cryptography

MDPI

Article

A Novel and Secure Fake-Modulus Based Rabin-3 Cryptosystem

Raghunandan Kemmannu Ramesh [1], Radhakrishna Dodmane [1,*], Surendra Shetty [2], Ganesh Aithal [3], Monalisa Sahu [4,*] and Aditya Kumar Sahu [4,*]

[1] Department of Computer Science and Engineering, NMAM Institute of Technology, NITTE University, Karnataka 574110, India; raghunandan@nitte.edu.in
[2] Department of Master of Computer Applications, NMAM Institute of Technology, NITTE University, Karnataka 574110, India; hsshetty@nitte.edu.in
[3] Department of Computer Science and Engineering, SMVITM, Bantakal 574115, India; ganeshaithal@gmail.com
[4] Amrita School of Computing Amaravati, Amrita Vishwa Vidyapeetham, Amaravati 522503, India
* Correspondence: rkdodmane@gmail.com (R.D.); swetymona@gmail.com (M.S.); adityasahu.cse@gmail.com (A.K.S.)

Abstract: Electronic commerce (E-commerce) transactions require secure communication to protect sensitive information such as credit card numbers, personal identification, and financial data from unauthorized access and fraud. Encryption using public key cryptography is essential to ensure secure electronic commerce transactions. RSA and Rabin cryptosystem algorithms are widely used public key cryptography techniques, and their security is based on the assumption that it is computationally infeasible to factorize the product of two large prime numbers into its constituent primes. However, existing variants of RSA and Rabin cryptosystems suffer from issues like high computational complexity, low speed, and vulnerability to factorization attacks. To overcome the issue, this article proposes a new method that introduces the concept of fake-modulus during encryption. The proposed method aims to increase the security of the Rabin cryptosystem by introducing a fake-modulus during encryption, which is used to confuse attackers who attempt to factorize the public key. The fake-modulus is added to the original modulus during encryption, and the attacker is unable to distinguish between the two. As a result, the attacker is unable to factorize the public key and cannot access the sensitive information transmitted during electronic commerce transactions. The proposed method's performance is evaluated using qualitative and quantitative measures. Qualitative measures such as visual analysis and histogram analysis are used to evaluate the proposed system's quality. To quantify the performance of the proposed method, the entropy of a number of occurrences for the pixels of cipher text and differential analysis of plaintext and cipher text is used. When the proposed method's complexity is compared to a recent variant of the Rabin cryptosystem, it can be seen that it is more complex to break the proposed method—represented as $O(\mathfrak{n} \times \tau)$ which is higher than Rabin-P ($O(\mathfrak{n})$) algorithms.

Keywords: cryptography; differential analysis; entropy; Fermat's factorization; RSA; Rabin cryptography

Citation: Ramesh, R.K.; Dodmane, R.; Shetty, S.; Aithal, G.; Sahu, M.; Sahu, A.K. A Novel and Secure Fake-Modulus Based Rabin-3 Cryptosystem . *Cryptography* **2023**, *7*, 44. https://doi.org/10.3390/cryptography7030044

Academic Editor: Josef Pieprzyk

Received: 7 June 2023
Revised: 19 July 2023
Accepted: 31 July 2023
Published: 19 September 2023

1. Introduction

Secure transaction in e-commerce refers to the safe and secure exchange of information and money between buyers and sellers in an online marketplace. E-commerce has revolutionized the way people buy and sell goods and services, making it easy for customers to shop from anywhere in the world, at any time of the day. The convenience of online shopping has also led to the need for secure transactions to protect both buyers and sellers from online threats and fraud. However, with the growth of e-commerce, there have also been concerns about the security of online transactions. Here are some of the most common security issues of e-commerce:

- Payment Security: One of the biggest concerns for consumers when shopping online is the security of their payment information. Cybercriminals may intercept and steal sensitive data such as credit card numbers, names, and addresses. To prevent this, it's important for e-commerce websites to have strong encryption protocols to protect customer data.
- Data Privacy: Customers share a lot of personal information when they make an online purchase. This data may include names, addresses, phone numbers, and email addresses. If this data falls into the wrong hands, it can be used for identity theft or other criminal activities. Businesses must ensure that they are handling this data securely, with proper encryption, storage, and access controls.
- Phishing and Malware Attacks: Cybercriminals often use phishing and malware attacks to steal sensitive information from customers. Phishing attacks involve sending fake emails or websites that appear to be legitimate to trick customers into sharing their personal information. Malware attacks involve installing malicious software on a customer's computer to steal data. E-commerce businesses should be vigilant in monitoring for these attacks and should have strong anti-malware and anti-phishing measures in place.
- Website Security: The security of e-commerce websites is also critical to protect against hacking and data breaches. Businesses should ensure that their websites are secure with SSL/TLS encryption, firewalls, and other security measures. They should also monitor for suspicious activity, such as multiple failed login attempts.

Secure transactions in e-commerce are crucial to maintaining the trust of customers and ensuring the safety and security of online transactions. E-commerce platforms must employ various security measures to protect the sensitive information of buyers and sellers and prevent fraudulent activities. They are encryption, authentication, and secure payment systems.

Encryption ensures that sensitive information such as credit card details, passwords, and personal data are securely transmitted over the internet, making it difficult for hackers to intercept or steal such information. Authentication involves verifying the identity of users, ensuring that only authorized individuals have access to sensitive information. Secure payment systems ensure that the payment information is transmitted securely, preventing unauthorized access and fraudulent activities. This involves the use of secure payment gateways, which encrypt and process the payment information, ensuring that the transaction is secure and protected [1].

Encryption is the process of converting plaintext into a coded form, making it unreadable to unauthorized users. Public key cryptography, such as the RSA (Rivest, Shamir, and Adleman) and Rabin cryptosystems, are widely used encryption techniques that ensure electronic commerce transactions' confidentiality, integrity, and authenticity.

2. Related Work

RSA cryptography is the oldest, most used, and most efficient of the various public-key cryptosystems, developed by Rivest et al. [2] in 1978. Rivest et al. [3] first proposed the problem of factorization in the year 1978. However, RSA's security [4] cannot be guaranteed theoretically; it is a slow algorithm that can only encrypt a small amount of data simultaneously. There have been several attempts to overcome the limitations of the RSA algorithm. Michael O. Rabin [5] made one such attempt in 1979. To increase the speed of the encryption of RSA, he proposed a variant of RSA, later known as Rabin's cryptography. Rabin is essentially RSA with the optimal choice of public key exponent (e), where encryption uses integer two as the public key exponent, which takes a shorter computation time for encryption. This feature makes the Rabin cryptosystem relatively faster in encryption than Standard RSA. Rabin algorithm makes use of two keys like RSA. Here, the public key is a common modulus (n), and the private keys are the prime factors used to compute n. Hence, the security of the Rabin algorithm entirely depends on n. In cryptanalysis, determining the factors of common modulus n plays a vital role. If someone breaks the factor,

obtaining the message becomes an easy task. Using the Rabin cryptosystem, getting plaintext back from the cipher text is considered as hard as factoring. Because of this feature, Rabin cryptosystem is used in numerous research applications [6–9]. Rabin cryptography can secure e-commerce transactions by encrypting sensitive information using the public key and decrypting it using the private key.

Here is an example of a secure transaction using Rabin cryptography:

1. Alice wants to purchase a book from an online store.
2. The online store has a publicly available public key.
3. Alice uses Rabin encryption to encrypt her credit card information and other personal data using the online store's public key. This generates the ciphertext.
4. Alice sends the ciphertext to the online store.
5. The online store receives the ciphertext and uses its private key to decrypt the message.
6. The online store processes the transaction and sends a confirmation message to Alice.
7. The confirmation message is encrypted using Alice's public key.
8. Alice receives the encrypted confirmation message and uses her private key to decrypt it.

In this example, Rabin cryptography ensures that Alice's credit card information and personal data are secure during the transaction. The online store's public key encrypts Alice's information, and only the online store's private key can decrypt the ciphertext. Similarly, Alice's public key encrypts the confirmation message, and only Alice's private key can decrypt the message. This provides a secure way for Alice and the online store to exchange information without the risk of unauthorized access or interception. It's important to note that Rabin cryptography are susceptible to brute force attacks and side-channel attacks. Therefore, using a secure implementation of these algorithms and keeping the private keys secure is essential. To improve the security of the Rabin cryptosystem, researchers contributed several ideas to make the Rabin cryptosystem strong. The following section discusses enhancements made in the Rabin cryptosystem to achieve extraordinary results in security.

Williams [5] uses unique prime numbers to make the system more efficient using the quadratic residue theory and the Jacobi symbol in the decryption. This leads to obtaining a proper message back out of four decrypted values. However, this technique results in Poor performance due to the involvement of the Jacobi symbol computation in the encryption and decryption process, causing increased computational complexity and the need for extra bits, which increases cipher text overhead. The work proposed in [10] optimized the Rabin cryptosystem by using reciprocal numbers to solve Rabin's 4-to-1 situation in decryption in 1999. In this method, Encryptor calculates and sends two additional bits of information with its ciphertext to indicate the proper square root. However, it still requires more computational costs since it uses the Jacobian symbol for encryption and decryption. Lynn Margaret Batten and Hugh Cowie Williams [11] introduced a unique scheme known as the 'R-W signature scheme,' which is considered the most efficient decryption method compared to existing methods. This scheme uses the concept of the Chinese Reminder Theorem (CRT) to obtain the correct plaintext back out of 4 outcomes of the decryption algorithm using private keys α and β. In 1997, authors in [12] proposed an RSA-type system using n-adic expansions and permutation functions, showing that the proposed method is faster. The authors introduced a new concept [13] built on the hardness of factoring and pointed similarity of the trapdoor permutation of the proposed scheme with the Rabin cryptosystem. He also suggested that the proposed method is best suited for practical application by developing a hybrid encryption scheme using a new trapdoor one-way permutation. The work in [14] deals with deterministic aspects and identification problems of the Rabin cryptosystem during decryption. The paper [15] proposed a fault attack against the Rabin cryptosystem using a one-byte permutation on public key n. However, the above-discussed methods are either too complex or easy to crack.

Some researchers turn to the modulus process to improve the Rabin cryptosystem. In [16], the authors analyzed and compared three types of algebraic analysis on AAβ cryptosystem. The study includes congruence relation, which is used to solve the Aaβ equation. Continued fractions and Coppersmith's theorem are used to retrieve the factors from the equation. The authors developed an asymmetric scheme based on the integer factorization problem [IFP], including the square root scenario in [17], which is analytically proved to have 1 to 1 decryption. Mahad et al. [17] introduced a new optimized solution to correct the Rabin cryptosystem decryption failure of 4 to 1 by reducing the plaintext phase space from $x \epsilon \mathbb{Z}_{\alpha\beta}$, to $x \epsilon 2^{2n-2}, 2^{2n-1} \subset \mathbb{Z}_{\alpha\beta}$, where $\alpha\beta$ is a composite of 2 strong primes $\alpha\beta\epsilon\, 2^{2n}, 2^{2n+2}$. Also, the specified proposed [18] method makes the encryption process fast, and computation is not included much. In [19], the authors proposed two methods using common modulus $ŋ = \alpha^2\beta$. In the first cnge to $M \in Z_{\alpha\beta}$. In the second method, the range of plaintext is restricted between 0 to 2^{2n-2}. In both schemes, the authors introduced a mathematical notation to obtain actual plaintext x_i among four possible candidates x_1, x_2, x_3, x_4 which is calculated using,

$$\left(C_i - xi^2\right)/(n) = Wi \tag{1}$$

where W_i is an integer, C_i is the cipher.

All Rabin encryption variant techniques stated in the literature above, a one-time execution of modulo ŋ squaring is registered with complexity $O(ŋ^2)$. This feature of Rabin makes the system the quickest and most efficient compared to RSA. In the Rabin cryptosystem, encryption can be done using Equation (2).

$$C_i \equiv x^2 (mod\ ŋ) \tag{2}$$

However, most of the researchers majorly concentrated on the decryption side of the Rabin cryptosystem. The decryption side of the Rabin cryptosystem proposed in [11,14,15,19] uses two prime factors as the key and uses the Chinese Reminder Theorem (CRT) to obtain the plaintext. In these approaches, the decryption procedure produces four possible plaintexts, of which only one will always be correct. In addition to the correct plaintext, decryption has three false plaintext results to judge the actual answer. This is the main issue and significant disadvantage of Rabin-type algorithms. If the algorithm is used to encrypt a text message, then obtaining back in the decryption is not a difficult task. If the plaintexts are numerical values, this algorithm becomes challenging in decryption.

This limitation has been resolved in the paper [20] with a new Rabin-like cryptosystem without using the Jacobi symbol. In this approach, the decryption function needs a single prime p as the key by computing a single mod function and giving the required plaintext without any failure. In [21] work of Rabin P is assessed on the microprocessor platform in terms of runtime and energy consumption. The following points summarize the limitations of all existing Rabin cryptosystems.

- Case I: In the case of the existing works, it is easy to recover the plaintext if the intruder can efficiently factor in the public key ŋ.
- Case II: Not all the plaintexts can be used for encryption/decryption.
- Case III: It requires plaintext padding systems or sending extra bits to improve encryption and decryption.
- Case IV: Insufficient expansion of the plaintext-ciphertext ratio.

To overcome all these issues, this paper proposes a novel key generation process by applying the fake-modulus (3) concept. The remaining portions of the article are structured as follows: Section 2 introduces the background of Rabin-3 Cryptosystem and the previous security efforts. The mathematical preliminaries essential to propose the algorithm are described in Section 3. Section 4 suggests Rabin-3 cryptosystem using a fake-modulus algorithm. Section 5 explores the evaluation results and discussion. The conclusion is discussed in Section 6.

3. Mathematical Preliminaries

This section gives the preliminaries required to support the proposed methodology, which makes the decryption process more unique and robust. Also, we suggest one more RSA variant by introducing the fake-modulus principle $\mathfrak{Z}$, which improves the Rabin encryption process. This feature makes the proposed system hard to break using the factorization process.

3.1. Range of Plaintext

The proposed algorithm supports encryption and decryption functionality for a specific range of plaintext. If x is the plaintext that is to be encrypted, then the range of plaintext is defined as $\sqrt{\mathfrak{Z}} < x < \frac{\alpha^2}{2}$.

Theorem 1: *Uniqueness of Solutions in Fake-Modulus Based Rabin-*$\mathfrak{Z}$ *Cryptosystem.*

Let x denote the plaintext, and α and β represent the prime factors of $\mathfrak{n}$. For any plaintext x satisfying the condition $\sqrt{\mathfrak{Z}} < x < \frac{\alpha^2}{2}$, a unique solution exists obtained through the computation of, $C_i \equiv x^2 mod\ \mathfrak{Z}$.

Proof: Upper bound of x is $\frac{\alpha^2}{2}$ then we should have $x_1 + x_2 < \alpha^2$, which leads to the contradiction $x_1 + x_2 = \alpha^2$. Suppose if x_1 and x_2 are greater than $\frac{\alpha^2}{2}$, which gives $x_1 + x_2 > \alpha^2$ again which leads to a contradiction. Thus, one of x_1 or x_2 is always less than $\frac{\alpha^2}{2}$. Suppose $x_1 < \frac{\alpha^2}{2}$, then there exists a real number ψ_1 such that $x_1 + \psi_1 = \frac{\alpha^2}{2}$. Similarly, suppose $x_2 > \frac{\alpha^2}{2}$ then there is a real number ψ_2 such that $x_2 - \psi_2 = \frac{\alpha^2}{2}$.

$\Rightarrow (x_1 + \psi_1) + (x_2 - \psi_2) = \frac{\alpha^2}{2} + \frac{\alpha^2}{2} = \alpha^2$.

But we have $x_1 + x_2 = \alpha^2$

$\Rightarrow \psi_1 - \psi_2 = 0$

$\Rightarrow \psi_1 = \psi_2$.

$\Rightarrow$ Only one of x_1 or x_2 is always less than $\frac{\alpha^2}{2}$. Hence there exists a unique $x < \frac{\alpha^2}{2}$. □

Cipher values obtained from the proposed algorithm's encryption functionality also fall within a specific range. If C_i be the cipher value, then the range of cipher values restricted to fall within the range $0 < C_i < \mathfrak{n}$ using Equation (3)

$$C_i \equiv x^2 (mod\ \mathfrak{Z}) \tag{3}$$

The limitations specified in cases I, III, and IV can be eliminated using the fake-modulus concept, which is used to hide the public key $\mathfrak{n}$ during the time of encryption. The computation process to obtain a fake-modulus is explained as follows.

3.2. Fake-Modulus Principle

In the Rabin algorithm, let α and β are two large prime numbers, such that $(\alpha + 1) \bmod 4 \equiv 0$ and $(\beta + 1) \bmod 4 \equiv 0$. If $\mathfrak{n} = \alpha^2\beta$ then, let Fake-modulus key $\mathfrak{Z} \in \mathbb{Z}^+$ can be computed using the formula

$$\mathfrak{Z} = \mathfrak{n} + \left(\alpha^2 \times \tau\right), \tag{4}$$

where τ is the random integer that falls within the range $0 < \tau < \sqrt{\alpha}$, and the range of $\mathfrak{Z}$ should be $\frac{\alpha^2}{2} < \mathfrak{Z} < \frac{\alpha^4}{4}$. Where τ is generated using a linear feedback shift register (LFSR) falls within the range $0 < \tau < \pm\sqrt{\alpha}$ and range of $\mathfrak{Z}$ should be $\frac{\alpha^2}{2} < \mathfrak{Z} < \frac{\alpha^4}{4}$. The length of the key τ should be chosen to provide a sufficient level of security, while also ensuring that the encryption and decryption operations can be performed efficiently. A key length of 1024 bits is commonly used for the Rabin cryptosystem.

A PRNG [22,23] with a suitable seed value can be used to generate the secret key. Any integer value $\kappa \in \pm\sqrt{\alpha}$ is considered for initial seed values of LFSR. In this generation of

key sequence is based on the initial seed values $\kappa_1, \kappa_2, \kappa_3 \ldots \kappa_i$ are considered. As shown in the Figure 1, to randomize the key sequence it uses a function $f(\kappa_1, \kappa_2, \kappa_3 \ldots \kappa_i)\ mod\ \sqrt{\alpha}$. To get more randomized results, prime values is taken as the initial seed values of LFSR.

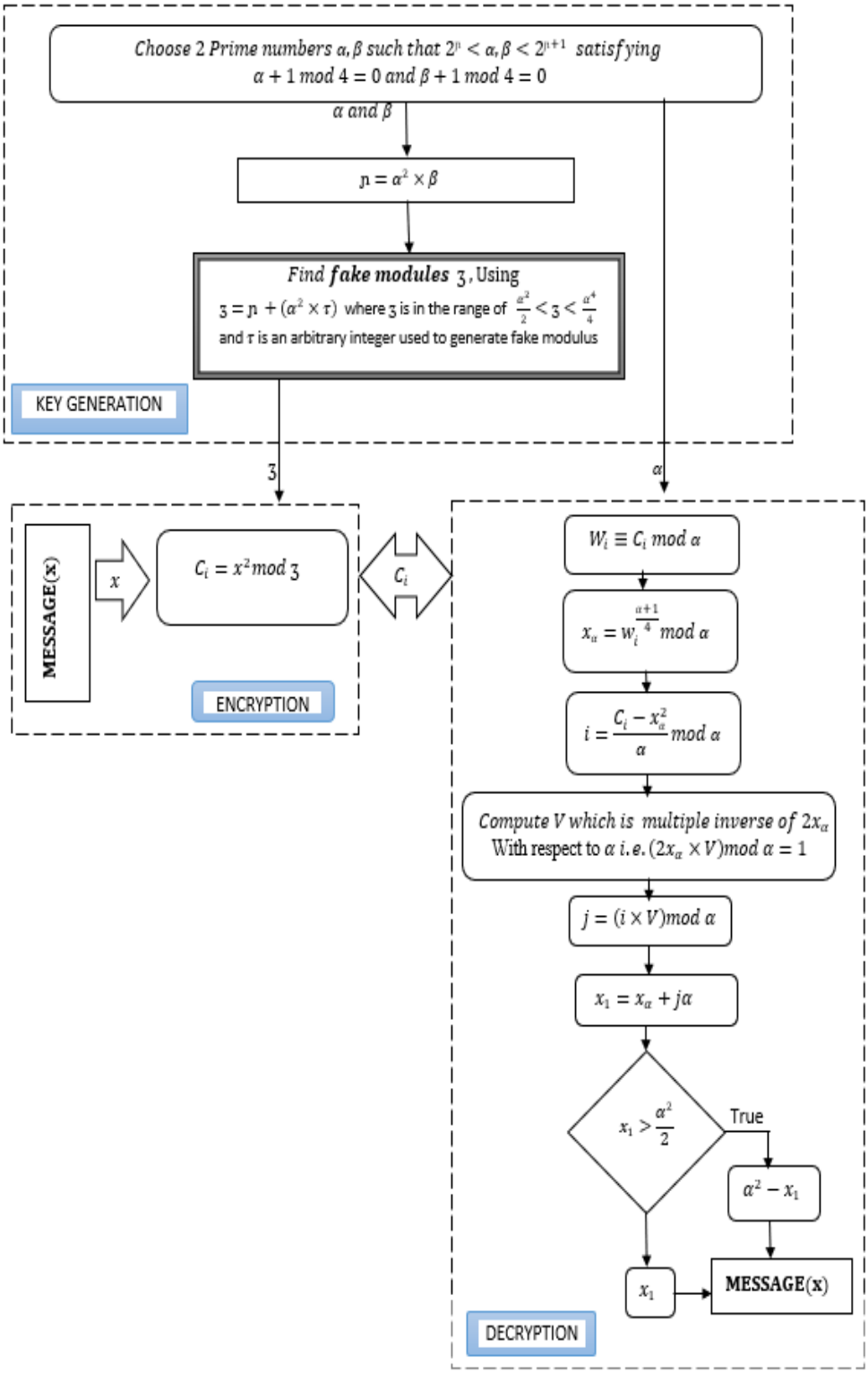


Figure 1. Block diagram of Key generation, encryption, and decryption process of Rabin-3 encryption with fake modules.

The decryption algorithm in the proposed methodology requires a single prime number as its key and performs with a single modular exponentiation process. This has more impact on the computational complexity of the proposed method over other variants. The following proof gives the justification for the methods used in decryption.

Theorem 2: *Let* $C_i \equiv x^2 mod\ \mathfrak{Z}$ *is the cipher text of Rabin. Then decryption algorithm produces a unique solution.*

Proof: Let C_i be the cipher value and α be the prime factor which is used as the private key at the decryption side. Suppose $C_i \equiv x^2 mod\ \mathfrak{Z}$ is the cipher text, and we obtain modulus $\mathfrak{n} = \alpha^2\beta$. We can write $C_i \equiv x^2 mod\ \mathfrak{Z}$ as $C_i - x^2 \equiv 0\ mod\ \mathfrak{Z}$. Then, $\frac{\alpha^2}{\mathfrak{Z}} \rightarrow \frac{\alpha^2}{C_i - x^2}$. Therefore $x < \alpha^2$, only solving is sufficient for $C_i \equiv x^2 mod\ \alpha^2$ which is effectively solved. Furthermore, there are exactly two separate x_1 and x_2 solutions that satisfy $C_i \equiv x^2 mod\ \alpha^2$. The decryption algorithm produces only a unique solution for $x < 2^{2k-1}$. Note that the upper limit of $x < \frac{\alpha^2}{2}$. Then either x_1 or x_2 is less than α^2 so $x_1 + x_2 = \alpha^2$ satisfies $x < 2^{2k-1}$. Lastly, we conclude that the decryption algorithm will produce only one unique $x < 2^{2k-1}$.

The discussed preliminaries can be readily adapted to the proposed method, which will be briefly discussed in the next section. This will encompass key generation, encryption, decryption, and will be supported by an illustrative experimental example. □

4. Methodology Proposed

The Rabin-3 cryptosystem is a variant of the Rabin cryptosystem that uses a fake modulus to reduce the computational cost of the decryption process.

This section proposes the methodology using Rabin-3 cryptosystem using the fake-modulus concept that is divided into three stages: key generation, encryption, and decryption. Specifically, the fake-modulus is chosen in the key generation step according to the selected large prime number. By this operation, we can keep α as a secret key and thus increase security. To design an efficient Rabin-3 lighter weight cryptosystem, we can consider the following steps:

4.1. Key Generation

In the process of key generation the fake modulus key $\mathfrak{Z} \epsilon\ \mathbb{Z}^+$ can be computed using the formula specified in Equation (4). The Algorithm 1 describes the process of obtaining fake modulus.

Algorithm 1: Key Generation

Input: 2 large prime numbers α and β by satisfying $(\alpha + 1)\ mod\ 4 == 0$ and $(\beta + 1)\ mod\ 4 == 0$.
Output: Fake-modulus $\mathfrak{Z}$.
Steps:

1. Select 2 large prime numbers α and β by satisfying $(\alpha + 1)\ mod\ 4 == 0$ and $(\beta + 1)\ mod\ 4 == 0$.
2. Calculate the modulus $\mathfrak{n} = \alpha^2\beta$. To hide the public key, compute fake-modulus $\mathfrak{Z}$ using function $\mathfrak{Z} = \mathfrak{n} + (\alpha^2 \times \tau)$ where $\mathfrak{Z}$ is in the range of $\frac{\alpha^2}{2} < \mathfrak{Z} < \frac{\alpha^4}{4}$ and τ is an arbitrary integer used to generate fake-modulus.
3. Share fake-modulus $\mathfrak{Z}$ as the public key to the encryption side and use α as a secret key on the decryption end.

4.2. Encryption

The encryption operation involves computing the ciphertext as the square of the plaintext modulo the fake modulus. This operation can be performed efficiently using standard modular exponentiation algorithms in Algorithm 2.

Algorithm 2: Encryption

Input: Plaintext x_i and fake-modulus $\mathfrak{Z}$.
Output: Cipher text C_i.
Steps:
Encrypt the plaintext x_i, where the range of x_i is $0 < x_i < \frac{\alpha^2}{2}$ using

$$C_i \equiv x^2 (mod\ \mathfrak{Z})$$

4.3. Decryption

The decryption operation involves computing the square roots of the ciphertext modulo, both the true and fake moduli in Algorithm 3. The square root modulo the fake modulus, can be computed efficiently using the LFSR, while the square root modulo, the true modulus, can be calculated using standard modular exponentiation algorithms. The correct plaintext can be obtained by combining the results of these computations using the Chinese remainder theorem.

Algorithm 3: Decryption

Input: Cipher text C_i and secret key α
Output: Plaintext x_i.
Steps:

1. Compute
$$w_i = C_i\ (mod\ \alpha) \tag{5}$$
2. Find
$$x_\alpha = w_i^{\frac{\alpha+1}{4}}\ (mod\ \alpha) \tag{6}$$
3. Obtain
$$i = \frac{C_i - x_\alpha^2}{\alpha}\ mod\ \alpha \tag{7}$$
4. Compute v which is multiple inverses of $2x_\alpha$ for α, i.e.,
$$(2x_\alpha * v) mod\ \alpha = (2x_\alpha * v)\ mod\ \alpha = 1 \tag{8}$$
5. Obtain
$$j = (i * v)(mod\ \alpha) \tag{9}$$
6. Compute
$$x_1 = x_\alpha + j\alpha \tag{10}$$
7. If $x_1 > \frac{\alpha^2}{2}$
$$\text{then plaintext } x = \alpha^2 - x_1 \text{ otherwise } x = x_1 \tag{11}$$

4.4. Example

In this section, we study a case of problem by assuming two prime numbers α = 263, β = 283 and plaintext x = 21,017 and use the proposed algorithm to encrypt and decrypt this plaintext.

4.4.1. Key Generation

Let α = 263 and β = 283 are the two prime numbers selected by satisfying $(\alpha + 1)\ mod\ 4 == 0$ and $(\beta + 1)\ mod\ 4 == 0$. Compute modulus $\mathfrak{n} = \alpha^2\beta = 19,574,827$. Select arbitrary integer τ within the range $0 < \tau < \sqrt{\alpha}$ and compute the fake-modulus using Equation (4). In this example, we selected $\tau = 5$ and obtained $\mathfrak{Z} = 19,920,672$. Share fake-modulus $\mathfrak{Z}$ as the public key and keep $\alpha = 263$ as secret.

4.4.2. Encryption

Using the fake-modulus $\mathfrak{Z} = 19,920,672$ perform encryption operation on plaintext $x = 21,017$. Using equation $C_i \equiv x^2 (mod\ \mathfrak{Z})$ obtain the cipher text $C_i = 3,459,505$.

4.4.3. Decryption

Upon receiving the cipher value $C_i = 3,459,505$ from the sender using private key $\alpha = 263$, the receiver follows the following steps. Compute w using Equation (5) and obtain $w = 3$. Compute $x_\alpha = 23$ using Equation (6). The value $i = 2$ was obtained using Equation (7). Compute v which is multiple inverses of $2x_\alpha$ with respect to α is computed using Equation (8) and obtained $v = 227$. Using Equation (9) calculate $j = 183$. Compute $x_1 = 48,152$ using Equation (10). According to Equation (11), the value of $x_1 > \frac{\alpha^2}{2}$ then plaintext $x = \alpha^2 - x_1$. In this case $\alpha^2 = 69{,}169$ and $x_1 = 48,152$. The difference between $\alpha^2 - x_1 = 21{,}017$, which is plaintext x.

The flow diagram of the Rabin-3 algorithm for key generation, encryption, and decryption is explored in Figure 1.

5. Cryptanalysis

In all the versions of the Rabin cryptography algorithms stated in [11,14,15,19], the public key component ŋ is shared publicly. Hence the hacker can crack the system very easily using the following two cryptanalysis methods:

- By factoring the prime numbers using Fermat's Factorization method [24]
- Breaking the plaintext using cipher value and shared public key by brute force.

The following subsections shows that Rabin-3 with fake-modulus is secure for the above two hacking strategies.

5.1. Obtaining Private Keys from Fermat's Factorization Method

Fermat factorization method known as Fermat's Difference of Squares Methods, which uses the concept of quadratic disputes.

Let ŋ be the composite number, which is written as $ŋ = \alpha^2 * \beta$; where $1 < \beta < \sqrt{ŋ}$, hence $\alpha > \beta$,

$ŋ = \left[\frac{\alpha+\beta}{2}\right]^2 - \left[\frac{\alpha-\beta}{2}\right]^2$ Where $\mathcal{S} = \frac{\alpha+\beta}{2}$, $\mathcal{T} = \frac{\alpha-\beta}{2}$ then $\alpha = \mathcal{S} + \mathcal{T}$ and $\beta = \mathcal{S} - \mathcal{T}$, it can also written as,

$$ŋ = \mathcal{S}^2 - \mathcal{T}^2$$

where,

$$ŋ = (\mathcal{S} + \mathcal{T})(\mathcal{S} - \mathcal{T}) = \alpha{\cdot}\beta \tag{12}$$

In the strategy of Fermat factorization, the algorithm searches for the value of $Y^2 - ŋ$ until it discovers an ideal root value. The search process begins from $\left|\sqrt{ŋ}\right| + 1$, $\left|\sqrt{ŋ}\right| + 2$, and so on. The above explanation demonstrates that this algorithm is guaranteed to eventually succeed in finding the factor value associated with the discovered root value. Let's utilize the Fermat method to factorize ŋ = 21,473. After determining $_i$ which is the ideal root value of ŋ, we find $\sqrt{ŋ} = \sqrt{21,473} = 146.536\ldots$ rounded to the nearest integer gives $\left|\sqrt{ŋ}\right| = 146 \rightarrow {}_0$. We initiate the process by incrementing from the initial root $_0$ until we find an integer. The progression of obtaining an integer from the ideal root value is presented in Table 1. By employing Equation (12), the factorization process can be expressed as $ŋ = (153 + 44) \times (153 - 44)$. Therefore, the factors of the given ŋ are $\alpha = 197$ and $\beta = 109$.

Table 1. Fermat's Factorization Process.

I	Y_i	Y_i^2	$Y_i^2 - n$	$T = \sqrt{Y_i^2 - n}$
1	147	21,609	136	11.661903789690601
2	148	21,904	431	20.760539492026695
3	149	22,201	728	26.981475126464083
4	150	22,500	1027	32.046840717924134
5	151	22,801	1328	36.4417343165772
6	152	23,104	1631	40.38564101261734
7	153	23,409	1936	44

Table 2 presents the outcomes of the Fermat's factorization process applied in the Rabin p algorithm, while Table 3 illustrates the utilization of Rabin 3 with the fake-modulus approach. A comparison was conducted using prime factors of various key sizes, showcasing the steps taken to factor the given modulus (ŋ), the processing time required for factorization, and the obtained factors through Fermat's factorization. The results depicted in Tables 2 and 3 demonstrate that the Rabin 3 with fake-modulus algorithms involve more steps and time to factorize the modulus (ŋ), and the resulting factors are not perfect. This observation highlights the robustness of the proposed algorithm and its ability to conceal the private key utilized in the decryption process.

Table 2. Use of Fermat's Factorization Process in Rabin-P algorithm.

Key Size	$ŋ = \alpha^2\beta$	**Steps k**	**Factoring Time in μs**	**Factors Obtained**
8	4,307,411	6631	6.2408447265625	17,161, 251
10	278,726,051	120,579	62.55626678466797	273,529, 1019
12	17,411,169,179	1,998,079	1064.565896987915	4,255,969, 4091
14	1,105,352,737,843	32,732,805	17,681.19716644287	67,551,961, 16,363
16	70,363,372,715,879	528,613,693	3,430,048.5668182373	1,073,938,441, 65,519

Table 3. Use of Fermat's Factorization Process in Rabin-3 algorithm.

Key Size	$ŋ = \alpha^2\beta$	**Steps k**	**Factoring Time in μs**	**Factors Obtained**
8	4,307,411	5899	12.034177780151367	17,161, 501
10	278,726,051	7253	11.652231216430664	50,731, 10,983
12	17,411,169,179	1137	0.8997917175292969	208,363, 167,103
14	1,105,352,737,843	818	0.7925033569335938	1,536,953, 1,438,325
16	70,363,372,715,879	12,748,236	144,303.49683761597	46,174,339, 3,047,703

5.2. Obtaining Plaintext from Cipher Text and Modulus in Rabin Cryptosystem Using Brute Force Method

Consider $M_i \in \mathbb{Z}$ as the plaintext to be encrypted using the encryption function $C_i = M_i^2 \bmod ŋ$, where $C_i \in \mathbb{Z}$ represents the ciphertext, and $ŋ_i \in \mathbb{Z}_{pq}$ serves as the modulus used as a public key in this function. Let $\kappa \in \mathbb{Z}$ be an integer that is iteratively incremented until the resulting M_i becomes an integer, using the function $M_i = \sqrt{C_i + \kappa \times ŋ}$. The following case studies provide illustrations of the process involved in recovering a message M_i from the ciphertext C_i and modulus n through a brute force attack on the Rabin cryptosystem. Figure 2 visually presents the step-by-step procedure for recovering the message M_i using the ciphertext C_i and modulus n through the brute force attack on the Rabin cryptosystem.

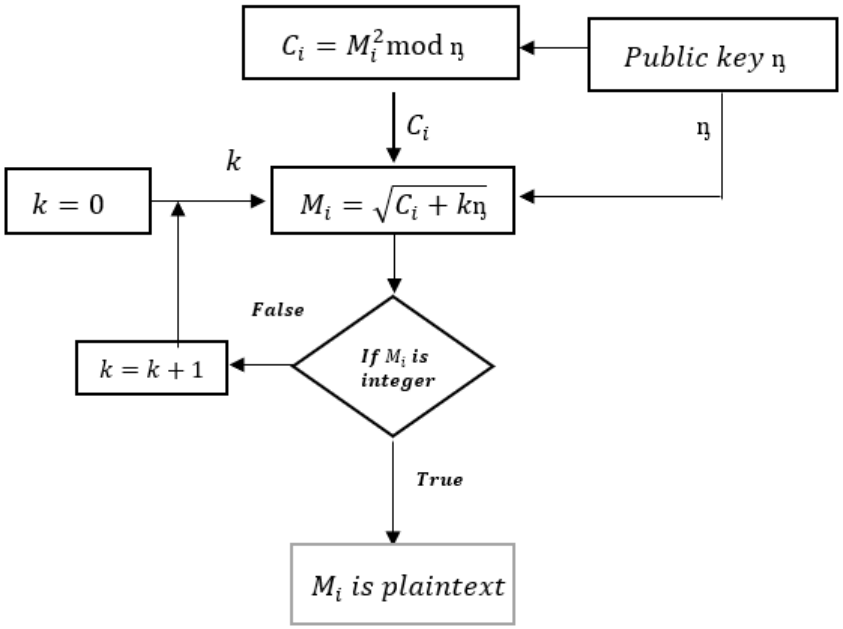


Figure 2. Flow diagram of brute-force attack on Rabin cryptosystem.

5.3. Case Study

Let $C = 11,544,473$ represent the cipher value, and the public key component $\eta = 19,574,827$. The eavesdropper, using these parameters, performs the following operation using the equation $M_i = \sqrt{C_i + \kappa \times \eta}$, where κ is the integer value that indicates the number of iterations or steps required to break the cipher. We initiate the incrementing process from 0 until we obtain an integer value for M_i. In this example, when $\kappa = 8$, the equation yields an integer value of 12,967. Since this integer value, 12,967, corresponds to the plaintext for the given cipher value, Table 4 displays the step-by-step process of obtaining the plaintext from the given cipher value.

Table 4. Process of obtaining Plaintext from the given cipher text.

k	$M_i = \sqrt{C_i + k \times n}$
0	3397.7158503912597
1	5578.467531500027
2	7119.980828625875
3	8382.657931706386
4	9478.59594032787
5	10,460.334985075764
6	11,357.52767991344
7	12,188.858108945235
8	12,967

The number of steps needed to break the plaintext using Rabin p versus Rabin 3 with fake-modulus for various key sizes is presented in Figure 3. The figure illustrates the comparison of the number of steps required to break the plaintext between Rabin p and Rabin 3 with fake-modulus for different key sizes.

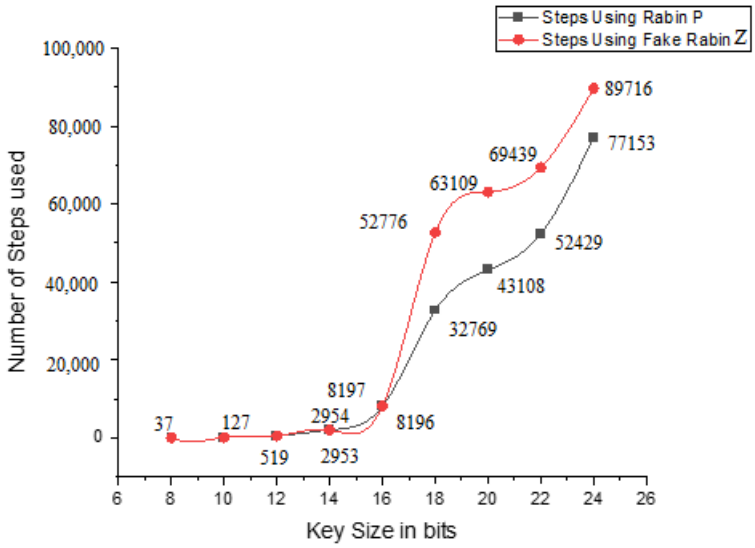


Figure 3. Number of steps required to break the plaintext using Rabin p v/s Rabin 3 using fake-modulus for different Key size.

Observations:

- It is observed that Rabin-P, with the fake-modulus approach, denoted as fake Rabin-P, requires a higher number of steps to crack the plaintext from the given ciphertext.
- The time consumption for Rabin-P and Rabin-P with the fake-modulus is approximately equivalent for prime numbers with lower bit lengths (e.g., 8, 10, and 12 bits). However, as the bit length increases beyond 16 bits, the gap between the time curves widens significantly.
- Based on the statistical comparison, it is evident that breaking the code using the proposed fake-modulus approach, demands more time and steps compared to the traditional Rabin-P algorithm.

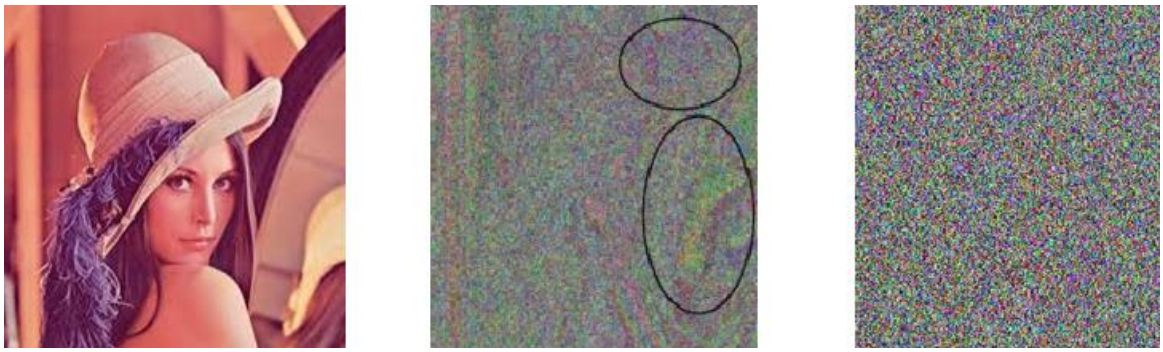
6. Results and Analysis

In this section, we aim to highlight the significance of the Rabin 3 cryptosystem in relation to the Rabin P cryptosystem, particularly regarding its robustness. We thoroughly examine the investigations conducted on the security provided by the proposed algorithms, comparing them to the Rabin P cryptosystem through performance and complexity analysis. Our objective is to showcase the importance of the proposed method in terms of its robustness compared to the Rabin P cryptosystems. To assess the significance of the proposed systems, we establish an experimental setup utilizing an existing Intel P4 CPU 1.7 GHz, 1.24 GB RAM, and the Windows 10 platform. To showcase the performance of the proposed algorithms, we applied them to a variety of test images, including Lena, and Baboon. These test images have dimensions of ($512 \times 512 \times 3$) pixels. By using these representative images, we aim to demonstrate the effectiveness and capabilities of the algorithms in different scenarios. The subsequent metrics defined herein effectively articulate the comparisons.

6.1. Visual Analysis

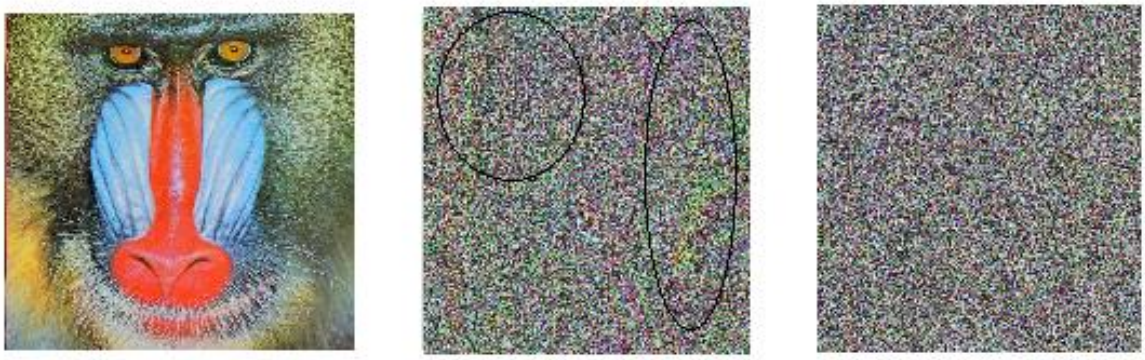
To evaluate the extent of distortion or degradation introduced during the encryption process, a visual comparison was performed between the plaintext and encrypted images. This allowed us to gauge the impact of encryption on the visual quality and fidelity of the images. If the encrypted image contains many unidentified pixels from the original image, it can be deemed secure. Figures 4b and 5b displays the encrypted image generated by the Rabin-P algorithm, wherein certain areas still exhibit evidence of the original image. However, upon employing the proposed Rabin-3 algorithm, the encrypted image (as

depicted in Figures 4c and 5c) exhibited no discernible traces of the original image. This demonstrates the algorithm's resilience against statistical attacks.



(**a**) (**b**) (**c**)

Figure 4. (**a**) Test Lena image, (**b**) Encrypted image using Rabin-P, (**c**) Encrypted image using Rabin-3.



(**a**) (**b**) (**c**)

Figure 5. (**a**) Test Baboon image ,(**b**): Encrypted image using Rabin-P, (**c**): Encrypted image using Rabin-3.

6.2. Histogram Analysis

A histogram is the schematic representation of the number of occurrences of the value of each pixel. In this work, the Lena color image shown in Figure 4a has been considered for evaluation. i.e., the numbers of occurrences of each pixel value are expressed separately for Red, Green, and Blue in the histogram. Figure 6a,b shows the histogram for the distribution of the occurrences of RGB components of the original image. From the visual analysis shown in the histogram of Figures 7 and 8, the histogram results of the Rabin-3 cryptosystem have flat and uniform pixel distribution compared to Rabin P. In the figure the x-axis represents the range of pixel values, while the y-axis represents the number of pixels in the image that fall within that range. These histogram results are significant enough to suggest that the proposed approach is cryptographically secure pixel distribution.

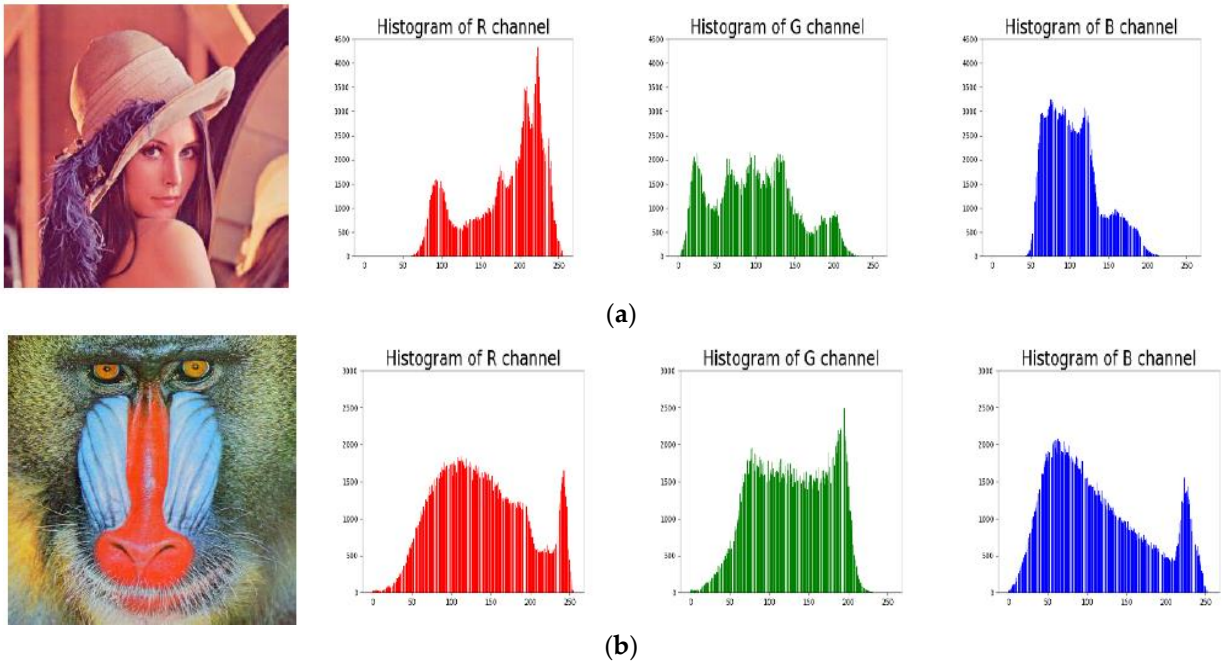


Figure 6. Number of occurrences of pixel values (*Y*-axis) vs. pixel values (*X*-axis) of (**a**) Lena image and (**b**) Baboon image with RGB components respectively.

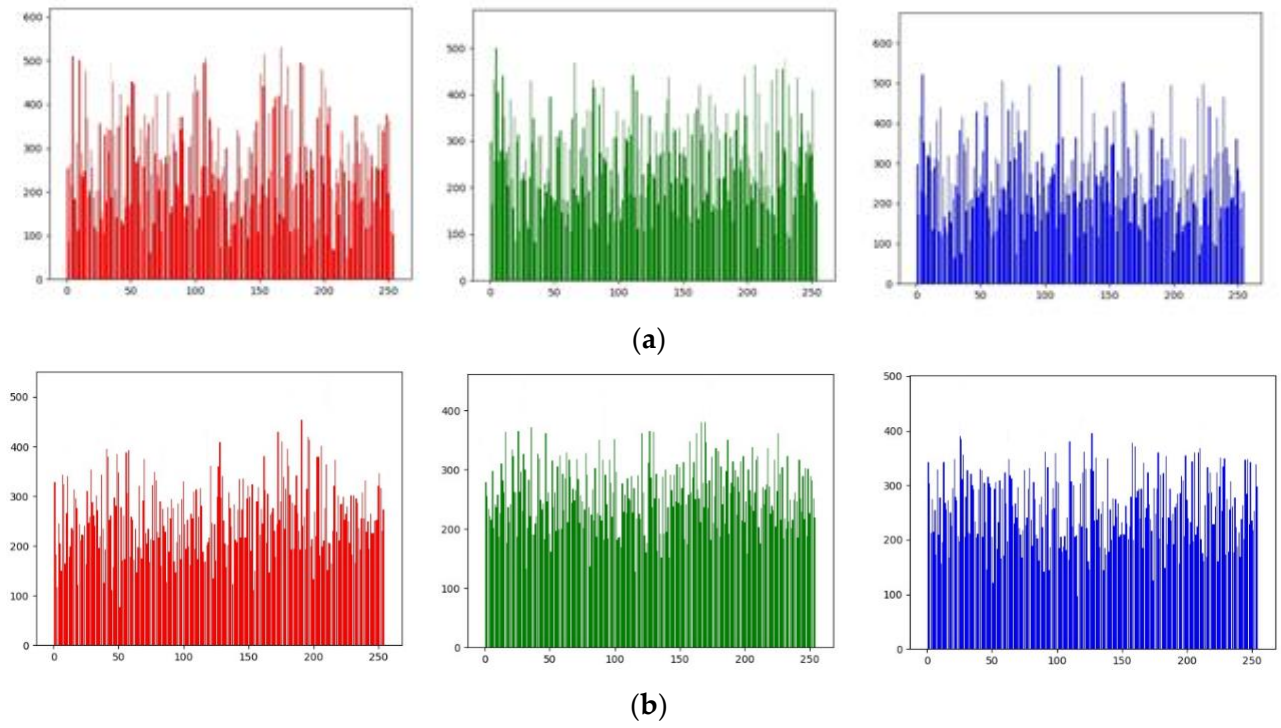


Figure 7. Number of occurrences of pixel values (*Y*-axis) v/s pixel values (*X*-axis) of cipher image of Rabin-P algorithm for (**a**) Lena image with RGB components, (**b**) Baboon image with RGB components.

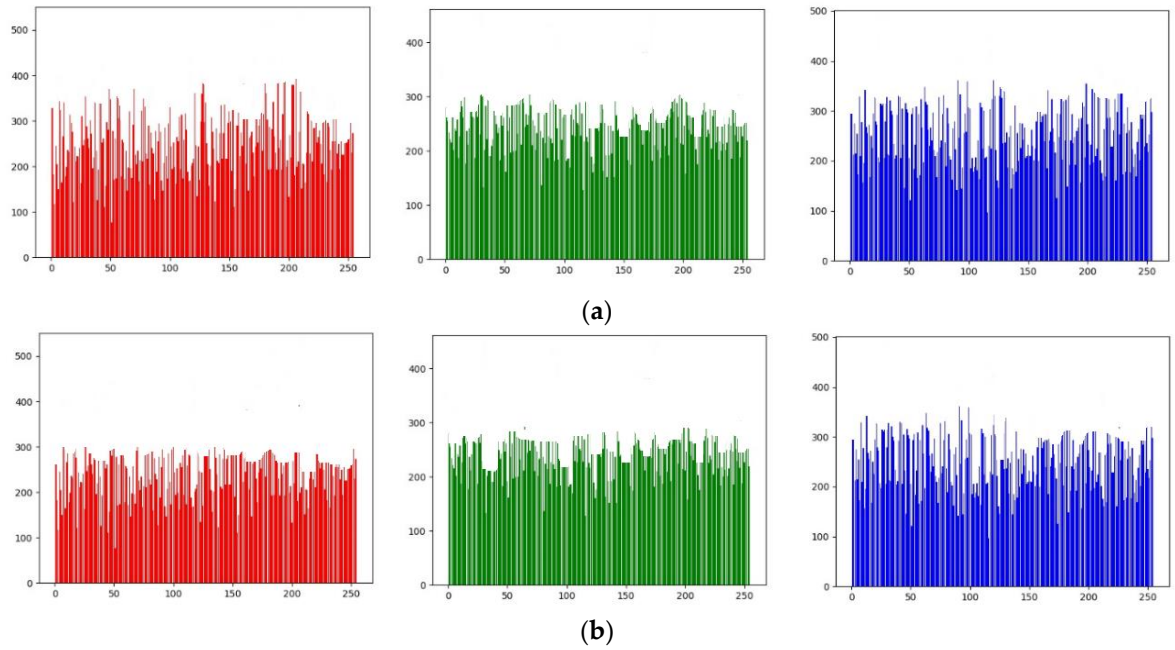


Figure 8. Number of occurrences of pixel values (Y-axis) v/s pixel values (X-axis) of cipher image of Rabin-3 algorithm for (**a**) Lena image with RGB components, (**b**) Baboon image with RGB components respectively.

6.3. Entropy Analysis

The degree of uncertainty in the system is defined as the entropy of information. The greater the entropy, the greater the image's randomness or uniformity [25]. Entropy can be mathematically defined using Equations (13) and (14). Let p_i be the probability of occurrence of pixel i in the cipher image of length N number of pixels, where $i = 0, 1, 2, \ldots, M-1$.

$$p_i = \lim_{N \to \infty} \frac{N_i}{N} \tag{13}$$

$$\text{Entropy}(\text{H}) = \lim_{\text{ŋ} \to \infty} \sum_{\text{i}=0}^{\text{M}-1} \text{p}_\text{i} \log_2 \left(\frac{1}{\text{p}_\text{i}} \right) \tag{14}$$

In the experimentation the entropy (H) of the cipher image is calculated separately for RGB components of the color image using the equation $\sum_{i=0}^{25} p_i \log_2 \left(\frac{1}{p_i} \right)$, where p_i is the probability of occurrence of the cipher values and is given by $p_i \approx \frac{N_i}{N}$, where N_i is the number of events of p_i in N different pixels of an image.

A comparison between the proposed algorithm and existing Rabin algorithms is conducted, and the resulting entropy values are presented in Table 5. The entropy values of the encrypted images generated by the proposed Rabin-3 algorithm are found to be in close proximity to 8. This signifies that the cipher image exhibits exceptional uncertainty and a significant degree of permutation and substitution effects. Consequently, it can be concluded that the proposed algorithm is highly resistant to attacks and provides a secure defense against statistical entropy attacks.

Table 5. Entropy result comparison of the number of occurrences of pixel values of proposed Rabin-3 with Rabin-P algorithm.

Methods Proposed		Entropy of RGB Components		
		Red	**Green**	**Blue**
Ref. [11]		7.59	7.68	7.71
Ref. [14]		7.65	7.70	7.68
Ref. [15]		7.73	7.76	7.72
Ref. [16]		7.71	7.73	7.70
Rabin-P algorithm [19]	(Lena)	7.63	7.71	7.75
	(Baboon)	7.68	7.74	7.69
Rabin-3 with fake-modulus	(Lena)	7.93	7.95	7.94
	(Baboon)	7.92	7.97	7.94

6.4. Differential Analysis

Differential analysis is a metric used in differential attack analysis to check the cipher resistance. When an attacker makes little changes to the original image (flipping one bit), notice the difference in the cipher image [26]. Such disparity can be calculated using two criteria: the Number of Pixel Change Rate (NPCR) [27] and the Unified Average Pixel Change Intensity (UACI) [28]. The proposed cryptosystem will guarantee two separate ciphered images, although there is only one bit of difference between them. The NPCR focuses on the total number of pixels that affect the value of differential attacks, and to evaluate the impact of the pixel change on the encrypted image using NPCR is given in (15)

$$NCPR = \left(\frac{1}{W_i H_i} \sum_{i,j=1}^{n,m} D(i,j) \right) \times 100 \tag{15}$$

$$with\ D(i,j) = 1\ if\ \mathrm{C}_1(i,j) \neq \mathrm{C}_2(i,j)\ and\ D(i,j) = 0\ if\ \mathrm{C}_1(i,j) = \mathrm{C}_2(i,j)$$

where W_i is the image width and H_i be the height. $\mathrm{C}_1(i,j)$ is the image before the change in one-bit pixel position and $\mathrm{C}_2(i,j)$ are the ciphered images after the change in one pixel of the plain image. For the pixel at the position (i,j) calculation was made $if\ \mathrm{C}_1(i,j) \neq \mathrm{C}_2(i,j)$, then set $D(i,j) = 1$ else set $D(i,j) = 0$.

UACI focuses on the average difference between two paired ciphertext images. UACI is specified in Equation (16)

$$ACI = \left(\frac{1}{L_i} \sum_{i,j=1}^{n,m} \frac{|\mathrm{C}_1(i,j) - \mathrm{C}_2(i,j)|}{255} \right) \times 100 \tag{16}$$

where L_i = length of the image, which contains the total number of pixels.

The outcomes of the NPCR and UACI equations are presented in Table 6. The analysis reveals that the encryption scheme exhibits a remarkable sensitivity to even minor modifications in the plaintext. For the proposed method, the NPCR values exceed 99%, while the UACI values surpass 33%. All measured values fall within the confidence interval of [98–99%]. While the NPCR results are comparable to those of existing Rabin algorithms, the UACI results significantly surpass them. These findings provide compelling evidence that our cryptosystem effectively safeguards against differential attacks.

Table 6. This is a table. Comparison of NPCR vs. UACI between plaintext and Cipher of Rabin-P and-3.

	NPCR			UACI		
	RED	**GREEN**	**BLUE**	**RED**	**GREEN**	**BLUE**
Ref. [11]	99.600	99.423	99.364	32.379	32.278	33.178
Ref. [14]	99.591	99.490	99.564	32.619	32.311	33.214
Ref. [15]	99.593	99.538	99.614	32.714	32.274	33.287
Ref. [16]	99.614	99.532	99.632	32.770	32.297	33.258
Rabin-P [19] (Lena)	99.619	99.629	99.636	32.799	32.382	33.284
Rabin-P [19] (Baboon)	99.608	99.587	99.478	32.798	32.492	33.01
Rabin-3 with fake-modulus (Lena)	99.641	99.638	99.646	32.957	32.300	33.310
Rabin-3 with fake-modulus (Baboon)	99.624	99.574	99.547	33.047	32.981	32.865

6.5. Complexityl Analysis

Time complexity is a statistical framework that determines the amount of time it takes to execute an algorithm. The complexity of the algorithms specified in this section is based on the number of mathematical operations involved in the function using the Newton-Raphson iteration method [29]. Table 7 summarizes the time the proposed methods took to prove the robustness in each step compared with the Rabin-P cryptosystem. To calculate modulus $\mathfrak{n} = \alpha^2\beta$ required $O(2\mathfrak{n})$ since $\mathfrak{n}$ uses multiplication operation on 3 integers twice. The equation used for computing fake-modulus $\mathfrak{Z} = \mathfrak{n} + (\alpha^2 \times \tau)$ also uses $O(2\mathfrak{n})$ for computing fake-modulus.

Table 7. Time complexity involved during Key generation, Encryption and Decryption phase of Rabin P And Rabin-3 algorithms.

Process	**Equation Used**	**Rabin-P**	**Rabin-3 Using Fake-Modulus**
Key Generation	$\mathfrak{n} = \alpha^2\beta$	$O(n^3)$	$O(n^3)$
	$\mathfrak{Z} = \mathfrak{n} + (\alpha^2 \times \tau)$	-	$O(n^3)$
Encryption	$C_i \equiv x^2 (mod\ \mathfrak{Z})$	$O(n^2\ log_2 n)$	$O(n^2\ log_2 n)$
Decryption	$w = C_i (mod\ \alpha)$	$O(n^2)$	$O(n^2)$
	$x_\alpha = C_i^{\frac{\alpha+1}{4}} (mod\ \alpha)$	$O(n^2\ log_2 n)$	$O(n^2\ log_2 n)$
	$i = \frac{C_i - x_\alpha^2}{\alpha} mod\ \alpha$	$O(3n^2)$	$O(3n^2)$
	$(2x_\alpha * v) mod\ \alpha = 1$	$O(M * 2n^2)$	$O(M * 2n^2)$
	$j = (i * v)(mod\ \alpha)$	$O(2n^2)$	$O(2n^2)$
	$x_1 = x_\alpha + j\alpha$	$O(log\ n)\ O(n)$	$O(log\ n)\ O(n)$

The proposed method Rabin-3 has a complexity of $O(\mathfrak{n}^2\ \log_2 \mathfrak{n})$ for encryption, which involves exponentiation and multiplication operations. The decryption process requires $O(n^2)$ and $O(n^2\ \log_2 n)$, operations to compute w and x_α, respectively. The complexity for computing v's multiplicative inverse is $O(M \times 2n^2)$, where M is the number of iterations required. The complexity for computing j is $O(2n^2)$ iterations, and computing x_1

requires $O(\log n)$ iterations. To break the Rabin algorithm using the equation $M_i = \sqrt{C_i + k_i \times \mathfrak{n}}$, the complexity involved relies on taking the square root in each iteration $(O(2\mathfrak{n}))$ and multiplying key k_i with $\mathfrak{n}$ requires $(O(\mathfrak{n}))$ can be specified as $O(3\mathfrak{n})$. The equation $\mathfrak{Z} = \mathfrak{n} + (\alpha^2 \times \tau)$ used for breaking the fake-modulus relies on the complexity of breaking $\mathfrak{n}$ and the key generated for the value τ using LFSR. Therefore, the complexity of breaking the Rabin algorithm using a Fake-modulus can be stated as $O(\mathfrak{n} \times \tau)$. The complexity of breaking $\mathfrak{n}$ and τ would depend on the specific techniques used to perform the attack, but in general, they would involve searching for a factor of the modulus or finding a linear relationship between the LFSR output and the fake-modulus.

Comparing the above statistics, it is observed that breaking Rabin-$\mathfrak{Z}$ using the fake-modulus is more complex than breaking Rabin-P algorithms. This makes the proposed Rabin-$\mathfrak{Z}$ with a fake-modulus more secure than the standard Rabin-P algorithm against attacks.

6.6. Randomness Analysis

In cryptographic applications that involve encrypting images, it is essential that the resulting encrypted images be immune to statistical attacks. Statistical attacks are a type of cryptanalysis technique that involves analyzing the statistical properties of the encrypted data to try to uncover information about the plaintext. The NIST statistical randomness test suite is a widely used tool for evaluating the randomness of encrypted images and other cryptographic outputs [30].

The significance level of the test should be higher than 0.01 in order to eliminate or accept the randomness of bit sequences. The results of the NIST randomness test for a 512 × 512 Lena and Baboon image are displayed in Table 8.

Table 8. NIST Encryption Test Results of proposed algorithm.

Test Name	Proposed Encryption Algorithm (Lena)	Proposed Encryption Algorithm (Baboon)	Result
Frequency	0.03427581	0.02989546	√
Block Frequency	0.02543914	0.02734212	√
Approximate Entropy	0.104512041	0.09128766	√
Linear Complexity	0.1382546	0.1087234	√
Random Excursions	0.16248531	0.10237231	√
Random Excursions Variant	0.09214753	0.10118763	√

According to tabulation results shown in Table 8, the proposed method passed (√) the randomness test when put through various tests as part of the NIST test suite. It suggests that the bit sequences generated by the method were able to pass the various statistical tests for randomness with a p-value greater than 0.01. This would indicate that the generated bit sequences are likely to be truly random [31–36].

7. Discussions

The proposed work aims to enhance the security and performance of existing Rabin cryptosystems by introducing a fake-modulus technique. The results specified that the proposed technique provides better immunity against differential attacks compared to existing Rabin type cryptosystems. It also mentions that the complexity involved in breaking the Rabin algorithm using the fake-modulus technique is higher than existing Rabin-P algorithms, making it difficult to break.

The proposed algorithm also produces a flat and uniform pixel distribution compared to existing Rabin cryptosystems, as demonstrated by the visual analysis of encrypted images and histograms. It highlights the paper's use of entropy and differential analysis to quantify the performance of the proposed method, which is considered a valuable con-

tribution to the field of electronic commerce and cryptography, as it offers a solution for ensuring secure communication in electronic commerce transactions.

8. Conclusions

Encryption using public key cryptography is widely used to ensure secure communication and protect sensitive information from unauthorized access. The proposed work aims to address the issues with existing Rabin cryptosystems by introducing a fake-modulus technique to enhance its security and performance against differential attacks. The paper presents a detailed analysis of the weaknesses of existing Rabin cryptosystems and proposes a solution that is validated through qualitative and quantitative studies. The proposed technique is shown to provide better immunity against differential attacks compared to existing Rabin cryptosystems. The complexity involved in breaking the Rabin algorithm using the fake-modulus technique is higher than existing Rabin-P algorithms, making it difficult to break. The proposed algorithm also produces flat and uniform pixel distribution compared to existing Rabin cryptosystems, as demonstrated by the visual analysis of encrypted images and histograms. The paper's use of entropy and differential analysis to quantify the performance of the proposed method is a valuable contribution to the field. The results show that the proposed algorithm provides excellent uncertainty, and its performance against differential attacks is superior to existing Rabin cryptosystems.

The proposed technique is supported by both visual and quantitative analysis, and its complexity makes it difficult to break. This work is relevant to the field of electronic commerce and cryptography, as it provides a solution for ensuring secure communication in electronic commerce transactions.

Author Contributions: Conceptualization, methodology: R.K.R. Validation, and investigation.: R.D. Resources & data curation; S.S. Writing—original draft preparation: G.A. Review and editing: M.S. supervision: M.S. and A.K.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data available on request from the authors.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cebeci, S.E.; Nari, K.; Ozdemir, E. Secure E-Commerce Scheme. *IEEE Access* **2022**, *10*, 10359–10370. [CrossRef]
2. Rivest, R.L.; Shamir, A.; Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM* **1978**, *21*, 120–126. [CrossRef]
3. Rabin, M.O. *Digitalized Signatures and Public-Key Functions as Intractable as Factorization*; Tech. Report MIT/LCS/TR-212; MIT Laboratory for Computer Science: Cambridge, MA, USA, 1979.
4. Imam, R.; Areeb, Q.M.; Alturki, A.; Anwer, F. Systematic and Critical Review of RSA Based Public Key Cryptographic Schemes: Past and Present Status. *IEEE Access* **2021**, *9*, 155949–155976. [CrossRef]
5. Williams, H. A modification of the RSA public-key encryption procedure (Corresp.). *IEEE Trans. Inf. Theory* **1980**, *26*, 726–729. [CrossRef]
6. Singh, D.; Kumar, B.; Singh, S.; Chand, S.; Singh, P.K. RCBE-AS: Rabin cryptosystem–based efficient authentication scheme for wireless sensor networks. *Pers. Ubiquitous Comput.* **2021**. [CrossRef]
7. Jain, M.; Lenka, S.K. Diagonal queue medical image steganography with Rabin cryptosystem. *Brain Inf.* **2016**, *3*, 39–51. [CrossRef]
8. Jain, M.; Kumar, A.; Choudhary, R.C. Improved diagonal queue medical image steganography using Chaos theory, LFSR, and Rabin cryptosystem. *Brain Inf.* **2017**, *4*, 95–106. [CrossRef]
9. Rachmawati, D.; Budiman, M.A. An implementation of the H-rabin algorithm in the shamir three-pass protocol. In Proceedings of the 2017 2nd International Conference on Automation, Cognitive Science, Optics, Micro Electro—Mechanical System, and Information Technology (ICACOMIT), Jakarta, Indonesia, 23–24 October 2017; pp. 28–33. [CrossRef]
10. Kurosawa, K.; Ogata, W. Efficient Rabin-type digital signature scheme. *Des. Codes Cryptogr.* **1999**, *16*, 53–64. [CrossRef]
11. Batten, L.M.; Williams, H.C. Unique Rabin-Williams Signature Scheme Decryption; Report 2019/915; Cryptology ePrint Archive: 2019. Available online: https://eprint.iacr.org/2019/915 (accessed on 30 July 2023).
12. Takagi, T. Fast RSA-type cryptosystems using n-adic expansion. In *Advances in Cryptology—CRYPTO '97*; CRYPTO 1997; Lecture Notes in Computer Science; Kaliski, B.S., Ed.; Springer: Berlin/Heidelberg, Germany, 1997; Volume 1294.

13. Schmidt-Samoa, K. A New Rabin-Type Trapdoor Permutation Equivalent To Factoring. *Electron. Notes Theor. Comput. Sci.* **2006**, *157*, 79–94. [CrossRef]
14. Elia, M.; Piva, M.; Schipani, D. The Rabin Cryptosystem Revisited. *Appl. Algebra Eng. Commun. Comput.* **2015**, *26*, 251–275. [CrossRef]
15. Kaminaga, M.; Yoshikawa, H.; Shikoda, A.; Suzuki, T. Crashing Modulus Attack on Modular Squaring for Rabin Cryptosystem. *IEEE Trans. Dependable Secur. Comput.* **2018**, *15*, 723–728. [CrossRef]
16. Asbullah, M.A.; Ariffin, M.R.K. Analysis on the AAβ cryptosystem. In Proceedings of the 5th International Cryptology and Information Security Conference 2016, CRYPTOLOGY 2016, Aksaray, Turkey, 21–22 September 2016; pp. 41–48.
17. Ariffin, M.R.K.; Asbullah, M.A.; Abu, N.A.; Mahad, Z. A New Efficient Asymmetric Cryptosystem Based on the Integer Factorization Problem. *Malays. J. Math. Sci.* **2013**, *7*, 19–37.
18. Zahari, M.; Ariffin, K.; Rezal, M. Rabin-RZ: A new efficient method to overcome Rabin cryptosystem decryption failure problem. *Int. J. Cryptol. Res.* **2015**, *5*, 11–20.
19. Zahari, M.; Muhammad Asyraf, A.; Ariffin, M.R.K. Efficient methods to overcome Rabin cryptosystem decryption failure. *Malays. J. Math. Sci.* **2017**, *11*, 9–20.
20. Asyraf, A.M.; Ariffin, K.; Rezal, M. Design of Rabin-like cryptosystem without decryption failure. *Malays. J. Math. Sci.* **2016**, *10*, 1–18.
21. Mazlisham, M.H.; Adnan, S.F.S.; Isa, M.A.M.; Mahad, Z.; Asbullah, M.A. Analysis of Rabin-P and RSA-OAEP Encryption Scheme on Microprocessor Platform. In Proceedings of the 2020 IEEE 10th Symposium on Computer Applications & Industrial Electronics (ISCAIE), Penang, Malaysia, 18–19 April 2020; pp. 292–296. [CrossRef]
22. Tutueva, A.V.; Nepomuceno, E.G.; Karimov, A.I.; Andreev, V.S.; Butusov, D.N. Adaptive chaotic maps and their application to pseudo-random numbers generation. *Chaos Solitons Fractals* **2020**, *133*, 109615. [CrossRef]
23. Bhattacharjee, K.; Das, S. A search for good pseudo-random number generators: Survey and empirical studies. *Comput. Sci. Rev.* **2022**, *45*, 100471. [CrossRef]
24. Kaur, M.; Kumar, V. A Comprehensive Review on Image Encryption Techniques. *Arch. Comput. Methods Eng.* **2020**, *27*, 15–43. [CrossRef]
25. Ruzai, W.N.A.; Ariffin, M.R.K.; Asbullah, M.A.; Mahad, Z.; Nawawi, A. On the Improvement Attack Upon Some Variants of RSA Cryptosystem via the Continued Fractions Method. *IEEE Access* **2020**, *8*, 80997–81006. [CrossRef]
26. Raghunandan, K.R.; Shetty, R.; Aithal, G. Key generation and security analysis of text cryptography using cubic power of Pell's equation. In Proceedings of the 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), Kerala, India, 6–7 July 2017; pp. 1496–1500. [CrossRef]
27. Raghunandan, K.R.; Dodmane, R.; Bhavya, K.; Rao, N.S.K.; Sahu, A.K. Chaotic-Map Based Encryption for 3D Point and 3D Mesh Fog Data in Edge Computing. *IEEE Access* **2023**, *11*, 3545–3554. [CrossRef]
28. Dodmane, R.; Rao, R.K.; Krishnaraj Rao, N.S.; Kallapu, B.; Shetty, S.; Aslam, M.; Jilani, S.F. Blockchain-Based Automated Market Makers for a Decentralized Stock Exchange. *Information* **2023**, *14*, 280. [CrossRef]
29. Zhou, N.-R.; Tong, L.-J.; Zou, W.-P. Multi-image encryption scheme with quaternion discrete fractional Tchebyshev moment transform and cross-coupling operation. *Signal Process.* **2023**, *211*, 109107, ISSN 0165-1684. [CrossRef]
30. Afolabi, A.O.; Oshinubi, K.I. Derivation of a Numerical Scheme to find any Root of any Real Number k using Newton Raphson Iterative Method. In Proceedings of the 13th iSTEAMS Multidisciplinary Conference, Accra, Ghana, 11 August 2018; pp. 107–112.
31. Sahu, A.K.; Sahu, M. Digital image steganography techniques in spatial domain: A study. *Int. J. Pharm. Technol.* **2016**, *8*, 5205–5217.
32. Hemalatha, J.; Sekar, M.; Kumar, C.; Gutub, A.; Sahu, A.K. Towards improving the performance of blind image steganalyzer using third-order SPAM features and ensemble classifier. *J. Inf. Secur. Appl.* **2023**, *76*, 103541. [CrossRef]
33. Sahu, A.K. A logistic map based blind and fragile watermarking for tamper detection and localization in images. *J. Ambient. Intell. Humaniz. Comput.* **2022**, *13*, 3869–3881. [CrossRef]
34. Rukhin, A.; Soto, J.; Nechvatal, J.; Smid, M.; Barker, E. *Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*; U.S. Department of Commerce: Washington, DC, USA, 2010.
35. Puneeth, B.R.; Raghunandan, K.R.; Bhavya, K.; Shetty, S.; Krishnaraj Rao, N.S.; Dodmane, R.; Ramya; Sarda, M.N.I. Preserving Confidentiality against Factorization Attacks using Fake-modulus (ζ) Approach in RSA and its Security Analysis. In Proceedings of the 2022 IEEE 9th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), Prayagraj, India, 2–4 December 2022; pp. 1–6. [CrossRef]
36. Wang, X.; Liu, P. A New Full Chaos Coupled Mapping Lattice and Its Application in Privacy Image Encryption. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2022**, *69*, 1291–1301. [CrossRef]