

Article

A Number-Theoretic Generalization of the ElGamal Cryptosystem with Applications to Digital Signatures and Image Encryption

Hayder R. Hashim 

Faculty of Computer Science and Mathematics, University of Kufa, P.O. Box 21, Al Najaf 54001, Iraq;
hayderr.almuswi@uokufa.edu.iq

Abstract

In this paper, we propose an image encryption and digital signature scheme based on a modification of ElGamal cryptosystem over a large modulus represented by $2p^n$, where p is a large prime number and $n > 2$ is a dynamic exponent derived from a shared secret established by the communicating participants through a modification of the Diffie–Hellman Key Exchange Protocol. The proposed scheme combines a modified ElGamal framework with image encryption and digital signature mechanisms to support confidentiality, image authentication and integrity verification while preserving the standard security assumptions of discrete-logarithm-based cryptography. Empirical performance and image-statistical evaluations, including histogram analysis, entropy, correlation coefficients, NPCR and UACI, are conducted on applying the procedures of the proposed scheme on four standard images. The experimental results demonstrate the correctness and favorable empirical image-statistical behavior of the proposed scheme for grayscale image encryption and authentication under the classical DLP assumptions. Furthermore, comparisons with representative symmetric and asymmetric cryptographic schemes are provided to evaluate the performance of the proposed scheme.

Keywords: cryptography; public-key cryptosystem; image encryption; ElGamal cryptosystem; signature scheme; prime numbers; primitive roots; Hash function; Diffie–Hellman key exchange

MSC: 94A60; 94A08; 11T71; 11Z05; 11A41; 11A07

1. Introduction

The advent of technology and the need of secure communication in digital platforms have made the protection of images an important area, especially in military, healthcare and online communication. Because of the unique properties of image data including its redundancy, large size and pixel correlation, many cryptanalysis techniques have been invented to attack many well-known public-key cryptosystems such as the ElGamal cryptosystem, which is introduced in 1985 [1] by Taher ElGamal. The security of this cryptosystem is based on the computational difficulty in solving the discrete logarithm problem (DLP) in finite cyclic groups, namely it concentrates on finding an integer d satisfying the equation

$$g^d \equiv a \pmod{p},$$



Received: 30 June 2026

Revised: 29 July 2026

Accepted: 31 July 2026

Published: 3 August 2026

Copyright: © 2026 by the author.

Licensee MDPI, Basel, Switzerland.

This article is an open access article

distributed under the terms and

conditions of the [Creative Commons](#)

[Attribution \(CC BY\)](#) license.

where g and a belong to the group $\mathbb{Z}_p^*$. This problem provides a robust public-key encryption scheme such as ElGamal cryptosystem, that has become an essential system used in securing digital communications. Another computational problem widely used in designing public-key cryptosystems is the integer factorization problem that focuses on factoring an integer into its product of large primes. Many public-key cryptosystems were designed using this problem such as the RSA cryptosystem and its variants, see e.g., [2–4].

One of the useful features of the ElGamal cryptosystem is its support for a data digital signature, which is a cryptographic method used to authenticate messages between the communicating parties to ensure non-repudiation and the integrity and of the transmitted data. This feature makes the ElGamal cryptosystem suitable for various security applications.

Indeed, the ElGamal cryptosystem is considered as an extension of the ideas in Diffie-Hellman Key Exchange Protocol (DHKEP), which was introduced in 1976 by Diffie and Hellman [5] and was considered as one of the earliest methods used for securely exchanging cryptographic keys over a public channel. With this method, two parties can exchange a key that can be used for public-key encryption without directly transmitting the key itself. Similarly, the security of this scheme depends on the DLP, and it can be defined as follows. Suppose that p is prime and g is a generator of $\mathbb{Z}_p^*$, which both are known for everyone, then a secret key can be exchanged by Alice and Bob as follows:

- Alice and Bob separately select their private keys d_1 and d_2 from the group $\mathbb{Z}_p^*$, respectively. Then they respectively compute their public keys $A = g^{d_1} \pmod{p}$ and $B = g^{d_2} \pmod{p}$.
- After receiving each other public keys, each participant uses his or her private key to compute the shared secret $K = B^{d_1} = A^{d_2} \equiv g^{d_1 d_2} \pmod{p}$.

In the case of image encryption, the ElGamal cryptosystem has been adapted to successfully handle the structured of the image data. However, this cryptosystem has good security due to the hardness of solving the discrete logarithm problem, it encounters many attacks. Therefore, many researchers have proposed enhancements, modifications and variations of the original scheme to gain better performance and security. For instance, some of these improvements include applying the scheme to other mathematical groups such as elliptic curves (EC) and multiplicative groups of modular integers. In 1987, Koblitz [6] proposed an improvement of the ElGamal cryptosystem that is based on EC and its security depends on the hardness of solving the DLP defined on an abelian group formed by points on elliptic curves. In fact, many techniques of image encryption based on elliptic curve cryptography have been proposed by many authors. For instance, a color image encryption method based on the combination of a 2D discrete chaotic map and Menezes Vanstone elliptic curve cryptography was proposed by Liu and et al. in [7]. Another cryptographic technique for image encryption based on combining DNA encoding and elliptic curve Diffie-Hellman key exchange protocol was proposed by Kumar and et al. in [8]. Moreover, Luo and et al. [9] proposed a scheme for image encryption based on ElGamal-ECC and chaotic theory in which they used the hash function, SHA-512, to generate the initial values of a chaotic system. Another enhancement of the ElGamal cryptosystem based on elliptic curves and high-dimensional chaotic map was introduced by Wu and et al. in [10], which was applied on encrypting color images. Last but not least, Li and et al. [11] suggested an enhancement of the ElGamal cryptosystem by using the additive homomorphism of Elliptic Curve ElGamal cryptosystem. For more improvements to the ElGamal cryptosystem, we refer to [12–14] and the references given there.

In order to achieve high computational efficiency in image encryption, symmetric image encryption algorithms are widely used since both communicating parties share and manage a secret key before any data can be exchanged. But in many real world applications, especially those with open or distributed networks, distributing keys between

communicating parties can be difficult and risky. To avoid this limitation, it is generally safer to employ asymmetric cryptographic schemes. Therefore, with the advent of technology and the existence of quantum computing, increasing the security of asymmetric cryptosystems such as ElGamal cryptosystem has become more important in order to avoid any attacks. Furthermore, it is known that the image encryption technique effectively protects the confidentiality of image data during transmission, but it does not guaranty the authenticity or integrity of the transmitted image since an attacker may replace an encrypted image without the recipient being able to detect the tampering. Therefore, to enable authentication and integrity verification of the recovered plaintext image after decryption, it is very important to incorporate a digital signature mechanism into the proposed image encryption scheme. Therefore, in this paper we propose a modification of the ElGamal cryptosystem and its signature scheme by using the multiplicative group of units modulo $2p^n$ with a large prime number p and $n > 2$ is a dynamic exponent derived from the shared secret established by the communicating parties through a modified Diffie–Hellman Key Exchange Protocol (MDHKEP). More precisely, this scheme is based on modifying the algebraic mathematical structures of the encryption, decryption and digital signature scheme in the classical algorithms by using a larger modulus $2p^n$ whose exponent $n > 2$ is derived from a secret key shared by the communicating parties using the MDHKEP, that allows the proposed scheme to employ session-dependent system parameters without modifying the security assumptions of the underlying discrete logarithm.

To be more precise, before the encryption procedure, the communicating parties compute the shared secret using the MDHKEP, from which the dynamic exponent n is derived. The corresponding modulus $2p^n$ is then independently constructed by the communicating parties and used in the proposed encryption, decryption and digital signature algorithms.

Therefore, the security of the proposed scheme firstly depends on the computational difficulty in solving the underlying discrete logarithm problem, which protects the cryptosystem's private keys. Furthermore, since the exponent n is dynamically derived from a shared Diffie–Hellman secret, thus determining the corresponding session-derived modulus $2p^n$ requires recovering the shared secret whose security relies on the computational hardness of the discrete logarithm problem on which the MDHKEP is based.

On the other hand, there is a one-way function that sends data with any size to a fixed size value (called by a hash or digest) called by a hash function, which is really important in cryptography for different reasons. First, it is considered as a kind of signature for a text or a data file. Moreover, finding collisions should be computationally infeasible for a secure hash function. In fact, there are various types of such functions such as the cryptographic hash function SHA-256, which generates a unique signature given by a hash value of 256-bits for a data file that has 64 hexadecimal characters. Therefore, the modification of the digital signature scheme of the proposed scheme includes the hash function (SHA-256) in the digital signature scheme.

In fact, this scheme is designed to encrypt grayscale image pixels directly (although, the exact scheme can be applied to color images) using modular operations to ensure confidentiality, while also offering support for digital signatures to verify image authenticity. The performance and empirical image-statistical behavior of the proposed encryption and decryption algorithms are evaluated through a series of empirical performance and image-statistical analyses including image histogram analysis, encrypting and decryption timing, MSE (Mean Squares Error), PSNR (Peak Signal-to-Noise Ratio), correlation coefficients, entropy and sensitivity to pixel changes with determining the Number of Pixels Change Rate (NPCR) and the Unified Average Changing Intensity (UACI). Thus, the proposed scheme provides a mathematically motivated approach for grayscale image encryption

and authentication whose correctness and empirical image-statistical performance are demonstrated under the classical DLP assumptions.

The main contributions of this work can be summarized as follows. We propose an image encryption and digital signature scheme based on a modification of ElGamal cryptosystem, where the cryptographic parameters are generated dynamically rather than being fixed. More precisely, the proposed scheme is defined over a large modulus $2p^n$, where the exponent n is dynamically and secretly computed through a Diffie–Hellman key exchange procedure. Moreover, the proposed scheme integrates image encryption and digital signature into a single scheme, together providing confidentiality, integrity, authentication and non repudiation. The main aim of this work is to establish the functional correctness of the construction of the proposed scheme, provide a security discussion under the computational difficulty assumptions of the DLP, and present empirical performance and image-statistical evaluations.

Note that the remainder of this paper is organized as follows. Section 2 describes in detail the algorithms of the encryption, decryption and digital signature scheme of the proposed scheme. Section 3 provides a detailed implementation of the algorithms of the proposed scheme. Section 4 presents the results of experiments conducted on standard grayscale images of size 256×256 . Section 5 discusses the security analysis and parameter considerations of the proposed scheme and the associated parameters used to evaluate the performance of its algorithms. Furthermore, this section provides comparisons with existing methods in terms of computational cost, correlation coefficients, NPCR, UACI, and the entropy of the encrypted and decrypted images. Finally, Section 6 concludes the results of the paper emphasizing the security of the proposed scheme and its potential applications.

2. The Proposed ElGamal Image Encryption and Signature Scheme

This section describes in detail the proposed modification of the ElGamal cryptosystem for image encryption and the generation of digital image signature schemes. The encryption, decryption and signature scheme algorithms are performed with the concept of primitive roots of a large modulus $2p^n$, where p is a large prime and $n > 2$ is computed through the MDHKEP. To be more precise, a positive integer r is called a primitive root modulo a positive integer m if $\gcd(r, m) = 1$ and $\text{ord}_m r = \phi(m)$, where $\text{ord}_m r$ denotes the order of r modulo m and defines the smallest positive integer x such that $r^x \equiv 1 \pmod{m}$, and $\phi(m)$ denotes the Euler phi function that defines the number of positive integers t less than m and $\gcd(t, m) = 1$. For more details on these results, we refer to [15,16]. For later use, we recall the following result represented by Lemma 1, concerning a primitive root modulo p , p^2 , p^n or $2p^n$, which follows from Theorem 8.10 (p. 162) in Burton [17]. Note that the procedures of all these algorithms are presented in Appendix A in a standard pseudocode format.

Lemma 1. *Suppose that r is a primitive root modulo an odd prime number p , then r or $r + p$ is a primitive root modulo p^2 . Furthermore, if r is an odd primitive root modulo p^2 , then it is a primitive root modulo $2p^n$ with $n > 2$.*

2.1. Public Parameters and Keys Generation

- The communicating parties (e.g., Bob and Alice) agree on a large odd prime $p \approx 256, 2048$ or 3072 bits, an odd primitive root r of p and p^2 (which is also a primitive root of $2p^n$ for all $n > 2$ according to the result of Lemma 1), and fixed positive integers n_0 and D .
- Bob and Alice respectively choose the secret keys b_d and a_d with $2 \leq b_d, a_d \leq p^{n_0} - 2$ to compute and publish the values of B_d and A_d :

$$B_d \equiv r^{b_d} \pmod{p^{n_0}} \text{ and } A_d \equiv r^{a_d} \pmod{p^{n_0}}. \quad (1)$$

- The exponent n of p is computed by

$$n \equiv n_0 + (K_d \bmod D), \quad (2)$$

where K_d represents the secret share between the communicating parties. Namely, it is computed as follows:

$$K_d \equiv B_d^{a_d} \equiv A_d^{b_d} \pmod{p^{n_0}}. \quad (3)$$

If n is not greater than 2, then the communicating parties should choose different values for n_0 and D . Note that n is generated only by the communicating parties, and the recovery of n by a third party requires first determining the shared secret K_d , which is computationally infeasible under DLP. It is important to explain the choice of a large modulus $2p^n$ in this scheme. By Lemma 1, it is recalled that if r is a primitive root modulo p and p^2 , then it is also a primitive root modulo $2p^n$. That is, this modulus generates a cyclic group called the group of units whose elements are relatively prime to $2p^n$, and it is generated by r . Therefore, to extend the classical ElGamal cryptosystem, which operates over the cyclic group $\mathbb{Z}_p^*$, we use a large modulus $2p^n$, where n is generated dynamically by the communicating parties using the MDHKEP. Furthermore, increasing the value of the exponent n increases the order of the underlying cyclic group and consequently enlarges the algebraic domain in which the proposed cryptographic operations are performed. Therefore, the use of this modulus does not increase the theoretical hardness of the DLP, but it enables the generation of dynamic parameters within the proposed scheme.

- Bob and Alice generate the main private and public keys by, respectively, selecting the parameters b and a with $2 \leq b, a \leq 2p^n - 2$ (which should be kept secret) and publish their public keys respectively by the values of r_b and r_a :

$$r_b \equiv r^b \pmod{2p^n} \text{ and } r_a \equiv r^a \pmod{2p^n}. \quad (4)$$

Note that the following image encryption, digital image signature, image decryption and signature verification algorithms are performed only after the successful completion of the modified Diffie–Hellman key-establishment phase, with which the communicating parties derive the shared secret K_d and independently compute the exponent n and the corresponding modulus $2p^n$.

2.2. Image Encryption (By Alice)

Given a grayscale image $I \in [0, 255]^{Q \times J}$ or a color image $[0, 255]^{Q \times J \times 3}$, Alice uses Bob's public key r_b and the primitive root r of $2p^n$ to encrypt the intended image as follows (here we focus on grayscale images as any color image can be converted into a grayscale image):

- Convert the image into a pixel matrix $M = [m_{ij}]$, where $m_{ij} \in \{0, 1, \dots, 255\} \subset \mathbb{Z}_{2p^n}$ and $1 \leq i, j \leq 256$. Since in practice $2p^n > 255$, each pixel value is indeed represented modulo $2p^n$. Therefore, the pixels of $M = [m_{ij}]$ are not required to belong to the group of units modulo $2p^n$. Consequently, pixel values such as zero and even values can be encrypted directly.
- For each m_{ij} , choose a random $f_{ij} \in [2, 2p^n - 2]$.
- Compute the entries of the encrypted matrices as follows:

$$C_{1,ij} \equiv r^{f_{ij}} \pmod{2p^n}, \quad C_{2,ij} \equiv m_{ij} \cdot r_b^{f_{ij}} \pmod{2p^n}. \quad (5)$$

Thus, two encrypted image matrices $C_1 = [C_{1,ij}]$, $C_2 = [C_{2,ij}]$ are generated and sent to Bob for decryption. It should be noted that the first ciphertext component

is generated in the multiplicative group of units modulo $2p^n$, whereas the second ciphertext component is represented as an element of the residue ring $\mathbb{Z}_{2p^n}$ and is therefore not necessarily a unit.

2.3. Digital Signature Scheme (By Alice)

She uses her private key a and the primitive root r to sign the image as follows:

- Flatten the matrix M into a byte array.
- Compute its SHA-256 hash (hexadecimal) and then convert it to an integer.
- Reduce the obtained integer from the previous step modulo

$$\varphi(2p^n) = \varphi(2)\varphi(p^n) = \varphi(p^n) = p^{n-1}(p-1)$$

as follows:

$$R \equiv \text{int}(\text{hash}(M)) \pmod{\varphi(2p^n)}. \quad (6)$$

- Select an integer $w_s \in [1, \varphi(p^n) - 2]$ with $\gcd(w_s, \varphi(p^n)) = 1$ to compute the signature (S_1, S_2) such that

$$S_1 \equiv r^{w_s} \pmod{2p^n}, \quad S_2 \equiv w_s^{-1} \cdot (R - a \cdot S_1) \pmod{\varphi(2p^n)}. \quad (7)$$

A secure random nonce w_s must be generated for each signature and should not be reused with different images, as it may compromise the private signing key.

- Send the encrypted images with the signature (S_1, S_2) to Bob.

2.4. Image Decryption (By Bob)

Given $C_1 = [C_{1,ij}]$, $C_2 = [C_{2,ij}]$ with $1 \leq i, j \leq 256$, Bob uses his private key b to decrypt the cipher images as follows:

- For each pixel, compute

$$m_{ij} \equiv C_{2,ij} \cdot (C_{1,ij}^b)^{-1} \pmod{2p^n}. \quad (8)$$

Note that one can easily verify the above congruence as follows:

$$C_{2,ij} \cdot (C_{1,ij}^b)^{-1} \pmod{2p^n} \equiv m_{ij} \cdot r_b^{f_{ij}} \cdot (r_b^{bf_{ij}})^{-1} \equiv m_{ij} \cdot r_b^{bf_{ij}} \cdot (r_b^{bf_{ij}})^{-1} \equiv m_{ij} \pmod{2p^n}.$$

- Recover the matrix $M = [m_{ij}]$ and convert it back to the original image.

2.5. Image Signature Verification (By Bob)

- Flatten the matrix M into a byte array.
- Compute its SHA-256 hash (hexadecimal) and then convert it to an integer.
- Reduce the obtained integer from the previous step modulo $\varphi(2p^n) = p^{n-1}(p-1)$ using Equation (6):

$$R \equiv \text{int}(\text{hash}(M)) \pmod{\varphi(2p^n)}.$$

- Check whether the following congruence holds:

$$r^R \equiv r_a^{S_1} \cdot S_1^{S_2} \pmod{2p^n}. \quad (9)$$

- Accept the signature if the congruence is valid, otherwise, reject it. Moreover, if $S_1 \notin [1, 2p^n - 1]$ or $\gcd(S_1, 2p^n) \neq 1$, or $S_2 \notin [1, \varphi(2p^n) - 1]$, then reject the signature.

Note that the latter congruence can be verified as follows (with using the fact that $r^{w_s w_s^{-1}} \equiv r \cdot r^{\varphi(p^n)v} \equiv r \pmod{2p^n}$):

$$\begin{aligned} r_a^{S_1} \cdot S_1^{S_2} &\equiv r^{a r^{w_s}} \cdot r^{w_s (w_s^{-1} \cdot (R - a \cdot r^{w_s}))} \pmod{2p^n} \\ &\equiv r^{a r^{w_s}} \cdot r^{(R - a \cdot r^{w_s})} \pmod{2p^n} \\ &\equiv r^R \pmod{2p^n}. \end{aligned}$$

Note that the digital signature is indeed generated from the SHA-256 hash of the original grayscale image prior to encryption. Then after the decryption procedure, the receiver also computes the SHA-256 hash of the decrypted image and verifies the signature to confirm the integrity and authenticity of the recovered image. Therefore, the main aim of the signature is to ensure the integrity and authenticity of the recovered image, which must represent the original image.

3. Implementation of the Proposed Scheme

In this example, we illustrate the steps of each phase of the proposed scheme applied to image encryption, digital signature, decryption, and signature verification using some small parameters and a 3×3 grayscale image matrix. However, in real cryptographic applications these parameters should be very large. For example, the use of a small prime $p = 7$ is intended only to illustrate the correctness of the proposed scheme. In practical applications, p must be chosen as a sufficiently large odd prime in order to achieve a good security level.

3.1. Phase 1: Public Parameters and Keys Generation

- Bob and Alice agree on the prime $p = 7$, a primitive root modulo $p = 7$ and $p^2 = 49$ $r = 3$, and the parameters $n_0 = 2$ and $D = 5$.
- Bob and Alice respectively select the private keys $b_d = 3$ and $a_d = 4$, which are used in MDHKEP, to announce their public keys represented by the values of B_d and A_d as given in (1):

$$B_d \equiv 3^3 \pmod{49} \equiv 27 \text{ and } A_d \equiv 3^4 \pmod{49} \equiv 32.$$

- The exchanged key satisfies the Equation (3) and is given by

$$K_d \equiv 27^4 \equiv 32^3 \pmod{49} \equiv 36.$$

- Therefore, the exponent n is computed using Equation (2):

$$n \equiv 2 + (36 \bmod 5) \equiv 3,$$

which implies that

$$p^n = 343, \quad 2p^n = 686 \quad \text{and} \quad \varphi(2p^n) = 294.$$

- Bob and Alice generate the main private and public keys by respectively selecting the private keys $b = 9$ and $a = 5$, and their corresponding public keys are the values of r_b and r_a as given in (4):

$$r_b \equiv 3^9 \pmod{686} \equiv 475 \text{ and } r_a \equiv 3^5 \pmod{686} \equiv 243.$$

3.2. Phase 2: Encryption Procedure

Suppose that Alice has the following partition of a pixel matrix of a grayscale image:

$$M = \begin{bmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \\ 70 & 80 & 90 \end{bmatrix},$$

which she wants to encrypt before sending it to Bob. So, she performs the following steps:

- For each m_{ij} with $1 \leq i, j \leq 3$, she chooses a random $f_{ij} \in [2, 2p^n - 2] = [2, 684]$ given in the following matrix:

$$F = \begin{bmatrix} 3 & 4 & 5 \\ 6 & 7 & 8 \\ 9 & 10 & 11 \end{bmatrix}.$$

- She uses the primitive root $r = 3$ and Bob's public key $r_b = 475$ to compute the entries of the encrypted matrices using the formulas given in (5):

$$C_{1,ij} = r^{f_{ij}} = \begin{bmatrix} 3^3 & 3^4 & 3^5 \\ 3^6 & 3^7 & 3^8 \\ 3^9 & 3^{10} & 3^{11} \end{bmatrix} \equiv \begin{bmatrix} 27 & 81 & 243 \\ 43 & 129 & 387 \\ 475 & 53 & 159 \end{bmatrix} \pmod{686}$$

and

$$\begin{aligned} C_{2,ij} = m_{ij} \cdot r_b^{f_{ij}} &= \begin{bmatrix} 10 \cdot 475^3 & 20 \cdot 475^4 & 30 \cdot 475^5 \\ 40 \cdot 475^6 & 50 \cdot 475^7 & 60 \cdot 475^8 \\ 70 \cdot 475^9 & 80 \cdot 475^{10} & 90 \cdot 475^{11} \end{bmatrix} \\ &\equiv \begin{bmatrix} 158 & 552 & 222 \\ 656 & 538 & 18 \\ 28 & 402 & 358 \end{bmatrix} \pmod{686}. \end{aligned}$$

- The encrypted image matrices are given by $C_1 = [C_{1,ij}]$ and $C_2 = [C_{2,ij}]$.

3.3. Phase 3: Digital Signature Scheme Procedure

Alice signs the image by using her private key $a = 5$ and the primitive root $r = 3$ to sign the image as follows:

- Flatten M into byte array: $[10, 20, 30, 40, 50, 60, 70, 80, 90]$.
- Compute the SHA-256 Hash of M (hex): `a51f9a0b6ea3523acf4697ac74fc8e3653ae8cbe1abf58 adffb069e1f5e8c2e5`.
- Compute the Hash as integer H : `74687455452395441613621917567455303537012853393425368066774525802745313149669`.
- Reduce the obtained integer modulo $\varphi(686) = 294$ by using Equation (6):

$$R \equiv \text{int}(\text{hash}(M)) \pmod{294} = 137.$$

- Choose an integer $w_s = 5$ such that $\gcd(5, 294) = 1$ and $w_s^{-1} \equiv 59 \pmod{294}$.
- Compute the signature (S_1, S_2) as given in (7):

$$S_1 = 3^5 \equiv 243 \pmod{686}, \quad S_2 = 59 \cdot (137 - 5 \cdot 243) \pmod{294} \equiv 196.$$

- Send the encrypted images with the signature $(S_1, S_2) = (243, 196)$ to Bob.

3.4. Phase 4: Decryption Procedure

When Bob gets the encrypted images represented by the matrices

$$C_1 = [C_{1,ij}] = \begin{bmatrix} 27 & 81 & 243 \\ 43 & 129 & 387 \\ 475 & 53 & 159 \end{bmatrix}$$

and

$$C_2 = [C_{2,ij}] = \begin{bmatrix} 158 & 552 & 222 \\ 656 & 538 & 18 \\ 28 & 402 & 358 \end{bmatrix}.$$

He uses his main private $b = 9$ and does the following steps:

- Compute $M = [m_{ij}]$ as follows using Formula (8):

$$\begin{aligned} M = [m_{ij}] &\equiv [C_{2,ij} \cdot (C_{1,ij}^b)^{-1}] \pmod{2p^n} \\ &\equiv \begin{bmatrix} 158 \cdot 27^{-9} & 552 \cdot 81^{-9} & 222 \cdot 243^{-9} \\ 656 \cdot 43^{-9} & 538 \cdot 129^{-9} & 18 \cdot 387^{-9} \\ 28 \cdot 475^{-9} & 402 \cdot 53^{-9} & 358 \cdot 159^{-9} \end{bmatrix} \pmod{686} \\ &\equiv \begin{bmatrix} 158 \cdot 139 & 552 \cdot 435 & 222 \cdot 167 \\ 656 \cdot 113 & 538 \cdot 97 & 18 \cdot 575 \\ 28 \cdot 615 & 402 \cdot 449 & 358 \cdot 349 \end{bmatrix} \pmod{686} \\ &\equiv \begin{bmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \\ 70 & 80 & 90 \end{bmatrix}, \end{aligned}$$

which recovers the original matrix of the grayscale image.

3.5. Phase 5: Image Signature Verification Procedure

The last phase that Bob performs when obtaining the matrix M with the signature $(S_1, S_2) = (243, 196)$ is to perform the following steps in order to verify the signature:

- Doing the same first four steps mentioned in Phase 3 to compute reduced integer value modulo $\varphi(2p^n) = \varphi(686) = 294$ (i.e., R given in Equation (6)) of the SHA-256 of M :

$$R \equiv \text{int}(\text{hash}(M)) \pmod{294} \equiv 137.$$

- Checking whether the congruence given in (9) holds:

$$r^R \equiv r_a^{S_1} \cdot S_1^{S_2} \pmod{686}$$

using the primitive root $r = 3$ and Alice's public key $r_a = 243$. Namely,

$$453 \equiv 3^{137} = r^R \equiv r_a^{S_1} \cdot S_1^{S_2} = 243^{243} \cdot 243^{196} \equiv 453 \pmod{686}.$$

- Accepting the signature.

4. Experimental Results and Performance Evaluation

In this section, a set of experiments is performed using grayscale images of size 256×256 . All of the following results are executed in the SageMath computational environment through the CoCalc cloud-based platform [18]. The experiments are conducted on a system with Windows 11 Pro, featuring a 12th Generation, Intel(R), Core(TM) i7-12650H

processor (2.70 GHz), and 16 GB of RAM. For these experiments, we use four well-known images (Lena, Mandrill, House, Moon), which are used in various cryptographic papers since they represent a wide range of image features such as textures, edges, gradients and uniform regions. Such features are important for image-statistical evaluations and visual impact of image encryption algorithms. More precisely, these images are retrieved from the USC-SIPI Image Database [19]. Without loss of generality and for the purpose of illustrating the correctness and successful execution performance of the encryption, decryption and signature scheme algorithms of the proposed scheme's algorithms, we run the algorithms on the images using relatively small cryptographic parameters during the key generation and public parameter initialization stages. These parameters give a clearer demonstration of the functionality of the proposed scheme. Moreover, to evaluate the performance of the algorithms of the proposed scheme with larger cryptographic parameters, experiments were carried out with prime numbers of 256 bits, producing significantly larger moduli $2p^n$ in the encryption, decryption and signature procedures. Furthermore, to strengthen the security of the proposed signature scheme and to avoid nonce reutilization, we use large and distinct nonce values for each signature process. As a result, the use of larger parameters increases the computational cost, especially for modular exponentiation operations. The corresponding performance analysis and computational metrics are presented in the following subsection. Appendix B indicates values of the global parameters used in the key generation, public and private keys, nonces and signature results corresponding to each of the four images used in the experimental evaluation of the proposed scheme.

Note that in real world applications, the cryptographic parameters used in our proposed scheme must be significantly larger to ensure practical security, particularly using primes with at least 2048-bits (preferably 3072-bits). Moreover, they should not be reused with different images as it may disclose the security of the private key. For each signature generation, the signing nonce value must be chosen independently and uniformly at random. From the above parameters and their corresponding results, in the following, we give a comprehensive performance and security evaluation of our proposed scheme's algorithms with their results for each of the four images used in the experiments. Note that the proposed scheme is evaluated mainly with certain statistical measures such as histogram analysis, entropy, correlation coefficients, NPCR and UACI. It is important to mention that these metrics provide empirical indicators of diffusion and confusion properties in the encrypted images. In other words, having optimal measures of these metrics does not constitute formal security proofs against statistical and differential attacks. Note that for more details about the sources of the used metrics in all of the following sections, we refer to [20–22].

4.1. Histogram Analysis

Histogram analysis is used mainly to evaluate the effectiveness of image encryption schemes by showing the distribution of pixel intensities between 0 and 255 in 8-bit grayscale images. In fact, histogram analysis shows how uniformly the encrypted image's pixels are distributed. An effective image encryption scheme must have significantly different histograms for the encrypted image and the original image. Ideally, the histogram of the encrypted image should appear uniformly distributed to indicate that the encryption procedure has removed any statistical patterns presented in the original image. This histogram does not reveal any useful information about the original image and indicates a good statistical randomization of the encrypted image. By comparing the original and encrypted images' histograms, one can visually evaluate and indicate how well the encryption algorithm is effective and covers the original contents.

More precisely, from the experimental results of the performance of our encryption and decryption schemes on the Lena image (shown in Figure 1), Mandrill image (shown in Figure 3), House image (shown in Figure 5) and Moon image (shown in Figure 7), we obtain their corresponding histograms given in Figures 2, 4, 6 and 8, respectively. Note that for visualization purposes, the encrypted image shown in each of Figures 1, 3, 5 and 7 represents a visualization of the corresponding ciphertext matrix $C_2 = [C_{2,ij}]$ (whose entries are computed according to (5)). Accordingly, for visualization and image-statistical analysis, each value in the ciphertext is mapped to an 8-bit grayscale intensity by reducing it modulo 256, as standard grayscale images are displayed using 8-bit pixel intensities with values in the range $\{0, 1, \dots, 255\}$. Therefore, the displayed encrypted images and their corresponding histograms are based on the 8-bit grayscale representation of the ciphertext matrix rather than on the full ciphertext values. However, during the decryption procedure, the decryption algorithm operates directly on the original ciphertext matrices C_1 and C_2 to recover the original image. Thus, the following encrypted images and their corresponding histograms are represented using the 8-bit grayscale representation of the ciphertext matrix obtained by reducing each ciphertext value modulo 256.

Likewise, all image-based statistical measures presented throughout the paper, including the correlation coefficients, NPCR, UACI and entropy values, are computed from the same 8-bit grayscale representation of the ciphertext matrix. It should be noted that these measures are only intended to evaluate the statistical properties of the encrypted image representation and should not be interpreted as statistical measures computed directly from the original ciphertext values modulo $2p^n$.

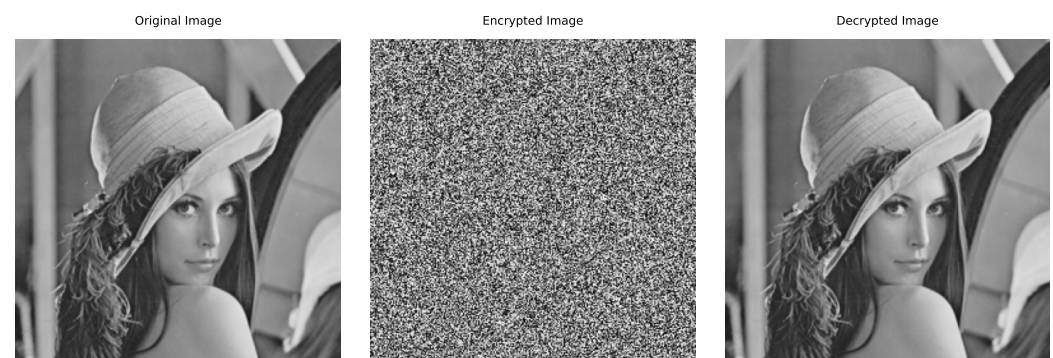


Figure 1. Original, Encrypted, and Decrypted Lena Images.

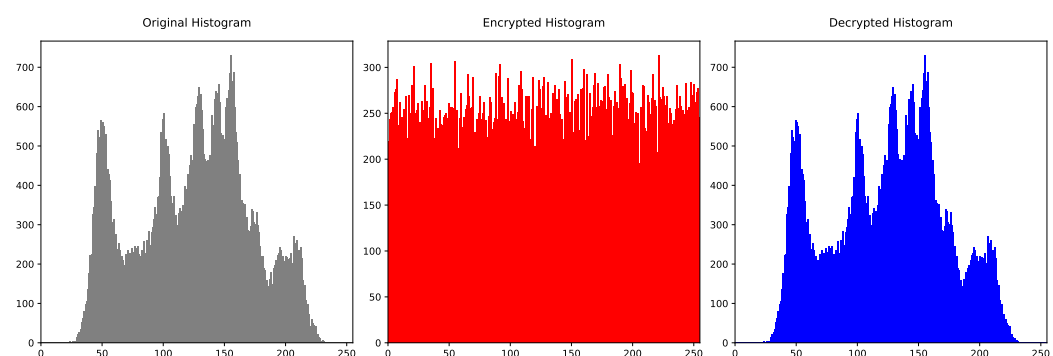


Figure 2. Histograms of Original, Encrypted, and Decrypted Lena Images. The histogram of the encrypted image is computed from the 8-bit grayscale representation of the ciphertext matrix.

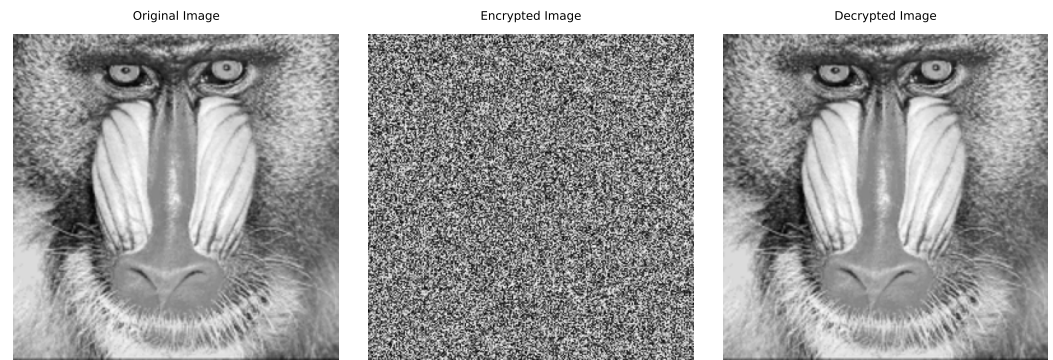


Figure 3. Original, Encrypted, and Decrypted Mandrill Images.

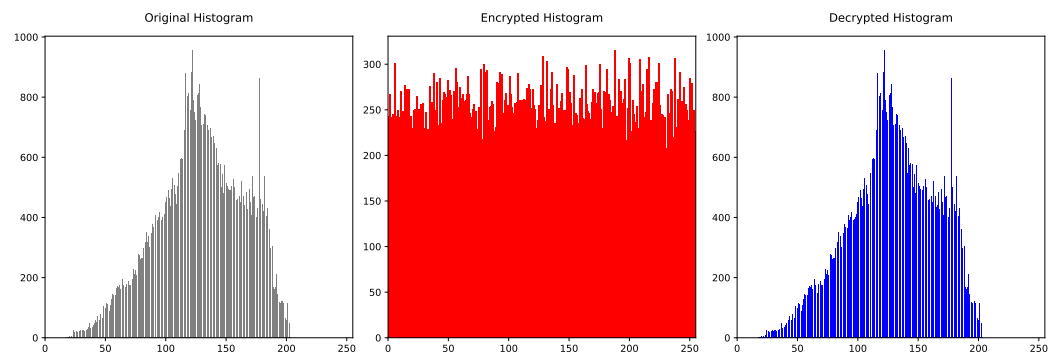


Figure 4. Histograms of Original, Encrypted, and Decrypted Mandrill Images. The histogram of the encrypted image is computed from the 8-bit grayscale representation of the ciphertext matrix.

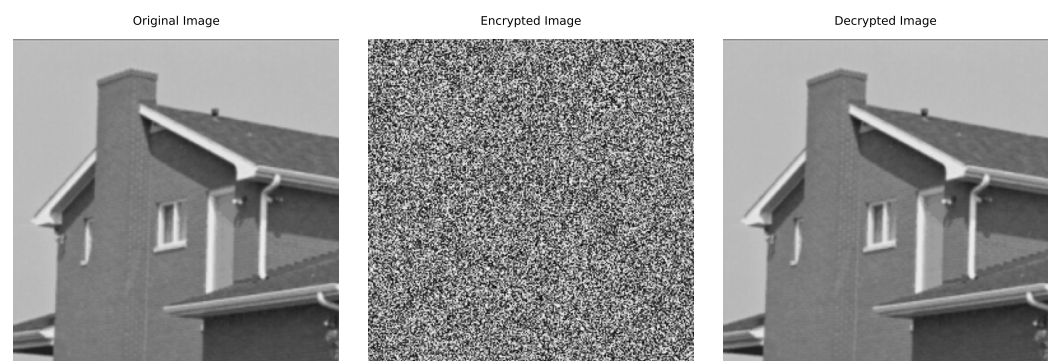


Figure 5. Original, Encrypted, and Decrypted House Images.

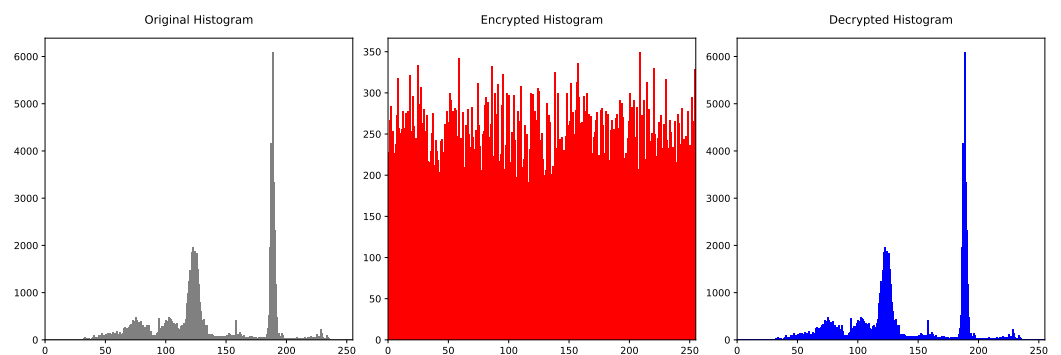


Figure 6. Histograms of Original, Encrypted, and Decrypted House Images. The histogram of the encrypted image is computed from the 8-bit grayscale representation of the ciphertext matrix.

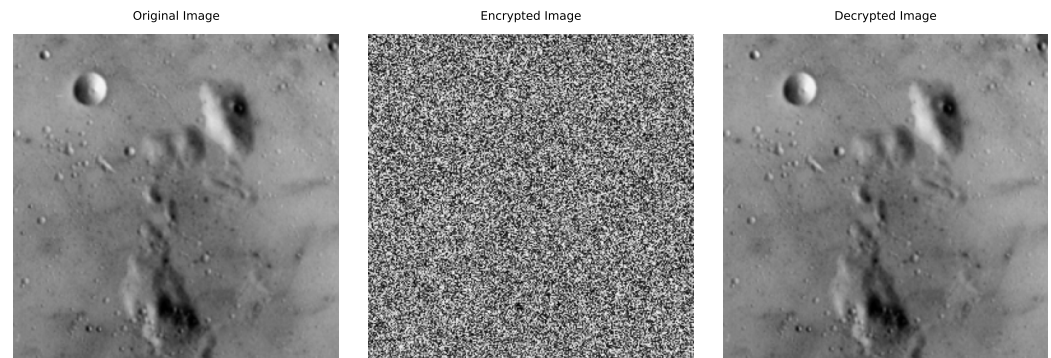


Figure 7. Original, Encrypted, and Decrypted Moon Images.

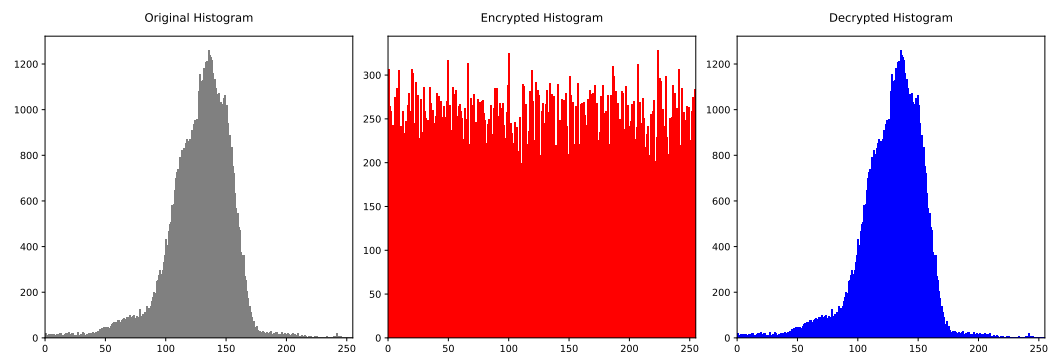


Figure 8. Histograms of Original, Encrypted, and Decrypted Moon Images. The histogram of the encrypted image is computed from the 8-bit grayscale representation of the ciphertext matrix.

By comparing the original, encrypted and decrypted images' histograms, firstly the original and decrypted images' histograms are exactly the same, which proves validation of the decryption algorithm. Also, for all of the images, the histograms of the original and encrypted images are significantly different. Furthermore, the histograms of the encrypted images are uniformly distributed.

4.2. Performance and Accuracy Metrics Analysis

Important metrics to evaluate the accuracy of image reconstruction after decryption are the Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR). The MSE is used to measure the average squared difference between the pixel values of the original and decrypted images. A lower MSE value (the best MSE value is 0) indicates a higher similarity between the original image and decrypted image. On the other hand, the PSNR (expressed in decibels (dB)) is applied to quantify the ratio between the highest possible value of pixels and the distortion's power given by encryption and decryption. A higher PSNR value (the best is theoretically ∞) implies better image quality and accuracy and gives that the error between the original and decrypted images is very small or close to zero. Furthermore, an essential tool for evaluating a cryptosystem's performance is the timing analysis for the processes of encryption and decryption. However, with the advent of quantum computing, security has become more important than execution time. Note that for all four images, the values of $MSE = 0$ and $PSNR = \infty$, which indicate that the decrypted images are exactly identical to the original images, with zero errors between them. These values indicate that our scheme has good image quality and computational efficiency. Moreover, the execution times of the encryption and decryption processes demonstrate favorable computational performance under the tested conditions. Table 1 summarizes the execution time values for the four tested images shown in Figures 1, 3, 5 and 7.

Table 1. Execution Time.

Image	Enc. Time (s)	Dec. Time (s)
Lena	0.5847	2.0203
Mandrill	0.4144	2.0373
House	0.8765	2.1229
Moon	0.7269	2.2594

From the results of the above table, it is clear that the algorithms of the proposed scheme do not introduce a low computational cost due to the use of large primes of length 256 bit, providing larger moduli $2p^n$ with $n \geq 2$ in the modular arithmetic operations. From the structure of the encryption and decryption algorithms of the proposed scheme, the decryption procedure is slightly slower than the encryption procedure since it requires a modular inversion in addition to modular exponentiation for each pixel.

4.3. Correlation Coefficient Analysis

The correlation coefficient analysis mainly evaluates the effectiveness of image encryption algorithms in different directions. In original images, there exist high correlation coefficients (often close to 1) due to the strong similarity in the adjacent pixels in vertical, horizontal and diagonal directions. On the other hand, in an effective image encryption scheme, the correlation coefficients of adjacent pixels in the encrypted image should be reduced to be ideally close to zero or slightly negative. In this case, the image encryption scheme produces a high level of randomness and a successful disruption of spatial redundancy, which indicates good confusion and diffusion characteristics.

Table 2 below gives a comparison of correlation coefficients in horizontal, vertical and diagonal directions for the original and encrypted standard test images given in Figures 1, 3, 5 and 7.

Table 2. Correlation Coefficients of Original and Encrypted Images.

Image	Type	Horizontal	Vertical	Diagonal
Lena	Original	0.9456	0.9727	0.9213
	Encrypted	0.0008	0.0008	−0.0047
Mandrill	Original	0.9326	0.9160	0.8769
	Encrypted	−0.0003	0.0008	0.0051
House	Original	0.9853	0.9737	0.9617
	Encrypted	0.0031	0.0008	0.0008
Moon	Original	0.9624	0.9796	0.9582
	Encrypted	0.0077	−0.0036	0.0008

From the results in the above table, we observe that the original images exhibit strong positive correlations near to 1 for the three different directions, while the encrypted images yield very low correlation values close to zero. These results indicate that the proposed scheme reduces the correlation between adjacent pixels in the encrypted images.

4.4. Differential Attacks Analysis

Differential attacks are a type of cryptanalysis techniques used to determine and evaluate the sensitivity of an encryption algorithm to small changes in the input. In case of image encryption, this involves changing or modifying a single pixel in the original image and noticing the effect on the resulting encrypted image. An effective image encryption scheme should lead to a significant and unpredictable change in the encrypted output even with a small modification of the input. In fact, the most well-known metrics used

to evaluate resistance against such attacks are called by the Number of Pixels Change Rate (NPCR) and the Unified Average Changing Intensity (UACI). The former is used to measure the percentage of pixels in the encrypted image that change values when a single pixel of the original image is changed before encryption. On the other hand, the UACI is used to quantify the average intensity difference between two encrypted images obtained from a slight change in two different inputs. High values for NPCR (ideally above 99%) and UACI (around 33%) indicate a strong sensitivity of the pixels in the encrypted image to small changes in the pixels of the plain image, providing desirable diffusion characteristics for image encryption.

Table 3 summarizes the values of NPCR and UACI obtained from encrypting the four images given in Figures 1, 3, 5 and 7, then encrypting them again after altering only a single pixel in each image.

Table 3. Differential Attack Resistance.

Image	NPCR (%)	UACI (%)
Lena	99.4774	33.4164
Mandrill	99.1396	33.3020
House	99.2569	33.3925
Moon	99.2279	33.0612

The results of NPCR and UACI in the above table for all of the images are almost equal to the ideal standard values, and this indicates that our scheme is very sensitive to small modifications in the input image. Therefore, the proposed scheme exhibits favorable diffusion and confusion properties, indicating satisfactory statistical performance for image encryption.

4.5. Entropy Analysis

Entropy is an important concept in cryptography and information theory, and it is used to measure the average level of randomness or uncertainty in certain data. In case of image encryption, entropy is used to determine the degree of unpredictability in pixel intensity distributions. A higher entropy value (maximum value is 8 for an 8-bit grayscale image) indicates greater randomness, which suggests resistance to entropy based attacks under statistical evaluation. On the other hand, original images generally have lower entropy values compared to the values of encrypted images due to the presence of patterns, redundancies and structured content.

The following table summarizes the entropy values (denoted by H) of the original and encrypted images for the four studied images shown in Figures 1, 3, 5 and 7.

From Table 4, we observe that the encrypted images have entropy values approaching 8. This result shows that the proposed scheme gives encrypted images with high entropy, providing empirical evidence of a high degree of statistical randomness.

Table 4. Entropy Analysis of Images.

Image	H (Original Image)	H (Encrypted Image)
Lena	7.4313	7.9955
Mandrill	7.1396	7.9942
House	6.3264	7.9857
Moon	6.6457	7.9940

5. Security Analysis and Parameter Considerations

In this section, we first present the security model of the proposed scheme, while the second part of this section discusses the security properties and practical parameter considerations.

5.1. Security Model and Parameter Classification

The purpose of this work is to suggest a modification of the classical ElGamal scheme to establish the correctness of the proposed scheme's algorithms and to provide confidentiality together with authentication and integrity of the recovered plaintext image. Our proposed scheme is analyzed under the classical computational model, where all public keys and algorithms are assumed to be known to everyone, while the scheme's private keys and the shared secret conducted by the modified Diffie–Hellman key exchange protocol remain secret. The parameter n is dynamically generated using the Diffie–Hellman shared secret K_d , where K_d is computed according to Equation (3), and is used to construct the corresponding session-dependent modulus $2p^n$. Hence, the recovery of n requires first determining the shared secret K_d , which is computationally infeasible without solving the underlying discrete logarithm problem in order to obtain one of the secret exponents a_d or b_d modulo p^{n_0} . For clarity, in Table 5 we summarize the classification of the parameters used throughout the proposed scheme, indicating the phase in which they are generated.

Table 5. Classification of the Cryptographic Parameters Used in the Proposed Scheme.

Parameter	Classification	Generation Phase
p	Public	System initialization.
r	Public	System initialization.
n_0	Public	System initialization.
D	Public	System initialization.
A_d, B_d	Public	Modified Diffie–Hellman key-establishment.
a_d, b_d	Private	Modified Diffie–Hellman key-establishment.
K_d	Session-derived secret	Modified Diffie–Hellman key-establishment.
n	Session-derived parameter	Dynamic parameter generation.
$2p^n$	Session-derived modulus	Dynamic parameter generation.
a, b	Private	Key generation.
r_a, r_b	Public	Key generation.
f_{ij}	Ephemeral	Encryption.
w_s	Ephemeral	Signature generation.

Thus, the security of the proposed scheme depends on the computational hardness of solving the DLPs underlying the modified Diffie–Hellman key-establishment procedure and the modified ElGamal cryptosystem.

Note that since the aim of the present work is to establish the functional correctness of the proposed scheme, provide a security discussion under the computational difficulty assumptions of the DLP, and present empirical performance and image-statistical evaluations. Accordingly, formal security notions such as indistinguishability under chosen-plaintext attacks (IND-CPA), indistinguishability under chosen-ciphertext attacks (IND-CCA), existential unforgeability under adaptive chosen-message attacks (EUF-CMA), and ciphertext integrity attack are beyond the scope of this work and are left for future research investigation. Furthermore, the proposed scheme is not resistant to quantum attacks since the DLP can be efficiently solved using quantum algorithms like Shor's algorithm [23]. In other words, increasing the security parameters leads to increased computational effort required for classical attacks but does not provide resistance against quantum attacks. For clarity, Table 6 summarizes the security assumptions and the attack models considered or not considered in this work, together with those left for future investigation.

Table 6. Scope of the Security Analysis.

Attack	Status
Discrete logarithm problem (DLP)	Considered. The security of the proposed scheme is based on the computational hardness of the DLP.
Chosen-plaintext attack (IND-CPA)	Future work.
Chosen-ciphertext attack (IND-CCA)	Future work.
Adaptive chosen-message attack (EUF-CMA)	Future work.
Quantum attacks (e.g., Shor's algorithm)	Not considered. The proposed scheme is not post-quantum secure.

5.2. Security and Parameters Discussion

The numerical values used in the experimental section, as listed in Appendix B, such as 256-bit prime number p , $r = 3$, $n_0 = 2$, $a_d = 4$, $b_d = 3$, $D = 5$, $A_d = 81$, $B_d = 27$, $K_d = 531,441$, $n = 3$ and modulus $2p^n$, were chosen only to simplify the presentation and show the verification of our proposed encryption, decryption, key generation and signature scheme procedures.

For clarity, Table 7 summarizes the representative parameters, key sizes, nonce sizes, ciphertext expansion and signature sizes used in the experimental evaluation.

Table 7. Representative Parameters and Communication Overhead Used in the Experiment.

Parameter	Value	Remarks
Prime (p)	256 bits	Randomly generated prime.
Primitive root (r)	3	Generator of the group modulo p and $2p^n$.
Initial exponent (n_0)	2	Initial exponent used to generate n .
Parameter (D)	5	Determines the dynamic exponent n .
Secret values (a_d, b_d)	(4, 3)	Secret values used in the MDHKEP and selected during the key-establishment stage.
Public values (A_d, B_d)	(81, 27)	Published values and computed by (1) and used in the MDHKEP.
Shared secret (K_d)	531,441	The secret share of the MDHKEP computed by (3) and used to compute n .
Exponent (n)	3	The exponent of p computed by (2).
Modulus ($2p^n$)	768 bits	Public modulus used for encryption and digital signatures.
Public key size	768 bits	Each public key (r_a, r_b) is regenerated for every execution.
Private key size	Variable	Secret exponents (a, b) generated independently for each execution.
Encryption nonce (f_{ij})	Random	Fresh random value selected from $[2, 2p^n - 2]$ for every encrypted pixel.
Signature nonce (w_s)	Random	Fresh random value selected from $[1, \varphi(p^n) - 2]$ and chosen differently for every image signature.
Hash size	256 bits	SHA-256 message digest.
Signature size	2×768 bits	Signature consists of the two components (S_1, S_2).
Ciphertext size	2×768 bits/pixel	Each pixel of the image matrix $M = [m_{ij}]$ is represented by two 768-bit ciphertext components.
Ciphertext components	2:1	Each pixel of the image matrix $M = [m_{ij}]$ is encrypted as the ciphertext pair ($C_{1,ij}, C_{2,ij}$).

In fact, these small parameters should not be used to provide secure environments. In real-world applications and security standards, the strength of our scheme is based on the

hardness of solving the DLP, which in fact requires very larger primes with at least 2048-bits or more for higher security levels. In fact, larger parameter sizes such as 2048-bit and 3072-bit are commonly recommended for practical DLP-based cryptosystems. However, they were not experimentally evaluated in this present work and are left for future investigation.

More precisely, the security of the proposed scheme depends on the computational difficulty in solving the DLP underlying the modified ElGamal cryptosystem. In addition, the exponent n is dynamically derived from the shared secret established from the procedure of Diffie–Hellman key-establishment.

In other words, the proposed modification to the classical ElGamal cryptosystem is not intended to suggest that the modulus $2p^n$ makes the DLP fundamentally more difficult. Rather, it extends the framework of the classical scheme by incorporating dynamically generated system parameters obtained during the process of Diffie–Hellman key-establishment. Consequently, the proposed scheme continues to depend on the security assumptions of discrete-logarithm-based cryptography while providing a more flexible mechanism for generating parameters.

However, the proposed scheme incurs additional computational cost, especially due to modular exponentiation operations. Therefore, in a real world implementation, an appropriate balance between efficiency and the security requirements of the application should be considered. Note that similar to conventional asymmetrical cryptosystems, our proposed scheme can also be incorporated into a hybrid encryption scheme, which combines public-key cryptography with symmetric encryption. Since it already employs a Diffie–Hellman key protocol to derive the exponent n , the resulting exchanged secret can be extended to establish a symmetric session key. Such hybrid constructions have become standard approaches for combining secure key establishment with efficient symmetric encryption in practical cryptographic applications; see e.g., [24].

In Table 8, we give a comparison regarding the operating cost between our proposed scheme with the classical ElGamal cryptosystem [1] and one of its variants introduced by Mohit and Biswas [14] in which they used a session key instead of the Diffie–Hellman key to extract the message. In order to present the computational costs, we denote the computational cost of a modular exponentiation modulo q , the cost of a modular multiplication modulo q , and the cost of a modular inverse modulo q (where $q = p$ or $2p^n$) by E_q , M_q and I_q , respectively.

Table 8. Computational Complexity Comparison of ElGamal-Based Cryptosystems.

Operation	Classical ElGamal	Session-Key Variant	Proposed Scheme
Key Generation	E_p	E_p	E_{2p^n}
Encryption	$2E_p + M_p$	$2E_p + M_p$	$2E_{2p^n} + M_{2p^n}$
Decryption	$E_p + I_p + M_p$	$E_p + I_p + M_p$	$E_{2p^n} + I_{2p^n} + M_{2p^n}$
Signature Generation	$2E_p + I_p$	$2E_p + I_p$	$2E_{2p^n} + I_{2p^n}$
Signature Verification	$3E_p + M_p$	$3E_p + M_p$	$3E_{2p^n} + M_{2p^n}$

The additional computational cost of the proposed scheme observed in Table 8 can be considered an acceptable trade-off for its modified cryptographic design and additional functionality. Our proposed scheme is more expensive than the other schemes due to its modified cryptographic construction.

Furthermore, our proposed scheme is compared with other related encryption methods with respect to correlation coefficients, NPCR, UACI, and entropy of the encrypted decrypted images. The comparison is performed using the Lena image. Table 9 gives the results of the comparison on the correlation of two adjacent pixels in the

original and encrypted Lena image of the proposed scheme with different asymmetric encryption methods.

Table 9. Comparison of Correlation Coefficients with Asymmetric Methods for the Lena Image.

Method	Type	Horizontal	Vertical	Diagonal
Ours	Original	0.9456	0.9727	0.9213
	Encrypted	0.0008	0.0008	−0.0047
Ref. [7]	Original	0.9325	0.9139	0.9469
	Encrypted	0.0074	−0.0094	−0.0054
Ref. [9]	Original	0.9858	0.9801	0.9669
	Encrypted	0.0019	−0.0024	0.0011
Ref. [10]	Original	0.9769	0.9596	0.9444
	Encrypted	−0.0001	0.0089	0.0091

The results in the above table indicate that for all considered methods, including our method, the original images exhibit strong positive correlations near to 1 for all directions, while the encrypted images yield very low correlation values close to zero. This shows that the proposed scheme is consistent with these encryption methods in the literature, demonstrating comparable image-statistical behavior in terms of the correlation coefficients.

Furthermore, Table 10 shows the results of the comparison with respect to NPCR, UACI, and entropy of the original and encrypted Lena image between the proposed scheme and asymmetric encryption methods mentioned in Table 9.

Table 10. Comparison of NPCR, UACI and Entropy with Asymmetric Methods for the Lena Image.

Method	NPCR	UACI	Entropy (Original)	Entropy (Encrypted)
Ours	99.4774	33.4164	7.4313	7.9955
Ref. [7]	99.62	33.46	7.2531	7.9992
Ref. [9]	99.6113	33.4682	7.3871	7.9993
Ref. [10]	100	33.47	7.758377	7.991226

The results of the above table show that the performance of our scheme compared to other existing image encryption methods mentioned in the literature is very ideal, as the entropy values of the encrypted images indicate a high degree of randomness similar to the other methods. Moreover, the values of NPCR and UACI are close to the theoretical optimal benchmarks, which give strong sensitivity to small changes in the original image. These results demonstrate empirical evidence of favorable image-statistical properties such as diffusion and statistical randomization in the encrypted grayscale images.

In addition to the above comparisons, a future work will include broader comparisons with RSA-based and hybrid asymmetric encryption schemes, together with additional implementation details to facilitate reproducibility.

It should be noted that our proposed scheme is an asymmetric cryptographic construction, which requires the use of modular exponentiation operations and the generation of two ciphertext components, and these naturally result in higher computational and communication overhead than symmetric cryptosystems. Consequently, the increased computational cost represents the usual trade-off for additional security features such as key establishment and digital signatures. Therefore, to provide a comparative empirical performance reference with well-known and commonly used cryptographic methods, in Table 11 we provide a comparison between our proposed scheme and two widely used symmetric

encryption algorithms, which are known by the AES-CBC [25] and ChaCha20 [26]. These methods are designed for high-speed and efficient data protection. Thus, the following comparison does not give a direct equivalence between symmetric and asymmetric cryptographic functionalities, it is mainly intended to provide a comparative evaluation regarding the encryption time, decryption time, NPCR, UACI and entropy under identical experimental conditions. More precisely, the evaluation is conducted using large prime numbers of 256 bits and large and distinct nonce values for each signature process as the ones indicated in Section 4. The corresponding experimental results for the Lena image are summarized in the following table:

Table 11. Comparison of Encryption and Decryption Times, NPCR, UACI and Entropy with Symmetric Methods for the Lena Image

Method	Enc. Time	Dec. Time	NPCR	UACI	Entropy
Ours	0.5847	2.0203	99.4774	33.4164	7.9955
AES-CBC	0.0009	0.0003	99.63	34.22	7.9973
ChaCha20	0.1090	0.0004	99.62	34.19	7.9970

From Table 11, it can be observed that the values of NPCR, UACI, entropy are close to the values of the benchmark algorithms. Therefore, these results indicate that our proposed scheme exhibits favorable empirical image-statistical properties. On the other hand, the execution time results indicate that the AES-CBC and ChaCha20 achieve lower computational cost and faster encryption and decryption times due to their symmetric-key structures and optimization for high-speed encryption. In contrast, our proposed scheme requires a higher cost because it requires more modular exponentiation operations over large parameters within a public-key cryptographic framework. In fact, the use of 256-bit prime numbers and large nonces and moduli of the form $2p^n$ with $n \geq 3$ increases the overall computational complexity of the encryption, decryption and signature procedures. Note that this additional computational cost is a trade-off for the proposed scheme, whose security is based on the hardness of DLP. To be more specific, the security of the proposed scheme depends on the difficulty of solving two related DLP cases: one arising from the MDHKEP used for a sharing key, and another originating from the classical ElGamal-type cryptographic structure.

6. Conclusions

This paper presents a modified version of the classical ElGamal cryptosystem designed for image encryption and digital signature applications. This scheme is based on a modification of the classical scheme by operating the encryption, decryption and signature scheme algorithms over a larger modulus presented by $2p^n$ for an odd prime p and integer $n \geq 2$ derived by the communicating parties using a modification of the DHKEP. This leads to a modified algebraic structure, while the underlying security assumption reduces to the computational hardness of the classical DLP rather than introducing a fundamentally harder DLP. Instead, the modified modulus provides a more flexible parameter generation mechanism. In addition, the recovery of n (or the modulus $2p^n$) requires the knowledge of the shared secret K_d whose recovery is computationally infeasible under the discrete logarithm assumption underlying the MDHKEP.

Furthermore, the experimental evaluations show and confirm that the decrypted images are completely identical to the originals by indicating $MSE = 0$ and $PSNR = \infty$, which highlight the capability of the system to achieve perfect reconstruction without introducing distortion. However, the proposed scheme requires additional cost in comparison

to other cryptographic methods, the execution times for encryption and decryption show favorable computational performance even when larger moduli and nonces are employed. Based on the conducted empirical image-statistical evaluations, the results indicate that the proposed modification of the ElGamal cryptosystem achieves high fidelity image recovery and favorable diffusion and confusion properties under the classical computational assumptions of the DLP.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The author declares no conflicts of interest.

Appendix A. Algorithms of the Proposed Scheme

Algorithm A1 Public Parameters and Key Generation

- 1: **Input:** Prime p , primitive root r , positive integer parameters n_0, D
 - 2: Choose a large prime p
 - 3: Select a primitive root r of p and p^2
 - 4: Fix positive integers n_0 and D
 - 5: Alice chooses a secret key $a_d \in [2, p^{n_0} - 2]$
 - 6: Bob chooses a secret key $b_d \in [2, p^{n_0} - 2]$
 - 7: Alice computes $A_d = r^{a_d} \bmod p^{n_0}$ and announces it publicly.
 - 8: Bob computes $B_d = r^{b_d} \bmod p^{n_0}$ and announces it publicly.
 - 9: The shared secret is computed by Alice and Bob: $K_d = A_d^{b_d} \equiv B_d^{a_d} \bmod p^{n_0}$
 - 10: Both compute $n = n_0 + (K_d \bmod D)$ such that $n > 2$
 - 11: Both parties independently derive the modulus $2p^n$
 - 12: Alice selects a private key $a \in [2, 2p^n - 2]$
 - 13: Bob selects a private key $b \in [2, 2p^n - 2]$
 - 14: Alice publishes the key $r_a = r^a \bmod 2p^n$
 - 15: Bob publishes the key $r_b = r^b \bmod 2p^n$
 - 16: **Output:** (r_a, r_b, a, b, p, n)
-

Algorithm A2 Image Encryption (Alice)

- 1: **Input:** Image matrix $M = [m_{ij}]$, Bob's public key r_b
 - 2: **for** each pixel m_{ij} **do**
 - 3: Choose a random value $f_{ij} \in [2, 2p^n - 2]$
 - 4: Compute $C_{1,ij} = r^{f_{ij}} \bmod 2p^n$
 - 5: Compute $C_{2,ij} = m_{ij} \cdot r_b^{f_{ij}} \bmod 2p^n$
 - 6: **Output:** Cipher images C_1, C_2 , that are sent to Bob
-

Algorithm A3 Digital Signature Generation (Alice)

- 1: **Input:** Image matrix M , Alice's private key a
 - 2: Flatten M into byte array
 - 3: Compute $H = \text{SHA-256}(M)$
 - 4: Convert H to integer h
 - 5: Compute $R = h \bmod \varphi(2p^n)$
 - 6: Choose the integer $w_s \in [1, \varphi(2p^n) - 2]$ with $\gcd(w_s, \varphi(2p^n)) = 1$
 - 7: Compute $S_1 = r^{w_s} \bmod 2p^n$ and $S_2 = w_s^{-1}(R - aS_1) \bmod \varphi(2p^n)$
 - 8: **Output:** Signature (S_1, S_2)
-

Algorithm A4 Image Decryption (Bob)

- 1: **Input:** Cipher image matrices C_1, C_2 , Bob's private key b
- 2: **for** each pixel (i, j) **do**
- 3: Compute $T_{ij} = C_{1,ij}^b \bmod 2p^n$
- 4: Compute $m_{ij} = C_{2,ij} \cdot T_{ij}^{-1} \bmod 2p^n$
- 5: Reconstruct decrypted image $M = [m_{ij}]$
- 6: **Output:** Original image M

Algorithm A5 Image Signature Verification (Bob)

- 1: **Input:** Image matrix M , signature (S_1, S_2) , Alice's public key r_a
- 2: Flatten the image matrix M into a byte array
- 3: Compute $H = \text{int}(\text{SHA-256}(M))$
- 4: Compute $R = H \bmod \phi(2p^n)$
- 5: Compute $L = r^R \bmod (2p^n)$
- 6: Compute $V = r_a^{S_1} \cdot S_1^{S_2} \bmod (2p^n)$
- 7: **if** $L = V$ **then**
- 8: **Accept** the signature
- 9: **else**
- 10: **Reject** the signature
- 11: **Output:** Verification result

Appendix B. Global Parameters and Signature Results**Lena Image**

==== Global Cryptographic Parameters ====

$p=98578635836944235406119012897575144524604016751664828785795326468201239$
 $324091, r = 3, n_0 = 2, a_d = 4, b_d = 3, D = 5, A_d = 81, B_d = 27, K_d =$
 $531441, n = 3,$

Modulus= $19159245727710150457671401530015954372955466285608075233760880474$
 $7153531183762899958504392166566956562703569236380315917207264776823490894$
 $6984370532482446636522397056807764315007274742700279646422097110008511668$
 $3344902746653949771142,$

$\phi(2p^n)=957962286385507522883570076500797718647773314280403761688044023$
 $7357676559188047820450784879665393993944603323184852882322473122769769111$
 $1592069014405427098382008313389275463158077652027418334660447187304093727$
 $170504636724441909290.$

Public and Private parameters: $a = 439, b = 217, r_a = 28591097196737693$
 $977405413612190367644445996443363606354443906175091935530090594454548186$
 $729766431331404629765682940775134923457108472267, r_b = 34301433818510654$
 $479997622149056072840806381240712147713459454414705167747070180824150668$
 $119248233312163.$

==== Signature Results ====

SHA-256 hash = $1138074980937568176714277105910606034339399224460652591857$
 $90113792390808016108.$

S_1=237864303465277145025636664074824560383458424113580530311260977211776
0101071753509624651203211850599744748207457016687900764563917311433831105
3051841507073272153993278022670491620521272065108962367742359680230201373
7758919367483095,

S_2=394457341960301138444139269268838367772970632130035523964263956478197
0979366708948204414672666151525796240278566431720282886188281458303050501
1815129317939488969961470476423121064406249070921494177369033905858159197
03446049555144967,

where

w_s=332532843091198025865232001464405724539068679982480482840590853936998
5374351092500568057182377111988216905583049813372201113376585612892724592
7096721288363336487448354421281319333298336149413197302668848130879230210
5608709003714649.

Signature VALID? Yes.

Mandrill Image

==== Global Cryptographic Parameters ====

p=79462267655397755339907079368051626444417313791682220781524192143938182
295737, r = 3, n_0 = 2, a_d = 4, b_d = 3, D = 5, A_d = 81, B_d= 27, K_d=
531441, n = 3,

Modulus=10034895619058531701242733330990698899047008405109339044451846030
2549285765536010995970658678490123266511024137588907118323483425480263839
4874488363917713849772640743164341510963878451592644787283589083205189489
025663983169307761106,

$\phi(2p^n)=501744780952926585062136666549534944952350420255466952222592301$
5127464288276737407278723553201690293706586633701555219611254838302358281
3114099660094521797619931079004009183002685694648042803354088980495634741
6251223175236925507384.

Public and Private parameters: a = 28, b = 215, r_a = 22876792454961,
r_b=3811270424278961608888624683228452537867375693412460857051050490522
796416341131202683407568805359256907.

==== Signature Results ====

SHA-256 hash = 560671248945282384243348128021587571917457572465833921835
3209891157098508725.

S_1=598362139176982668019546357151367542454750629693010019779736309067640
9473621500842708508646319624809292053789481953348390577971668743354425370
7369517792921246701916133685765609857644209754778677050345998697345940106
3849809509859641,

S_2=225553830505487173563311951526949882409657729083810865144457469653587
2896152013561400595891094484071467684596947407121503389211068320736758969
9628223001558023292584643292669522724749934658315508820672390130141467542

9383695804050201,

where

w_s=723640315919957107194670881973198706384258989432874626562509943055004
9545899776492957327955692145313280514797780240787501972125015429141367157
7100032071185697925103389602252514854359103039049651317685614966543615231
05231174154161.

Signature VALID? Yes.

House Image

==== Global Cryptographic Parameters ====

p=77258579327625907531919211048839620891184193700112804934110332090141008
221489, r = 3, n_0 = 2, a_d = 4, b_d = 3, D = 5, A_d = 81, B_d = 27, K_d =
531441, n = 3,

Modulus=92229562641001021397439165099121918310466561313773813132058446227
7885000356537759647217586522964088965056406136932567767769207252329128360
4279238529414598814132309051504747541092842376927576488925746141576454845
7175064580661030638,

$\phi(2p^n)=461147813205005106987195825495609591552332806568869065660292231$
1389425001782629109355290701962557027964652204938242416237526999655142698
5209718824932493669527960997264713989904223906175930902843931235514720570
227399275412423776048.

Public and Private parameters: a = 42, b = 79, r_a = 10941898913151235920
9, r_b= 49269609804781974438694403402127765867.

==== Signature Results ====

SHA-256 hash = 1106618483246622102777663527442310933944973826037709870715
60624254181871295286.

S_1=857341617359707845257314619810561693087966774676054072452270720371197
2225053040369288145043573040184262142970575551073003620405387166137442016
5108567895203507760351510939595173744729759534099505566857176208917381583
6105032655209761,

S_2=153724149642385474558728713306563914763108048100569019318490218986785
4346724269273946260006485288530514490102506181023042129898592627933586065
1588012558826726814359854289681176139480822400896255093196863041639485738
2193274445758452,

where

w_s=193099995179266740671632924208901260176949072222872665472233131874600
0033767765279241280680565737432824291534122019219932704070766243299019391
3999712391639523470987170994165330331151473394584749504597650798081493461
6532219334464179.

Signature VALID? Yes.

Moon Image

==== Global Cryptographic Parameters ====

$p=87644027003410589950475865236450109175126180524574545329132543752555866$
 300289 , $r = 3$, $n_0 = 2$, $a_d = 4$, $b_d = 3$, $D = 5$, $A_d = 81$, $B_d = 27$, $K_d =$
 531441 , $n = 3$,

Modulus= $13464708869278007651333037042089083547619264343764727931539860274$
 $6659238151238700481446656742167179333572320668587274639972830447781604216$
 $6151794767711532945292105576999877481009744515123773985456692704879241172$
 11424157515502075138 ,

$\phi(2p^n)=673235443463900382566651852104454177380963217188236396576993013$
 $7332961907561858209317639091614081671638710864578233556979180284146612517$
 $7889176390195336007290377550570025648896052962211395860468065666254064748$
 6292879625777029554048 .

Public and Private parameters: $a = 102$, $b = 99$, $r_a = 4638397686588101979$
 $328150167890591454318967698009$, $r_b = 171792506910670443678820376588540424$
 234035840667 .

==== Signature Results ====

SHA-256 hash = $9236095690873859812168112447797402112285964879731071337695$
 7339786856624954750 .

$S_1=490168425413232430558752435533387846637858067892718548570526202700386$
 $5472178275733053327245166690012357521146749494567336308471301944987605602$
 $7069956063497157700273507438367315423615893309421306270231927292436252771$
 512091730423401 ,

$S_2=379317730766479557501658050563653718112330954284262380523685564088444$
 $9794411496826575451203878909536596327889064656430694131491379650391711724$
 $61462020494883170121349757544261956290198325380517549086220584002396087586$
 213520473977928 ,

where

$w_s=578707288233739386444477231293894738973353089730262092535685387381700$
 $5299807798701573674134701544317758026465585427512853740906011270162535086$
 $1867032906253621028536076511390291827906302024442437543436794132551412951$
 0045766943527069 .

Signature VALID? Yes.

References

1. ElGamal, T. A public-key cryptosystem and a signature scheme based on discrete logarithms. In *Advances in Cryptology—Proceedings of CRYPTO '84 (a Workshop on the Theory and Application of Cryptographic Techniques Held at the University of California, Santa Barbara, August 19–22, 1984)*; Springer: Berlin, Germany, 1985; pp. 10–18. [\[CrossRef\]](#) [\[PubMed\]](#)
2. AlSabti, K.D.; Hashim, H.R. A new approach for image encryption in the modified RSA cryptosystem using MATLAB. *Glob. J. Pure Appl. Math.* **2016**, *12*, 3631–3640.
3. Hashim, H.R.; Molnár, A.; Tengely, S. Cryptanalysis of ITRU. *Rad. Hrvat. Akad. Znan. Umjet. Mat. Znan.* **2021**, *546*, 181–193. [\[CrossRef\]](#)
4. Rivest, R.L.; Shamir, A.; Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM* **1978**, *21*, 120–126. [\[CrossRef\]](#)

5. Diffie, W.; Hellman, M.E. New directions in cryptography. *IEEE Trans. Inf. Theory* **1976**, *22*, 644–654. [CrossRef]
6. Koblitz, N. Elliptic curve cryptosystems. *Math. Comput.* **1987**, *48*, 203–209. [CrossRef]
7. Liu, Z.; Xia, T.; Wang, J. Image encryption technique based on new two-dimensional fractional-order discrete chaotic map and Menezes–Vanstone elliptic curve cryptosystem. *Chin. Phys. B* **2018**, *27*, 030502. [CrossRef]
8. Kumar, M.; Iqbal, A.; Kumar, P. A new RGB image encryption algorithm based on DNA encoding and elliptic curve Diffie–Hellman cryptography. *Signal Process.* **2016**, *125*, 187–202. [CrossRef]
9. Luo, Y.; Ouyang, X.; Liu, J.; Cao, L. An Image Encryption Method Based on Elliptic Curve ElGamal Encryption and Chaotic Systems. *IEEE Access* **2019**, *7*, 38507–38522. [CrossRef]
10. Wu, J.; Liao, X.; Yang, B. Color image encryption based on chaotic systems and elliptic curve ElGamal scheme. *Signal Process.* **2017**, *141*, 109–124. [CrossRef]
11. Li, L.; Abd El-Latif, A.A.; Niu, X. Elliptic curve ElGamal based homomorphic image encryption scheme for sharing secret images. *Signal Process.* **2012**, *92*, 1069–1078. [CrossRef]
12. Ajeena, R.K.K.; Kamarulhaili, H. On the distribution of scalar k for elliptic scalar multiplication. *AIP Conf. Proc.* **2015**, *1682*, 020052. [CrossRef]
13. Luhaib, Q.M.; Ajeena, R.K.K. Elliptic curve matrices over group ring to improve elliptic curve–discrete logarithm cryptosystems. *J. Discret. Math. Sci. Cryptogr.* **2023**, *26*, 1699–1704. [CrossRef]
14. Mohit, P.; Biswas, G.P. Modification of ElGamal Cryptosystem into Data Encryption and Signature Generation. In *Proceedings of International Conference on Big Data, Machine Learning and Applications*; Patgiri, R., Bandyopadhyay, S., Balas, V.E., Eds.; Springer: Singapore, 2021; pp. 119–129.
15. Hashim, H.R. On the solutions of $2^x + 2^y = z^2$ in the Fibonacci and Lucas numbers. *J. Prime Res. Math.* **2023**, *19*, 27–33.
16. Rosen, K.H. *Elementary Number Theory and Its Applications*, 7th ed.; Pearson: Boston, MA, USA, 2023.
17. Burton, D.M. *Elementary Number Theory*, 7th ed.; McGraw-Hill Companies, Inc.: New York, NY, USA, 2011.
18. Stein, W.; SageMath, I. CoCalc: Collaborative Calculation in the Cloud. 2024. Available online: <https://cocalc.com> (accessed on 15 July 2025).
19. University of Southern California, Signal and Image Processing Institute. USC-SIPI Image Database. 2025. Available online: <https://sipi.usc.edu/database/> (accessed on 15 July 2025).
20. Al-Saffar, N.; Alkhayyat, H.; Obaid, K. A novel image encryption algorithm involving a logistic map and a self-invertible matrix. *Malays. J. Math. Sci.* **2024**, *18*, 107–126. [CrossRef]
21. Yousif, S.F.; Abboud, A.J.; Radhi, H.Y. Robust Image Encryption with Scanning Technology, the El-Gamal Algorithm and Chaos Theory. *IEEE Access* **2020**, *8*, 155184–155209. [CrossRef]
22. Kaner, S.; Garipcan, A.M.; Erdem, E. A novel deep learning-based statistical randomness evaluation test methodology for cryptographic applications. *J. King Saud Univ. Comput. Inf. Sci.* **2025**, *37*, 264. [CrossRef]
23. Shor, P.W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Rev.* **1999**, *41*, 303–332. [CrossRef]
24. Fujisaki, E.; Okamoto, T. Secure integration of asymmetric and symmetric encryption schemes. *J. Cryptol.* **2013**, *26*, 80–101. [CrossRef]
25. Vaidehi, M.; Justus Rabi, B. Design and analysis of AES-CBC mode for high security applications. In *Proceedings of the Second International Conference on Current Trends in Engineering and Technology (ICCTET 2014)*, Coimbatore, India, 8 July 2014; pp. 499–502. [CrossRef]
26. RFC 8439; ChaCha20 and Poly1305 for IETF Protocols. RFC: Wilmington, DE, USA, 2018. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.