階層的なアクセスレベル制御を行う RDF データに対する集約計算結果の推定

三上　英明†　　杉原　弘祐†　　横田　治夫†

† 東京工業大学大学院情報理工学研究科計算工学専攻　〒 152-8552 東京都目黒区大岡山 2-12-1
E-mail: †{mikami,sugihara}@de.cs.titech.ac.jp, ††yokota@cs.titech.ac.jp

あらまし　被災地における救援物資提供のための対象人数調査など, 個人情報を含むデータの集計が必要な場合がある. 従来は, プライバシーに関わる情報の特定を防ぐため, 属性単位等のアクセス範囲に対する集計値を返さないようにしたり乱数を付加して返したりする方法がとられていた. 本研究では, より細かいアクセス制御を行うために, RDF によるグラフデータベースを想定し, グラフのノード単位で, 情報の持ち主の意向を取り入れた階層的なアクセスレベル制御を行い, ユーザのクラスに応じた集約計算を行うことを目指す. プライバシーに関わる情報が特定できないように配慮しながら, それぞれのユーザクラスがアクセス可能なノード数の割合を利用し集計値を推定する. 被災情報に関する RDF ベンチマークを用いて, 集計値の精度の評価を行う.

キーワード　情報セキュリティ, プライバシー, RDF

1. はじめに

近年, 災害発生時や発生後の被災地における復旧活動において情報技術を活用することが注目され, 安否確認を含む情報共有システムが利用されている. 実現されたシステムとしては, 利用者の安否回答の状況を地図上に表示し, 家族や同僚, 友人同士でその情報を確認できるアプリケーションなどが挙げられる. このようなシステムで利用するデータの中には, 個人のプライバシーに関係するものが含まれる可能性があるため, システムの利用にあたっては不正利用を防止するためのアクセス制御やデータの暗号化が必要不可欠である. しかしながら, 必要があればそのデータを個人のプライバシーを侵害することがないように提供することも同時に必要である.

例えば, 被災者支援団体が救援物資の提供のために支援対象である被災者の総数を知りたいという場合, 被災者の情報を保存したデータベース上でユーザが指定した条件を満たす被災者数の集計計算を行いユーザに提供するシステムが必要になる. このとき, 悪意を持つユーザがシステムに集計計算を要求し, 集計計算の検索条件と集計計算結果からプライバシーに関わる情報を特定する危険があるため, その危険に対処しなければならない.

既存手法では集計計算結果からプライバシーにかかわる情報を特定可能と考えられる場合に集計計算結果を返さないようにしたり, 雑音として乱数を加算したりするといった方法がとられていた. しかし, これらの手法は得られる集計計算結果が確実でないことや, ユーザがアクセス可能なデータ数を考慮しておらず, アクセス可能なデータ数の多いユーザがより正確な集計計算結果を得られるようにすることができないことなどの問題点がある.

そこで本論文では, アクセス制御が行われた暗号化データベースに対してユーザのアクセス可能なデータ数に応じた精度で確実な集計値を提供するために, ユーザがアクセス可能な部分の集計値から全体の集計値を実用に堪える速度で推定する手法を提案する. 全てのデータにアクセス可能なアクセス権限をもつユーザには全体の集計値と誤差のない集計値が提供されるが, アクセス権限が低くアクセス可能なデータが少ないユーザにはアクセス不可能な部分の値を特定できないような全体の集計値との誤差が大きい集計値が提供されることを示す. また, RDF を利用した, 個々のデータに階層的なアクセスレベルを設定することが可能な暗号化データベース上で提案手法を実装し, 被災地の避難者情報を再現したベンチマークを用いて性能を評価する.

本論文は, 本章を含めて 6 章により構成される. 第 2 章では本論文に関して前提となる知識を述べる. 第 3 章では関連する既存研究を紹介する. 第 4 章では提案する手法について述べる. 第 5 章で評価実験の方法を述べ, その結果を考察する. 最後に, 第 6 章で本論文のまとめと今後の課題を述べる.

2. 前提知識

2.1 アクセス制御

アクセス制御とは, ある情報資源に対し, 特別に認可された利用者のみがアクセスできるようにすることである [1]. アクセス制御ポリシーとは, どの利用者がどの情報資源にアクセスできるようにするかを判断する基準である. システムは, ユーザによるアクセス要求があると, まず認証により登録されたユーザであるかを確認し, 登録されたユーザであればそのユーザ情報を取得する. そして, システムはユーザの属性・ユーザがアクセスしようとしている情報資源の属性・ユーザが要求するアクセスの種類などの情報をアクセス制御ポリシーと照らし合わせ, 最終的にユーザのアクセス要求に対し認可を与えるかどうかの判断を行う. このような仕組みにより, システムは情報資源に対し管理者の意図しないアクセスが行われないことを保証できる.

ただし, 情報資源内の情報をその情報の所有者が意図しないアクセスから守るには, このようなアクセス制御だけでは不十分である. 情報資源の管理者がシステムのアクセス制御設定を不正に書き換え, 情報資源に許可なくアクセスする場合があるからである. このような場合に備えて, 公開範囲を制限すべき情報については暗号化による保護も必要である. 所有者や許可された利用者のみが保持する鍵で個々のデータを暗号化することにより, たとえ管理者が情報資源に許可なくアクセスして内部のデータを取得しようとしても, それを利用するために復号することができないからである.

このような暗号化による情報資源の保護技術の応用として, Popa らが提案した手法 [2] がある. この手法では, ユーザとデータベースサーバの間に CryptDB と呼ばれるアプリケーションが設けられ, プロキシとして暗号や復号に関わる処理を行う. CryptDB は, データを暗号化したまま演算が可能な暗号化手法を利用し, それらの暗号化をデータに対して複数回行い, 最後にどの演算も不可能な暗号化をデータに施す. ユーザが検索要求を出すと, そのクエリで計算が必要になるデータベース内の属性列の値を, 対応する演算が可能な暗号化を施された状態になるまで復号してから処理を行う. CryptDB はこのような仕組みを利用しているので, 検索時にデータベースの内容を全て復号化する必要がないため, データベースの管理者が暗号化前のデータを取得する危険を最小限に抑えることが可能である.

2.2 RDF

情報共有においては情報への不正なアクセスの防止の他に, 情報をどのように表現し保存するかということも重要である. 特に災害時は避難所情報を速やかに被災者へ提供することが重要であり, そのためには自動処理プログラムにより二次利用のための編集や加工などが可能な形式で避難所情報を表現する必要がある. 政府はこのようなデータ形式での情報公開を行っており [3], これはオープンデータと呼ばれる. これを実現するデータ表現のモデルとして RDF(Resource Description Framework) [4] が注目されている.

RDF は, 情報を 3 つの要素からなるトリプル単位で表現する. トリプルはある事物を指す識別子である主語と, 他の事物の識別子あるいは主語の持つ属性値を表すリテラル値である目的語と, 主語と目的語の関係を示す有向辺である述語で構成される. 例えば, Alice さんという人物に対し一意な識別子が与えられ, その省略形が ex:Alice と表現できたとする. Alice が 24 歳であるという情報は図 1 に示すトリプルとして表現される. トリプルの集合も RDF グラフとよばれ, 同様に可視化できる.

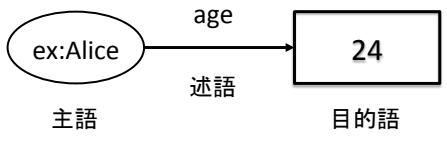

図 1　RDF トリプル (ex:Alice, age, 24)

RDF グラフからデータを検索するための言語としては SPARQL が標準的に用いられている [5]. SPARQL クエリは通常の SQL と同様に SELECT 節, WHERE 節を持つ. SPARQL クエリによるデータの検索は, WHERE 節に記述されるグラフパターンとのマッチングにより行われる. 以下に SPARQL クエリの一例を示す.

```
SELECT ?age WHERE
{
  ex:Alice age ?age
}
```

グラフパターンはトリプルパターンの集合であり, トリプルパターンの主語, 述語, 目的語部分には RDF グラフにおいてそれらになりうる要素に加えて, どの値にもマッチする変数を置くことができる. 変数は頭文字に?をつけることで表現する. また, 同じ変数が置かれた要素については同じ値であることを示す. クエリ結果は WHERE 節にマッチするグラフパターンの中で SELECT 節に書かれた変数にマッチした要素となる. SELECT 節では変数そのものだけでなく COUNT などの集計計算を行った結果を取得するようにすることも可能である. 例に示したクエリは ex:Alice に対応する人の年齢を取得するクエリである. 図 1 のグラフに対してこのクエリを発行した場合, クエリ結果は {{?age =: 24 }} となる.

RDF でデータを記述すると, SPRAQL クエリなどを用いて他の RDF で記述されたデータも対象にした検索や集計を行うことが可能である. この性質から, RDF は日本を含む世界において相互にリンクするオープンデータ (Linked Open Data [4]) を実現する手段として普及し始めている. 特に日本においては東日本大震災以降避難所情報を RDF 形式で公開する自治体が出てきており [6], 災害時の情報共有に用いる情報の表現に RDF を用いることが今後標準になっていくのではないかと考えられる. したがって, 災害時の情報共有に関係している本研究において情報表現の手段として最も適していると考えられるため, 本研究においては使用するデータを RDF で表現する.

3. 関 連 研 究

3.1 プロキシ再暗号化を用いた暗号化データ検索

児玉らはデータやユーザの効率的な追加および削除を効率的に行うために, データに対し論理的なクラスを単位とした暗号化を行う手法を提案した [7]. この手法は, プロキシ再暗号化可能な暗号を利用することにより, アクセス制御の柔軟性を維持しながら, 従来の暗号化手法と比較して少ない計算量でデータやユーザの追加および削除を行うことを可能にしている. また, 2.2 で説明したような理由から, データを表現する形式としては RDF を用いている. この手法を用いて実装された暗号化データ検索システムは, 暗号化されたデータを保存するデータベースと以下の要素を組み合せている.

アプリケーション　ユーザが直接通信するアプリケーション

プロキシサーバ　暗号化されたデータが保存されたデータベースとアプリケーションの間にあり, データを再暗号化する能力をもつ

アクセス制御情報　ユーザがどのクラスに属するか・どのクラ

スがどのアクセスレベルに対応するかという情報と, アクセスレベル間の階層関係が保存される

秘密鍵生成局　アクセスレベルの公開鍵, ユーザの公開鍵・秘密鍵およびアクセスレベルからユーザへ再暗号化のために使用する鍵を提供する

これらの要素を組み合わせて暗号化データベースで検索を行う手順は以下のとおりである. また, 各要素間の連携を図 2 に示した.

（1） アプリケーションはユーザの認証を行うと, ユーザのクエリを受け付け, ユーザの id に紐づく公開鍵でクエリに含まれる RDF タームを暗号化し, ユーザの id とともにプロキシサーバに送信する

（2） プロキシサーバはユーザの id からユーザのクラスと対応付けられたアクセスレベル l を取得し, l およびそれ以下のアクセスレベルが設定されそのレベルで暗号化されたデータを検索可能なようにクエリの RDF タームを再暗号化しデータベースに送信する

（3） プロキシサーバはデータベースにおける検索結果を受け取ると, 結果に含まれる暗号文をアクセスレベルからユーザの id に紐づくように再暗号化してアプリケーションに送信する

（4） ユーザはアプリケーションから結果を受け取ると, ユーザの秘密鍵で結果を復号化する

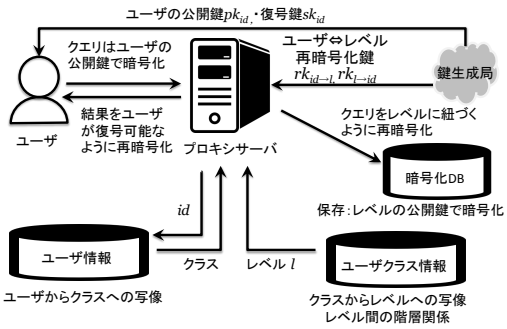

図 2　児玉らの手法

3.2　プライバシー保護データマイニング

個人の行動に関連した実社会情報を扱う広告モデルや SNS などのオンラインサービスが登場している近年では, サービスの提供者が個人のプライバシーに関わる情報を利用できないことは経済的な機会損失につながるため, そのような情報の利用は必要不可欠となっているが, 利用に際しては所有者が望まない流出が起こることもある. プライバシー保護データマイニング (以下 PPDM) とは, 個人のプライバシーに関わる情報を含むデータ群から個人のプライバシーが流出しないように保護しつつ有用な情報を抽出するための技術の総称である.

PPDM に関する研究は大きく分けて次の 3 つに分類される [8].

入力プライバシー　1 つ 1 つのデータをある個人のものと特定できないように公開する

出力プライバシー　データ群に集計計算などのマイニング処理を施した結果を, 個別のデータの値が特定できないように公開する

（狭義の）PPDM　所有者の異なる複数のデータ群に対する計算結果を, 各所有者に対し他の所有者の所有するデータの値を特定できないないように計算し公開する

入力プライバシーに関する技術としては, 個々のデータのもつ値が特定の範囲内であることまではわかるようにしながらも元の値がわからないようにする手法 (匿名化) や, 各データ内で個人を特定しうる属性を隠す手法 (抑圧) などが挙げられる. 出力プライバシーに関する技術としては, 集計計算結果から個々のデータのもつ値を特定できないように集計計算結果を歪ませる手法 (摂動法) や, 同一のユーザから類似した集計計算クエリが連続して送信された場合に集計計算結果を返さないようにする手法 (クエリ監査) などが挙げられる. 摂動法としては雑音として乱数を加算する手法などがある.

摂動法により歪められた集計値の安全性を評価する基準として, Dwork は差分プライバシーという概念を提唱した [9]. これは, データベースを入力とし歪められた集計計算結果を出力とするメカニズムを f とし, あるデータベース D とその中にある任意の個人のデータ t に対して, D から t のみを除外したデータベース D' を考えたときに, $f(D)$ と $f(D')$ の区別がほとんどつかなければ f は安全であるとみなすものである. 差分プライバシーを満たすメカニズムとしては, 集計計算結果にラプラス分布 $L(\frac{\Delta}{\varepsilon}) = \frac{\varepsilon}{2\Delta} e^{-\frac{\varepsilon|x|}{\Delta}}$ ($\Delta = \max_{D,D'} |f(D) - f(D')|$) にしたがう乱数を加算するものなどが同氏らの研究により発見されている [10].

これらの動きに加えて, 近年ではデータの所有者によるプライバシー意識の多様性に対応するため, PPDM の保護レベルをパラメータ化しユーザが調整できるようにする動き (Multilevel Trust [11] の導入) も見られる. Yaping らは入力プライバシーの保護レベルをパラメータ化し, データから作成した分散共分散行列とパラメータに比例する値の積で定義される分散をもつ正規雑音を各データに加算する手法とともにこの概念を提唱した [11]. また, 出力プライバシーに関する研究としては, データの所有者が指定した確率でデータの値を別の値に置き換えてサーバに送信し, サーバが値が特定のカテゴリに属すユーザ数を集計するという状況において, 所有者が指定した確率・ユーザ数・カテゴリ数などの値を利用して真の集計計算結果を推定する手法が提案されている [12].

本研究は出力プライバシーに深く関係している. 既存の摂動法は乱数が関わるため, 歪められた集計計算結果の正しい集計計算結果との誤差が膨大になる場合があるので, 集計計算結果の値が確実性に欠ける. クエリ監査ではシステムの規模が大きくなった場合に記録するログの量が膨大になるという問題点がある. また, 両者ともに, データベース内の同じ属性列に対して, ユーザがその中の一部にのみアクセス可能とするようなアクセス制御を考えた場合に, アクセス可能な属性列が多いユーザが少ないユーザよりも正確な集計計算結果を得られるようにする

ことができない．Multilevel Trust を導入した PPDM に関する既存研究においても，先述のような集計計算結果を提供するためのパラメータ設定を個々のデータ所有者の意向とユーザのアクセス権限から求める手法が確立されていない．本研究では個々のデータ所有者がデータに設定したアクセスレベルとユーザのクラスに対応付けられたアクセスレベルの高低関係を考慮した上で先述のような集計計算結果をユーザが得られるようにする．災害時や災害後の復興にあたり被災者や自治体関係者，被災者支援団体との情報共有を考える上でそれぞれのユーザに提供する集計結果が出力プライバシーを守れるようにすることも考えていく．

3.3 データベースにおける選択性推定

データベースにおいて結合を含むクエリを実行する際には，実行速度や使用メモリの削減のために各結合演算の対象となるタプルの数ができるだけ小さくなるような結合順序で結合を行う必要がある．選択性推定 (Selectivity estimation) とは，結合対象となるそれぞれの途中計算結果について，タプル数やデータサイズを推定することである [13]．

データベースにおいてデータの検索を行う演算は選択 (SELECT) であり，これは属性列の値が特定の値になっているタプルを取得するなどの処理である．関係 R に対して選択演算 F を行った結果に含まれるタプル数の推定のためには，R に対して F がどの程度の割合でタプルを取得するかを推定する必要があり，これを選択率 (selectivity factor) と呼ぶ [14]．R に対して F を行った結果を $\sigma_F(R)$ とし，F に対する選択率を $SF(F)$ とすると，$\sigma_F(R)$ に含まれるタプル数は式 (1) のようにして求められる [14]．

$$|\sigma_F(R)| = SF(F) \times |R| \tag{1}$$

結合を行う場合，結合結果に含まれるタプル数が最も大きくなるのは，お互いに含まれるタプル全てに対して結合が成り立ち結合結果が結合に関わる関係の直積集合となる場合である．この濃度に対してどれだけ結合が成立するかを選択率とすれば式 (1) と同様に結合結果に含まれるタプル数を推定できる．関係 R と S の結合に対し，選択率を SF_J とすれば，$R \bowtie S$ に含まれるタプル数は式 (2) のようにして推定できる [14]．

$$|R \bowtie S| = SF_J \times |R||S| \tag{2}$$

選択率の求め方には，データベース全体の統計情報を利用する手法と，データベース内の一部のタプルを利用する手法などがある．データベース全体の統計情報を利用する手法では，関係に含まれるタプルにおける演算に利用する属性値の値域や最大値・最小値などから選択率を推定する [14]．

データベース全体に対する統計情報ではなく，データベース内の一部のタプルから選択率を推定する手法もある．Hou らはランダムに抽出したタプルの総数 m と，その中で選択演算の論理式を満たすタプルの総数 s を用いて選択率を式 (3) のように論理式を満たすタプルの割合として求めた [15]．なお，このとき複数の関係を含む結合演算結果に含まれるタプル数 σ_J は式 (2) と同様に考えると式 (4) のようにして求められる [15]．

$$SF = \frac{s}{m} \tag{3}$$

$$\sigma_J = SF_J \times \prod_{N \in \{\text{ 結合に関わる関係 }\}} |N| \tag{4}$$

本研究では選択性推定の手法を，データベース上のデータに対する集計結果をユーザのアクセス可能なデータから推定するために使用する．特に Hou らの手法に対し，ユーザがアクセス可能なデータのみを利用することを考える．こうすることにより，アクセス権限が十分に高く全てのデータにアクセス可能なユーザに対しては正しい選択率を推定できるが，アクセス権限が限定され一部のデータにしかアクセスできないユーザに対しては誤差を含んだ選択率を推定することになると考えられる．

3.4 避難場所情報に対する RDF データセットベンチマーク

Nguyen らは被災地の避難所場所の情報や，避難所における作業の関係者の情報を表現する RDF データを自動的に生成するベンチマークツール SIBM(Shelter Information Benchmark) を開発した [16]．

既存の RDF によるベンチマークツールとしては LUBM [17] があるが，生成されるデータは大学の学部ごとの学生や職員の個人情報や授業履修・担当情報であり，学生同士の細かい個人間関係を含んでいるわけではないため，避難所における避難者間の個人間関係を再現できず，被災地における情報共有システムの評価には適さないという問題点があった．

それに対し SIBM は避難所における避難者間の家族関係や，LUBM よりも詳細な個人情報，また避難所における災害の発生状況や物資の備蓄情報なども含んでいるため，LUBM よりも被災地における情報共有システムの評価に適していると考えられる．それに加えて SIBM は，生成する避難所情報は実際に存在し避難所として利用可能な施設の情報をもとにしていることや，生成する個人の情報は日本の人口構造情報をもとにしていることから，現実と近いデータセットの作成が可能である．これらの理由から，本研究ではこのベンチマークを利用して提案手法の評価を行う．SIBM が生成するデータの詳細を図 3 に示す．

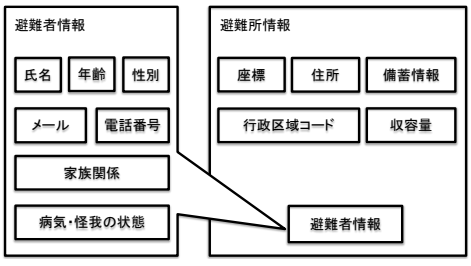

図 3　避難所情報

ベンチマークは，ユーザが指定した被災地と中心からの距離 $d(d > 0)$ を引数とするプログラムが作成する．ユーザが指定した被災地の中心地を中心とし，半径 $d$km 以内に存在する避難所とその避難所に避難している人の情報が生成される．

4. 提案手法

アクセス制御が行われた RDF データベース内で，ユーザの

アクセス可能なデータ数に対応した集計計算結果を推定する手法を説明する．本研究では集計計算の中でも COUNT(条件を満たす結果の総数) 計算の結果の推定を行う．以下では, RDF データベースに含まれるトリプル全体の集合を R とする．

SPARQL クエリの結合パターンとしては主語-主語が圧倒的に多く, 主語-目的語が後を追い, それ以外の組み合わせは 2 者に比べて非常に少ない [18] ことを利用すると, ほとんどの SPARQL クエリの WHERE 節は述語に変数でない値が, 主語に変数がそれぞれ指定されているトリプルパターンの集合に一般化できる．以降ではこれを踏まえ, クエリのトリプルパターンはこの構成であると仮定する．

SPARQL クエリの WHERE 節を G とすると, G はトリプルパターンの集合である．G にマッチするグラフパターンは R 内のトリプルの結合により得られるので, その最大数を Hou らによる選択性推定手法の式 (4) の $\prod|N|$ に対応させることができる．さらに (4) の選択率 SF_J は式 (3) の SF のように推定できることを利用する．ユーザ u がクエリを発行したとき, 児玉らの RDF データベースシステムは, G にマッチするグラフパターンのうち全ての要素に u の所属するクラスからアクセス可能なグラフパターンのみに対する検索結果を返す．これを利用するとユーザがアクセス可能なデータのみに対する COUNT 計算結果と, G に含まれる各トリプルパターンと述語部分がマッチする R 内のトリプルの中でユーザがアクセス可能なトリプルの総数を取得できる．G にマッチするグラフパターンは R 内のトリプルの結合により得られるので, 式 (3) の s を前者に, またユーザがアクセス可能なトリプルパターンの結合により得られるグラフパターン数は多くともユーザがアクセス可能で述語部分がマッチするトリプルパターンの直積集合の濃度なので m を後者の総積に対応させることができる．

以上の過程を経てユーザ u が発行したクエリのグラフパターン G にマッチするグラフパターンの総数の推定値 $ALL_{G,u}$ を推定するにあたって, Hou らの手法と対応させた部分の詳細な計算式を以下に示す．式 (4) の $\prod|N|$ に対応する値を T_G, 式 (3) の s に対応する値を $SA_{G,u}$, m に対応する値を $TA_{G,u}$ とする．(4) の SF_J に対応する値を $SF_{G,u}$ とする．以下では, ユーザ u がアクセス可能なアクセスレベルの集合を L_u とし, RDF 要素 e に設定されたアクセスレベルを l_e と表現する．また, G に含まれるトリプルパターンの数を $|G|$ とする．

$$T_G = \prod_{(s_G,p_G,o_G)\in G} |\{(s,p,o)\in R | p = p_G\}| \tag{5}$$

$$TA_{G,u} = \prod_{(s_G,p_G,o_G)\in G} |\{(s,p,o)\in R | l_s, l_p, l_o \in L_u \wedge p = p_G\}| \tag{6}$$

$$SA_{G,u} = |\{g \in R^{|G|} | g \text{ が } G \text{ にマッチ} \wedge \forall (s,p,o)\in g \; l_s, l_p, l_o \in L_u\}| \tag{7}$$

$$SF_{G,u} = \frac{SA_{G,u}}{TA_{G,u}} \tag{8}$$

$$ALL_{G,u} = SF_{G,u} T_G \tag{9}$$

具体的な例として, RDF データベース R に対する以下のクエリの集計計算結果の推定を考える．

```
SELECT (COUNT(*) AS  ?CNT) WHERE{
  ?p stayAt <shelter1>
  ?p state  <flu>
}
```

stayAt は主語が目的語である避難所に避難していることを表し, state は主語が目的語である病気に罹っていることを表す. これは避難所 shelter1 に避難しており, インフルエンザに罹っている人の総数を取得するクエリである. このクエリの WHERE 節内のグラフパターンを P とする．あるユーザ u_1 が, このクエリを児玉らのシステムに対し発行して, {{?CNT=10}} という結果が得られたとする．このとき $SA_{P,u_1} = 10$ である．P に含まれるトリプルパターンは (?p, stayAt, <shelter1>) と (?p state <flu>) であるので, T_P は u_1 がアクセス不可能なトリプルも含めた R 全体で述語が stayAt であるトリプルの総数と述語が state であるトリプルの総数の積になる．同様に考えると TA_{P,u_1} は u_1 がアクセス可能で述語が stayAt であるトリプルの総数と述語が state であるトリプルの総数の積になる. このとき R に対し次の 2 つのクエリが発行される．

```
SELECT (COUNT(*) AS ?CNT1) WHERE
{?p stayAt <shelter1>}
SELECT (COUNT(*) AS ?CNT2) WHERE{?p state <flu>}
```

R 全体に対する集計結果がそれぞれ {{?CNT1=3000}}, {{?CNT2=1500}} ならば, $T_P = 3000 \times 1500 = 4500000$ である． 同様に, ユーザ u_1 がアクセス可能なトリプルのみに限定した場合の集計結果がそれぞれ {{?CNT1=2000}}, {{?CNT2=500}} ならば, $TA_{P,u_1} = 2000 \times 500 = 1000000$ である. 結果として $ALL_{P,u_1} = 10 \times \frac{4500000}{1000000} = 45$ となる．

なお, ここで発行する COUNT クエリの計算は, 児玉らのシステムに機能を追加して SPARQL の COUNT 関数を用いてデータベース内でデータを暗号化したま行っている．クライアント側で値を復号して検索条件と比較し集計するよりも高速に集計が行えると考えられるからである．

ここで推定した COUNT 計算結果は, ユーザがアクセス不可能な RDF 要素の値に関する情報を使用していないため, ユーザに提供してもユーザがアクセス不可能な情報を露出したことにはならない．乱数を使用していないため, 得られる値はデータベースの内容とユーザのクラスが同じであれば常に同じである．さらに, ユーザがアクセス可能なアクセスレベルが増え, それに伴ってアクセス可能な RDF 要素が増加するにつれて, G にマッチするグラフパターンの選択率 ($= SF_{G,u}$) が真の割合に近づくので, 推定結果の誤差が小さくなる．

5. 実　　験

提案手法と, 差分プライバシーを満たすラプラス乱数の加算による出力プライバシー保護技術の性能を比較する．両者ともに児玉ら [7] の提案した手法を利用した RDF データベースシ

ステム上での評価を行った. また, 実行時間について, 児玉らのシステムに機能を追加しデータベース内で集計する場合にかかる時間と, クエリ結果に含まれるリテラル値をクライアント側で復号し検索条件を満たすか確認してカウントして集計結果を求めた場合にかかる時間を比較した. また, RDF データベースの作成, 検索にあたっては Java で開発された Jena [19] というライブラリを使用した.

5.1 環境とデータセット・アクセス制御ポリシー

実験には, 以下の環境を用いた.

CPU　Intel Xeon E5620
RAM　DDR3 SDRAM 24GB
OS　Ubuntu 11.10 64bit
Java　1.7.0_51
Jena　2.13.0

実験にあたり, 個人のプライバシーにかかわる情報を含んだデータセットとして, SIBM [16] を利用した. 被災地は福島県とし, 中心からの距離を 2.0, 3.0, 4.0, 4.5, 4.7, 5.1km と変更しながら 6 種類のデータセットを生成した.

実験で発行したクエリは以下の通りである.

```
PREFIX sibm: <http://lab.ene.im/Sibm/property#>
SELECT (COUNT(*) AS ?CNT)
WHERE{
  ?p       sibm:isOld                    “true”
  ?p       sibm:stayAt                   ?shelter
  ?shelter sibm:administrativeAreaCode 7203
}
```

このクエリ内で, 述語 stayAt は主語である人が目的語である避難所に避難していることを表し, 述語 administrativeAreaCode は目的語が主語である避難所の行政区域コードであることを示す. また, 述語 isOld は生成したデータセットに対し実験用に追加したものであり, 目的語の値 (true or false) が主語である人の年齢が 60 以上か否かに対応することを示す. 児玉らの手法では数値データの大小比較ができないので, 大小比較でなく等価か否かの比較だけで集計計算が行えるようにするために追加した. このクエリは行政区域コード 7203(福島県郡山市) の避難所に避難している 60 歳以上の人の総数を取得するクエリである. このクエリに含まれるリテラル値は true と 7203 であるので, クライアント側で復号する手法ではそれらを変数に置き換えてそれらの値を取得するクエリを発行し, それらの変数の値が true および 7203 であるようなグラフパターンの総数をカウントする.

なお, 今回生成した 6 種類のデータセットの詳細と発行するクエリに対する真の集計結果は表 1 のとおりである. クエリの発行元ユーザは, Press(報道関係者), Volunteer(ボランティア), Evacuee(避難者), Public Officer(自治体関係者), Medical Person(医療関係者) の 5 種類とし, それぞれについて検索を行った. なお, クライアント側でリテラルを復号する場合の検索時間の計測については Volunteer ユーザのみで行った. 実験におけるアクセスレベルと前述のユーザクラスとの対応関係は, 5

表 1　データセットの詳細

d	避難所総数	避難者総数	トリプル数 (暗号化前)	真の集計結果
2.0	15	3289	49994	236
3.0	24	7514	114093	509
4.0	37	10674	162022	748
4.5	54	14712	223494	986
4.7	61	17724	269262	1205
5.1	73	21621	327592	1443

つのアクセスレベルを想定し, 図 4 のように設定した.

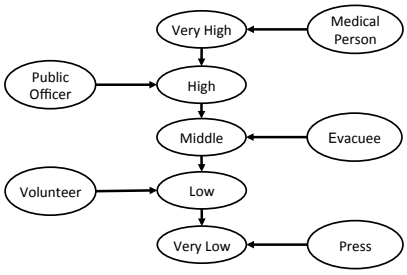

図 4　アクセスレベルとユーザクラスの対応関係

また, SIBM に含まれる述語へのアクセスレベルの設定は, 所有者の意向が多様であることを想定し, 所有者により異なるアクセスレベルを設定できるようにした. 各述語に対し, 各アクセスレベルに設定される確率を決めておき, 乱数を利用してその確率でアクセスレベルが設定されるようにした. 今回の実験で実行したクエリ内の述語に対するアクセスレベル設定確率の分布は図 5 と図 6 に示してある.

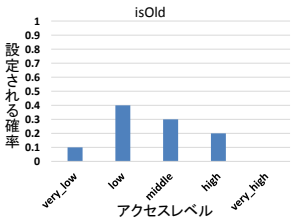

図 5　設定確率分布

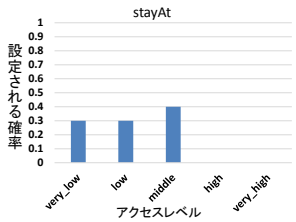

図 6　設定確率分布

なお, administrativeAreaCode などの避難所情報は個人情報でないため, アクセスレベルは Very Low よりも低いアクセスレベルを設定しており, どのユーザからでもアクセス可能になっている.

暗号方式としては, 児玉らの実験と同様に Blaze らの BBS アルゴリズム [20] を使用した. 比較対象で使用したラプラス乱数 $L(\lambda)$ のパラメータ $\lambda = \frac{\Delta}{\varepsilon}$ は $\frac{1}{0.1}$ に設定した. COUNT 計算に対して差分プライバシーを満たすラプラス雑音の Δ は 1 になることが知られている [21] ためである.

5.2 集計計算結果の相対誤差

提案手法で推定した集計計算結果と, 真の集計計算結果 (アクセスレベルを無視した集計計算結果) にラプラス雑音を加算した値のそれぞれについて, 真の集計計算結果との相対誤差を算出し, 暗号化前のトリプル数を横軸とするグラフにまとめたのが

図 7 である．凡例は noisy は真の集計結果にラプラス雑音を加算した値の平均値 (試行回数は 10 回) の相対誤差を示し，noisy 以外はクエリ発行元のユーザクラスを示す．noisy については信頼区間も求め，その範囲もグラフ内に示した．信頼度は 95%とし，分散はラプラス乱数の分散 $2 \times \left(\frac{\Delta}{\varepsilon}\right)^2 = 2 \times \left(\frac{1}{0.1}\right)^2 = 200$ を用いた．図 7 のグラフより，提案手法で推定した集計計算結

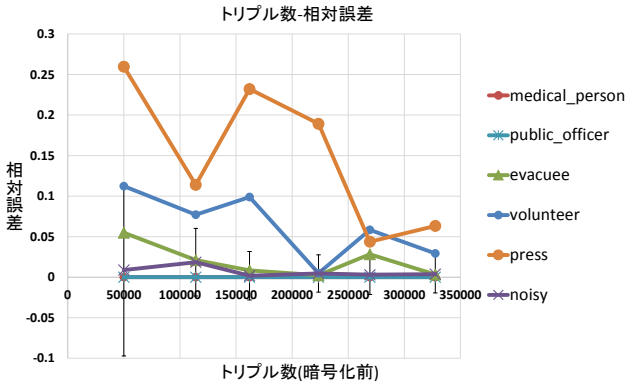

図 7　ユーザクラスごとの相対誤差比較

果は，データセットの規模が大きくなるにつれて，真の集計計算結果との相対誤差が小さくなる傾向があることがわかる．

ユーザクラスごとに考えると，isOld と stayAt の両方について全てにアクセス可能な Medical Person および Public Officer は誤差のない集計計算結果を得ることができ，isOld へアクセス可能な割合が 2 割だけ減少した Evacuee は noisy の信頼区間の最大値よりも小さい相対誤差で集計計算結果を得られた一方で，両者ともに約半分までしかアクセスできない Volunteer は noisy の信頼区間の最大値と同程度か 2~3 倍の相対誤差がある集計計算結果しか得られず，さらに両者へのアクセス可能な割合が減少した Press が得た集計計算結果に至っては，noisy の信頼区間の最大値と比較しても最大で約 7 倍の相対誤差があった．

さらに，図 7 で noisy の相対誤差の平均値と信頼区間を観察すると，ラプラス雑音が加算された集計計算結果は相対誤差がほぼ 0 から 1 割に及ぶこともあるので，実用に際し得られた集計計算結果の確実性が低いことによる問題が発生することが考えられる．一方，提案手法で得られた集計計算結果はデータセットとアクセス制御と問い合わせ元ユーザのクラスが同じであれば常に同じ結果が返ってくるため，確実性は高いと考えられる．

提案手法はクエリが含むトリプルパターンに関係する統計値を利用しているため，クエリによって相対誤差が変わることが考えられる．データセットの規模との関係も含めて観察するため，Volunteer が属するユーザからのクエリについて，5.1 で紹介したクエリの他に，含まれるトリプルパターンを述語が isOld であるトリプルパターンのみにしたクエリに対しても集計計算結果を推定し，相対誤差を計測した．図 8 は，それをグラフにまとめたものである．図 8 のグラフより，推定した集計計算結果の相対誤差は，クエリに含まれるトリプルパターン数が大きいと大きくなる傾向があることがわかる．これはユーザクラスに与えられたアクセス権限と無関係に集計計算結果の精度が低下することを示す．提案手法にはこの点において改善の余地があると考えられる．また，相対誤差の差はデータセットの規模が大きくなるにつれて小さくなる傾向があることもわかった．

5.3　集計計算の実行時間

提案手法は検索対象のデータに含まれるリテラル値を暗号化されたままデータベース内で検索を行っている．そのため検索結果からリテラル値を復号して検索条件に指定した値と等しいかどうか確認するよりも高速に集計計算を実行できるのではないかと考えられる．このことを確かめるため，集計計算結果の相対誤差のほかに集計計算の実行時間も計測した．

Volunteer に属するユーザからのクエリについて，データベース内で集計計算を行った場合の実行時間と，リテラル値をクライアント側で復号して集計を行った場合の実行時間を比較し，図 9 のグラフにまとめた．図 9 のグラフより，実行時間の差はデータセットが増加するにつれ大きくなっていることがわかる．また，性能比は最大で 14 倍になった．データベース内で暗号化したまま検索を行うことにより，値をクライアント側で復号して確認するよりも集計計算がより高速に行えることがわかった．

しかしながら，実用化を目指すにあたってはさらなる高速化が必要であることも考えられる．今回使用したデータセットの中で最も規模の大きい $d = 5.1$ のデータセットに対しては集計計算に 2 時間以上もの時間を要している．

6.　おわりに

6.1　まとめ

階層的なアクセス制御を行ったデータベース上で，ユーザがアクセス可能なデータのみに対する集計計算結果と個人情報の値と無関係なデータベース内の統計情報からユーザがアクセス

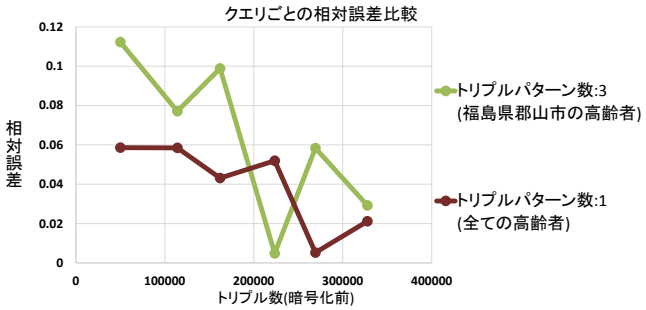

図 8　クエリごとの相対誤差比較

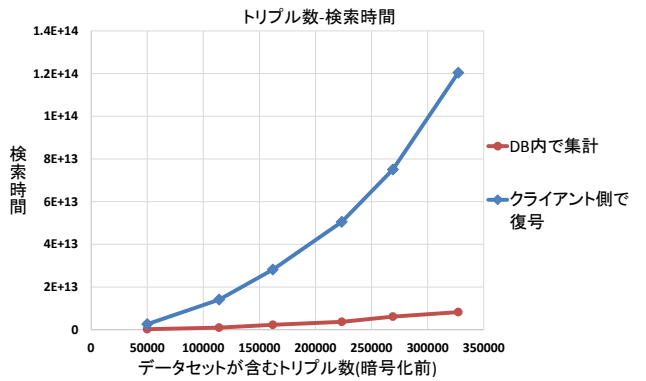

図 9　集計計算の実行時間

できないデータも含めたデータ全体に対する集計計算結果を推定する方法を提案した.

被災地の避難所情報を再現したデータセット上で階層的なアクセス制御を行い, 提案手法の評価を行った結果, 提案手法で推定した集計計算結果の相対誤差は, ユーザのアクセス権限が高く全てのデータにアクセス可能なときは 0 であり, アクセス可能なデータ数が 8 割以上であれば差分プライバシーを考慮した処理を施した集計計算結果の信頼区間の最大値よりも小さいが, ユーザのアクセス権限が低くなりアクセス可能なデータ数が減少するにつれて差分プライバシーを考慮した処理を施した集計計算結果の信頼区間の最大値より大きくなる傾向が見られた. クエリに含まれるトリプル数が増加すると集計計算結果の相対誤差が大きくなる傾向も見られた. また, リテラル値を暗号化したままデータベース内で集計計算を行う場合とクライアント側でリテラル値を復号して集計を行う場合の実行時間を比較すると, データベース内で集計する方が最大で 14 倍高速に集計計算を行えることがわかった. 実行時間は実用化にあたりさらなる削減が必要である.

6.2 今後の課題

まず, クエリに含まれるトリプルパターン数が増加するとユーザのアクセス権限と無関係に集計計算結果の相対誤差が大きくなる傾向がある点について改善が必要である. 提案手法はトリプルの結合結果に含まれるグラフパターン数がトリプルの直積集合と等しくなる場合をグラフパターン数の上限としており, 結合が成立する確率が 100%でない場合を反映できていないことが原因であると考えられる. 式 (5) の T_G や式 (6) の $TA_{G,u}$ を求める際に, 結合が成立する確率を考慮してトリプル数を掛け合わせることで, 相対誤差の増加抑制を図れるのではないかと考えられる.

次に, データセットによる相対誤差の違いについて, 今回使用したデータセットとは異なるデータセットを使用した実験による観察が必要である. 同じ SIBM を使用するとしても, 被災地を変更すると異なる避難所の情報が生成されるので, その違いが相対誤差に与える影響について考察する必要がある. また, SIBM と構成そのものが異なっているデータセットを使用した場合の相対誤差がどうなるかについても実験の後考察が必要である.

最後に, 本研究で推定の対象にした集計計算は COUNT のみであるが, 集計計算としては SUM(総和), AVERAGE(平均) なども存在する. これらの集計計算にも対象にする必要がある. 特に SUM 計算結果の推定は準同型暗号を使用することで値を暗号化したまま安全に行えるのではないかと考えられる.

謝　　辞

本研究の一部は, 日本学術振興会科学研究費補助金基盤研究 (A)(#25240014) の助成により行われた. ここに謝意を表す.

文　　献

[1] Glossary of computer security terms (ncsc-tg-04), 1988.

[2] Raluca Ada Popa, Catherine M. S. Redfield, Nickolai Zeldovich, and Hari Balakrishnan. Cryptdb: Protecting confidentiality with encrypted query processing. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*, SOSP '11, pp. 85–100. ACM, 2011.

[3] 総務省情報流通行政局情報流通振興課. オープンデータ戦略の推進. Retrieved January 8, 2016, from: `http://www.soumu.go.jp/menu_seisaku/ictseisaku/ictriyou/opendata/`.

[4] David Wood, Marsha Zaidman, Luke Ruth, and Michael Hausenblas. *Linked Data*. Manning Publications Co., 1st edition, 2014.

[5] Florian Haag, Steffen Lohmann, and Thomas Ertl. Sparql-filterflow: Sparql query composition for everyone. In *The Semantic Web: ESWC 2014 Satellite Events*, pp. 362–367. Springer, 2014.

[6] ［オープンデータ＋ QGIS］ 統計・防災・環境情報がひと目でわかる地図の作り方. 株式会社技術評論社, 2014.

[7] 児玉快, 横田治夫. データやユーザの効率的な追加・削除が可能な秘匿情報アクセス手法. 第 7 回データ工学と情報マネジメントに関するフォーラム, 2015.

[8] 佐久間淳. プライバシー保護データマイニング (私のブックマーク). 人工知能学会誌, 第 26 巻, pp. 533–536. 社団法人人工知能学会, 2011.

[9] Cynthia Dwork. Differential privacy. In *Automata, Languages and Programming*, Vol. 4052 of *Lecture Notes in Computer Science*, pp. 1–12. Springer Berlin Heidelberg, 2006.

[10] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Conference on Theory of Cryptography*, TCC'06, pp. 265–284. Springer-Verlag, 2006.

[11] Yaping Li, Minghua Chen, Qiwei Li, and Wei Zhang. Enabling multilevel trust in privacy preserving data mining. *Knowledge and Data Engineering, IEEE Transactions on*, Vol. 24, No. 9, pp. 1598–1612, 2012.

[12] 清雄一, 大須賀昭彦. Randomized response を用いた柔軟な匿名データ収集. 電子情報通信学会論文誌 D, Vol. 97, No. 5, pp. 953–963, 2014.

[13] Evaggelia Pitoura. Selectivity estimation. In *Encyclopedia of Database Systems*, p. 2548. Springer US, 2009.

[14] M. Tamer Özsu and Patrick Valduriez. *Principles of Distributed Database Systems*. Computer science. Springer, 3 edition, 2011.

[15] Wen-Chi Hou, Gultekin Ozsoyoglu, and Baldeo K. Taneja. Statistical estimators for relational algebra expressions. In *Proceedings of the Seventh ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, PODS '88, pp. 276–287. ACM, 1988.

[16] Nguyen Hoai Nam, 荒堀喜貴, 横田治夫. SIBM: 避難場所情報に対する RDF データセットベンチマークツール. 第 7 回データ工学と情報マネジメントに関するフォーラム, 2015.

[17] Yuanbo Guo, Zhengxiang Pan, and Jeff Heflin. Lubm: A benchmark for owl knowledge base systems. *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 3, No. 2–3, pp. 158–182, 2005.

[18] Mario Arias Gallego, Javier D Fernández, Miguel A Martínez-Prieto, and Pablo de la Fuente. An empirical study of real-world sparql queries. *CoRR*, Vol. abs/1103.5043, , 2011.

[19] Apache jena – home. Retrieved January 8, 2016, from: `http://jena.apache.org/`.

[20] Matt Blaze, Gerrit Bleumer, and Martin Strauss. Divertible protocols and atomic proxy cryptography. In *Advances in Cryptology—EUROCRYPT'98*, pp. 127–144. Springer, 1998.

[21] 寺田雅之, 竹内大二朗, 齊藤克哉, 本郷節之. 差分プライバシー基準に基づく情報秘匿手法の一考察. マルチメディア、分散協調とモバイルシンポジウム 2014 論文集, pp. 224–233, 2014.