Extending the ElGamal Cryptosystem to the Third Group of Units of $\mathbb{Z}_n$

Jana Hamza *[1], Mohammad EL Hindi †[2], Seifeddine Kadri ‡[3], Therrar Kadri §[4], and Yahya Awad ¶[5]

[1,5] *Department of Mathematics and Physics, Faculty of Arts and Sciences, Lebanese International University, Bekaa, Lebanon*
[2] *Department of Mathematics and Computer Science, Faculty of Science, Beirut Arab University, Beirut, Lebanon*
[3] *Faculty of Applied Computing and Technology, Norof University College, Kristiansand, Norway*
[4] *Department of Science, Nourthwestern Polytechnic, Grande Prairie, Canada*

Abstract

In this paper, we extend the ElGamal cryptosystem to the third group of units of the ring $\mathbb{Z}_n$, which we prove to be more secure than the previous extensions. We describe the arithmetic needed in the new setting. We also provide some numerical simulations that shows the security and efficiency of our proposed cryptosystem.

Keywords: ElGammal Cryptosystem,Group of units, Order of groups, Cyclic groups, Rings

1 Introduction

ElGamal crptosystem was first introduced by T. ElGamal in [7]. Classically the system was defined on the multiplicative group $\mathbb{Z}_p^*$, the group of integers modulo a prime p, which is a cyclic group generated by one of its elements, yet this cryptosystem can work in the setting of any cyclic group G. The intractability of the discrete logarithm problem in the group G is the basis for the security of the generalized ElGamal cryptosystem. Moreover, this group G should be carefully chosen so that the operations of G are relatively easy to apply for efficiency. This cryptosystem has been generalized several times over different

*jana.hamza115@gmail.com
†mohammadyhindi98@gmail.com
‡Seifedine.kadry@norof.no
§TKadri@nwpolytech.ca
¶yehya.awad@liu.edu.lb

groups. For more information about these generalizations, we guide the reader for the following ([9],[10],[11],[13],[14],[16]).

In 2006, El-Kassar and Chehade in [1] introduced a generalization of the group of units of the ring $\mathbb{Z}_n$ denoted by the kth group of units of $\mathbb{Z}_n$, $U^k(\mathbb{Z}_n)$. For more information about the kth group of units, see ([1],[6],[9]). The authors in [1] determined all rings $R = \mathbb{Z}_n$ having the 2^{nd} group of units cyclic. These groups were used as an extension of the ElGamal Cryptosystem given by Haraty et al in [9]. They examined two cases of $U^2(Z_n)$:

1. Both $U(\mathbb{Z}_n)$ and $U^2(\mathbb{Z}_n)$ are cyclic.
2. $U^2(\mathbb{Z}_n)$ is cyclic, while $U(\mathbb{Z}_n)$ is not cyclic.

Kadri and El-Kassar in [9] examined the third group of units of $\mathbb{Z}_n$ and determined all rings $\mathbb{Z}_n$ having $U^3(\mathbb{Z}_n)$ cyclic and proposed to extend the ElGamal cryptosystem to these groups in the case when they are cyclic.

In this paper, we extend the ElGamal cryptosystem to the third group of units of $\mathbb{Z}_n$ in the case when they are cyclic. In other words, we modify the ElGamal public key encryption scheme from its classical domain natural to the domain of $U^3(\mathbb{Z}_n)$ by extending the arithmetic needed for the modifications in this domain.

In Section 2 we describe the construction of $U^3(\mathbb{Z}_n)$ and some theorems related to this group. Section 3 provides the description of our proposed cryptosystem. Finally, Section 4 shows a comparison between our work and some previous results.

2 Preliminaries

In this section we give a brief presentation of the Classical ElGamal public key cryptosystem, and the modified ElGamal cryptosystem in the setting of the second group of units of the ring of integers modulo n.

The basic algorithms for the functioning of this cryptosystem are described in the following three algorithms:

Algorithm 1 *Key generation*

1. Find a generator α of the $\mathbb{Z}_p^*$.
2. Select a random integer a, $1 \leq a \leq p-2$, and compute a^α mod p.
3. A's public key is (p, α, α^a); A's private key is a.

The following algorithm shows how B can encrypt a message m to A.

Algorithm 2 *Encryption*

1. Obtain A's authentic public key (p, α, α^a).
2. Represent the message as an integer m in the range $(0, 1, ..., p-1)$.

3. Select a random integer k, $1 \leq k \leq p-2$.
4. Compute $\gamma = \alpha^k \bmod p$ and $\delta = m \cdot (\alpha^a)^k \bmod p$.
5. Send the ciphertext $c = (\gamma, \delta)$ to A.

Here is the algorithm that A uses to recover the message m.

Algorithm 3 *Decryption*

1. Use the private key a to compute $\gamma^{p-1-a} \bmod p = \gamma^{-a}$
2. Recover m by computing $(\gamma^{-a}) \cdot \delta \bmod p$.

Now for the modified ElGamal cryptosystem over $U^2(\mathbb{Z}_n)$, as we mentioned before, two cases were considered.
Case 1: $U(\mathbb{Z}_n)$ and $U^2(\mathbb{Z}_n)$ are cyclic: The elements of $U^2(\mathbb{Z}_n)$ in this case have the form $U^2(\mathbb{Z}_n) = \{r^i \bmod n : \gcd(i, \varphi(n)) = 1\}$, where r is the generator of $U(\mathbb{Z}_n)$.

The extended ElGamal public key cryptosystem over $U^2(\mathbb{Z}_n)$ follows the next four algorithms:

Algorithm 4 *Generator of* $U^2(\mathbb{Z}_n)$

1. Find a generator θ_1 of $U(\mathbb{Z}_n)$.
2. Write the order of $U^2(\mathbb{Z}_n)$ as $p_1^{\alpha_1} p_2^{\alpha_2} ... p_k^{\alpha_k}$.
3. Select a random integer s, $0 \leq s \leq \varphi(n) - 1$, $(s, \varphi(n)) = 1$.
4. For $j = 1$ to i, do:
   4.1. Compute $\theta_1^{N/p_j} \bmod n$.
   4.2. If $\theta_1^{s(N/p_j)} \bmod n \equiv \theta_1$, then go to step 3.
5. Return s.

For the key generation, use this algorithm:

Algorithm 5 *Key generation*

1. Find a generator θ_1 of $U(\mathbb{Z}_n)$.
2. Find s using the previous algorithm.
3. Compute the order of $U^2(\mathbb{Z}_n)$ using $\varphi^2(n)$.
4. Select a random integer a, $2 \leq a \leq \varphi^2(n) - 1$, and compute $f = s^a (\bmod\ \varphi(n))$.
5. A's public key is (n, θ_1, s, f) and A's private key is a.

For the encryption of a message m, the following algorithm is used:

Algorithm 6 *Encryption*

1. B obtains A's authentic public key (n, θ_1, s, f).
2. Represent the message as an integer in $U^2(\mathbb{Z}_n)$.
3. Select a random integer k, $2 \leq k \leq \varphi^2(n) - 1$.
4. Compute $q = s^k(\text{mod}\varphi(n)), r = f^k(\text{mod}\varphi(n))$, $\gamma = \theta^k = \theta_1^q(\text{mod}n)$ and $\delta = m^r(\text{mod}n)$.
5. Send the cipher text $c = (q, \delta)$ to A.

Finally, to decrypt the message, use the next algorithm:

Algorithm 7 *Decryption*

1. Use the private key a to compute $b = \varphi^2(n) - a$.
2. Recover the message by computing $t = q^b(\text{mod}\varphi(n))$ and $\delta^t(\text{mod}n)$.

Case 2: $U^2(\mathbb{Z}_n)$ is cyclic, while $U(\mathbb{Z}_n)$ is not cyclic: The extended ElGamal public key cryptosystem over $U^2(\mathbb{Z}_n)$ follows the next four algorithms:

Algorithm 8 *Generator of* $U^2(\mathbb{Z}_n)$

1. Find a generator θ_1 of $U(\mathbb{Z}_n)$.
2. Write the order of $U^2(\mathbb{Z}_n)$ as $p_1^{\alpha_1} p_2^{\alpha_2} ... p_k^{\alpha_k}$.
3. Select a random integer s, $0 \leq s \leq \varphi(p) - 1$, $(s, \varphi(p)) = 1$.
4. For $j = 1$ to i, do:
   - 4.1. Compute $\theta_1^{N/p_j} \bmod p$.
   - 4.2. If $\theta_1^{s(N/p_j)} \bmod p \equiv \theta_1$, then go to step 3.
5. Use the Chinese Remainder Theorem to find θ, and s by solving the system of congruencies: $x \equiv 2(\text{mod}3)$ and $x \equiv \theta_1^s(\text{mod}p)$.
6. return s.

Now, for the key generation, A uses the following algorithm:

Algorithm 9 *Key generation*

1. Find a generator θ_1 of $U(\mathbb{Z}_p)$.
2. Find s using the previous algorithm.
3. Compute the order of $U^2(\mathbb{Z}_p)$ using $\varphi^2(p)$.
4. Select a random integer a, $2 \leq a \leq \varphi^2(p) - 1$, and compute $f = s^a(\text{mod}\ \varphi(p))$.
5. A's public key is (p, θ_1, s, f) and A's private key is a.

For B to encrypt a message m for A, he uses this algorithm:

Algorithm 10 *Encryption*

1. B obtains A's authentic public key (n, θ_1, s, f).
2. Represent the message as an integer in $U^2(\mathbb{Z}_p)$.
3. Select a random integer k, $2 \leq k \leq \varphi^2(p) - 1$.
4. Compute $q = s^k (\text{mod}\varphi(p)), r = f^k (\text{mod}\varphi(p))$, $\gamma = \theta^k = \theta_1^q (\text{mod}p)$ and $\delta = m^r (\text{mod}p)$.
5. Send the cipher text $c = (q, \delta)$ to A.

Finally to decrypt the message m, A applies the next algorithm:

Algorithm 11 *Decryption*

1. Use the private key a to compute $b = \varphi^2(p) - a$.
2. Recover the message by computing $t = q^b (\text{mod}\varphi(p))$ and $\delta^t (\text{mod}p)$.

3 Construction Of $U^3(\mathbb{Z}_n)$

In this paper, in order to apply the ElGamal cryptosystem on $U^3(\mathbb{Z}_n)$, it must be a cyclic group, so we are concerned about the values of n that makes $U^3(\mathbb{Z}_n)$ cyclic.

Lemma 12 *$U(\mathbb{Z}_{3^\alpha})$, $U(\mathbb{Z}_{\varphi(3^\alpha)})$, and $U(\mathbb{Z}_{\varphi(\varphi(3^\alpha))})$ are cyclic for all $\alpha > 0$.*

Lemma 13 *$U(\mathbb{Z}_{2.3^\alpha})$, $U(\mathbb{Z}_{\varphi(2.3^\alpha)})$, and $U(\mathbb{Z}_{\varphi(\varphi(2.3^\alpha))})$ are cyclic for all $\alpha > 0$.*

Now, we define the operation that gives the group isomorphic to $U^3(\mathbb{Z}_n)$ as follows:

Theorem 3.1 *Let $U(\mathbb{Z}_n)$, $U(\mathbb{Z}_{\varphi(n)})$, and $U(\mathbb{Z}_{\varphi(\varphi(n))})$ be cyclic groups. Then $f : U^3(\mathbb{Z}_n) \longrightarrow \mathbb{Z}_{\varphi(\varphi(\varphi(n)))}$ given by*

$$f(a) = \log_{g_3} \log_{g_2} \log_{g_1} a \bmod n$$

is an isomorphism, where g_1, g_2, and g_3 are the generators of $U(\mathbb{Z}_n)$, $U(\mathbb{Z}_{\varphi(n)})$, and $U(\mathbb{Z}_{\varphi(\varphi(n))})$ respectively.

Proof. Let $U(\mathbb{Z}_n)$, $U(\mathbb{Z}_{\varphi(n)})$, and $U(\mathbb{Z}_{\varphi(\varphi(n))})$ be cyclic groups. Let g_1 be a generator of $U(\mathbb{Z}_n)$. Since $U(\mathbb{Z}_n)$ is cyclic and finite of order $\varphi(n)$, then $U(\mathbb{Z}_n) \approx \mathbb{Z}_{\varphi(n)}$ by a function $h_1 : U(\mathbb{Z}_n) \longrightarrow \mathbb{Z}_{\varphi(n)}$ defined by $h_1(a) = log_{g_1} a \bmod n$. Now since $U^2(\mathbb{Z}_n)$ is a subset of $U(\mathbb{Z}_n)$, and $U(\mathbb{Z}_{\varphi(n)})$ is a subset of $\mathbb{Z}_{\varphi(n)}$, then $h_1 : U^2(\mathbb{Z}_n) \longrightarrow U(\mathbb{Z}_{\varphi(n)})$ is an isomorphism.

Let g_2 be a generator of $U(\mathbb{Z}_{\varphi(n)})$. Since $U(\mathbb{Z}_{\varphi(n)})$ is cyclic and finite of order $\varphi(\varphi(n))$, then $U(\mathbb{Z}_{\varphi(n)}) \approx \mathbb{Z}_{\varphi(\varphi(n))}$ by a function $h_2 : U(\mathbb{Z}_{\varphi(n)}) \longrightarrow \mathbb{Z}_{\varphi(\varphi(n))}$ defined by $h_2(b) = \log_{g_2} b \bmod \varphi(n)$ or $h_2 \circ h_1 : U^2(\mathbb{Z}_n) \longrightarrow \mathbb{Z}_{\varphi(\varphi(n))}$ defined by $h_2 \circ h_1(a) = \log_{g_2} \log_{g_1} a \bmod n$. Now since $U^3(\mathbb{Z}_n)$ is a subgoup of $U^2(\mathbb{Z}_n)$ and $U(\mathbb{Z}_{\varphi(\varphi(n))})$ is a subgroup of $\mathbb{Z}_{\varphi(\varphi(n))}$, then $h_2 \circ h_1 : U^3(\mathbb{Z}_n) \longrightarrow U(\mathbb{Z}_{\varphi(\varphi(n))})$ is an isomorphism.

Now let g_3 be a generator of $U(\mathbb{Z}_{\varphi(\varphi(n))})$. Since $U(\mathbb{Z}_{\varphi(\varphi(n))})$ is cyclic and finite of order $\varphi(\varphi(\varphi(n)))$, then the function $h_3 : U(\mathbb{Z}_{\varphi(\varphi(n))}) \longrightarrow \mathbb{Z}_{\varphi(\varphi(\varphi(n)))}$ defined by $h_3(c) = \log_{g_3} c \bmod \varphi(\varphi(n))$ is an isomorphism.

This implies that $h_3 \circ h_2 \circ h_1(a) : U^3(\mathbb{Z}_n) \longrightarrow \mathbb{Z}_{\varphi(\varphi(\varphi(n)))}$ defined by $h_3 \circ h_2 \circ h_1(a) = \log_{g_3} \log_{g_2} \log_{g_1} a \bmod n$ is an isomorphism. ■

Now for the construction of $U^3(\mathbb{Z}_n)$, we use the following algorithm:

1. Find a generator g_1 for the group $U(\mathbb{Z}_n)$.
2. Write each element in $U(\mathbb{Z}_n)$ as a power of g_1.

   $U(\mathbb{Z}_n) = \{g_1^i \bmod n, 0 \le i \le \varphi(n)\}$.
3. Find a generator g_2 for the group $U(\mathbb{Z}_{\varphi(n)})$.
4. Write each element in $U(\mathbb{Z}_{\varphi(n)})$ as a power of g_2.

   $U(\mathbb{Z}_{\varphi(n)}) = \{g_2^i \bmod \varphi(n), 0 \le i \le \varphi(\varphi(n))\}$.
5. Find $U^3(\mathbb{Z}_n) = \{g_1^{g_2^i \bmod \varphi(n)} \bmod n, \gcd(i, \varphi(\varphi(n))) = 1\}$.

The following example shows how to find $U^3(\mathbb{Z}_{11})$, and the isomorphic element corresponding to each of its elements in $\mathbb{Z}_2$

Example 14 *A generator g_1 of $U(\mathbb{Z}_{11})$ is 2.*

We have $U(\mathbb{Z}_{11}) = \{1 = 2^0, 2 = 2^1, 3 = 2^8, 4 = 2^2, 5 = 2^4, 6 = 2^9; 7 = 2^7, 8 = 2^3, 9 = 2^6, 10 = 2^5\}$ all $\bmod 11$, and $U(\mathbb{Z}_{\varphi(11)}) = U(\mathbb{Z}_{10})$. A generator g_2 of $U(\mathbb{Z}_{10})$ is 3.

$U(\mathbb{Z}_{10}) = \{1 = 3^0, 3 = 3^1, 7 = 3^3, 9 = 3^2\}$ *all* $\bmod 10$.

$$\begin{aligned} U^3(\mathbb{Z}_{11}) &= \{2^{3^i} \bmod 11, \gcd(i,4) = 1\} \\ &= \{2^{3^1} \bmod 11, 2^{3^3} \bmod 11\} \\ &= \{2^3 \bmod 11, 2^7 \bmod 11\} \\ &= \{8, 7\} \end{aligned}$$

$U(\mathbb{Z}_{\varphi(\varphi(11))}) = U(\mathbb{Z}_{\varphi(10)}) = U(\mathbb{Z}_4) = \{1, 3\}$.

Now to find the isomorphism group of $U^3(\mathbb{Z}_{11})$, we find g_3, the generator of $U(\mathbb{Z}_4)$, $g_3 = 3$.

$$\begin{aligned} f(7) &= \log_3 \log_3 \log_2 7 \bmod 11 \\ &= \log_3 \log_3 \log_2 2^7 \bmod 11 \\ &= \log_3 \log_3 7 \bmod 11 \\ &= \log_3 \log_3 3^3 \bmod 11 \\ &= \log_3 3 \bmod 11 \\ &= 1 \end{aligned}$$

$$
\begin{aligned}
f(8) &= \log_3 \log_3 \log_2 8 \bmod 11 \\
&= \log_3 \log_3 \log_2 2^3 \bmod 11 \\
&= \log_3 \log_3 3 \bmod 11 \\
&= \log_3 1 \bmod 11 \\
&= 0
\end{aligned}
$$

Therefore, 7 in $U^3(\mathbb{Z}_{11})$ is isomorphic to 1 and 8 in $U^3(\mathbb{Z}_{11})$ is isomorphic to 0, and thus $U^3(\mathbb{Z}_{11}) = \{7, 8\} \approx \{0, 1\} = \mathbb{Z}_2$.

The following Theorem explains how to find a generator of $U^3(\mathbb{Z}_n)$.

Theorem 3.2 *Let g be a generator of $U^3(\mathbb{Z}_n)$. Then g has the form $g_1^{g_2^{g_3} (\bmod \varphi(n))} (\bmod n)$, where g_1, g_2, and g_3 are the generators of $U(\mathbb{Z}_n)$, $U(\mathbb{Z}_{\varphi(n)})$, and $U(\mathbb{Z}_{\varphi(\varphi(n))})$ respectively.*

Proof. We have from Theorem 3.1 that $U^3(\mathbb{Z}_n) \approx \mathbb{Z}_{\varphi(\varphi(\varphi(n)))}$ by a function $f : U^3(\mathbb{Z}_n) \longrightarrow \mathbb{Z}_{\varphi(\varphi(\varphi(n)))}$ defined by $f(a) = \log_{g_3} \log_{g_2} \log_{g_1} a \mod n$, where f is a group isomorphism under addition in $\mathbb{Z}_{\varphi(\varphi(\varphi(n)))}$. Moreover, if $U^3(\mathbb{Z}_n)$ is cyclic of generator g, then $\mathbb{Z}_{\varphi(\varphi(\varphi(n)))}$ is cyclic of generator $f(g)$.

However, $(\mathbb{Z}_{\varphi(\varphi(\varphi(n)))}, +)$ is cyclic of generator 1, then $f(g) = 1$. Therefore,

$$
\begin{aligned}
g &= f^{-1}(1) \\
&= (\log_{g_3} \log_{g_2} \log_{g_1})^{-1}(1) \\
&= \log_{g_1}^{-1} \log_{g_2}^{-1}(g_3^1) \\
&= \log_{g_1}^{-1}(g_2^{g_3}) \\
&= g_1^{g_2^{g_3} (\bmod \varphi(n))} (\bmod n).
\end{aligned}
$$

■

Definition 15 *Let f be the function defined in Theorem 3.1. The operations in $(U^3(\mathbb{Z}_n), \oplus, \otimes)$ are defined as follows:*

1. *$x \oplus y = x^{\log_r y} \bmod n$, where r is the generator of $U(\mathbb{Z}_n)$.*
2. *$x \otimes y = f^{-1}(f(x) + f(y))$.*
3. *$x^n = f^{-1}(nf(x))$.*

4 ElGamal Cryptosystem over $U^3(\mathbb{Z}_n)$

The following three algorithms illustrate the ElGamal Cryptosystem over $U^3(\mathbb{Z}_n)$.

For key generation, entity A must do the following:

Algorithm 16 *(key generation)*

1. find a generator g of $U^3(\mathbb{Z}_n)$.
2. select a random integer b, $1 \leq b \leq \varphi^3(n)$.

3. compute $B = g^b$.
4. A's public key is (g, B) and A's private key is b.

To encrypt a message m for A, entity B must use the following algorithm:

Algorithm 17 *(Encryption)*

1. obtain A's public key (g, B).
2. represent the message as an integer m in $U^3(\mathbb{Z}_n)$.
3. select a random integer a, $1 \leq a \leq \varphi^3(n)$.
4. compute $s = B^a$.
5. compute $A = g^a$.
6. compute $X = m \otimes s$.
7. send the cipher text $c = (A, X)$.

To recover the message m, entity A uses this algorithm:

Algorithm 18 *(decryption)*

1. use the private key to compute $s = A^b$.
2. compute s^{-1}.
3. recover the message m by computing $m = X \otimes s^{-1}$.

Theorem 4.1 *Given a generator g of $U^3(\mathbb{Z}_n)$. Define $B = g^b$, $A = g^a$, $s = B^a = A^b$, and $X = m \otimes s$. If $k \in U^3(\mathbb{Z}_n)$ such that $k = X \otimes s^{-1}$, then $k = m$.*

Proof. We have $s = B^a = A^b = g^{ab}$, and $X = m \otimes s = m \otimes g^{ab}$, then $k = X \otimes s^{-1} = m \otimes g^{ab} \otimes (g^{ab})^{-1} = m$.

■

The following example is an application of our cryptosystem.

Example 19 *Let $n = 3^4 = 81$.*
By applying Theorem 3.1, we get that $U^3(\mathbb{Z}_{81}) \approx \mathbb{Z}_6$, where

$$5 \approx 4, 23 \approx 3, 32 \approx 0, 50 \approx 1, 59 \approx 2, 77 \approx 5$$

Key Generation

1. *$g = 50$ is a generator of $U^3(\mathbb{Z}_{81})$*
2. *select $b = 4$*
3. *compute*

$$B = g^b = 50^4 = f^{-1}(4f(50)) = f^{-1}(4) = 5$$

4. *Public key is $(50, 5)$ and private key is 4.*

Encryption

1. *choose* $a = 2$
2. *compute*

$$s = B^a = 5^2 = f^{-1}(2f(5)) = f^{-1}(2.4 \bmod 6) = f^{-1}(2) = 59$$

3. *compute*

$$A = g^a = 50^2 = f^{-1}(2f(50)) = f^{-1}(2) = 59$$

4. *choose* $m = 77$.
5. *compute*

$$X = m \otimes s = 77 \otimes 59 = f^{-1}(f(77) + f(59)) = f^{-1}(5 + 2 \bmod 6) = f^{-1}(1) = 50.$$

6. *cipher text is* $(59, 50)$.

Decryption

1. *compute*

$$s = A^b = 59^4 = f^{-1}(4f(59)) = f^{-1}(4.2 \bmod 6) = f^{-1}(2) = 59$$

2. *compute*

$$s^{-1} = f^{-1}([f(59)]^{-1}) = f^{-1}(2^{-1}) = f^{-1}(4) = 5$$

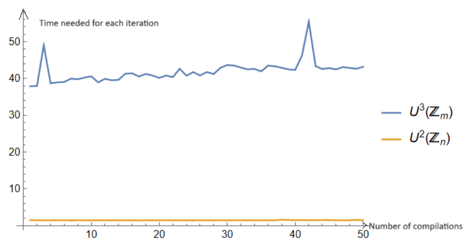
3. *compute*

$$m = X \otimes s^{-1} = 50 \otimes 5 = f^{-1}(f(50) + f(5)) = f^{-1}(1 + 4 \bmod 6) = f^{-1}(5) = 77.$$

5 Efficiency and security of the cyptosystem

In this section we present a comparative study between the efficiency and security of our cryptosystem and that present in [9]. Since the Baby step- Giant step attack algorithm depends basically on the operation g^i, where g is the generator of the selected group, it was enough for us to compare the compiling time of g^i for both groups. We generated our algorithms on Wolfram Mathematica 12. We used two groups $U^2(\mathbb{Z}_n)$ and $U^3(\mathbb{Z}_m)$ of approximately equal orders (difference between orders is 2), and after running both programs 50 times, on randomly chosen elements from both groups, the results are presented in Figure 1. The results prove that the timing for each iteration in $U^3(\mathbb{Z}_m)$ was around 30 times that of $U^2(\mathbb{Z}_n)$, which indicates that the iterations are way more complex in our new cryptosystem, and made it way harder to crack the system. Note that the blue curve corresponds to timing of $U^3(\mathbb{Z}_m)$, and the orange curve corresponds to that of $U^2(\mathbb{Z}_n)$.

Figure 1: Time comparison between iterations done on algorithms of $U^2(\mathbb{Z}_n)$ and $U^3(\mathbb{Z}_m)$



References

[1] A.-N. El-Kassar, H. Y. Chehade. Generalized group of units. *Math. Balkanica.* 2006;20(3):275-286.

[2] A.-N. El-Kassar, H. Y. Chehadi, D. Zentout. Quotient rings of polynomials over finite fields with cyclic group of units. *Proceedings of the International Conference on Research Trends in Science and Technology.* 2002; 257-266.

[3] A.-N. El-Kassar, R. A. Haraty. ElGamal public-key cryptosystem using reducible polynomials over a finite field. *IASSE.* 2004;189-194.

[4] A.-N. El-Kassar, R. A. Haraty. ElGamal Public-key cryptosystem in multiplicative groups of quotient rings of polynomials over finite fields. *Computer Science and Information Systems.* 2005;2(1):63-77.

[5] A.-N. El-Kassar, M. Rizk, N. M. Mirza, Y. A. Awad. ElGamal public key cryptosystem in the domain of Gaussian integers. *Int. J. Appl. Math.* 2001;7(4):405-412.

[6] M. El Hindi, T. Kadri. On the generalized multiplicative euler phi function. *arXiv: 2111.13474.* 2021.

[7] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *Advances in Cryptography.* Springer, Berlin, Heidelberg. 1985;10-18.

[8] J. A. Gallian. *Contemporary Abstract Algebra*, 7th ed., Cengage: Learning. 2012.

[9] R. Haraty, A. -N. Kassar, S. Fanous. Hardening the ElGamal cryptosystem in the setting of the second group of units, *International Arab Journal of Information Technology (IAJIT).* 2014;11(5):514-520.

[10] Irawadi, S. Nonsingular matrix as private key on ElGamal cryptosystem. In Journal of Physics: Conference Series (Vol. 1821, No. 1, p. 012018). IOP Publishing. 2021.

[11] Irawadi, S. Discrete Logarithmic Improvement for ElGamal Cryptosystem Using Matrix Concepts. In 2020 8th International Conference on Cyber and IT Service Management (CITSM) (pp. 1-5). IEEE. 2020.

[12] T. Kadri, A.-N. El-Kassar. The third group of units of the ring $\mathbb{Z}_n$. *JP Journal of Algebra, Number Theory and Applications.* 2016;38(4):385-413.

[13] Mahalanobis, A. A simple generalization of the ElGamal cryptosystem to non-abelian groups. Communications in Algebra®. 2008;36(10), 3878-3889.

[14] Mahalanobis, A. A simple generalization of the ElGamal cryptosystem to non-abelian groups II. Communications in Algebra. 2012;40(9), 3583-3596.

[15] R. Rivest, A. Shamir, L. Aldeman. A method for obtaining digital signatures and public key cryptosystems. *Communications of the ACM.* 1978;21(2):120-126,

[16] Ranasinghe, R., Athukorala, P. A generalization of the ElGamal public-key cryptosystem. Journal of Discrete Mathematical Sciences and Cryptography; 2022;25(8), 2395-2403.