6Y-01

クラウド環境におけるゲノム秘匿検索に向けた暗号スキームの比較

山田 優輝 † 小口 正人 †
† お茶の水女子大学

1. はじめに

近年ゲノムデータの統計解析が可能になり，医療分野に留まらず様々な分野でゲノムデータ利用の実用化が期待されている．ゲノムデータを統計処理する際には大型のストレージと計算機が必要になるため，クラウドを用いたゲノムデータ委託システムが普及していくと考えられる．この際プライバシ保護が必須となるが，暗号化されたデータ同士での演算が可能な完全準同型暗号を用い，クラウドに秘密鍵を渡すこと無く秘匿演算を行う秘匿検索手法が近年盛んに研究されている．完全準同型暗号を用いたゲノム秘匿検索システムを実現する際には，システムデザインやアルゴリズムだけでなくデータ構造や暗号スキーム，また採用する暗号ライブラリなど様々な要素が影響し合うため，これらを総合的に分析・評価する必要がある．本研究では，クラウド上で行われる完全準同型暗号を用いたゲノム秘匿検索演算の性能を左右する要素について，主に暗号スキーム・暗号ライブラリに着目して分析する．

2. 完全準同型暗号

以下の式 (1) 及び式 (2) のように暗号文同士での加算，乗算が成立する性質をそれぞれ加法準同型性，乗法準同型性と言う．完全準同型暗号 FHE はこの両方の性質を持ち合わせた暗号化手法である．

加法準同型性，乗法準同型性

$$Encrypt(m) \oplus Encrypt(n) = Encrypt(m+n) \quad (1)$$

$$Encrypt(m) \otimes Encrypt(n) = Encrypt(m \times n) \quad (2)$$

FHE は公開鍵暗号方式の機能を持つが，秘密鍵を用いることなく暗号文同士の演算を行い，平文同士の演算を暗号化した値を導くことが出来るため，ユーザは平文上で行うのと同様に暗号文同士での加法演算・乗法演算を行うことが出来る．課題としては計算量が大きいことの他に，暗号文に含まれるノイズが演算の度に増加し，閾値を越えると正しく復号することが出来なくなる，というものが挙げられる．演算の回数を限定するか，bootstrap と呼ばれる計算量は大きいがノイズをリセットすることが出来る手法を導入することで復号を保証することが出来る．

完全準同型暗号の実用化が期待されるようになるにつれ，複数の実装が公開されるようになった．本研究では現在広く利用されているライブラリの一つである HElib[1] とより新しい実装でありアクティブに開発されている PALISADE[2] とを用いる．

Comparison of Encryption Schemes for Privacy-Preserving Genome Search in Cloud Environment
† Yuki Yamada, † Masato Oguchi
Ochanomizu University (†)

3. 先行研究

石巻らによる先行研究 [3] 及び [4] で提案されたシステムデザインをそれぞれデザイン 1，デザイン 2-1 とする．デザイン 1 では一文字分の問い合わせを繰り返すことで最終的な結果を得ることでサーバ上での FHE の計算量を削減しているが，クエリ長が増大するとともにクライアントでの計算負荷が増加する．また，毎回の通信で大容量のデータが転送されるため，通信ネットワークに負荷をかけ，計算資源の乏しいクライアントには適さない．これに対してデザイン 2-1 では，クエリの文字列長に関わらず一往復の通信で検索を行うことが出来る．これはデザイン 1 よりも計算資源の乏しいクライアントには適しているが，暗号文の容量やサーバ上での毎回の FHE 演算の計算量はデザイン 1 と比べて増大してしまう．デザイン 2-1 では bootstrap を導入しているが，代わりに大きなパラメータを用いる手法を挙げることも出来る．本研究ではこれをデザイン 2-2 とする．

4. 実験及び分析

本研究では暗号スキーム・暗号ライブラリに着目し，HElib[1] が提供する BGV[5] と PAILSADE[2] が提供する BFV[6] とに関する比較実験を行う．サンプルとして一塩基多型を並べた SNP 配列を 1,000 サンプル用いる．それぞれのサンプルの長さは 10,000 文字とする．クエリ長は 1 から 8 もしくは 20 まで，指定するポジション（検索開始位置）の数は 1 から 8 まで変化させて計測実験を行う．また，いずれの実験においても Smart et al. による パッキング技術 [7] を利用する．

実験に用いた環境を表 1 に，パラメータを表 2 に示す．

表 1: 実験環境

Server	OS	CentOS 6.10
	CPU	Intel®Xeon®Processor E5-2643 v3 (3.4GHz) 6 Cores × 2 Sockets
	Main Memory	512GB
	SSD	80GB
	HDD	2TB

表 2: パラメータ

	デザイン 1	デザイン 2-2
BGV (HElib)	*level = 9*	*level = 9* * クエリ長
BFV (PALISADE)	*NumMults = 15*	*NumMults = min(50, 9* * クエリ長*)*

クエリ長によるサーバ上での実行時間の比較結果を図 1 に，クエリ長によるクライアント上での実行時間の比較結果を図 2 に示す．また，クエリ長によるサーバからクライアントへのデータ転送量の比較結果を図 3 に，クエリ長によるクライアントからサーバへのデータ転送量の比較結果を図 4 に示す．

Copyright ©2020 Information Processing Society of Japan.
All Rights Reserved.

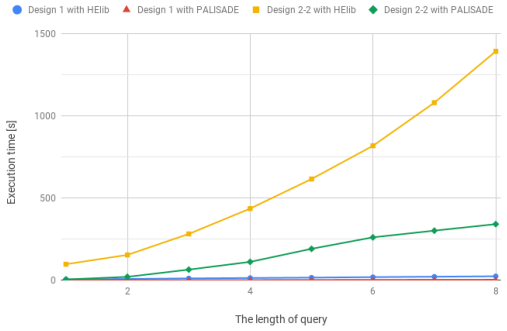

図 1: クエリ長による暗号ライブラリごとのサーバ上での実行時間の比較

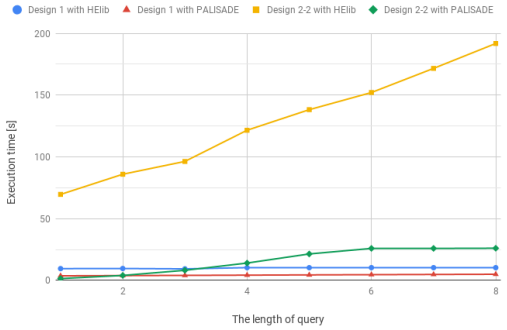

図 2: クエリ長による暗号ライブラリごとのクライアント上での実行時間の比較

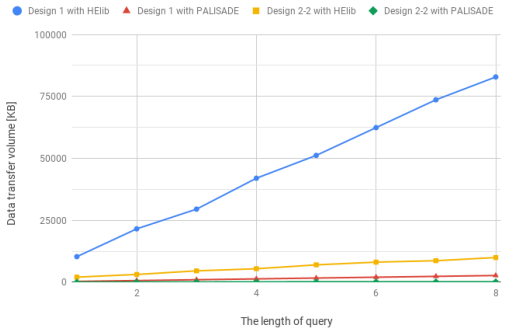

図 3: クエリ長による暗号ライブラリごとのサーバからクライアントへのデータ転送量の比較

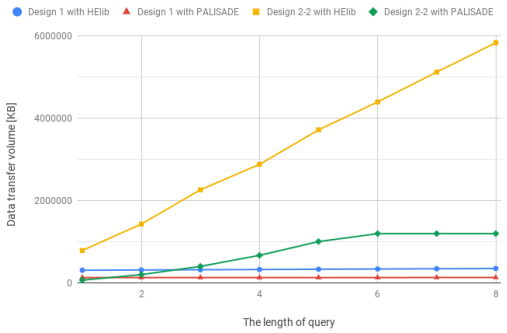

図 4: クエリ長による暗号ライブラリごとのクライアントからサーバへのデータ転送量の比較

実行時間についての実験結果を示した図 1 及び図 2 より，本システムにおいては HElib により提供される BGV よりも PALISADE により提供される BFV の方が高速であることが読み取れる．同様に，データ転送量についての実験結果を示した図 3 及び図 4 の各グラフより，本システムにおいては HElib により提供される BGV よりも PALISADE により提供される BFV の方がデータ転送量を少なく抑えることが出来ることが読み取れる．

以上の実験結果より，本システムにおいては HElib により提供される BGV よりも PALISADE により提供される BFV の方が高いパフォーマンスを発揮すると考えることが出来るが，PALISADE は bootstrap をサポートしていないため，bootstrap が必須となるシステムにおいては PALISADE ではなく HElib を採用せざるを得ない場合も考えられる．

5. 結論

先行研究に基づき，完全準同型暗号を用いたゲノム秘匿検索システムを二種類のデザイン及び二種類の暗合スキーム・ライブラリを用いて実装し，クラウドコンピューティングを想定した環境下で実験を行った．また得られた実験結果について，システムデザインと暗号スキーム・ライブラリの観点から分析を行った．その結果，クエリ長やポジション数などによって適するデザインが変わること，本アプリケーションでは PALISADE により提供される BFV スキームが良い性能を示すことが確認された．今後はデザイン 1 とデザイン 2 を組み合わせたシステムデザインを提案するなど，実用化に向けた取り組みを行っていきたい．

謝辞

本研究は JST CREST JPMJCR1503 の支援を受けております．

参考文献

[1] homenc, *Helib: An implementation of homomorphic encryption*, `https://github.com/homenc/HElib/`, visited on 12/2019.

[2] PALISADE, *Palisade homomorphic encryption software library*, `https://palisade-crypto.org/software-library/`, visited on 12/2019.

[3] Y. Ishimaki, K. Shimizu, K. Nuida, and H. Yamana, “Poster: Privacy-preserving string search for genome sequences using fully homomorphic encryption,” in *IEEE Symposium on Security and Privacy*, 2016.

[4] Y. Ishimaki, H. Imabayashi, K. Shimizu, and H. Yamana, “Privacy-preserving string search for genome sequences with fhe bootstrapping optimization,” in *2016 IEEE International Conference on Big Data (Big Data)*, IEEE, 2016, pp. 3989–3991.

[5] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, “Fully homomorphic encryption without bootstrapping,” *IACR Cryptology ePrint Archive*, vol. 2011, p. 277, 2011.

[6] J. Fan and F. Vercauteren, “Somewhat practical fully homomorphic encryption,” *IACR Cryptology ePrint Archive*, vol. 2012, p. 144, 2012.

[7] N. P. Smart and F. Vercauteren, “Fully homomorphic simd operations,” *Designs, codes and cryptography*, vol. 71, no. 1, pp. 57–81, 2014.

Copyright ©2020 Information Processing Society of Japan.
All Rights Reserved.