A Brief Description of Our Submissions for Task 1 of IDASH 2023

The LARC Team*

August 16, 2023

1 Introduction

We have three entities:

- A query entity (QE): QE holds genomes of target individuals, containning 400 genomes with the genotypes for $16,344$ genetic variants.
- A database owner (DE): DE contains $2,000$ genomes with genotypes for the same set of $16,344$ variants.
- A computing entity (CE): CE performs genome detection using the encrypted data from QE and DE, non-colluding with QE or DE.

The goal is to decide if any of the query individuals from QE are related to any individuals in the genomic database from DE.

The three entities do as follows:

- QE initializes for a homomorphic encryption (HE) scheme, including generating secret key, public key, and keys for further HE operations.
- QE using the public key encrypts the private query genotype data.
- DE using QE's public key encryts the genotype data of the database.
- CE homomorphically evaluates a query mechanism, using QE's encrypted data and the public information of the HE scheme, and returns an encrypted score for each query individual.
- QE decrypts the encrypted results.

*In alphabetic order: Jingwei Chen, Xiaokang Dai, Xiangyu Hu, Rui Li, Weijie Miao, Qingyu Mo, Haoyong Wang, Shuai Wang, Wenyuan Wu, Chen Yang, Linhan Yang, Haonan Yuan, and Jindou Zhang are affiliated with Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 400714, China; Jia Liu and Li Yang are affiliated with Sansure Biotech, Changsha 410221, China.
Email: chenjingwei@cigit.ac.cn, wuwenyuan@cigit.ac.cn

2 Encryption Scheme

For encryption, decryption, and homomorphic evaluation, we use the Brakerski-Gentry-Vaikuntanathan (BGV) scheme [2] implemented in Microsoft SEAL [3].

Parameters. Let $R = \mathbb{Z}[X]/\langle X^n + 1\rangle$. The plaintext and ciphertext space of BGV is $R/p\mathbb{Z}$ and $R/q\mathbb{Z}$, respectively.

- The plain modulus $p = 33538049 \approx 2^{25}$.
- The polynomial degree $n = 2^{13} = 8,192$.
- The bit size of the ciphertext modulus $\log q \approx 200 = 41+39+40+40+40$.
- The standard derivation of the error distribution is set to be the default value, i.e., $\frac{8}{\sqrt{2\pi}} \approx 3.19$.

Security. According to the latest version of Lattice Estimator [1], the set of above parameters ensures that the security parameter archieves 128.

3 Query Mechanism

We, the LARC team, propose two different mechanisms, which we call Average-Max and Minority-Max, respectively. Note that both mechanisms are enlighted by the "Single-SNP averages" method presented in [4]. For convenience, we fix the following notations:

- $\mathbf{Q} \in \mathbb{Z}^{16,344\times 400}$: the query data.
- $\mathbf{A} \in \mathbb{Z}^{2,000\times 16,344}$: the database data.
- $\mathbf{r} \in \mathbb{Z}^{1\times 400}$: the resulting scores for the 400 query individuals.

3.1 Submission 1: Average-Max

The basic idea of this mechanism is to replace max by the negative average, based on the observations on the challenging data. The scores $\mathbf{r} = \mathbf{1}(\mathbf{A} - \mathbf{E})\mathbf{Q}$, where all entries of $\mathbf{1} \in \mathbb{Z}^{1\times 2,000}$ and $\mathbf{E} \in \mathbb{Z}^{2,000\times 16,344}$ are 1. Although the matrix $\mathbf{E}$ is indeed a public parameter of the Average-Max query mechanism, we encrypt it and deal with it in its encrypted form as well.

3.2 Submission 2: Minority-Max

The basis idea is to replace max by the minor commponent. The resulting scores $\mathbf{r} = (\mathbf{1}(10 \cdot \mathbf{A}) - \mathbf{u})\mathbf{Q}$, where $\mathbf{u} \in \mathbb{Z}^{1\times 16,344}$ is a constant vector (somehow like $\mathbf{1}$) stored in `model/u.txt` (relative to repository root). As the matrix $\mathbf{E}$ in the

Average-Max mechanism, we read and encrypt the public parameter **u** and deal with it in its encrypted form. [1]

4 Compile and Run

4.1 Dependencies

- `cmake` 3.13 or higher
- `clang`
- `openmp`
- Microsoft `SEAL`

4.2 Compile

Download the source code from GitHub, unzip and enter the repository root directory, and then run the following:

```
cd submission_X
mkdir build
cd build
cmake ..
make
```

where `X` should be specified to '`1`' for Average-Max or '`2`' for Minority-Max.

4.3 How to Use

- Query genotype data must be renamed as `QUERY_SITE_genotypes.txt`, and must be put in the `CHALLENGE_DATA_DIR` directory (relative to repository root).
- Database genotype data must be renamed as `DATABASE_SITE_genotypes.txt`, and must be put in the `CHALLENGE_DATA_DIR` directory (relative to repository root).
- Open a terminal, enter the `submission_X/build` directory (relative to repository root, and '`X`' should specified to `1` or `2`), and run

  ```
  ./IDASH23
  ```

- The above command produces a file in the `submission_X/build` directory (relative to repository root, and `X` should specified to `1` or `2`) named `result.txt`. The resulting file consists of 400 rows and 1 column. Each row is the score of the corresponding query individual.

[1] We note that the vector **u** can be interpreted as a model parameter, which is trained from the training data, and is also expected to work for unknown data to be classified, predicted, or queried.

5 Experimental Results

We test our code on a PC with Intel i9-12900K CPU, 32GB DDR5 memory, and Ubuntu 22.04 system. The performance is recorded as in the following table, where the score is computed by AUC/exp(total time in minutes / 5). All the following statistics are based on the challenge data published on http://www.humangenomeprivacy.org/2023/competition-tasks.html.

Submission	Memory Used (MB)	Total Time (sec.)	AUC	Score
1	3426.66	4.10	0.85	0.8384
2	3434.73	4.18	0.99	0.9763

References

[1] M. R. Albrecht, R. Player, and S. Scott. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 9(3):169–203, 2015. https://doi.org/10.1515/jmc-2015-0016. Lattice Estimator: https://github.com/malb/lattice-estimator. 2

[2] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In S. Goldwasser, editor, *Proc 3rd ITCS (January 8–10, 2012, Cambridge, MA, USA)*, pages 309–325. ACM, New York, 2012. https://doi.org/10.1145/2633600. 2

[3] Microsoft. Microsoft SEAL (release 4.1.1), Accessed in July, 2023. https://github.com/microsoft/SEAL. 2

[4] D. Speed and D. J. Balding. Relatedness in the post-genomic era: is it still useful? *Nature Reviews Genetics*, 16:33–44, 2014. https://doi.org/10.1038/nrg3821. 2