準同型演算による差分プライバシ保護処理の検討

関　萌乃　(指導教員：小口　正人)

1　はじめに

IoT デバイスの普及によりユーザから収集可能となった様々なデータの利活用においては，データ解析者が解析結果から個人の情報を推測できないように，適切なプライバシ保護を行う必要がある．また，データ収集者においても，外部からの攻撃やデータの不正利用の可能性があるため，暗号化などによりデータの保管や処理における機密性を保持しなくてはならない．

本研究では，暗号化した状態でのデータ処理を可能とする準同型暗号と，近年注目されているプライバシ保護基準である差分プライバシを組み合わせ，暗号変換を用いることで，クライアントへの負荷を低減しつつ，データ収集・蓄積・解析を安全に行えるシステムのデザインを示す．また，提案システムの構成要素として，準同型演算による差分プライバシの保護処理について検討する．

2　関連研究

2.1　準同型暗号

準同型暗号とは，データを暗号化した状態での演算が可能な暗号のことである．加法と乗法の両方に対応した暗号は，パラメータによりあらかじめ設定した任意回数の演算を行える Leveled 準同型暗号（LHE: Leveled Homomorphic Encryption）や，演算回数に制限のない完全準同型暗号（FHE: Fully Homomorphic Encryption）などに分類される．

2.2　差分プライバシ

差分プライバシは，2006 年に Dwork[1] により提唱されたプライバシの定義である．当初のアイデアである CDP（Centralized Differential Privacy）は，2 つの似たデータベースの解析結果が似た値となるよう，データ解析者に提供する解析結果にランダム化処理を施すものであった．これを拡張した局所差分プライバシ（LDP: Local Differential Privacy）では，クライアントはデータ解析者だけでなくデータ収集者も信頼せず，収集者に送信するデータそのものを加工する．

LDP のメカニズムに異なる入力を与えたとき，応答値が任意の出力の部分集合に含まれる確率の比は，プライバシパラメータ ϵ で定められる有限値で抑えられることが保証される．

定義 1 *ϵ-LDP*

任意の異なる入力 x, x' と，メカニズム M の出力の任意の部分集合 S について，

$$\frac{\Pr(M(x) \in S)}{\Pr(M(x') \in S)} \leq \exp(\epsilon)$$

ならば，M は *ϵ-LDP* を満たす（$\epsilon > 0$）．

メカニズムの応答値の有用性とプライバシ保護の程度はトレードオフの関係にあり，ϵ の値が大きいほど有用性は高く，プライバシ保護の程度は低くなる．

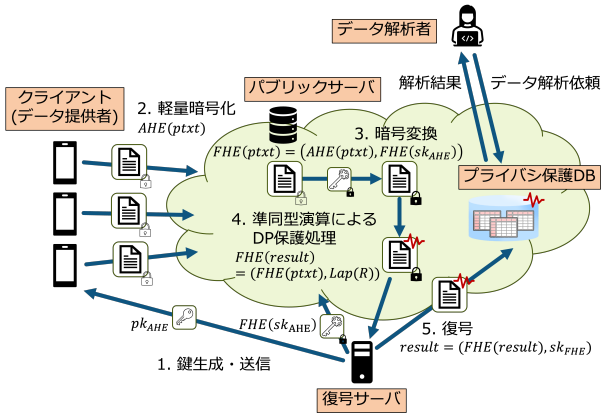

図 1: システム概要図

3　提案システム

3.1　既存手法の問題点と解決策

各クライアントがデータにノイズを加える LDP は，CDP と比べ解析結果の精度が低い．サーバ・クライアント間の対話回数を増やすことで解析精度は向上するものの，クライアントへの負荷は増加する．

そこで，提案システムでは，演算回数の自由度の高い LHE や FHE で暗号化したパブリックサーバ上のデータに繰り返しアクセスしながら，差分プライバシを保証したデータを生成する．データは暗号化されたままプライバシ保護処理を受けるため，LDP 同様，信頼のおけないデータ収集者が生のデータを閲覧することはない．一方，クライアントからのデータ収集は 1 回のみで，反復的な処理は計算資源の豊富なパブリックサーバ上で行うため，クライアントへの負荷を低減しつつ，対話可能性の高い LDP のモデルと同等の高精度な解析に対応可能である．

しかしながら，LHE や FHE は暗号文サイズが大きく，また暗号化処理に時間がかかる．このため，計算・通信処理能力の低い IoT デバイスにおいて，LHE や FHE によるデータの暗号化は実用的ではない．

これを解決するために，松本ら [2] が提案した，IoT デバイスではデータをより軽量な暗号を用いて暗号化し，パブリックサーバで LHE もしくは FHE への変換を行うシステムデザインを用いる．軽量暗号の例としては，暗号文の加算のみ可能な加法準同型暗号（AHE: Additive Homomorphic Encryption）が挙げられる．

3.2　システムの流れ

図 1 に示した提案システムの流れを以下に述べる．

まず，復号サーバは，準同型暗号（LHE や FHE など）および処理の軽い軽量暗号（AHE など）の公開鍵と秘密鍵のペアを生成する．IoT デバイスでは，軽量暗号の公開鍵を用いてデータを暗号化し，パブリックサーバに送信する．パブリックサーバは，準同型暗号で暗号化された軽量暗号の秘密鍵を用いて，軽量暗号による暗号化データを準同型暗号に変換する．その後，

表 1: 実験環境

サーバ	PowerEdgeR430
OS	Ubuntu 22.04.5 LTS
CPU	Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz
コア数	10
メモリ	64GB

準同型演算を用いて暗号化データに差分プライバシの保護処理を適用し，復号サーバとの通信を経てデータを復号する．こうしてパブリックサーバ上に構築されたプライバシ保護データベースに対し，データ解析者は任意の解析を行うことができる．

4 実験

4.1 実験概要

LDP を満たすメカニズムの一つであるラプラスメカニズムを準同型演算で実装し，単一サーバ上で実験を行った．サーバのマシンスペックは表 1 に示した通りである．

データセットとしては，Wine Quality[3] の白ワインのデータ（レコード数 4098）のうち，連続値を取る数値属性のみを使用した（属性数 $d = 11$）．また，各属性の値が $[-1, 1]$ の範囲に収まるように，事前にスケーリングを行い，Microsoft の準同型暗号ライブラリである SEAL[4] を用いて，実数を扱える CKKS 方式で各レコードを暗号化した．

実験では，暗号化データに平均 0，スケールパラメータ R のラプラス分布 $\mathrm{Lap}(R) = \frac{1}{2R}\exp(-\frac{|x|}{R})$ からサンプリングしたノイズを準同型加算した後，復号を行った．ここでは，LDP のプライバシパラメータ ϵ に対し，$R = \frac{2d}{\epsilon}$ に設定した．

各 ϵ の値に対し 10 回ずつ実験を行い，ノイズ加算および復号にかかる平均実行時間を測定した．同時に，出力データを用いて各属性の平均値推定を行い，以下で定義される平均値の推定値の L_∞ 誤差を求めた．

定義 2 L_∞ 誤差

属性 A_i の真の平均値を μ_i，平均値の推定値を μ'_i とする．$\boldsymbol{\mu} = (\mu_1, \mu_2, ..., \mu_d)$，$\boldsymbol{\mu'} = (\mu'_1, \mu'_2, ..., \mu'_d)$ とすると，平均値推定における出力データの L_∞ 誤差は，以下の式で定義される．

$$\|\boldsymbol{\mu'} - \boldsymbol{\mu}\|_\infty = \max_{1 \le i \le d} |\mu'_i - \mu_i|$$

4.2 結果

図 2 には，ノイズ加算，復号，およびそれらの合計の平均実行時間 [s] を示した．ϵ の値にかかわらず，各項目の平均実行時間はほぼ一定であり，ノイズ加算は約 37 秒，復号は約 9 秒である．また，ノイズ加算と復号の合計は約 46 秒で，1 レコードあたりに換算すると約 0.0057 秒となることがわかる．このノイズ加算と復号の合計時間に，クライアントでの軽量暗号化，パブリックサーバでの暗号変換，各エンティティ間の通信にかかる時間を加えた時間が，プライバシ保護手法としてラプラスメカニズムを用いた際の提案システムの合計実行時間となることが推測される．

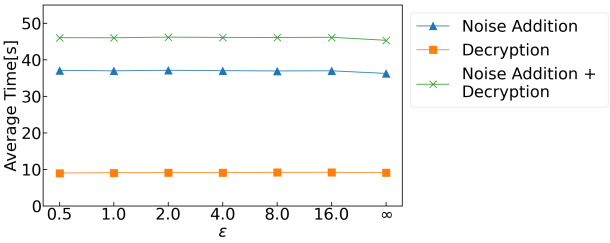

図 2: 平均実行時間

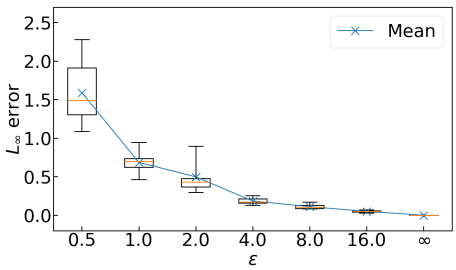

図 3: 平均値推定における L_∞ 誤差

図 3 には，平均値推定における L_∞ 誤差を示した．青色の線は L_∞ 誤差の平均値，オレンジ色の線は中央値を表している．ϵ の値が増加するほど，平均値推定における L_∞ 誤差，および四分位範囲は減少している．

5 まとめと今後の課題

IoT デバイスでデータを軽量暗号化し，パブリックサーバで準同型暗号へ変換した後，準同型演算による差分プライバシ保護処理を行うことで，IoT デバイスに対して低負荷，かつ安全なデータベースシステムを提案した．また，単一サーバ上で準同型演算を用いて LDP のラプラスメカニズムを実装し，実行時間と平均値推定における誤差を評価した．

今後は，上記の実装の各工程を各エンティティに割り振り，通信を含めた実行時間を測定する予定である．また，より高精度な解析を実現し，平均値推定以外のタスクにも対応可能な差分プライバシ保護手法の検討を行う．

参考文献

[1] Cynthia Dwork. Differential privacy. In *International colloquium on automata, languages, and programming*, pp. 1–12. Springer, 2006.

[2] Marin Matsumoto and Masato Oguchi. IoT Device Friendly Leveled Homomorphic Encryption Protocols. In *Proc. the 8th IEEE International Conference on Smart Data(SmartData2022)*, pp. 525–532, 2022.

[3] Cerdeira A. Almeida F. Matos T. Cortez, Paulo and J. Reis. Wine Quality. UCI Machine Learning Repository, 2009. DOI: https://doi.org/10.24432/C56S3T.

[4] Microsoft SEAL (release 4.1). `https://github.com/Microsoft/SEAL`, January 2023. Microsoft Research, Redmond, WA.