DEIM Forum 2013 F8-5

検索効率を重視したプライバシ保護型検索における タプル情報の漏洩を防止する索引手法

宮崎　昇幸†　　吉川　正俊†

† 京都大学大学院 情報学研究科　〒606-8501 京都市左京区吉田本町
E-mail: †tmiyazaki@db.soc.i.kyoto-u.ac.jp, ††yoshikawa@i.kyoto-u.ac.jp

あらまし　暗号をベースとしたプライバシ保護型検索の枠組みでは，検索に伴う情報漏洩と検索効率とはトレードオフの関係にある．特に，これまで提案されている検索効率化を重視した研究では，索引付けによってタプル間の関係情報を漏洩しており，特定の攻撃に対する安全性を著しく損なってしまう．そこで本稿では，検索効率を重視したプライバシ保護検索において，検索効率を大きく損なうことなくタプル単位での情報漏洩を防止する索引手法を提案する．この索引手法は，互いに識別不能であることを保証したタプルの集合を関係情報漏洩の最小単位としており，かつ安全な更新が可能である．

キーワード　プライバシ保護型検索, データベースアウトソーシング, Database as a Service

Takayuki MIYAZAKI† and Masatoshi YOSHIKAWA†

† Graduate School of Informatics, Kyoto University
Yoshida Honmachi, Sakyo, Kyoto 606-8501 Japan
E-mail: †tmiyazaki@db.soc.i.kyoto-u.ac.jp, ††yoshikawa@i.kyoto-u.ac.jp

1. はじめに

DaaS (Database as a Service) [1] [2] においては，利用者はデータベースシステムの運用・保守に要する管理コストを削減できる一方で，サービス提供者に委託する機密データの安全性をいかにして保証するかが問題となる．データの暗号化は，こうした環境においてデータの安全性を保証するための強力な手段であるが，一般に，安全性を損なうことなく効率的に種々の検索（範囲検索など）を行うことを困難にしてしまう．こうした背景から，安全性と検索効率の両立を目的としたデータ暗号化，および検索手法は盛んに研究されている [3] [4] [1] [2] [5]．本論文では，こうした手法を，検索効率を重視したプライバシ保護型検索と呼ぶ．

委託する機密データに対する脅威は，DaaS のモデルに含まれない外部の攻撃者によるものと，モデル内部の悪意ある主体によるものに分類できる．Shamir の研究 [6] や Blakley らの研究 [7] では，機密データを n 個の断片に分割したうえ複数のサーバに分散配置し，そのうち k 個が収集されたときはじめて元データを復号可能となる k-out-of-n 秘密分散法を提案している．秘密分散法は情報量的安全性を保証しており，実サービスとして提供されている事例もある [8] ．しかし，サービス提供者，ひいてはサーバ（群）の悪意ある挙動は想定しておらず，データ断片を分散したサーバの結託による情報漏洩の危険が存在する．すなわち，外部の脅威への対策は秘密分散法によって一つの解決をみたと言えるが，内部の脅威への対策は未だ研究の余地を残している．

効率的な検索とは，データベースに格納されている全てのデータに対する照合を必要としないことを意味する．そのため，これらの研究課題は，いかにして暗号化データ上に安全な索引付けを行うかという問題に帰着される．アプローチの一つとして，平文の大小関係が暗号文の大小関係にそのまま反映されるよう暗号化を施し，その順序関係を索引付けに利用する手法が提案されている [3]．しかし，この手法はタプル単位の順序関係や平文の出現数が漏洩することを許容しており，ドメインの分布に関する知識を持つ攻撃者に対して脆弱である．

立場の異なるアプローチとして，秘匿したい属性の索引付け，および問合せは全て複数のタプルを含む集合（バケット）単位で行うことで，本来問合せ対象となっていたタプルの匿名性を保証する手法が提案されている [1] [2]．この手法での検索は非常に効率的であると同時に，匿名性の保証によってタプル間の順序関係や値の出現頻度漏洩を防止する．しかし，問合せ結果の中に含まれる偽陽性をフィルタリングするコストが利用者に課されるうえ，更新に対応した動的な安全性の保証にも難がある．

本論文では，暗号化データに対する効率的検索を目的とした上記 2 つのアプローチを統合することで，両者が持つ課題を補完した安全な索引手法を提案する．サービス提供者（サーバ，サーバ群）は悪意を持っていると仮定し，従来タプル単位で許容されていた順序関係の漏洩を，複数の暗号化タプルを含んだバケット単位でしか許容しない．また，バケット内ではタプル間の区別ができないことを保証し，各平文の出現頻度を秘匿する．これらのバケットを葉とする暗号化 B+木索引を構築し，この索引によって各平文の出現頻度に応じた匿名性を保証するバケット分割，および問合せをサポートする．

2. 関 連 研 究

本節では，データベース上の安全な検索を対象とする関連研究を紹介する．

2.1 検索可能暗号

検索可能暗号 (Searchable Encryption : SE) [9] [10] [11] は，秘匿したいデータの暗号文に検索用タグを付加することで，暗号化したままでのキーワード完全一致検索（もしくは部分一致検索）を可能とする暗号化手法である．検索用タグは元のデータから抽出したキーワードを反映した暗号文であり，検索したいキーワードを入力として生成した復号鍵によりこれらのタグが正しく復号されるか否かによってキーワードの一致を判定する．

SE は共通鍵暗号をベースとしたものと公開鍵暗号をベースとしたものに大別されるが，一般には高速性を重視し共通鍵暗号をベースとするものが多い．一方，どちらの場合も検索用タグおよびトークンを用いた照合処理は格納されている全てのデータに対して行う必要があり，全データ数 n に対して $O(n)$ のコストを要する．そのため，対象データが非常に大規模である場合は検索が非効率になるという課題がある．

2.2 プライバシ保護型検索

プライバシ保護型検索は，データベースサービスを提供する信頼できないデータ保持者（サーバ）と利用者のインタラクションによる情報漏洩，およびそれに基づく統計的攻撃も考慮した上で，利用者が問い合わせた内容を秘匿する検索を行うことを目的とする研究である．

Private Information Retrieval (PIR) [12] [13] [14] は，問合せに関する情報をデータベースサーバに一切知られることなくデータを取得することを目的とする研究である．PIR では，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に対して，利用者は i 番目のビット値を i を秘匿したまま問い合わせることが可能である．一方，複数の結託しないことが保証できるサーバおよび膨大な通信コストを必要とするなどの特徴から，実現性のある手法を与えることが困難である．

金子らの研究 [15] では，要素集合に対するある要素の存在検査に用いられる Bloom Filter [16] を索引として利用し，タプルを暗号化したまま一致・範囲問合せを行う手法を提案している．Bloom Filter はビット列同士の OR 演算を伴うため元の属性値を復元することはできず，問合せもハッシュした上でフィルタ化してからサーバに渡されるため対象となる属性およびその値を知られることはない．また，問合せ結果にはハッシュ値の衝突による偽陽性が含まれ，特定のフィルタとタプルとを結びつけられるリスクを低減している．しかし，問合せの度に Bloom Filter の生成，および索引全てに対する存在検査が必要であり，ハッシュ値の計算等に要するオーバーヘッドが問題となる．

川本らの研究 [17] では，摂動を加えた内積範囲述語暗号を検索トークンとして用いるプライバシ保護範囲 (一致) 問合せ手法を提案している．この手法は格納データおよび問合せの匿名性を保証するが，全てのタプルに対する行列演算による照合処理が必要であり，全データ数 n に対して $O(n)$ の検索コストを要する．これに対し，Bloom Filter と局地性検知可能ハッシュを利用した検索効率化も提案されている [18] が，金子らの研究同様，検索結果に含まれる偽陽性のフィルタリングが問題となる．また，暗号化問合せの頻度を用いた統計的攻撃の実現性についても十分な検証はなされていない．

Lu の研究 [5] では，範囲述語暗号を利用した暗号化 B+木を構築することにより，タプルの順序関係漏洩を許容する代わりに対数オーダーでのプライバシ保護一致・範囲検索を実現している．この手法は安全な更新，アクセスコントロール，およびデータ完全性の検証をサポートするが，索引付けによって順序保存暗号化と同様にタプル間の順序関係を即座に漏洩してしまう．先述のようにこれらタプル単位での順序関係漏洩は非常にリスクが高いと考えられるが，検索効率を重視する場合に許容できる程度の情報漏洩であるのかは議論されていない．

2.3 順序保存暗号

順序保存暗号 (Order-Preserving Encryption Scheme : OPES) は，平文の順序関係を保存したまま暗号化を施すことで，暗号化データ上での効率的な検索を可能とする暗号化手法である．順序関係を保存することから，OPES により暗号化されたデータベースを観察した者は立ちどころに全タプルの順序関係情報を入手することが可能である．

Agrawal らの研究 [3] では，暗号化対象となる数値属性のドメインを順序関係を保ったまま別の分布に再マップすることによる OPES を提案している．一方，この手法ではある平文に対する暗号文は一意に決定されるため，当該属性値の頻度分布を知る攻撃者による統計的攻撃に対して脆弱であるという課題がある．これに対し Kadhem らの研究 [4] では，ある平文に対して複数の暗号文を割り当てることが可能な順序保存暗号化スキーム MV-OPES (Multivalued-Order Preserving Encryption Scheme) を提案している．しかしこの研究では，各平文に別のドメインの部分空間を割り当てる際に入力とした頻度分布の偏りが後から変化する場合，全ての部分空間の再割当をしなければ全体が一様分布になることを保証できない．

OPES の暗号学的安全性については近年盛んに議論が成されている [19] [20] が，主眼は「暗号文から順序関係以外の情報を漏洩していないか」という点であり，知識を持つ攻撃者による統計的攻撃に対する脆弱性は考慮されていない．しかし，攻撃者が当該属性値の頻度分布についての完全な知識を持つ場合だけでなく，ドメインの最小・最大値，属性の性質，頻度分布の

一部，もしくは問合せの履歴など不完全な知識しか持たない場合にも，タプル間の順序関係情報との組み合わせにより推論攻撃の信頼度を幾ばくか高められる危険性がある (例 1, 例 2)．このように，OPES にみられるタプル単位での順序関係漏洩が許容できるか否かについては議論の余地が残る．

例 1: 暗号化された属性が「試験の得点」であることを，攻撃者が知り得た事例を考える (図 1)．この知識から，暗号化前の属性値の値域は $[0, 100]$ である可能性が高いことがわかる．ここで，暗号文が順序保存かつ決定論的である（平文と暗号文が一意に対応する）という性質を持つ場合には，テーブルに格納された暗号文の多様性（何種類存在するか）が高いほど，各暗号文の元値推定の信頼度を高めることできる．

また，暗号文が決定論的でない場合にも，問合せ履歴を収集出来るという現実的な仮定のもとでは，攻撃者が暗号文の元となった平文の多様性を収集可能である可能性がある．すなわち，平文が共通するタプルの数を漏洩しない問合せ手法を用いなければ，同様に推論攻撃の信頼度を高められる危険性がある．

例 2: 暗号化された属性の部分的な頻度分布，および問合せ履歴を攻撃者が知り得た事例を考える (図 2)．この場合，ドメインの最小・最大値を高い信頼度のもとに推定することは困難であるため，例 1 のような格納データの多様性に基づく推論攻撃の精度向上は非現実的であると考えられる．その一方で，攻撃者が知る頻度分布の範囲内では，精度の高い推論攻撃が可能である．

例えば，図 2 のような部分的頻度分布（図中実線部）と問合せ履歴の知識を攻撃者が持つことを考える．まず，問合せ履歴から，結果集合 A に属するタプル群は，範囲検索の結果ではなくある共通の平文に対する一致検索の結果である可能性が高いことがわかる．すなわち，結果集合 A は，出現頻度が 10 で

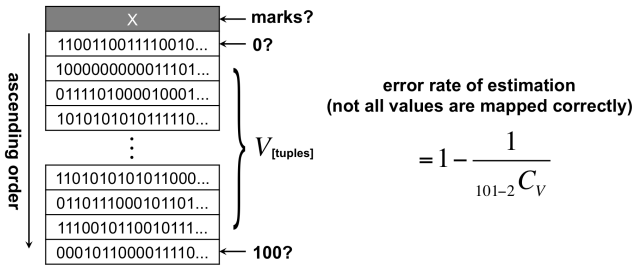

図 1　ドメインの最小値，最大値に関する知識

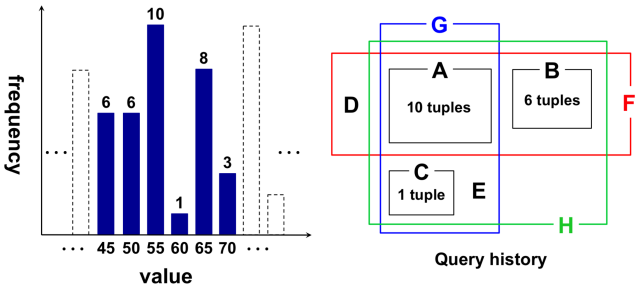

図 2　部分的頻度分布および問合せ履歴に関する知識

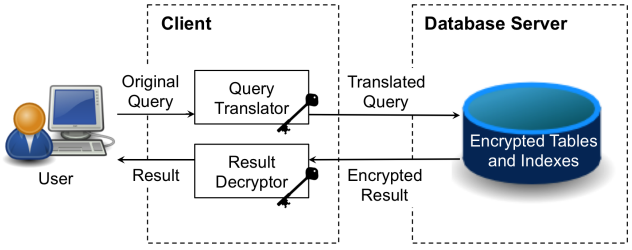

図 3　Database as a Service モデル

ある平文 $ptext_A$ を元とする暗号文の集合に一致する．このことから，$ptext_A$ は攻撃者が知る部分頻度分布のうち，頻度 10 を持つ値 55 である可能性が浮上する．この時点では，推定の信頼度は当該ドメインにおける頻度 10 を持つ属性値の数に依存するため，攻撃者は自身の推定の確からしさを検証することができない．

一方，問合せ履歴を更に観察すると，結果集合 B, C も出現頻度がそれぞれ 6, 1 である平文 $ptext_B, ptext_C$ を元とする暗号文の集合に一致する可能性が高く，かつ昇順に B, A, C，もしくは C, A, B の大小関係があることがわかる（ここではタプル間の順序関係から，B, A, C の順であるとわかったとする）．先ほど推定した $ptext_A = 55$ の前後の頻度分布と $ptext_B, ptext_C$ の頻度とを総合すると，両者には一致が見られ，結果集合 A の性質のみに基づいた場合よりも $ptext_A = 55$ という推定の信頼度は高まったといえる．これを随時繰り返すことで，攻撃者が持つ部分頻度分布に出現する値と暗号文とを，高い信頼度のもとに対応付けすることが可能となる．加えて，部分頻度分布に対応付けされた暗号文との前後関係や問合せ履歴から，分布の知識を持たなかった暗号文にまで推論攻撃を波及できる危険性も存在する．

2.4　バケット化

文献 [1] [2] では，検索対象となる sensitive 属性のドメインをタグによって識別される複数の部分タプル集合（バケット）に分割し，これらのタグを検索語として問合せに利用するバケット化（Bucketization）手法を提案している．バケットのタグは包含する値とは無関係に選ばれ，各バケット内では，暗号化タプルが互いに識別不能であることを保証する．利用者は属性値 a に対する問合せ用の属性値 a^* として a が属するバケットのタグを利用し，k 個以上の要素を含むバケット単位で取得することで，k-匿名性による問合せ結果の安全性を保証する．

一方，この方法では，問合せ結果中に含まれる偽陽性のフィルタリングコストと保証される安全性（匿名性）とがトレードオフの関係にある．また，更新の結果として匿名性が保証されないバケットが生じた場合には，分割の境界を再構築する必要がある．しかし，その際のタプルの移動を追跡することによって，識別不能であったタプル間の順序関係が徐々に漏洩してしまう可能性がある．

3.　準　　備

3.1　提案手法の概要

従来，暗号化データに対する効率的検索を目指す研究では，

索引付けのためタプル単位での順序関係を漏洩することを許容してきた．一方，こうしたタプル単位の順序関係情報は攻撃者にとって有用な知識となり，利用者が委託する機密データに対する脅威を著しく増大させる危険性がある．そこで我々は，OPES [3] や Lu の研究 [5] において許容されるタプル単位の情報漏洩を，ドメイン分割（バケット化）[1] [2] に類似する匿名化手法によってタプル集合単位まで粗くし，各タプルに対する推論攻撃のリスクを低減する索引手法を提案する．

この手法では，検索対象となるドメインを要素数と順序関係に基づき有限個のバケットに分割し，これらバケットを葉として暗号化 B+木を構築する．バケットに含まれる各タプルは互いに識別できない形で暗号化しており，かつ平文の値が同一である複数の暗号文を異なるバケットに分配可能とする索引のキー定義を与えることで，タプル単位での情報漏洩が起きないことを保証する．また，内部ノードが持つ暗号化キーの安全性，および分割に際してバケット内のタプルが追跡不能であることを保証し，偽陽性のフィルタリングコストと安全性を考慮した動的なバケット更新手法を与える．

3.2 信用範囲と攻撃モデル

本研究においては，図 3 のように表される DaaS モデル [1] [2] を想定する．このモデルは，データの所有者かつ検索者である 1 人の利用者（図 3 中の User と Client とを合わせて利用者と呼ぶ）とデータベースサービスの提供者である 1 台のサーバで構成される．ここで，サーバは外部からの侵入に対して脆弱な可能性があるため，委託された利用者の機密データに対して能動的に攻撃を仕掛ける可能性がある潜在的な攻撃者であると見なす．したがって利用者はサーバを完全には信用しておらず，情報は信頼できるクライアントを介し全て暗号化したうえでサーバに送信する．攻撃者はこれらの暗号化データを全て見渡すことができるが，利用者によって施された暗号化そのものを破ることは出来ない．また，利用者 - サーバ間の通信路は保護されており安全だが，利用者から送信されるメッセージは攻撃者にとって可視であるものとする．

攻撃者は，委託された暗号化データや利用者とのインタラクションから得られる情報，および事前知識を駆使し，出来る限り高い信頼度で各暗号文の真の値を推論しようとする．

4. タプル情報の漏洩を防止する匿名化索引

4.1 バケット分割

秘匿の対象となる属性のドメインを分割し，問合せの単位となるタプル集合（バケット）を生成する．この分割は，対象属性のドメインに出現する値の集合 V，各値 v $(\in V)$ の頻度 $f(v)$ の集合 F，および各バケットに含まれるタプル数の最小値と最大値を表すパラメータ b_{min}, b_{max} $(\in \mathbb{N}, b_{min} < b_{max})$ を入力として行う．

ここで，同じ属性値に対する暗号化タプルを必ず同一のバケットに格納すると定めると，分布や問合せ頻度に大きな偏りがある場合にバケット中の暗号化タプルの真値を高い信頼度で推定可能となってしまう危険がある．そこで，バケット内の属性値の偏りを緩和するための平滑化を，安全性に関するパラメータ s $(0 < s < 1)$ を入力として行う．具体的には，各バケット中での要素数が占める割合が s を超える属性値を偏りがあると判断し，平滑化の対象とする．

(a) Original Table

id	name	salary	addr
23	Tom	70K	Maple
860	Mary	60K	Main
320	John	50K	River
875	Jerry	55K	Hopewell

(b) Bucketized + Encrypted Table

etuple	X
1100110011110010...	31
1000000000011101...	7
1111101000010001...	59
1010101010111110...	59

図 4　変換前と変換後のテーブル例

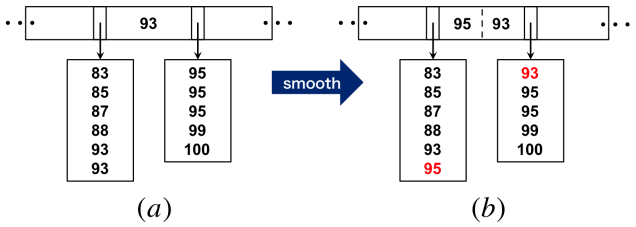

図 5　匿名化バケットと索引付け $(s = \frac{1}{2})$

また，これらのパラメータのみを用いた平滑化では，ある値がバケット間を次々と移動して飛び（値 v がバケット B_i, B_{i+2} に含まれるが B_{i+1} に含まれない等）が生じ，その値の問合せ結果中に全要素が偽陽性となるバケットが含まれてしまう可能性がある．このようなバケットの混入は，問合せ対象と同バケットに含まれ付随して得られる偽陽性に比べ，利用者によるフィルタリングコストを不必要に増大させてしまう．そこで，平滑化の対象とする属性値がどのバケットに含まれているかをあらかじめ把握し，飛びを生じさせる移動は許容しないという制約を付ける（以下，分配制約と呼ぶ）．これらの操作の基本的な流れを以下に示す．

（1） ドメインの最小値 V_{min} を始点とし，順に整数乱数 n $(b_{min} \leqq n \leqq b_{max})$ 個のタプルを 1 つのバケットとしてランダムなラベルを割り当てる．

（2） 各バケット B_i 中での要素数が占める割合が s を超える属性値 d を，順序関係における前後のバケット B_{i-1} 中の d を超えない最大値，もしくは B_{i+1} 中の d より大きい最小値と入れ替え，頻度を平滑化する．前後のバケットに交換可能な値が無ければ，b_{min}, b_{max}, および s の制約を破らない限りで d を分配する．

図 4 (a) は変換前のテーブル，(b) は属性 salary についてバケット分割による匿名化を行いレコード単位の暗号化を施したうえでサーバに送信されるテーブルである．また，図 5 (a) は上述の手順 1 で得られる初期分割後のバケット，(b) は手順 2 により平滑化を施したバケットである．

4.2 暗号化 B+木索引

先述のドメイン分割および頻度平滑化によって得られたバケットに対し，範囲述語暗号 (Range Predicate Encryption : RPE) を利用した暗号化 B+木による索引付けを施す．RPE は公開鍵暗号をベースとしたもの [21] と 共通鍵暗号をベースと

したもの [22] [5] [17] に大別されるが，ここでは，Lu の研究 [5] にて提案されている共通鍵方式による RPE を利用する．

以下に，文献 [5] において提案されている共通鍵 RPE の暗号化スキームを概観する．ここでは，安全性に関する証明は省略する．この構築法は，Shen らの研究 [23] で提案されている共通鍵内積述語暗号（SSW と略記する）をベースとしている．SSW の暗号化スキームは，以下の確率的多項式時間アルゴリズムで与えられる．ここで $\mathcal{G}$ は，1^k をセキュリティパラメータとし，組 $(p, q, r, s, \mathbb{G}, \mathbb{G}_T, e)$ を生成するアルゴリズムである．ただし p, q, r, s は 4 つの相異なる素数であり，$\mathbb{G}, \mathbb{G}_T$ はそれらの積 $N = pqrs$ を位数とする巡回群，e は $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ を満たす双線形写像である．

SSW_Setup(1^k): $\mathcal{G}(1^k)$ を実行し，組 $(p, q, r, s, \mathbb{G}, \mathbb{G}_T, e)$ を得る．ここで，位数 p, q, r, s に対応した部分群により，$\mathbb{G} = \mathbb{G}_p \times \mathbb{G}_q \times \mathbb{G}_r \times \mathbb{G}_s$ と表す．次に，$\mathbb{G}_p, \mathbb{G}_q, \mathbb{G}_r, \mathbb{G}_s$ から生成元 g_p, g_q, g_r, g_s をそれぞれ取り出す．また，$h_{1,i}, h_{2,i}, u_{1,i}, u_{2,i} \in \mathbb{G}_p$ を各 i $(= 1, 2, \cdots, n)$ に対してランダムに選ぶ．秘密鍵 SK は，以下のようになる．

$$SK = (g_p, g_q, g_r, g_s, \{h_{1,i}, h_{2,i}, u_{1,i}, u_{2,i}\}_{i=1}^n).$$

SSW_Encrypt$(SK, \overrightarrow{x})$: $\overrightarrow{x} = (x_1, \cdots, x_n) \in \mathbb{Z}_N^n$ とする．$y, z, \alpha, \beta \in \mathbb{Z}_N$，$S, S_0 \in \mathbb{G}_s$ をランダムに選び，$R_{1,i}, R_{2,i} \in \mathbb{G}_r$ を各 i $(= 1, 2, \cdots, n)$ に対しランダムに選ぶ．暗号文 $CT_{\overrightarrow{x}}$ は，以下のようになる．

$$CT = \begin{pmatrix} C = S \cdot g_p^y, \\ C_0 = S_0 \cdot g_p^z, \\ \{C_{1,i} = h_{1,i}^y \cdot u_{1,i}^z \cdot g_q^{\alpha x_i} \cdot R_{1,i}, \\ C_{2,i} = h_{2,i}^y \cdot u_{2,i}^z \cdot g^{\beta x_i} \cdot R_{2,i}\}_{i=1}^n \end{pmatrix}.$$

SSW_GenToken$(SK, \overrightarrow{v})$: $\overrightarrow{v} = (v_1, \cdots, v_n) \in \mathbb{Z}_N^n$ とする．$f_1, f_2 \in \mathbb{Z}_N$，$R, R_0 \in \mathbb{G}_r$ をランダムに選び，$r_{1,i}, r_{2,i} \in \mathbb{Z}_N$，$S_{1,i}, S_{2,i} \in \mathbb{G}_s$ を各 i $(= 1, 2, \cdots, n)$ に対しランダムに選ぶ．検索トークン $TK_{\overrightarrow{v}}$ は，以下のようになる．

$$TK_{\overrightarrow{v}} = \begin{pmatrix} K = R \cdot \prod_{i=1}^n h_{1,i}^{-r_{1,i}} \cdot h_{2,i}^{r_{2,i}}, \\ K_0 = R_0 \cdot \prod_{i=1}^n u_{1,i}^{-r_{1,i}} \cdot u_{2,i}^{-r_{2,i}}, \\ \{K_{1,i} = g_p^{r_{1,i}} \cdot g_q^{f_1 \, v_i} \cdot S_{1,i}, \\ K_{2,i} = g_p^{r_{2,i}} \cdot g_q^{f_2 \, v_i} \cdot S_{2,i}\}_{i=1}^n \end{pmatrix}.$$

SSW_Query$(TK_{\overrightarrow{v}}, CT_{\overrightarrow{x}})$: $CT_{\overrightarrow{x}} = (C, C_0, \{C_{1,i}, C_{2,i}\}_{i=1}^n)$，$SK_{\overrightarrow{v}} = (K, K_0, \{K_{1,i}, K_{2,i}\}_{i=1}^n)$ と表記する．SSW_Query は，以下の等式が成立するとき，すなわち条件 $\langle \overrightarrow{x}, \overrightarrow{v} \rangle = 0 \bmod N$ を満足するときだけ “1” を出力し，それ以外では “0” を出力する．

$$e(C, K) \cdot e(C_0, K_0) \cdot \prod_{i=1}^n e(C_{1,i}, K_{1,i}) \cdot e(C_{2,i}, K_{2,i}) \stackrel{?}{=} 1.$$

次に，SSW をベースとした RPE の暗号化スキームを以下に示す．文献 [5] ではセグメント木を利用し効率化した構築手法を提案しているが，ここでは基本的な手法のみに言及する．

RPE_Setup$(1^k, [0, T-1])$: セキュリティパラメータ 1^k および範囲 $[0, T-1]$ を入力とし，**SSW_Setup**(1^k) によって秘密鍵 SK を生成，出力する．

RPE_Encrypt(SK, t): 点 t を入力とし，$i = t$ のとき $x_i = 1$，$i \neq t$ のとき $x_i = 0$ としてベクトル $\overrightarrow{x} = (x_0, \cdots, x_i, \cdots, x_{T-1})$ を生成する．次に，**SSW_Encrypt**$(SK, \overrightarrow{x})$ により暗号文 $CT_{\overrightarrow{x}}$ を生成し，出力する．

RPE_GenToken(SK, Q): 範囲クエリ Q を入力とし，$i \in Q$ のとき $v_i = 0$，$i \notin Q$ のとき $v_i = 1$ としてベクトル $\overrightarrow{v} = (v_0, \cdots, v_i, \cdots, v_{T-1})$ を生成し，**SSW_GetToken**$(SK, \overrightarrow{v})$ により検索トークン $TK_{\overrightarrow{v}}$ を生成，出力する．

RPE_Query$(TK_{\overrightarrow{v}}, CT_{\overrightarrow{x}})$: **SSW_Query**$(TK, CT_{\overrightarrow{x}})$ の結果を出力する．

暗号化 B+木の各内部ノードに格納する RPE キーに反映する点情報は，葉ノードにあたるバケットの内容からボトムアップに決定する．ある内部ノード I の 2 つの子ノード B_i, B_{i+1} が図 5 (a) のように重複範囲を持つ場合，その I を二重化して I_L, I_R とし，I_L に B_{max}^i，I_R に B_{min}^{i+1} (B_{min}^{i+1}：バケット B_{i+1} の最小値, B_{max}^i：バケット B_i の最大値) を反映した RPE をキーとして格納する．一方，図 5 (b) のように子ノードに重複範囲が無い場合，その I は二重化せず，B_{max}^i (B_{max}^i：バケット B_i の最大値) を反映した RPE をキーとして格納する．

ここでは簡単のため，Lu らの研究 [5] と同様に，検索可能な数値属性が暗号化テーブル上に 1 つだけしか存在しないものとする．多次元範囲検索への拡張については，今後の課題とする．初期動作として利用者はデータの暗号化，バケット分割，および暗号化 B+木による索引付けを自身の秘密鍵を用いて行った後，それらをサーバに送信する．

4.3 問 合 せ

以下に，提案手法における範囲問合せの流れを示す．一致問合せは範囲問合せの範囲の最小値と最大値が一致する特殊型と見なすことができ，同様の手続きによって実行可能である．ただし，検索対象のドメインを $[D_{min}, D_{max}]$ とし，秘密鍵 SK を **RPE_Setup**$(1^k, [D_{min}, D_{max}])$ によって予め生成しているとする．

（1） 利用者は平文の範囲検索クエリ $Q = [Q_{min}, Q_{max}]$ を生成する．

（2） 利用者は **RPE_GenToken**$(SK, [D_{min}, q_{min} - 1])$ によって検索トークン tk_L，**RPE_GenToken**$(SK, [q_{max} + 1, D_{max}])$ によって検索トークン tk_R をそれぞれ生成し，サーバに送信する．

（3） サーバは tk_L をもとに，暗号化 B+木の各内部ノードキー k_i $(i = 1, \cdots, m)$ を昇順に走査する．内部ノード I が二重化されている場合，I_L のキーと照合を行う．

RPE_Query(tk_L, k_i) が “1” となる場合，枝 b_i が指すノードに進む．**RPE_Query**(tk_L, k_m) も “1” にならない場合，枝 b_{m+1} が指すノードに進む．

（4） サーバは tk_R をもとに，暗号化 B+木の各内部ノードキー k_i $(i = 1, \cdots, m)$ を降順に走査する．内部ノード I が二重化されている場合，I_R のキーと照合を行う．**RPE_Query**(tk_R, k_i) が “1” となる場合，枝 b_{i+1} が指すノードに進む．**RPE_Query**(tk_R, k_1) も “1” にならない場合，(3) で得られた枝が指すノードに進む．

（5） (3), (4) の走査が葉ノードのバケット B_{min}, B_{max} に達したら，バケット B_n $(n = min, min+1, \cdots, max)$ を全て取得し結果として利用者に返す．

（6） 利用者は問合せ結果を受け取り，自身の秘密鍵で中身を復号して偽陽性をフィルタリングする．

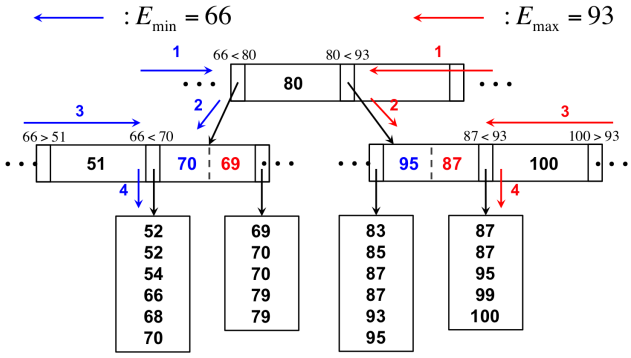

図 6　範囲問合せの流れ $(s = \frac{1}{2})$

4.4　バケットの更新

タプルの追加（削除）に伴い，匿名化バケットを分割（マージ）することを考える．以下に，利用者があるタプルを追加（削除）する際の手続きを示す．

タプルの追加，バケットの分割

（1） 利用者は追加したいタプルの値を反映した述語暗号 E を自身の秘密鍵を用いて生成し，サーバに送信する．

（2） サーバは E をもとに該当するバケット（もしくはバケット群） B を探索し，利用者に返す．

（3） 利用者は B を受け取り，対象タプルを追加する．ここで，要素の偏りが s を超える場合は (4)，バケットの要素数が上限を超える場合は (5)，に進み，そうでなければ (7) に進む．

（4） 利用者は，B 中の最小値および最大値に基づいて B の前後の隣接するバケット B_{-1}, B_{+1} をサーバから取得し（※ 片側しか存在しない場合は一方），B とマージする．

（5） 利用者は B に含まれる最小値，最大値をそれぞれ反映した述語暗号 E_{min}, E_{max} を利用し，サーバから B 中の要素に関連する索引のノードを取得する（※ バケット分割における分配制約の判定に利用する）．

（6） 利用者は B 中のタプルをソートし，4.1 項に示す手法で分割・ラベル付けを行うことでバケット B_1, B_2 を得る．この際，全てのタプルは摂動を加えて暗号化しなおす．

（7） 利用者は更新後のバケット中の最大値を反映した述語暗号 E' を自身の秘密鍵で生成し，更新情報と共にサーバに送信する．

（8） サーバは受け取った情報をもとに暗号化 B+木を更新する．

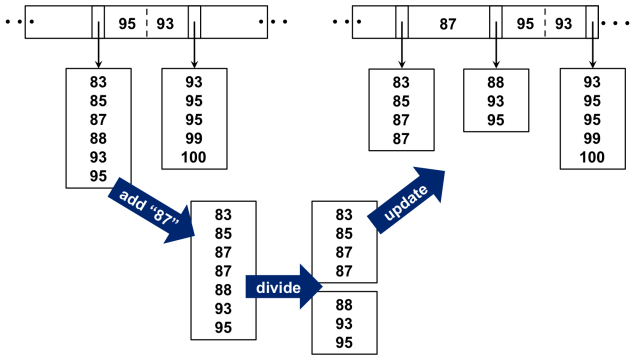

図 7　バケットの分割 $(s = \frac{1}{2})$

タプルの削除，バケットのマージ

（1） 利用者は削除したいタプルの値を反映した述語暗号 E を自身の秘密鍵を用いて生成し，サーバに送信する．

（2） サーバは E をもとに該当するバケット（もしくはバケット群） B を探索し，利用者に返す．

（3） 利用者は B を受け取り，対象タプルを削除する．ここで，バケットの要素数が下限を下回る場合は (4) に進み，そうでなければ (6) に進む．

（4） 利用者は，B 中の最小値および最大値に基づいて B の前後の隣接するバケット B_{-1}, B_{+1} をサーバから取得し（※ 片側しか存在しない場合は一方），B とマージする．

（5） B_{-1}, B, B_{+1} 中のタプルをソートし，4.1 項に示す手法で分割・ラベル付けを行うことでバケット B_1, B_2 を得る．

（6） 利用者は更新後のバケット中の最大値を反映した述語暗号 E' を自身の秘密鍵で生成し，更新情報と共にサーバに送信する．

（7） サーバは受け取った情報をもとに暗号化 B+木を更新する．

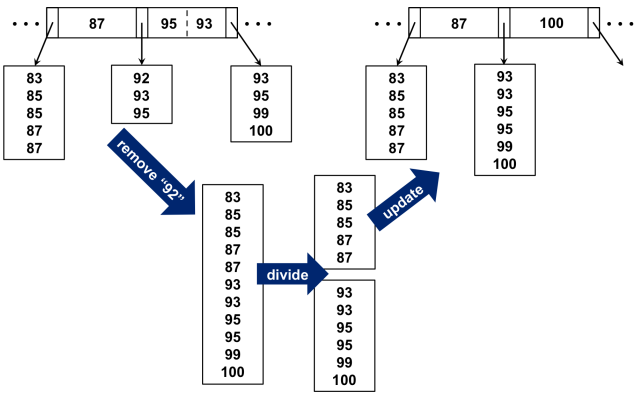

図 8　バケットのマージ $(s = \frac{1}{2})$

本手法が利用するバケット化は，偽陽性を多く含むタプル集

合を結果として返すとフィルタリングコストが高くなり，偽陽性の少ないタプル集合を返すと安全性が低下するというトレードオフを抱えている．そこで，4.1 項にて述べたように，各バケットが保持できるタプル数を制限するパラメータを与え，要素数をベースとしたバケット分割およびマージを行うことで安全性とフィルタリングコストのバランスを取る．一方，各バケットの中に含まれる暗号化タプルがサーバにとって可視であることを考慮すると，このようなバケット分割に伴い，当初は匿名化されていたタプル単位の順序関係が徐々に漏洩してしまう可能性がある．したがって，バケット分割の際には包含される全てのタプルを摂動を加えた上で再暗号化し，分割前後で追跡不能とすることでタプル単位での順序関係漏洩を防ぐ．

索引の更新

暗号化 B+木索引については，以下の **IDX** 命令セットを介してサーバ側で更新する．

IDX_Insert$(TK_{\overrightarrow{v}}, CT_{\overrightarrow{x}})$: 挿入したい索引キー k から **RPE_Encrypt**(SK, k)，**RPE_GenToken**$(SK, [k])$ によって利用者が生成した暗号文 $CT_{\overrightarrow{x}}$，トークン $TK_{\overrightarrow{v}}$ を受け取り，トークンによって指定された葉ノードの該当位置（二重化されたノードのキーを用いる場合，その最小値，最大値それぞれを用いて位置を定める）にキーを挿入する．ノードに格納できるキーの最大数を超えた場合，葉ノードを分割し，中央のキーを親ノードに挿入する．以後，分割と挿入を再帰的に行う．

IDX_Delete$(TK_{\overrightarrow{v}}, CT_{\overrightarrow{x}})$: 削除したい索引キー k から，**RPE_Encrypt**(SK, k)，**RPE_GenToken**$(SK, [k])$ によって利用者が生成した暗号文 $CT_{\overrightarrow{x}}$，トークン $TK_{\overrightarrow{v}}$ を受け取り，トークンによって該当する葉ノードを発見する．この際，通過した内部ノードキーの位置を記憶しておく．続いて葉ノード中の該当位置（二重化されたノードのキーを用いる場合，その最小値，最大値それぞれを用いて位置を定める）のキーを削除するが，ノードに格納できるキーの最小数を下回った場合，通過した内部ノードのキーをもとにノードをマージし，キーを置換・削除する．

IDX_Replace$(TK_{\overrightarrow{v}}, CT_{\overrightarrow{x}})$: 置換対象とする索引キー k_1 から **RPE_GenToken**$(SK, [k_1])$ によって利用者が生成したトークン $TK_{\overrightarrow{v}}$，置換する新キー k_2 から **RPE_Encrypt**(SK, k_2)，によって利用者が生成した暗号文 $CT_{\overrightarrow{x}}$ を受け取り，トークンによって指定された葉ノードの該当位置のキーを置換する．

以下に，この命令セットを用いた索引の更新動作の例を示す．

例 3: 図 9 に示すような索引に，値 85 を持つタプルを追加する際の動作を考える．ここで，各バケットは最小 3 個，最大 6 個の要素を持つことができるとする．

利用者は値 85 を反映した述語暗号を自身の秘密鍵で生成し，サーバに送信する．サーバは与えられたトークンをもとにバケット $B_1 = \{83, 85, 87, 87, 93, 95\}$ を発見し，これを利用者に返す．利用者はバケット B_1 にタプルを追加するが，要素数が 7 となり最大数を超過してしまう．そこで，利用者は B_1 中のタプルを一旦全て復号し，4.1 項の手続きにより分割されたバケット $B_{11} = \{83, 85, 87\}, B_{12} = \{85, 87, 93, 95\}$ を得る．この際，バケット内のタプルは全て摂動を加えて再暗号化する．

この分割によって B_1 を指すポインタに対応したキー **RPE_Encrypt**$(SK, [93, 95])$ (SK : 利用者の秘密鍵) を更新する必要が生じるため，利用者は B_{11}と B_{12} に付加したラベルをテーブル内の各タプルに反映させると共に，**RPE_Encrypt**$(SK, [85, 87])$ を **IDX_Insert** によって挿入する．この例では，挿入先の内部ノードが格納するキー数が最大値を超えたとし，キー **RPE_Encrypt**$(SK, 120)$ が平衡化のため移動する．

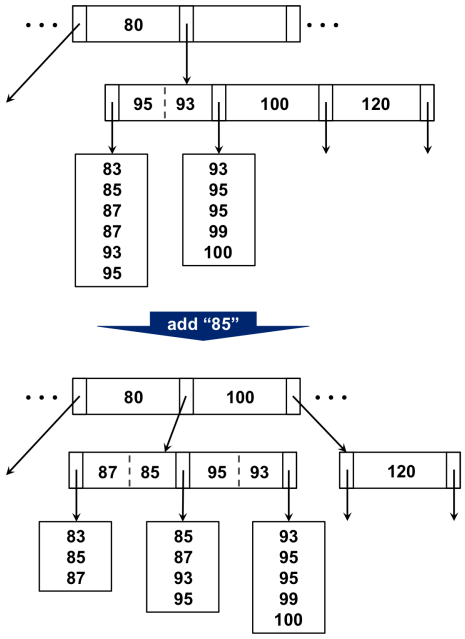

図 9 索引の更新 ($s = \frac{1}{2}$)

5. まとめと今後の課題

従来の検索効率を重視するプライバシ保護型検索手法では，暗号化データに対する効率的な範囲検索をサポートすることに主眼を置き，タプル単位での順序関係を漏洩することを許容してきた．しかし，こうした順序関係情報の漏洩は完全な知識を持たない攻撃者に対しても推論攻撃の糸口を与える危険性がある．そこで我々は，タプル集合単位でしか順序関係情報を漏洩しない暗号化索引により，大きなオーバーヘッドを課すこと無く各タプルに対する推論攻撃のリスクを低減する手法を提案した．

今後の喫緊の課題として，提案手法の安全性や効率に関する評価実験，およびバケット内の偏りがどの程度許容できるのかに関する検討を行う予定である．また，本論文では，1 人の利用者と 1 台のサーバを想定し議論してきたが，DaaS の現実的なモデルとしては複数の利用者と複数のサーバを想定する必要があると考えられる．複数の利用者を想定する場合には，提案

手法の根幹である述語暗号化に用いる秘密鍵の運用が問題となる．これについては，Lu らの研究 [5] のように単一の信頼できる管理者を想定し，各利用者はその管理者を介してデータの暗号化や検索用トークンの発行を行うようモデルを修正することが考えられる．この他に，岡本らが提案している階層的述語暗号 (Hierarchical Predicate Encryption : HPE) [24] を用いた権限委譲が利用できると考えているが，これらの拡張は今後の課題である．一方，複数のサーバが存在する環境においても提案手法はそのまま適用できるのではないかと考えているが，これについても詳細な検討が必要である．

また，提案手法は複数の（秘匿の対象となる）検索可能属性を有する暗号化リレーショナルデータベースに適用した場合，バケット分割・マージに際してタプルの追跡不能性を保証することができない．これについては，空間充填曲線および多次元空間分割手法を導入し，複数の検索可能属性を単一属性として表現することで解決できると考えている．一方，こうした多次元空間分割手法を導入すると，提案手法を 1 次元データにおいて運用した場合に比べ利用者側でフィルタリングすべき偽陽性が大きく増加する可能性が高い．この点についての検証，および手法の修正も今後の課題である．

謝　　辞

本論文の執筆にあたり，筑波大学大学院 システム情報工学研究科の川本 淳平氏より多くの有益な助言を頂きました．心より感謝申し上げます．

文　　献

[1] H. Hacıgümüş, B. Iyer, C. Li, and S. Mehrotra. Executing SQL over encrypted data in the database-service-provider model. *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, 2002.

[2] B. Hore, S. Mehrotra, and G. Tsudik. A Privacy-Preserving Index for Range Queries. *Proceedings of 30th international conference on Very large data bases*, Vol. 30, pp. 720–731, 2004.

[3] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Order Preserving Encryption for Numeric Data. *SIGMOD '04 Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pp. 563–574, 2004.

[4] H. Kadhem, T. Amagasa, and H. Kitagawa. MV-OPES: Multivalued-Order Preserving Encryption Scheme: A Novel Scheme for Encrypting Integer Value to Many Different Values. *IEICE TRANSACTIONS on Information and Systems*, Vol. E93-D, No. 9, pp. 2520–2533, 2010.

[5] Y. Lu. Privacy-Preserving Logarithmic-time Search on Encrypted Data in Cloud. *In 19th Annual Network and Distributed System Security Symposium (NDSS Symposium'12)*, 2012.

[6] A. Shamir. How to share a secret. *Communications of the ACM*, Vol. 22, No. 11, pp. 612–613, 1979.

[7] G. R. Blakley and C. Meadows. Security of ramp schemes. *In Proceedings of CRYPTO'88, LNCS403*, pp. 252–268, 1990.

[8] NRI セキュアテクノロジーズ株式会社. SecureCube Secret Share, 2010. `http://www.nri-secure.co.jp/service/cube/secretshare.html`.

[9] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. *In Proceedings of EUROCRYPT'04*, Vol. LNCS 3027, pp. 506–522, 2004.

[10] D. X. Song, D. Wagner, and A. Perrig. Practical techniques for searches on encrypted data. *Proceedings of the 2000 IEEE Symposium on Security and Privacy*, pp. 44–, 2000.

[11] M. Bellare, A. Boldyreva, and A. ONeill. Deterministic and efficiently searchable encryption. *CRYPTO'07 Proceedings of the 27th annual international cryptology conference on Advances in cryptology*, pp. 535–552, 2007.

[12] B. Chor, O. Goldreich, E. Kusheilevitz, and M. Sudan. Private Information Retrieval. *J. ACM*, Vol. 45, No. 6, pp. 965–982, 1998.

[13] C. Cachin, S. Micali, and M. Stadler. Computationally private information retrieval with polylogarithmic communication. *EUROCRYPT'99 Proceedings of the 17th international conference on Theory and application of cryptographic techniques*, pp. 402–414, 1999.

[14] S. Wang, D. Agrawal, and A.E. Abbadi. Generalizing PIR for Practical Private Retrieval of Public Data. *Proceedings of the 24th annual IFIP WG 11.3 working conference on Data and applications security and privacy*, pp. 1–16, 2010.

[15] S. Kaneko and C. Watanabe. Semi-Shuffled BF : Performance Improvement of a Privacy-Preserving Query Method for a DaaS Model using a Bloom Filter. *The 2011 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA '11)*, 2011.

[16] E. Goh. Secure indexes, 2004. `http://eprint.iacr.org/2003/216/`.

[17] J. Kawamoto and M. Yoshikawa. Private Range Query by Perturbation and Matrix Based Encryption. *In 6th International Conference on Digital Information Management (ICDIM 2011)*, pp. 211–216, September 2011.

[18] J. Kawamoto. A Locality Sensitive Hashing Filter for Encrypted Vector Databases. *Proceedings of International Workshop on Privacy-AwaRe Intelligent Systems (PARIS 2012)*, December 2012.

[19] A. Boldyreva, N. Chenette, and A. O'Neill. Order-Preserving Encryption Revisited: Improved Security Analysis and Alternative Solutions. *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference*, Vol. 6841, p. 575, 2011.

[20] A. Boldyreva, N. Chenette, Y. Lee, and A. O'Neill. Order-Preserving Symmetric Encryption. *EUROCRYPT '09 Proceedings of the 28th Annual International Conference on Advances in Cryptology: the Theory and Applications of Cryptographic Techniques*, pp. 224–241, 2009.

[21] D. Boneh and B. Waters. Conjunctive, subset, and range queries on encrypted data. *Proceedings of the 4th conference on Theory of Cryptography*, pp. 535–554, 2007.

[22] C. Blundo, V. Lovino, and G. Persiano. Private-key hidden vector encryption with key confidentiality. *Cryptology and Network Security Lecture Notes in Computer Science*, Vol. 5888, pp. 259–277, 2009.

[23] E. Shen, E. Shi, and B. Waters. Predicate Privacy in Encryption Systems. *In Theory of Cryptography*, Vol. 5444, pp. 457–473, 2009.

[24] T. Okamoto and K. Takashima. Hierarchical Predicate Encryption for Inner-Products. *In 15th International Conference on the Theory and Application of Cryptology and Information Security*, pp. 214–231, 2009.