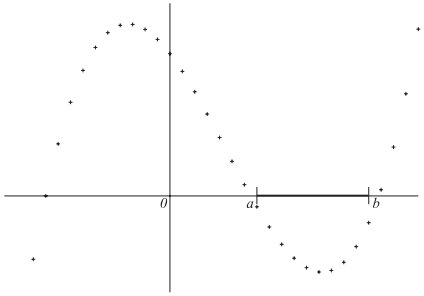

図 3　$\tilde{p}_{a,b,d}(x)$ のグラフ
Fig. 3　Graph of $\tilde{p}_{a,b,d}(x)$.

$$\tilde{p}_{a,b,d}(k) = \left(k - a + \frac{1}{2}\right)\left(k - b - \frac{1}{2}\right)(k + d) \tag{3}$$

とする．この式は，キー属性値が $\tilde{k}_\delta$ の場合次のように計算される．

$$\tilde{p}_{a,b,d}(\tilde{k}_\delta) = \left(k + \delta - a + \frac{1}{2}\right)\left(k + \delta - b - \frac{1}{2}\right)(k + \delta + d)$$

仮定より k，a，b，d は自然数であり，δ は $|\delta| < 1/2$ である．よって，式 (3) の述語が真，すなわち $\tilde{p}_{a,b,d}(\tilde{k}_\delta) \leq 0$ となるのは $k \in [a, b]$ のときに限られ（**図 3**），問合せは期待どおりに実行される．

3.2　問合せとキー属性値の暗号化

摂動を加えた問合せ述語 $\tilde{p}_{a,b,d}$ とキー属性値 $\tilde{k}_\delta$ から直接元の値 a，b および k が計算されること防ぐために暗号化を行う．暗号化された問合せ述語とキー属性値をサーバ上で復号化することなく問合せ処理を実現させるために，行列演算を用いた暗号を利用する．そのために，まずは問合せ述語とキー属性の値をそれぞれベクトルを用いて表現する．

一般的に，多項式は定数ベクトルと変数ベクトルの内積として表現できる．ベクトル化の対象である問合せ述語 $\tilde{p}_{a,b,d}$ は 1 変数多項式である．そこで，1 変数 n 次多項式 $f(x) = \sum_{i=0}^{n} c_i x^i$（$c_0, c_1, \cdots, c_n$ は定数）について考えると，この式はベクトルの内積演算を用いて次のように書ける[⋆1]．

$$f(x) = \sum_{i=0}^{n} c_i x^i = <(c_n, c_{n-1}, \cdots, c_0)^t, (x^n, x^{n-1}, \cdots, x^0)^t>$$

$(c_n, c_{n-1}, \cdots, c_0)^t$ は定数ベクトルであり，$(x^n, x^{n-1}, \cdots, x^0)^t$ は変数ベクトルである．このように多項式は定数ベクトルと変数ベクトルに分離できる．よって，定数ベクトルに問合せ述語を，変数ベクトルにキー属性を対応付けることでベクトル化を行う．式 (3) は，

$$\tilde{p}_{a,b,d}(k) = x^3 - (a+b-d)x^2 + \left(\left(a - \frac{1}{2}\right)\left(b + \frac{1}{2}\right) - (a+b)d\right)x + \left(a - \frac{1}{2}\right)\left(b + \frac{1}{2}\right)d$$

と計算できるから，問合せ述語ベクトル $\tilde{\mathbf{p}}_{a,b,d}$ とキーベクトル $\tilde{\mathbf{k}}_\delta$ は次のようになる．

$$\tilde{\mathbf{p}}_{a,b,d} = \left(1, -(a+b-d), \left(a - \frac{1}{2}\right)\left(b + \frac{1}{2}\right) - (a+b)d, \left(a - \frac{1}{2}\right)\left(b + \frac{1}{2}\right)d\right)^t \tag{4}$$

$$\tilde{\mathbf{k}}_\delta = (\tilde{k}_\delta^3, \tilde{k}_\delta^2, \tilde{k}_\delta^1, \tilde{k}_\delta^0)^t = \left((k+\delta)^3, (k+\delta)^2, (k+\delta), 1\right)^t \tag{5}$$

これらのベクトルを用いたサーバ上での問合せ処理は，保持しているタプルの中からキーベクトルとクライアントから受け取った問合せ述語ベクトルの内積が 0 以下となるタプルの検索になる．この内積演算は，

$$\begin{aligned} <\tilde{\mathbf{p}}_{a,b,d}, \tilde{\mathbf{k}}_\delta> &= \tilde{\mathbf{p}}_{a,b,d}^t \cdot \tilde{\mathbf{k}}_\delta \\ &= (k+\delta)^3 - (a+b-d)(k+\delta)^2 \\ &\quad + \left(\left(a - \frac{1}{2}\right)\left(b + \frac{1}{2}\right) - (a+b)d\right)(k+\delta) + \left(a - \frac{1}{2}\right)\left(b + \frac{1}{2}\right)d \\ &= \left(k + \delta - a + \frac{1}{2}\right)\left(k + \delta - b - \frac{1}{2}\right)(k + \delta + d) \end{aligned} \tag{6}$$

となる．したがって，内積 $<\tilde{\mathbf{p}}_{a,b,d}, \tilde{\mathbf{k}}_\delta>$ が 0 以下となるのは，$(k+\delta-a+1/2)(k+\delta-b-1/2)(k+\delta+d) \leq 0$ となる場合に等しい．この式は，条件 $|\delta| < 1/2$ より，キー属性の値 k が閉区間 $[a, b]$ に含まれる場合のみ真となる．

次に，正則行列を用いた暗号によって問合せ述語ベクトルとキーベクトルを暗号化する．この暗号スキームでは，問合せ述語ベクトルの暗号鍵は 4 次正則整数行列 M である．キーベクトルの暗号鍵は，M の逆行列 M^{-1} を用いた $D_M = |\det(M)|M^{-1}$ となる．なお，$\det(M)$ は行列 M の行列式であり，Cramer's fomula により行列 D_M も 4 次正則整数行列となる[⋆2]．これらの暗号鍵を用いた問合せ述語ベクトルの暗号化手続き $\mathrm{EP}_M(\mathbf{p})$ とキー

⋆1 $<\mathbf{x}, \mathbf{y}>$ でベクトル $\mathbf{x}$ と $\mathbf{y}$ の内積を，$\mathbf{x}^t$ で，ベクトル $\mathbf{x}$ の転置を表す．

⋆2 ある正則行列 A の逆行列は，余因子行列を $\tilde{A}$，行列式 $\det(A)$ とすると，$A^{-1} = \tilde{A}/\det(A)$ で得られる．正数行列の余因子行列はすべての成分が整数であるため，$|\det(A)|A^{-1}$ で得られる行列の各成分も整数となる．

ベクトルの暗号化手続き $\mathrm{EK}_{D_M}(\mathbf{k})$ は次のとおりである．

$$\mathrm{EP}_M(\mathbf{p}) = r_p M^t \mathbf{p},\ \mathrm{EK}_{D_M}(\mathbf{k}) = r_k D_M \mathbf{k}$$

ここで，r_p と r_k は暗号化処理ごとにに異なる自然数乱数である．この暗号化ベクトルに対する問合せ処理は，暗号化前と同様に，保持しているタプルの中からキーベクトルとクライアントから受け取った問合せ述語ベクトルの内積が 0 以下となるタプルの検索になる．この内積計算は次のようになる．

$$\begin{aligned}<\mathrm{EP}_M(\tilde{\mathbf{p}}_{a,b,d}), \mathrm{EK}_{D_M}(\tilde{\mathbf{k}}_\delta)> &= (r_p M^t \tilde{\mathbf{p}}_{a,b,d})^t r_k D \tilde{\mathbf{k}}_\delta \\ &= r_p r_k \tilde{\mathbf{p}}^t_{a,b,d} \cdot M \cdot |\det(M)| M^{-1} \cdot \tilde{\mathbf{k}}_\delta \\ &= r_p r_k |\det(M)| \tilde{\mathbf{p}}^t_{a,b,d} \cdot \tilde{\mathbf{k}}_\delta \\ &= r_p r_k |\det(M)| <\tilde{\mathbf{p}}_{a,b,d}, \tilde{\mathbf{k}}_\delta>\end{aligned}$$

$r_p r_k |\det(M)|$ は正数であるから，$<\mathrm{EP}_M(\tilde{\mathbf{p}}_{a,b,d}), \mathrm{EK}_{D_M}(\tilde{\mathbf{k}}_\delta)>$ は，式 (6) で示したようにキー属性の値 k が $a \le k \le b$ を満たす場合のみ 0 以下となる．したがって，暗号化したベクトルを用いた場合でもキー属性の値 k が閉区間 $[a, b]$ に含まれるときのみ 0 以下となる．いい換えれば，暗号化ベクトルに対する問合せ処理はサーバ上で期待どおりに実行される．このようにして計算される問合せ $\mathrm{EP}_M(\tilde{\mathbf{p}}_{a,b,d})$ およびキー属性値 $\mathrm{EK}_{D_M}(\tilde{\mathbf{k}}_\delta)$ には，それぞれ d，δ という追加のパラメータが含まれている．このパラメータを用いることで問合せ述語ベクトルとキーベクトルの値の書き換え候補を無数に用意することができる．

4. ベクトル成分の整数化

式 (4)，(5) によって定義した暗号化前の問合せ述語ベクトルとキーベクトルの成分は実数値をとる．そのため，サーバにおける内積計算は丸め誤差を含む浮動小数点演算として実行される．その結果，暗号鍵 M，D や問合せ範囲 $[a, b]$，キー属性値 k，および摂動として加えた乱数 d，δ の組合せによっては，誤差によりサーバ上での問合せ処理結果に誤りを含む可能性がある．そこで，サーバ上での計算を整数範囲にとどめ誤りを取り除くために暗号化前の問合せ述語ベクトルとキーベクトルの成分を整数化する．

IPP 法ではキーベクトル成分の整数化に一次近似式を用いる．一次近似式は 1 変数多項式 $f(k)$ に対して δ/k が十分小さい場合に，

$$f(k+\delta) \simeq f(k) + \delta \frac{d}{dk} f(k) \tag{7}$$

が成り立つというものである．近似式を用いることで 2 つのベクトルは誤差を含んだ値となるが，本章では近似による誤差が問合せ処理に影響を与えないことの証明も行う．したがって，ベクトル成分の整数化によって問合せ処理結果が誤りを含まないことを保証できる．

4.1 ベクトルの整数化

式 (4) で定義した暗号化前の問合せ述語ベクトル $\tilde{\mathbf{p}}_{a,b,d}$ において分母の最小公倍数は 4 である．したがって，問合せ述語ベクトルの各成分を整数化するためにはこの値を乗じればよい．よって，整数化された問合せ述語ベクトル $\hat{\mathbf{p}}_{a,b,d}$ は，

$$\begin{aligned}\hat{\mathbf{p}}_{a,b,d} &= 4\tilde{\mathbf{p}}_{a,b,d} \\ &= (4, -4(a+b-d), (2a-1)(2b+1) - 4(a+b)d, (2a-1)(2b+1)d)^t\end{aligned} \tag{8}$$

となる．仮定より a，b，d は整数であるから，このベクトルの成分は整数である．

次にキーベクトルを整数化する．式 (5) で定義した暗号化前のキーベクトルにおいて，δ/k が十分小さいと仮定する．このとき，各成分に式 (7) の一次近似式が使用できて，

$$\tilde{\mathbf{k}}_\delta \simeq (k^3 + 3\delta k^2, k^2 + 2\delta k, k + \delta, 1)^t$$

となる．次に，十分大きな自然数 ϕ を用いて $\delta = 1/\phi$ とする（条件 $|\delta| < 1/2$ を満足する）．k は整数だから $\tilde{\mathbf{k}}_{1/\phi}$ の近似値における分母の最小公倍数は ϕ となる．したがって，キーベクトルの各成分を整数化するためには，ϕ を乗じればよいことになる．よって，整数化されたキーベクトル $\hat{\mathbf{k}}_{1/\phi}$ は，

$$\hat{\mathbf{k}}_{1/\phi} = \phi \left(k^3 + 3\frac{1}{\phi}k^2, k^2 + 2\frac{1}{\phi}k, k + \frac{1}{\phi}, 1 \right)^t = (\phi k^3 + 3k^2, \phi k^2 + 2k, \phi k + 1, \phi)^t \tag{9}$$

となる．仮定より，k および ϕ は整数であるから，キーベクトルの各成分は整数である．

4.2 整数化ベクトルを用いた問合せ処理に関する証明

キーベクトルの整数化では一次近似式を利用した．そのため，整数化されたキーベクトルの各成分は誤差を含んだ値となる．ここでは，この誤差を含むキーベクトルに対してもサーバ上での問合せ処理が期待どおりに実行されることを証明する．そのためにまず，

$$g_{a,b}(k, \phi, d) = <\hat{\mathbf{k}}_{1/\phi}, \hat{\mathbf{p}}_{a,b,d}> \tag{10}$$

という関数を考える．$a \le b$ は問い合わせる範囲，k はキー属性の値，そして ϕ と d は摂動のための値であり，すべて自然数である．問合せが期待どおりに処理されるためには，次の 4 条件

$$g_{a,b}(a, \phi, d) \le 0 \tag{11}$$

$$g_{a,b}(b,\phi,d) \le 0 \tag{12}$$
$$g_{a,b}(a-1,\phi,d) > 0 \tag{13}$$
$$g_{a,b}(b+1,\phi,d) > 0 \tag{14}$$

が満たされていればよい．

まず，$g_{a,b}(a,\phi,d)$ を計算すると，

$$g_{a,b}(a,\phi,d) = -\left(2(b-a)+1\right)(a+d)\phi - 4(b-a)d - 2(2a+1)(b-a) - 1$$

となる．仮定より $0 < a \le b$，$0 < d$，$0 < \phi$ であるから，つねに条件 (11) を満足する．次に，$g_{a,b}(b,\phi,d)$ を計算すると，

$$g_{a,b}(b,\phi,d) = -2\left((b+d)(\phi-2)+1\right)(b-a) - (b+d)\phi - 1$$

となる．よって，同様に $0 < a \le b$，$0 < d$，$0 < \phi$ という仮定のもとで，つねに条件 (12) を満足する．また，$g_{a,b}(a-1,\phi,d)$ および $g_{a,b}(b+1,\phi,d)$ を計算すると，それぞれ，

$$\begin{aligned} g_{a,b}(a-1,\phi,d) &= \left(2(b-a)(\phi-2)+3\phi-2\right)d \\ &\quad + \left((\phi-2)(a-1)+1\right)(2b-2a+3) + 7(2a-3) + 23 \end{aligned}$$
$$g_{a,b}(b+1,\phi,d) = \phi\left(2(b-a)+3\right)(b+d+1) + (4d+2b+3)(b-a+2) - 1$$

となる．どちらも $0 < a \le b$，$0 < d$，$0 < \phi$ という仮定のもとで，つねに条件 (13)，(14) を満足する．以上より，近似を用いた整数化がサーバ上での問合せ処理に影響を与えないことが示された．なお，これらの計算における詳細な式変形は付録に掲載している．

5. IPP 法のスキーム

信頼できるクライアントによる書き換えスキームを用いた IPP 法についてまとめる．IPP 法はキー・バリュー型のデータベースを対象としており次のことを仮定している．

- キー属性の定義域 $\mathbb{D}_K$ は l_K bit の自然数とする．
- キー属性値が閉区間 $[a,b]$（$a,b \in \mathbb{D}_K$）に入るタプルを求める範囲問合せのみを考える．

この仮定を満足するキー・バリュー型データベースに IPP 法を適用するとする．初めに信頼できるクライアントは問合せ述語ベクトルとキーベクトル用の暗号鍵ペア (M, D_M) を生成する．鍵の生成は 16 個の l_m bit 正数乱数からなる 4 次正則整数行列 M を生成した後，$D_M = |\det(M)|M^{-1}$ を計算する．D_M の各成分は Cramer's fomula により M の余因子であり 3 次正方行列の行列式である．したがって各成分は $3l_m$ bit の記憶容量を必要とする．生成された鍵ペアはこのキー・バリュー型データベースサービスを利用するすべての利用者の信頼できるクライアントにあらかじめ配布しておく．

$\mathbf{TK}_{D_M}(k)$
Require: $k \in \mathbb{D}_K$，D_M はキーベクトル用の暗号用鍵
1: 摂動用の l_ϕ bit 自然数乱数 ϕ を生成する
2: キーベクトルを求める: $\hat{\mathbf{k}}_{1/\phi} \leftarrow (\phi k^3 + 3k^2, \phi k^2 + 2k, \phi k + 1, \phi)^t$
3: 暗号化用の l_{rk} bit 自然数乱数 r_k を生成する
4: キーベクトルを暗号化する: $\mathbf{k}^* \leftarrow \mathrm{EK}_{D_M}(\hat{\mathbf{k}}_{1/\phi})$ $(= r_k D_M \hat{\mathbf{k}}_{1/\phi})$
5: **return** $\mathbf{k}^*$

図 4　キー属性値の書き換えアルゴリズム
Fig. 4　Transformation algorithm for Key attribute values.

$\mathbf{TQ}_M(q)$
Require: $a,b \in \mathbb{D}_K$ は問合せ q における問合せ区間，M は問合せベクトル用の暗号鍵
1: 摂動用の l_d bit 自然数乱数 d を生成する
2: 問合せ述語ベクトルを求める:
$\hat{\mathbf{p}}_{a,b,d} \leftarrow (4, -4(a+b-d), (2a-1)(2b+1) - 4(a+b)d, (2a-1)(2b+1)d)^t$
3: 暗号化用の l_p bit 自然数乱数 r_p を生成する
4: 問合せ述語ベクトルを暗号化する: $\mathbf{p}^* \leftarrow \mathrm{EP}_M(\hat{\mathbf{p}}_{a,b,d})$ $(= r_p M^t \hat{\mathbf{p}}_{a,b,d})$
5: **return** $\mathbf{p}^*$

図 5　問合せの書き換えアルゴリズム
Fig. 5　Transformation algorithm for queries.

5.1 IPP 法における変換手続きとサーバにおける問合せ処理

利用者がキー属性とバリュー属性の値がそれぞれ k，v であるタプル (k,v) をデータベースへ追加する場合，利用者の信頼できるクライアントは**図 4** に示す手続き TK によってキー属性値の書き換えを行う．そして得られたキーベクトルを用いたタプル $(\mathbf{k}^*, v)$ を作成しサーバへ送信する．また，利用者がキー属性の値が閉区間 $[a,b]$ に含まれるタプルを問い合わせる場合，信頼できるクライアントは**図 5** に示す手続き TQ によって問合せを書き換える．そして，得られた問合せ述語ベクトル $\mathbf{p}^*$ を用いてサーバへ問い合わせる．

サーバは問合せ述語ベクトル $\mathbf{p}^*$ を受け取ると，保持しているタプルそれぞれに対してそのキー属性値ベクトル $\mathbf{k}^*$ が $<\mathbf{p}^*, \mathbf{k}^*> \le 0$ を満たすか調べる．条件を満足するタプルが問合せ結果でありクライアントへ送信する．この処理はサーバが保持するタプル数を n として $O(n)$ の計算量が必要になる．サーバ上で必要な計算量は既存の PIR 手法と同じである．一方で，IPP 法による問合せ処理手続きは分散化が可能である．タプルを m 台のサー

表 1　IPP 法における必要な記憶容量
Table 1　Comparison of necessary memory size.

種類	本来の記憶容量 (bit)	IPP 法における記憶容量 (bit)
キー属性値	l_K	$12l_K + 4(l_\phi + 3l_m + l_{rk})$
問合せ	$2l_K$	$8l_K + 4(l_d + l_m + l_{rp})$

バに分散させて保存する場合，それぞれのサーバが保持するタプル数は効率的に分散すれば n/m タプルにすることができる．各サーバは並行して問合せ処理を行うことができるため，分散システム環境をうまく構築すれば全体として問合せ処理にかかる計算量は $O(n/m)$ まで改善されることが期待できる．

5.2　必要となる記憶容量

ここでは，IPP 法における書き換え手続きで得られた問合せ述語ベクトルとキーベクトルが必要とする記憶容量と本来の問合せおよびキー属性値が必要とする記憶容量を比較する．まず，変換手続き TK によって得られたキーベクトル $\mathbf{k}^* = r_k D_M(\phi k^3 + 3k^2, \phi k^2 + 2k, \phi k + 1, \phi)^t$ に必要な記憶容量を見積もる．このベクトルのうち，r_k に l_{rk} bit，キーベクトルの暗号鍵 D_M の各成分に $3l_m$ bit が必要である．また，$(\phi k^3 + 3k^2, \phi k^2 + 2k, \phi k + 1, \phi)^t$ で最も大きい値は，$\phi k^3 + 3k^2$ であり $l_\phi + 3l_K$ bit 必要である．したがって，キーベクトル $\mathbf{k}^*$ の各成分には $l_{rk} + 3l_m + l_\phi + 3l_K$ bit 必要であり，全体ではその 4 倍である $12l_K + 4(l_\phi + 3l_m + l_{rk})$ bit が必要である．

次に，生成される安全な問合せベクトル，

$$\mathbf{p}^* = r_p M^t \, (4, -4(a+b-d), (2a-1)(2b+1) - 4(a+b)d, (2a-1)(2b+1)d)^t$$

に必要な記憶容量を見積もる．このベクトルのうち，r_p と暗号鍵 M の各成分にそれぞれ l_{rp} bit と l_m bit が必要である．また，

$$(4, -4(a+b-d), (2a-1)(2b+1) - 4(a+b)d, (2a-1)(2b+1)d)^t$$

の中で最も大きい値は $(2a-1)(2b+1)d$ である．a，b はキー属性と同じく l_K bit，d は l_d bit がそれぞれ必要であるから全部で $2l_K + l_d$ bit が必要になる．よって，安全な問合せベクトル $\mathbf{p}^*$ の各成分には $2l_K + l_d + l_m + l_{rp}$ bit 必要となり，全体で $8l_K + 4(l_d + l_m + l_{rp})$ bit が必要である．**表 1** は，IPP 法を用いた場合に必要となる記憶容量を本来の記憶容量と比較したものである．範囲問合せにおける本来の記憶容量は，区間の下限と上限を送信すると考え $2l_K$ bit 必要であるとしている．

6. 評 価 実 験

IPP 法における，サーバが保存するタプル数と実行速度の比較と，分散数と問合せ実行速度の比較を検証するために評価実験を行った．実験用のサーバは Python（Version: 2.6.4）と Python 上のフレームワークである Tornado [⋆1] を用いて，クライアントも Python（Version: 2.6.4）を用いて実装した．サーバは Intel Core i7-920（2.66 GHz）の CPU と 12 GB の記憶容量を搭載し Windows 7 を OS とするホスト計算機で動作する Oracle Virtual Box [⋆2] 上の仮想機械として実行した．各仮想機械の記憶容量は 512 MB，CPU 数は 1 とし，OS には Ubuntu 10.10 を用いた．また，IPP 法におけるパラメータは次のように定めた．

$$l_K = l_\phi = l_m = l_d = l_{rk} = l_{rd} = 32$$

サーバに保存されているデータ数と問合せ時間の関係についての評価では，データベースは分散を行わず単一のサーバを用いサーバが保持するタプルはキー属性値が一様分布に従うように作成した．問合せ時間の計測は，ランダムな範囲を問い合わせる場合と幅を 100 に固定したランダムな小範囲を問い合わせる場合の 2 通りを行った．また，比較対象として 2 章で紹介したバケット化手法[7] を用い，バケットサイズ k が 9 と 99 の 2 通りについての計測もあわせて行った．なお，問合せ時間の計測はクライアント上で行っており，バケット化手法においては後処理時間[⋆3] も含んでいる．

図 6 は，ランダムな範囲（幅は 1 から 15,000 の間）を計 100 回問い合わせた場合の平均問合せ時間とサーバ上の総タプル数との関係を表したものである．グラフは横軸に総タプル数を対数尺度で，縦軸に問合せ処理時間を記しており，問合せ時間を通常の尺度で記したものと対数尺度を用いたものの 2 種類のグラフを掲載している．図より対数尺度で記したグラフでは直線になっており，IPP 法を用いた場合に要した時間はサーバ上の総タプル数に比例している，すなわち総タプル数を n として $O(n)$ に従うことが分かる．一方，バケット化手法による問合せ処理時間は総タプル数との関係は薄くばらつきのある値となっている．これは，以降で議論するが，バケット化手法においては総タプル数よりも問合せの範囲の方が処理時間に与える影響は大きいためである．

図 7 は，IPP 法の問合せ時間と問合せ範囲の幅を固定したバケット化手法の問合せ時間を比較したものである．IPP 法の問合せ時間は図 6 に記したものと同じである．問合せ範

⋆1 Tornado: http://www.tornadoweb.org/
⋆2 Virtual Box: http://www.virtualbox.org/
⋆3 バケット化は偽陽性を持つため，サーバから受け取った問合せ結果を走査し誤りを取り除く処理が必要となる．

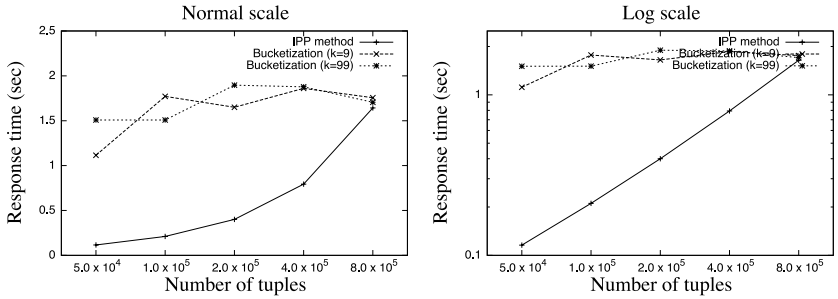

図 6　タプル数と問合せ時間の関係
Fig. 6　Relation between the numbers of tuples and query processing times.

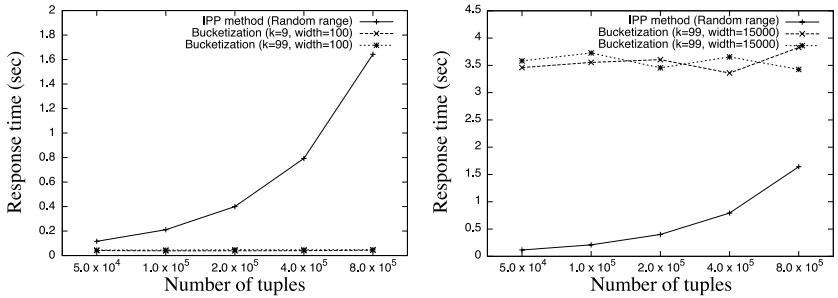

図 7　IPP 法と固定範囲を問い合わせたバケット化手法の問合せ時間比較
Fig. 7　Comparison of querying time between IPP method and Bucketization.

囲が比較的小さい（width = 100）場合，バケット化手法の問合せ処理時間は IPP 法に比べて小さい．特に，問合せ処理に B+ 木などの索引構造を利用できるバケット化手法は，総タプル数の増加にほとんど影響を受けず，IPP 法よりも短い計算時間を実現している．その一方で，問合せ範囲が広い（width = 15,000）場合，バケット化手法は IPP 法に比べてより多くの問合せ処理時間を必要とする．これはバケット化手法がクライアント上での後処理を必要とするためである．バケット化手法では，クライアントはサーバより受け取った全タプルに対して本来求めていた範囲に含まれるタプルか否かを判定する処理が必要である．この処理は，サーバより受け取ったタプル数を n_{res} とすると $O(n_{res})$ の計算量を要する．そのため，問合せ範囲が広く n_{res} が大きくなると，この後処理が無視できない影響を与え

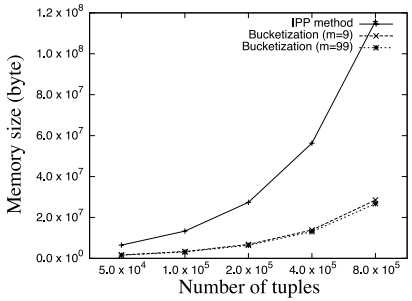

図 8　IPP 法とバケット化手法における必要記憶容量の比較
Fig. 8　Comparison of necessary memory size between IPP method and Bucketization.

IPP 法よりも遅い結果になる．以上より，IPP 法はバケット化手法に比べて計算量クラスとしては遅い処理であるが，比較的広い範囲を問い合わせる場合においてはバケット化手法よりも短い処理時間となる．なお，IPP 法ではランダムな範囲を問い合わせても $O(n)$ に従う問合せ時間を要していることから，問合せ範囲の幅と問合せ処理時間の間に有意な影響は見られず，総タプル数の影響の方が大きい．

図 8 は，IPP 法とバケット化手法のそれぞれにおいて必要な記憶容量を比較したものである．どちらの手法も総タプル数に比例する記憶容量を必要とするが，表 1 に示したように，本来の属性値が要する記憶容量よりも少なくとも 12 倍の記憶容量を必要とする IPP 法の方が，バケット化手法よりも多くの記憶容量を必要としている．なお，バケット化手法の記憶容量には問合せ処理の高速化用に用いる B+ 木索引も含まれている．この索引が必要とする記憶容量は，バケットのサイズ k に依存し，パケットサイズが小さい，つまり総バケット数が大きい方がより大きな記憶容量を必要とする．実際，この実験では $k = 9$ の方が $k = 99$ よりも大きな容量を必要としている．しかし，全体として必要となる記憶容量に比べて大きな差とはならなかった．

サーバ数と問合せ時間の関係について評価では，1.0×10^5 タプルを保存するデータベースと 1.0×10^6 タプルを保存するデータベースの 2 種類を 1〜5 台のサーバに分散させて問合せを行った．サーバが保持するタプルは，先ほどと同様にキー属性値が一様分布に従うように作成した．分散システムの構成は，利用者からの問合せを直接受け取る親サーバを 1 台と，実際にタプルを保存している子サーバ（1〜5 台）からなり，親サーバは受け取った問合せを子サーバへ転送する．タプルの追加時において，それぞれのタプルをどの分散サーバ

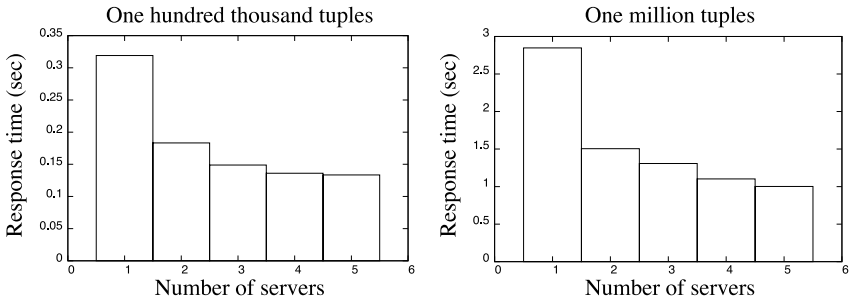

図 9　問合せにおけるサーバ分散数と問合せ時間の関係
Fig. 9　Relation between the numbers of servers and query processing times.

に保存するかの決定は親サーバが行い，分散先の決定はキー属性値ベクトルのハッシュ値をもとに決定した．したがって，ハッシュ値のランダム性を仮定すると，それぞれの子サーバは等しい個数のタプルを保存することになる．

この分散システムに対し，ランダムな範囲を計 100 回問い合わせて平均問合せ処理時間を計測した．**図 9** は子サーバ数と問合せ処理時間の関係を記したものである．分散数を 1 台（未分散）から 2 台に変化させた場合，問合せ処理時間は半分に近い値まで減少することが分かる．その一方で，分散数が 3 台以上では，計算時間は減少するものの $O(n/m)$ からは離れてしまっている．原因としては，分散化によって親サーバと子サーバ間の通信コストが必要となることおよびクライアントとの通信に親サーバを介する必要があり，このサーバ能力がボトルネックとなることが考えられる．したがって，IPP 法におけるサーバ上での問合せ処理は分散可能性を持っているが，分散処理によって高速化を行うためには，効率的な分散システムの構築が必要であり本手法の課題の 1 つである．

7. おわりに

本論文では，広義キー・バリュー型データベースにおける利用者の安全性を考慮した新しい範囲問合せ手法である IPP 法を提案した．IPP 法は，**範囲問合せを主な対象**としており**キー属性の値に重複を許す**という特徴がある．IPP 法では，問合せとキー属性の値に摂動を加え暗号化することで，クライアントによる書き換えモデルにおいて，(1) 内容が自明な問合せの書き換え候補を無数に用意できる，(2) 平文キー属性値の書き換え候補を無数に用意できる，(3) 書き換え後の問合せおよびキー属性値から直接元の値が計算できない，という特徴を持つ問合せ手法を実現している．

IPP 法を用いることで，問合せ内容をサーバを含む攻撃者に秘匿したまま範囲問合せを行うことができる一方で，通常の問合せと比べてサーバの記憶容量やサーバにおける問合せ処理時間が多く必要となる．各タプルのキー属性値および問合せに必要なデータサイズは，表 1 に示すとおりで，平文データを保存する場合と比較してそれぞれ 12 倍以上，4 倍以上のサイズが必要になる．

本論文では，キー属性値にのみ注目しバリュー属性に対する安全性については議論していない．しかし，バリュー属性に関しては一般的な暗号化技術を用いることで安全性を保証することができる．また，タプルの更新および削除に関しては同じキー属性値を持つタプルをすべて取得し更新および削除を行った後再度サーバへ送信するという方法をとることになる．効率的なタプルの更新および削除方法については今後の課題である．さらに，問合せ結果としてクライアントに送信されたタプルデータを大量に集めることで問合せ内容が判別できてしまう可能性も考えられる．この問題も今後の課題である．しかし，一般的に問合せに比べて問合せ結果のデータ量は大きくなる．特に範囲問合せでは複数のタプルを返却するためより大きなデータ量となる．したがって，問合せ結果の解析には大規模なデータ量を保存しなければならず現実的な攻撃手法か否かの議論も必要である．

評価実験により，総タプル数を n としたときに提案手法のサーバにおける処理時間が $O(n)$ に従うことが確認された．また，m 台のサーバを用いた分散環境においては $O(n/m)$ の問合せ時間となる一方でサーバへの通信コストが存在するため一定以上に分散化しても問合せ速度に改善が見られないことも分かった．実用段階においては，ネットワークの構成や想定される保存タプル数などを基に分散数を決定することになる．1 章で述べたように，$O(n)$ のサーバ計算量は利用者のプライバシを考慮した問合せを実現するための下限計算量であると考えている．その一方で，保護するプライバシを一部限定する代わりに高速化する代替案についての議論も必要であると考えている．この $O(n)$ の処理時間を軽減する手法については局所性検知可能ハッシュを用いる手法などが考えられるが今後の課題である．

参考文献

1) Brewer, E.A.: Towards robust distributed systems, *Proc. 19th annual ACM symposium on Principles of distributed computing*, Vol.19, Portland, Oregon, United States, ACM, p.7 (online), DOI:10.1145/343477.343502 (2000).
2) Gilbert, S. and Lynch, N.: Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services, *ACM SIGACT News*, Vol.33, No.2, p.51

(online), DOI:10.1145/564585.564601 (2002).
3) DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., Sivasubramanian, S., Vosshall, P. and Vogels, W.: Dynamo: amazon's highly available key-value store, *Proc. 21st ACM SIGOPS symposium on Operating systems principles - SOSP '07*, New York, New York, USA, p.205, ACM Press (online), DOI:10.1145/1294261.1294281 (2007).
4) Schütt, T., Schintke, F. and Reinefeld, A.: Scalaris: reliable transactional p2p key/value store, *Proc. 7th ACM SIGPLAN workshop on ERLANG*, Victoria, BC, Canada, ACM, pp.41–48 (online), DOI:10.1145/1411273.1411280 (2008).
5) Fielding, R.T.: Architectural styles and the design of network-based software architectures, Ph.D. Thesis, University of California, Irvine (2000).
6) Davida, G.I., Wells, D.L. and Kam, J.B.: A database encryption system with subkeys, *ACM Trans. Database Syst.*, Vol.6, No.2, pp.312–328 (online), DOI:10.1145/319566.319580 (1981).
7) Hacıgümüş, H., Iyer, B., Li, C. and Mehrotra, S.: Executing SQL over encrypted data in the database-service-provider model, *Proc. 2002 ACM SIGMOD international conference on Management of data*, New York, New York, USA, ACM, pp.216–227 (online), DOI:10.1145/564716.564717 (2002).
8) Chor, B., Goldreich, O., Kushilevitz, E. and Sudan, M.: Private Information Retrieval, *J. ACM*, Vol.45, No.6, pp.965–982 (1998).
9) Muralidhar, K. and Sarathy, R.: Security of random data perturbation methods, *ACM Trans. Database Syst.*, Vol.24, No.4, pp.487–493 (online), DOI:10.1145/331983.331986 (1999).
10) Ostrovsky, R. and Skeith, III, W.E.: A Survey of Single-Database PIR: Techniques and Applications, *Proc. 10th international conference on Practice and theory in public-key cryptography*, Beijing, China, pp.393–411, Springer-Verlag (2007).
11) Kushilevitz, E. and Ostrovsky, R.: Replication is not needed: Single database, computationally-private information retrieval, *Proc. 38th Annual Symposium on Foundations of Computer Science*, Washington, DC, USA, IEEE Computer Society, pp.364–373 (online), DOI:10.1109/SFCS.1997.646125 (1997).
12) Wang, S., Agrawal, D. and Abbadi, A.E.: Generalizing PIR for Practical Private Retrieval of Public Data, *Proc. 24th annual IFIP WG 11.3 working conference on Data and applications security and privacy*, Rome, Italy, pp.1–16, Springer-Verlag (2010).
13) Sweeney, L.: k-anonymity: A model for protecting privacy, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, Vol.10, No.5, pp.1–14 (2002).
14) Ghinita, G., Kalnis, P., Khoshgozaran, A., Shahabi, C. and Tan, K.-L.: Private Queries in Location Based Services: Anonymizers are not Necessary Categories and Subject Descriptors, *Proc. 2008 ACM SIGMOD international conference on Management of data*, No.ii, Vancouver, Canada, ACM, pp.121–132 (online), DOI:10.1145/1376616.1376631 (2008).
15) Hore, B., Mehrotra, S. and Tsudik, G.: A Privacy-Preserving Index for Range Queries, *Proc. 30th international conference on Very large data bases – Volume 30*, Toronto, Canada, VLDB Endowment, pp.720–731 (2004).
16) Agrawal, R., Kiernan, J., Srikant, R. and Xu, Y.: Order preserving encryption for numeric data, *Proc. 2004 ACM SIGMOD international conference on Management of data – SIGMOD '04*, New York, NY, USA, ACM, p.563 (online), DOI:10.1145/1007568.1007632 (2004).
17) Machanavajjhala, A., Kifer, D., Gehrke, J. and Venkitasubramaniam, M.: L-diversity: Privacy beyond k-anonymity, *ACM Trans. Knowledge Discovery from Data*, Vol.1, No.1, pp.1556–4681 (online), DOI:10.1145/1217299.1217302 (2007).
18) Wong, R.C.-W., Fu, A.W.-C., Liu, J., Wang, K. and Xu, Y.: Global Privacy Guarantee in Serial Data Publishing, *Proc. 26th International Conference on Data Engineering*, Long Beach, CA, USA (2010).
19) Wong, W.K., Cheung, D.W.-L., Kao, B. and Mamoulis, N.: Secure kNN Computation on Encrypted Databases Categories and Subject Descriptors, *Proc. 35th SIGMOD international conference on Management of data*, Providence, Rhode Island, USA, ACM, pp.139–152 (online), available from ⟨http://doi.acm.org/10.1145/1559845.1559862⟩ (2009).

付　　録

4.2 節で用いた式の詳細な計算を与える．初めに式 (10) の関数 $g_{a,b}(k,\phi,d)$ を変形する．

$$\begin{aligned}
g_{a,b}(k,\phi,d) &= <\hat{\mathbf{k}}_{1/\phi}, \hat{\mathbf{p}}_{a,b,d}> \\
&= 4(\phi k^3 + 3k^2) - 4(a+b-d)(\phi k^2 + 2k) \\
&\quad + ((2a-1)(2b+1) - 4(a+b)d)(\phi k + 1) + (2a-1)(2b+1)\phi d \\
&= 4\phi\left(k^3 - (a+b-d)k^2 + (ab-ad-bd)k + abd\right) \\
&\quad + 12k^2 - 8(a+b-d)k + (2a-2b-1)\phi k \\
&\quad + (2a-1)(2b+1) - 4(a+b)d + (2a-2b-1)\phi d \\
&= 4\phi(k-a)(k-b)(k+d) + (2a-2b-1)(k+d)\phi + 4(2k-a-b)d \\
&\quad + 12k^2 - 8(a+b)k + (2a-1)(2b+1)
\end{aligned}$$

条件式 (11) の $g_{a,b}(a,\phi,d)$ は，$k=a$ のとき $4\phi(k-a)(k-b)(k+d)$ は 0 となるから，

$$\begin{aligned} g_{a,b}(a,\phi,d) &= (2a-2b-1)(a+d)\phi + 4(2a-a-b)d \\ &\quad + 12a^2 - 8(a+b)a + (2a-1)(2b+1) \\ &= -\left(2(b-a)+1\right)(a+d)\phi - 4(b-a)d - 2(2a+1)(b-a) - 1 \end{aligned}$$

と計算できる．次に，式 (12) における $g_{a,b}(b,\phi,d)$ も，$k=b$ のときに $4\phi(k-a)(k-b)(k+d)=0$ であるから，

$$\begin{aligned} g_{a,b}(b,\phi,d) &= (2a-2b-1)(b+d)\phi + 4(2b-a-b)d \\ &\quad + 12b^2 - 8(a+b)d + (2a-1)(2b+1) \\ &= -2(b-a)(b+d)\phi - (b+d)\phi + 4(b-a)d + 2(2b+2d-1)(b-a) - 1 \\ &= -2\left((b+d)(\phi-2)+1\right)(b-a) - (b+d)\phi - 1 \end{aligned}$$

となる．条件式 (13) の $g_{a,b}(a-1,\phi,d)$ は d を含む項と含まない項に分けて整理する．

$$\begin{aligned} g_{a,b}(a-1,\phi,d) &= 4\phi(a-1-a)(a-1-b)(a-1+d) \\ &\quad + (2a-2b-1)(a-1+d)\phi + 4(2a-2-a-b)d \\ &\quad + 12(a-1)^2 - 8(a+b)(a-1) + (2a-1)(2b+1) \\ &= \phi(2b-2a+3)(a+d-1) \\ &\quad + 4(a-b-2)d + 2(a-b-2)(2a-3) + 23 \\ &= \left(\phi(2b-2a+3) + 4(a-b-2)\right)d \\ &\quad + \phi(2b-2a+3)(a-1) + (2a-2b+4)(2a-3) + 23 \\ &= \left(2(b-a)(\phi-2) + 3\phi - 2\right)d \\ &\quad + \left((\phi-2)(a-1)+1\right)(2b-2a+3) + 7(2a-3) + 23 \end{aligned}$$

最後に，条件式 (14) の $g_{a,b}(b+1,\phi,d)$ を計算する．ここでは $b+d+1$ という共通項に注目することで次のように計算できる．

$$\begin{aligned} g_{a,b}(b+1,\phi,d) &= 4\phi(b+1-a)(b+1-b)(b+1+d) \\ &\quad + (2a-2b-1)(b+1+d)\phi + 4(2a+2-a-b)d \\ &\quad + 12(b+1)^2 - 8(a+b)(b+1) + (2a-1)(2b+1) \\ &= \left(4\phi(b-a+1) + (2a-2b-1)\phi\right)(b+d+1) \\ &\quad + 4(b-a+2)d + 2(b-a+2)(2b+3) - 1 \\ &= \phi\left(2(b-a)+3\right)(b+d+1) + (4d+2b+3)(b-a+2) - 1 \end{aligned}$$

(平成 23 年 3 月 20 日受付)
(平成 23 年 7 月 6 日採録)

（担当編集委員　中野 美由紀）

川本　淳平（学生会員）

2008 年京都大学大学院情報学研究科修士課程修了．現在，同大学院同研究科博士課程に在学中．Web における利用者のプライバシ問題に興味を持つ．2008 年より人工知能学会学生編集委員，2009 年より日本学術振興会特別研究委員（DC1）．日本データベース学会，人工知能学会，ACM 各学生会員．

吉川　正俊（正会員）

1985 年京都大学大学院工学研究科博士後期課程修了．工学博士．京都産業大学，奈良先端科学技術大学院大学，名古屋大学を経て，2006 年より京都大学大学院情報学研究科教授．この間，南カリフォルニア大学客員研究員，ウォータルー大学客員准教授．XML データベース，多次元空間索引，異種情報源の統合等の研究に従事．電子情報通信学会，ACM 各会員．