Implementation of Learning with Errors in Non-Commuting Multiplicative Groups

Aleksejus Mihalkovich[1], Lina Dindiene[2] and Eligijus Sakalauskas[3]

[1] Kaunas University of Technology, Kaunas LT-44249, Lithuania
aleksejus.michalkovic@ktu.lt
[2] lina.dindiene@ktu.lt
[3] eligijus.sakalauskas@ktu.lt

Abstract. In this paper, we demonstrate a way to generalize learning with errors (LWE) to the family of so-called modular-maximal cyclic groups which are non-commuting. Since the group $\mathbb{M}_{2^t}$ has two cycles of maximal multiplicative order, we use this fact to construct an accurate criterion for restoring the message bit with overwhelming probability. Furthermore, we implement the original idea by O. Regev in the considered group to gain benefits from the non-commutativity of $\mathbb{M}_{2^t}$. Also we prove that using this approach we can achieve a level of security comparable to the original idea.

Keywords: Learning with errors · Matrix power function · Public key encryption · Post-quantum cryptography.

1 Introduction and Previous Work

At the end of the XX century P. Shor in his paper [1] proposed an algorithm for solving the discrete logarithm and integer factorization problems in polynomial time using quantum computations. Therefore, cryptographic primitives based on such problems are on the edge of extinction due to ongoing progress in the development of quantum processors. To our knowledge, as of September 2024, the record holders for most qubits are IBM with their 1121-qubit Condor [2] and Atom Computing with their 1180-qubit computer [3]. Moreover, IBM can offer 10 minutes per month of free access to their 127-qubit system to get a handle on quantum computing [4].

The threat of quantum cryptanalysis to the widely used classic algorithms like the RSA cryptosystem is so great that at the end of 2016, NIST announced a call for proposals for quantum-safe cryptographic algorithms [5]. As of 2022, one public-key encryption and three digital signature schemes have been selected [6].

Three out of four selected schemes use hard problems defined in lattices [7, 8, 9]. The one we consider in this paper is the learning with errors (LWE) problem first introduced by M. Ajtai in his paper [10]. This problem was originally formulated for m-dimensional lattice defined in the finite field $\mathbb{Z}_q$, where q is prime, as follows:

Definition 1. Let $\mathbf{A}$ be an $m \times n$ matrix with entries $a_{ij} \in \mathbb{Z}_q$ sampled uniformly at random and let χ be a probability distribution on $\mathbb{Z}_q$. The $LWE(\mathbb{Z}_q, \chi)$ problem is to recover a secret vector $\vec{x} \in \mathbb{Z}_q^n$ given a pair $(\mathbf{A}, \vec{b})$, where $\vec{b} = \mathbf{A} \cdot \vec{x} + \vec{\varepsilon}$ and the errors $\varepsilon \in \mathbb{Z}_q$ are sampled according to χ, with probability exponentially close to 1.

The basic idea behind LWE is using error terms added to linear equations. This composes a problem that is computationally infeasible to solve without revealing the

secret vectors. O. Regev used this idea in his paper [11] to define the following public key encryption (PKE) scheme – a triplet of algorithms (KeyGen, Enc, Dec):

PKE Scheme 1. *Let m and n denote the number of equations and unknowns respectively. For a prime number $q \geq 2$ between n^2 and $2n^2$ let χ denote the probability distribution on $\mathbb{Z}_q$. Also, let $m = (1+\epsilon)(n+1)\log q$ for some constant $\epsilon > 0$. Then two entities called Alice and Bob, where Alice is the receiver and Bob is the sender, can execute the following PKE scheme:*

- KeyGen *algorithm:*
  - *Alice chooses a secret vector $\vec{x} \in \mathbb{Z}_q^n$ uniformly at random, where n is the security parameter of the scheme. Her private key is $\vec{x}$.*
  - *For $i = 1, \ldots, n$, Alice chooses m vectors $\vec{a}_1, \vec{a}_2, \ldots, \vec{a}_m \in \mathbb{Z}_q^m$ independently from the uniform distribution. Also she chooses elements $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m \in \mathbb{Z}_q$ independently according to χ. Her public key is given by $(\mathbf{A}, \vec{b})$, where $\mathbf{A}$ is an $m \times n$ matrix with columns $\vec{a}_1, \vec{a}_2, \ldots, \vec{a}_m$, and $\vec{b} = \mathbf{A}\vec{x} + \vec{\varepsilon}$.*
- Enc *algorithm:*
  - *To encrypt a bit, Bob chooses k equations by picking a random set of row indices among all 2^m subsets of the set $[m] = \{1, 2, \ldots, m\}$. We denote this set by $\mathbb{S} = \{i_1, i_2, \ldots, i_k\}$.*
  - *Bob adds the chosen equations together to obtain an expression:*

$$\Big(\sum_{i\in\mathbb{S}} a_{i1}\Big)x_1 + \Big(\sum_{i\in\mathbb{S}} a_{i2}\Big)x_2 + \ldots + \Big(\sum_{i\in\mathbb{S}} a_{im}\Big)x_m + \Big(\sum_{i\in\mathbb{S}} \varepsilon_i\Big) = \sum_{i\in\mathbb{S}} b_i. \quad (1)$$

  - *If the original bit is 0, the encryption is a pair $\Big(\vec{a}, b\Big)$, where $\vec{a} = \sum_{i\in\mathbb{S}} \vec{a}_i$ and $b = \sum_{i\in\mathbb{S}} b_i$.*
  - *If the original bit is 1, the encryption is $\Big(\vec{a}, \lfloor \frac{p}{2} \rfloor + b\Big)$.*
- Dec *algorithm:*
  - *To decrypt a bit from the pair $(\vec{a}, b)$ Alice calculates $d = b - \vec{a} \cdot \vec{x}$, where $\cdot$ denotes the inner product of two vectors $\vec{a}$ and $\vec{x}$.*
  - *Alice outputs 0 if d is closer to 0 than to $\frac{q}{2}$. Otherwise, she outputs 1.*

Analyzing the PKE Scheme 1 we can see that the extra term $\lfloor \frac{q}{2} \rfloor$ used during encryption to encrypt 1 is exactly the middle of the group $\mathbb{Z}_q$. Also note that during decryption the subtraction of the term $\vec{a} \cdot \vec{x}$ can always be performed since $\mathbb{Z}_q$ is a commuting group under addition. Furthermore, the decision Alice makes during the decryption step to restore the original bit relies on the properties of χ, which is usually chosen to be the Gaussian distribution.

Because LWE is conjectured to be intractable for quantum computers, this type of cryptography has become an important element in the search for a secure way for parties to communicate in today's digitized society. As a result over the recent years encryption-decryption scheme [12, 13], digital signature [14], and fully homomorphic encryption [15, 19] were created. Moreover, LWE can also be used to provide zero-knowledge proof [18, 20].

In 2010 Lyubashevsky et al. [21] extended the LWE problem to ring-based structures. This extension paved the way for a new modified LWE method. Ring LWE (RLWE) was established to be more efficient in creating cryptographical primitives and developing

lattice-based cryptosystems. Since it reduces the computational complexity by leveraging the structure of polynomial rings, ring LWE is the most significant and widely analysed modification to LWE [22, 23].

Various modifications of algebraically structured LWE were introduced later [25]. Module learning with errors problem (MLWE) is a generalization of RLWE and was proven to be at least as hard as ring LWE problem [24]. Another generalization of RLWE among others is order-LWE where the distribution of secret elements has a specific algebraic structure [26]. Thus, the potential of LWE in cryptography is obvious and it seems that the possibilities of using this type of cryptographic primitives have not yet been exhausted.

Another branch of cryptography which is currently considered perspective is based on the study of hard problems defined in non-commuting groups. The first steps in this direction were made on the verge of the millennium, and so cryptographic primitives like Ko-Lee and Anshel-Anshel-Goldfield key exchange protocols were presented [27, 28].

In this paper, we also use the so-called right-sided matrix power function (RMPF) first introduced in [29] as an example of the one-sided MPF. Formally, MPF is a mapping defined for a base matrix $\mathbf{W}$, whose entries come from a multiplicative group $\mathbb{G}$ (abelian or otherwise), and takes either one (for one-sided MPFs) or two (for the two-sided MPF) scalar matrices $\mathbf{X}$ and $\mathbf{Y}$, whose entries come from a ring of integers $\mathbb{Z}_{\text{ord}(\mathbb{G})}$, as an input [29, 30, 31]. Here the notation ord($\mathbb{G}$) stands for the multiplicative order of the group $\mathbb{G}$. The explicit definition of the right-sided MPF is as follows [29]:

Definition 2. Let $\mathbf{W}$ be a $m \times n$ matrix with entries $w_{ij} \in \mathbb{G}$, where $\mathbb{G}$ is a multiplicative group, and let $\mathbf{X}$ be an $n \times p$ matrix with entries $x_{ij} \in \mathbb{Z}_{\text{ord}(\mathbb{G})}$. The right-sided MPF is a mapping $\mathbb{G}^{m\times n} \times \mathbb{Z}^{n\times p}_{\text{ord}(\mathbb{G})} \to \mathbb{G}^{m\times p}$ denoted as

$$\text{RMPF}_{\mathbf{W}}(\mathbf{X}) = \mathbf{W}^{\mathbf{X}},$$

where the entries of matrix $\mathbf{V} = \text{RMPF}_{\mathbf{W}}(\mathbf{X})$ are calculated as follows:

$$v_{ij} = \prod_{k=1}^{n} w_{ik}^{x_{kj}}. \tag{2}$$

The left-sided MPF can be defined in a similar way. Furthermore, the two-sided MPF can be defined as a combination of both one-sided MPFs [29]. However, since these mappings are not used in this paper, we omit their explicit definitions.

Since in this paper, the right-sided MPF is also used as an analog of the inner product for vectors $\vec{w} \in \mathbb{G}^m$ and $\vec{x} \in \mathbb{Z}^m_{\text{ord}(\mathbb{G})}$ we introduce an extra notation $\vec{w}^{\vec{x}}$ to denote the following operation:

$$\vec{w}^{\vec{x}} = \prod_{i=1}^{m} w_i^{x_i}. \tag{3}$$

Formally, according to the definition of the right-sided MPF, the vector $\vec{w}$ in (3) should be transposed. However, since the upper index denoting transposing of $\vec{w}$ is inconvenient, we omit it in (3) and onward.

Over the years several papers (see, for example [32, 33, 34]) have been published, where the authors have shown how MPFs can be applied in cryptography. Their results have shown that problems based on MPFs are hard and can potentially be quantum resistant [35, 36].

In this paper, we demonstrate how to expand the LWE problem to non-commuting groups therefore combining two perspective branches of cryptography together. We also present a security analysis of the presented schemes and compare them to the original Regev's idea. As usual, we present conclusions at the end of this paper.

Note, however, that in this paper we establish a basis of our future work: we plan to explore the possibility of encrypting multiple bits, just as was done after the introduction of the original Regev's PKE.

2 A direct multiplicative approach to the generalization of Regev's PKE

In this section, we present the simplest way to generalize Regev's PKE to multiplicative algebraic structures. Looking back at the PKE Scheme 1, we can make the following changes to adapt it for a multiplicative group $\mathbb{G}_q$:

PKE Scheme 2. *Security parameters m and n are defined as in PKE Scheme 1. For a prime number $q \geq 2$ between n^2 and $2n^2$ let $\mathbb{G}_q$ be the Sylow subgroup of $\mathbb{Z}_p^*$, where $p = \alpha q + 1$ is a prime and $\gcd(\alpha, q) = 1$. Also, let χ denote the discrete Gaussian distribution $\mathcal{N}(0, \sigma)$ on $\mathbb{Z}_q$.*

- KeyGen *algorithm:*
  - *Alice chooses the coefficients a_{ij} of the equations from $\mathbb{G}_q$ uniformly at random. Note that every element in $\mathbb{G}_q$ has a multiplicative order of a prime number $q|(p-1)$.*
  - *The secret key is now a vector of exponents $\vec{x}$, where $x_i \in \mathbb{Z}_q$.*
  - *Alice generates the error terms ε_i of the form $\varepsilon_i = g^{r_i}$, where g is a generator of $\mathbb{G}_q$ and $r_i \in \mathbb{Z}_q$ independently according to distribution χ. Therefore, Alice obtains the following system of equations defined in $\mathbb{G}_q$:*

$$\begin{cases} a_{11}^{x_1} & a_{12}^{x_2} & \dots & a_{1m}^{x_m} & \varepsilon_1 & = & b_1 \\ a_{21}^{x_1} & a_{22}^{x_2} & \dots & a_{2m}^{x_m} & \varepsilon_2 & = & b_2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ a_{m1}^{x_1} & a_{m2}^{x_2} & \dots & a_{mm}^{x_m} & \varepsilon_m & = & b_m \end{cases} \tag{4}$$

- *Alice's private key is the vector $\vec{x}$. Her public key is $\vec{b}$, where the entries b_i are defined by the system of equations (4).*
- Enc *algorithm:*
  - *To encrypt a bit, Bob chooses k equations. We denote the set of chosen rows by $\mathbb{S} = \{i_1, i_2, \dots, i_k\}$. Bob multiplies these equations in $\mathbb{G}_q$ to obtain the following result:*

$$\Big(\prod_{i\in\mathbb{S}} a_{i1}^{x_1}\Big) \cdot \Big(\prod_{i\in\mathbb{S}} a_{i2}^{x_2}\Big) \cdot \ldots \cdot \Big(\prod_{i\in\mathbb{S}} a_{im}^{x_m}\Big) \cdot \Big(\prod_{i\in\mathbb{S}} \varepsilon_i\Big) = \prod_{i\in\mathbb{S}} b_i. \tag{5}$$

    Let us denote $a_j = \prod_{i\in\mathbb{S}} a_{ij}^{x_j}$, $b = \prod_{i\in\mathbb{S}} b_i$.
  - *If a bit is 0, the encryption is $(\vec{a}, b)$, and if the bit is 1, the encryption is $(\vec{a}, g^{\lfloor \frac{q}{2} \rfloor} b)$.*
- Dec *algorithm:*
  - *To decrypt a bit, calculate $b(\vec{a}^{\vec{x}})^{-1} = g^r$, where $\vec{a}^{\vec{x}} = \prod_{i=1}^m a_i^{x_i}$.*
  - *Alice makes a decision based on the values of g^r. If r is closer to 0 than to $\lfloor \frac{q}{2} \rfloor$ then the decrypted bit is 0. Otherwise, it is 1.*

However, this approach is of little interest to us, since it is basically the original idea written in terms of multiplicative notations. Furthermore, taking the discrete logarithm of all equations base g we have the original Regev's scheme defined in $\mathbb{Z}_q$. On the other hand, for this reason the presented scheme is as secure as the original idea.

Interestingly enough, the presented scheme can also be rewritten using matrix notation. Let us define:

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mm} \end{pmatrix}, \vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_m \end{pmatrix}, \vec{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \dots \\ \varepsilon_m \end{pmatrix}, \vec{b} = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{pmatrix}.$$

Then we can rewrite system (4) as follows:

$$\mathbf{A}^{\vec{x}} \odot \vec{\varepsilon} = \vec{b}, \tag{6}$$

where $\mathbf{A}^{\vec{x}}$ is the right-sided MPF defined by (2) and $\odot$ is the Hadamard product of two vectors.

We can draw inspiration from a couple of observations in the PKE Scheme 2 to adapt the original idea to another multiplicative group. First, we can see that we used only a certain subgroup of a larger multiplicative group $\mathbb{Z}_p^*$ to define the PKE Scheme 2. This was done to ensure that every element of the Sylow subgroup $\mathbb{G}_q$ aside from 1 has a multiplicative order of q. Therefore, the private key exponents are chosen from $\mathbb{Z}_q$. Also note that the multiplicative algebraic structure to define the PKE Scheme 2 (in our case, $\mathbb{G}_q$) must be a group, since otherwise the multiplicative inverse may not exist and hence the decryption step fails. Furthermore, the decision on the decrypted bit is more complicated as compared to the PKE Scheme 1 since it is harder to consider the leftover term g^r. This issue may be resolved by making a list of values for which the decrypted bit is 0, and for which the decrypted bit is 1. Weather this is a reasonable solution depends on the value of q. Another simple solution is to calculate the discrete logarithm of g^r base g, i.e. $r = \operatorname{dlog}_g g^r$. In this case, the decryption works just as in the Regev's PKE.

3 Description of the multiplicative group

In our work, we mainly consider multiplicative groups. Recently our attention turned to the study of non-commuting groups in cryptography, since problems defined in such structures are generally hard to solve and have the potential to withstand quantum analysis.

In this paper, we focus on one specific family of modular-maximal cyclic groups of order 2^t generally denoted as $\mathbb{M}_{2^t}$ [37, 38, 39], where $t \geq 3$ is the size-defining parameter. The general representation of these groups is presented below:

$$\mathbb{M}_{2^t} = \langle a, b | a^{2^{t-1}} = e, b^2 = e, ab = ba^{2^{t-2}+1} \rangle, \tag{7}$$

where a and b are the generators and e is the identity of the group. We can see from the definition of $\mathbb{M}_{2^t}$ that the generators a and b do not commute if $t \geq 3$. Furthermore, due to the third relation, any element of the form $a^{k_1}b^{\alpha}$ can be represented in the form $b^{\alpha}a^{k_2}$ (here $\alpha \in \{0, 1\}$ and $k_{1,2} \in \{0, 1, \dots, 2^{t-1} - 1\}$). Throughout this paper we use the form $b^{\alpha}a^{k}$ to represent the elements of $\mathbb{M}_{2^t}$.

Originally the group $\mathbb{M}_{16}$ caught our attention since it is one of the seven non-commuting groups of size 16 which cannot be decomposed into a direct product of smaller ones [37]. Furthermore, similar results have been proven for groups $\mathbb{M}_{32}$ and $\mathbb{M}_{64}$ [38, 39]. We think that this feature beneficially distinguishes these groups from commuting ones, since most of the abelian groups can be split into either a direct or a free product of smaller groups. Therefore, for commuting multiplicative groups their generators can be considered separately, which results in more LWE equations for the attacker to study.

As we have seen from the presented direct multiplicative approach given by PKE Scheme 2, to construct an LWE problem using a multiplicative group we need the multiplication and exponentiation operations. Also we should be able to calculate the multiplicative inverse of an element. Let us now present explicit expressions of these operations.

First, we consider the multiplication of two elements $\omega_1 = b^{\alpha_1}a^{k_1}$ and $\omega_2 = b^{\alpha_2}a^{k_2}$ in $\mathbb{M}_{2^t}$. We have:

$$\omega_1 \cdot \omega_2 = \begin{cases} b^{\alpha_1+\alpha_2}a^{k_1+k_2}, \text{ if } k_1 \text{ is even;} \\ b^{\alpha_1}a^{k_1+k_2}, \text{ if } k_1 \text{ is odd and } \alpha_2 = 0; \\ b^{\alpha_1+1}a^{k_1+k_2+2^{t-2}}, \text{ if } k_1 \text{ is odd and } \alpha_2 = 1. \end{cases} \tag{8}$$

The expression for raising the element $\omega = b^{\alpha}a^{k}$ to a power n can be derived from (8) and is as follows:

$$\omega^n = \begin{cases} a^{kn}, \text{ if } \alpha = 0; \\ b^n a^{kn}, \text{ if } \alpha = 1 \text{ and } k \text{ is even;} \\ b^n a^{kn+2^{t-2}\left[\frac{n}{2}\right]}, \text{ if } \alpha = 1 \text{ and } k \text{ is odd.} \end{cases} \tag{9}$$

Immediately we see from (9) that the maximal multiplicative order of the elements in $\mathbb{M}_{2^t}$ is equal to 2^{t-1}. Hence any power n in (9) can be considered modulo 2^{t-1}. This is one of the key facts we use in this paper.

Finally, the inverse of the element $\omega = b^{\alpha}a^{k}$ can be calculated as:

$$\omega^{-1} = \begin{cases} a^{-k}, \text{ if } \alpha = 0; \\ ba^{-k}, \text{ if } \alpha = 1 \text{ and } k \text{ is even;} \\ ba^{2^{t-2}-k}, \text{ if } \alpha = 1 \text{ and } k \text{ is odd.} \end{cases} \tag{10}$$

This expression can also be derived from (9) by calculating $\omega^{2^{t-1}-1}$.

All of the presented formulas can be easily proven using the definition of $\mathbb{M}_{2^t}$. For the case of $t = 4$ these proofs can be found in [32].

In our paper, we also make use of the following important observation: an extra summand of 2^{t-2} appears only in the third case of expressions (8) and (9). We use this fact to help us control the appearance of extra factors $a^{2^{t-2}}$ when defining the LWE scheme in $\mathbb{M}_{2^t}$. More precisely, setting $f(n) = 2^{t-2}\left[\frac{n}{2}\right] \bmod 2^{t-1}$ we can see from Table 1 when the extra summand appears in (9).

Table 1: Values of function $f(n)$ [32].

n	0	1	2	3	4	5	6	7	...	$2^{t-1}-4$	$2^{t-1}-3$	$2^{t-1}-2$	
$f(n)$	0	0	2^{t-2}	2^{t-2}	0	0	2^{t-2}	2^{t-2}	...	0	0	2^{t-2}	

Despite the word 'cyclic' in its name, the $\mathbb{M}_{2^t}$ does not contain any single element, which generates the whole group. However, the are two cycles of order 2^{t-1}. These are presented below:

$$\langle a \rangle = \{e, a, a^2, \ldots, a^{2^{t-1}-1}\};$$
$$\langle ba \rangle = \{e, ba, a^2, \ldots, ba^{2^{t-1}-1}\}.$$

It is easy to see that the presented cycles intersect at even powers of generator a, which define the center of $\mathbb{M}_{2^t}$. Also, the presented cycles contain only 75% of total 2^t elements of the considered group.

Notably there are other cycles of smaller orders in $\mathbb{M}_{2^t}$. These are less interesting to us, since we desire to make use of the non-commutativity of $\mathbb{M}_{2^t}$.

Before we end this section, let us present the following proposition which is used in this paper to construct a working scheme:

Proposition 1. *For any two elements $\omega_1, \omega_2 \in \mathbb{M}_{2^t}$ and $n \in \mathbb{Z}_{2^{t-1}}$ we have:*

$$(\omega_1\omega_2)^n = a^{\alpha 2^{t-2}}\omega_1^n\omega_2^n, \tag{11}$$

where $\alpha \in \{0, 1\}$.

Proof. This fact follows from the definitions (8) and (9) of basic operations in group $\mathbb{M}_{2^t}$. □

Note that in (11) we have $\alpha = 0$, if both ω_1 and ω_2 come from the same cycle. This is one of the facts, which helps us to control additional factors during the decryption step.

4 PKE from the LWE in the non-commuting multiplicative group

Let us now present the main idea of this paper, i.e. we show how LWE can be adapted to work in a non-commuting group $\mathbb{M}_{2^t}$. Inspired by the original idea of Regev we propose to divide the cycle $\langle ba \rangle$ into two equal parts. However, we also need several other changes to our proposal to achieve the benefits from a non-commuting group while working properly.

PKE Scheme 3. *We assume that the number of equations m and the number of unknowns n are fixed. For $t \geq 3$ such that 2^{t-1} is between n^2 and $2n^2$ we consider the non-commuting group $\mathbb{M}_{2^t}$, and a discrete Gaussian distribution on $\mathbb{Z}_{2^{t-1}}$ denoted as $\mathcal{N}(0, \sigma)$, where $\sigma \leq 2^{\frac{t-1}{4}}$. As in PKE Scheme 1, let $m = (t-1)(1+\epsilon)(n+1)\log 2$ for some constant $\epsilon > 0$. Then we can define the following one-bit PKE scheme from LWE in $\mathbb{M}_{2^t}$ as follows:*

- KeyGen *algorithm:*
  - *Alice generates her private key as a vector $\vec{x}$, where $x_i \in \mathbb{Z}_{2^{t-1}}$.*
  - *She picks an integer n_c close to $\frac{n}{2}$. Then Alice uniformly and independently chooses entries of the base matrix $\mathbf{W}$ from $\mathbb{M}_{2^t}$, so that the first n_c columns of $\mathbf{W}$ contain elements from the cycle $\langle ba \rangle$, and the leftover columns contain elements from $\langle a \rangle$. Therefore, entries in all the columns of matrix $\mathbf{W}$ commute whereas row entries are divided into two commuting parts.*
  - *Alice now calculates the vector $\vec{u} = \mathbf{W}^{\vec{x}}$ and samples the errors ε_i from the discrete Gaussian distribution $\mathcal{N}(0, \sigma)$. Alice publishes the pair $(\mathbf{W}, \vec{v})$ as her public key, where $\vec{v} = \vec{\varepsilon} \odot \vec{u}$, and $\odot$ is the Hadamard (element-wise) product of two vectors.*
- Enc *algorithm:*
  - *Bob chooses a random subset $\mathbb{S} = \{s(1), s(2), \ldots, s(r)\}$ of $[m]$. Also, for each of the chosen equations he sample error terms $\delta_{s(i)} \in \{e, ba, (ba)^{-1}\}$ as follows:*
    - *For the first of the chosen equations he chooses an error $\delta_{s(1)}$ so that the product $v'_{s(1)} = \delta_{s(1)} v_{s(1)}$ is of the form ba^k for some k, i.e. the generator b is present;*
    - *For all of the subsequent chosen equations Bob chooses errors $\delta_{s(i)}$ so that the product $v'_{s(i)} = \delta_{s(i)} v_{s(i)} \in \langle a \rangle$, i.e. the generator b is absent;*
    - *The error terms ba and $(ba)^{-1}$ alternate starting with the ba term.*
  - *He calculates $\vec{w}$, where $w_j = \prod_{i \in \mathbb{S}} a_{ij}$, and $v' = \prod_{i=1}^{r} v'_{s(i)}$.*
  - *If the original bit is 0 Bob outputs $(\vec{w}, v')$, otherwise the encryption is $(\vec{w}, a^{2^{t-2}} v')$.*
- Dec *algorithm:*
  - *Given a pair $(\vec{w}, v')$, Alice uses her private key $\vec{x}$ to calculate $h = v' \cdot (\vec{w}^{\vec{x}})^{-1}$.*
  - *Since $h \in \langle ba \rangle$, Alice can calculate the discrete logarithm of h base ba. The decrypted bit is 0 if $\operatorname{dlog}_{ba} h$ is closer to 0 than to 2^{t-2} and 1 otherwise.*

Evidently, the non-commutativity of $\mathbb{M}_{2^t}$ cannot be ignored in this case. Moreover, the two columns can only be switched if they commute. Otherwise, it becomes hard to control the extra factors.

Another major difference is the fact that Bob does not use the vector $\vec{v}$ directly to produce the ciphertext of a bit. In other words, the PKE Scheme 3 does not work correctly if the errors are absent unless the entries of $\vec{u}$ are contained in $\langle a \rangle$ in the first place. Simply put, the error term in the LWE problem produced by PKE Scheme 3 always exists.

Now we prove the correctness of our proposal, i.e.

Proposition 2. *The PKE Scheme 3 decrypts the encrypted bits correctly with overwhelming probability.*

Proof. Recall from the definition of the encryption algorithm Enc that Bob modifies the chosen equations in such a way that only the first element $v'_{s(1)}$ contains the generator b whereas all the subsequent elements $v'_{s(i)}$, where $i > 1$, do not contain the generator b. Without loss of generality, we assume that Bob considers the first two equations i.e. $\mathbb{S} = \{1, 2\}$. Furthermore, we divide the left hand side products into two parts just as Alice has done. Hence we have:

$$\begin{array}{cccccc} \delta_1 & \varepsilon_1 & \prod_{i=1}^{n_c} w_{1i}^{x_i} & \prod_{j=n_c+1}^{n} w_{1j}^{x_j} & = & ba^*; \\ \delta_2 & \varepsilon_2 & \prod_{i=1}^{n_c} w_{2i}^{x_i} & \prod_{j=n_c+1}^{n} w_{2j}^{x_j} & = & a^*, \end{array} \tag{12}$$

where $*$ indicates any power of a. Since the right hand side of the second equation is a^*, the product $\delta_2\varepsilon_2 \prod_{i=1}^{n_c} w_{2i}^{x_i}$ has the form a^{2*}, i.e. any even power of a, and therefore is in the center of $\mathbb{M}_{2^t}$. For this reason and the associativity property, we have:

$$\begin{aligned} &\Big(\delta_1\varepsilon_1 \prod_{i=1}^{n_c} w_{1i}^{x_i}\Big)\Big(\prod_{j=n_c+1}^{n} w_{1j}^{x_j}\Big)\Big(\delta_2\varepsilon_2 \prod_{i=1}^{n_c} w_{2i}^{x_i}\Big)\Big(\prod_{j=n_c+1}^{n} w_{2j}^{x_j}\Big) = \\ &= \Big(\delta_1\varepsilon_1 \prod_{i=1}^{n_c} w_{1i}^{x_i}\Big)\Big(\delta_2\varepsilon_2 \prod_{i=1}^{n_c} w_{2i}^{x_i}\Big)\Big(\prod_{j=n_c+1}^{n} w_{1j}^{x_j}\Big)\Big(\prod_{j=n_c+1}^{n} w_{2j}^{x_j}\Big) = \\ &= (\delta_1\delta_2)(\varepsilon_1\varepsilon_2)\Big(\prod_{i=1}^{n_c}(w_{1i}\cdot w_{2i})^{x_i}\Big)\Big(\prod_{j=n_c+1}^{n}(w_{1j}\cdot w_{2j})^{x_j}\Big) = ba^*\cdot a^*. \end{aligned}$$

We can now apply the same idea to a larger number of equations, since for all of them, aside from the first one, the product $\delta_k\varepsilon_k \prod_{i=1}^{n_c} w_{ki}^{x_i}$ is in the center of $\mathbb{M}_{2^t}$.

By multiplying r equations randomly selected by Bob, we obtain v':

$$v' = \delta\varepsilon\vec{w}^{\vec{x}}.$$

During the decryption step Alice calculates $h = v' \cdot (\vec{w}^{\vec{x}})^{-1} = \delta\varepsilon$, where $\delta = \prod_{i\in\mathbb{S}} \delta_i$ and $\varepsilon = \prod_{i\in\mathbb{S}} \varepsilon_i$. Notably, the extra error term δ can only be one of two options, i.e., $\delta = e$ if an even number of non-identity errors $\delta_{s(i)}$ was generated, and $\delta = ba$ otherwise.

Let us now consider the case of $\delta = e$. Then the value $h = \varepsilon$ is the product of r independent discrete Gaussian random variables of the form ε_k implies that the exponent of the term $(ba)^\beta$, where $\beta = \sum_k^r \beta_k$ is itself a discrete Gaussian random variable, whose distribution can be readily written down as $\beta \sim \mathcal{N}(0, \sigma\sqrt{r})$.

In lattice-based cryptography, the parameter σ is often chosen significantly smaller than the modulus $\rho = 2^{t-1}$. The ratio $\sigma \ll \frac{p}{8}$ is common to prevent wrap-around modulo ρ and to maintain hardness assumptions [41, 42]. Therefore, below in Table 2 we present the probability of failure results for selected values of the discrete Gaussian distribution defined in $\mathbb{Z}_\rho$ parameter $\sigma = \rho^{1/4}$ and r equations.

We can see from Table 2 that the probability of error increases with the number of chosen equations r. However, we can see that the failure probability remains acceptably

Table 2: Probability of failure for different values of r, $\rho = 2^{t-1}$, $\sigma = \rho^{1/4}$

r	$\rho = 2^{10}$	$\rho = 2^{11}$	$\rho = 2^{12}$
16	4.8e-28	3.78e-78	1.13e-220
20	1.7e-22	1.81e-62	2.53e-176
25	5.0e-18	6.44e-50	7.72e-141
30	4.7e-15	1.53e-41	3.54e-117
50	4.4e-09	9.0e-25	7.88e-70
100	1e-4	3.73e-12	2.83e-34
200	0.03	7.85e-06	1.8e-16

low given that proper parameter values are chosen. As ρ increases, a sharp decrease in the error probability is observed, for fixed r. Based on the performed calculations, we can estimate the dependency of the failure probability on ρ by an exponential decay in ρ:

$$\Pr[\text{fail}] = A(r) \cdot e^{-B(r)\cdot\rho}$$

where $A(r)$ and $B(r)$ are parameters depending on r. Due to this exponential decay we claim that with an appropriately chosen value of σ, the probability of incorrect decryption becomes negligibly small.

The proof for the case of $\delta = ba$ is analogous with the exception that $h = (ba)\varepsilon$ is now a discrete Gaussian variable shifted by 1. Therefore, despite the distribution of δ leaning to the latter case, the probability of incorrect decryption remains negligibly small. □

5 Example calculations

5.1 Toy Example

Let us present an example of our idea. Assume that Alice and Bob settled on a platform group $\mathbb{M}_{2^9}$. Therefore, all the calculations are performed using modulo $2^8 = 256$.

Alice runs the KeyGen algorithm. The output is as follows:

- The private key: $\vec{x} = \big(228, \quad 245, \quad 140, \quad 35, \quad 38, \quad 65, \quad 215, \quad 65\big)$.
- The following 8×8 base matrix:

$$\mathbf{W} = \begin{pmatrix} a^{208} & ba^{245} & ba^{235} & ba^{173} & a^{70} & a^{112} & a^{181} & a^{245} \\ ba^{103} & ba^{119} & a^{106} & ba^{193} & a^{11} & a^{97} & a^{193} & a^{87} \\ a^{32} & a^{40} & a^{74} & a^{62} & a^{24} & a^{195} & a^{70} & a^{149} \\ ba^{233} & a^{248} & ba^{245} & a^{100} & a^{210} & a^{203} & a^{174} & a^{57} \\ ba^{161} & ba^{245} & ba^{39} & ba^{39} & a^{177} & a^{47} & a^{167} & a^{192} \\ a^{24} & a^{124} & ba^{9} & ba^{171} & a^{81} & a^{125} & a^{41} & a^{65} \\ ba^{199} & a^{204} & ba^{217} & a^{180} & a^{243} & a^{114} & a^{30} & a^{129} \\ a^{140} & a^{36} & ba^{111} & a^{8} & a^{8} & a^{165} & a^{127} & a^{178} \end{pmatrix}.$$

  We can see that $n_c = 4$ in this example.

- The public key: $\vec{v} = \mathbf{W}^{\vec{x}} = \big(ba^{237}, \quad a^{237}, \quad ba^{235}, \quad ba^{89}, \quad a^{62}, \quad ba^{14}, \quad a^{49}, \quad a^{102}\big)$.

The error vector generated during KeyGen execution was

$$\vec{\varepsilon} = \big(ba^{253}, \quad a^{130}, \quad ba^{127}, \quad ba^{131}, \quad a^{130}, \quad a^{130}, \quad a^{122}, \quad a^{134}\big).$$

This vector remains hidden. Note that all the vectors are presented in transposed form for convenience.

Bob wants to encrypt the bit 0 and runs the Enc algorithm. The following data is generated during the execution:

- The set of chosen equations $\mathbb{S} = \{4, 7, 2, 8, 1, 6\}$.
- The vector $\vec{w} = \left(ba^{11},\ a^{80},\ ba^{155},\ ba^{185},\ a^{111},\ a^{48},\ a^{234},\ a^{249}\right)$.
- Extra errors for equations were generated as presented in Table 3:

Table 3: Extra errors generated for the chosen equations

$v_4 = ba^{89}$	$v_7 = a^{49}$	$v_2 = a^{237}$	$v_8 = a^{102}$	$v_1 = ba^{237}$	$v_6 = ba^{14}$
$\delta_4 = e$	$\delta_7 = e$	$\delta_2 = e$	$\delta_8 = e$	$\delta_1 = ba$	$\delta_6 = ba^{127}$

- The value $v' = ba^{89} \cdot a^{49} \cdot a^{237} \cdot a^{102} \cdot a^{110} \cdot a^{13} = ba^{88}$.
- The output is $(\vec{w}, ba^{88})$.

Upon receiving the pair $(\vec{w}, ba^{88})$, Alice runs the Dec algorithm:

- She calculates $h = ba^{88} \cdot (\vec{w}^{\vec{x}})^{-1} = a^4$.
- Alice calculates the discrete logarithm $\mathrm{dlog}_{ba}\, a^4 = 4$.
- Since $4 < 64$, i.e., it is closer to 0, Alice outputs 0 as the decrypted bit.

We can see that Alice correctly decrypted the bit.

If Bob wishes to encrypt 1, he sends $(\vec{w}, ba^{216})$ to Alice. She calculates $h = ba^{216} \cdot (\vec{w}^{\vec{x}})^{-1} = a^{132}$. The discrete logarithm $\mathrm{dlog}_{ba}\, a^{132} = 132$ is now in the range $64 < 132 < 192$ and hence Alice outputs 1, which is correct.

5.2 Correctness of decryption

Now that we have shown how our idea works, let us perform experiments using parameters chosen according to the setup of our protocol. We consider n variables and $m = (t-1)(1+\epsilon)(n+1)\log 2$ equations defined in $\mathbb{M}_{2^t}$, where $n^2 \leq 2^{t-1} \leq 2n^2$. We perform[1] 1000 executions of PKE Scheme 3 each time randomly generating the base matrix, where $n_c = \lceil \frac{n}{2} \rceil$. For each experiment a the KeyGen algorithm generates a new pair of keys for Alice using discrete Gaussian errors with $\sigma = 2^{\frac{t-1}{4}}$. Bob encrypts a randomly chosen bit using the Enc algorithm, where the number of selected equations is in the range $\left[\frac{m}{4}; \frac{3m}{4}\right]$. Alice decrypts the obtained ciphertext using Dec algorithm and her private key. The decrypted bit is compered to the original one. We keep count of the correctly decrypted bits therefore indicating the correct execution of our scheme.

First, we consider the following parameters: $n = 15$ variables and $m = 90$ equations defined in $\mathbb{M}_{2^9}$. Note that $(t-1)(1+\epsilon)(n+1)\log 2 = 90$ for the value $\epsilon \approx 0.0144$. Also, we repeat this experiment 10 times. The obtained results are presented in Table 4.

We can see that in our setup, the probability of correct decryption is 97.8% on average. This result is stable with a standard deviation of roughly 0.46. Moreover, after the number of executions was increased to 10000, we obtained a 98.03% probability of successful decryption.

Now we can consider a more realistic setup: $n = 80$ variables and $m = 730$ equations defined in $\mathbb{M}_{2^{14}}$. Under this setup, all of 1000 executions resulted in correctly decrypted bits.

[1]The experiments were performed in MATLAB R2020b. The program is available at https://drive.google.com/drive/folders/1cg-UsS8OoFVcgzwc22nWdzGBqA7v5O6X?usp=sharing

Table 4: Ratio of correctly decrypted bits

Experiment	Ratio, %
1	97.2
2	97.4
3	97.2
4	97.8
5	97.8
6	98.5
7	98.0
8	98.5
9	97.8
10	97.8
Average	97.8

6 Security analysis

We now consider the security of the presented scheme.

Inspired by LWE hardness assumption (see Section 2.6 in [43]) we define two oracles $\mathcal{O}_{enc}$ and $\mathcal{O}_{rand}$ whose behaviors are presented below:

- $\mathcal{O}_{enc}$ outputs samples $(\vec{w}_i, \varepsilon_i \cdot \vec{w}_i^{\vec{x}})$, where the entries of vectors $\vec{w}_i$ are uniformly sampled from one of the cycles as presented in PKE Scheme 3 with a fixed value of n_c, i.e. for any query index i and for a fixed column index $j \leq n_c$ we have $w_{ij} \in \langle ba \rangle$, otherwise $w_{ij} \in \langle a \rangle$. As previously, the vector $\vec{x}$ is uniformly chosen from $\mathbb{Z}_{2^t}^n$. The error ε_i is an element of $\langle ba \rangle$.
- $\mathcal{O}_{rand}$ generates random samples of the form $\vec{w}_i, v_i$, where $\vec{w}_i$ is generated as above and $v_i \in \mathbb{M}_{2^t}$.

Defining the advantage of $\mathcal{A}$ to decide the LWE problem $LWE1(\mathbb{M}_{2^t}, n)$ presented above as

$$AdvLWE1(\mathbb{M}_{2^t}, n) = |\Pr[\mathcal{A}^{\mathcal{O}_{enc}} = 1] - \Pr[\mathcal{A}^{\mathcal{O}_{rand}} = 1]| \tag{13}$$

we show that the following proposition holds:

Proposition 3. *The PKE Scheme 3 oracle $\mathcal{O}_{enc}$ produces samples statistically indistinguishable from the ones produced by $\mathcal{O}_{rand}$ if $\lim_{n\to\infty} \frac{n_c}{n-n_c} \to 1$.*

Proof. Let us consider a sample $(\vec{w}, \varepsilon \cdot \vec{w}^{\vec{x}})$ produced by the oracle $\mathcal{O}_{enc}$. Due to the properties of $\mathbb{M}_{2^t}$, the product $\varepsilon \cdot \vec{w}^{\vec{x}}$ can be rewritten as follows:

$$\varepsilon \cdot \vec{w}^{\vec{x}} = (ba)^{r+\sum_{j=1}^{n_c} \alpha_j x_j} a^{\sum_{j=n_c+1}^{n} \alpha_j x_j}, \tag{14}$$

where $\varepsilon = (ba)^r$, $w_1 = (ba)^{\alpha_1}, \ldots, w_{n_c} = (ba)^{\alpha_{n_c}}, w_{n_c+1} = a^{\alpha_{n_c+1}}, w_n = a^{\alpha_n}$.

Now we consider the distribution of exponents of each part individually.

In the first part, the exponent of ba is the sum $r + \sum_{j=1}^{n_c} \alpha_j x_j$, which is reduced modulo 2^{t-1}, where the power r is a discrete Gaussian variable. Hence we are interested in the distribution of the sum $\sum_{j=1}^{n_c} \alpha_j x_j$, where the coefficients α_j's are fixed.

Consider a single term $\alpha_1 x_1$, and assume that the fixed coefficient α_1 is odd. Then it is invertible modulo 2^{t-1}, and hence the probability

$$\Pr[\alpha_1 x_1 = s_0] = \Pr[x_1 = (\alpha_1)^{-1} s_0],$$

where s_0 is a fixed value. However, x_1 is a uniform random variable, and therefore

$$\Pr[\alpha_1 x_1 = s_0] = \frac{1}{2^{t-1}}. \tag{15}$$

Now assume that $\alpha_1 = 2^k \beta_1$ and β_1 is odd. Then the congruence $\alpha_1 x_1 \equiv s \bmod 2^{t-1}$ has a solution if and only if $s = 2^k u$. In this case, it is equivalent to the congruence $\beta_1 x_1 \equiv u \bmod 2^{t-1-k}$, since we can cancel 2^k from both sides of the initial congruence including the modulus. Hence, we have:

$$\Pr[\alpha_1 x_1 = s_0] = \Pr[\beta_1 x_1 = u_0] = \begin{cases} \frac{1}{2^{t-1-k}}, \text{ if } \gcd(\alpha_1, s_0) = 2^k; \\ 0, \text{ otherwise.} \end{cases} \tag{16}$$

Therefore, instead of one-to-one mapping, we get a 2^k-to-one mapping. However, each of the achievable values remains equally likely.

Now consider the sum of two terms denoted as $S_2 = \alpha_1 x_1 + \alpha_2 x_2$. Then, if α_1 is odd, the term $\alpha_1 x_1$ is uniformly distributed, and therefore due to (15) we have:

$$\Pr[S_2 = S_2^0] = \sum_{x_2^0=0}^{2^{t-1}} \Pr[\alpha_1 x_1 = S_2^0 - \alpha_2 x_2^0, x_2 = x_2^0] = \frac{1}{2^{t-1}} \sum_{x_2^0=0}^{2^{t-1}} \Pr[x_2 = x_2^0] = \frac{1}{2^{t-1}}, \tag{17}$$

because $\alpha_1 x_2$ and x_2 are independent random variables. Through similar actions we can also show that S_2 is uniformly distributed if α_2 is odd. Moreover, if both $\alpha_1 = 2^{k_1}\beta_1$ and $\alpha_2 = 2^{k_2}\beta_2$ are even, we can perform similar actions after eliminating all of the common evenness $2^{\min(k_1,k_2)}$ thus obtaining a distribution as in (16).

The main takeaway from these observations is that the uniform distribution of the sum result value S_2 is achieved if and only if at least one of the fixed coefficients α_1 or α_2 is odd.

Now we perform an induction on the number of terms in the sum $S_{n_c} = \sum_{j=1}^{n_c} \alpha_j x_j$. Given that for $m < n_c$ the sum $S_{m-1} = \sum_{j=1}^{m-1} \alpha_j x_j$ is uniformly distributed, we have:

$$\Pr[S_{m-1} + \alpha_m x_m = S_m^0] = \sum_{x_m^0=0}^{2^{t-1}} \Pr[S_{m-1} = S_m^0 - \alpha_m x_m^0, x_m = x_m^0] =$$

$$= \frac{1}{2^{t-1}} \sum_{x_m^0=0}^{2^{t-1}} \Pr[x_m = x_m^0] = \frac{1}{2^{t-1}},$$

since the sum $\sum_{x_m^0=0}^{2^{t-1}} \Pr[x_m = x_m^0]$ covers all the possible values of x_m. The discrete Gaussian distribution of r is irrelevant at this point as the overall sum $r + \sum_{j=1}^{n_c} \alpha_j x_j$ remains uniformly distributed.

In the second part, the exponent of a is the sum $\sum_{j=n_c+1}^{n} \alpha_j x_j$, which is also reduced modulo 2^{t-1}. However, due to similar observations, this sum is also uniformly distributed in $\mathbb{Z}_{2^{t-1}}$, unless all of the fixed α_j's are simultaneously even.

For the entire product to be uniformly distributed in $\mathbb{M}_{2^t}$ both exponents in the considered parts have to be uniformly distributed in $\mathbb{Z}_{2^{t-1}}$. This can be shown using a multiplication table for the value of the product $\omega_1 \cdot \omega_2$, where $\omega_1 \in \langle ba \rangle$ and $\omega_2 \in \langle a \rangle$. An example of this table for the group $\mathbb{M}_{16}$ is presented in Table 5.

Therefore, the oracle $\mathcal{O}_{enc}$ produces a uniformly distributed random variable $\varepsilon \cdot \vec{w}^{\vec{x}}$ with probability $1 - (2^{-n_c} + 2^{n_c - n} - 2^{-n})$, since all α_j's are sampled independently. However, the quantity $(2^{-n_c} + 2^{n_c-n} - 2^{-n})$ has a dominant term $2^{-min(n_c, n-n_c)}$, which is negligibly small, given that the value of n_c is properly chosen. A good choice of this parameter is $n_c = \lceil \frac{n}{2} \rceil$.

Therefore, the samples generated by $\mathcal{O}_{enc}$ are statistically indistinguishable from those produced by $\mathcal{O}_{rand}$. □

Table 5: Multiplication table for $\langle ba \rangle \times \langle a \rangle$

$\cdot$	e	a	a^2	a^3	a^4	a^5	a^6	a^7
e	e	a	a^2	a^3	a^4	a^5	a^6	a^7
ba	ba	ba^2	ba^3	ba^4	ba^5	ba^6	ba^7	b
a^2	a^2	a^3	a^4	a^5	a^6	a^7	e	a
ba^3	ba^3	ba^4	ba^5	ba^6	ba^7	b	ba	ba^2
a^4	a^4	a^5	a^6	a^7	e	a	a^2	a^3
ba^5	ba^5	ba^6	ba^7	b	ba	ba^2	ba^3	ba^4
a^6	a^6	a^7	e	a	a^2	a^3	a^4	a^5
ba^7	ba^7	b	ba	ba^2	ba^3	ba^4	ba^5	ba^6

Note that due to Bob's errors the element v' is of the form ba^k for some k. Therefore, this element has a special structure which is an important fact to keep in mind for the security analysis. In other words, we can define the following Attack Game between the adversary $\mathcal{A}$ and the challenger $\mathcal{C}$:

1. $\mathcal{C}$ creates an LWE instance as presented in this paper and sends the pair $(\mathbf{W}, \vec{v})$ to $\mathcal{A}$;

2. $\mathcal{A}$ samples a random bit $\mu \in \{0, 1\}$, and sends it to $\mathcal{C}$;

3. $\mathcal{C}$ samples a random bit $\beta \in \{0, 1\}$;

4. If $\beta = 0$, then $\mathcal{C}$ encrypts the bit μ by executing Bob's steps. Hence $\mathcal{C}$ obtains a ciphertext $\vec{c}_0 = (\vec{w}, a^{\beta \cdot 2^{t-2}} v')$;

5. If $\beta = 1$, then $\mathcal{C}$ calculates $\vec{w}$ as Bob does, samples a random k uniformly from the set $\left[0, 2^{t-1}\right]$ and outputs $\vec{c}_1 = (\vec{w}, ba^k)$;

6. $\mathcal{C}$ sends $\vec{c}_\beta$ to $\mathcal{A}$;

7. $\mathcal{A}$ outputs a guess $\hat{\beta}$. He wins if $\hat{\beta} = \beta$.

Defining the advantage of $\mathcal{A}$ win the Attack Game presented above as

$$\text{IND-CPA}(LWE(\mathbb{M}_{2^t}, n)) = |\Pr[\hat{\beta} = \beta] - \frac{1}{2}| \tag{18}$$

we show that the following proposition holds:

Proposition 4. *$\mathcal{A}$'s advantage in winning the Attack Game presented above is negligible, i.e., IND-CPA$(LWE(\mathbb{M}_{2^t}, n)) \leq \epsilon$.*

Proof. Since the entries of $\vec{v}$ are distributed uniformly, Bob's errors only place these entries in an appropriate set as presented in PKE Scheme 3. Therefore, powers of the generator a are uniformly distributed in the set $\left[0, 2^{t-1}\right]$. Furthermore, the set $\mathbb{S}$ is chosen at random as well. Since only the first entry of v' contains the generator b, it cannot be eliminated and no additional factors appear during encryption. Hence, we can consider the distribution of the power of a in $v' = \prod_{i=1}^{r} v'_{s(i)}$. However, since $\mathbb{Z}_{2^{t-1}}$ is an additive group, the uniform distribution is preserved for the sum of powers of a. On the other hand, the element ba^k from $\vec{c}_1$ was generated using a uniformly distributed k. Therefore, the adversary cannot distinguish between $a^{\beta \cdot 2^{t-2}} v'$ and ba^k. □

Therefore, we claim that our scheme is IND-CPA secure.

As for the recovery of the secret key, we have seen that the non-commutativity of $\mathbb{M}_{2^t}$ cannot be ignored. However, due to definitions (8) and (9) of the basic operations, we can define a subproblem of $LWE1(\mathbb{M}_{2^t}, n)$ by applying the following mapping:

$$\varphi : \mathbb{M}_{2^t} \to \mathbb{Z}_2 : \quad \varphi(b^\alpha a^k) = k \bmod 2.$$

It is easy to see that φ is essentially a parity mapping. By applying it entry-wise to $LWE1(\mathbb{M}_{2^t}, m)$ we have:

- $\mathcal{O}_{enc}$ outputs samples $(\vec{w}_i, \varepsilon_i \cdot \vec{w}_i^{\vec{x}})$, where the entries of vectors $\vec{w}_i$ are uniformly sampled from $\mathbb{Z}_2^m$, the vector $\vec{x}$ is uniformly chosen from $\mathbb{Z}_2^n$, and the error ε_i is an element of $\mathbb{Z}_2$.
- $\mathcal{O}_{rand}$ generates random samples of the form $\vec{w}_i, v_i$, where $\vec{w}_i$ are uniformly sampled from $\mathbb{Z}_2^m$ and $v_i \in \mathbb{Z}_2$ is chosen at random.

We define the advantage of $\mathcal{A}$ to decide the LWE problem $LWE(\mathbb{Z}_2, m)$ presented above as

$$AdvLWE(\mathbb{Z}_2, n) = |\Pr[\mathcal{A}^{\mathcal{O}_{enc}} = 1] - \Pr[\mathcal{A}^{\mathcal{O}_{rand}} = 1]| \tag{19}$$

Evidently, solving $LWE(\mathbb{Z}_2, m)$ would reveal the parity of the secret vector $\vec{x}$. Therefore, we claim that

$$AdvLWE1(\mathbb{M}_{2^t}, n) \leq AdvLWE(\mathbb{Z}_2, n), \tag{20}$$

i.e., the $LWE1(\mathbb{M}_{2^t}, n)$ problem is at least as secure as $LWE(\mathbb{Z}_2, n)$. As of now, we make a conjecture that the inequality is strict due to additional randomness introduced by the non-commutative nature of $\mathbb{M}_{2^t}$, which cannot be taken into the account by the $LWE(\mathbb{Z}_2, n)$ problem. Moreover, since the group $\mathbb{M}_{2^t}$ is not cyclic, we cannot apply the discrete logarithm mapping to the equations directly. Also, the considered group cannot be split into a direct product of smaller groups. Hence the analogue of the discrete logarithm mapping cannot be introduced in this structure. For example, consider the element $ba^2 \in \mathbb{M}_{16}$. We have $ba^2 = (ba) \cdot a = (ba^3) \cdot a^7 = a^3 \cdot (ba^3)$. Therefore, there is no unique representation of the element ba^2 using two subgroups $\langle ba \rangle$ and $\langle a \rangle$ we consider in this paper.

Moreover, we can also replace 2 with a prime q in the definition of modular-maximal cyclic groups, i.e. in the future we may also consider a family of groups $\mathbb{M}_{q^t}$. Therefore, we think that a security level of $LWE(\mathbb{Z}_p, n)$ can be achieved for $LWE1(\mathbb{M}_{q^t}, n)$ problem. Moreover, if the inequality in (20) is strict, the prime q can be less than p.

7 Comparison to the original LWE

Let us now compare the presented schemes to Regev's original idea.

The first major difference is that the PKE Scheme 3 uses two cycles of $\mathbb{M}_{16}$ in such a way that error terms must exist for it to work properly. In other words, the PKE Scheme 1 without errors is easily broken using Gaussian elimination technique whereas in the PKE Scheme 3 equations without errors are worthless, since the scheme is not able to decrypt a bit correctly (at least, not with an overwhelming probability). In other words, extra error terms δ_i are necessary for the scheme to function correctly. Also, since the non-commutativity of $\mathbb{M}_{2^t}$ cannot be ignored, even without errors the complexity of solving a system of equations obtained from the PKE Scheme 3 is not known thus far. We think that this fact beneficially distinguishes our idea from the original one.

Let us now consider the original idea of Regev. Assume that the public matrix $\mathbf{A}$ has a left pseudo-inverse matrix $\mathbf{A}'$, i.e. $\mathbf{A}'A$ is an identity matrix. Then the LWE problem

$\mathbf{A}\vec{x} + \vec{e} = \vec{b}$ can be restated as follows by multiplying the original problem by $\mathbf{A}'$ on the left:

$$\vec{x} + \mathbf{A}'\vec{e} = \mathbf{A}'\vec{b},$$

where now $\vec{x}$ serves as an error vector and $\vec{e}$ is the private key. This is called the duality of LWE problem. In other words, the secret key $\vec{x}$ and the error term $\vec{e}$ are interchangeable in the original idea.

This property is lost in our scheme, since the discrete logarithm cannot be applied to the considered equations. Furthermore, the are two types of errors used in the scheme. As a consequence we may obtain a shifted distribution for the error term h, if the product $\prod_{i=1}^{r} \delta_i = ba$.

From the security point of view, both schemes presented here have at the very least a comparable level of security, considering that the size of the non-commuting group can be increased to any power of a prime q. This comes from the fact that all of the schemes rely on similar systems of equations. Regev's scheme relies on a classic system of linear equations with errors. The direct multiplicative approach is essentially the same after using a discrete logarithm mapping. However, as opposed to the direct approach the discrete logarithm mapping cannot be applied to LWE problem defined in non-commuting group $\mathbb{M}_{2^t}$, since this group itself brings an additional randomness factor into LWE problem.

8 Conclusions

In this paper, we have presented a way for adapting the LWE problem to non-commuting multiplicative groups, which could lead to new directions of research regarding the LWE problem. In the presented scheme we consider a group with two non-commuting generators which cannot be considered separately, since in definitions (8), (9) and (10) the exponent of generator b also affects the exponent of generator a in the output value. We think that this fact is beneficial for our scheme, since the additive group $\mathbb{Z}_q$ used in the original LWE is cyclic generated by 1.

We think that additional factor of non-commutativity can be used as one of the means to complicate the cryptanalysis of our scheme and potentially other schemes which use non-commuting structures to define a LWE problem. The open question may be to find a way to encrypt two bits using Regev's original idea by implementing non-commuting structures with two generators. This idea could be expanded to longer messages just as it was done previously with Regev's PKE scheme thus potentially halving the secret key while maintaining the security level. Interestingly enough, this is not possible with commuting groups, since any such group would be immediately split into either direct or free product of two distinct groups thus splitting a single LWE problem into two disjoint LWE problems.

Notably, the non-commuting group $\mathbb{M}_{2^t}$ contains two cycles of size 2^{t-1}. We used this fact to show that it can be used to construct an accurate criterion for the correct decryption of a bit with overwhelming probability. We have proven that the ciphertext produced by the general concept scheme oracle $\mathcal{O}_{enc}$ is indistinguishable from a truly random output of the oracle $\mathcal{O}_{rand}$. Furthermore, we adapted the LWE hardness assumption for our scheme due to the special structure of the ciphertext generated by our scheme. Using the defined attack game, we have shown that an effective adversary cannot distinguish between two ciphertexts generated by our scheme.

The security of our scheme is at the very least comparable to the security of the original scheme by Regev. For now it is an open question how the non-commutativity factor contributes to the overall security of the scheme. However, we think that the original security can be easily achieved simply by switching the group $\mathbb{M}_{2^t}$ for a larger one, namely $\mathbb{M}_{q^t}$. In this paper, we focused more on establishing a working scheme leaving larger

groups for future exploration. Moreover, we plan to adapt the dual Regev PKE scheme presented in [44] for our setup.

References

References

[1] Shor, P W.: Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th annual symposium on foundations of computer science IEEE, 1994. Santa Fe, NM, USA, 124-134, 20-22 November 1994. doi.org/10.1109/SFCS.1994.365700

[2] Keumars A.-S.: Scientists just built a massive 1,000-qubit quantum chip, but why are they more excited about one 10 times smaller?, 2023-12-11. Available from: https://www.livescience.com/technology/computing/ibm-scientists-built-massive-condor-1000-qubit-quantum-computer-chip-133-qubit-heron-system-two.

[3] Smith-Goodson P Forbes, Atom Computing Announces Record-Breaking 1,225-Qubit Quantum Computer. Available from: https://www.forbes.com/sites/moorinsights/2023/10/24/atom-computing-announces-record-breaking-1225-qubit-quantum-computer/.

[4] Pricing: IBM Quantum Computing. https://www.ibm.com/quantum/access-plans, Date accessed: September 27, 2024.

[5] Computer Security Division, I. T. L. (2017, January 3). Call for Proposals—Post-Quantum Cryptography. https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/Call-for-Proposals, Date accessed: September 27, 2024.

[6] NIST Announces First Four Quantum-Resistant Cryptographic Algorithms. https://www.nist.gov/news-events/news/2022/07/nist-announces-first-four-quantum-resistant-cryptographic-algorithms, Date accessed: September 27, 2024.

[7] Avanzi, R., et al. CRYSTALS-Kyber: Algorithm Specifications And Supporting Documentation (version 3.01). https://pq-crystals.org/kyber/data/kyber-specification-round3-20210131.pdf, Date accessed: September 27, 2024.

[8] Bai, S., et al. CRYSTALS-Dilithium: Algorithm Specifications And Supporting Documentation (version 3.1). https://pq-crystals.org/dilithium/data/dilithium-specification-round3-20210208.pdf, Date accessed: September 27, 2024.

[9] Fouque, P.-A., et al. Falcon: Fast-Fourier Lattice-based Compact Signatures over NTRU. https://falcon-sign.info/falcon.pdf, Date accessed: September 27, 2024.

[10] Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing - STOC 1996. ACM Press, Philadelphia, Pennsylvania, USA, 99–108, May 22 - 24, 1996 https://dl.acm.org/doi/10.1145/237814.237838

[11] Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing, STOC 2005. Association for Computing Machinery, New York, USA, 84–93, May 22-24, 2005. doi.org/10.1145/1060590.1060603

[12] Zuo, B. , Li, J., Zhang, Y., and Shen, J.: Identity-Based Online/Offline Encryption Scheme from LWE. Information (Basel), 15 (9), 539, (2024-09). https://eprint.iacr.org/2004/321

[13] Bandara, H., Herath, Y., Weerasundara, T., and Alawatugoda, J.: On Advances of Lattice-Based Cryptographic Schemes and Their Implementations. Cryptography, 6(4), 56, (2022). doi.org/10.3390/cryptography6040056

[14] Brakerski, Z.; Gentry, C.; Vaikuntanathan, V.: New Constructions of Strongly Unforgeable Signatures Based on the Learning with Errors Problem. In Proceedings of the 48th Annual ACM Symposium on Theory of Computing, Cambridge, MA, USA,19–21 June 2016. 19–21, (2016).

[15] Brakerski, Z., Vaikuntanathan, V.: Efficient Fully Homomorphic Encryption from (Standard) LWE. IEEE 52nd Annual Symposium on Foundations of Computer Science, Palm Springs, CA, USA, 2011. IEEE, 97-106, (2011). doi.org/10.1109/FOCS.2011.12

[16] Xie, X., Xue, R., and Wang, M.: Zero Knowledge Proofs from Ring-LWE. In: Abdalla, M., Nita-Rotaru, C., Dahab, R. (eds) Cryptology and Network Security. CANS 2013. Springer, Cham., 8257, 57–73, (2013). doi.org/10.1007/978-3-319-02937-5_4

[17] Lyubashevsky, V., Peikert, C., and Regev, O.: On Ideal Lattices and Learning with Errors over Rings. In Advances in Cryptology – EUROCRYPT 2010. Springer Berlin Heidelberg, 1–23, (2010). doi.org/10.1007/978-3-642-13190-5_1

[18] Bootle, J., Lyubashevsky, V., Nguyen, N.K., and Seiler, G.: More Efficient Amortization of Exact Zero-Knowledge Proofs for LWE. In: Bertino, E., Shulman, H., Waidner, M. (eds) Computer Security – ESORICS 2021. Springer, Cham., 12973, 608–627, (2021). doi.org/10.1007/978-3-030-88428-4_30

[19] Brakerski, Z., Vaikuntanathan, V.: Fully homomorphic encryption from ring-LWE and security for key dependent messages. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS Springer, Heidelberg, 6841, 505–524, (2011). doi.org/10.1007/978-3-642-22792-9_29

[20] Xie, X., Xue, R., and Wang, M.: Zero Knowledge Proofs from Ring-LWE. In: Abdalla, M., Nita-Rotaru, C., Dahab, R. (eds) Cryptology and Network Security. CANS 2013. Springer, Cham., 8257, 57–73, (2013). doi.org/10.1007/978-3-319-02937-5_4

[21] Lyubashevsky, V., Peikert, C., and Regev, O.: On Ideal Lattices and Learning with Errors over Rings. In: Advances in Cryptology – EUROCRYPT 2010. Springer Berlin Heidelberg, 1–23, (2010). doi.org/10.1007/978-3-642-13190-5_1

[22] Lyubashevsky, V., Peikert, C., and Regev, O.: A Toolkit for Ring-LWE Cryptography. In: Johansson, T., Nguyen, P.Q. (eds) Advances in Cryptology – EUROCRYPT 2013. Springer, Berlin, Heidelberg, 7881, 35–54, (2013). doi.org/10.1007/978-3-642-38348-9_3

[23] Ortiz, J.N., de Araujo, R.R., Aranha, D.F., Costa, S.I.R., and Dahab, R.: The Ring-LWE Problem in Lattice-Based Cryptography: The Case of Twisted Embeddings. Entropy, 23(9), 1108, (2021). doi.org/10.1007/978-3-642-38348-9_3

[24] Albrecht, M.R., and Deo, A.: Large Modulus Ring-LWE $\geq$ Module-LWE. In Advances in Cryptology—ASIACRYPT 2017—23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, 3–7 December 2017. Cham: Springer International Publishing, 10624, 267–296, (2017). doi.org/10.1007/978-3-319-70694-8_10

[25] Peikert, C., and Pepin, Z.: Algebraically Structured LWE, Revisited. Journal of Cryptology, 37, 28, (2024). doi.org/10.1007/s00145-024-09508-3

[26] Bolboceanu, M., Brakerski, Z., Perlman, R., and Sharma, D.: Order-LWE and the Hardness of Ring-LWE with Entropic Secrets. In Advances in Cryptology – ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security. Springer Science and Business Media B.V., (2019). doi.org/10.1007/978-3-030-34621-8_4

[27] Ko, K. H., et al.: New public-key cryptosystem using braid groups. In: Advances in Cryptology—CRYPTO 2000: 20th Annual International Cryptology Conference. Santa Barbara, California, USA, Springer Berlin Heidelberg 166-183, August 20–24, 2000. doi.org/10.1007/3-540-44598-6_10

[28] Anshel, I., Anshel, M., Goldfeld, D.: An algebraic method for public-key cryptography. Mathematical Research Letters, vol. 6: pp. 287—291. https://dx.doi.org/10.4310/MRL.1999.v6.n3.a3

[29] Sakalauskas, E., Listopadskis, N., Tvarijonas, P.: Key agreement protocol (KAP) based on matrix power function. Advanced Studies in Software and Knowledge Engineering, 2(4), pp. 92–96, 2008.

[30] Sakalauskas, E., Luksys, K.: Matrix power function and its application to block cipher s-box construction. International Journal of Innovative Computing, Information and Control, 8(4), pp. 2655–2664, 2012.

[31] Mihalkovich, A., Sakalauskas, E., Levinskas, M.: Key Exchange Protocol Based on the Matrix Power Function Defined Over $\mathbb{M}_{16}$. In: Science and Information Conference. London, UK, Cham: Springer International Publishing, July 14-15, 2022, 511–531.

[32] Mihalkovich, A., Sakalauskas, E., Luksys, K.: Key exchange protocol defined over a non-commuting group based on an NP-complete decisional problem. Symmetry, 12(9): p. 1389, 2020. https://doi.org/10.3390/sym12091389

[33] Mihalkovich A, et al.: CBC Mode of MPF Based Shannon Cipher Defined Over a Non-Commuting Platform Group. Informatica, 33(4), pp. 833–856, 2022. doi.org/10.15388/22-INFOR499

[34] Mihalkovich, A., Luksys, K., Sakalauskas, E.: Sigma identification protocol construction based on MPF defined over non-commuting platform group. Mathematics, 10(15), p. 2649, 2022. https://doi.org/10.3390/math10152649

[35] Sakalauskas, E.: The multivariate quadratic power problem over $\mathbb{Z}_n$ is NP-Complete. Information Technology and Control, 41(1), pp. 33–39, 2012, https://doi.org/10.5755/j01.itc.41.1.821

[36] Sakalauskas, E., Mihalkovich, A.: MPF Problem over Modified Medial Semigroup Is NP-Complete. Symmetry, 10(11), p. 571, 2018. https://doi.org/10.3390/sym10110571

[37] Grundman, H., Smith, T. Automatic realizability of Galois groups of order 16. In: Proceedings of the American Mathematical Society, 124(9), pp. 2631–2640, 1996. https://doi.org/10.1090/S0002-9939-96-03345-X

[38] Grundman, H., Smith, T.: Realizability and automatic realizability of Galois groups of order 32. Central European Journal of Mathematics, vol. 8, pp. 244–260, 2010. doi.org/10.2478/s11533-009-0072-x

[39] Grundman, H., Smith, T. Galois realizability of groups of order 64. Central European Journal of Mathematics, vol. 8, pp. 846–854, 2010. doi.org/10.2478/s11533-010-0052-1

[40] Mihalkovich, A.: On the associativity property of MPF over M16. Proceedings of the Lithuanian Mathematical Society, Ser. A.. Vilnius : Vilniaus universitetas. 2018, vol. 59, pp. 7-12. doi.org/10.15388/LMR.A.2018.02

[41] Micciancio, D., Schultz-Wu, M.: Error Correction and Ciphertext Quantization in Lattice Cryptography. In: Handschuh, H. and Lysyanskaya, A. (eds.) Advances in Cryptology – CRYPTO 2023. pp. 648–681. Springer Nature Switzerland, Cham (2023). https://doi.org/10.1007/978-3-031-38554-4_21.

[42] Lyubashevsky, V., Peikert, C., Regev, O.: A Toolkit for Ring-LWE Cryptography. In: Johansson, T. and Nguyen, P.Q. (eds.) Advances in Cryptology – EUROCRYPT 2013. pp. 35–54. Springer Berlin Heidelberg, Berlin, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38348-9_3.

[43] Agrawal, S., Boneh, D., Boyen, X.: Efficient Lattice (H)IBE in the Standard Model, In: Gilbert H (Ed.), Advances in Cryptology – EUROCRYPT 2010, Berlin, Heidelberg, Springer Berlin Heidelberg, pp. 553–572, 2010. https://doi.org/10.1007/978-3-642-13190-5_28

[44] Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Proceedings of the fortieth annual ACM symposium on Theory of computing. pp. 197–206. ACM, Victoria British Columbia Canada (2008). https://doi.org/10.1145/1374376.1374407