SUPPLEMENTARY MATERIAL
FACT: Fast and Accelerated FHE with Threshold Decryption for Real-Time Applications

I. Introduction

We provide some low-level details of our work in this document. We would like to emphasize that these details are not necessary to get an understanding of our work. But this document can be helpful to interested readers, as it contains **(i)** some detailed mathematical background, **(ii)** formal proofs of our observation of various patterns in the distribution matrix and the secret shares generated during the execution of Benaloh-Leichter Linear Integer Secret Sharing Scheme (LISSS), **(iii)** a diagram illustrating various functions of Torus-FHE library, **(iv)** some minute details of our prototype software implementation of threshold FHE, **(v)** some elaborated discussion on the steps used to implement homomorphic K-nearest neighbours (KNN) algorithm over Torus-FHE library for a real-world large medical dataset. For the ease of explanation, we have frequently referred to various sections of our main paper in this document.

II. Basic Definitions and Mathematical Background

In this section we provide some preliminary background material on cryptographic primitives used in this paper.

Access Structure. Access structures are commonly encountered in many different cryptographic primitives, including secret-sharing [1]–[3], and are heavily referenced in this paper. So we recall the informal definition here. Let $\mathcal{P} = \{P_1, \dots, P_T\}$ be a set of T parties, and suppose that some secret k is shared among them, such that only certain "qualified" subsets of the T parties can collaboratively recover k by combining their key shares. Informally, this defines an *access structure* over $\mathcal{P}$, which is nothing but the collection of all such "qualified" subsets $\overline{\mathcal{P}} \subseteq \mathcal{P}$ of parties that can recover the secret k by combining their key shares.

Threshold Access Structure. In this paper, we focus primarily on threshold access structures. For any $T, t \in \mathbb{N}$ such that $t \leq T$, a (t,T)-threshold access structure over any set $\mathcal{P} = \{P_1, \dots, P_T\}$ is defined as a collection of qualified subsets of the form

$$\mathbb{A}_{(t,T)} = \{\overline{\mathcal{P}} \subseteq \mathcal{P} : |\overline{\mathcal{P}}| \geq t\},$$

which (informally) states that any subset with t or more parties is a qualified subset. Finally, if $\mathbb{A}_{(t,T)}$ is a *minimal* (t,T)-threshold access structure, then it only consists of subsets of size exactly t; in other words, we have $|\mathbb{A}_{(t,T)}| = \binom{T}{t}$.

Monotone Boolean Formula (MBF). A Boolean formula is monotone if it has a single output and it consists of only AND and OR combination of Boolean variables. It turns out that any (t,T)-threshold access structure $\mathbb{A}_{(t,T)}$ can be represented by a monotone Boolean formula. If $x_1, \dots, x_T$ are T Boolean variables corresponding to T parties involved in secret sharing, then any AND combination of the form $(x_{i_1} \wedge \ldots \wedge x_{i_t})$ represents collaboration of t parties together, where $i_1, \dots, i_t \in [1, T]$. As each such collaboration is capable of reconstructing the secret individually, $\mathbb{A}_{(t,T)}$ can be represented by OR combination of $\binom{T}{t}$ such terms.

Rényi Divergence. Let $Supp(P)$ and $Supp(Q)$ denote the supports of distributions P and Q respectively, such that $Supp(P) \subseteq Supp(Q)$. For $a \in (1, +\infty)$, the Rényi divergence of order a is defined as

$$R_a(P||Q) = \left(\sum_{x \in Supp(P)} \frac{P(x)^a}{Q(x)^{a-1}} \right)^{\frac{1}{a-1}}.$$

In addition, the Rényi divergence of order 1 and $+\infty$ are defined as

$$R_1(P||Q) = \exp\left(\sum_{x \in Supp(P)} P(x) log \frac{P(x)}{Q(x)} \right).$$

$$R_\infty(P||Q) = max_{x \in Supp(P)} \frac{P(x)}{Q(x)}.$$

The definition extends in a natural way to continuous distributions; see [4] for details.

Gaussian Distribution. Let, $\omega \geq 1$ and $\rho \in \mathbb{R}^+$, now $\forall x, \mu \in \mathbb{R}^\omega$, the Gaussian function with mean μ and standard deviation ρ is denoted by $f_{\mu,\rho}(x) = \exp(-||x-\mu||^2/2\rho^2)$. Let $\mathbb{H}$ be a subset of $\mathbb{R}^\omega$, then $f_{\mu,\rho}(\mathbb{S})$ is defined in [5] as follows:

$$f_{\mu,\rho}(\mathbb{S}) = \begin{cases} \sum_{x \in \mathbb{S}} f_{\mu,\rho}(x), & \text{if } \mathbb{S} \text{ is discrete,} \\ \int_{x \in \mathbb{S}} f_{\mu,\rho}(x) \cdot dx, & \text{if } \mathbb{S} \text{ is continuous.} \end{cases}$$

Throughout this paper, when we refer to Gaussian distribution, we refer to the discrete Gaussian distribution.

III. Observing the Pattern of Secret Shares

We state our observation on the pattern of the secret shares, generated by the (t,T)-threshold secret sharing using Benaloh-Leichter LISSS (Section IV-C of our main paper), in the form of a theorem and provide the corresponding proof here.

Theorem 1. *$\mathcal{P}' = \{P_{id_1}, P_{id_2}, \dots, P_{id_t}\} \subset \mathcal{P} = \{P_1, P_2, \dots, P_T\}$ is a t-sized group with* group_id *value of gid, where $id_1 < id_2 < \cdots < id_t$. $\forall 1 \leq i \leq t$, P_{id_i} has a key share SH_i, tagged with* group_id *value of gid. Then all key shares except SH_1, have only binary coefficients in their k polynomials, while SH_1 will have coefficient value upper-bounded by t in its k polynomials.*

In order to prove Theorem 1, we will first state two lemmas related to the structure of the distribution matrix M for (t,T) threshold secret sharing of a TRLWE secret key S. We consider the number of polynomials in S is k and I_k denotes the identity matrix of dimension k.
The first lemma is about the pattern of the distribution matrix for Boolean formula of the form $x_1 \wedge x_2 \wedge \cdots \wedge x_t$ for any t.

Lemma 1. *We consider* $\mathbf{0}$ *to be a notation of zero matrix of dimension* $k \times k$*. Then, distribution matrix* M_f *for Boolean formula* $f = x_1 \wedge x_2 \wedge \cdots \wedge x_t$ *follows the following structure.*

$$\begin{bmatrix} I_k & I_k & I_k & \dots & I_k & I_k \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & I_k \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & I_k & \mathbf{0} \\ \vdots \\ \mathbf{0} & \mathbf{0} & I_k & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & I_k & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \end{bmatrix}_{kt \times kt}$$

Proof of Lemma 1. We prove the lemma by induction on the value of t.
For $t = 1$, $f = x_1$ and $M_f = I_k$. Hence, the stated matrix structure is satisfied by default.
For $t = 2$, $f = x_1 \wedge x_2$. We follow the ANDing procedure (see Section IV-C in the paper) of $M_{x_1} = I_k$ and $M_{x_2} = I_k$ and get $M_{x_1 \wedge x_2} = \begin{bmatrix} I_k & I_k \\ \mathbf{0} & I_k \end{bmatrix}$, which clearly satisfies the claimed structure.
Let us assume that the claimed structure of the distribution matrix holds for $t = i$, i.e., for $f = x_1 \wedge x_2 \wedge \cdots \wedge x_i$, M_f is as shown below. Also, x_{i+1} being a Boolean variable, $M_{x_{i+1}} = I_k$. ANDing M_f and $M_{x_{i+1}}$ produces $M_{f_1} = M_{f \wedge x_{i+1}}$ as shown below. M_f has a dimension of $ki \times ki$ and M_{f_1} has a dimension of $k(i+1) \times k(i+1)$.

$$M_f = \begin{bmatrix} I_k & I_k & I_k & \dots & I_k & I_k \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & I_k \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & I_k & \mathbf{0} \\ \vdots \\ \mathbf{0} & \mathbf{0} & I_k & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & I_k & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \end{bmatrix}$$

$$M_{f_1} = \begin{bmatrix} I_k & I_k & I_k & I_k & \dots & I_k & I_k \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & I_k \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & I_k & \mathbf{0} \\ \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & I_k & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_k & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & I_k & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \end{bmatrix}$$

Clearly, the structure is maintained for $t = i + 1$. Hence, by induction, the lemma is true for any $t \geq 1$. $\square$

And the second lemma is about the pattern of distribution matrix for Boolean formula consisting of disjunction of l number of such t-sized conjunctive terms, i.e., $(x_{1,1} \wedge x_{1,2} \wedge \cdots \wedge x_{1,t}) \vee \cdots \vee (x_{l,1} \wedge x_{l,2} \wedge \cdots \wedge x_{l,t})$.

Lemma 2. *Let us assume that* $f' = (x_{1,1} \wedge x_{1,2} \wedge \cdots \wedge x_{1,t}) \vee \cdots \vee (x_{l,1} \wedge x_{l,2} \wedge \cdots \wedge x_{l,t})$ *is a Boolean formula, where* $\forall 1 \leq i \leq l, 1 \leq j \leq t$, $x_{i,j}$ *is a binary variable and each of the* $(x_{i,1} \wedge x_{i,2} \wedge \cdots \wedge x_{i,t})$ *terms is represented by distribution matrix* M_f*, as stated in Lemma 1. We denote first* k *columns of* M_f *by* F *of dimension* $kt \times k$ *and the rest of the columns of* M_f *by* R *of dimension* $kt \times k(t-1)$*.* $\mathbf{0}$ *denotes zero matrix of dimension* $kt \times k(t-1)$*. Then distribution matrix* $M_{f'}$ *has the following structure:*

$$\begin{bmatrix} F & R & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ F & \mathbf{0} & R & \mathbf{0} & \dots & \mathbf{0} \\ \vdots \\ F & \mathbf{0} & \dots & \mathbf{0} & R & \mathbf{0} \\ F & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & R \end{bmatrix}_{lkt \times (lkt-(l-1)k)}$$

Proof of Lemma 2. We prove the lemma by induction on the value of l.
For $l = 1$, $f' = f = (x_{1,1} \wedge x_{1,2} \wedge \cdots \wedge x_{1,t})$ and $M_{f'} = M_f = \begin{bmatrix} F & R \end{bmatrix}$, which satisfies the claimed structure by default.
For, $l = 2$, $f' = (x_{1,1} \wedge x_{1,2} \wedge \cdots \wedge x_{1,t}) \vee (x_{2,1} \wedge x_{2,2} \wedge \cdots \wedge x_{2,t})$. We perform ORing on $M_{x_{1,1} \wedge x_{1,2} \wedge \cdots \wedge x_{1,t}} = M_f$ and $M_{x_{2,1} \wedge x_{2,2} \wedge \cdots \wedge x_{2,t}} = M_f$ (see Section IV-C in the paper) and get

$$M_{f'} = \begin{bmatrix} F & R & \mathbf{0} \\ F & \mathbf{0} & R \end{bmatrix}_{2kt \times (2kt-k)}$$

This structure follows the lemma.
Let us assume that the structure is maintained $\forall l \leq j$. So, with $f' = (x_{1,1} \wedge x_{1,2} \wedge \cdots \wedge x_{1,t}) \vee \cdots \vee (x_{j,1} \wedge x_{j,2} \wedge \cdots \wedge x_{j,t})$ and $f'' = (x_{j+1,1} \wedge x_{j+1,2} \wedge \cdots \wedge x_{j+1,t})$, $M_{f'}$ has a dimension of $jkt \times jkt - (j-1)k$ and $M_{f''}$ has a dimension of $kt \times kt$. $M_{f'}$ follows the structure as shown below. $M_{f''} = \begin{bmatrix} F & R \end{bmatrix}$. Now, ORing $M_{f'}$ and $M_{f''}$ produces $M_{f_2} = M_{f' \vee f''}$ with dimension $(j+1)kt \times ((j+1)kt - jk)$ as shown below.

$$M_{f'} = \begin{bmatrix} F & R & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ F & \mathbf{0} & R & \mathbf{0} & \dots & \mathbf{0} \\ \vdots \\ F & \mathbf{0} & \dots & \mathbf{0} & R & \mathbf{0} \\ F & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & R \end{bmatrix}$$

$$M_{f_2} = \begin{bmatrix} F & R & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ F & \mathbf{0} & R & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \vdots \\ F & \mathbf{0} & \dots & \mathbf{0} & R & \mathbf{0} & \mathbf{0} \\ F & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & R & \mathbf{0} \\ F & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & R \end{bmatrix}$$

So, the lemma is true for $l = (j+1)$.
Hence, by induction the lemma is true for any $l \geq 1$. $\square$

Now we use Lemma 1 and Lemma 2 to provide here the proof of Theorem 1.

Proof of Theorem 1. Let us recall from Section IV-C of the paper that the monotone Boolean formula for (t,T)-threshold secret sharing can be written as $f=(x_{1,1}\wedge x_{1,2}\wedge\cdots\wedge x_{1,t})\vee\cdots\vee(x_{l,1}\wedge x_{l,2}\wedge\cdots\wedge x_{l,t})$, where $l=\binom{T}{t}$. If $\mathbf{0}$ denotes zero matrix of dimension $kt\times(kt-k)$, from Lemma 1 and Lemma 2, we know that structure of the corresponding distribution matrix M with dimension $\binom{T}{t}kt\times(\binom{T}{t}kt-(\binom{T}{t}-1)k)$ is as follows:

$$M=\begin{bmatrix} F & R & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0}\\ F & \mathbf{0} & R & \mathbf{0} & \dots & \mathbf{0}\\ \vdots & & & & & \\ F & \mathbf{0} & \dots & \mathbf{0} & R & \mathbf{0}\\ F & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & R\end{bmatrix}$$

$$F=\begin{bmatrix} I_k\\ \mathbf{0}\\ \mathbf{0}\\ \vdots\\ \mathbf{0}\end{bmatrix}\qquad R=\begin{bmatrix} I_k & I_k & I_k & \dots & I_k\\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & I_k\\ \vdots & & & & \\ \mathbf{0} & I_k & \mathbf{0} & \dots & \mathbf{0}\\ I_k & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0}\end{bmatrix}$$

A detailed look into the above matrix M reveals that F has a structure of dimension $kt\times k$ and R has a structure with dimension $kt\times(kt-k)$ as shown in above matrix structure. In F and R, $\mathbf{0}$ denotes a zero matrix of dimension $k\times k$. It is obvious from the structure of M that each of its $\binom{T}{t}$ horizontal sections contain exactly one F and one R along with $(\binom{T}{t}-1)$ zero matrices $\mathbf{0}_{kt\times(kt-k)}$. Now, the structure of F shows that each of its first k rows contains one '1' entry. No other row below has any '1' in it and the structure of R reveals that each of its first k rows contains exactly $(t-1)$ number of '1' in it. Each of the other rows below contains exactly one '1' in it. Hence, each of the first k rows of any one horizontal section (out of total $\binom{T}{t}$ sections) of M has exactly t number of '1' in it. Each of the rest of the rows below in that section contains exactly one '1' in it.

Let us recall that, each section of M corresponds to one section of $shares$ ($shares=M\cdot\rho$ from Section IV-C in the paper, i.e, the key shares of any t-sized subset of collaborating parties.

ρ is a binary matrix. During matrix multiplication, dot product between one row of M and one column of ρ produces an entry in $shares$. Dot product between two binary vectors is always upper bounded by the number of '1' in any of the two vectors. As, each of first k rows of any section of M contains exactly t number of '1', the entries of first k rows of any section in $shares$ are always upper bounded by t. First k rows of any section of $shares$ form one key-share. Clearly, that key share will have non-binary entries in it. Similarly, each of the other $(kt-k)$ rows below in any section of M contains exactly one '1', so the entries of the $(kt-k)$ number of rows below in any section of $shares$ are upper bounded by 1. In other words, those entries can be either 0 or 1. Hence, rest of the $(t-1)$ key shares of any t-sized subset of parties, have only binary entries in it.

Hence we conclude that, in our proposed (t,T) threshold LISSS for a t-sized subset of parties $PT'=\{P_{id_1},P_{id_2},\dots,P_{id_t}\}$, where $id_1<id_2<\cdots<id_t$, all the parties except P_{id_1} will have binary key shares. $\square$

IV. Additional Implementation Details

In this section, we present some additional details about our software implementation, as well as some explanation about our experiments and the case study.

Workflow of Torus-FHE Library. We provide a block diagram (shown in Figure 1) which shows the flow of crypographic API's of the Torus-FHE library functions.

A. *Conversion of* Torus*-LWE (TLWE) Ciphertext into* Torus*-RLWE (TRLWE) Ciphertext*

The outputs of the encryption and homomorphic computation of our proposed scheme are TLWE ciphertexts. In our implementation, this TLWE ciphertext is converted to a TRLWE ciphertext before it is decrypted in a thresholdized manner in order to exploit the packing advantage in TRLWE. To illustrate this, let us assume that we need to encrypt m bits. For this, we need m number of TLWE ciphertexts with, each with N dimensional vectors. However, if $m<N$, all the m bits can be encrypted together in a single TRLWE ciphertext with N coefficients, by constructing the plaintext as a polynomial of degree m with the message bits as coefficients.

We demonstrate the process of converting one TLWE ciphertext to a TRLWE ciphertext in Algorithm 1. Note that according to Torus-FHE library, a TLWE ciphertext $ct=(\boldsymbol{a},b)$, $\boldsymbol{a}$ is a n-sized vector, whereas a TRLWE ciphertext $CT=(A,B)$, A is a collection of k N-sized polynomials. To make the TLWE to TRLWE conversion feasible, we assume two things: firstly, n is equal to N, secondly, k equals to 1. k being 1, $A=\sum_{j=1}^{N}A_jx^{j-1}$ and $S=\sum_{j=1}^{N}S_jx^{j-1}$. Although we have described the process of converting one TLWE ciphertext to one TRLWE ciphertext here, several works [6] exist concentrating on various techniques of packing multiple LWE ciphertexts into single Ring-LWE ciphertext.

Algorithm 1 Convert TLWE ciphertext to TRLWE ciphertext

Input: a TLWE ciphertext $ct=(\boldsymbol{a},b)$ and corresponding TLWE Key $\boldsymbol{s}$
Output: a TRLWE ciphertext $CT=(A,B)$ and corresponding TRLWE key S

```
function CONVERTLWETORLWE(ct, s)
    Initialize a TRLWE ciphertext CT = (A, B) and corresponding TRLWE key S
    for i = 1 to N do
        S_i ← s_i
    B_1 ← b
    A_1 ← a_1
    for i = 2 to N do
        A_i ← −a_{N−i}
    return ⟨CT, S⟩
```

B. *Additional Details for KNN Computation over Encrypted Data*

In this section, we provide the additional details related to different modules used in the encrypted KNN classification

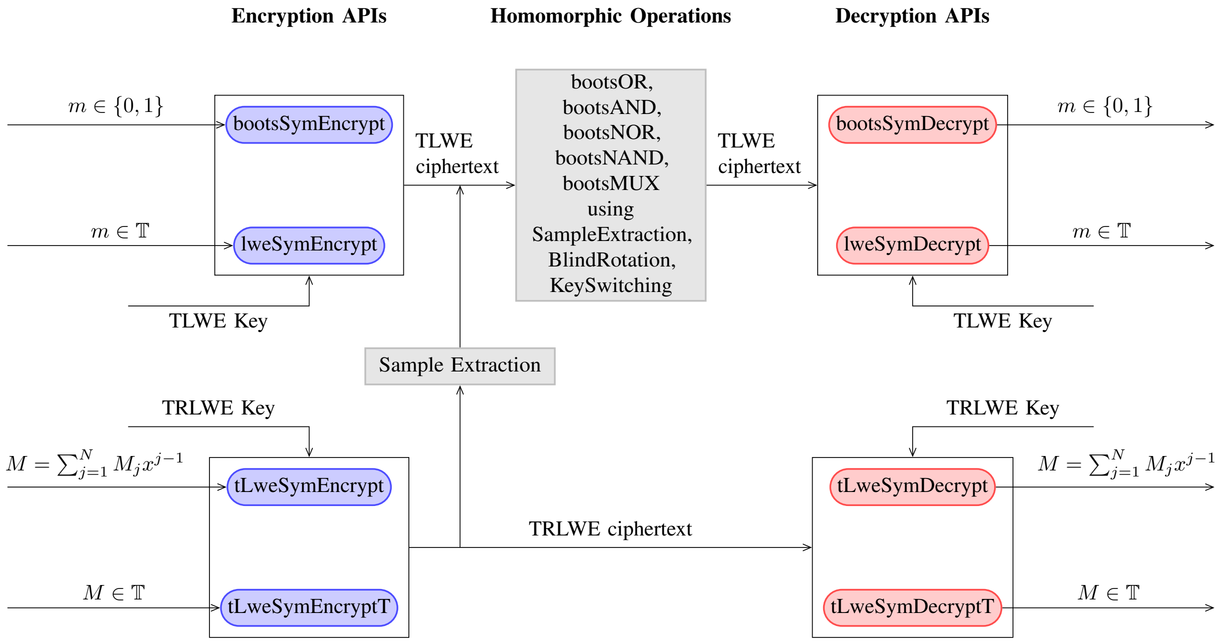

Figure 1: Diagrammatic Illustration of Torus FHE library functions

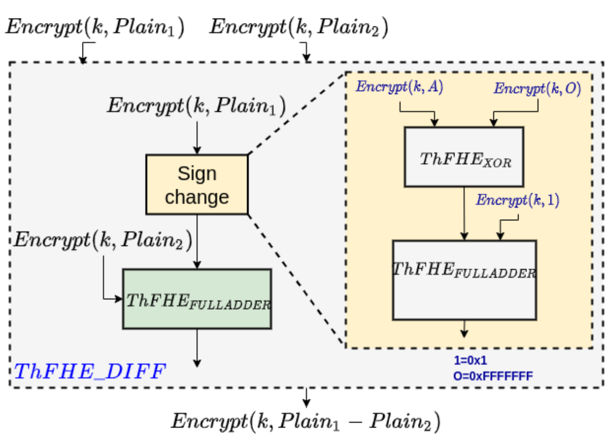

Figure 2: ThFHE Difference Module

algorithm. The encrypted KNN classification algorithm is broadly divided into three stages; 1) *Computing Manhattan distance over encrypted data*, 2) *Computing Bubble-Sort over encrypted data*, and 3) *Computing Prediction over encrypted data*. We require certain modules like, ThFHE$_{FULLADDER}$, ThFHE$_{DIFF}$, and ThFHE$_{MUX}$ (can be found in our code base repository) to compute all the three stages of KNN algorithm.

Manhattan Distance over Encrypted Data. The encrypted Manhattan distance between two encrypted data Encrypt$(k, Plain_1)$ and Encrypt$(k, Plain_2)$ is given by Encrypt$(k, |Plain_1 - Plain_2|)$. It is derived from the ThFHE$_{DIFF}$ module, which in turn is implemented using the ThFHE$_{FULLADDER}$ and ThFHE$_{MUX}$ modules as building blocks. We explain the design of ThFHE$_{DIFF}$ module and the computation of the encrypted Manhattan distance

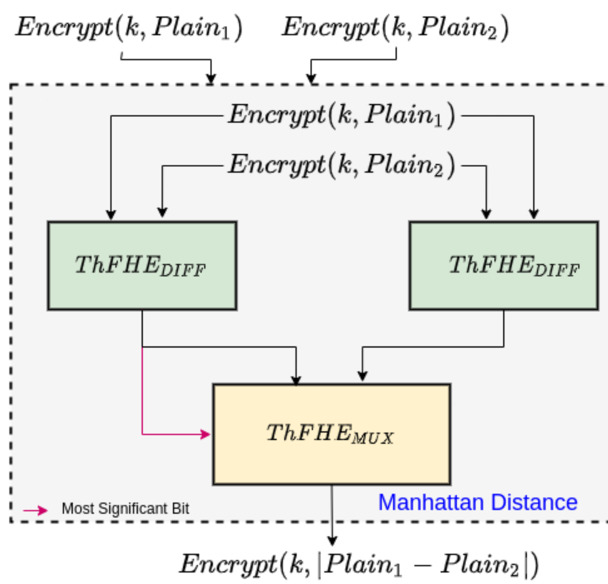

Figure 3: ThFHE Manhattan Distance Module

through Figures 2 and 3. The encrypted Manhattan distance computation module takes two ciphertexts Encrypt$(k, Plain_1)$ and Encrypt$(k, Plain_2)$ at a time, and homomorphically computes the encrypted values Encrypt$(k, Plain_1 - Plain_2)$ and Encrypt$(k, Plain_2 - Plain_1)$ using ThFHE$_{DIFF}$ module. The encrypted decision making module, i.e. ThFHE$_{MUX}$ is then used to homomorphically select the absolute positive distance by using the most significant bit (MSB) of the ThFHE$_{DIFF}$ output as select line. We describe this in more details below.

Algorithm 2 Bubble Sort over encrypted data

Input: Manhattan distances from test data $\boldsymbol{distant} = \{\textsc{Encrypt}(k, distance_1), \ldots, \textsc{Encrypt}(k, distance_n)\}$, $\boldsymbol{train\_data} = \{\textsc{Encrypt}(k, patient_1), \ldots, \textsc{Encrypt}(k, patient_n)\}$, $\boldsymbol{bk}$ = bootstrapping key, K = KNN Parameter.
Output: a Manhattan distance-wise sorted train data $\boldsymbol{sorted\_train\_data}$

Initialize $dist_{smaller}$ and $dist_{bigger}$ to store smaller and bigger distance out of the two elements from $\boldsymbol{distant}$ respectively.
Initialize $smaller$ and $bigger$ to store smaller and bigger out of the two data from $\boldsymbol{train\_data}$ respectively.
for $itr = 0$ to K **do**
  for $i = n - 1$ to 1 **do**
    $diff \leftarrow \textsc{difference}(distant_{i-1}, distant_i, bk)$
    $dist_{bigger} \leftarrow \text{ThFHEMUX}(MSB(diff), distant_i, distant_{i-1}, bk)$
    $dist_{smaller} \leftarrow \text{ThFHEMUX}(MSB(diff), distant_{i-1}, distant_i, bk)$
    $bigger \leftarrow \text{ThFHEMUX}(MSB(diff), train\_data_i, train\_data_{i-1}, bk)$
    $smaller \leftarrow \text{ThFHEMUX}(MSB(diff), train\_data_{i-1}, train\_data_i, bk)$
    $distant_i \leftarrow dist_{bigger}$
    $distant_{i-1} \leftarrow dist_{smaller}$
    $train\_data_i \leftarrow bigger$
    $train\_data_{i-1} \leftarrow smaller$
  $sorted\_train\_data_{itr} \leftarrow train\_data_{itr}$
return $sorted\_train\_data$

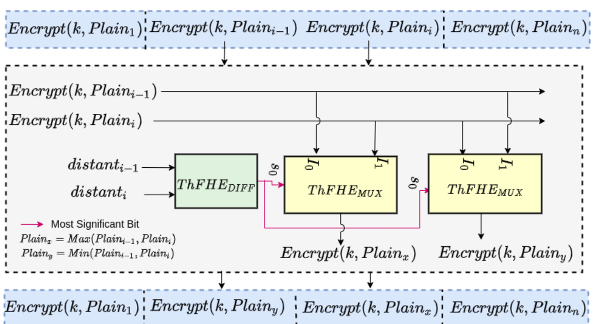

Figure 4: ThFHE Bubble Sort Comparison Module

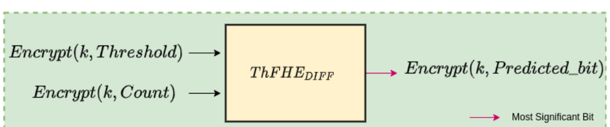

Figure 5: ThFHE Prediction Module

In order to compute $\text{Encrypt}(k, |Plain_1 - Plain_2|)$, we utilise the ThFHE_{DIFF} module shown in Figure 2 to calculate both $\text{Encrypt}(k, Plain_1 - Plain_2)$ and $\text{Encrypt}(k, Plain_2 - Plain_1)$ and out of these two computation only one is selected based on the output of ThFHE_{MUX} as shown in Figure 3. Figure 2 depicts the homomorphic computation of difference between any two ciphertext. It makes use of $\text{ThFHE}_{FULLADDER}$ and XOR gate to perform the difference using simple 2's complement technique. The ThFHE_{MUX} uses the encrypted most significant bit (MSB) of $\text{Encrypt}(k, |Plain_1 - Plain_2|)$ as the select line and outputs $\text{Encrypt}(k, Plain_1 - Plain_2)$ if the select line is the encryption of 0, otherwise it outputs $\text{Encrypt}(k, Plain_2 - Plain_1)$ as the Manhattan distance between $\text{Encrypt}(k, Plain_1)$ and $\text{Encrypt}(k, Plain_2)$. Note that the modules like ThFHE_{MUX}, $\text{ThFHE}_{FULLADDER}$, ThFHE_{DIFF}, etc. are all build on the top of Boolean gates available in Torus-FHE library.

Bubble Sort over encrypted data. As given in algorithm 2, the encrypted bubble sort intakes the bootstrap key, patients data and their corresponding Manhattan distance from test data and outputs sorted patients data. The algorithm runs for K iteration to sort the first K smaller data. The inner loop begins from the end of the list and fixes one of the smallest data at its correct position. This way, when the outer loops has executed for K iteration the algorithm will have brought all the K smallest data at the first K indices in the ascending order. The inner loop picks two elements $distant_i$ and $distant_{i-1}$ and computes the difference between them using ThFHE_{DIFF} module as shown in Figure 2. The encrypted most significant bit of the calculated difference is further used by ThFHE_{MUX} as select line to separate the smaller and bigger distances out of $distant_i$ and $distant_{i-1}$. In a similar fashion, next step separates $train\_data_i$ and $train\_data_{i-1}$ into $smaller$ and $bigger$ using ThFHE_{MUX} module. The next step is to store smaller element at smaller index and bigger element at bigger index in the list. The operations inside inner loop is depicted in bubble sort comparison module (see Figure 2) which takes two ciphertexts and compare them on the basis of Manhattan distance. Let us take two ciphers $\text{Encrypt}(k, Plain_i)$ and $\text{Encrypt}(k, Plain_{i-1})$ and their corresponding Manhattan distance as $distant_i$ and $distant_{i-1}$. Now, compare $distant_i$ and $distant_{i-1}$ using ThFHE_{DIFF} module; if $distant_{i-1}$ is greater than $distant_i$ then the positions of ciphers need to be swapped otherwise positions remain intact. In this way, the smallest distant ciphertext will be placed at the beginning of all the ciphertexts in the first iteration and eventually in K iteration the minimum K-ciphertexts are bubble sorted.

Prediction over encrypted data. In this step, the decisions of k-nearest neighbours are added using $\text{ThFHE}_{FULLADDER}$ and then compared with the threshold value to arrive on the decision for the testing data. Figure 5 shows a comparison between $\text{Encrypt}(k, Count)$ and $\text{Encrypt}(k, Threshold)$ value to output the predicted decisional bit. The $\text{Encrypt}(k, Predicted\_bit)$ is further used by t-out-of-T threshold decryption unit to arrive on the conclusion if the treatment need to be given or not.

References

[1] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[2] R. Bendlin and I. Damgård, "Threshold decryption and zero-knowledge proofs for lattice-based cryptosystems," in *Theory of Cryptography Conference*. Springer, 2010, pp. 201–218.

[3] D. Boneh, R. Gennaro, S. Goldfeder, A. Jain, S. Kim, P. M. R. Rasmussen, and A. Sahai, "Threshold cryptosystems from threshold fully homomorphic encryption," in *CRYPTO 2018, Part I*, ser. LNCS, H. Shacham and A. Boldyreva, Eds., vol. 10991. Springer, Heidelberg, Aug. 2018, pp. 565–596.

[4] S. Bai, T. Lepoint, A. Roux-Langlois, A. Sakzad, D. Stehlé, and R. Steinfeld, "Improved security proofs in lattice-based cryptography: using the rényi divergence rather than the statistical distance," *Journal of Cryptology*, vol. 31, no. 2, pp. 610–640, 2018.

[5] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, "Tfhe: fast fully homomorphic encryption over the torus," *Journal of Cryptology*, vol. 33, no. 1, pp. 34–91, 2020.
[6] H. Chen, W. Dai, M. Kim, and Y. Song, "Efficient homomorphic conversion between (ring) lwe ciphertexts," in *Applied Cryptography and Network Security*, K. Sako and N. O. Tippenhauer, Eds. Cham: Springer International Publishing, 2021, pp. 460–479.