Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 11, No. 01, March 2025.

# A NEW APPROACH COMBINING RSA AND ELGAMAL ALGORITHMS: ADVANCEMENTS IN ENCRYPTION AND DIGITAL SIGNATURES USING GAUSSIAN INTEGERS

Yahia Awad[1], Douaa Jomaa[1],Yousuf Alkhezi[2], Ramiz Hindi[1]

*(Received: 16-Jul.-2024, Revised: 11-Sep.-2024, Accepted: 2-Oct.-2024)*


## *ABSTRACT*

*This article introduces a novel approach that integrates the ElGamal and RSA algorithms to advance the security and efficiency of public-key cryptosystems. By combining these two established asymmetric-key algorithms, our method leverages their individual strengths and addresses the limitations of traditional systems, particularly in relation to the integer-factorization and discrete-logarithm problems. The application of Gaussian integers enhances the robustness of both encryption and digital signature processes, offering a more secure cryptographic framework. Our study involves a comprehensive analysis of the integrated algorithms, including practical implementations and extensive cryptanalytic evaluations focused on the integer-factorization and discrete-logarithm challenges. Quantitative assessments are provided to evaluate the effectiveness and computational efficiency of the proposed system. While key generation is slightly slower compared to using RSA or ElGamal individually, our approach delivers comparable performance in encryption and decryption, with notable improvements in robustness and versatility. In contrast to existing research predominantly focused on optical-image processing, our work extends the scope to a broader range of applications, enhancing both theoretical insights and practical implementations of cryptographic schemes. Future research will focus on optimizing key generation, exploring integration with existing security frameworks and evaluating performance in diverse real-world scenarios to further refine and validate the proposed approach.*



## *KEYWORDS*

*Combined RSA-ElGamal public-key cryptosystem, RSA, ElGamal, Digital signature, Gaussian integers.*


## 1. INTRODUCTION

Cryptography, an intricate fusion of art and science, has long been fundamental to ensuring secure communication throughout human history. From early simple ciphers to today's sophisticated digital-encryption techniques, the field has continually adapted to meet increasing demands for data security. In the contemporary digital era, where massive volumes of information are exchanged and stored globally, the urgency for robust and adaptable encryption solutions has never been greater. Public-key cryptography represents a significant breakthrough, revolutionizing security protocols with its dual-key system: a public key for encryption and a private key for decryption. This innovative approach allows for secure communication even when the encryption method is known, relying on the mathematical intricacies of cryptographic processes to maintain confidentiality and trust. For further details, see [4][8][11][21][25][36] and the references therein.

As computational power advances and cyber-threats become more sophisticated, the field of public-key cryptography continues to evolve. Recent research has made significant strides in several key areas. Extensions of classical systems, such as RSA, ElGamal and Rabin, have been explored through their application in Gaussian integers and finite fields, enhancing their security and resilience against attacks [6]-[7], [13]-[15]. Hybrid encryption systems, like the one introduced by Kuppuswamy et al. [24], combine public and private-key algorithms to enhance security and authentication. Novel hybrid algorithms, including the HRSA proposed by Panda et al. [28], use multiple prime numbers to complicate factorization, while Iswari et al. [22] and Ahmed et al. [3] have combined RSA with ElGamal and integrated integer factorization with discrete logarithms to improve efficiency and security. Additionally, Adeniyi et al. [2] have focused on integrating RSA and ElGamal with hash functions to bolster data integrity through enhanced digital signatures. Meanwhile, numerous studies have addressed public-key cryptosystems' application in optical-image processing, tackling specific


1. Y. Awad, D. Jomaa and R. Hindi are with Department of Mathematics and Physics, Lebanese Int. University, Faculty of Arts and Sciences, Bekaa Campus, Lebanon. Emails: yehya.awad@liu.edu.lb, o.a.douaa@gmail.com and r.math090@gmail.com
2. Y. Alkhezi is with College of Basic Education Mathematics Department, Public Authority for Applied Education and Training, Kuwait. Email: ya.alkhezi@paaet.edu.kw

challenges and opportunities in this specialized field [5][19][21][29][36]. These contributions advance security measures, but are often confined to particular applications.

The novelty of our research lies in the innovative integration of RSA and ElGamal algorithms, which are traditionally viewed as distinct entities in cryptographic practice. By strategically merging these two algorithms, we have developed a combined RSA-ElGamal public-key cryptosystem that harnesses their individual strengths while mitigating their respective weaknesses. This novel approach not only enhances the overall security of the system, but also provides a versatile framework adaptable to various cryptographic functions, including encryption, decryption and digital signatures. Our work is distinguished by a thorough analysis of the mathematical foundations of this new approach, rigorous cryptographic evaluations and a comprehensive comparative study. These elements collectively advance the field of cryptography, offering deeper insights and new possibilities for future developments in secure communication protocols.

The structure of this paper is as follows: Section 1 introduces the research objectives and context. Section 2 provides an overview of the essential mathematical concepts relevant to our work. In Section 3, we present our public-key generation, encryption and decryption algorithms, supported by formal proofs and numerical examples. Section 4 introduces our combined RSA-ElGamal algorithms and ElGamal digital-signature scheme, detailing key generation, signature creation and verification processes. Section 5 focuses on the security analysis of our combined RSA-ElGamal cryptosystem, evaluating its efficiency and comparing it with classical RSA and ElGamal schemes. This section also includes a comparative complexity analysis, offering insights into the computational costs and advantages of our proposed system.

## 2. PRELIMINARIES

In this section, we provide a concise overview of the mathematical concepts required for our work. For additional details, please refer to [9], [10] and [25].

### 2.1 Arithmetic in Z

In algebra, it is widely known that if we consider a group $G$ and an element g within that group, the order of g, represented as,$|g|$ refers to the smallest positive integer t for which $g^t \equiv e$. Furthermore, if there exists an element g in a group G such that G can be generated entirely by g, denoted as $G = \langle g \rangle = \{g^n | n \in Z\}$, we say that $G$ is a cyclic group and g is known as the generator of G, where the order of g is equal to the order of $G$ (i.e., $|g| = |G|$ ). Euler's phi function, represented as $\phi(n)$, denotes the count of positive integers that are both relatively prime to n and less than n. Additionally, the set of $\phi(n)$ integers that are relatively prime to n and do not contain different elements congruent to each other modulo n is referred to as a reduced residue system modulo n, denoted as Un. This set $U_n$ is cyclic if and only if n takes on the values 2, 4, $p^k$ or $2p^k$, where p is an odd prime and $k \geq 1$. For more information, we refer to [9] and the references therein.

**Theorem 2.1** [25] (Euler's Theorem) If $n$ is a positive integer and $a$ is an integer relatively prime to $n$, then $a^{\phi(n)} \equiv 1 (\text{mod } n)$.

**Theorem 2.2** [25] (Fermat's Theorem) Let $a$ be a positive integer and p be any prime number. If $p$ doesn't divide $a$, then $a^{p-1} \equiv 1 (\text{mod } p)$.

### 2.2 Arithmetic in Z[$i$]

The domain of Gaussian integers is the subring $|Z_i| = \{x + iy | a, b \in Z \text{ and } i^2 = -1\}$. It is well known that Z[$i$] is an Euclidean domain of norm $N(z) = x^2 + y^2$. Let $\gamma$ be a Gaussian integer. If $\gamma$ divides 1, then $\gamma$ is called a unit. As $\gamma$ is a unit, we call $\gamma\alpha$ an associate of the Gaussian integer $\alpha$. An element $\gamma \in$ Z[$i$] is said to be a unit if and only if $N(\gamma) = 1$. This implies that the only units in Z[$i$] are 1, -1, $i$ and $-i$. If a non-zero non-unit Gaussian integer $\pi$ is divisible only by units and associates, then it is called a Gaussian prime. The only Gaussian primes are $1 \pm i$, those Gaussian integers $\pi$ such that $N(\pi) = \pi\bar{\pi}$ which is a natural prime number of the form $4k + 1$ and those natural prime numbers of the form $4k + 3$. For more information, see [10], [20] and [27].

**Definition 2.1** [10] The complete residue system modulo β ∈ Z[$i$] is the set $A(\beta) = \{z \mid z \in \langle\beta\rangle\}$.

**Theorem 2.3** [10] Suppose that γ and β are any two non-zero relatively prime Gaussian integers. Then, $A(\gamma\beta) = \{s + r\gamma : s \in A(\gamma), r \in A(\beta)\}$.

**Theorem 2.4** [10] For any positive integer n, if we consider $\alpha = 1+i$, $p$ as a Gaussian prime in the form $4k + 3$ and $\pi$ as a Gaussian prime where $N(\pi) = \pi\bar{\pi}$ is a natural prime number $q$ in the form $4k + 1$, then the complete residue systems modulo prime powers in Z[$i$] are given as follows:

1. $A(\alpha^{2n}) = \{x + iy : 0 \le x \le 2^n - 1,\ 0 \le y \le 2^n - 1\}$ and it has an order of $2^{2n}$.
2. $A(\alpha^{2n+1}) = \{x + iy : 0 \le x \le 2^{n+1} - 1,\ 0 \le y \le 2^n - 1\}$ and it has an order of $2^{2n+1}$.
3. $A(p^n) = \{x + iy : 0 \le x \le p^n - 1, 0 \le y \le p^n - 1\}$ and it has an order of $p^{2n}$.
4. $A(\pi^n) = \{x : 0 \le x \le q^n - 1\}$ and it has an order of $q^n$.

**Theorem 2.5** [10] For any positive integer $n$, if we consider $\alpha = 1+i$, $p$ as a Gaussian prime in the form $4k + 3$ and $\pi$ as a Gaussian prime where $N(\pi) = \pi\bar{\pi}$ is a natural prime number $q$ in the form $4k + 1$, then the reduced residue systems modulo prime powers in Z[$i$] are given as follows:

1. $R(\alpha^n)=\{x + iy \in A(\alpha^n) : x \not\equiv y \pmod 2\}$ *and* it has an order of $\phi(\alpha^n) = 2^n - 2^{n-1}$.
2. $R(p^n)=\{x + iy \in A(p^n)$: $\gcd(x, p) \sim 1$ *or* $\gcd(y, p) \sim 1\}$and it has an order of $\phi(p^n)=p^{2n-2}(p^2 - 1)$.
3. $R(\pi^n)=\{x \in A(\pi^n)$: $\gcd(x, q) \sim 1\}$ *and* it has an order of $\phi(\pi^n) = q^{n-1}(q - 1)$.

**Remark 2.1** [10] Let $\beta$ be a Gaussian integer, then the factor ring of Z[$i$] modulo $\langle\beta\rangle$ is the set of all cosets of $\langle\beta\rangle$ denoted by $G\beta$ or Z[$i$]/ $\langle\beta\rangle$. Its elements are the equivalence classes of the form $[x+iy] = (x+iy)+\langle\beta\rangle$. The operations are defined by $[\alpha]+[\gamma] = [\alpha + \gamma]$ and $[\alpha][\gamma] = [\alpha\gamma]$, for every $\alpha, \gamma \in$ Z[$i$]/ $\langle\beta\rangle$. Note that the order of a factor ring modulo $\langle\beta\rangle$ is equal to the number of elements in $A(\beta)$. $G_\beta$ is a complete residue system modulo β and of order $q(\beta)$. In addition, the units form a group under multiplication, denoted by $U(\beta)$ or $G_\beta^*$, which is the reduced residue system modulo $\beta$.

**Definition 2.2** [10] Let $\beta$ be a Gaussian integer, then the order of $G_\beta^*$ is defined as $\phi(\beta)$, which is the extension of Euler's phi function to be the domain of Gaussian integers Z[$i$].

**Theorem 2.6** [10] $G_\beta^*$ is cyclic if and only if $\beta$ is of the form $\alpha$, $\alpha^2$, $\alpha^3$, $\pi^n$, $p$, $\alpha\pi^n$ *or* $\alpha p$.

**Theorem 2.7** [10] Suppose that $\eta = \beta_1\beta_2$ is a composite Gaussian integer such that both $\beta_1$ and $\beta_2$ are odd prime integers of the form $4k_1 + 3$ and $4k_2 + 3$, respectively. Then, the complete residue system modulo $\eta$ is the set $G_\eta = \{x + iy : 0 \le x \le \beta_1\beta_2 - 1,\ 0 \le y \le \beta_1\beta_2 - 1\}$.

## 2.3 Classical RSA Public-key Cryptosystem

The RSA public-key cryptosystem is widely recognized as one of the most prominent cryptographic systems, initially introduced by Ronald Rivest, Adi Shamir and Leonard Adleman in 1977 (refer to [32]). The security of RSA is rooted in two fundamental problems: the integer-factorization problem and the RSA problem. The integer-factorization problem involves finding the prime factorization of a positive integer $n = p_1^{e_1} p_2^{e_2} \ldots p_k^{e_k}$, where $p_i$'s are distinct primes and $e_i \ge 1$. On the other hand, the RSA problem entails finding an integer $m$ that serves as the $e^{th}$ root of $c$ modulo a composite integer $n$. In this scenario, $n$ is a product of two distinct odd primes $p$ and $q$ and $e$ is a positive integer satisfying $\gcd(e, (p-1)(q-1))=1$. It is widely acknowledged that while the integer-factorization problem and the RSA problem share similarities, this resemblance has not been formally proven yet (see [8] and [25]).

The RSA cryptosystem operates through the following steps: Entity $A$ generates two large, distinct random primes, $p$ and $q$ (approximately of the same size). They compute $n = pq$ and $\phi(n) = (p-1)(q-1)$ and then choose a random integer $e$ such that $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1$. Entity $A$ also computes the multiplicative inverse of $e$ modulo $\phi(n)$ and obtains the value $d$. The resulting public key is denoted as $(n, e)$, while the private key is denoted as $(p, q, d)$. To encrypt a plaintext $m \in Zn$, entity B employs the public key $(n, e)$ to compute the ciphertext $c \equiv m^e \pmod n$ and transmits it to entity $A$. Subsequently, entity $A$ utilizes the private key $d$ to recover the original plaintext by computing $m \equiv c^d \pmod n$.

## 2.4 Classical ElGamal Public-key Cryptosystem

The ElGamal public-key cryptosystem, introduced by Taher ElGamal in 1985 (refer to [12]), stands as a widely adopted and robust cryptographic technique. Its security is fundamentally based on the discrete logarithm problem (DLP), which poses the challenge of finding an integer $k$ within the range of $0 \leq k \leq p-1$, such that $\alpha^k \equiv \beta \pmod{p}$, with $p$ denoting a prime, $\alpha$ serving as a generator of $Z_p^*$ and $\beta$ representing an element in $Z_p^*$.

The ElGamal cryptosystem unfolds as follows: Entity $A$ initiates the process by generating a large random prime integer $p$, along with a generator $\alpha$ of the multiplicative cyclic group $Z_p^*$. Subsequently, a random integer $a$ is selected, adhering to the condition $1 \leq a \leq p-2$. Entity $A$ then computes $\alpha^a \pmod{p}$. The resulting public key is represented as $(p, \alpha, \alpha^a)$, while the private key remains as $a$.

To encrypt a plaintext $m \in Z_p$, entity $B$ proceeds by choosing another random integer $k$, satisfying $1 \leq k \leq p-2$. Subsequently, $\gamma \equiv \alpha^k \pmod{p}$ and $\delta \equiv m(\alpha^a)^k \pmod{p}$ are computed. The resulting ciphertext is then given by $c = (\gamma, \delta)$. Finally, for the decryption and recovery of the plaintext, entity $A$ applies the private key $a$ to compute $\gamma^{p-1-a} \pmod{p}$, from which the original message $m$ is obtained as $m = (\gamma^{-a}).\delta \pmod{p}$.

## 2.5 RSA and ElGamal Digital Signatures

Let's define some notations before discussing the RSA and ElGamal signature algorithms, including key generation, signature and verification algorithms (refer to [25]).

### 2.5.1 Prerequisite Notations

1. $M$ (Message Space): This represents a collection of elements to which a signer can attach a digital signature.
2. $M_S$ (Signing Space): It refers to a collection of components on which the signature transformations are applied.
3. $S$ (Signature Space): It denotes a collection of items in $M$ that are associated with messages. These components establish a link between the signer and the message.
4. $R$ (Redundancy Function): It represents a one-to-one mapping from $M$ to $M_S$. It is important that $R$ is not multiplicative, meaning that $R(ab) \neq R(a)R(b)$ for all pairs of relatively prime elements $a$ and $b$ in $M$.
5. $M_R$: It refers to the image of $R$.
6. $R^{-1}$: It represents the inverse of $R$ and maps elements from $M_R$ back to $M$.
7. $h$ (Hash Function): It is a one-way function with its domain defined as $M$ .
8. $M_h$ (Hash Value Space): If $h : M \rightarrow M_h$, then $M_h$ is a sub-set of $M_S$.

### 2.5.2 Hash Function

The hash function [25] is a fundamental cryptographic tool widely employed in protocols. It generates a hash value denoted as $\tilde{m} = h(m)$, a concise, fixed-length bit string used to represent a specific message (e.g. fingerprints). To ensure the security of the hash function, three fundamental properties must be satisfied:

1. Preimage Resistance (or the one-way property): This ensures computing the original message $m$ given that the hash value m is computationally infeasible.
2. Weak Collision Resistance: A form of pre-image resistance, making it computationally infeasible to find two distinct messages $m_1 \neq m_2$ that produce the same hash values; i.e., $m_1 = m_2$.
3. Collision Resistance: It ensures it's challenging to find two distinct inputs $m_1 \neq m_2$ that hash to the same value; i.e., $h(m_1) = h(m_2)$.

Thus, it's crucial to highlight that when dealing with the hash-value representation of a message, both signature generation and verification operate on the hash value itself rather than the original message. Moreover, digital signatures are broadly categorized into two main types.

### 2.5.3 Digital Signature

There are two types of digital signatures

1. Digital signature with an appendix: This type of signature requires the original message as an input during the verification process. It utilizes cryptographic hash functions instead of custom redundancy functions, making it less vulnerable to existential forgery attacks. One example of this method is the ElGamal signature, introduced by Taher ElGamal in 1985. It is a digital-signature scheme that relies on the discrete-logarithm problem (DLP). It is a probabilistic algorithm used to generate digital signatures for messages of any length. The scheme requires a hash function, denoted as $h$, which maps messages to integers modulo a large prime number $p$. It is described as follows: Entity $B$ signs the message $m \in Z_p$ by selecting a random secret integer $k$, such that $1 \leq k \leq p\text{-}2$ with $\gcd(k, p\text{-}1) = 1$. Then, entity $B$ computes $r \equiv \alpha^k \pmod p$, $k^{-1} \pmod{p\text{-}1}$ and $s \equiv k^{-1}\,(h(m)\text{-}ar) \pmod{p\text{-}1}$. The signature is $(r, s)$. Now, entity $A$ verifies $B$'s signature by verifying that $1 \leq r \leq p\text{-}1$, otherwise the signature is rejected. Then, entity $A$ computes $v_1 \equiv (\alpha^a)^r r^s \pmod p$ and $v_2 \equiv \alpha^{h(m)} \pmod p$. The signature is accepted if and only if $v_1 = v_2$. For further information, please see [25].
2. Digital signature with message recovery: Unlike the previous type, this method does not need the original message for the verification process. The original message can be extracted from the signature. The RSA signature is an example of a technique that provides digital signatures with message recovery. It was introduced in 1978 and it is the most commonly used digital-signature system in practice since its verification process is fast and easy. Its security is also based on the integer-factorization problem. It is described as follows: Entity $B$ signs a message $m \in M$ by computing $\tilde{m} = R(m)$, where $m \in [0, n-1]$ and computes the signature $s \equiv \tilde{m}^d \pmod n$. Now, entity $A$ verifies $B$'s signature by computing $\tilde{m} \equiv s^e \pmod n$, which should be in $M_R$ and recovers $m = R^{-1}(m)$. For further information, please see [25].

**Remark 2.2** We will employ the hash function $h(m) = m^3$ with a specified modulus depending on the cryptographic context. In the case of ElGamal encryption, we take mod $p$, where $p$ is a prime, while in RSA encryption, we take mod $n$, where $n$ is a composite integer. This hash function is chosen for multiple reasons. Firstly, it maintains pre-image resistance, making it computationally difficult to find any pre-image m given $\tilde{m}$, such that $h(m) \equiv \tilde{m} \bmod$ modulus. Secondly, it upholds second preimage resistance since, given a pre-image $m_1$, it is computationally infeasible to generate another distinct preimage $m_2$ such that $h(m_1) \equiv h(m_2) \bmod$ modulus and $m_1 \neq m_2$. Thirdly, it preserves collision resistance, making it computationally impracticable to discover any two distinct inputs $m_1$ and $m_2$ where $m_1 \neq m_2$ and $h(m_1) \equiv h(m_2) \bmod$ modulus.

## 3. COMBINED RSA-ELGAMAL ALGORITHMS AND ELGAMAL CRYPTOSYSTEM

In this section, we present a novel combined RSA-ElGamal public-key encryption scheme that combines the RSA and ElGamal encryption schemes. We provide the algorithms for public-key generation, encrypion and decryption, along with accompanying proofs. Additionally, we illustrate the concepts with a numerical example.

### 3.1 Methodology

The ElGamal public-key cryptosystem relies on the discrete-logarithm problem, while the strength of the RSA public key cryptosystem lies in the difficulty of the integer-factoring problem. In this proposal, we present a novel algorithm that combines both RSA and ElGamal public-key cryptosystems. To achieve this, we first implement a modified ElGamal scheme using Gaussian integers and then utilize the RSA scheme in the domain of Gaussian integers.

Here is a brief overview of the process: We start by generating a large prime number $p$ along with a generator $\alpha$ for the group $G_p^*$. Next, we select a random positive integer $a < p^2\text{-}1$ and compute $\alpha^a \pmod p$. Following that, we choose two Gaussian primes $q$ and $r$ and find their product $\eta = qr$. Subsequently, we select a random integer $e$ and using the extended Euclidean algorithm, we determine its unique inverse $d \in G_\eta$, ensuring that $\gcd(e, \phi(\eta)) = 1$ and $1 < e, d < \phi(\eta)$. The resulting public key is given by $(p, \alpha, \alpha^a, \eta, e)$ and the private key is represented as $(a, q, r, d)$.

To encrypt a message $m \in G_p$, we randomly choose a positive integer $k< p^2 -1$ and compute the ciphertext $c \equiv \text{Me}(\text{mod } \eta)$, where $M = \gamma + \delta i$ with $\gamma \equiv \alpha^k$ and $\delta \equiv m(\alpha^a)^k$, which are elements in $G_p$. For the decryption of the sent ciphertext $c$, we utilize the private keys $a$ and $d$ to recover the original message. This is achieved by computing

$$m = \left[\left(\left(Re\left(c^d mod\ \eta\right)\right)^{q(p)-1-a}(modp)\right).\left(Im\left(c^d mod\eta\right)(modp)\right)\right](mod\eta). \quad (1)$$

### 3.1.1 Choice of the Gaussian Primes

In the following discussion, we will present an analysis of the primes $p$, $q$ and $r$ that will be selected in our novel approach. Initially, ElGamal scheme will be applied within the complete residue system $G_p$, which is defined as mentioned in Theorems 2.4 and 2.7 as follows:

1. If p is any natural prime integer, then $G_p = Z_p$.
2. If $p$ is a Gaussian prime such that $p\bar{p}$ is a natural prime of the form $4k+1$, then $G_p = \{x: 0 \leq x \leq p\bar{p} - 1\}$ and it has an order of $q(p) = p\bar{p}$.
3. If $p$ is a Gaussian prime of the form $4k+3$, then $G_P = \{x + iy: 0 \ll x \leq p - 1, 0 \leq y \leq p - 1\}$ and it has an order of $q(p) = p^2$.

For the sake of simplicity, we can utilize the initial implementation. Nevertheless, we shall employ the third implementation.

Second, the RSA scheme will be implemented in the complete residue system $G_\eta$ such that $\eta$ is a product of two Gaussian primes $q$ and $r$; i.e., $\eta = qr$, where we have three possible cases:

1. If $q = \pi_1$ and $r = \pi_2$, where $\pi_1\overline{\pi_1}$ and $\pi_2\overline{\pi_2}$ are two prime integers of the form $4k + 1$, then the complete residue system modulo $\eta$ is $G_\eta = \{x + qy : x \in G_q, y \in G_r\}$ and of order $q(\eta) = qr$. But, this case will be neglected due to its similarity to the classical settings.
2. If $q = \pi_1$ is a Gaussian prime such that $\pi_1\overline{\pi_1}$ is a prime integer of the form $4k + 1$ and $r$ is a prime integer of the form $4k + 3$, then the factorization of $\eta = \pi_1 r$ which has the form $x+ yi$ could be easily solved by simply finding the gcd($x$, $y$) which will be equal to r. Hence, this case will be also neglected, since our aim is to ensure the infeasibility of the factorization of $\eta$.
3. If $q$ and $r$ are both Gaussian primes of the form $4k + 3$, then the complete residue system modulo $\eta$ is $G_\eta = \{x + qy: x \in G_q, y \in G_r\}$ and of order $q(\eta) = q^2r^2$, which is huge enough to enhance the security of our approach compared to that of the classical one. Hence, this case will be chosen, since it is the best choice for the new implementation of the RSA scheme.

Thus, to provide a clearer justification: when using Gaussian primes of the form $4k + 3$ for both $q$ and $r$, the order of $G_\eta$ is $q(\eta) = q^2r^2$, meaning that the message space is not just doubled, but squared. This increase in size is crucial, because it exponentially expands the variety of possible plaintexts, making brute-force attacks, including exhaustive search methods, computationally infeasible. The complexity of deciphering the original message from the ciphertext becomes exponentially harder, requiring much more effort than in classical RSA systems with the same prime numbers.

Moreover, by increasing the size of the message space, the number of possible combinations of plaintexts grows exponentially. This means that any adversary attempting to recover the plaintext would face a significantly more difficult task, as the size of the problem space grows much larger. Traditional algorithms for factorization or solving the discrete-logarithm problem become less effective, further strengthening the cryptographic security of our approach.

### 3.1.2 Choice of Plaintext $m$

The plaintext, denoted as $m \in G_p$, can be expressed in two possible forms. The first form is $m=x +iy$, where both $x, y \in Z_p$ and $y \not\equiv 0 \pmod p$. The second form is $m = x$, where $x \in Z_p$.

## 3.2 Combined RSA-ElGamal Algorithms and ElGamal Public-key Scheme

In the subsequent sub-sections, we present a comprehensive explanation of our novel concept for

the "Combined RSA-ElGamal public-key cryptosystem." We elucidate the procedures for key generation, encryption and decryption in the following manner:

**Algorithm 3.1** Key generation for the combined RSA-ElGamal public-key scheme by entity A.

1. Generate three distinct large random odd prime integers $p$, $q$ and $r$ of the form $4k+3$ and approximately the same size.
2. Find a generator $\alpha$ of $G_p^*$.
3. Select a random integer $a$, such that $2 \leq a \leq p^2-2$ *and* then compute $\alpha^a \pmod p$.
4. Compute $\eta = qr$ *and* $\phi_\eta = (q^2-1)(r^2-1)$.
5. Select a random integer $e$ such that $1 < e < \phi_\eta$ *and* $\gcd(e, \phi_\eta) = 1$.
6. Use the extended Euclidean division algorithm to compute $d$, such that $ed \equiv 1 \pmod{\phi_\eta}$.
7. The public key is $(p, \alpha, \alpha^a, \eta, e)$ *and* the private key is $(a, q, r, d)$.

**Algorithm 3.2** Combined RSA-ElGamal public-key encryption by entity *B*.

1. Obtain A's public key $(p, \alpha, \alpha^a, \eta, e)$.
2. Choose a random integer $k$, such that $2 \leq k \leq p^2-2$.
3. Compute the ciphertext $c \equiv M^e \pmod \eta$, where $M = \gamma + \delta i$, $\gamma \equiv \alpha^k \pmod p$ and $\delta \equiv m(\alpha^a)^k \pmod p$.
4. Send the ciphertext $c$ to entity *A*.

**Algorithm 3.3** Combined RSA-ElGamal public-key decryption.

By using the private keys $a$ and $d$, entity *A* recovers the plaintext $m$ such that:

$$m \equiv (\mathrm{Re}(c^d \bmod \eta))^{p^2-1-a} \pmod p \; (\mathrm{Im}(c^d \bmod \eta)) \pmod p \pmod \eta.$$

**Theorem 3.1** The original message m is recovered by reducing

$$\left[\left(\left(Re(c^d mod\ \eta)\right)^{p^2-1-a}(mod\ p)\right).\left(Im(c^d mod\ \eta)(mod\ p)\right)\right](mod\ \eta).$$

**Proof 3.1** Consider the Gaussian integer $m' \in G_p$ such that

$$m' \equiv \left[\left(\left(Re\left(c^d mod \eta\right)\right)^{p^2-1-a}(mod\ p)\right).\left(Im\left(c^d mod\ \eta\right)(mod\ p)\right)\right](mod \eta) \tag{2}$$

Since $ed \equiv 1 \pmod{\phi(\eta)}$, then there exists an integer $k'$, such that $ed = 1 + k'\phi(\eta)$. Hence, there are two cases:

1. Suppose that the $\gcd(M, q)=1$. Then, by using the modified Euler's theorem to the domain of Gaussian integers, we have $M^{\phi(\eta)} \equiv 1 \pmod \eta$. After raising both sides of the congruence to the power of $k'$ and then multiplying them by $M$. We get,
$$M^{1+k'\phi(\eta)} \equiv M^{ed} \equiv c^d \pmod \eta \equiv M \pmod \eta. \tag{3}$$
2. Suppose that $\gcd(M, q) = q$. Then, we have $M \equiv 0 \pmod q$. Hence, $M^{k'(q^2-1)(r^2-1)} \equiv 0(mod\ q)$. After multiplying both sides by $M$, we get $M^{1+k'(q^2-1)(r^2-1)} \equiv 0(mod\ q)$ and hence, $M^{1+k'\emptyset(\eta)} \equiv M^{ed} \equiv c^d \equiv 0(mod\ q)$, since $M \equiv 0 \pmod q$. Then, $c^d \equiv M \pmod q$. By the same argument, we also get $c^d \equiv M \pmod r$. Since $q$ and $r$ are two distinct Gaussian primes, we obtain that $c^d \equiv M \pmod \eta$.

Hence, for any Gaussian integer $M$, we have $c^d \equiv M \pmod \eta$.Therefore,

$$\begin{aligned} m' &\equiv \left[\left(\left(Re(c^d mod\ \eta)\right)^{p^2-1-a}(mod\ p)\right).\left(Im(c^d mod\ \eta)(mod\ p)\right)\right](mod\ \eta) \\ &\equiv \left[\left(\left(Re(M)\right)^{p^2-1-a}(mod\ p)\right).\left(Im(M)(mod\ p)\right)\right](mod\ \eta) \end{aligned} \tag{4}$$

But, $M = \gamma + \delta i$. Then,

$$\begin{aligned} m' &\equiv \left[\left(\gamma^{p^2-1-a}(mod\ p)\right).\left(\delta(mod\ p)\right)\right](mod\ \eta) \equiv \left[\left(\alpha^{-ak}(mod\ p)\right).\left(\delta(mod\ p)\right)\right](mod\ \eta) \\ &\equiv [\alpha^{-ak}.m.\alpha^{ak} \pmod p] \pmod \eta \;\equiv\; m \pmod{qr}. \end{aligned} \tag{5}$$

**Example 3.1** (Combined RSA-ElGamal public-key scheme) Entity $A$ generates the keys as follows: If $p = 3$, and $\alpha = 2$ is a generator of $G_3^*$, then entity $A$ chooses the private key $a = 2$ and computes $\alpha^a = 1(\text{mod } 3)$. Also, if $q = 7$ and $r = 11$, then entity $A$ computes $\eta = 77$ and $\phi(\eta) = 5760$. After that, entity $A$ chooses $e = 971$ and by using the extended Euclidean division algorithm, finds $d = 611$ such that $ed \equiv 1(\text{mod } \phi(\eta))$. The public-key is (3, 2, 1, 77, 971) and the private key is (2, 7, 11, 611). Now entity $B$ encrypts the message $m = 2$ by selecting a random integer $k = 6$ and computing $\gamma \equiv 1(\text{mod } 3)$ and $\delta \equiv 2(\text{mod } 3)$. Then, entity $B$ assumes that $M = 1+2i$ and computes $c = 1-24i$. Entity $B$ then sends $c$ to entity $A$, which decrypts and recovers the message $m$ by computing $m \equiv [((\text{Re}(c^d \text{ mod } \eta))^{p2-a-1}(\text{mod } p)). (\text{Im}(c^d \text{ mod } \eta)(\text{mod } p))](\text{mod } \eta) \equiv 2$.

### 3.3 Security of the Proposed Combined RSA-ElGamal Cryptosystem

As the new proposed scheme combines elements of both the modified ElGamal and RSA schemes, each relying on distinct mathematical problems (the discrete-logarithm problem and the integer-factorization problem, respectively), the security of our combined RSA-ElGamal public-key scheme is predicated on both of these cryptographic challenges. To decrypt a message encrypted using this new scheme, one must first solve the integer-factorization problem, followed by solving the discrete-logarithm problem to obtain the plaintext. Consequently, the time required to compromise the new proposed scheme is influenced by the hacking times of both classical ElGamal and RSA schemes, as demonstrated in the comparative study outlined in Section 5. Additionally, the new scheme implements RSA in the domain of Gaussian integers by generating two odd primes, designated as $q$ and $r$, in the form of $4k + 3$. This choice results in the complete residue system $A(\eta)$ containing $q^2r^2$ elements, as opposed to just $qr$ elements in the classical scheme. Moreover, if we implement the ElGamal in the domain of Gaussian integers modulo a Gaussian prime p of the form $4k + 3$, the cyclic group $G_p^*$ has $p^2$-1 elements and the private key a can range from 2 to $p^2$-1. In contrast, the cyclic group of the classical scheme, $Z_p^*$, has $p$-1 elements and the private key a can range from 2 to $p$-1. Consequently, with equivalent effort to that in classical settings, our new scheme offers an expanded set of choices for plaintext and private keys by more than the square of the choices in the classical case. This extension bolsters the security provided by the new proposed scheme without necessitating any additional efforts.

## 4. COMBINED RSA-ELGAMAL SIGNATURE SCHEME

In this section, we introduce our proposed signature called the combined RSA-ElGamal signature scheme, where the key generation, signature and verification algorithms are given with proofs and a numerical example.

### 4.1 Description of the Combined RSA-ElGamal Signature

The concept behind our proposed signature arises from the necessity to enhance the security of our cryptosystem. Our signature approach combines elements from the classical ElGamal signature and the modified RSA signature within the domain of Gaussian integers. Its security is dependent on both the discrete-logarithmic and integer-factorization problems. In our proposed signature scheme, the message space, denoted as $M$, is represented by $Z_p$, while the ciphertext signing and signature spaces are all denoted as $G_\eta$. The redundancy function, denoted as $R : Z_p \rightarrow G_\eta$, can be made public and the hash function, denoted as $h : M \rightarrow Z_p$, is selected in a manner such that $p$ represents a large prime number.

The procedure is as follows: Firstly, a natural prime integer p is chosen, along with a generator $\alpha$ for $Z_p^*$. Then, a random positive integer $a$ is selected such that $a < p$-1 and $\alpha^a(\text{mod } p)$ is computed. In the next step, two Gaussian primes, $q$ and $r$, are chosen in the form $4k + 3$ and their product $\eta = qr$ is determined. Following this, a random integer e is selected and its unique (up to associates) inverse $d \in G_\eta$ is calculated using the extended Euclidean algorithm, satisfying gcd $(e, \phi(\eta)) = 1$ and $1<e, d<\phi(\eta)$. The public key comprises $(p, \alpha, \alpha^a, \eta, e)$, while the private key comprises $(a, q, r, d)$. To sign a message $m \in Z_p$, a random positive integer k is chosen such that $k<p$-1 and $\varsigma$ is computed as $\varsigma \equiv z^d(\text{mod } \eta)$, where $z = r' + si = R(\tilde{m})$, with $r' \equiv \alpha^k(\text{mod } p)$, and $s \equiv k^{-1}(\tilde{m} - ar')(\text{mod } p\text{-}1)$. To verify the signature $\varsigma$ and recover the original message $m$, $z$ is calculated as $z \equiv \varsigma^e(\text{mod } \eta)$, where it should belong to $M_R$ and $\tilde{m}$ is recovered such that $\tilde{m} = R^{-1}(z)$.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 11, No. 01, March 2025.

Finally, $h(m)$, $v_1$ and $v_2$ are computed so that $h(m) \equiv ks+ar' (\text{mod } p\text{-}1)$, $v_1 \equiv y^{\text{Re}(z)}.\text{Re}(z)^{\text{Im}(z)} (\text{mod } p)$, where $1 \leq \text{Re}(z) \leq p\text{-}1$ and $v_2 \equiv \alpha^{h(m)} (\text{mod } p)$. The signature is accepted if $v_1 = v_2$.

The aforementioned description is presented in a step-by-step manner in the following algorithms.

**Algorithm 4.1** Key generation for the combined RSA-ElGamal signature scheme:

1. Generate a random large odd prime $p$ and $a$ generator $\alpha$ *of* $Z_p^*$ *and* choose a random integer $a$, where $1 \leq a \leq p-2$.
2. Compute $y \equiv \alpha^a (\text{mod } p)$.
3. Generate two large, distinct odd primes, $q$ and $r$, each of roughly the same size.
4. Compute $\eta = qr$ *and* $\phi(\eta) = (q^2\text{-}1)(r^2\text{-}1)$.
5. Select a random integer $e$ such that $1 \leq e \leq \phi(\eta)$ with $\gcd(e, \phi(\eta)) = 1$.
6. Use the extended Euclidean algorithm to compute the unique integer $d$, such that $ed \equiv 1 (\text{mod } \phi(\eta))$.
7. The public key is $(p, \alpha, y, \eta, e)$ *and* the private key is $(a, q, r, d)$.

**Algorithm 4.2** Combined RSA-ElGamal signature generation by entity *B*.

1. Select a random secret integer $k$, such that $1 \leq k \leq p\text{-}2$ with $\gcd(k, p\text{-}1) = 1$.
2. Compute $r' \equiv \alpha^k (\text{mod } p)$, $k^{-1} (\text{mod } p\text{-}1)$, $h(m) = \tilde{m} = m^3 (\text{mod } p)$ *and* $s \equiv k^{-1}(\tilde{m} - ar') (\text{mod } p\text{-}1)$.
3. Take $z = r' + si = R(\tilde{m})$ *and* compute $\varsigma \equiv z^d (\text{mod } \eta)$.
4. *B's* signature for $m$ *is* $\varsigma$.

**Algorithm 4.3** Combined RSA-ElGamal verification by entity *A*.

1. Obtain *B*'s authentic public key $(p, \alpha, y, \eta, e)$.
2. Compute $z \equiv \varsigma^e (\text{mod } \eta)$.
3. Verify that $z \in M_R$, if not, reject the signature.
4. Recover $\tilde{m} = R^{-1}(z)$.
5. Verify that $1 \leq \text{Re}(z) \leq p\text{-}1$, if not, reject the signature.
6. Compute $v_1 \equiv y^{\text{Re}(z)}.\text{Re}(z)^{\text{Im}(z)} (\text{mod} p)$ *and* $v_2 \equiv \alpha^{h(m)} (\text{mod} p)$.
7. Accept signature *if* $v_1 = v_2$.

**Theorem 4.1** The signature verification method works.

**Proof 4.1** Let $\varsigma \equiv z^d (\text{mod } \eta)$ such that $z = r' + si$. Since $ed \equiv 1 (\text{mod } \phi(\eta))$, we have had $\varsigma^e \equiv z^{ed} \equiv z (\text{mod } \eta)$. Then, $R^{-1}(z) = R^{-1}(R(\tilde{m})) = \tilde{m} = h(m)$. Hence, $s \equiv k^{-1}(h(m) - ar') (\text{mod } p - 1)$. Multiply both sides by $k$, $ks \equiv h(m) - ar' (\text{mod } p - 1)$. Then, $h(m) \equiv ks + ar' (\text{mod } p - 1)$. Hence, $\alpha^{h(m)} \equiv \alpha^{ar'+ks} \equiv (\alpha^a)^{r'}.r^{s'} \equiv y^{r'}.r^{s'}$. Therefore, $v_1 = v_2$.

**Theorem 4.2** The redundancy function $R(\tilde{m}) = r' + si = \alpha^k + i\ [k - 1(\tilde{m} - ar') (\text{mod}(p - 1))\ ]$ is a $1-1$ mapping from $M$ *to* $M_S$.

**Proof 4.2** Suppose that $R(\tilde{m}_1) = R(\tilde{m}_2)$ such that $R(\tilde{m}_1) = \alpha^k + i\,[k^{-1}(\tilde{m}_1 - ar') (\text{mod}(p - 1))] \in G_\eta$ *and* $R(\tilde{m}_2) = \alpha^k + i\ [k^{-1}(\tilde{m}_1 - ar') (\text{mod}(p - 1))] \in G_\eta$. *Then,*

$$\alpha^k + i\ [k^{-1}(\tilde{m}_1 - ar') (\text{mod}(p - 1))] = \alpha^k + i[k^{-1}(\tilde{m}_2 - ar') (mod(p - 1))]. \quad (6)$$

Thus,

$$i[k^{-1}(\tilde{m}_1 - ar') (mod(p - 1))] = i[k^{-1}(\tilde{m}_2 - ar') (mod(p - 1))]. \quad (7)$$

Multiplying both sides by $(ik)$, we get $\tilde{m}_1 - ar' (mod(p - 1)) = \tilde{m}_2 - ar' \big(mod(p - 1)\big)$, which implies that $\tilde{m}_1 = \tilde{m}_2$.

**Theorem 4.3** The redundancy function $R(\tilde{m}) = r' + si = \alpha^k + i[k^{-1}(\tilde{m} - ar') (\text{mod}(p\text{-}1))]$ is not multiplicative.

**Proof 4.3** It is clear that

$$R(\tilde{m}_1).R(\tilde{m}_2) = (\alpha^k + i\ [k^{-1}(\tilde{m}_1 - ar') (\text{mod}(p - 1))]).(\alpha^k + i[k^{-1}(\tilde{m}_2 - ar') (mod(p - 1))] =$$

$$[\alpha^{2k} + i\alpha^k k^{-1}(\tilde{m}_1 + \tilde{m}_2 - 2ar') - k^{-2}(\tilde{m}_1 - ar')(\tilde{m}_2 - ar')]\ (mod(p - 1)) \quad (8)$$

But, $R(\tilde{m}_1\tilde{m}_2) = \alpha^k + i[k^{-1}(\tilde{m}_1\tilde{m}_2) - ar') (mod(p - 1))]$. Therefore, $R$ is not multiplicative.

**Example 4.1** (Combined RSA-ElGamal Signature) Entity B generates the keys as follows: If $p$=61 and $\alpha$=33 is a generator of $Z_{61}^*$. Then, entity $B$ chooses the private key $a$=58 and computes $y \equiv 33^{58} (\text{mod } 61) \equiv 27$. After that, entity $B$ selects $q$=9871 and $r$=5107 and computes both $\eta$=50411197 and $\phi(\eta)$=2541288659454720. Entity $B$ chooses $e$=1844480063626867 and solves $ed \equiv 1844480063626867$. $d \equiv 1 (\text{mod } 50411197)$, yielding $d$=993514318001083. Hence, the public-key is: ($p$=61, $\alpha$=33, $\alpha^a$ =27, $\eta$=50411197, $e$=1844480063626867) *and* the private key is ($a$ = 58, $q$ = 9871, $r$ = 5107, $d$ = 993514318001083). Assume that the hash function is $h(m) = \tilde{m} = m^3$. To sign a message $m$=42, entity $B$ selects a random integer $k$=7 and computes $r' \equiv 33^7 (\text{mod } 61) \equiv 38$, $k^{-1} \text{ mod}(p\text{-}1) \equiv 43$ *and* $\tilde{m}$=74088. Finally, entity $B$ computes $s \equiv 34(71884) (\text{mod } 60) \equiv 34$ and assumes $z$=38+34$i$ to compute $\varsigma \equiv 23812157\text{-}23285899i$. As a result, the signature for $m$ *is* $\varsigma$. Now, to verify the signature, entity A first computes $z$=38+34$i$, then computes $\tilde{m} = R^{-1}(z) = 74088 \in M_R$. After that, entity $A$ computes $v_1 \equiv 52$, $h(m)$=74088 *and* $v_2 \equiv 52$. Entity A accepts the signature since $v_1 = v_2$.

## 5. COMPARATIVE STUDY

In this section, we undertake a comparative analysis to position our novel cryptosystem against existing methodologies.

### 5.1 Security Evaluation and Comparative Analysis

In this study, we evaluate the security of our novel cryptographic scheme through assessments of attack, encryption and decryption times, supported by numerical simulations to measure its efficacy. Experimental investigations were conducted using an ALIENWARE laptop, specifically the Alienware 15 R4 model, equipped with an Intel(R) Core(TM) i7-8750H CPU, 16384MB RAM and BIOS version 1.20.0 (UEFI type). The laptop's robust specifications, including compatibility with Windows 11 Pro 64-bit, DirectX 12 and UEFI BIOS, along with features like Miracast Support and Microsoft Graphics Hybrid Compatibility, make it well-suited for computationally intensive experiments. Following this experimental setup, we perform a comparative analysis involving traditional RSA and ElGamal schemes alongside our novel hybrid approach, followed by a discussion of the identified strengths and weaknesses of our proposed cryptosystem.

### 5.2 Data Collection and Cryptanalysis

In this study, we employed Mathematica 10 to implement the algorithms for key generation, encryption, decryption and cryptanalysis of our new scheme. The prime numbers $p$, $q$ and $r$ were randomly selected from nineteen distinct intervals. These intervals, numbered from 1 to 19, encompass the ranges $10^1$ to $10^{38}$ for both q and r and $10^1$ to $10^{11}$ for $p$.

Due to the computational limitations of our current hardware, we were unable to explore higher exponents beyond $10^{38}$. Our personal computer, despite its capabilities, was unable to efficiently handle the larger key sizes required for more advanced cryptanalysis. In future work, we plan to leverage high-performance computing resources or cloud platforms to extend our analysis to larger primes, which will allow for a more thorough evaluation of the scheme's performance and security with larger key sizes.

#### 5.2.1 Key Generation

It has been observed that the total time needed for key generation in the new scheme is roughly equal to the sum of the individual times required for generating RSA and ElGamal keys.

#### 5.2.2 Encryption and Decryption

Regarding the encryption processes, the time slots required for all three cryptosystems are approximately the same, which is very significant, because our proposed cryptosystem does not require excessive durations to be done compared to the classical ones. The same applies to the decryption process.

### 5.2.3 Hacking Time Results

Hacking time denotes the period taken by an unauthorized entity to successfully decrypt or compromise the security of the combined RSA-ElGamal public-key cryptosystem under consideration. In our manuscript, this temporal measure is quantified in seconds and serves as a crucial metric for evaluating the system's resistance to potential breaches. The subsequent table (Table 1) provides comprehensive results for Hacking Time (HT) in seconds across each cryptosystem, delineating the time necessary to solve either the integer-factorization problem or the discrete-logarithm problem.

The first column of the table represents the data sizes, which are expressed as the $i^{th}$ power of 10, ranging from $10^1$ to $10^{19}$. These values correspond to bit lengths ranging from approximately 4 bits (for $10^1$) to approximately 63 bits (for $10^{19}$). The y-axis signifies the hacking time measured in seconds.

The table provides insights into the key-generation time for RSA, ElGamal and the proposed combined RSA-ElGamal scheme, elucidating the temporal dynamics of each cryptographic system across diverse data sizes. This analysis highlights that the key-generation time of the combined RSA-ElGamal scheme is the sum of the times required to generate keys for both RSA and ElGamal. This detailed examination underscores the performance attributes of the proposed cryptographic methodology and its implications for practical applications.

In addition, the figures presented below visually portray the obtained results, providing a graphical representation of the hacking time (HT) needed to initiate an attack on each cryptosystem, measured in seconds. This hacking time pertains to the duration taken to resolve either the integer-factorization problem or the discrete-logarithm problem.

Table 1. Time required (in seconds) to compromise RSA, ElGamal and the Combined RSA-ElGamal scheme through hacking attempts. This figure illustrates the comparative performance of each encryption scheme based on its respective vulnerability to attacks.

| Data Size | RSA | ElGamal | Combined RSA-ElGamal Scheme |
|---|---|---|---|
| 1 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 |
| 6 | 0 | 0.015 | 0.016 |
| 7 | 0 | 0.015 | 0.016 |
| 8 | 0.031 | 0.109 | 0.125 |
| 9 | 0.047 | 0.015 | 0.062 |
| 10 | 0.14 | 0.032 | 0.172 |
| 11 | 0.297 | 0.86 | 1.125 |
| 12 | 2.797 | 2.36 | 5.157 |
| 13 | 5.359 | 10.5 | 16.25 |
| 14 | 11.016 | 6.89 | 18.047 |
| 15 | 60.922 | 117.016 | 168.5 |
| 16 | 307.61 | 213.563 | 519.218 |
| 17 | 661.328 | 8826.2 | 9737.52 |
| 18 | 2635.02 | 23808.9 | 26482.3 |
| 19 | 10993.7 | 71228.1 | 82294 |

### 5.2.4 Observations

The analysis of Figures 1, 2 and 3 reveals an interesting observation regarding the impact of data size on the time required to attack the combined RSA-ElGamal scheme in comparison to the classical RSA and ElGamal schemes. Initially, when the data size is relatively small, there is no noticeable difference between the three schemes. However, as the data sizes increase, significant differences arise, with the time required to attack the combined RSA-ElGamal scheme surpassing that of the RSA and ElGamal schemes by several thousands of seconds. Furthermore, Figure 4 provides

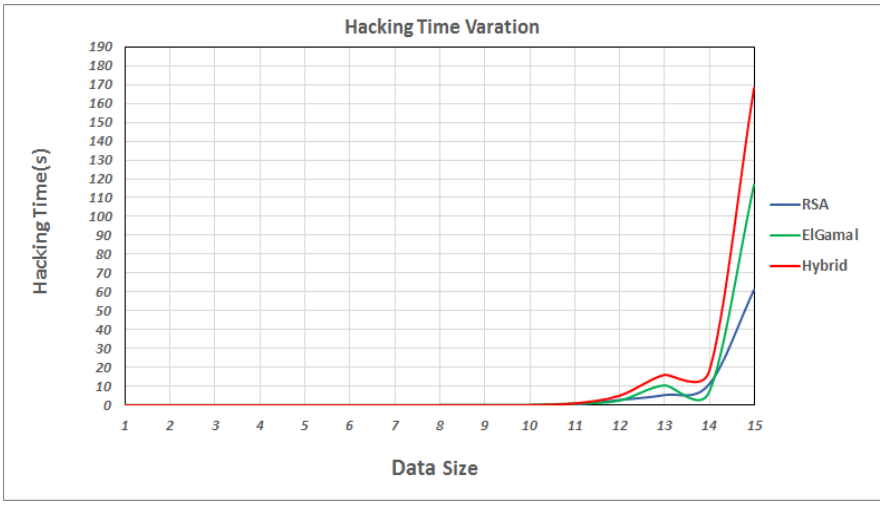


Figure 1. Comparison of performance and efficiency between the classical RSA and ElGamal cryptosystems and the innovative combined RSA-ElGamal cryptosystem, with data sizes ranging from $10^1$ to $10^{15}$ (approximately 4 to 50 bits). This figure illustrates how each system performs across different data sizes.

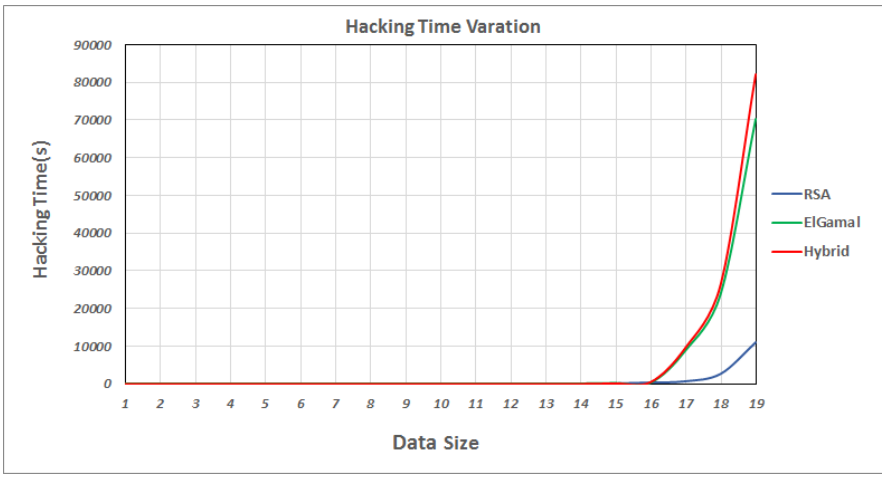


Figure 2. Comparison of performance and efficiency of the classical RSA and ElGamal cryptosystems *versus* the innovative combined RSA-ElGamal cryptosystem across data sizes ranging from $10^1$ to $10^{19}$ (approximately 4 to 67 bits).

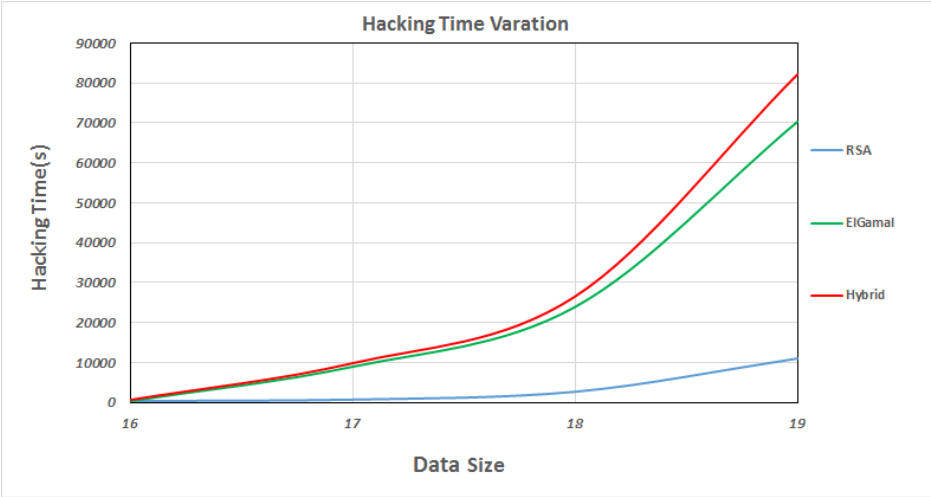


Figure 3. Comparison of performance and efficiency of the classical RSA and ElGamal cryptosystems with the innovative combined RSA-ElGamal cryptosystem across data sizes ranging from $10^{16}$ to $10^{19}$ (approximately 54 to 64 bits).

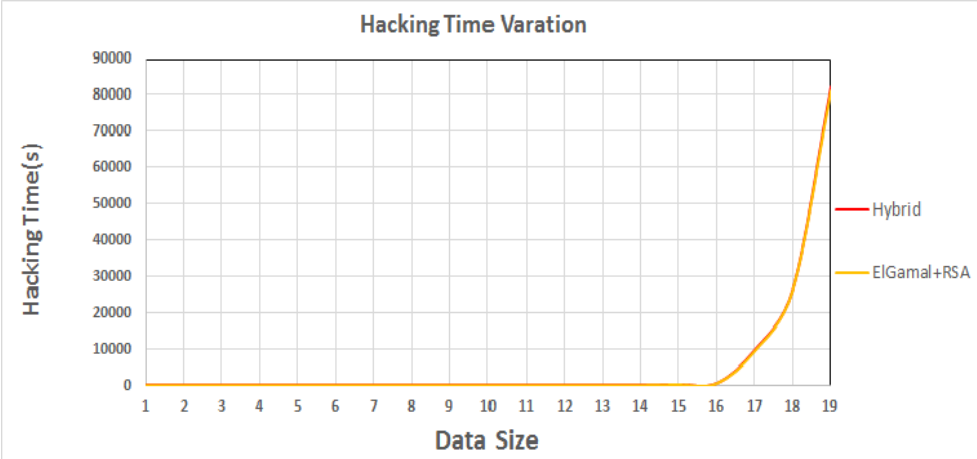


Figure 4. Analysis of the combined RSA-ElGamal algorithm's hacking time (HT) compared to the combined hacking time of the classical RSA and ElGamal algorithms. This figure illustrates the performance differences between the combined RSA-ElGamal approach and the sum of individual classical algorithms' hacking times.

additional insight, demonstrating that the attacking time of the new scheme is notably greater and equivalent to the cumulative attacking times of the classical RSA and ElGamal cryptosystems combined. Based on this evidence, we can deduce that the combined RSA-ElGamal scheme offers enhanced security compared to both the classical RSA and ElGamal schemes. In summary, the analysis showcases the advantage of the combined RSA-ElGamal scheme, highlighting its resistance to attacks as the data size grows larger. The substantial increase in attacking time for the combined RSA-ElGamal scheme, compared to the classical RSA and ElGamal schemes, suggests its heightened level of security and reinforces its suitability for cryptographic applications.

## 5.3 Complexity Analysis

In the following sub-sections, we provide a comparative analysis of the complexity of the RSA, ElGamal and the proposed combined RSA-ElGamal scheme algorithms. Time complexity of an algorithm is commonly expressed using the asymptotic notation of O(n), which is determined by counting the number of basic operations performed during the algorithm's execution, such as addition, subtraction, multiplication and division. The space complexity of a cryptographic algorithm refers to the amount of memory required for the algorithm to run, relative to the length of its input. Space complexity depends on the size of the input. When considering the maximum complexity for a given input size, it is referred to as worst-case complexity. Conversely, when considering the average complexity across all inputs of a given size, it is known as the expected complexity.

### 5.3.1 Complexity of the RSA Scheme

Complexity of key generation: To generate the key, the complexity of selecting random primes $p$ and $q$ and computing their product $n = pq$ is either $O(log_2^2 p)$ using the Fermat's primality test or $O(log_2^3 p)$ using the Miller-Rabin test. Computing $n=pq$ in the domain of natural integers, $Z$, has a complexity of $O(\log_2 p \log_2 q) \approx O(log_2^2 p)$ since $p < q$. Computing Euler's totient function $\phi(n) = (p-1)(q-1)$ has a complexity of $O(\log_2 p \log_2 q) ) \approx O(log_2^2 p)$, since $p < q$. The complexity of selecting a random number $e$ such that $0<e<\phi(n)$ with $(e, \phi(n)) = 1$ using Euclidean division is $O(log_2^3 \phi(n)) \approx O(log_2^3 pq) \approx O(log_2^3 n)$. Thus, the overall time complexity of the key-generation process is $O(log_2^3 n)$.

Complexity of the encryption process: The complexity of computing $c=m^e \pmod n$ is $O(log_2^3 n)$, since the size of e is proportional to that of n.

Complexity of the encryption process: The complexity of computing $m=c^d \pmod n$ is $O(log_2^3 n)$, since the size of e is proportional to that of $n$.

### 5.3.2 Complexity of ElGamal Scheme

Complexity of key generation: Selecting a random prime number $p$ has a complexity of either $O(log_2^2 p)$ if the Fermat's primality test is used or $O(log_2^3 p)$ if the Miller-Rabin test is used. Selecting a value $k$ between 2 and $p-2$ has a complexity of $O(\log_2 p)$. Finding a generator $\alpha$ of the multiplicative group $Z_p^*$has a complexity of $O(log_2^2 p)$. Computing $\alpha^a(\text{mod } p)$ has a complexity of $O(log_2^3 p)$. The overall complexity of key generation is determined to be $O(log_2^3 p)$.
Complexity of the encryption process: Computing $\gamma \equiv \alpha^k (mod\, p)$ has a complexity of $O(log_2^3 p)$. Computing $d \equiv (\alpha^k m)(mod\, p)$ has a complexity of $O(log_2^2 p)$. The overall complexity of the encryption process is $O(log_2^3 p)$.
Complexity of the decryption process: Computing $\gamma^{p-a-1} \equiv \alpha^{-ak} (mod\, p)$ using the extended Euclidean algorithm has a complexity of $O(log_2^3 p)$. Computing $m \equiv \alpha^{-ak}\delta (mod\, p)$ has a complexity of $O(log_2^2 p)$. The overall complexity of the decryption process is $O(log_2^3 p)$.

### 5.3.3 Complexity of the Combined RSA-ElGamal Scheme

Key-generation Complexity: Picking random primes $p$, $q$ and $r$ has a complexity of $O(log_2^2 p)$ if the Fermat's primality test is used or $O(log_2^3 p)$ if the Miller-Rabin test is used. Selecting a value $k$ such that $2 \leq k \leq p^2 - 1$ has a complexity of $O(2log_2 p)$. Finding a generator $\alpha$ of the group $G_p^*$ has a complexity of $O(log_2^2 p^2)$ using a specific algorithm. Computing $\alpha^a \pmod p$ has a complexity of $O(log_2^3 p)$. Computing $\phi(\eta) = (q^2 - 1)(r^2 - 1)$ has a complexity of $O(log_2^4 p)$. Selecting a value $e$ such that $1 \leq e \leq \phi(\eta)$ and $\gcd(e, \phi(\eta)) = 1$ has a complexity of $O(log_2^3 \eta)$. Computing $d \equiv e^{-1} \pmod{\phi(\eta)}$ has a complexity of $O(log_2^3 \phi(\eta)) = O(log_2^3 \eta^2)$ using the extended Euclidean algorithm. Hence, the overall complexity of key generation is $O(log_2^4 q)$.

Encryption-process Complexity: Computing $\gamma \equiv \alpha^k \pmod p$ has a complexity of $O(log_2^3 p)$. Computing $\delta \equiv (\alpha^k m) \pmod p$ has a complexity of $O(log_2^3 p)$. Computing $c \equiv (\gamma + \delta i)^e \pmod \eta$ has a complexity of $O(log_2^3 \eta)$. Hence, the overall complexity of the message-encryption process is $O(log_2^3 p)$.

Decryption-process Complexity: Computing $M \equiv c^d \pmod \eta$ has a complexity of $O(log_2^3 \eta)$. Computing $f \equiv Re(M)^{p^2-a-1} \pmod p$ has a complexity of $O(log_2^2 p)$. Computing $h \equiv \text{Im}(M) \pmod p$ has a complexity of $O(\log_2 p)$. Computing $t \equiv fh \pmod \eta$ has a complexity of $O(log_2^2 \eta)\eta)$. Hence, the overall complexity of the message-decryption process is $O(log_2^3 \eta)$.

In summary, our analysis has thoroughly examined the computational complexities inherent in three prominent cryptographic schemes: RSA, ElGamal and the combined RSA-ElGamal schemes. We assessed these complexities in terms of key generation, encryption and decryption operations. For RSA, the key generation, encryption and decryption exhibit a time complexity of approximately $O(log_2^3 n)$, leveraging efficient key generation, but demanding multiple modular exponentiations for encryption and decryption. Conversely, ElGamal scheme demonstrates a similar time complexity of $O(log_2^3 p)$ for key generation, encryption and decryption, excelling in key generation and encryption processes while requiring an additional modular exponentiation during decryption. The combined RSA-ElGamal scheme combines the strengths of both RSA and ElGamal schemes, featuring key generation complexity of $O(log_2^4 p)$and encryption complexity of $O(log_2^3 p)$, akin to ElGamal scheme. However, decryption complexity increases to $O(log_2^3 \eta)$, reflecting a slight trade-off for the inclusion of RSA's capabilities. In conclusion, the combined RSA-ElGamal scheme offers a balanced approach, leveraging RSA's advantages in key management alongside ElGamal's encryption efficiency, with the choice of scheme dependent on specific security needs and computational considerations.

## 5.4 Real-world Applicability

### 5.4.1 Impact of Key Generation on Overall Performance

The combined RSA-ElGamal scheme presents a balance between enhanced security and computational efficiency. As discussed in our analysis, the key-generation process for this novel scheme is inherently more complex and time-intensive compared to the individual RSA or ElGamal

schemes. This complexity arises from the need to generate and manage three distinct prime numbers and perform additional arithmetic operations within the domain of Gaussian integers.

In practical applications, the extended key-generation time can affect the performance of systems that rely on frequent key rotations or require rapid key creation. For instance, in scenarios such as secure communications or real-time applications where keys are generated dynamically, the increased key-generation time might impact the system responsiveness. To address this issue, future research could explore optimizing key-generation processes or leveraging parallel-computing techniques to reduce the required time.

### 5.4.2 Integration into Existing Security Frameworks

The combined RSA-ElGamal scheme can be integrated into existing security frameworks with minimal disruption. Its dual-layer approach enhances security while requiring only minor adjustments to the existing cryptographic infrastructure. Key points include:

1. **Backward Compatibility:** The combined RSA-ElGamal scheme can be deployed alongside existing RSA or ElGamal systems, facilitating gradual adoption. This allows organizations to apply the new approach to new applications or incrementally transition from older systems.
2. **Modular Integration:** The architecture of the combined RSA-ElGamal scheme supports modular integration into existing security frameworks. It can be incorporated into established protocols, such as TLS or VPNs, either as a replacement or a complement to existing encryption algorithms, thereby enhancing security without necessitating an overhaul of the entire system.
3. **Adaptability for Specific Use Cases:** The flexibility in the combined RSA-ElGamal scheme's parameter choices enables customization for specific security needs. For example, in environments with stringent security requirements, the scheme's enhanced resistance to attacks can be particularly advantageous.
4. **Compatibility with Modern Hardware:** Given that the computational demands of the new scheme are manageable with current hardware, it can be effectively utilized in both software-based and hardware-accelerated cryptographic systems.

In summary, while the combined RSA-ElGamal scheme introduces additional computational overhead, it offers significant security benefits that can be applied to various real-world scenarios. By carefully considering the impact of key generation and thoughtfully integrating the scheme into existing frameworks, its advantages can be maximized and potential performance challenges can be mitigated.

## 5.5 Practical Limitations

### 5.5.1 Increased Computation Time

While the theoretical analysis of computational complexities provides a foundation, practical implementations often encounter additional challenges. The combined RSA-ElGamal scheme, due to its combined use of RSA and ElGamal schemes, involves complex operations that can impact performance:

1. **Key Generation:** The key-generation process for the combined RSA-ElGamal scheme requires generating three primes and performing additional arithmetic operations within the domain of Gaussian integers. This complexity can lead to significantly longer key generation times compared to RSA and ElGamal schemes individually. This extended time might affect systems that require frequent key updates or rapid key generation, such as secure-communication systems and real-time applications.
2. **Encryption and Decryption:** Although the encryption process for the combined RSA-ElGamal scheme shows comparable time requirements to traditional methods, the combined computational steps from both RSA and ElGamal schemes can lead to longer processing times in practical scenarios. For instance, the combined RSA-ElGamal scheme involves modular exponentiations and additional arithmetic operations that could contribute to a slower overall encryption and decryption process.

### 5.5.2 Memory Requirements

The combined RSA-ElGamal scheme's increased complexity also impacts memory usage:

1. **Storage for Intermediate Results:** The computations involved in key generation and encrypttion/decryption require storing intermediate results, which can increase memory usage. For instance, handling large integers and matrices in Gaussian-integer arithmetics can lead to higher memory demands compared to simpler encryption schemes.
2. **Ciphertext Size:** As discussed, the new scheme may result in ciphertexts that are larger due to the inclusion of both real and imaginary components. This increase in ciphertext size can impact storage requirements and bandwidth, particularly in systems with limited resources.

### 5.5.3 Mitigation Strategies

To address these practical limitations, several strategies can be considered:

1. **Algorithm Optimization:** Future research could focus on optimizing the combined RSA-ElGamal scheme's algorithm to reduce computation time and memory usage.
2. **Hardware Acceleration:** Implementing hardware acceleration for cryptographic operations could help manage the increased computational load and memory requirements, making the scheme more feasible for practical applications.
3. **Efficient Storage Solutions:** Exploring efficient storage and management solutions for intermediate results and ciphertexts could help mitigate memory overhead.

### 5.5.4 Advantages and Disadvantages

**Advantages:** The combined RSA-ElGamal scheme offers several significant advantages:

1. **Integration of RSA and ElGamal Schemes:** By combining RSA and ElGamal encryption schemes, the new approach delivers a dual-layered security solution. The first layer leverages the discrete-logarithm problem, while the second layer relies on the integer-factorization problem. This combination creates a robust encryption framework similar to a double onion routing shield.
2. **Smooth Implementation:** Implementing the combined RSA-ElGamal scheme requires no additional effort beyond what is expected with traditional encryption schemes. The transition to the new approach can be smoothly achieved without introducing added complexities or burdensome requirements.
3. **Expanded Parameter Range:** With computational efforts comparable to traditional encryption methods, the combined RSA-ElGamal scheme allows for a broader selection of plaintexts and private keys. In fact, the number of available options exceeds the square of those in classical encryption settings, offering increased flexibility for customized-encryption processes.
4. **Efficient Encryption and Decryption:** The combined RSA-ElGamal scheme maintains comparable time requirements for encryption and decryption processes relative to traditional encryption methods. No extra time is required for these operations, ensuring that the new approach remains efficient and practical.
5. **Enhanced Security:** The combined RSA-ElGamal scheme significantly increases resistance to attacks compared to traditional encryption methods. The time required for an attacker to compromise the combined RSA-ElGamal scheme is substantially greater than or equal to the combined time needed to break both underlying classical encryption schemes.

**Disadvantages:** Despite its strengths, the combined RSA-ElGamal scheme has a few drawbacks that must be considered:

1. **Increased Ciphertext Length:** In certain cases, the ciphertext generated by the combined RSA-ElGamal scheme may be twice the length of the original message. This occurs when the plaintext is real, resulting in a complex-number ciphertext that includes both real and imaginary components. The increased ciphertext length may impact storage requirements or communication bandwidth, which is an important consideration in resource-constrained

environments.

2. **Computational Overhead:** The combined RSA-ElGamal scheme involves extensive computations, particularly when dealing with large logarithmic calculations. This computational burden can result in longer processing times, potentially affecting the feasibility of encryption processes, especially when handling large datasets. The algorithm may perform sluggishly when encrypting extensive data on a single machine, necessitating optimization or the use of distributed-computing strategies.
3. **Slower Key Generation:** Key generation in the combined RSA-ElGamal scheme is slower compared to RSA and ElGamal schemes. This is because the combined RSA-ElGamal scheme requires the generation of three primes (as opposed to one in ElGamal or two in RSA) and additional computations. Moreover, arithmetic operations within the domain of Gaussian integers add further computational requirements, depending on the specific forms of the selected Gaussian primes. As a result, key generation in the combined RSA-ElGamal scheme demands more time and more computational resources.

Therefore, the proposed combined RSA-ElGamal encryption scheme combines the strengths of RSA and ElGamal schemes while avoiding excessive implementation complexities. However, careful consideration of the potential expansion of ciphertext length, computational overhead and slower key-generation processes is essential. These factors should be evaluated to determine the suitability of the combined RSA-ElGamal scheme for various cryptographic applications, taking into account the specific requirements and constraints of the intended use cases.

## 6. CONCLUSION

In summary, our investigation into the combined RSA-ElGamal scheme highlights its effectiveness in striking a balance between enhanced security and computational efficiency. Despite the inherent complexity in key generation compared to standalone RSA or ElGamal schemes, the new approach delivers significant security benefits that warrant this added complexity. When compared to other advanced cryptographic methods, such as elliptic curve cryptography (ECC) and lattice-based cryptography, the combined RSA- ElGamal scheme presents a distinctive combination of security advantages and practical utility.

Our study emphasizes the theoretical robustness of this new approach. Future work will focus on several key areas to further enhance and validate the scheme. We plan to provide a comprehensive analysis of the cryptanalysis methods used, including specific algorithms and their implementations. We will also include comparative studies with recent cryptographic techniques to contextualize our results and demonstrate the scheme’s relative efficacy. Additionally, optimizing key generation processes, evaluating performance in various real-world scenarios and integrating the combined RSA-ElGamal scheme with existing security frameworks will be pivotal. Exploring variants and extensions of the combined RSA-ElGamal scheme will offer deeper insights into its practical advantages and limitations, guiding its future development and application.

## ACKNOWLEDGEMENTS

We extend our profound gratitude to the editor and the referees for their meticulous review and invaluable feedback. Their expert insights and constructive comments have been instrumental in refining the manuscript and enhancing its quality. We deeply appreciate the time and effort dedicated by them to review our work, which has significantly enriched the depth and impact of our research. Thank you for your exceptional contributions and support.

## REFERENCES

[1] T. Adamski and W. Nowakowski, "The Average Time Complexity of Probabilistic Algorithms for Finding Generators in Finite Cyclic Groups," Bulletin of the Polish Academy of Sciences: Technical Sciences, vol. 63, no. 4, pp. 989-996, 2015.

[2] E. A. Adeniyi et al., "Secure Sensitive Data Sharing Using RSA and ElGamal Cryptographic Algorithms with Hash Functions," Information, vol. 13, no. 10, p. 442, 2022.

[3] J. M. Ahmed and Z. M. Ali, "The Enhancement of Computation Technique by Combining RSA and ElGamal Cryptosystems," Proc. of the 2011 Int. Conf. on Electrical Engineering and Informatics, pp. 1-5, DOI: 10.1109/ICEEI.2011.6021779, Bandung, Indonesia, 2011.

[4] A. Aman and R. K. Aggarwal, "A Survey: Analysis of Existing Hybrid Cryptographic Techniques," Proc. of the Int. Conf. on Recent Developments in Cyber Security, Cyber Security and Digital Forensics (REDCYSEC 2023), Part of the Book: Lecture Notes in Networks and Systems, vol. 896, pp. 259-269, Springer, 2023.

[5] S. Anjana et al., "Security-enhanced Optical Nonlinear Cryptosystem Based on Phase-truncated Fourier Transform," Optical and Quantum Electronics, vol. 55, no. 12, p. 1099, 2023.

[6] Y. Awad et al., "Comparative Study between a Novel Deterministic Test for Mersenne Primes and Well-known Primality Tests," Baghdad Science Journal, vol. 20, no. 5 (Suppl.), 2023.

[7] Y. Awad, A. N. El-Kassar and T. Kadri, "Rabin Public-key Cryptosystem in the Domain of Gaussian Integers," Proc. of the 2018 Int. Conf. on Computer and Applications (ICCA), pp. 1-340, DOI: 10.1109/COMAPP.2018.8460338, Beirut, Lebanon, 2018.

[8] M. Bunder, A. Nitaj, W. Susilo and J. Tonien, "A Generalized Attack on RSA Type Cryptosystems," Theoretical Computer Science, vol. 704, pp. 74-81, 2017.

[9] J. J. Cogswell, The Theory of Indices Modulo n, Ph.D. Dissertation, Emporia State Univ., Emporia, KS, 2012.

[10] J. T. Cross, "The Euler ϕ-function in the Gaussian Integers," American Mathematical Monthly, vol. 90, no. 8, pp. 518-528, 1983.

[11] A. A. El-Douh, S. F. Lu, A. Elkony and A. S. Amein, "A Systematic Literature Review: The Taxonomy of Hybrid Cryptography Models," Proc. of Future of Information and Communication Conf., Advances in Information and Communication (FICC 2022), Part of the Book Series: Lecture Notes in Networks and Systems, vol. 439, pp. 714-721, Springer International Publishing, 2022.

[12] T. ElGamal, "A Public-key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," IEEE Transactions on Information Theory, vol. 31, no. 4, pp. 469-472, 1985.

[13] A. N. El-Kassar, R. A. Haraty, Y. A. Awad and N. C. Debnath, "Modified RSA in the Domains of Gaussian Integers and Polynomials over Finite Fields," Proc. ISCA 18th Int. Conf. Comput. Appl. Ind. Eng. (CAINE), pp. 298-303, Honolulu, USA, 2005.

[14] A. N. El-Kassar, R. Haraty and Y. Awad, "Rabin Public-key Cryptosystem in Rings of Polynomials over Finite Fields," Proc. of the Int. Conf. on Computer Science, Software Engineering, Information Technology, e-Business and Applications (CSITeA'04), Cairo, Egypt, 2004.

[15] A. N. El-Kassar and S. Habre, "Greatest Common Divisor and Least Common Multiple Matrices on Factor Closed Sets in a Principal Ideal Domain," J. of Mathematics and Statistics, vol. 5, no. 4, pp. 342-347, 2009.

[16] A. A. Emmanuel, A. E. Okeyinka, M. O. Adebiyi and E. O. Asani, "A Note on Time and Space Complexity of RSA and ElGamal Cryptographic Algorithms," Int. J. of Advanced Computer Science and Applications, vol. 12, no. 7, pp. 143-147, 2021.

[17] S. M. Hardi, J. T. Tarigan and N. Safrina, "Hybrid Cryptosystem for Image File Using ElGamal and Double Playfair Cipher Algorithm," Journal of Physics: Conference Series, IOP Publishing, vol. 978, no. 1, p. 012068, 2018.

[18] J. Hoffstein, "Integer Factorization and RSA," Proc. of An Introduction to Mathematical Cryptography, Part of the Book: Undergraduate Texts in Mathematics (UTM), pp. 1-75, New York, USA, 2008.

[19] K. S. Gaur et al., "Cryptanalysis of the Optical Cryptosystem Titled 'An Asymmetric Image Encryption Based on Phase Truncated Hybrid Transform'," Journal of Optics, vol. 53, pp. 605-609, 2023.

[20] C. F. Gauss, "The Arithmetic of the Gaussian Integers," [Online], Available: https://personal.math.ubc.ca/~anstee/math444/GaussianIntegersfinal.pdf, 2020.

[21] R. Imam et al., "Systematic and Critical Review of RSA Based Public- key Cryptographic Schemes: Past and Present Status," IEEE Access, vol. 9, pp. 155949-155976, 2021.

[22] N. M. S. Iswari, "Key Generation Algorithm Design Combination of RSA and ElGamal Algorithm," Proc. of the 2016 8th IEEE Int. Conf. on Information Technology and Electrical Engineering (ICITEE), pp. 1-5, Yogyakarta, Indonesia, 2016.

[23] M. Iavich et al., "Hybrid Encryption Model of AES and ElGamal Cryptosystems for Flight Control Systems," Proc. of the 5th IEEE Int. Conf. on Methods and Systems of Navigation and Motion Control (MSNMC), pp. 229-233, Kiev, Ukraine, 2018.

[24] P. Kuppuswamy and S. Q. Al-Khalidi, "Hybrid Encryption/Decryption Technique Using New Public-key and Symmetric Key Algorithm," Int. J. of Information and Computer Security, vol. 6, no. 4, pp. 372-382, 2014.

[25] A. J. Menezes et al., Handbook of Applied Cryptography, CRC Press, 2018.

[26] M. Mohammadi, A. Zolghadr and M. Pourmina, "Comparison of Two Public-key Cryptosystems," Journal of Optoelectronical Nanostructures, vol. 3, no. 3, pp. 47-58, 2018.

[27] B. Molelekeng, Arithmetic in the Ring of Gaussian Integers, Ph.D. Dissertation, University of the Witwatersrand, Johannesburg, 2022.

[28] P. K. Panda and S. Chattopadhyay, "A Hybrid Security Algorithm for RSA Cryptosystem," Proc. of the

2017 4th IEEE Int. Conf. on Advanced Computing and Communication Systems (ICACCS), pp. 1-6, Coimbatore, India, 2017.

[29] J. M. Parenreng and A. Wahid, "The E-mail Security System Using El-Gamal Hybrid Algorithm and AES (Advanced Encryption Standard) Algorithm," Internet of Things and Artificial Intelligence J., vol. 2, no. 1, pp. 1-9, 2022.

[30] K. S. Patil, I. Mandal and C. Rangaswamy, "Hybrid and Adaptive Cryptographic-based Secure Authentication Approach in IoT Based Applications Using Hybrid Encryption," Pervasive and Mobile Computing, vol. 82, p. 101552, DOI: 10.1016/j.pmcj.2022.101552, 2022.

[31] I. S. Permana, T. Hidayat and R. Mahardiko, "Raw Data Security by Using ElGamal and SHA 256 Public-key Algorithm," Teknokom, vol. 4, no. 1, pp. 1-6, 2021.

[32] R. L. Rivest, A. Shamir and L. Adleman, "A Method for Obtaining Digital Signatures and Public-key Cryptosystems," Communications of the ACM, vol. 21, no. 2, pp. 120-126, 1978.

[33] L. B. Rivera et al., "Hybrid Cryptosystem Using RSA, DSA, ElGamal and AES," Int. Journal of Scientific & Technology Research, vol. 8, no. 10, pp. 1777-1781, 2019.

[34] A. P. U. Siahaan, B. O. Elviwani and B. Oktaviana, "Comparative Analysis of RSA and ElGamal Cryptographic Public-key Algorithms," Proc. of the Joint Workshop KO2PI and the 1st Int. Conf. on Advance & Scientific Innovation (ICASI'18), pp. 163-172, DOI: 10.4108/eai.23-4-2018.2277584, 2018.

[35] H. Singh, R. Girija and M. Kumar, "A Cryptoanalysis of Elliptic Curve Cryptography Based on Phase Truncation in the Domain of Hybrid Gyrator Hartley Transform," Optical and Quantum Electronics, vol. 55, no. 6, p. 487, 2023.

[36] N. Tahat et al., "A New RSA Public-key Encryption Scheme with Chaotic Maps," Int. Journal of Electrical and Computer Engineering (IJECE), vol. 10, no. 2, pp. 1430-1437, 2020.

[37] Q. Zhang, "An Overview and Analysis of Hybrid Encryption: The Combination of Symmetric Encryption and Asymmetric Encryption (CDS)," Proc. of the 2nd IEEE Int. Conf. on Computing and Data Science (CDS), pp. 616-622, DOI: 10.1109/CDS52072.2021.00111, Stanford, USA, Jan. 2021.

## ملخص البحث:

تُقدّم هذه الورقة منهجيةً مبتكرةً تجمع بين خوارزمية RSA وخوارزمية الجَمَل لتحسين أمان وفاعلية أنظمة التّشفير ذات المفاتيح العامّة. ومن خلال الجمْع بين الخوارزميتين المذكورتين، فإنّ طريقتنا تجمع بين الاستفادة من نقاط القوّة والحدّ من نقاط الضّعْف في الأنظمة التقليدية، وبخاصّةٍ فيما يتعلق بمشكلة عَومَلة الأعداد الصّحيحة ومشكلة اللّوغاريثم المجرّد. ويعمل استخدام أعداد غاوْسْ الصّحيحة على تعزيز متانة عمليات التّشفير والتّوقيع الرّقمي، موفّراً إطارَ عملٍ أكثر أماناً. وتتضمّن دراستنا تحليلاً شاملاً للخوارزميتين اللّتين يتمّ الجمع بينهما، إلى جانب تطبيقاتٍ عمليةٍ وتقييمات لأنظمة التّشفير الّتي تُصمَّم بالمنهجية المبتكرة في هذه الدّراسة مع التركيز على التّحدّيات المتمثّلة في عَومَلة الأعداد الصّحيحة ومشكلة اللّوغاريثم المجرّد. كذلك تمّ إجراء تقييماتٍ للوقوف على جودة النّظام المقترح وفاعليته الحسابية.

فبينما كان توليد المفاتيح أبطأ مقارنةً باستخدام خوارزمية RSA أو خوارزمية الجَمَل على انفراد، فإنّ المنهجية المقترحة تنِمُّ عن أداءٍ جيّدٍ في التّشفير وإزالة التّشفير. وعلى العكس من الأنظمة المشابهة في دراسات أخرى الّتي تركّز على معالجة الصُّوَر الضوئية، فإنّ دراستنا تعملُ على توسيع نطاق أنظمة التّشفير والتّوقيع الرّقمي إلى تطبيقات أكثر، ساعيةً بذلك إلى تحسين الجانب النّظري المتعلّق بموضوع الدّراسة وتطوير التّطبيقات العملية لأنظمة التّشفير والتّوقيع الرّقمي. والجدير بالذّكر أنّ البحث المستقبلي حَوْل موضوع الدراسة سيركّز على توليد المفاتيح بطرقٍ أكثر مثاليةً، واستكشاف دمج النّظام المقترح في أُطُر العمل القائمة المتعلّقة بأمان الأنظمة، وتقييم الأداء في ظلّ سيناريوهاتٍ من العالم الحقيقي؛ وذلك من أجْل تحسين أداء النّظام المقترح والتّحقُّق من نجاعته.

This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).