23rd International Symposium on Mathematical Theory of Networks and Systems
Hong Kong University of Science and Technology, Hong Kong, July 16-20, 2018

A New Symmetric Key Homomorphic Encryption Scheme

Uddipana Dowerah[1] and Srinivasan Krishnaswamy[2]

***Abstract*— This work describes a new Symmetric Key Homomorphic Encryption scheme. We define a noisy variant of the Subspace Membership problem called the Hidden Subspace Membership problem and show that the proposed scheme is IND-CPA secure based on the hardness of this problem. We then discuss the homomorphic properties of the proposed scheme.**

***Index Terms*— Homomorphic Encryption; Hidden Subspace Membership**

I. INTRODUCTION

Homomorphic encryption allows mathematical computations to be performed on encrypted data without knowing the actual data or the decryption key. In other words, given a bilinear operation $* : \mathbf{P} \times \mathbf{P} \to \mathbf{P}$, where $\mathbf{P}$ denotes the plaintext space, an encryption scheme is said to be homomorphic with respect to $*$ if for any two plaintexts $\boldsymbol{m}_1, \boldsymbol{m}_2 \in \mathbf{P}$, $Enc(\boldsymbol{m}_1 * \boldsymbol{m}_2)$ can be efficiently computed from $Enc(\boldsymbol{m}_1)$ and $Enc(\boldsymbol{m}_2)$ without using the decryption key. An encryption scheme is called fully homomorphic if it is homomorphic with respect to both addition and multiplication over the underlying algebraic structure. Homomorphic encryption has wide applications in multiparty computation [9], secure electronic voting [4], private information retrieval [13] etc.

Several lattice-based schemes [6], [7], [11] have been proposed following the breakthrough work in [10], which gives the first feasible construction of a fully homomorphic encryption scheme based on *ideal lattices*. Other algebraic structures such as multivariate polynomials and linear codes yields additively homomorphic schemes [1], [2] but for multiplicative homomorphism, the size of the ciphertexts grows exponentially. The construction of a secure fully homomorphic encryption scheme based on these structures remains an open problem.

In this paper, we propose a symmetric key encryption scheme using simple linear algebra operations which is shown to be homomorphic with respect to addition and modular convolution. We define a noisy variant of the subspace membership problem called the Hidden Subspace Membership (HSM) problem and show that the proposed scheme is IND-CPA secure based on the hardness of this problem.

The remainder of this paper is ordered as follows: In section II, we discuss the Hidden Subspace Membership problem. Section III contains the construction of the proposed scheme. In section IV, we discuss the homomorphic properties of the scheme. Section V deals with the security and possible attacks on the proposed scheme. Finally, in section VI, we suggest some experimental parameter choices for the proposed scheme.

The following notations are used in the paper. λ denotes the security parameter of the scheme. $\mathbb{R}$ and $\mathbb{N}$ denote the set of real and natural numbers respectively. $\mathbb{Z}$ denotes the set of integers. $x \leftarrow y$ is used to assign the value y to x. $x \xleftarrow{\$} \mathcal{S}$ means that x is sampled from a set $\mathcal{S}$ uniformly at random. $\mathbb{F}$ denotes a field and $\mathbb{F}_q$ denotes a finite field of order q, where q is a prime power. $\mathbb{F}_q[x]_{\leq d}$ denotes the set of polynomials of degree $\leq d$ in the variable x over $\mathbb{F}_q$ for some $d \in \mathbb{N}$. Given a set $\mathcal{S}$, $|\mathcal{S}|$ denotes the cardinality of $\mathcal{S}$. For any positive integer n, $[n]$ denotes the set of integers $\{1, 2, \ldots, n\}$. We use uppercase script letters $\mathcal{A}, \mathcal{B}, \ldots$ to denote tensors and uppercase letters $A, B, \ldots$ to denote matrices. The i^{th} column of a matrix A is denoted by $A(:, i)$. Scalars are denoted using lower case letters $a, b, \ldots$ and vectors are denoted using lowercase bold letters $\boldsymbol{a}, \boldsymbol{b}, \ldots$ etc. A function $f(x) : \mathbb{N} \to \mathbb{R}$ is called negligible if, for every $\omega \in \mathbb{N}$, there exists an integer n_ω such that $|f(x)| < \frac{1}{x^\omega}$ for all $x > n^\omega$.

II. HIDDEN SUBSPACE MEMBERSHIP PROBLEM

Given a set of noisy samples from a subspace $\mathcal{S}$ of a vector space $\mathcal{V}$, where the noise comes from a distribution $\mathcal{N}$ over $\mathcal{V}$, the aim of the Hidden Subspace Membership (HSM) problem is to compute a basis for $\mathcal{S}$. A decisional variant of the problem, the Decisional Hidden Subspace Membership (DHSM) problem involves distinguishing elements of the subspace from uniformly sampled elements of the vector space.

[1] Researcher is with the Department of Electronics and Electrical Engineering, Indian Institute of Technology Guwahati, Guwahati, Assam 781039, India d.uddipana@iitg.ernet.in

[2] Faculty is with the Department of Electronics and Electrical Engineering, Indian Institute of Technology Guwahati, Guwahati, Assam 781039, India srinikris@iitg.ernet.in

Definition 1: **(Hidden Subspace Membership (HSM) Problem)**. Let $\mathcal{S}$ be a subspace of a vector space $\mathcal{V}$ and $\mathcal{N}$ be a noise ditribution on $\mathcal{V}$. The HSM problem is defined as: given polynomially many samples of the form $(\boldsymbol{v} + \boldsymbol{n})$ such that $\boldsymbol{v} \xleftarrow{\$} \mathcal{S}$ and $\boldsymbol{n} \xleftarrow{\$} \mathcal{N}$, output a basis for $\mathcal{S}$ with high probability.

Observe that, the noise-free variant of the HSM problem is extremely easy to solve. If $\mathcal{S}$ is a k-dimensional subspace of a vector space $\mathcal{V}$, then k linearly independent samples of $\mathcal{S}$ can be used to construct a basis for $\mathcal{S}$.

We define the decisional variant using the game playing framework adapted from [3]. A game is run with an adversary $\mathcal{A}$, where $\mathcal{A}$ is considered as a Probabilistic Polynomial Time (PPT) algorithm and consists of procedures called oracles. We consider an algorithm $\mathcal{G}$ that takes as input the security parameter λ and a description of the vector space $\mathcal{V}$ and generates a basis $B_{\mathcal{S}}$ for a subspace $\mathcal{S} \subset \mathcal{V}$.

Definition 2: **(Decisional Hidden Subspace Membership (DHSM) Problem)**. The DHSM problem can be defined in terms of the game shown in Figure 1. A PPT adversary $\mathcal{A}$ wins the game if it can guess the value of c with a non-negligible advantage ϵ, where ϵ is defined as $\epsilon := \left| Pr[c = c'] - \frac{1}{2} \right|$.

Initialize
1. $B_{\mathcal{S}} \leftarrow \mathcal{G}(\lambda, \mathcal{V})$
2. $c \xleftarrow{\$} \{0, 1\}$

Challenge()
1. $\boldsymbol{v} \xleftarrow{\$} \mathcal{V}$
2. if $c = 1$, $\boldsymbol{v} \xleftarrow{\$} \mathcal{S}$
3. return $\boldsymbol{v}$

Sample()
1. $\boldsymbol{v} \xleftarrow{\$} \mathcal{S}$, $\boldsymbol{n} \xleftarrow{\$} \mathcal{N}$
2. set $\boldsymbol{v} \leftarrow \boldsymbol{v} + \boldsymbol{n}$
3 return $\boldsymbol{v}$

Finalize (c')
1. return $(c = c')$

Fig 1: DHSM Game

It can be easily proved that the two problems are equivalent. If there is an adversary that can solve the DHSM problem, then it can be used to detect k linearly independent elements of the subspace $\mathcal{S}$ and thus contsruct a basis for $\mathcal{S}$. Conversely, if the HSM problem can be solved and we have a basis for $\mathcal{S}$ then it can be used to construct a basis for $\mathcal{S}^{\perp}$. Then, one can check whether a given vector lies in $\mathcal{S}$ by checking if it lies in the kernel of $\mathcal{S}^{\perp}$.

We now demonstrate that the hardness of the HSM problem is related to the well-known Learning With Errors (LWE) problem [16].

Definition 3: **(Learning With Errors (LWE) Problem)**. Given a noise distribution $\mathcal{N}$ over $\mathbb{Z}_q$, the LWE problem can be defined in terms of the game shown in Figure 2 as mentioned in [1]. A PPT adversary $\mathcal{A}$ wins the game if it can guess the value of $\boldsymbol{s}$ with non-negligible advantage ϵ, where $\epsilon := Pr[s = s']$.

Initialize
1. $m \leftarrow m(\lambda)$
2. $\boldsymbol{s} \xleftarrow{\$} \mathbb{Z}_q^m$

Sample()
1 $\boldsymbol{a} \xleftarrow{\$} \mathbb{Z}_q^m$, $e \xleftarrow{\$} \mathcal{N}$
2. set $b \leftarrow \left(\boldsymbol{a}^T \boldsymbol{s} + e\right) \bmod q$
3 return $(\boldsymbol{a}, b)$

Finalize ($\boldsymbol{s}'$)
1. return $(\boldsymbol{s} = \boldsymbol{s}')$

Fig 2: LWE Game

Given samples from the HSM game, recovering the subspace $\mathcal{S}$ is equivalent to recovering its perpendicular space $\mathcal{S}^{\perp}$. Therefore, the HSM problem over the vector space $\mathbb{Z}_q^m$ can be modified as: given noisy samples $\boldsymbol{v}_i$, determine a vector $\boldsymbol{s}$ in $\mathcal{S}^{\perp}$ such that $\boldsymbol{v}_i^T \boldsymbol{s} = 0 \pmod q$. Observe that, an LWE sample $(\boldsymbol{a}_i, b_i)$ is such that $\boldsymbol{a}_i^T \boldsymbol{s} \approx b_i \pmod q$ which is a noisy equation with the noise being sampled according to some probability distribution $\mathcal{N}$ on $\mathbb{Z}_q$. It can be alternatively viewed as,

$$\begin{bmatrix} \boldsymbol{a}_i^T & -b_i \end{bmatrix} \begin{bmatrix} \boldsymbol{s} \\ 1 \end{bmatrix} \approx 0 \pmod q \tag{1}$$

Hence, we can construct a PPT adversary against the LWE problem from a PPT adversary against the HSM problem with the same advantage. Thus, the HSM problem is as difficult as the LWE problem.

III. The Proposed Scheme

In this section, we describe the construction of the proposed scheme. The scheme encrypts messages over the space $\mathbb{F}_q^m$ and maps it to a matrix $C \in \mathbb{F}_{q^\ell}^{m \times n}$ for some $m, n \in \mathbb{N}$ such that $m < n$, $\mathbb{F}_{q^\ell}$ is an extension field of $\mathbb{F}_q$ of order q^ℓ for some $\ell \in \mathbb{N}$ and m and n are functions of the security parameter λ. The scheme can be summarized in terms of the following algorithms:

- **KeyGen**(λ, m, n): The key generation algorithm takes the parameters λ, m, n as inputs and generates the secret key sk. Sample $L \xleftarrow{\$} GL_m(\mathbb{F}_{q^\ell})$ and $R \xleftarrow{\$} GL_n(\mathbb{F}_{q^\ell})$. Set $sk := (L, R)$.
- **Encrypt**$(sk, \boldsymbol{m})$: It takes a message $\boldsymbol{m} \in \mathbb{F}_q^m$ and the secret key sk as inputs and outputs a block of encryptions $\boldsymbol{C} := \{C^i \in \mathbb{F}_{q^\ell}^{m \times n} | 1 \leq i \leq \eta\}$, for some $\eta \in \mathbb{N}$. Consider the set

$$\mathcal{D} := \left\{ X \in \mathbb{F}_{q^\ell}^{m \times n} \;\middle|\; \sum_{i=1}^{n} X(:, i) = 0 \bmod q^\ell \right\} \tag{2}$$

Algorithm 1: KeyGen(λ, m, n)

Input : λ, m, n
Output: sk
sample $L \xleftarrow{\$} GL_m(\mathbb{F}_{q^\ell})$, $R \xleftarrow{\$} GL_n(\mathbb{F}_{q^\ell})$
set secret key $sk := (L, R)$
return sk

and sample $Q^i \xleftarrow{\$} \mathcal{D}$. Set $Q^i(:, \beta) \leftarrow Q^i(:, \beta) + \boldsymbol{m}$, where $\beta \in_\$ [n]$. Compute, $C^i = L \cdot Q^i \cdot R + c_i \cdot N^i$, where $N^i \in \mathbb{F}_{q^\ell}^{m \times n}$ is a random noise matrix and $c_i \in \{0, 1\}$ is sampled according to a distribution $\mathcal{E}$ such that for every block of η samples $c_\gamma = 0$ for a randomly chosen $\gamma \in [\eta]$ and $c_i = 1$ for $1 \leq i \leq \eta, i \neq \gamma$. Thus, every η encryptions contain one noise-free encryption of $\boldsymbol{m}$. **Encrypt** then outputs the ciphertext block $\boldsymbol{C} = \{C^i\}_{1 \leq i \leq \eta}$.

Algorithm 2: Encrypt($\boldsymbol{m}, sk$)

Input : $\boldsymbol{m}$, sk, $\mathcal{D}$, $\mathcal{E}$, η
Output: ciphertext $\boldsymbol{C} = \{C^i \in \mathbb{F}_{q^\ell}^{m \times n} | 1 \leq i \leq \eta\}$
sample $Q^i \xleftarrow{\$} \mathcal{D}$, $c_i \xleftarrow{\mathcal{E}} \{0, 1\}$, $N^i \xleftarrow{\$} \mathbb{F}_{q^\ell}^{m \times n}$
set $Q^i(:, \beta) \leftarrow Q^i(:, \beta) + \boldsymbol{m}$ for $\beta \xleftarrow{\$} [n]$
set $C^i = L \cdot Q^i \cdot R + c_i \cdot N^i$
return $\boldsymbol{C} = \{C^i\}_{1 \leq i \leq \eta}$

- **Decrypt**$(sk, \boldsymbol{C})$: It takes a ciphertext block $\boldsymbol{C}$ and the secret key sk and outputs the corresponding message $\boldsymbol{m}$. A decrypter with the knowledge of L and R can determine $\boldsymbol{m}_i = \sum_{j=1}^{n} (L^{-1} \cdot C^i \cdot R^{-1})(:, j)$ and outputs $\boldsymbol{m} = \boldsymbol{m}_i$ with the smallest i such that $\boldsymbol{m}_i \in \mathbb{F}_q^m$.

Algorithm 3: Decrypt($\boldsymbol{C}, sk$)

Input : ciphertext $\mathbf{C} = \{C^i\}_{1 \leq i \leq \eta}$, secret key sk
Output: message $\boldsymbol{m}$
for $i = 1$ **to** η **do**
  set $\boldsymbol{m}_i = \sum_{j=1}^{n} (L^{-1} C^i R^{-1})(:, j)$
end
if $\boldsymbol{m}_i \in \mathbb{F}_q^m$ **then**
  set $\boldsymbol{m} = \boldsymbol{m}_i$
return $\boldsymbol{m}$

The following lemma shows that, if the decryption of a ciphertext gives an element of $\mathbb{F}_q^m$, then the decrypted vector is a valid message with very high probability. For the simplicity of the calculations, we assume a uniform distribution of noise in this paper for the noisy elements of the subspace.

Lemma 1: If the decryption of a ciphertext $\boldsymbol{C}$ of the respective message $\boldsymbol{m}$ is an element of $\mathbb{F}_q^m$, then the probability that it comes from the noiseless element of $\boldsymbol{C}$ is given by $1/((\eta - 1)q^{m(1-\ell)} + 1)$.

Proof: Let A be the event that $\boldsymbol{C}$ decrypts to a vector in $\mathbb{F}_q^m$ and B be the event that it comes from a noisy element of $\boldsymbol{C}$. An outcome of B is of the form $C^i = L \cdot Q^i \cdot R + N^i = (L \cdot Q^i + \widehat{N}^i)R$ for some $\widehat{N}^i \in \mathbb{F}_{q^\ell}^{m \times n}$ and there exists a one-to-one correspondence between N^i and $\widehat{N}^i$. Then, the required probability can be determined as follows,

$$Pr(\overline{B}|A) = \frac{Pr(A|\overline{B}) \cdot Pr(\overline{B})}{Pr(A)} \tag{3}$$

where $\overline{B}$ denotes the complement of the event B and $Pr(A|\overline{B})$ is the probability that the decryption of a noiseless element of $\boldsymbol{C}$ lies in $\mathbb{F}_q^m$. Hence, $Pr(A|\overline{B}) = 1$ and $Pr(\overline{B}) = \frac{1}{\eta}$, given that there exits one noise-free encryption of $\boldsymbol{m}$ in $\boldsymbol{C}$. $Pr(A)$ can be determined as,

$$Pr(A) = Pr(A|B) \cdot Pr(B) + Pr(A|\overline{B}) \cdot Pr(\overline{B}) \tag{4}$$

where, $Pr(A|B)$ is the probability that the decryption of a noisy element of $\boldsymbol{C}$ lies in $\mathbb{F}_q^m$. Therefore, $Pr(A|B) = \frac{q^m}{q^{\ell m}} = q^{m(1-\ell)}$. Hence,

$$Pr(A) = q^{m(1-\ell)} \left(\frac{\eta - 1}{\eta} \right) + \frac{1}{\eta} \tag{5}$$

Therefore,

$$Pr(\overline{B}|A) = \frac{1}{(\eta - 1)q^{m(1-\ell)} + 1} \tag{6}$$

■

IV. Homomorphic Properties

The proposed scheme can be used to remotely perform computations on encrypted data without explicit decryption. The remote server which is used to perform these computations is provided with an evaluation key ek by the client. If a bilinear operation M has to be performed on two messages $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$, the server is provided with encrypted blocks $\boldsymbol{C}_1 := \{Enc(\boldsymbol{m}_1, sk)\}$ and $\boldsymbol{C}_2 = \{Enc(\boldsymbol{m}_2, sk)\}$ which contains encryptions of $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ under the secret key sk. Taking the evaluation key ek and the individual elements C_1^i and C_2^j of $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ as inputs, the server is capable of evaluating a function $M'(ek, C_1^i, C_2^j)$ such that if C_1^i and C_2^j denote the respective noiseless elements of $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$, then $M'(ek, C_1^i, C_2^j) = M(\boldsymbol{m}_1, \boldsymbol{m}_2)$. The remote evaluation of the function M can be done as per the following algorithm:

1) The server performs a random permutation π on the set $\{1, 2, \ldots, \eta\}$.

2) It then creates a block $\boldsymbol{C}'$ which contains the elements $M'\left(ek, C_1^i, C_2^{\pi(i)}\right)$ for $1 \leq i \leq \eta$ and sends it to the client.
3) The client decrypts the block $\boldsymbol{C}'$ and checks if it contains a vector in the base field $\mathbb{F}_q^m$. If not, the client requests the server for another evaluation using a different permutation.

In the remainder of the section, we will demonstrate the utility of the above scheme in remotely performing addition and modular convolution.

A. Addition

Using the above mentioned schematic, we now see how two messages can be homomorphically added as elements of $\mathbb{F}_q^m$. Let $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ be two messages that need to be added. The server is provided with their encryptions $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$. The elements of $\boldsymbol{C}_1$ are added with the permuted elements of $\boldsymbol{C}_2$ and the resulting block $\boldsymbol{C}_{add}$ is given to the client for decryption. If the decryption of the block $\boldsymbol{C}_{add}$ yields an element of $\mathbb{F}_q^m$, i.e., if $\sum_{j=1}^n \left(L^{-1}(C_1^i + C_2^{\pi(i)})R^{-1}\right)(:, j) \in \mathbb{F}_q^m$ for some $i \in [\eta]$, then the block $\boldsymbol{C}_{add}$ is accepted as an encryption of $\boldsymbol{m}_1 + \boldsymbol{m}_2$. If not, the client requests the server to repeat the operation. The algorithm implemented by the server at each stage is shown in Algorithm 4. Here $\mathcal{P}$ denotes the set of permutations on the set $\{1, 2, \ldots, \eta\}$ and '+' denotes addition over $\mathbb{F}_{q^\ell}$ unless stated otherwise.

Algorithm 4: Add

Input : ciphertexts $\boldsymbol{C}_1 = \{C_1^i\}_{1 \leq i \leq \eta}, \boldsymbol{C}_2 = \{C_2^i\}_{1 \leq i \leq \eta}$
Output: ciphertext $\boldsymbol{C}_{add}$
set $\pi \xleftarrow{\$} \mathcal{P}([\eta])$
set $\boldsymbol{C}_{add} := \left\{C_1^i + C_2^{\pi(i)} : 1 \leq i \leq \eta\right\}$
return $\boldsymbol{C}_{add}$

1) Correctness of Addition: The underlying assumption of the above mentioned scheme is that if the decryption of the resulting block $\boldsymbol{C}_{add}$ yields an element of $\mathbb{F}_q^m$, this element should have necessarily come from the addition of two noiseless elements of $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$. If C_1^i and $C_2^{\pi(i)}$ denotes the noiseless elements of $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$, then

$$\sum_{i=1}^{n} \left(L^{-1}(C_1^i + C_2^{\pi(i)})R^{-1}\right) = \boldsymbol{m}_1 + \boldsymbol{m}_2 \pmod q \tag{7}$$

Lemma 2: Let $C_1^{\pi_0(i)} + C_2^{\pi_1(i)} + \cdots + C_k^{\pi_{k-1}(i)} \leftarrow \mathbf{Add}(\boldsymbol{C}_1, \ldots, \boldsymbol{C}_k)$ denotes the addition of k respective elements of $\boldsymbol{C}_1, \ldots, \boldsymbol{C}_k$. Given that, the decryption of $C_1^{\pi_0(i)} + \cdots + C_k^{\pi_{k-1}(i)}$ yields an element of $\mathbb{F}_q^m$, the probability that it comes from the addition of the noiseless elements of $\boldsymbol{C}_1, \ldots, \boldsymbol{C}_k$ is given by $1/\left(\left(\eta^k - 1\right) q^{m(1-\ell)} + 1\right)$.

Proof: Let A be the event that $C_1^{\pi_0(i)} + \cdots + C_k^{\pi_{k-1}(i)}$ decrypts to a vector in $\mathbb{F}_q^m$ and B be the event that at least one of the $C_j^{\pi_{j-1}(i)}$s is noisy for some $j \in [k]$. Then $\overline{B}$ is event of adding the noiseless elements of the respective ciphertexts. Observe that, $\overline{B} \subseteq A$. Therefore, the required probability can be determined as,

$$Pr(\overline{B}|A) = \frac{Pr(\overline{B})}{Pr(A)} \tag{8}$$

where, $Pr(\overline{B}) = \frac{1}{\eta^k}$, given that each $\boldsymbol{C}_j$ contains one noiseless element for $1 \leq j \leq k$. Observe that, $Pr(A|\overline{B}) = 1$ and $Pr(A|B) = \frac{q^m}{q^{\ell m}}$, considering that the first $(k-1)$ elements are chosen uniformly at random and the last element in such a way that their sum lies in $\mathbb{F}_q^m$. Therefore,

$$\begin{aligned} Pr(A) &= Pr(A|B) \cdot Pr(B) + Pr(A|\overline{B}) \cdot Pr(\overline{B}) \\ &= \left(1 - \frac{1}{\eta^k}\right) q^{m(1-\ell)} + \frac{1}{\eta^k} \end{aligned} \tag{9}$$

Hence,

$$Pr(\overline{B}|A) = \frac{1}{(\eta^k - 1)q^{m(1-\ell)} + 1} \tag{10}$$

■

Lemma 3: The expected number of trials till the decryption of $\boldsymbol{C}_{add} \leftarrow \mathbf{Add}(\boldsymbol{C}_1, \boldsymbol{C}_2)$ outputs a vector in $\mathbb{F}_q^m$ is $\frac{\eta^2}{(\eta^2-1)q^{m(1-\ell)}+1}$.

Proof: Let A be the event that $\boldsymbol{C}_{add}$ outputs a vector in $\mathbb{F}_q^m$. Then, from Lemma 2, it can be easily seen that

$$Pr(A) = \left(1 - \frac{1}{\eta^2}\right) q^{m(1-\ell)} + \frac{1}{\eta^2} \tag{11}$$

Hence, the expected number of trials till $\boldsymbol{C}_{add}$ decrypts to a vector in $\mathbb{F}_q^m$ is $\frac{1}{Pr(A)} = \frac{\eta^2}{(\eta^2-1)q^{m(1-\ell)}+1}$. ■

B. Modular Convolution

A message $\boldsymbol{m} \in \mathbb{F}_q^m$ can be viewed as a polynomial of degree $< m$ over $\mathbb{F}_q[x]$. There exists a natural map $\phi : \mathbb{F}_q^m \to \mathbb{F}_q[x]_{\leq (m-1)}$ from the elements of $\mathbb{F}_q^m$ to the polynomials of $\mathbb{F}_q[x]_{\leq (m-1)}$ defined as

$$\phi(a_0, \ldots, a_{m-1}) = a_0 + a_1 x + \ldots + a_{m-1} x^{m-1} \tag{12}$$

Given a polynomial $f(x) \in \mathbb{F}_q[x]_{\leq m}$, the modular convolution of $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ with respect to $f(x)$, denoted as $*_f$, can be defined as:

$$\boldsymbol{m}_1 *_f \boldsymbol{m}_2 = \phi^{-1}\left[\phi\left(\boldsymbol{m}_1\right) * \phi\left(\boldsymbol{m}_2\right) \mod f(x)\right] \tag{13}$$

1) The Public Evaluation Key: The modular convolution of two messages $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ can be homomorphically evaluated using an order-4 tensor $\mathcal{T}$ that acts on two respective elements C_1^i and $C_2^{\pi(i)}$ of $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ as $C_1^i * C_2^{\pi(i)} = {C_1^i}^T \mathcal{T} C_2^{\pi(i)}$, where $\mathcal{T}$ is given as the public evaluation key. The rest of the section deals with generating the tensor $\mathcal{T}$.

The modular convolution with respect to $f(x)$ is a bilinear operation that can be represented by an $m \times m \times m$ order-3 tensor $\mathcal{B}$ with respect to a basis of $\mathbb{F}_q^m$. Using $\mathcal{B}$, $\boldsymbol{m}_1 *_f \boldsymbol{m}_2$ of two messages $\boldsymbol{m}_1, \boldsymbol{m}_2 \in \mathbb{F}_q^m$ can be obtained as

$$\begin{aligned} \boldsymbol{m}_1 *_f \boldsymbol{m}_2 &= \boldsymbol{m}_1^T \mathcal{B} \boldsymbol{m}_2 \\ &= \begin{bmatrix} \boldsymbol{m}_1^T \mathcal{B}_1 \boldsymbol{m}_2 & \dots & \boldsymbol{m}_1^T \mathcal{B}_m \boldsymbol{m}_2 \end{bmatrix}^T \end{aligned} \quad (14)$$

where $\mathcal{B}_j = \mathcal{B}(:,:,j)$ denotes the frontal slices of $\mathcal{B}$ for $1 \le j \le m$. Then, the modular convolution of $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ can be homomorphically computed using the following tensor

$$\mathcal{X}_j = \left(\mathcal{B} \times_1 L^{-T} \times_2 L^{-T}\right) \otimes \left(R^{-1}(:,j) \cdot \left(\sum_{k=1}^{n} R^{-1}(:,k) \right)^T \right) \quad (15)$$

for $1 \le j \le n$. Here, $\times_i$ denotes the mode-i product of tensors and $\otimes$ denotes the usual tensor product. If $(\boldsymbol{c}_1^{(1)}, \dots, \boldsymbol{c}_n^{(1)})$ and $(\boldsymbol{c}_1^{(2)}, \dots, \boldsymbol{c}_n^{(2)})$ represents the column vectors of C_1^i and $C_2^{\pi(i)}$ and R^{-1} is the $n \times n$ matrix with entries $R^{-1}(j,k) = r_{jk}$ for $1 \le j, k \le n$, then

$$C_1^i * C_2^{\pi(i)} = \begin{bmatrix} {\boldsymbol{c}_1^{(1)}}^T & \dots & {\boldsymbol{c}_n^{(1)}}^T \end{bmatrix} \mathcal{X}_i \begin{bmatrix} \boldsymbol{c}_1^{(2)} \\ \vdots \\ \boldsymbol{c}_n^{(2)} \end{bmatrix} \quad (16)$$

where $\mathcal{X}_j$ is given by the following $mn \times mn \times m$ tensor.

$$\mathcal{X}_j = \begin{bmatrix} r_{1j}\sum_k r_{1k}\widetilde{\mathcal{B}} & r_{1j}\sum_k r_{2k}\widetilde{\mathcal{B}} & \dots & r_{1j}\sum_k r_{nk}\widetilde{\mathcal{B}} \\ r_{2j}\sum_k r_{1k}\widetilde{\mathcal{B}} & r_{2j}\sum_k r_{2k}\widetilde{\mathcal{B}} & \dots & r_{2j}\sum_k r_{nk}\widetilde{\mathcal{B}} \\ \vdots & \vdots & \ddots & \vdots \\ r_{nj}\sum_k r_{1k}\widetilde{\mathcal{B}} & r_{nj}\sum_k r_{2k}\widetilde{\mathcal{B}} & \dots & r_{nj}\sum_k r_{nk}\widetilde{\mathcal{B}} \end{bmatrix} \quad (17)$$

For $1 \le j \le n$, $\mathcal{X} := (\mathcal{X}_1, \dots, \mathcal{X}_n)$ represents an order-4 tensor of dimension $mn \times mn \times m \times n$ over $\mathbb{F}_{q^\ell}^m$. The tensor $\mathcal{X}$ is then pre and post-multiplied by two matrices L_1 and R_1 chosen uniformly at random from $GL_m(\mathbb{F}_{q^\ell})$ and $GL_n(\mathbb{F}_{q^\ell})$ respectively, to obtain $\mathcal{T} = \mathcal{X} \times_3 L_1 \times_4 R_1$.

The server homomorphically computes the modular convolution of two messages as per Algorithm 5. Similar to addition, if the decryption of $\boldsymbol{C}_{con}$ yields a vector in $\mathbb{F}_q^m$, it is accepted as an encryption of $\boldsymbol{m}_1 *_f \boldsymbol{m}_2$.

Algorithm 5: Mult

Input : ciphertexts $\boldsymbol{C}_1 = \{C_1^i\}_{1 \le i \le \eta}, \boldsymbol{C}_2 = \{C_2^i\}_{1 \le i \le \eta}$, evaluation key $\mathcal{T}$
Output: ciphertext $\boldsymbol{C}_{con}$
set $\pi \xleftarrow{\$} \mathcal{P}([\eta])$
set $\boldsymbol{C}_{con} := \left\{ C_1^i * C_2^{\pi(i)} = {C_1^i}^T \mathcal{T} C_2^{\pi(i)} : \ 1 \le i \le \eta \right\}$
return $\boldsymbol{C}_{con}$

2) Correctness of Convolution: The decryption key for the convoluted ciphertexts is (L_1, R_1). If C_1^i and $C_2^{\pi(i)}$ denote the noiseless elements of $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$, then

$$L_1^{-1} \left(C_1^i * C_2^{\pi(i)} \right) R_1^{-1} = {C_1^i}^T \mathcal{X} C_2^{\pi(i)} \quad (18)$$

Using equation (16) and (17), $C'_{con} = {C_1^i}^T \mathcal{X} C_2^{\pi(i)}$ is an $m \times n$ matrix, which can be written in terms of its column vectors as:

$$\left[\left({Q_1^i(:,1)}^T \mathcal{B} \sum_{i=1}^{n} Q_2^{\pi(i)}(:,i) \right) \ \dots \ \left({Q_1^i(:,n)}^T \mathcal{B} \sum_{i=1}^{n} Q_2^{\pi(i)}(:,i) \right) \right]$$

The decryption function retrieves,

$$\sum_{j=1}^{n} C'_{con}(:,j) = \boldsymbol{m}_1 *_f \boldsymbol{m}_2 \quad (19)$$

Lemma 4: Let $C_1^{\pi_0(i)} * C_2^{\pi_1(i)} * \dots * C_k^{\pi_{k-1}}(i) \leftarrow \textbf{Mult}(\boldsymbol{C}_1, \dots, \boldsymbol{C}_k, ek)$ denotes the convolution of k respective elements of $\boldsymbol{C}_1, \dots, \boldsymbol{C}_k$. Given that the decryption of $C_1^{\pi_0(i)} * \dots * C_k^{\pi_{k-1}(i)}$ yields an element of $\mathbb{F}_q^m$, the probability that it comes from the modular convolution of the noiseless elements of $\boldsymbol{C}_1, \dots, \boldsymbol{C}_k$ is given by $\frac{1}{(\eta^k - 1)\left[\left(1 - \left(1 - \frac{1}{q^{\ell m}}\right)^k\right) + \left(1 - \frac{1}{q^{\ell m}}\right)^k \left(\frac{q^m - 1}{q^{\ell m} - 1}\right)\right] + 1}$.

Proof: Let A be the event that $C_1^{\pi_0(i)} * \dots * C_k^{\pi_{k-1}(i)}$ yields an element of $\mathbb{F}_q^m$ and B be the event that atleast one of the $C_j^{\pi_{j-1}(i)}$s is noisy for some $j \in [k]$. Observe that, $\overline{B} \subseteq A$. Therefore, the required probability can be determined as,

$$Pr(\overline{B}|A) = \frac{Pr(\overline{B})}{Pr(A)} \quad (20)$$

The probability that $C_1^{\pi_0(i)} * \dots * C_k^{\pi_{k-1}(i)}$, containing a noisy element, decrypts to a vector in $\mathbb{F}_q^m$ can be determined as follows. Two cases may arise: first, when at least one of the individual elements decrypts to zero

and second, when all the elements decrypt to non-zero elements of $\mathbb{F}_{q^\ell}^m$. Considering that the first $(k-1)$ elements are chosen uniformly at random and the last element in such a way that their modular convolution lies in $\mathbb{F}_q^m$, the required probability for the first case can be determined as

$$\left(1-\left(1-\tfrac{1}{q^{\ell m}}\right)^k\right)$$

and for the second case, it can be obtained as

$$\left(1-\tfrac{1}{q^{\ell m}}\right)^k \cdot \left(\tfrac{q^m-1}{q^{\ell m}-1}\right)$$

Therefore,

$$Pr(A|B)=\left(1-\left(1-\frac{1}{q^{\ell m}}\right)^k\right)+\left(1-\frac{1}{q^{\ell m}}\right)^k\left(\frac{q^m-1}{q^{\ell m}-1}\right) \tag{21}$$

Observe that, $Pr(A|\overline{B})=1$ and $Pr(B)=1-\frac{1}{\eta^k}$. Therefore, $Pr(A)$ can be determined as

$$\begin{aligned}Pr(A) &= Pr(A|B)\cdot Pr(B)+Pr(A|\overline{B})\cdot Pr(\overline{B})\\ &=\left(1-\frac{1}{\eta^k}\right)\left[\left(1-\left(1-\frac{1}{q^{\ell m}}\right)^k\right)+\left(1-\frac{1}{q^{\ell m}}\right)^k\left(\frac{q^m-1}{q^{\ell m}-1}\right)\right]\\ &\quad+\frac{1}{\eta^k}\end{aligned} \tag{22}$$

Hence,

$$Pr(\overline{B}|A)=\frac{1}{(\eta^k-1)\left[\left(1-\left(1-\frac{1}{q^{\ell m}}\right)^k\right)+\left(1-\frac{1}{q^{\ell m}}\right)^k\left(\frac{q^m-1}{q^{\ell m}-1}\right)\right]+1} \tag{23}$$

■

Lemma 5: The expected number of trials till the decryption of $\boldsymbol{C}_{con} \leftarrow \textbf{Mult}(\boldsymbol{C}_1, \boldsymbol{C}_2, ek)$ outputs a vector in $\mathbb{F}_q^m$ is $\frac{\eta^2}{(\eta^2-1)\left[\left(1-\left(\frac{q^{\ell m}-1}{q^{\ell m}}\right)^2\right)+\left(1-\frac{1}{q^{\ell m}}\right)^2\left(\frac{q^m-1}{q^{\ell m}-1}\right)\right]+1}$.

Proof: Let A be the event that $\boldsymbol{C}_{con}$ outputs a vector in $\mathbb{F}_q^m$. Then, from Lemma 4, it can be easily verified that,

$$\begin{aligned}Pr(A) &= \left(1-\frac{1}{\eta^2}\right)\left[\left(1-\left(1-\frac{1}{q^{\ell m}}\right)^2\right)+\left(1-\frac{1}{q^{\ell m}}\right)^2\left(\frac{q^m-1}{q^{\ell m}-1}\right)\right]\\ &\quad+\frac{1}{\eta^2}\end{aligned} \tag{24}$$

Hence, the expected number of trials till $\boldsymbol{C}_{con}$ decrypts to a vector in $\mathbb{F}_q^m$ is

$$\frac{1}{Pr(A)}=\frac{\eta^2}{(\eta^2-1)\left[\left(1-\left(\frac{q^{\ell m}-1}{q^{\ell m}}\right)^2\right)+\left(1-\frac{1}{q^{\ell m}}\right)^2\left(\frac{q^m-1}{q^{\ell m}-1}\right)\right]+1}$$

■

C. Compactness

An encryption of a message is a block of $(m \times n)$ matrices $\{C^i \in \mathbb{F}_{q^\ell}^{m\times n}$ for $1 \le i \le \eta\}$. The number of bits required to represent an element C^i is $(mn\ell \log_2 q)$. Since m and n are polynomial functions of the security parameter λ, the proposed scheme is compact.

V. Security

We show that the proposed scheme is IND-CPA secure based on the hardness of the Decisional Hidden Subspace Membership (DHSM) problem.

A. Chosen Plaintext Security

In a Chosen Plaintext Attack (CPA) model, the adversary has a number of plaintext-ciphertext pairs at its disposal. A symmetric key encryption scheme is said to be indistinguishable under a chosen plaintext attack (IND-CPA) if, given sufficient samples (plaintext-ciphertext pairs), an adversary is unable to distinguish between the encryptions of two distinct messages of its choice with probability more than $\frac{1}{2}$.

Definition 4: **(IND-CPA Security)**. The IND-CPA security of a symmetric encryption scheme can be defined in terms of the game shown in Figure 3. A PPT adversary $\mathcal{A}$ selects two messages $(\boldsymbol{m}_0, \boldsymbol{m}_1)$ of its choice and the Left-Right oracle outputs the encryption of one of the messages by choosing $c \xleftarrow{\$} \{0,1\}$. $\mathcal{A}$ wins the game if it can guess the value of c with a non-negligible advantage ϵ, where $\epsilon := \left|Pr[c=c'] - \frac{1}{2}\right|$.

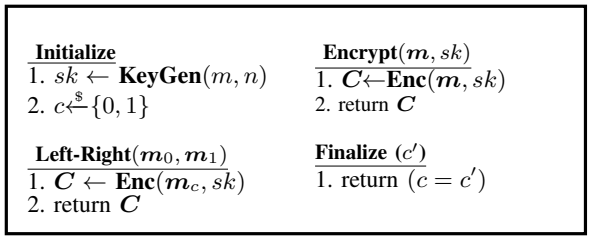

Fig 3: IND-CPA Game

Before proving the security of the proposed scheme, observe that a noise-free variant of the scheme is insecure. It can be easily verified that for $c_i = 0$ in $C^i = L \cdot Q^i \cdot R + c_i \cdot N^i$, the encryptions of zero in the proposed scheme forms an $m(n-1)$ dimensional subspace $\mathcal{V}_0$ of the vector space $\mathbb{F}_{q^\ell}^{m\times n}$. Given sufficient samples, a simple linear algebra attack that recovers a basis of the subspace then distinguishes an encryption of zero from that of a non-zero message.

We now show that the proposed scheme is IND-CPA secure based on the hardness of the DHSM problem. The DHSM problem can be defined with respect to the

proposed scheme in terms of the game shown in Figure 4. The algorithm $\mathcal{G}$ in the following game takes as input (m, n) and generates a basis $B_{\mathcal{S}}$ for a subspace $\mathcal{S} \subset \mathbb{F}_{q^\ell}^{m\times n}$ of dimension $m(n-1)$ such that any $X \in \mathcal{S}$ satisfies the condition, $\sum_{i=1}^{n} X(:, i) = 0$.

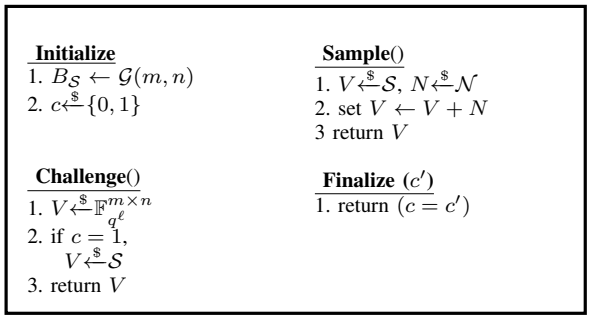

Fig 4: DHSM Game

Theorem 1: A PPT adversary $\mathcal{A}$ that breaks the IND-CPA security of the proposed scheme with non-negligible advantage ϵ can be converted into a PPT adversary $\mathcal{B}$ that can solve an instance of the DHSM problem for the case, where $\mathcal{S} = \mathcal{V}_0$, with advantage at least $\frac{\epsilon}{2}$.

Proof: $\mathcal{B}$ initializes $\mathcal{A}$ with the parameters (m, n). Let $\mathcal{N}'$ be a noise distribution on $\mathbb{F}_{q^\ell}^{m\times n}$. Given $\mathcal{N}'$, we can create a new distribution $\mathcal{N}$ such that $\mathcal{N} := c_0 \cdot \mathcal{N}'$, where c_0 is a random variable that takes values from $\{0, 1\}$ according to a distribution $\mathcal{E}$ as described in section III. When $\mathcal{A}$ asks for an encryption of $\boldsymbol{m}$, $\mathcal{B}$ queries the procedure **Sample** of the DHSM game to get $V_i \leftarrow V_i + N_i$ for $1 \le i \le \eta$, where $N_i \xleftarrow{\$} \mathcal{N}$ and returns the block $\boldsymbol{V} = \{V_i\}_{1\le i\le \eta}$ after setting $V_i(:, \beta) = V_i(:, \beta) + \boldsymbol{m}$ for $1 \le i \le \eta$. Hence, every block of η samples from the **Sample** oracle of the DHSM game contains one element from the subspace $\mathcal{S}$. Therefore, the sample returned to $\mathcal{A}$ is a valid encryption of $\boldsymbol{m}$.

When $\mathcal{A}$ queries the **Left-Right** oracle of the IND-CPA game, $\mathcal{B}$ queries the procedure **Challenge** of the DHSM game to get V and returns a block $\boldsymbol{V} = \{V_i\}_{1\le i\le \eta}$ after choosing $c \xleftarrow{\$} \{0, 1\}$ and setting $V_i(:, \beta) \leftarrow V_i(:, \beta) + \boldsymbol{m}_c$, for some $\beta \xleftarrow{\$} [n]$ such that $V_\gamma = V$ for some $\gamma \xleftarrow{\$} [\eta]$ and $V_i \xleftarrow{\$} \mathbb{F}_{q^\ell}^{m\times n}$ for $1 \le i \le \eta, i \ne \gamma$.

If the sample returned from the **Challenge** oracle to $\mathcal{B}$ is an element of the subspace $\mathcal{S}$, then $\mathcal{A}$ runs in a similar environment to that of the IND-CPA game and hence, $\mathcal{B}$ outputs c with probability $\frac{1}{2} + \epsilon$ which is same as the probability that $\mathcal{A}$ wins the IND-CPA game. On the other hand, if the sample returned is uniform in $\mathbb{F}_{q^\ell}^{m\times n}$, then $\mathcal{B}$ outputs c with probability $\frac{1}{2}$. If $\text{DHSM}_{\mathcal{B}}(\lambda)$ denotes the experiment of the DHSM game, then $\mathcal{B}$ solves the DHSM problem with probability,

$$\begin{aligned} Pr[\text{DHSM}_{\mathcal{B}}(\lambda) = 1] &= \sum_{j\in\{0,1\}} Pr[(c = c') \cap (c = j)] \\ &= \sum_{j\in\{0,1\}} Pr[c = c' | c = j] \cdot Pr[c = j] \\ &\ge (\frac{1}{2} + \epsilon) \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} \\ &= \frac{1}{2} + \frac{1}{2}\epsilon \end{aligned}$$

Hence, $\mathcal{B}$ solves the DHSM problem with advantage at least $\frac{\epsilon}{2}$. ■

B. Exploiting the Public Key

Given the tensor $\mathcal{T}$, an attacker may try to exploit the public key in order to recover the secret key. Observe that, the entries of the public key tensor forms an overdefined system of $(mn)^3$ equations in $m^3 + 2(m^2 + n^2)$ variables of degree $d \le 5$. Therefore, the security of the scheme with respect to the homomorphic properties depends on the problem of solving a system of multivariate polynomial equations over a finite field $\mathbb{F}_{q^\ell}$. This problem is known to be NP-hard in general.

Not many results are known for the exact solvability of this problem in the general case. Further, the algorithms proposed for certain special cases have exponential complexity in the worst case [5], [12]. A recently proposed algorithm in [14] solves this problem that beats brute force search in deciding the satisfiability of the problem. For the proposed parameter choices in section VI, the complexity of this algorithm is exponential in the number of variables.

VI. PARAMETERS

We suggest parameter choices for the proposed scheme based on the attacks discussed in section V. In order to rule out exhaustive search over the key space, we need

$$m \ge \lceil \sqrt[3]{\lambda \log_{q^\ell} 2} \rceil$$

We choose $n \ge m + 1$ because for $n = m$, the ciphertexts associated with the encryptions of zero are rank deficient matrices and an attacker can easily distinguish an encryption of zero from the encryption of a non-zero message.

We need to ensure that the probability of decrypting a valid ciphertext should be ≈ 1. Hence, from Lemma 1, we need

$$(\eta - 1)\, q^{m(1-\ell)} \approx 0$$

We consider q to be polynomial in the security parameter λ. For some $d \in \mathbb{N}$ and $\xi \in \mathbb{R}$ such that $0 \leq \xi \leq 1$, we consider

$$q \approx \lambda^{d+\xi}$$

In the LWE problem [16], the noise distribution is a discrete Gaussian distribution $\mathcal{X}_{\alpha,q}$, where $0 < \alpha \in \mathbb{R}$ and q is polynomial in λ. The hardness of LWE over the extension field $\mathbb{F}_{q^\ell}$ is ensured from the condition that $\alpha q^\ell \geq 2\sqrt{L}$, where L is the dimension of the lattice (which translates to $L = mn$ in the proposed case) and

$$\alpha \geq 1.5 \max\left(\frac{1}{q^\ell}, 2^{-2\sqrt{L \log q \log \delta}}\right)$$

where, δ is the quality of approximation for the shortest vector problem [15]. Assuming a similar noise distribution in the DHSM problem, we consider $\delta = 1.005$, similar to the parameter choices in [1]. Taking these factors into consideration, we provide some example parameter choices for the proposed scheme in Table I.

λ	q	ℓ	m	n	**Ciphertext size ($\approx$)**	**Public Key size ($\approx$)**
80	1109	2	7	15	0.26 kB	3 MB
	15373	3	5	20	0.50 kB	5 MB
	57241	4	3	25	0.81 kB	8 MB
128	1447	2	10	12	0.30 kB	4 MB
	16381	3	6	18	0.55 kB	6 MB
	70237	4	4	26	0.82 kB	8 MB
256	2351	2	11	14	0.42 kB	9 MB
	21617	2	8	13	0.36 kB	4 MB
	114113	3	5	20	0.61 kB	6 MB

TABLE I: Experimental Parameter Choices

The size of the public key is quite large for the parameter choices in Table I. However, it is equivalent to that of other public key homomorphic schemes ([8] achieves a public key size of 10.3 MB for $\lambda = 72$ using public key compression technique for the scheme proposed in [17]).

VII. Conclusion

A new symmetric key encryption scheme has been proposed which is homomorphic with respect to addition and modular convolution. A noisy variant of the subspace membership problem called the Hidden Subspace Membership problem has been introduced and the proposed scheme has been shown to be IND-CPA secure based on its hardness. A possible extension towards future work is to convert the scheme into its public key variant.

References

[1] Martin Albrecht, Pooya Farshim, Jean-Charles Faugere, and Ludovic Perret. Polly cracker, revisited. *Advances in Cryptology–ASIACRYPT 2011*, pages 179–196, 2011.

[2] Frederik Armknecht and Ahmad-Reza Sadeghi. A new approach for algebraically homomorphic encryption. *IACR Cryptology ePrint Archive*, 2008:422, 2008.

[3] Mihir Bellare, Anand Desai, Eron Jokipii, and Phillip Rogaway. A concrete security treatment of symmetric encryption. In *Foundations of Computer Science, 1997. Proceedings., 38th Annual Symposium on*, pages 394–403. IEEE, 1997.

[4] Josh Daniel Cohen Benaloh. *Verifiable secret-ballot elections*. Yale University. Department of Computer Science, 1987.

[5] Luk Bettale, Jean-Charles Faugere, and Ludovic Perret. Hybrid approach for solving multivariate systems over finite fields. *Journal of Mathematical Cryptology*, 3(3):177–197, 2009.

[6] Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-lwe and security for key dependent messages. In *Advances in Cryptology–CRYPTO 2011*, pages 505–524. Springer, 2011.

[7] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. *SIAM Journal on Computing*, 43(2):831–871, 2014.

[8] Jean-Sébastien Coron, David Naccache, and Mehdi Tibouchi. Public key compression and modulus switching for fully homomorphic encryption over the integers. In *EUROCRYPT*, volume 7237, pages 446–464. Springer, 2012.

[9] Ronald Cramer, Ivan Damgård, and Jesper Nielsen. Multiparty computation from threshold homomorphic encryption. *Advances in cryptologyEUROCRYPT 2001*, pages 280–300, 2001.

[10] Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009.

[11] 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.

[12] Neeraj Kayal. Derandomizing some number-theoretic and algebraic algorithms. *Indian Institute of Technology Kanpur*, 2007.

[13] Helger Lipmaa. On diophantine complexity and statistical zero-knowledge arguments. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 398–415. Springer, 2003.

[14] Daniel Lokshtanov, Ramamohan Paturi, Suguru Tamaki, Ryan Williams, and Huacheng Yu. Beating brute force for systems of polynomial equations over finite fields. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2190–2202. SIAM, 2017.

[15] Daniele Micciancio and Oded Regev. Lattice-based cryptography. In *Post-quantum cryptography*, pages 147–191. Springer, 2009.

[16] Oded Regev. The learning with errors problem. *Invited survey in CCC*, page 15, 2010.

[17] Marten Van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 24–43. Springer, 2010.