辞書を用いた圧縮センシングによる準同型暗号文の通信量削減

理学専攻・情報科学コース　泉 湖雪

1　はじめに

準同型暗号は暗号文同士の加算や乗算が可能であり，データを秘匿したままで分析し活用可能な技術の一つである．準同型暗号の応用例としては暗号化データベースや暗号化回帰分析が挙げられる．クライアントが暗号化したデータをデータベースサーバで集計/分析し，データ分析者は集計/分析結果だけを得られるようになる．しかしながら，このようなアプリケーションを想定した場合，準同型暗号は平文に比べ暗号文サイズが大きいため，クライアント・データベースサーバ間の通信量が大きくなる．本研究では辞書を用いた圧縮センシングによる通信量削減を提案する．評価の結果，圧縮率を 50%に設定した場合は，圧縮しない場合に比べて 50%通信量を削減できることを示した．また，カウントクエリの誤差，ロジスティック回帰による 2 値分類の精度，画像の誤差によって圧縮・復元したデータの有用性を示した．

2　システム概要

提案手法を用いたシステムの流れを図 1 に示す．復元アルゴリズムには，一般逆行列による L_2 ノルムの最小化を用いた．

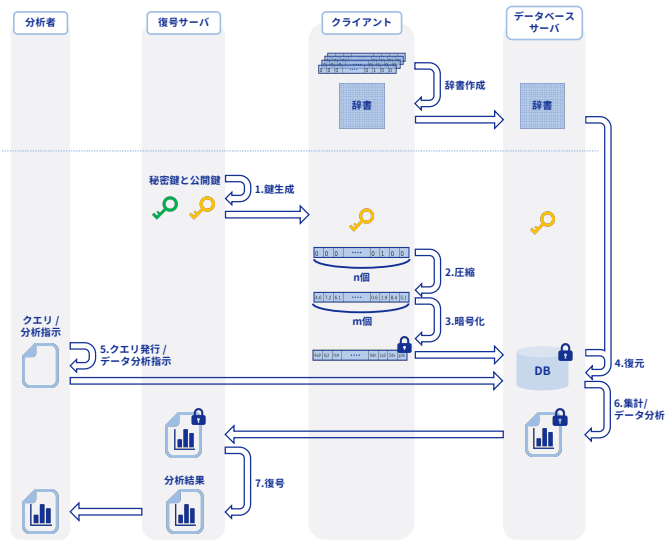


図 1: 提案手法を用いたシステムの流れ．
システムは，復号サーバでの鍵生成，クライアントでの圧縮・暗号化，事前に送信された辞書を用いたサーバでの復元，分析者からのデータ分析の指示，データベースサーバでのデータ分析，復号サーバでの復号といった流れで行われる．

3　評価

3.1　実験概要

SEAL ライブラリを使用して，Baseline と提案手法を C++で実装し，データセットを利用して評価した．ここで Baseline とは，単純に多次元データをクライアントで準同型暗号化して，圧縮せずにサーバに送信する方法を指す．評価指標として通信量 (暗号文サイズ)，復元精度，実行時間の 3 つを用いた．

3.2　実験設定

実験環境：実験に使用したマシンの性能を表 1 に示す．

表 1: マシン性能

	Raspberry Pi	サーバ
OS	Raspbian 11	Ubuntu 22.04.1
CPU	ARM Cortex-A72	Intel®Xeon®Gold 5115
コア数	4	10
プロセッサ速度	1.8GHz	2.4GHz
RAM	4GB	192GB

データセット：実験では復元精度を評価するために，カウントクエリ，ロジスティック回帰分析，画像生成を行った．以下に使用したデータセットを示す．

- カウントクエリ：POS データ[1]を使用．各カラムを分割し，それぞれ 2 進数に変換した．
- ロジスティック回帰分析：
  3 種類のデータセットを使用．正規化したデータを用いて以下のような 2 値分類を行った．
  - Wine quality dataset[2]：ワインが赤/白かを予想
  - NHANES III dataset[3]：高血圧かどうかを予想
  - Adult dataset[4]：年収が 5 万ドル超えかを予想
- 画像生成：画像データセット MNIST[5] を使用．

準同型暗号に関するパラメータ：表 2 に示す．

3.3　実験結果

3.3.1　通信量

暗号化前と暗号化後での圧縮：暗号化後に Zip コマンドで圧縮したところほとんどサイズ減少しなかった．準同型暗号文は同じ文字が何回も出現するという特徴を持たないためであり，暗号化前に圧縮する必要がある．

圧縮センシングの効果：図 2 にクライアントからサーバに送信する準同型暗号文のサイズを示す．Baseline に比べ提案手法では約半分の通信量に抑えられている．n/m の値を大きくすると通信量削減量は増加するが，同時に精度の低下にもつながる．

3.3.2　復元誤差

カウントクエリ：`param1` を用いて，提案手法による復元結果と Baseline に対して 5 つのカウントクエリを実行し，誤差を二乗平均平方根誤差 (RMSE: Root Mean Squared Error) で評価する．結果を図 3 に示す．RMSE は値が小さいほど誤差が少なく精度が高いことを表す．このときの 5 つそれぞれのカウントクエリの RMSE の平均は，学習量 50%で約 0.2 であり，実際のデータとの間に大幅な誤差は見られなかった．

ロジスティック回帰分析：`param2` を用いて，提案手法による復元結果と Baseline に対してロジスティック回

[1]https://www.mdpi.com/2306-5729/4/2/67
[2]http://www3.dsi.uminho.pt/pcortez/wine/
[3]https://rdrr.io/rforge/LogisticDx/man/nhanes3.html
[4]https://doi.org/10.24432/C5XW20
[5]https://yann.lecun.com/exdb/mnist/

表 2: SEAL の CKKS 方式に関するパラメータ．
param1 はカウントクエリや画像生成に，param2 はロジスティック回帰分析に使用される．

	param1	param2
暗号文の長さ slot_count	4096	8192
セキュリティ	128 bit	
Level	2	4
復号時の精度	60 bit	
復号時の実数値の精度	整数部分 20 bit 小数部分 20 bit	

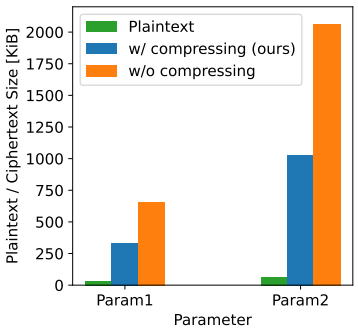


図 2: 暗号文のサイズ．
$n : m = 2 : 1$ で実行．Baseline と提案手法の通信量の比率は n と m の比率に対応しており，ここではサイズが $m/n = 1/2$ に削減されている．例えば，$n = 1024, m = 512$ と $n = 128, m = 64$ での通信量削減量は同等である．

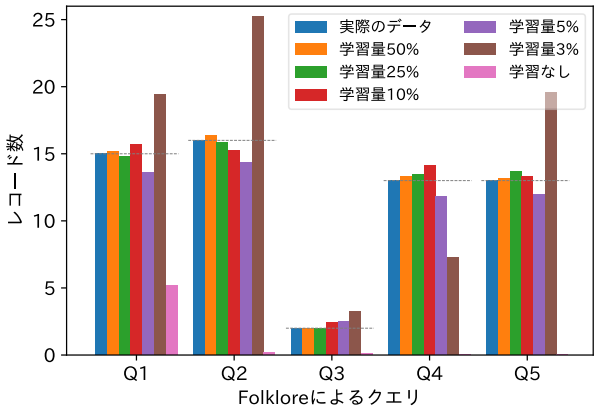


図 3: 圧縮率 50%でカウントクエリを実行した結果．
param1，$(n, m) = (256, 128)$ で実行．学習量を増やすほど精度が高くなっている．

帰分析を行い，2 値分類結果を比較することで評価する．結果を図 4 に示す．Baseline と提案手法での 2 値分類には，約 0.3%–1.6%の誤差が見られ，大幅な誤差は見られなかった．提案手法により復元したデータは予測に用いることが十分に可能であるといえる．

画像生成：param1 を用いて，提案手法による復元結果と Baseline(それぞれ画素データ) に対して画像を生成し，誤差をピーク信号対雑音比 (PSNR: Peak Signal to Noise Ratio) で評価する．結果を表 3 に示す．PSNR は値が大きいほど画像の劣化が少なく精度が高いことを表す．復元画像の PSNR の平均は約 32.2 であり，これは一般的に「拡大すると劣化がわかるが，通常の使用では問題ない品質」とされる．

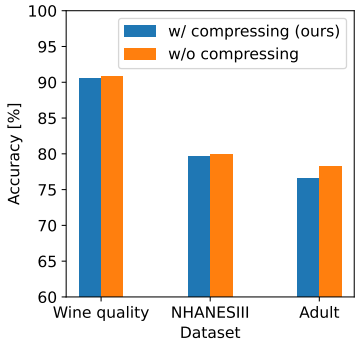
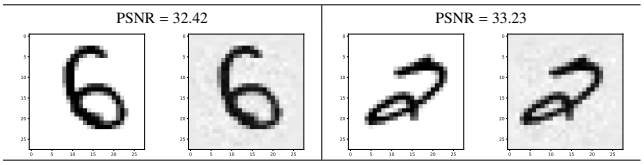


図 4: ロジスティック回帰分析で 2 値分類をした結果の比較． param2，$(n, m) = (16, 8)$ で実行．2 値分類の精度において，Baseline と提案手法の間で約 0.3%–1.6%の違いが見られた．

表 3: 元の画像と，提案手法により復元した画素から生成した画像． 各左側が元画像，各右側が復元画像．param1，$(n, m) = (1024, 512)$ で実行．全体的に，数字部分はしっかりと復元されており，背景部分がグレーがかっている．



3.3.3 実行時間

圧縮行列のサイズと実行時間：復元アルゴリズムでの準同型演算回数は m に比例するため，長いベクトルよりも短いベクトルを複数復元する (n, m を小さくする) ほうが高速であり適している．

圧縮の有無と実行時間：カウントクエリ，ロジスティック回帰分析，画像生成それぞれにおける実行時間を表 4 に示す．n, m の大きさを小さくするほど，トータルの平文の長さは同じでも，圧縮や復元に要する時間は短くなる．また，n, m の値が小さい場合，連結作業の増加の影響で暗号化に要する時間が増加する．

表 4: 1 ベクトルあたりの実行時間．
カウントクエリ，ロジスティック回帰分析，画像生成は，それぞれ $(n, m) = (256, 128), (16, 8), (1024, 512)$ で実行．

		Client		Server
		圧縮	暗号化	復元
カウントクエリ	圧縮あり (提案手法)	3.75 [ms]	2.1875 [ms]	5.75 [s]
	圧縮なし	–	3.75 [ms]	–
ロジスティック回帰分析	圧縮あり (提案手法)	0.0195 [ms]	0.71 [ms]	1.128 [s]
	圧縮なし	–	0.34 [ms]	–
画像生成	圧縮あり (提案手法)	72.625 [ms]	8 [ms]	94.5 [s]
	圧縮なし	–	15.25 [ms]	–

4 まとめ

本研究ではクライアント・サーバ間の通信量削減を目的として，圧縮センシングを用いて準同型暗号文サイズ削減に取り組んだ．提案手法で復元したデータは予測に用いることが十分に可能であることを示し，実行時間より通信量を重視する場合に有用であるといえる．

謝辞

本研究は一部，JST CREST JPMJCR22M2 の支援を受けたものである．