Noiseless Fully Homomorphic Encryption

Jing Li
School of computer science and software
Gaungzhou University
lijingbeiyou@163.com

Licheng Wang
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts & Telecom.
wanglc2012@126.com

ABSTRACT

We try to propose two fully homomorphic encryption (FHE) schemes, one for symmetric (aka. secret-key) settings and another under asymmetric (aka. public-key) scenario. The presented schemes are noiseless in the sense that there is no "noise" factor contained in the ciphertexts. Or equivalently, before performing fully homomorphic computations, our schemes do not incorporate any noise-control process (such as bootstrapping, modulus switching, etc) to refresh the ciphertexts, since our fully homomorphic operations do not induce any noise. Instead of decrypting approximately, our proposal works in an exact homomorphic manner, no matter the inputs are the first-hand ciphertexts that come from the encryptions of plaintexts, or the second-hand ciphertexts that come from homomorphic combinations of other ciphertexts. Therefore in essential, our schemes have no limitation on the depth of the fully homomorphic operations over the ciphertexts.

Our solution is comprised of three steps. First, Ostrovsky and Skeith's idea for building FHE from a multiplicative homomorphic encryption (MHE) over a non-abelian simple group is extended so that FHE can be built from an MHE over a group ring that takes an underlying non-abelian simple group as the natural embedding. Second, non-trivial zero factors of the underlying ring are plugged into the encoding process for entirely removing the noise after fully homomorphic operations, and a slight but significant modification towards Ostrovsky-Skeith's NAND gate representation is also introduced for avoiding computing inverse matrices of the underlying group ring. In such manner, a symmetric FHE scheme is produced. Finally, based on the proposed symmetric FHE scheme, an asymmetric FHE scheme is built by taking a similar diagram to the well-known GM84 scheme. But different from GM84 that only supports ciphertext homomorphism according to the logically incomplete gate XOR, our scheme supports ciphertext homomorphism according to the logically complete gate NAND.

ACM ISBN 000-0-0000-0000-0.
DOI: 00.0000/0000

Keywords

Noiseless FHE; Finite Non-abelian Simple Group; Group Ring Matrices

1. INTRODUCTION

Being viewed as one of the *holy grails* of modern cryptography [1], fully homomorphic encryption (FHE) is also regarded among the golden keys for securing cloud computation, multi-party computation, data banks [9], etc. After conceptualizing FHE in 1978 [9], it took three decades to discover the first plausible construction of FHE. At STOC 2009, Gentry proposed the first FHE scheme based on ideal lattices [4]. Since then, a lot of developments and improvements were witnessed [1, 10, 11, 12], but all such proposals use essentially the same "noisy" approach in the sense that they encrypt via a noisy encoding of the message, decrypt using an "approximate" ring homomorphism, and have to employ *noise control techniques* to keep a delicate balance between structure and randomness [3]. Although these noise control techniques, such as bootstrapping [4] and modulus switching [1], are full of creative ideas, however, noise control process in general requires computationally expensive steps to bound the noise before fully homomorphic operations over ciphertexts are performed [3], and thus becomes a *leg-pulling* factor towards improving the efficiency of underlying FHE schemes. Therefore, Gentry recently claimed that

> *"I would like nothing more than ... a radically different way of constructing fully homomorphic encryption ... that escapes the current paradigm of using noisy, approximate homomorphisms [3]."*

1.1 Our Methodology and Results

In fact, two years before Gentry's discovery of the first FHE scheme [4], Ostrovsky and Skeith concluded that constructing an FHE scheme is equivalent to constructing a multiplicative homomorphic encryption (MHE) scheme over any finite non-abelian simple group. Their core idea lies in that given an arbitrary finite non-abelian simple group, the logically complete gate NAND (i.e. the combing of the gate AND and the gate NOT) and thus any function $f : \{0,1\}^m \to \{0,1\}^n$ can be composably representable over the underlying group. Moreover, such kind of group-based representation of NAND gate is *noiseless* in sense of the following two features:

- $\mathbf{F_1}$: The corresponding decryption algorithms work in an exact manner, no matter the inputs are the first-hand ciphertexts that come from the encryptions of plaintexts, or the second-hand ciphertexts that come from homomorphic combinations of other ciphertexts;

- $\mathbf{F_2}$: Before performing homomorphic operations or decryption, there is no noise control process to "refresh" the ciphertexts.

Therefore, if a noiseless MHE scheme over some finite non-abelian simple group is attained, then a noiseless FHE scheme can be obtained. But up to now, it is little known how to design a noiseless MHE scheme over finite non-abelian simple groups. We think the main obstacles lie in, at least, the following two limitations:

- $\mathbf{L_1}$: The non-commutativity of the underlying group is both a boon and a bane. On one hand, it is necessary for representing NAND gate by using non-trivial commutators [8]; on the other hand, it also imposes a limitation on the flexibility of cryptographic constructions.

- $\mathbf{L_2}$: The monotonicity of the underlying algebraic operation (i.e. group multiplication) is blamed for. For instance, although the alternative group $\mathbb{A}_5$ is suggested in [8], we face much inconvenience in cryptographic constructions by using only permutations in $\mathbb{A}_5$.

Apparently, the limitation $\mathbf{L_1}$ is rigid and we must obey it; otherwise, we face a new problem of how to represent a logically complete gate. But through careful investigations of the related proofs in [8], we found that the limitation $\mathbf{L_2}$ is comparatively relaxable: If the underlying simple group can be embedded into a ring, then we can use the ring operations, including addition and multiplication for cryptographic constructions.

Based on the aforementioned considerations, we try to construct three noiseless HE schemes, including not only a symmetric MHE, but also a symmetric FHE and an asymmetric FHE. Ostrovsky and Skeith's framework (referred as OS07) is the most important base for our constructions. Besides, we follow the well known Goldwasser-Micali diagram of XOR homomorphic encryption (referred as GM84) [5], with a necessary adaption for transforming the underlying group from the Abelian group $\mathbb{Z}_n$ to the non-Abelian group ring $\mathbb{Z}_n[\mathbb{A}_5]$ that takes the suggested alternative group $\mathbb{A}_5$ as a natural embedding (where n is a big Blum integer).

2. CONSTRUCTIONS: NOISELESS (FULLY) HOMOMORPHIC ENCRYPTIONS

2.1 Review of Ostrovsky-Skeith Framework

Ostrovsky and Skeith [8] presented an elegent framework for building FHE from MHE over non-abelian simple groups. At first, the logically complete gate, NAND, is defined on two special elements of the underlying simple group. Thus, any function $\mathcal{F}:\{0,1\}^m \to \{0,1\}^n$ becomes representable (Lemma 1). Then, an FHE scheme can be obtained by applying the NAND gate to the corresponding MHE's ciphertexts according to the multiplicative homomorphism (Lemma 2).

Lemma 1. (Theorem 4.25 of [8]) Let G be a finite non-abelian simple group. Then any function $\mathcal{F}:\{0,1\}^m \to \{0,1\}^n$ is representable over G.

In the proof of the above lemma, Ostrovsky and Skeith [8] presented that there must exists some product for a 2-order element $x \in G$ such that $x = s_1 \cdots s_l$ $(l \geq 2)$ for some $s_i = [g_i x g_i^{-1}, h_i x h_i^{-1}]$ and $g_i, h_i \in G$ $(i = 1, \cdots, l)$, where the notation $[\cdot,\cdot]$ is the commutator operator. Let e be the the identity of group G. Then, e and x can be viewed as codewords of logic bits 0 and 1, respectively. Now, the NAND gate with two input codewords $g, h \in \langle x\rangle \subset G$ can be represented as follows:[1]

$$\mathrm{NAND}(g,h) = \overline{g \wedge h} = x\prod_{i=1}^{l}[g_i g g_i^{-1},\ h_i h h_i^{-1}].$$

For instance, let $G = \mathbb{A}_5$ be the 5-degree alternating group, and let $e = (1)$, $x = (12)(34)$, $s_1 = (12345)$, $s_2 = (345)$, $g_1 = (354)$, $g_2 = (345)$, $h_1 = (243)$, $h_2 = (1) = e$. Then, it is easy to check that x is an 2-order permutation satisfying $x = (s_1^4 s_2^2)^2 s_1^2$ and

$$s_i = [g_i x g_i^{-1}, h_i x h_i^{-1}],\ (i = 1, 2).$$

From the expression of x, we have that

$$\begin{aligned}\mathrm{NAND}(g,h) &= \overline{g \wedge h}\\ &= x \cdot ([g_1 g g_1^{-1},\ h_1 h h_1^{-1}]^4 [g_2 g g_2^{-1},\ h_2 h h_2^{-1}]^2)^2 \cdot \\ &\quad [g_1 g g_1^{-1},\ h_1 h h_1^{-1}]^2,\end{aligned}$$

and

$$\overline{e \wedge e} = \overline{e \wedge x} = \overline{x \wedge e} = x,\quad \overline{x \wedge x} = e.$$

Finally, the logical completeness of the NAND gate implies that any function $\mathcal{F}:\{0,1\}^m \to \{0,1\}^n$ is also representable over G.

Lemma 2. (Corollary 4.26 of [8]) Designing a fully homomorphic encryption (FHE) scheme over a ring with identity is equivalent to constructing a group multiplication homomorphic encryption (MHE) scheme over any finite non-abelian simple group G.

Suppose such an MHE scheme be at hand, then the corresponding decryption algorithm, denoted by $\mathrm{Dec_M}$, supports multiplication homomorphism over G, i.e.,

$$\mathrm{Dec_M}(sk, \mathrm{Mul}(C_1 C_2)) = \mathrm{Dec_M}(sk, C_1)\mathrm{Dec_M}(sk, C_2) \in G.$$

Thus, the proposed NAND gate in Lemma 1 also works over ciphertexts.

2.2 Our Constructions

To overcome the aforementioned limitation $\mathbf{L_2}$, the 5-degree alternative group $\mathbb{A}_5$ is embedded into the group ring $\mathbb{Z}_n[\mathbb{A}_5]$ (where n is a big Blum integer) that can be regarded as a modular space defined over the basis $\mathbb{A}_5 = \{g_1, \cdots, g_{60}\}$. The *embedding map* is given by

$$\nu:\ G \to \mathbb{Z}_n[\mathbb{A}_5] \subseteq \mathbb{Z}_n^{60},\ \ g_i \mapsto (a_{g_1}, \cdots, a_{g_{60}}), (i = 1, \cdots, 60)$$

with $a_{g_i} = 1$ and $a_{g_j} = 0$ for all $j \neq i$ $(j = 1, \cdots, 60)$. Reversely, we assume that the *unembedding map*, denoted by ν^{-1}, that takes as input a vector $\vec{a} = (a_{g_1}, \cdots, a_{g_{60}}) \in \mathbb{Z}_n^{60}$, will output g_i if the i-th component of $\vec{a}$ is the only non-zero one, and $\perp$ otherwise. Further, an ever large ring,

[1]For visual comfort and layout convenience, we interweavly use two equivalent notations $\mathrm{NAND}(\cdot,\cdot)$ and $\overline{\cdot \wedge \cdot}$ without further explanation.

$M_2(\mathbb{Z}_n[\mathbb{A}_5])$, can be defined as the set of all 2×2 matrices with entries taking from $\mathbb{Z}_n[\mathbb{A}_5]$. More details about group ring and matrices of group ring are given in Appendix A.

2.2.1 Random Homomorphic Encoding Maps

Suppose that the involved group elements $e, x, g_i, h_i \in \mathbb{A}_5$ are defined as the same way in Section 2.1. In addition, all of them, as well as their embedding $\nu(e), \nu(x), \nu(g_i), \nu(h_i) \in \mathbb{Z}_n[\mathbb{A}_5]$, are published publicly in case of necessity. For convenience in subsequent descriptions, we would like to in advance introduce one random map with respect to an invertible matrix $H \in M_2(\mathbb{Z}_n[\mathbb{A}_5])$ and two safe primes p, q:

- $\Phi : \mathbb{A}_5 \to M_2(\mathbb{Z}_n[\mathbb{A}_5])$, named as *free-phi* map that is given by

$$g \mapsto H \begin{pmatrix} pt_1 \cdot \nu(g) + q \cdot \vec{\alpha}_0 & \vec{\alpha}_1 \\ \vec{0} & \vec{\alpha}_2 \end{pmatrix} H^{-1},$$

  where $\vec{\alpha}_0, \vec{\alpha}_1, \vec{\alpha}_2 \in \mathbb{Z}_n[\mathbb{A}_5]$ and $t_1 \in \mathbb{Z}_n^*$ are picked at random.

2.2.2 Symmetirc MHE scheme over $\mathbb{A}_5$

Let n be a big Blum integer[2], i.e., $n = pq$ for two distinct safe primes p and q such that $p \equiv q \equiv 3 \pmod 4$ [6]. Over the group $\mathbb{A}_5$, we try to built a symmetric MHE scheme that consists the following four algorithms:

- $\mathrm{KeyGen_M}$: Select an invertible matrix $H \in M_2(\mathbb{Z}_n[\mathbb{A}_5])$ and output the secret key $sk = (H, p, q)$ and the system public parameter n.

- $\mathrm{Enc_M}(sk, m)$: A message $m \in \mathbb{A}_5$ is encrypted as a matrix $C = \Phi(m)$.

- $\mathrm{Dec_M}(sk, C)$: A ciphertext C is decrypted as $m = \nu^{-1}(p \cdot \vec{w}_{11})$, where $\vec{w}_{11} \in \mathbb{Z}_n[\mathbb{A}_5]$ is the left-top corner entry of the matrix $W = H^{-1}CH$.

- $\mathrm{Mul}(C_1, C_2)$: The multiplicative homomorphism algorithm is provided by Theorem 1.

Theorem 1. If C_1 and C_2 are the valid ciphertexts of m_1 and m_2 respectively, then the multiplication C_1C_2 is a valid ciphertext of the message m_1m_2. Or equivalently,

$$\mathrm{Dec_M}(sk, \mathrm{Mul}(C_1, C_2)) = \mathrm{Dec_M}(sk, C_1)\mathrm{Dec_M}(sk, C_2).$$

In other words, $\mathrm{Dec_M}(sk, \Phi(m_1)\Phi(m_2)) = \mathrm{Dec_M}(sk, \Phi(m_1m_2))$.

The proof of Theorem 1 is deferred a bit later.

Theorem 2. The above symmetric MHE scheme is consistent.

PROOF. Suppose C be a valid ciphertext on a message $m \in \mathbb{A}_5$. That is,

$$C = H \begin{pmatrix} pt_1 \cdot \nu(m) + q \cdot \vec{\alpha}_0 & * \\ \vec{0} & * \end{pmatrix} H^{-1},$$

for some $t_1 \in \mathbb{Z}_n^*$ and $\vec{\alpha}_0 \in \mathbb{Z}_n[\mathbb{A}_5]$, where the second column of two $*$ symbols indicate some random elements in $\mathbb{Z}_n[\mathbb{A}_5]$ that we do not concern. Then, taking as inputs the decryption key (H, p, q) and the ciphertext C, the $\mathrm{Dec_M}$ algorithm will output

$$\begin{aligned} \nu^{-1}(p \cdot (H^{-1}CH)_{11}) &= \nu^{-1}(p \cdot (pt_1 \cdot \nu(m) + q \cdot \vec{\alpha}_0)) \\ &= \nu^{-1}(p^2 t_1 \cdot \nu(m)) \\ &= m, \end{aligned}$$

where the second equality comes from the fact that

$$pq \cdot \vec{\alpha}_0 = n \cdot \vec{\alpha}_0 = 0 \cdot \vec{\alpha}_0 = \vec{0} \in \mathbb{Z}_n[\mathbb{A}_5],$$

while the third equality is derived from Lemma 3 and the fact $p^2 t_1 \not\equiv 0 \pmod n$ considering that t_1 is coprime with n. □

[PROOF OF THEOREM 1.] Suppose C_1 and C_2 be two ciphertexts on messages m_1 and m_2, respectively. That is,

$$C_1 = \Phi(m_1) = H \begin{pmatrix} pt_1 \cdot \nu(m_1) + q \cdot \vec{\alpha}_0 & \vec{\alpha}_1 \\ \vec{0} & \vec{\alpha}_2 \end{pmatrix} H^{-1}$$

and

$$C_2 = \Phi(m_2) = H \begin{pmatrix} pt'_1 \cdot \nu(m_2) + q \cdot \vec{\alpha}'_0 & \vec{\alpha}'_1 \\ \vec{0} & \vec{\alpha}'_2 \end{pmatrix} H^{-1}$$

for some $t_1, t'_1 \in \mathbb{Z}_n^*$ and $\vec{\alpha}_i, \vec{\alpha}'_i \in \mathbb{Z}_n[\mathbb{A}_5]$ $(i = 0, 1, 2)$. Then

$$\mathrm{Mul}(C_1, C_2) \triangleq C_1C_2 = H \begin{pmatrix} \vec{\alpha}^*_0 & \vec{\alpha}^*_1 \\ \vec{0} & \vec{\alpha}^*_2 \end{pmatrix} H^{-1}$$

for some $\vec{\alpha}^*_1, \vec{\alpha}^*_2 \in \mathbb{Z}_n[\mathbb{A}_5]$ that we do not much concern, while

$$\begin{aligned} \vec{\alpha}^*_0 &= (pt_1 \cdot \nu(m_1) + q \cdot \vec{\alpha}_0)(pt'_1 \cdot \nu(m_2) + q \cdot \vec{\alpha}'_0) \\ &= p^2 t_1 t'_1 \cdot (\nu(m_1)\nu(m_2)) + pqt_1 \cdot \nu(m_1)\vec{\alpha}'_0 \\ &\quad + pqt'_1 \cdot \vec{\alpha}_0 \nu(m_2) + q^2 \cdot \vec{\alpha}_0 \vec{\alpha}'_0 \\ &= p^2 t_1 t'_1 \cdot \nu(m_1m_2) + \vec{0} + \vec{0} + q^2 \cdot \vec{\alpha}_0 \vec{\alpha}'_0 \\ &= p^2 \tilde{t}_1 \cdot \nu(m_1m_2) + q^2 \cdot \tilde{\vec{\alpha}}_0 \end{aligned}$$

with $\tilde{t}_1 = t_1 t'_1 \in \mathbb{Z}_n^*$ and $\tilde{\vec{\alpha}}_0 = \vec{\alpha}_0 \vec{\alpha}'_0 \in \mathbb{Z}_n[\mathbb{A}_5]$. Therefore, we have that

$$\begin{aligned} \mathrm{Dec_M}(sk, \mathrm{Mul}(C_1, C_2)) &= \nu^{-1}(p \cdot (H^{-1}\mathrm{Mul}(C_1, C_2)H)_{11}) \\ &= \nu^{-1}(p \cdot \vec{\alpha}^*_0) \\ &= m_1m_2 \\ &= \mathrm{Dec_M}(sk, C_1)\mathrm{Dec_M}(sk, C_2). \end{aligned}$$

In other words, $\mathrm{Dec_M}(sk, \Phi(m_1)\Phi(m_2)) = \mathrm{Dec_M}(sk, \Phi(m_1m_2))$. This is the end of the proof of Theorem 1. □

2.2.3 Symmetirc FHE Scheme over $\mathbb{Z}_2$

Now, a symmetric FHE scheme over the ring $\mathbb{Z}_2$ is given by the following four algorithms:

- $\mathrm{KeyGen_F}$: Produce the secret key $sk = (H, p, q)$ by calling $\mathrm{KeyGen_M}$. Then, output sk and the system public parameters $p_{pub} = (n, K_1, K_2, K_3)$, where

$$K_1 = \Phi(x), \;\; K_2 = \Phi(g_1), \;\; K_3 = \Phi(h_1).$$

- $\mathrm{Enc_F}(sk, m)$: A message $m \in \mathbb{Z}_2$ is encrypted as a matrix $C = \Phi(e)$ if $m = 0$, and $C = \Phi(x)$ otherwise.

[2]Typically, we can choose four large primes p, p_0, q, q_0 such that $p = 2p_0 + 1$ and $q = 2q_0 + 1$ and let $n = pq$.

- $\mathrm{Dec_F}(sk, C)$: A ciphertext C is decrypted as

$$m = f(g) = \begin{cases} 0, & \text{if } g = e \\ 1, & \text{if } g = x \\ \perp, & \text{if } g = \perp \end{cases}$$

where $g = \mathrm{Dec_M}(sk, C)$. That is, for $g \in \{e, x\}$, we have

$$m = \mathrm{Dec_F}(sk, \Phi(g)) = f(\mathrm{Dec_M}(sk, \Phi(g))) = f(g).$$

- $\mathrm{NAND}(C_1, C_2)$: The NAND homomorphism algorithm is given by Theorem 3.

Theorem 3. If C_1 and C_2 are the ciphertexts of two bits m_1 and m_2 respectively, then the multiplication $K_1(\xi_1^4\xi_2^2)^2\xi_1^2$ is a ciphertext of the bit $\mathrm{NAND}(m_1, m_2)$, where

$$\xi_1 = (K_2C_1K_2^2K_3C_2K_3^2)^2, \quad \xi_2 = (K_2^2C_1K_2K_1^2C_2K_1^2)^2.$$

Or equivalently,

$$\mathrm{Dec_F}(sk, \mathrm{NAND}(C_1, C_2)) = \mathrm{NAND}(\mathrm{Dec_F}(sk, C_1), \mathrm{Dec_F}(sk, C_2)).$$

Remark 1. Note that the definition of the NAND gate over ciphertexts in our scheme is slightly different but essentially coherent with Ostrovsky and Skeith's framework [8]. In [8], the commutators over non-abelian groups are directly used to define NAND gate. However, we face the difficulty in directly computing commutators over ciphertexts, since our ciphertexts lie in the group ring matrices $M_2(\mathbb{Z}_n[\mathbb{A}_5])$ and they might not be invertible. Fortunately, this difficulty does not affect much on the feasibility of the FHE schemes. Actually, based on Theorem 1, we let the commutator operation directly act on the preimage set of function Φ. In other words, to avoid computing inversion over $M_2(\mathbb{Z}_n[\mathbb{A}_5])$, we adopt a *substitution* strategy based on the following observations:

- Suppose that $r = \mathrm{ord}(g)$ is the order of $g \in \mathbb{A}_5$. That is, $g^{r-1} = g^{-1}$. Then, in the NAND algorithm, any required appearance of the "inverse" of $\Phi(g)$ can be replaced with $\Phi(g)^{r-1}$, no matter whether $\Phi(g)$ is inverse or not. This is correct since $\mathrm{Dec_M}(sk, \Phi(g)^{r-1}) = \mathrm{Dec_M}(sk, \Phi(g^{r-1})) = \mathrm{Dec_M}(sk, \Phi(g^{-1}))$ based on the multiplication homomorphism of the MHE scheme.
- The feasibility of our FHE schemes depends on the correctness of the NAND gate on the preimages of Φ and the multiplication homomorphism of the MHE scheme. Thus, designing a FHE scheme over $\mathbb{Z}_2$ is equivalent to construct a MHE scheme with a random mapping Φ from a finite non-abelian simple group to a ring such that $\mathrm{Dec_M}(sk, \Phi(g_1)\Phi(g_2)) = \mathrm{Dec_M}(sk, \Phi(g_1g_2))$. In other words, the ciphertexts of the MHE scheme can be allowed to lie in a ring, but not necessarily a group.

2.2.4 Asymmetric FHE scheme over $\mathbb{Z}_2$

To proceed, an asymmetric FHE scheme over the ring $\mathbb{Z}_2$ is given by the following four algorithms:

- KeyGen: Produce the secret key $sk = (H, p, q)$ by calling $\mathrm{KeyGen_M}$. Then, output sk and the corresponding public key $pk = (n, K_1, K_2, K_3)$, where

$$K_1 = \Phi(x), \ K_2 = \Phi(g_1), \ K_3 = \Phi(h_1).$$

- $\mathrm{Enc}(pk, m)$: A message $m \in \mathbb{Z}_2$ is encrypted as a matrix

$$C = K_1^{2b_1}K_2^{3b_2}K_3^{3b_3}$$

if $m = 0$, and

$$C = K_1^{2b_1+1}K_2^{3b_2}K_3^{3b_3}$$

otherwise, where $b_i \in \{0,1\}^\lambda$ $(i = 1, 2, 3)$ are picked at random. Here, λ is the bit length of b_i $(i = 1, \cdots, 3)$, and it should be large enough for resisting brute force attack. In practice, the sum of the lengthes of b_i $(i = 1, 2, 3)$ is set to 160.

- $\mathrm{Dec}(sk, C)$: Same as $\mathrm{Dec_F}(sk, C)$ in Section 2.2.3.
- $\mathrm{NAND}(C_1, C_2)$: Same as $\mathrm{NAND}(C_1, C_2)$ in Section 2.2.3.

2.2.5 Noiseless Features Analysis

At first, as for the symmetric MHE scheme, the proof of the consistency (i.e. Theorem 2) suggests that on input a first-hand ciphertext $C = \mathrm{Enc_M}(sk, m)$, the decryption algorithms $\mathrm{Dec_M}(sk, C)$ outputs the corresponding plaintext $m \in \mathbb{A}_5$ *in an exact manner*. For a second-hand ciphertext $C = \mathrm{Mul}(C_1, C_2) = C_1C_2$, the proof of Theorem 1 also suggests that on one hand, C takes the same form as that of the first hand ciphertexts, and on the other hand, $\mathrm{Dec_M}(sk, \mathrm{Mul}(C_1, C_2))$ also outputs $\mathrm{Dec_M}(sk, C_1)\mathrm{Dec_M}(sk, C_2) \in \mathbb{A}_5$ *in an exact manner*, no matter C_i is a first-hand ciphertext (i.e. $C_i = \mathrm{Enc_M}(m_i)$ for some $m_i \in \mathbb{A}_5$) or a second-hand ciphertext (i.e. $C_i = \mathrm{Mul}(C_{i1}, C_{i2})$) $(i = 1, 2)$. Thus, the proposed MHE scheme meets the so-called noise-freeness feature $\mathbf{F_1}$ mentioned in Introduction.

Similarly, as for the symmetric FHE scheme, a first-hand ciphertext $C = \mathrm{Enc_F}(sk, m)$ (where $m \in \{0, 1\}$) will also be decrypted in an exact manner, since $\mathrm{Dec_F}(sk, C) = f(\mathrm{Dec_M}(sk, C))$ and f is an exact, well-defined map. As for a second-hand ciphertext $C = \mathrm{NAND}(C_1, C_2) = K_1(\xi_1^4\xi_2^2)^2\xi_1^2$, where $\xi_1 = (K_2C_1K_2^2K_3C_2K_3^2)^2$, $\xi_2 = (K_2^2C_1K_2K_1C_2K_1)^2$, the proof of Theorem 2 implies that the decryption algorithm $\mathrm{Dec_F}(sk, \mathrm{NAND}(C_1, C_2))$ outputs a bit $\mathrm{NAND}(\mathrm{Dec_F}(sk, C_1), \mathrm{Dec_F}(sk, C_2)) \in \{0, 1\}$ *in an exact manner* according to the OS07 framework. Thus, the proposed symmetric FHE scheme caters to the noise-freeness feature $\mathbf{F_1}$, too.

As for the asymmetric FHE scheme, the decryption algorithm Dec and the NAND homomorphism algorithm are directly inherited from the symmetric FHE scheme. That is, the proposed asymmetric FHE scheme also satisfies the noise-freeness feature $\mathbf{F_1}$.

Moreover, before performing decryption or the multiplicative (resp. NAND) homomorphic combinations, there is no additional noise cleaning or squashing process to "refresh" the ciphertexts. In fact, our trick lies in the encoding diagram used in definitions of the random maps Φ and Φ_γ. That is, the left-top corner elements is encoded in such a way that the products of two valid codewords automatically remove the noise term by taking modulo n (cf. the term $\vec{\alpha}_0^*$ in the proof of the Theorem 1). By doing so, the proposed three (fully) homomorphic encryption schemes meet the aforementioned noise-freeness feature $\mathbf{F_2}$, too.

Therefore, our proposals are noiseless in the sense that all of them meets the noise-freeness features $\mathbf{F_1}$ and $\mathbf{F_2}$.

Remark 2. In an even abstract perspective, our trick for achieving noiseless FHE schemes comes from the following simple observation: If the non-trivial zero factors (i.e. p and q in our proposals) were introduced into the encoding process, then after multiplication of two ciphertexts, the unexpected noise terms becomes zeros automatically. That is,

$$\begin{aligned} & (pt_1 \cdot \nu(m_1) + q \cdot \vec{\alpha}_0)(pt_1' \cdot \nu(m_2) + q \cdot \vec{\alpha}_0') \\ = \; & p^2 t_1 t_1' \cdot \nu(m_1 m_2) + \vec{0} + \vec{0} + q^2 \cdot \vec{\alpha}_0 \vec{\alpha}_0'. \end{aligned}$$

This observation also encourages us to propose the following conjecture that might have independent interests.

Conjecture 1. In our proposals, the underlying ring $\mathbb{Z}_n$ could be securely replaced by other ring R, if given an explicit description of R, it is still difficult in finding non-trivial zero factors of R.

3. FURTHER DISCUSSION

Lastly, we would like to present further explanation on some tricks used in our constructions.

3.1 H's Conjugation Actions

Before seeing H's role, we might need to notice the following *weird* facts about determinants and characteristic polynomials of matrices over a non-commutative ring $\mathcal{R}$:

- $\det(AB) \neq \det(A)\det(B)$ in general for $A, B \in M_2(\mathcal{R})$;
- $\chi(BAB^{-1}) \neq \chi(A)$ in general for $A, B \in M_2(\mathcal{R})$.

Then, in our constructions, suppose C be a MHE ciphertext on a message $m \in \mathbb{A}_5$. That is,

$$C = H \begin{pmatrix} pt_1 \cdot \nu(m) + q \cdot \vec{\alpha}_0 & \vec{\alpha}_1 \\ \vec{0} & \vec{\alpha}_2 \end{pmatrix} H^{-1},$$

for some $t_1 \in \mathbb{Z}_n^*$ and $\vec{\alpha}_0, \vec{\alpha}_1, \vec{\alpha}_2 \in \mathbb{Z}_n[\mathbb{A}_5]$. Based on the above two facts, none can work out $pt_1 \cdot \nu(m) + q \cdot \vec{\alpha}_0$ by computing $\det(C)$ or $\chi(C)$. In fact, one of our core tricks is to protect $pt_1 \cdot \nu(m) + q \cdot \vec{\alpha}_0$ by employing the conjugate action of H. In detail, suppose $H = \begin{pmatrix} \vec{h}_1 & \vec{h}_2 \\ \vec{h}_3 & \vec{h}_4 \end{pmatrix}$, $H^{-1} = \begin{pmatrix} \vec{y}_1 & \vec{y}_2 \\ \vec{y}_3 & \vec{y}_4 \end{pmatrix}$. The ciphertext is equal to $C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$, where

$$\begin{aligned} C_{11} &= pt_1 \cdot \vec{h}_1 \nu(m) \vec{y}_1 + q \cdot \vec{h}_1 \vec{\alpha}_0 \vec{y}_1 + \vec{h}_1 \vec{\alpha}_1 \vec{y}_3 + \vec{h}_2 \vec{\alpha}_2 \vec{y}_3, \\ C_{12} &= pt_1 \cdot \vec{h}_3 \nu(m) \vec{y}_1 + q \cdot \vec{h}_3 \vec{\alpha}_0 \vec{y}_1 + \vec{h}_3 \vec{\alpha}_1 \vec{y}_3 + \vec{h}_4 \vec{\alpha}_2 \vec{y}_3, \\ C_{21} &= pt_1 \cdot \vec{h}_1 \nu(m) \vec{y}_2 + q \cdot \vec{h}_1 \vec{\alpha}_0 \vec{y}_2 + \vec{h}_1 \vec{\alpha}_1 \vec{y}_4 + \vec{h}_2 \vec{\alpha}_2 \vec{y}_4, \\ C_{22} &= pt_1 \cdot \vec{h}_3 \nu(m) \vec{y}_2 + q \cdot \vec{h}_3 \vec{\alpha}_0 \vec{y}_2 + \vec{h}_3 \vec{\alpha}_1 \vec{y}_4 + \vec{h}_4 \vec{\alpha}_2 \vec{y}_4. \end{aligned}$$

Due to the randomness of $\vec{\alpha}_i$ and H, each component of C_{ij} $(1 \leq i, j \leq 2)$ contains sufficient randomness for hiding p, q and $\nu(m)$. Furthermore, H cannot be a triangular matrix; otherwise, the modulus n can be factorized. To see this, suppose $H = \begin{pmatrix} \vec{h}_1 & * \\ & \vec{h}_4 \end{pmatrix}$, then $H^{-1} = \begin{pmatrix} \vec{h}_1^{-1} & *' \\ & \vec{h}_4^{-1} \end{pmatrix}$. Then for the message $m = e$, we have $C_{11} = pt_1 \cdot \nu(e) + q \cdot \vec{h}_1 \vec{\alpha}_0 \vec{h}_1^{-1}$, where $\nu(e) = (1, 0, \ldots, 0)$. Let $\vec{h}_1 \vec{\alpha}_0 \vec{h}_1^{-1} = (l_1, \ldots, l_{60}) \in \mathbb{Z}_n[\mathbb{A}_5]$. Hence, $C_{11} = (pt_1 + ql_1, ql_2, \ldots, ql_{60})$ and $q = \gcd(n, ql_2)$.

3.2 Relations to GM84 Diagram

Compared to the well-known GM84 diagram, our construction of asymmetric FHE scheme has the following similarities.

- Similarities. Both the GM84 scheme and our asymmetric FHE scheme can be represented by the following common framework
$$C = Y^{2b_1 + m} \cdot X,$$
where $m \in \{0, 1\}$ is the message to be encrypted, b_1 is a random integer, and the similarities of the terms Y and X are depicted in Table 1. That is, the Y-term in GM84 is specified by a non-quadratic residue, while the Y-term in our construction is an encryption of non-identity, more precisely, a 2-order element. Meanwhile, the X-term in both schemes plays the role of introducing necessary randomness: In GM84, it is a random quadratic residue, while in our scheme it is a random encryption of identity.

Table 1: Similarities of the GM84 scheme and Ours

	Y	Y's meaning	X	X's meaning
GM84:	y	non-quadratic residue	x^2	quadratic residue
Ours:	K_1	Enc of non-identity	$K_2^{3b_2} K_3^{3b_3}$	Enc of identity

Actually, the encryption algorithm of the asymmetric FHE scheme can be given as $C = K_1^{2b_1+m} K_2^{3b_2}$ or $C = K_1^{2b_1+m} K_3^{3b_3}$ for message $m \in \{0, 1\}$.

4. REFERENCES

[1] Z. Brakerski and V. Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In P. Rogaway, editor, *CRYPTO*, volume 6841 of *Lecture Notes in Computer Science*, pages 505–524. Springer, 2011.

[2] C. W. Curtis and I. Reiner. *Representation Theory of Finite Groups and Associative Algebras*. Number XI in Pure and Applied Mathematics. Interscience Publishers, 1962.

[3] C. Gentry. Computing on the edge of chaos: Structure and randomness in encrypted computation. *IACR Cryptology ePrint Archive*, 2014:610, 2014.

[4] C. Gentry et al. Fully homomorphic encryption using ideal lattices. In *STOC*, volume 9, pages 169–178, 2009.

[5] S. Goldwasser, S. Micali, and A. Wigderson. Probabilistic encryption. *Journal of Computer and System Sciences*, 28:270–299, Apr. 1984.

[6] J. Hastad, A. W. Schrift, and A. Shamir. The discrete logarithm modulo a composite hides $O(n)$ bits. *Journal of Computer and Systems Sciences*, 47:376–404, 1993.

[7] A. D. Myasnikov and A. Ushakov. Quantum algorithm for discrete logarithm problem for matrices over finite group rings. *Groups Complexity Cryptology*, 6(1):31–36, 2014.

[8] Ostrovsky and Skeith. Algebraic lower bounds for computing on encrypted data. In *ECCCTR: Electronic Colloquium on Computational Complexity, technical reports*, 2007.

[9] R. L. Rivest, L. Adleman, and M. L. Dertouzos. On data banks and privacy homomorphisms. *Foundations of Secure Computation*, pages 169–179, 1978.

[10] P. Scholl and N. Smart. Improved key generation for gentrya's fully homomorphic encryption scheme. In *Proc. Cryptography and Coding*, pages 10–22. Springer LNCS 7089, Dec. 2011.

[11] N. P. Smart and F. Vercauteren. Fully homomorphic SIMD operations. *Des. Codes Cryptography*, 71(1):57–81, 2014.

[12] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. *IACR Cryptology ePrint Archive*, 2009:616, 2009.

APPENDIX

A. GROUP RING

Definition 1. (Group Ring [2]) Let G be a group and R a ring. The group ring $R[G]$ is the set of all formal sums $\sum\limits_{g_i \in G} a_{g_i} \cdot g_i$ (where $a_{g_i} \in R$) with the addition and multiplication defined below:

$$\sum_{g_i \in G} a_{g_i} \cdot g_i + \sum_{g_i \in G} b_{g_i} \cdot g_i = \sum_{g_i \in G} (a_{g_i} + b_{g_i}) \cdot g_i.$$

$$\left(\sum_{g_i \in G} a_{g_i} \cdot g_i \right) \left(\sum_{g_i \in G} b_{g_i} \cdot g_i \right) = \sum_{g_i \in G} \sum_{g_j g_k = g_i} (a_{g_j} b_{g_k}) \cdot g_i.$$

It is easy to check that the group ring $R[G]$ is indeed a ring. If we represent a group ring element as a vector $\vec{\alpha} = (a_1, a_2, \cdots, a_{|G|})$, then the addition is a direct sum of vectors and the multiplication is a *twist product* of vectors.

Definition 2. (Matrices of Group Rings [7]) A matrix over a group ring $R[G]$ is a matrix in which the entries are taken from the group ring $R[G]$. Usually, the ring of $d \times d$ matrix over $R[G]$ is denoted by $M_d(R[G])$.

For the given commutative ring R and the group ring $R[G]$ with $|G| = k$, Myasnikov and Ushakov [7] built the following ring monomorphism

$$\phi : M_d(R[G]) \to M_{d \cdot k}(R), \quad A \mapsto A^*$$

with

$$A^* = \begin{pmatrix} \psi(\vec{a}_{1,1}) & \cdots & \psi(\vec{a}_{1,d}) \\ \vdots & \ddots & \vdots \\ \psi(\vec{a}_{d,1}) & \cdots & \psi(\vec{a}_{d,d}) \end{pmatrix},$$

where $\psi : R[G] \to M_k(R)$ is also a ring monomorphism that is defined as follows:

$$\vec{a} = (a_{g_1}, \cdots, a_{g_k}) \mapsto \begin{pmatrix} a_{g_1 g_1^{-1}} & \cdots & a_{g_1 g_k^{-1}} \\ \vdots & \ddots & \vdots \\ a_{g_k g_1^{-1}} & \cdots & a_{g_k g_k^{-1}} \end{pmatrix}.$$

In this paper, we adopt the settings $R = \mathbb{Z}_n$ for a big Blum integer n, $G = \mathbb{A}_5$, and $d = 2$. Moreover, we have the following observations.

Lemma 3. Over the group ring $\mathbb{Z}_n[\mathbb{A}_5]$, we have that

- For any $t_1 \in \mathbb{Z}_n$ and $\vec{\alpha}_1, \vec{\alpha}_2 \in \mathbb{Z}_n[\mathbb{A}_5]$, $(t_1 \cdot \vec{\alpha}_1)(\cdot \vec{\alpha}_2) = t_1 \cdot (\vec{\alpha}_1 \vec{\alpha}_2)$.
- For $\forall m_1, m_2 \in \mathbb{A}_5$, $\nu(m_1)\nu(m_2) = \nu(m_1 m_2)$.
- For any positive integer z, if $z \not\equiv 0 \pmod{n}$, then $\nu^{-1}(z \cdot \nu(m)) = m$ for $\forall m \in \mathbb{A}_5$.

Here, ν an ν^{-1} are respectively the embedding and unembedding maps given in the beginning of Section 2.2.