Secure Searching of Biomarkers Using Hybrid GSW Encryption Scheme

Jung Hee Cheon, Miran Kim, Yongsoo Song

Seoul National University, South Korea

iDASH Privacy & Security Workshop, November 11, 2016

Table of contents

Motivation

Track 3: Testing for Genetic Diseases

- Database $\mathrm{Chr}[i] \in \{1, 2, \ldots, 22, X(=23), Y(=24)\}$, $\mathrm{POS}[i]$
- Corresponding nucleic acid sequence $\mathrm{SNPs}[i] \in \{A, T, G, C\}^*$
- Goal: find a query genome in database.

Encoding of database

- We make the use of 1-to-1 functions
  - $(\mathrm{Chr}[i], \mathrm{POS}[i]) \mapsto d_i = \mathrm{Chr}[i] + 24 \cdot \mathrm{POS}[i] \in \mathbb{Z}_{2^{32}}$.
  - $\mathrm{SNPs}[i] \mapsto \alpha_i \in \mathbb{Z}$.
- Check if there is an index k such that $(d, \alpha) = (d_k, \alpha_k)$.

Problem: comparison is expensive in Homomorphic Encryption

Motivation

Track 3: Testing for Genetic Diseases

- Database $\mathrm{Chr}[i] \in \{1, 2, \ldots, 22, X(=23), Y(=24)\}$, $\mathrm{POS}[i]$
- Corresponding nucleic acid sequence $\mathrm{SNPs}[i] \in \{A, T, G, C\}^*$
- Goal: find a query genome in database.

Encoding of database

- We make the use of 1-to-1 functions
  - $(\mathrm{Chr}[i], \mathrm{POS}[i]) \mapsto d_i = \mathrm{Chr}[i] + 24 \cdot \mathrm{POS}[i] \in \mathbb{Z}_{2^{32}}$.
  - $\mathrm{SNPs}[i] \mapsto \alpha_i \in \mathbb{Z}$.
- Check if there is an index k such that $(d, \alpha) = (d_k, \alpha_k)$.

Problem: comparison is expensive in Homomorphic Encryption

RLWE public-key encryption

- Cyclotomic Ring
  - $\mathcal{R} = \mathbb{Z}[X]/\Phi_m(X)$ for an integer m (: power of two).
  - $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$ is the residue ring modulo an integer q.
- KeyGen:
  - $sk \leftarrow (1, s)$ for a small s.
  - $pk \leftarrow (b, a)$ generated by $a \leftarrow \mathcal{R}_q$, $b = -as + e$ for a small e.
- Encryption: $\vec{c} \leftarrow \text{RLWE.Enc}(m)$
  - $\vec{c} \leftarrow v \cdot pk + (\frac{q}{t}m + e_0, e_1)$ for small e_1, e_2 and v.
  - $\langle \vec{c}, sk \rangle = \frac{q}{t}m + e \pmod q$ for some small e.
  - Free to convert RLWE encryption of $m = \sum_i m_i X^i$ into a LWE encryption of m_0

GSW encryption [GSW13, DM15]

Encryption: $C \leftarrow \mathrm{GSW.Enc}(m)$:

- A $2k \times 2$ matrix $(\vec{c}_0, \vec{c}_1) \leftarrow (-s \cdot \vec{a} + \vec{e}, \vec{a}) + m \cdot G$ for a small $\vec{e}$

  and the Gadget matrix $G = \mathcal{P}_B(1) \otimes I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ \vdots & \vdots \\ B^{k-1} & 0 \\ 0 & B^{k-1} \end{bmatrix}$

- An encryption C of m satisfies $C \cdot sk = m \cdot \mathcal{P}_B(sk) + \vec{e}$.

Multiplication of GSW & RLWE ciphertexts [CGGI16]

- GSW ciphertexts *act on* RLWE ciphertexts.

$$\begin{array}{rccccc} \text{Mult}: & \{\text{GSW ctxts}\} & \times & \{\text{RLWE ctxts}\} & \to & \{\text{RLWE ctxts}\} \\ & C \in \mathcal{R}_q^{2k\times 2} & , & \vec{c} = (c_0, c_1) \in \mathcal{R}_q^2 & \mapsto & WD_B(\vec{c}) \cdot C \end{array}$$

- If $C \cdot sk = m' \cdot \mathcal{P}_B(sk) + \vec{e}$ and $\langle \vec{c}, sk \rangle = \frac{q}{t} m + e$, then

$$\langle \vec{c}_{\text{mult}}, sk \rangle = (WD_B(\vec{c}) \cdot C) \cdot sk = WD_B(\vec{c}) \cdot (C \cdot sk) = \frac{q}{t} mm' + e^*$$

for $e^* = m'e + \langle WD_B(\vec{c}), \vec{e} \rangle$.

- $\vec{c}_{\text{mult}}$ is a RLWE encryption of mm' with the error e^*.

Encryption of VCF Files & Query Data

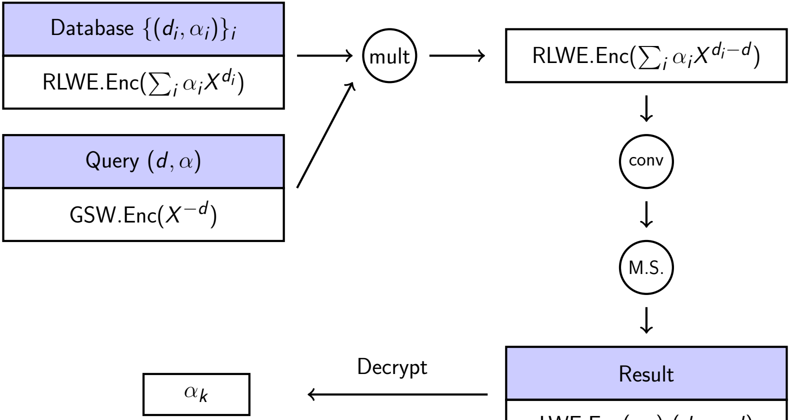
- Database file is encoded into $\{(d_i, \alpha_i) : 1 \leq i \leq \ell\}$. Construct the polynomial
$$\mathrm{DB}(X) = \sum_i \alpha_i X^{d_i},$$
and use the RLWE encryption scheme. Store the ciphertext $\vec{c}_{\mathrm{DB}}$.
- Use symmetric-key GSW scheme for encoded query (d, α). Encrypt the polynomial $X^{-d} = -X^{n-d}$ and send the ciphertext C_{Q} to the server.

Query Computation: Searching and Extraction

Given $\vec{c}_{\mathrm{DB}} \leftarrow \mathrm{RLWE.Enc}(\sum_i \alpha_i X^{d_i})$ and $C_{\mathrm{Q}} \leftarrow \mathrm{GSW.Enc}(X^{-d})$,

1. Compute $\vec{c}_{\mathrm{res}} \leftarrow \mathrm{Mult}(C_{\mathrm{Q}}, \vec{c}_{\mathrm{DB}}) \ \ (= \mathrm{RLWE.Enc}(\sum_i \alpha_i X^{d_i - d}))$.
2. Convert it into a LWE ciphertext, which is an encryption of α_k if $d_k = d$ for some k; otherwise an encryption of random value.
3. Carry out the modulus-switching to reduce the size of resulting LWE ciphertexts and communication cost.
4. Decrypt the LWE ciphertexts and compare with α.

Query Computation: Searching and Extraction



Database $\{(d_i, \alpha_i)\}_i$

RLWE.Enc($\sum_i \alpha_i X^{d_i}$)

Query (d, α)

GSW.Enc(X^{-d})

mult

RLWE.Enc($\sum_i \alpha_i X^{d_i - d}$)

conv

M.S.

Result

LWE.Enc(α_k) $(d_k = d)$

Decrypt

α_k

Optimization technique

- Construction of a single polynomial yields huge $n > 2^{31}$,
  $\Rightarrow$ take $n = 2^{16}$ and divide d_i into two 16-bit integers $d_{i,1}$, $d_{i,2}$.
- Size of the encoded nucleic acid sequences α_i is too large to be encrypted in a single ciphertext (e.g. 41 bits).
  - Split α_i into smaller integers
    $\Rightarrow$ smaller plaintext space $t = 2^{11}$ and modulus $q = 2^{32}$.
  - The use of variable type '*int32_t*' accelerates the speed of implementation and basic C++ std libraries.

#(SNPs)	Size	Complexity				Storage	
		Q-enc	DB-enc	Eval	Dec	DB	Res
5	10K	0.14s	0.11s	0.67s	0.15ms	1MB	0.25MB
	100K		0.27s	1.64s	0.29ms	2.5MB	0.625MB
20	10K		0.45s	2.75s	0.41ms	4MB	1MB
	100K		1.04s	6.88s	0.84ms	10MB	2.5MB

#(SNPs): maximal number of SNPs considered for comparison

Intel Core i5 running at 2.9 GHz processor

Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène.
Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds.
to be appeared in ASIACRYPT, 2016.

Léo Ducas and Daniele Micciancio.
Fhew: Bootstrapping homomorphic encryption in less than a second.
In *Advances in Cryptology–EUROCRYPT 2015*, pages 617–640. Springer, 2015.

Craig Gentry, Amit Sahai, and Brent Waters.
Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based.
In *Advances in Cryptology–CRYPTO 2013*, pages 75–92. Springer, 2013.