完全準同型暗号下での差分プライバシ適用 ―レンジクエリを対象として―

牛山 翔二郎† 髙橋 翼‡ 工藤 雅士§ 井上 紘太朗§ 鈴木 拓也§ 山名 早人〃

† 早稲田大学基幹理工学部 〒169-8555 東京都新宿区大久保 3-4-1
‡ LINE 株式会社 Data Labs 〒160-0004 東京都新宿区四谷 1-6-1 四谷タワー23 階
§ 早稲田大学基幹理工学研究科 〒169-8555 東京都新宿区大久保 3-4-1
〃 早稲田大学理工学術院 〒169-8555 東京都新宿区大久保 3-4-1

E-mail: † § 〃 {s-ushiyama, kudoma34, kinoue, t-suzuki, yamana}@yama.info.waseda.ac.jp

‡ tsubasa.takahashi@linecorp.com

あらまし 近年，データベースに対する問い合わせ応答システムとして，クラウドコンピューティングの活用が注目される．複数のデータ提供者が所有するパーソナルデータをクラウドサーバへ保存し，クラウドサーバがデータ解析者の問い合わせに対して応答を行うシステムを想定する．ここで，データ提供者が所有するデータが，クラウドサーバとデータ解析者に漏洩する問題が存在する．準同型暗号を使用することで，クラウドサーバに対して，データ提供者が所有するデータを秘匿することができる．また，差分プライバシを使用することで，データ解析者に対して，データ提供者が所有するデータのプライバシを保護することができる．これら 2 つを同時に実現する方法として，従来，準同型暗号と差分プライバシを組み合わせるプライバシ保護手法が提案されている．しかし，準同型暗号と差分プライバシを組み合わせるプライバシ保護手法には，準同型暗号に起因する応答速度低下と，差分プライバシに起因する応答回数制限の問題がある．本稿では，これらの問題を解決するため，解析対象データに対して暗号文上で差分プライバシを事前に適用した後，復号し，要約データとして保存する．そして，平文で保存される要約データに対して，問い合わせを行う．これにより，問い合わせ応答の速度を向上させる．また，要約構成時に差分プライバシを保証したデータを用いて，全ての問い合わせに応答することで応答回数制限の問題を解決する．提案手法は，データ解析者の問い合わせを高速化する一方で，要約の構成には時間がかかる．完全準同型暗号ライブラリ TFHE を用いて提案システムを構築し，要約構成にかかる時間を測定した結果，16 ビットで表現される 6 個のデータに対して約 2.5 時間が必要となることが分かった．

キーワード 差分プライバシ，準同型暗号，完全準同型暗号，TFHE

1. はじめに

近年，データベースに対する問い合わせ応答システムとして，クラウドコンピューティングの活用が注目されている．しかし，機密性の高いデータをクラウドサーバ上で処理する際に,データの漏洩が問題となる．具体的に,解析されるデータを提供するデータ提供者，提供されたデータに対して処理を行うクラウドサーバ，クラウドサーバに対する問い合わせ応答の出力から任意の解析を行うデータ解析者の 3 つのエンティティを想定する．データ提供者が提供するデータを以後，元データと呼ぶ．この時に，元データがクラウドサーバとデータ解析者に漏洩する危険性が存在する．本稿では，この問題に対する対策として，準同型暗号[1]と差分プライバシ[2]の 2 つのプライバシ保護手法を使用する．

準同型暗号[1]は，暗号化されたデータに対して復号せずに演算を行い，演算後の復号結果が正しい演算結果となる性質を持った暗号の総称である．準同型暗号を使用し，クラウドサーバ上で行われる演算を暗号文上で行うことで，クラウドサーバに対して元データを秘匿することができる．

差分プライバシ[2]は，元データもしくは元データを使用した計算結果の出力値にノイズを加算することにより，元データや元データを使用した計算結果の出力値に誤差を含ませることで元データを推定することを困難にするプライバシ保護手法である．差分プライバシの適用方法としては，(1)データ提供者が元データをクラウドサーバに送信する前に，データ提供者が差分プライバシを適用する方法，(2)受信した元データをクラウドサーバが集約した後に，クラウドサーバが差分プライバシを適用し，差分プライバシ適用後のデータを使用して問い合わせ応答を行う方法，(3)個々のデータ解析者からの問い合わせ応答結果ごとに，クラウドサーバが差分プライバシを適用する方法の 3 つがある．

(1)では，データ集約前にノイズが加算されるため，データ解析者に送信される問い合わせ応答結果の誤差が大きくなる．(2)及び(3)では，クラウドサーバが元データを受信するため，クラウドサーバを信頼する必要がある．また(3)の方法では，差分プライバシのプライバシ予算の観点から応答回数に制限が必要となる．

上記で述べた問題，すなわち差分プライバシを単体のみで用いた場合の問題を解決するため，本稿では差分プライバシに対して準同型暗号を組み合わせるプライバシ保護手法に注目する．準同型暗号と差分プライバシを組み合わせる研究への注目度が 2015 年ごろから増加している[3]．2020 年に Chowdhury ら[4]は，準同型暗号と差分プライバシを組み合わせることで，クラウドサーバとデータ解析者の両者に対して，元データを保護することを実現する問い合わせ応答システムを提案した．Chowdhury らが提案したシステムは，上記の差分プライバシの適用方法の(3)に該当する．データ解析者の問い合わせが入力されてから，準同型暗号下で問い合わせに対する応答を計算するため，応答速度の低下が問題となる．また，差分プライバシのプライバシ予算の観点から，データ解析者が行う問い合わせに回数制限を設ける必要がある．

本稿では，クラウドサーバとデータ解析者の両者に対して，元データのプライバシ保護を目的に，準同型暗号と差分プライバシを組み合わせる．また，準同型暗号による応答速度低下と，差分プライバシによる問い合わせの回数制限の問題を解決するために，上記で示した(2)のモデルを採用する．本稿では，事前に差分プライバシを保証したデータを完全準同型暗号[5]下で構築し，復号サーバを用いて復号し，平文でクラウドサーバに保存した上で，差分プライバシ保証済みデータを使用して問い合わせ応答を行うシステムを提案する．以後，平文で表現される差分プライバシ保証済みデータを要約と呼ぶ．提案手法では，データ解析者が行う問い合わせに対して，事前に構築した要約を用いて，クラウドサーバが応答する．データ解析者の問い合わせに対して平文で処理を行うことにより，高速な問い合わせ応答を可能にすることで，準同型暗号に起因する応答速度低下の問題を解決する．また，要約構成時に差分プライバシを適用したデータを用いて，全ての問い合わせに応答するため，データ解析者は多数の問い合わせを行ったとしても，データ提供者のデータに関する情報に対する統計的な推測を行うことができない．これは，データ解析者の行う問い合わせに回数制限を設ける必要がないことを意味しており，差分プライバシに起因する応答回数制限の問題を解決する．また，提案手法では，差分プライバシによって発生する誤差を低減することを目的に，2014 年に Li ら[6]が提案した差分プライバシアルゴリズムを完全準同型暗号下で実現する．

本稿の貢献を以下に示す．

- データ提供者が所有するデータをクラウドサーバ及びデータ解析者の両者から保護する，準同型暗号と差分プライバシを組み合わせる問い合わせ応答システムにおいて，準同型暗号に起因する応答速度低下の問題と，差分プライバシに起因する応答回数制限の問題を解決する．

本稿の構成は以下の通りである．2 節では本稿の主要な背景知識である準同型暗号と差分プライバシについて説明する． 3 節では本稿における関連研究を示す．4 節で提案手法の詳細を説明する．5 節で提案手法の性能を実験により評価する．最後に 6 節で結論を示す．

2. 背景知識

2.1 準同型暗号

準同型暗号[1]とは，暗号化されたデータに対して復号せずに演算を行い，演算後の復号結果が正しい演算結果となる性質を持った暗号の総称である．暗号文上で行われる加算を準同型加算，乗算を準同型乗算と呼ぶ．特に，任意回数の準同型加算と準同型乗算を可能にする準同型暗号を完全準同型暗号[5]と呼ぶ．準同型暗号では，攻撃に対する耐性を高めるために，ノイズを暗号文に加える[1]．暗号文に含まれるノイズは，暗号化や準同型加算，準同型乗算によって増加する．暗号文内に蓄積されるノイズが一定量を超えると正しく復号を行うことができなくなる．完全準同型暗号は，ノイズを削減する処理であるブートストラッピングを行うことで，任意回数の準同型加算と準同型乗算を可能にする．また，一般的に準同型乗算は準同型加算よりも多くのノイズを蓄積させる．

2.2 差分プライバシ

差分プライバシ[2]は，データ及びデータを演算した結果に対してノイズを加算することで，ノイズを加算する前のデータのプライバシを保護する．差分プライバシは任意の背景知識を持つ攻撃者に対して有効であるとされている[7]．一般的に，差分プライバシにおける，プライバシ保護強度と差分プライバシが適用されたデータの有用性はトレードオフの関係がある．具体的には，ノイズを多く加算するとプライバシ保護強度が向上する一方で，差分プライバシが適用されたデータは真の値との誤差が大きくなり，差分プライバシ適用後のデータの有用性が低下する．

差分プライバシにおけるノイズは，平均 0 の確率分

[1] 準同型暗号におけるノイズは，差分プライバシにおけるノイズとは異なる．本稿において，本項以外では，「ノイズ」は差分プライバシにおけるノイズを指す．

布に従ってサンプリングされるため，差分プライバシが適用されたデータを多数集めると，統計的性質を復元することができる．そのため，差分プライバシを保証したデータを多く集めた攻撃者は，差分プライバシが保証される前のデータに対して統計的な推測を行うことができる．これを防ぐため，差分プライバシではプライバシ予算を設定することで，ノイズ加算を行うことができる上限回数を設ける必要がある．

差分プライバシは数学的な根拠に基づいたプライバシ保護手法であり，以下の定義 1 を満たすことを「メカニズムmはϵ-差分プライバシを満たす」と言う．また，ϵはプライバシパラメータと呼ばれ，0 より大きい実数値を取る．ϵの大きさでプライバシ強度を調節することができる．具体的には，ϵが小さいほど保証されるプライバシ強度は強くなる．

定義 1([7][8]より引用)：**ϵ-差分プライバシ**

クエリqにおいて，$d(D,D')=1$なる任意のデータベースの組D,D'，およびクエリ応答の出力の部分集合Sについて，以下の不等式を満たす．

$$\frac{Pr_{z\leftarrow m(y),y\leftarrow q(D)}(z\in S)}{Pr_{w\leftarrow m(x),x\leftarrow q(D')}(w\in S)}\leq \exp(\epsilon) \quad (1)$$

$\Pr(a)$は，ある事象aが起こる確率を表す．また，$d(D,D')=1$とは，2 つのデータベースDおよびD'が 1 つのレコードを除いて，残りのレコードが全く同じデータベースであることを意味する．$d(D,D')=1$なる任意のデータベースの組(D,D')を隣接データベースと呼ぶ．

ランダムメカニズム(以後，メカニズム)とは，ノイズの付加等によって差分プライバシを満たすランダム性を持つ計算機構である．代表的なメカニズムとして，ラプラス分布に従ってノイズをサンプリングする，ラプラスメカニズム[2]がある．ラプラスメカニズムは，プライバシパラメータとクエリの敏感度によって設計されたラプラス分布を用いる．すなわち，ラプラスメカニズムの出力は，プライバシパラメータとクエリの敏感度に依存する．敏感度とは，隣接データベースに対して，単一のレコードがクエリqの出力に与える影響の最大値を表している．敏感度は以下の定義 2 で定義される．

定義 2([2][7]より引用)：**敏感度$\Delta_{1,q}$**

$$\Delta_{1,q}=max_{D\sim D'}\|q(D)-q(D')\|_1 \quad (2)$$

ここで$\|q(D)-q(D')\|_1$は，$d(D,D')=1$であるデータベースD,D'のクエリに対する出力の差をℓ_1ノルムで評価する．

敏感度を用いて，ラプラスメカニズムの定義を以下の定義 3 に示す．定義 3 における$Lap(R)$は平均 0 のラプラス分布を表しており，$Lap(R)=\frac{1}{2R}\exp\left(-\frac{|x|}{R}\right)$である．

定義 3([2][7]より引用)：**ラプラスメカニズム**

データベースD，プライバシパラメータϵ，クエリqの敏感度$\Delta_{1,q}$において，yを出力する．

$$y=q(D)+r \qquad \left(r\sim Lap\left(\frac{\Delta_{1,q}}{\epsilon}\right)\right) \quad (3)$$

ここでrはラプラス分布$Lap\left(\frac{\Delta_{1,q}}{\epsilon}\right)$からサンプリングされる乱数である．

3. 関連研究

3.1 準同型暗号と差分プライバシを組み合わせるプライバシ保護手法の関連研究

2020 年に提案された Chowdhury ら[4]の研究では，1 つの計算サーバと 1 つの復号サーバの計 2 つのクラウドサーバを使用して，準同型暗号と差分プライバシを組み合わせたプライバシ保護問い合わせ応答システムを構築している．Chowdhury らのシステムを図 1 に示す．Chowdhury らのシステムでは，データ提供者が暗号化したデータをクラウドサーバが受信した後，準同型演算を用いてデータの集約が行われる．クラウドサーバでの処理の過程で，復号サーバ内で復号が行われるが，暗号文に対して差分プライバシを保証するノイズを加算した後で復号を行うことで，クラウドサーバが元データの値を知ることができないことを保証している．また，データ解析者が受信する問い合わせ応答の出力には差分プライバシが保証されている．元データは準同型暗号によりクラウドサーバから秘匿されると共に，差分プライバシにより復号サーバとデータ解析者から保護される．一方で，Chowdhury らのシステムは，データ解析者が行う問い合わせの個々の応答に準同型演算を用いて差分プライバシ適用するため，問い合わせの応答速度が遅くなると共に，問い合わせに回数制限が生じる．具体的に，データ提供者の年齢に関する問い合わせとして，1 から 100 歳までの各年齢の累積分布を求める問い合わせに対して，約 30 分の応答時間を要する[4]．

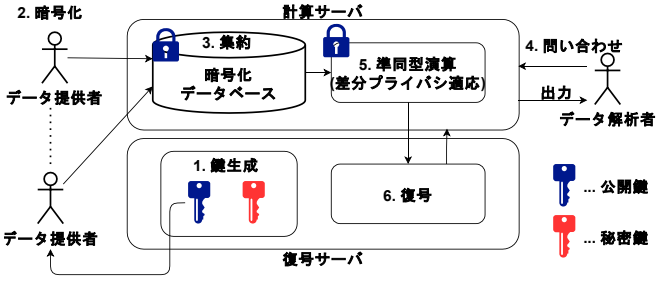

図 1　Chowdhury ら[4]のシステム

3.2 差分プライバシの関連研究

2014 年に提案された Li ら[6]の差分プライバシアル

ゴリズムは，ヒストグラムで表現される 1 次元及び 2 次元のデータを対象に，競合する手法と比較して低誤差を達成する[6]．また，Li らの手法は，問い合わせとして，レンジクエリのみを対象としている．Li らが提案したアルゴリズムは，ヒストグラムで表現される入力データに対して値が近いまとまりごとに分割を行う処理と，レンジクエリの集合に対して応答結果の誤差を低減するように最適化する処理から構成されている．ヒストグラムで表現される入力データに対して値が近いまとまりごとに分割を行う処理を以後，パーティショニングと呼ぶ．Li らのアルゴリズムでは，ヒストグラムで表現されるデータに対してパーティショニングを行い，それぞれのまとまりに対してノイズを加算する．個々のデータにノイズを加算するのではなく，いくつかのまとまり分割してからノイズを加算することで，ヒストグラム全体として加算するノイズの合計量を低減する．パーティショニング処理とパーティショニングによって分割したまとまりごとにノイズを加算する処理を合わせて以後，DA アルゴリズム(Data-Aware Algorithm)と呼ぶ．

なお，Li ら[6]が提案したアルゴリズムは，Hay ら[9]によって検証された．Hay らは，包括的な性能評価をすることが困難である差分プライバシアルゴリズムの評価原則を提案した．Hay らは，提案した評価原則に基づく評価フレームワークを用いて，15 個の差分プライバシアルゴリズムを評価した．実験結果から，総合的に評価して，Li らアルゴリズムは差分プライバシに起因する出力の誤差が小さいことを示した．

4. 提案手法

4.1 提案手法の概要

本節では，完全準同型暗号下で差分プライバシを適用したデータの要約を事前に構築することで，データ解析者の問い合わせに対して回数制限なく高速に応答する問い合わせシステムを提案する．提案手法は，Chowdhury ら[4]の準同型暗号と差分プライバシを組み合わせた問い合わせ応答システムにおける，準同型暗号に起因する応答速度低下と差分プライバシに起因する応答回数制限の問題を解決することを目的としている．提案手法では，事前に構築された平文で表現される要約を用いて，データ解析者の問い合わせに高速に応答することで，準同型暗号に起因する応答速度低下の問題を解決する．また，個々の問い合わせに差分プライバシ適用するのではなく，事前に構築された要約から全ての問い合わせに応答することで，差分プライバシに起因する応答回数制限の問題を解決する．

提案手法の概要図を図 2 に示す．提案手法は，暗号文で集約したデータに対する差分プライバシを適用した要約を，完全準同型暗号下で事前に構成し，要約を用いて問い合わせに応答することで，問い合わせの応答速度低下と応答回数制限の問題を解決する．差分プライバシを適用した要約を構成する際，Li ら[6]が提案した差分プライバシアルゴリズムの一部である，DA アルゴリズムを使用する．

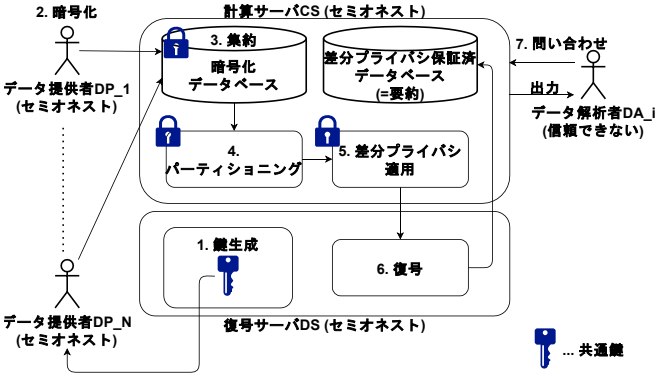

図 2　提案手法の概要

提案手法ではデータ提供者，計算サーバ，復号サーバ，データ解析者の 4 つのエンティティを想定する．データ提供者，計算サーバ，復号サーバはセミオネストであると仮定する．ここで，セミオネストであるとは，提案手法のシステムのプロトコルには従うが，データ提供者が所有するデータを盗み見ようとするエンティティであることを指す．データ解析者は信頼できないエンティティであると仮定する．また，計算サーバと復号サーバは相互に，及び他のエンティティと共謀しないことを前提とする．それぞれのエンティティの説明を以下に示す．以下において **N**，**M** はそれぞれ任意の正の整数である．

- **データ提供者(DP_j(1≤j≤N))**
  データ提供者の人数は N 人(N は 1 以上の整数)であるとする．データ提供者は，復号サーバから受信した共通鍵を用いて，自身のデータを暗号化して計算サーバに送信する．データ送信後，データ提供者はシステムに関与しない．
- **計算サーバ(CS)**
  計算サーバは，データ提供者から受信した暗号化データに対して，完全準同型暗号下でデータの集約，差分プライバシの適用を行う．また，復号サーバと協力して受信した暗号化データに対する，差分プライバシを保証したデータの要約を構築する．計算サーバ上に保存されるデータは，完全準同型暗号及び差分プライバシのいずれか，または両方によって常に保護されている．
- **復号サーバ(DS)**
  復号サーバは鍵の生成と，計算サーバから受信

する暗号文データの復号を行う．復号サーバが復号するデータには，差分プライバシが保証されているため，データ提供者の元データを知ることはできない．

- **データ解析者(DA_i(1≤i≤M))**
  データ解析者の人数はM人(Mは1以上の整数)であるとする．データ解析者は，多数の問い合わせを行うことで，データ提供者が所有する元データに対して統計的な推測を試みることができる．データ解析者は計算サーバに対して問い合わせを行うことで，問い合わせに対する応答を計算サーバから得る．

提案手法のシステムでは，データ提供者が所有するデータのプライバシを，計算サーバと復号サーバ，データ解析者に対して保護する．一方で，差分プライバシが保証されたデータを計算サーバ，復号サーバが保持することは許容する．また，算出されるパーティショニングの結果に対して差分プライバシを保証することで，計算サーバと復号サーバがパーティショニングの結果を平文で保持することを許容する．共通鍵はデータ提供者と復号サーバのみが保持する．暗号化はデータ提供者のみが行い，復号は復号サーバのみで行われる．

次に，図2に基づいて提案手法の手順を示す．以下の手順において，「**1. 鍵生成**」から「**6. 復号**」までを前処理として，データ解析者の問い合わせが行われる前に完了する．

1. **鍵生成：**DSが，鍵を生成する．生成した共通鍵をDP_j(1≤j≤N)に安全に送信する．
2. **暗号化：**DP_jは，DSから受信した共通鍵を用いて，自身のデータを暗号化し，CSに送信する．
3. **集約:**DP_jから暗号文データを受信したCSは，準同型演算を使用して，複数の暗号文データを集約する．
4. **パーティショニング：**CSは集約した暗号文データに対して，準同型演算を使用して，パーティショニングを行う．復号サーバを使用することで，パーティショニングの結果を平文で得る．
5. **差分プライバシ適用：**CSは，パーティショニングによって分割されたまとまりごとに，準同型演算とラプラスメカニズムを用いてノイズを加算する．CSはノイズ加算後の暗号文データをDSに送信する．
6. **復号:**DSはCSから受信した暗号文を復号する．復号したデータをCSに送信する．
7. **問い合わせ：**DA_i(1≤i≤M)は，CSに対して問い合わせを行い，問い合わせ応答結果を得る．

4.2 完全準同型暗号下での差分プライバシ適用

本項では，完全準同型暗号下で差分プライバシを適用する方法を示す．提案手法では，差分プライバシを適用する際に，Liら[6]が提案した差分プライバシアルゴリズムの一部である，DAアルゴリズム使用する．DAアルゴリズムは，差分プライバシを保証することによって発生する問い合わせ応答の誤差を低減する目的で使用する．提案手法では，Liらのアルゴリズムと同様に，ヒストグラムで表現されるデータに対するレンジクエリのみを対象とする．また，Liらのアルゴリズムは1次元と2次元のデータに対応しているのに対して，提案手法は1次元のデータを想定する．本項で示す完全準同型暗号下での差分プライバシ適用方法は，Liらが提案した差分プライバシアルゴリズムの一部であるDAアルゴリズムを完全準同型暗号下で実現することで達成される．

集約後の，ヒストグラムで表現される暗号文データに対して，単純にラプラスメカニズムを用いて差分プライバシを保証するナイーブな手法は，最低限以上の誤差を生む．Liら[6]のDAアルゴリズムでは，ヒストグラムに対して，値が近いまとまりごとに分割を行う処理であるパーティショニングを行い，分割されたまとまりごとのデータの合計値に対してノイズを加算する．ここで，値が近いまとまりごとに分割を行うとは，まとまり内のデータの偏差が小さくなるように分割することを意味する．個々のデータに対してノイズを加算するのではなく，いくつかの分割されたまとまりごとにノイズを加算することで，加算するノイズ量の合計値を低減する．真にまとまり内の偏差が最小になるパーティショニングを行うことは，プライバシの侵害をもたらすため，Liらの手法ではプライバシパラメータϵ_1を使用して，差分プライベートなパーティショニングを行う．分割されたまとまりごとのデータの合計値に対してノイズを加算する時のプライバシパラメータをϵ_2とすると，$\epsilon = \epsilon_1 + \epsilon_2$として，DAアルゴリズムにおける出力値は$\epsilon$-差分プライバシを満たす．

ヒストグラムにおける各評価項目，すなわち横軸をドメインと呼ぶ．ヒストグラムの縦軸は各ドメインに対応する数値を示し，この数値を数値データと呼ぶ．パーティショニングによって区切られるドメインのまとまりをバケットと呼ぶ．また，バケットの集合をパーティションと呼ぶ．ヒストグラムのドメイン数を$\boldsymbol{n}$とする．データ提供者から計算サーバへ送信される入力データを集約したヒストグラムの各ドメインの数値データを$\boldsymbol{x} = (\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_n-1}, \boldsymbol{x_n})$と定義する．$\boldsymbol{x}$はベクトルであり，$\boldsymbol{x}$の要素である$\boldsymbol{x_i}(\boldsymbol{1 \le i \le n})$は$\boldsymbol{i}$番目のドメインの数値データを示す．パーティションを$\boldsymbol{B} = \{\boldsymbol{b_1}, \boldsymbol{b_2}, \ldots, \boldsymbol{b_k-1}, \boldsymbol{b_k}\}$と定義する．$\boldsymbol{B}$は集合であり，$\boldsymbol{B}$

の要素である$b_j(1 \le j \le k)$はバケットを表し，j番目のバケットのドメインの集合である．具体的に，b_jが 3 番目のドメインから 6 番目のドメインまでの集合である場合は，$b_j = [3,4,5,6]$と表現される．1 つのパーティションにおいて，バケットごとの数値データの合計値を$S = (s_1, s_2, \ldots, s_k - 1, s_k)$とする．$S$はベクトルであり，$S$の要素である$s_j(1 \le j \le k)$を以下の式(4)に示す．

$$s\_j = \sum_{i \in b_j} x_i \qquad (4)$$

Sに対してラプラスメカニズムで生成されるノイズを加算する．差分プライバシが保証されたバケットごとの数値データの合計値を$S' = (s'_1, s'_2, \ldots, s'_k - 1, s'_k)$とする．計算サーバは，$S'$を復号サーバに送信する．復号サーバは共通鍵を用いて，S'を復号し，計算サーバに送信する．計算サーバは，復号されたS'を均一展開する．ここで，均一展開とは，s'_jをb_jの要素数で除算し，均一に展開する処理を指す．具体的に，$s'\_j = 10$，$b_j = [3,4,5,6]$である場合は，$(x'_3, x'_4, x'_5, x'_6) = (2.5, 2.5, 2.5, 2.5)$となる．計算サーバは，均一展開されたデータ$x' = (x'_1, x'_2, \ldots, x'_n - 1, x'_n)$を得る．$x'$を要約として，データ解析者からの問い合わせに応答する．ここで，x，Sは暗号文，B，x'は平文で表現される．S'は復号サーバで復号されるまでは暗号文で表現され，復号された後は平文で表現される．

以下の図 3 は$x = (3,2,6,5,6,3,4)$，$B = \{[1,2],[3,4,5],[6,7]\}$である時の例を示している．この時，$S = (5,17,7)$に定まる．差分プライバシ適用後のバケットごとの合計値を$S' = (4.6, 16.2, 7.8)$とすると，均一展開された データの値は$x' = (2.3, 2.3, 5.4, 5.4, 5.4, 3.9, 3.9)$となる．

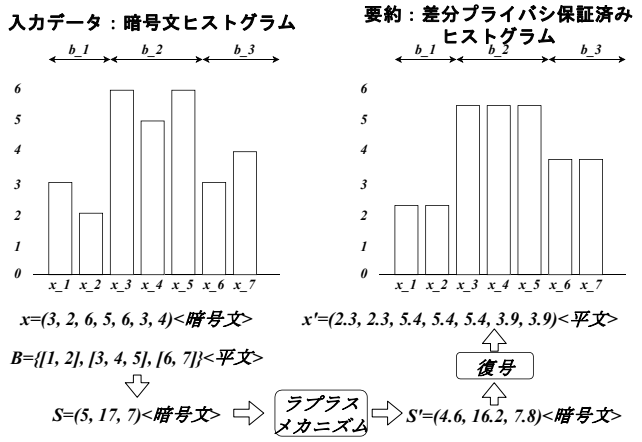

図 3　完全準同型暗号下での差分プライバシ適用

Li ら[6]が提案したパーティショニングを完全準同型暗号下で実現するためには，暗号文上で絶対値と最小値を取得する必要がある．平文上では，条件分岐を用いることで，絶対値と最小値を取得することができる．しかし，暗号文上では条件分岐を行うことができない．条件分岐を使用せずに絶対値や最小値を取得するために，提案手法では TFHE(Torus Fully Homomorphic Encryption)方式[10]を実装した TFHE ライブラリ[2] を使用する．TFHE ライブラリとは完全準同型暗号のライブラリの 1 つである．TFHE 方式の特徴はバイナリで表現される平文を暗号化しており，バイナリゲートごとの高速なブートストラッピングを実現していることである．TFHE 方式を用いることで，暗号化されたデータに対して，バイナリゲートで構成される任意の論理回路を構築することが可能となる．絶対値や最小値を取得する論理回路を，条件分岐を使用せずに構成することで，暗号文上で絶対値や最小値の暗号文を取得することができる．

5. 評価実験

本節では，提案手法を実験的に評価する．評価実験は，要約の構成にかかる時間と，構成された要約の精度の観点から行う．

5.1　実験環境

評価実験に用いられる実験プログラムは TFHE ライブラリを使用して，C++で記述されたものである．TFHE ライブラリは version1.1 を使用する．プログラム実行はシングルスレッドで行う．本評価実験を実施した際の実験環境を表 1 に示す．実験プログラムは固定小数点方式を採用している．また，2 の補数表現を使用している．固定小数点方式による実装おいて，表現することができない小数部分の値は切り捨てることとする．

評価実験で使用するプライバシパラメータの値は$\epsilon = 1.00$としている．ϵ_1とϵ_2の比は Li ら[6]と同様に 1 : 3 とする．すなわち，$\epsilon_1 = 0.25$，$\epsilon_2 = 0.75$となる．実験で使用される数値データは 0 から 10 の整数値をランダム生成することで取得している．数値データの上限値は計算過程でオーバーフローが起こらないようにするという観点から決定している．また，本評価実験では，数値データとして負の数を想定しないため，差分プラ

表 1　実験環境

名称	値
CPU モデル	Intel(R) Xeon(R) Platinum 8280
ソケット数	2
コア数	56
メモリサイズ	1.5TB
OS	CentOS Linux release 7.6.1810(Core)
Linux version	3.10.0-957.21.3
g++ version	7.3.1

[2] https://github.com/tfhe/tfhe

イバシにおけるノイズにより，差分プライバシ適応後の数値データが負の数となった場合は，その数値データを 0 に置き換えることとする．

5.2　要約構成時間の評価実験

提案手法は，データ解析者の問い合わせに対する応答速度を高速化する一方で，要約を構成するには時間がかかる．提案手法において，パーティショニングから差分プライバシ保証済みデータベースを構築するのにかかる時間を要約構成時間と呼ぶ．要約構成時間は，主にドメインサイズ及び TFHE において暗号文を表現するビット数に依存して変化する．本項の評価実験ではドメインサイズ及び暗号文を表現するビット数を変化させた時の要約構成時間の変化を測定する．

測定方法はドメインサイズ 2 から 6 までのヒストグラムにおいて 10(2)ビット，12(4)ビット，16(8)ビットの 3 種類のビット数で，要約構成時間をそれぞれ 10 回測定し，平均を算出するという方法で行う．ここで，ビット数の()内の数字は小数部分のビット数を示している．すなわち，10(2)ビットとは，符号部 1 ビット，整数部 7 ビット，小数部 2 ビットである．具体的に，要約構成時間として，パーティショニングの開始から，均一展開の終了までを測定した．要約構成時間の計測には C++の標準ライブラリに含まれている chrono を使用した．

ドメインサイズの変化に注目した要約構成時間の測定結果を以下の図 4 に示す．図 4 より，要約構成時間は，ドメインサイズに応じて指数的に増加することが確認された．提案手法では，ドメインサイズに応じて要約構成時間が指数的に増加するため，ドメインサイズが大きくなると要約構成時間が膨大となり，現実的には活用することが困難となることが分かった．

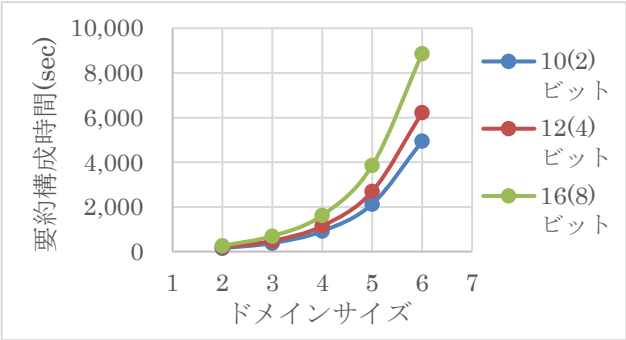

図 4　ドメインサイズごとの要約構成時間

次に，ビット数の変化に注目した要約構成時間の測定結果を図 5 に示す．図 5 より，要約構成時間は，暗号文を表現するビット数に応じて，線形に増加することが確認された．提案手法では，要約構成時間はビット数に応じて線形増加であるため，ビット数を増加させたとしても，現実的な時間で要約を構成することができることが分かった．

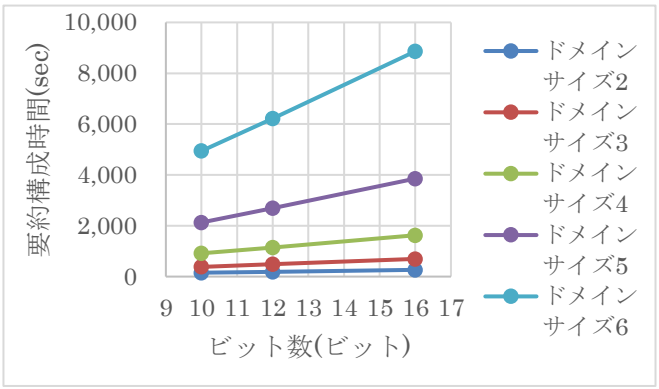

図 5　ビット数ごとの要約構成時間

図 4 から，提案手法はドメインサイズに応じて，指数的に増加することが確認された．図 5 から，提案手法ははビット数に応じて，線形に増加することが確認された．すなわち，提案手法はドメインサイズの増加に対応することは困難であるが，ビット数の増加に対応することは可能である．

5.3　精度の評価実験

本項では，提案手法における精度の観点から評価実験を行う．TFHE では浮動小数点方式を使用することが困難であるため，本実装は固定小数点方式を採用する．本実装では，小数部分を表現するビット数の大きさに応じて，小数部分を切り捨てている．小数部分のビット数が小さい場合は，切り捨てる小数部分の値が大きくなる．TFHE はバイナリゲートごとの高速なブートストラッピングを実現することで，任意回数の演算を可能にしている．そのため，ビット列で表現される数値を暗号化し，任意の演算を行った後，復号された計算結果の値は準同型暗号に起因するノイズを含まない．これは，暗号化することによって精度が低下しないことを表している．精度の低下は小数部分を切り捨てることで発生する．すなわち，提案手法では小数部分のビット数に応じて，要約の精度が変化する．本項の評価実験は，小数部分のビット数を変化させた時の，構成されるデータの要約の精度の変化を測定する目的で行われる．集約直後のデータを差分プライバシ適用前データと呼ぶ．小数部分のビット数を変化させた時に，差分プライバシ適用前データに対する，構成された要約のデータの誤差の変化を測定する．

本項の評価実験は，計算にかかる時間を短縮する目的で，平文のプログラムで実施する．平文のプログラムにおいても，TFHE を使用するプログラムと同様に，数値データをビット列で表現し，演算を論理回路で構成するため，精度は TFHE を使用したプログラムと比

較して変化しない．

実験方法はドメインサイズ 2 から 10 までのヒストグラムにおいて 10(2)ビット，12(4)ビット，16(8)ビットの 3 種類のビット数で，構成された要約と差分プライバシ適用前データの誤差をそれぞれ 100 回測定し，平均を取るという方法で行う．ここで，構成された要約と差分プライバシ適用前データの誤差は，各ドメインにおける誤差の偏差の合計を，ドメイン数で除算するという方法で算出する．また，小数部分を切り捨てることが精度に及ぼす影響を調べる目的で，提案手法と同様の処理を浮動小数点方式で行う平文のプログラムを実装し，同様の実験方法で精度を測定した．使用する浮動小数点方式は，符号部に 1 ビット，仮数部に 52 ビット，指数部に 11 ビット使用する，64 ビットで表現される．

精度の測定結果を図 6 に示す．提案手法において，小数部分のビット数を増加させると，表現することができる小数部分の値が真の値に近づくことから，精度が向上することが期待される．しかし，図 6 より，実験結果からビット数に応じた，精度の規則的な変化は確認することができなかった．原因として，提案手法において，プライバシパラメータ $\epsilon = 1.00$ の時に発生する，差分プライバシにおけるノイズの大きさに対して，数値データをビット列で表現することで切り捨てられる小数部分の大きさが無視できるほど小さいことが考えられる．これは，小数部分のビット数を変化することで発生する精度への影響は，大きくないことを表している．

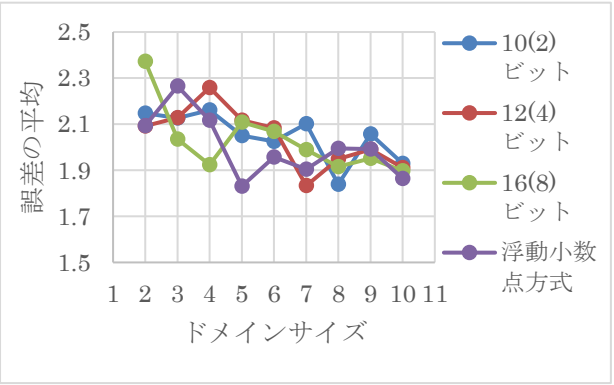

図 6　小数部分ビット数ごとの精度の比較

6. おわりに

本稿では，クラウドサーバとデータ解析者の両者に対してデータ提供者の所有するデータを保護することを目的として，準同型暗号と差分プライバシを組み合わせる問い合わせ応答システムについて議論した．本稿では，差分プライバシを保証したデータの要約を完全準同型暗号下で事前に構築し，平文下で要約を用いて問い合わせ応答を行う手法を提案した．データ解析者の問い合わせに応じて，平文上で演算を行うことで応答速度低下の問題を解決した．また，要約構成時に差分プライバシを保証したデータを用いて，全ての問い合わせに応答することで応答回数制限問題を解決した．提案手法はデータ解析者の問い合わせに高速に応答する一方で，要約構成に高い計算コストを必要とする．評価実験の結果から，ドメインサイズ 10 のヒストグラムにおいて，符号部 1 ビット，整数部 7 ビット，小数部 8 ビットの固定小数点方式を用いて要約構成を行うには，約 2.5 時間の時間がかかることが分かった．

提案手法のアルゴリズムは，ドメインサイズが大きい場合，計算量的観点から実用的ではない．今後の課題として，準同型暗号下で処理される計算を効率化することで，計算量を削減する必要があることが挙げられる．

謝辞　本研究の一部は，2020 年度国立情報学研究所 CRIS 共同研究の助成を受けています．

参　考　文　献

[1] Rivest, R., Adleman, L. and Dertouzos, M., “On data banks and privacy homomorphisms.” In Foundations of Secure Computation, Academic Press, pp.169-177, 1978.

[2] Dwork, C., McSherry, F., Nissin, K. and Smith, A., “Calibrating Noise to Sensitivity in Private Data Analysis,” Proc. of TCC 2006, pp.265-284, 2006.

[3] 牛山　翔二郎，工藤　雅士，髙橋　翼，井上　紘太朗，鈴木　拓也，山名　早人，“差分プライバシと準同型暗号の組み合わせに関する研究動向調査，” コンピューターセキュリティシンポジウム 2020，pp.207-214，2020.

[4] Chowdhury, A. R., Wang, C., He, X., Machanavajjhala, A. and Jha, S., “Cryptϵ: Crypto-Assisted Differential Privacy on Untrusted Servers,” Proc. of SIGMOD 2020, pp.603-619, 2020.

[5] Gentry, C., “Fully Homomorphic Encryption Using Ideal Lattices,” Proc. of the 41st annual STOC 2009, pp.169-178, 2009.

[6] Li, C., Hay, M., Miklau, G. and Wang, Y., “A Data- and Workload-Aware Algorithm for Range Queries under Differential Privacy,” Proc. of VLDB 2014, pp.341-352, 2014.

[7] 佐久間　淳，“データ解析におけるプライバシー保護，” pp.85-111，株式会社 講談社，2016.

[8] Dwork, C., “Differential Privacy,” Proc. of the 33rd ICALP 2006, LNCS, Vol.4052, pp.1-12, 2006.

[9] Hay, M., Machanavajjhala, A., Miklau, G., Chen, Y. and Zhang, D. “Principled evaluation of differentially private algorithms using DPBENCH,” Proc of SIGMOD 2016, pp.139-154, 2016.

[10] Chillotti, L., Gama, N., Georgieva, M. and Izabachene, M., “Faster Fully Homomorphic Encryption: Bootstrapping in less than 0.1 seconds,” Proc. of ASIACRYPT 2016, pp.3-33, 2016.