準同型暗号を用いたデータベースシステムの処理性能向上に関する研究

理学専攻・情報科学コース　内藤 華

1　研究背景

近年データ利用が急速に拡大しており，機密データのセキュリティ対策が求められている．準同型暗号という暗号化手法を用いると，データを復号せずに直接演算を行うことが可能となる．準同型暗号の中でも，暗号化状態での加算と乗算を任意の回数行うことのできるものを，完全準同型暗号（以下 FHE）という．FHE には処理コストや暗号分のサイズが大きいといった課題があり，実用化に向けてストレージや計算の効率化が求められている．本研究では，エンコード手法やデータ構造，データ圧縮手法の最適化により，FHE を用いたデータ解析の処理性能向上を目指す．

2　準同型暗号

準同型暗号とは，暗号化されたデータに対して直接演算を行うことのできる暗号技術である．従来の暗号化手法では，データを暗号化しても分析や計算処理に利用する際には復号が必要となり，その過程で秘匿性が損なわれる可能性がある．一方で，準同型暗号という暗号化手法を用いると，暗号化状態のままで安全に計算を実行することが可能となる．準同型暗号には，加算あるいは乗算どちらか一方の計算が可能な部分準同型暗号，どちらも計算可能であるが乗算の回数に制限のある Somewhat 準同型暗号，任意の回数の加算と乗算が可能な FHE などが存在する．FHE を用いる際の基本的な操作は，以下の通りである．

KeyGen 暗号システムのセキュリティ強度を決定するパラメータ λ を基に，公開鍵と秘密鍵を作成する．公開鍵はデータの暗号化，秘密鍵は復号にそれぞれ用いられる．

Enc 公開鍵を利用して平文のデータを暗号文に変換する．

Dec 秘密鍵を用いて暗号文を再び平文に戻し，最終的な結果を取り出す．

Eval 暗号文のデータに対して直接演算処理を実行する．暗号化状態で加算や乗算，暗号文を回転させる Rotation などの計算を行うことができる．多くの FHE は，複数の暗号文に対して単一の操作を同時に実行することのできる SIMD 演算をサポートしている．パッキングした暗号文では特定の要素のみに対する処理を行うことが困難であるため，Rotation を活用し暗号文内のデータを適切な位置に移動させる．FHE では，計算を繰り返すと暗号文にノイズが蓄積されていくため，bootstrapping という操作によってこのノイズをリセットする．

3　比較演算に適したエンコード手法

3.1　実験概要

比較演算は，データ解析において重要な演算の一つであるが，計算の複雑さや FHE を用いることによるオーバヘッドなどの課題が存在する．本実験では，比較演算における計算コストを削減するため，各シナリオに適したデータのエンコーディング手法について検討する．以下の 2 種類のエンコーディング手法で整数データのエンコードを行い，最大値関数とフィルタリング関数の性能評価を行う．

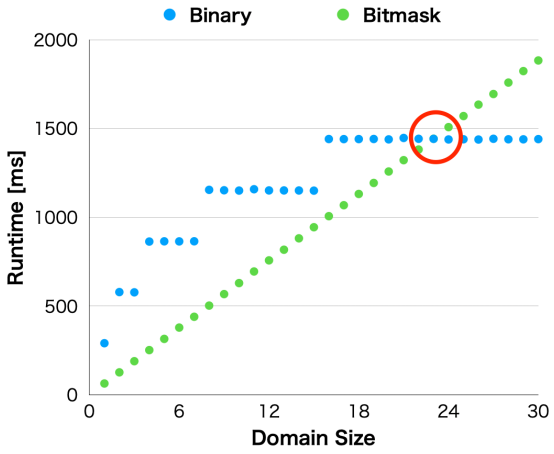


図 1: 最大値関数におけるドメインサイズと実行時間の関係

- バイナリエンコーディング
  数値データを 2 進数のビット列で表現する．
- ビットマスクエンコーディング
  エンコードする値と同じ個数のビットを 1 にセットする．例えば，属性 $Age \in [1, 75]$ が 22 である場合，$[\underbrace{1,\ldots,1}_{22},\underbrace{0,\ldots,0}_{53}]$ のように表現される．

3.2　実験結果

図 1 は，各ドメインサイズにおける最大値関数の実行時間を示している．ドメインサイズが 23 未満の範囲では，ビットマスクコードの実行時間がバイナリコードよりも短く効率的であることが読み取れる．しかし，ビットマスクエンコーディングではビット長が線形に増加するため，ドメインサイズの大きい数値データに対しては，バイナリエンコーディングの方が効率的であると考えられる．フィルタリング関数においても，同様の傾向が見られた．

4　ゲノム情報検索に適した FHE ライブラリ

4.1　実験概要

ゲノムデータには個人の遺伝的特徴など機密性の高い情報が含まれているため，ゲノム解析を行う際には FHE によるプライバシの保護が有用である．本稿では，Positional-Burrows Wheeler Transform というデータ構造を用いることで，暗号化状態で効率的なゲノム情報検索を実現する．FHE ライブラリとして PALISADE，およびその後継の OpenFHE で実装を行い，ゲノム情報検索に適した暗号ライブラリについて考察する．

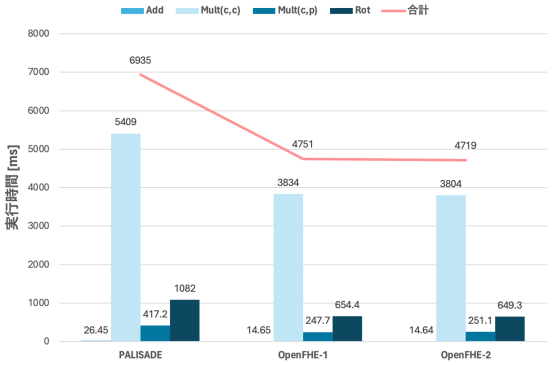
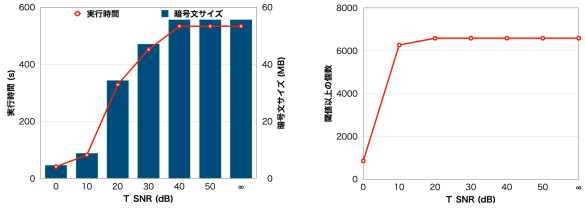


図 2: ゲノム秘匿情報検索処理全体における実行時間の内訳



(a) SNR による圧縮効果　(b) 検出した信号数

図 3: 閾値解析の結果

4.2 実験結果

PALISADE および OpenFHE の BFV 方式を用いた際の演算処理全体の実行時間と，FHE ライブラリごとの加算，乗算，Rotation の実行時間の内訳を，図 2 に示す．OpenFHE については，PALISADE と同様の演算子を利用した実装と，OpenFHE で新たに追加された演算子を用いた実装の 2 通りについて実行時間の計測を行った．それぞれの実装を OpenFHE-1，OpenFHE-2 として図中に示している．OpenFHE では，計算の際に加わるノイズの低減や乗算のアルゴリズムが改善されており，OpenFHE-1 の実行時間は，PALISADE と比較して 30%程度短縮された．また，OpenFHE から新たに導入された EvalAddMutable() や EvalMultMutable() などの演算子を用いた OpenFHE-2 では，メモリ使用量が大幅に削減された．

5 SNR を用いた周波数成分の圧縮

5.1 実験概要

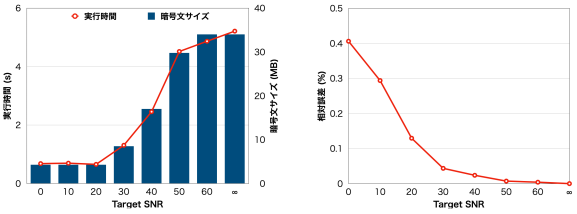
周波数解析において個人や企業の情報に関わるような機密性の高いデータを扱う際には，FHE を用いたプライバシの保護が有用である．本稿では，有効な信号とノイズの比率を示す指標である信号対雑音比（以下 SNR）に基づいてノイズを除去することで，暗号化状態におけるストレージと周波数解析の効率化を目指す．時系列データと顔画像データの周波数成分に対して SNR を用いて圧縮処理を行い，さらに周波数領域で閾値解析と顔画像の類似度解析を実行する．

5.2 閾値解析

家庭の電力消費量に関する時系列データの周波数成分に対して，暗号化状態で閾値解析を実行した．圧縮後の暗号文のデータサイズと解析の実行時間の関係を，図 3a に示す．なお，$T_{SNR} = \infty$ は圧縮をせず元データを用いた場合を表している．結果より，$T_{SNR} \leq 30$ dB の範囲でデータサイズおよび実行時間が削減されていることがわかる．図 3b は，閾値検索で検出されたデータポイント数を表しており，$T_{SNR} \geq 20$ dB の範囲で全ての要素が正しく検出できることが確認できた．すなわち，T_{SNR} を 20~30 dB 程度に設定することで，閾値解析の精度を維持しながら計算効率も向上させることが可能であると考えられる．

表 1: 各 T_{SNR} におけるコサイン類似度

T_{SNR}	0	10	20	30	40	50	60	∞
画像 A’	**0.988**	**0.987**	**0.985**	**0.984**	**0.984**	**0.984**	**0.984**	**0.984**
画像 B	0.957	0.952	0.949	0.948	0.948	0.947	0.947	0.947
画像 C	0.986	0.979	0.975	0.972	0.972	0.972	0.972	0.972
画像 D	0.987	0.979	0.975	0.971	0.971	0.971	0.971	0.971
画像 E	0.952	0.944	0.939	0.938	0.938	0.937	0.937	0.937
画像 F	0.978	0.976	0.969	0.967	0.967	0.967	0.967	0.967



(a) SNR による圧縮効果　(b) 圧縮なしとの誤差

図 4: 顔画像類似度解析の結果

5.3 類似度解析

クライアントが保持する顔画像 A とサーバの保持する顔画像 A’ および B~F が同一人物であるかを検証するため，周波数成分同士のコサイン類似度を計算した．なお，画像 A’ は，画像 A と同一人物の表情の異なる画像とする．表 1 に，各 T_{SNR} における画像 A とのコサイン類似度の計算結果を示す．全ての T_{SNR} において，画像 A’ の類似度が最も高い結果となった．これは，データ量を削減しても画像 A と画像 A’ が同一人物であると正しく認識できていることを示している．圧縮後の暗号文データサイズ，類似度解析の実行時間の関係を図 4a に示す．$T_{SNR} \leq 40$ dB の範囲でデータサイズおよび実行時間が 50%以上削減された．データ圧縮を行わずに類似度解析を実行した結果との差異を相対誤差として評価し，その結果を図 4b に示す．T_{SNR} の増加に伴い相対誤差が減少し，元データでの解析結果に近づいていることが読み取れる．効率と精度の両方の観点から，T_{SNR} は 20~40 dB 程度が適していると考えられる．

6 まとめと今後の課題

本研究では，FHE を用いたデータ解析の効率化を目的とし，3 種類の実験を行った．データや解析の特性に応じて適切なエンコーディング手法，FHE ライブラリ，データ圧縮手法を選択することにより，データサイズや計算コストを大きく削減することができた．今後は，他のデータ形式や解析手法へ応用することで，提案手法の汎用性を評価したい．