SECURED PAILLIER HOMOMORPHIC ENCRYPTION SCHEME BASED ON THE RESIDUE NUMBER SYSTEM

Daniel Asiedu[1] and Abdul-MuminSalifu[2]

[1]Department of Computer Science, Tamale Technical University, Box 3 E/R, Tamale, Ghana
[2]Department of Computer Science, C. K. T. University of Technology and Applied Sciences, Navrongo, Ghana

ABSTRACT

In this paper, we present an improved Paillier Cryptosystem for a secured data transmission based on the Residue Number System (RNS). The current state of Paillier Cryptosystem allows the computation of the plaintext from the cipher text without solving its security assumption of Decisional Composite Residuosity or the knowledge of its private keys under mathematical attacks. The proposed RNS based cryptosystem involving two stages of encryption and two stages of decryption has never been adequately studied before. This paper attempts to solve by introducing two stages of encryption and two stages of decryption. The first stage of the encryption process maintains the traditional Paillier encryption process and the second stage process is the encryption using the recommended moduli set $\{2^n + 1, 2^n, 2^n - 1, 2^{n-1} - 1\}$ *by the RNS Forward converter. At the first stage of the decryption process, our proposed RNS based reverse converter is adopted and finally, the traditional Paillier decryption process will be used at the second stage of the decryption process. Because the entire encryption technique is randomized, it can withstand chosen brute-force attacks. The suggested algorithm's security study reveals that it has a wide key space (*2^{4n} *key space*), *a high level resistance to key sensitivity attacks, and an acceptable level of resilience. In terms of security, it has been discovered that the proposed system outperforms the present algorithm.*

KEYWORDS

Cryptography, RNS, Information Security, Forward Converter, Reverse converter, Paillier Cryptosystem, Dynamic Range.

1. INTRODUCTION

Cryptography is a branch of information science that explores methods for establishing secure communication and using codes to protect plain text messages. It's when the original sender sends a message or information to the intended receiver while preventing an adversary valid access to cause any repetition. Cryptography's core idea is to allow two parties to communicate via an unsecured channel in such a way that an adversary cannot decipher what is being transmitted. Information security, often called Cryptography, is a major issue in data communication networks. This is because, the broadcast signal may travel beyond the conversing parties in both wired and wireless communications. With the correct equipment, anyone might easily intercept the data being transferred. To prevent intruders from deciphering intercepted signals, it is critical to encrypt data before transmission. Information security is extremely very important and a serious consider decisive the standard of service in information transmission. There's no such factor as excellent security; we'd like to concentrate additional on creating our information troublesome to steal and making that meaning out of it.

Constructing and analyzing cryptographic protocols which are often trounced by adversaries and in information security is often referred to as data integrity, data confidentiality, data privacy, non-repudiation, reliability, and data authentication. In Cryptology, data encryption is classified as Symmetric key cryptographic, Asymmetric key cryptographic, and Hash function. In symmetric key cryptography, both the sender and receiver make use of a single key for encryption and decryption. Commonly used ones are Block Cipher, DES (Data Encryption System), Blowfish, RC2, and Stream Cipher. Asymmetric key cryptographic uses a couple of keys for encryption and decryption process both for the sender and the receiver. Commonly used ones are RSA, DSA, PKCs, Pailliar, and Elliptic curve. Hash function instead of using predetermined keys uses mathematical equations by taking numerical data as input and produces hash message as the resulting output. Commonly used ones are MD5, RIPEMD, Whirlpool, and SHA. ATM cards, web encryption (HTTPS), computer passwords, time stamping, digital signature, and electronic commerce are some of the areas of application of cryptography.

The Paillier cryptosystem is the most generally used public-key encryption system to hide information from unauthorized access and different malicious activities due to its intensive application in e-voting, e-cash and e-commerce systems. However, in (Asiedu & Salifu, 2020), Asiedu and Salifu conducted a security risk assessment on Paillier Cryprosystem to identify threats and weaknesses. It was found that, the Paillier Cryptosystem can be broken under a series of mathematical attacks. That is, revealing the plaintext from the cipher text without solving its security assumption of Decisional Composite Residuosity (DCRA) or the knowledge of its private keys. As a result, the most valuable research is on improved Paillier public-key cryptosystems.

The Residue Number System (RNS) is an integer number system that exhibits supporting capabilities of carry-free addition, parallel computation, borrow-free subtraction, one step multiplication without considering partial product which are the difficulties to binary and decimal number system.

In this paper, the Residue Number System (RNS) is utilized to improve the Paillier public-key cryptosystem by passing the cypher text from the traditional Paillier encryption scheme through a smaller moduli set. Because the chosen moduli set is part of the private key, the key length is also increased. Also, the intractability of solving its security assumption of Decisional Composite Residuosity (DCRA) will not be used exclusively in this cryptosystem. In terms of security, the suggested system outperforms the existing system.

2. OVERVIEW OF RNS

The Residue Number System (RNS) is an integer number system that supports parallel, carry-free addition, borrow-free subtraction, and single-step multiplication with no partial product. Therefore RNS offers the properties of parallelism (Flores, 1969). The inherent properties of RNS have led to its intensive and widespread applications, such as image processing, communications, Digital Signal Processing (DSP), Fast Fourier Transform (FFT), Digital filtering, Discrete Cosine Transform (DCT), correlation, convolution, highly computing applications, and cryptography (Schoinianakis, 2020).

Nonetheless, magnitude comparison, sign detection, moduli selection, overflow detection and correction, data conversion, division, and other complex computing operations are still research problems in RNS.

Forward conversion is the process of converting a conventional number system to a residue number system, and Reverse/Backward conversion is the process of converting a residue number system to a conventional number system, both of which are accomplished using the Chinese Remainder Theorem or Mixed Radix Conversion or any variations of the two can be utilized to achieve reverse conversion.

2.1. The Algebra of RNS

RNS is defined by a set of moduli set $\{m_1, m_2, \dots, m_n\}$ that are relatively prime to each other and GCD $(m_i, m_j) = 1$ where $i \neq j$. The dynamic range $M = \prod_{i=1}^{n} m_i$ denotes total permissible numbers that can be represented by this RNS system. An integer X can be represented by the residues $(x_1, x_2, \dots, x_n)$ where $x_i = X \; mod \; m_i, m_i > x_i \geq 0 \; and \; X \in [0, M-1]$.

2.2. Residue Representation

Given any base, the RNS representation, $\{r_i\}_{i=1}^{N}$ where r_i are integers defined by a set of N equations $x = q_i m_i + r_i$. Where, $i = 1,2,3, \dots, N$ and q_i is an integer so chosen that $0 \leq r_i < m_i$. It is clear that q_i is an integer value of a quotient x/m_i which is denoted by $[\frac{x}{m_i}]$. The quantity r_i is the least positive integer (remainder) of the division of x by m_i and is represented as $x \; mod \; m_i = |x|_{m_i}$. $x = q_i m_i + r_i$ can be rewritten as $x = m_i \left[\frac{x}{m_i}\right] + |x|_{m_i}$.

Example: Given that $m_1 = 2, m_2 = 3 \; and \; m_3 = 5$, determine $[\frac{x}{m_i}]$ and the RNS representation of x if $x = 25$.

Solution: $m_1 = 2, \left[\frac{25}{2}\right] = 12$. $m_2 = 3, \left[\frac{25}{3}\right] = 8. m_3 = 5, \left[\frac{25}{5}\right] = 5$. $|x|_{m_1} = r_1 = x - m_1 \left[\frac{x}{m_1}\right] = 25 - 12 \times 2 = 1$. $|x|_{m_2} = r_2 = x - m_2 \left[\frac{x}{m_2}\right] = 25 - 8 \times 3 = 1$. $|x|_{m_3} = r_3 = x - m_3 \left[\frac{x}{m_3}\right] = 25 - 5 \times 5 = 0$. Therefore the RNS representation of 25 is $\{1,1,0\}$.

3. RELATED PREVIOUS WORKS

In 1999, a new probabilistic public cryptographic encryption method with a homomorphic property was proposed by Pascal Paillier (Fontaine & Galand, 2007; Paillier, 1999). The Paillier scheme is viewed as an extension of Okamoto-Uchiyama. The security assumption of the scheme has been proven under Decisional Composite Residuosity Assumption (DCRA). With its additive homomorphic property, the Paillier scheme has gained a lot of attention in numerous applications, such as electronic voting, machine learning on encrypted data, threshold schemes, and cloud computing (Albugmi et al., 2016; Shihab Ahmed & Zolkipli, 2016). The scheme is based on computation over $Z_{n^2}^*$, n being RSA modulus. The security of the Paillier scheme is based on the assumption that deciding nth composite residuosity: $z = y^n \; mod \; n^2$ is considered to be computationally difficult. That is, it is hard to determine whether z is n-residue modulo n^2 given n as a composite number and z as an integer.

Damgard, et al. (I. Damgård et al., 2010), in their paper titled "A Generalization of Paillier's Public-Key System with Applications to Electronic Voting", proposed a useful application of Paillier's scheme in the area of Electronic Voting. Jurik (Jurik, 2003), in his thesis titled "Paillier's original scheme. Extensions to the Paillier Cryptosystem with Applications to Cryptological Protocols", proposed some useful length flexibility. This is based on the ability to extend the plaintext space at encryption time rather than at key generation time, when the public

key is chosen, which was only available for symmetric ciphers in literature. In (Gupta & Sharma, 2013; Gupta Iti Sharma Asso, 2013), an asymmetric key encryption scheme with fully homomorphic evaluation capabilities was proposed. The operations are matrix-based, that is, the scheme consists of mapping the operations on integers to operations on a matrix. They further include a protocol that uses the proposed scheme for private data processing in clouds. (Catalano et al., 2001), this paper evaluates the hardcore bits of Paillier's new trapdoor scheme. The assumption was to prove that the least significant bit of c, $c = Class(w)$ is a hard-core bit if we assume computing residuosity classes is hard. In other words, we show that given a random $w \in Z_{n^2}^*$, if one can guess$lsb(Class(w))$ better than at random, then one can compute the whole $Class(w)$ efficiently.

The related works so far focused on the following areas of the scheme under consideration, Paillier Cryptosystem: Applications of the scheme (Jiang & Pang, 2020; Pettersen & Gjøsteen, 2016), Implementation of the scheme (Moore et al., 2014), Length Flexibility (I. B. Damgård et al., 2003) and Cloud Computing (El Makkaoui et al., 2020; Moulay et al., 2017; Papisetty, 2017).

However, much attention has not been drawn to the stability of the Paillier Cryptosystem. In other words, how can the scheme be broken without solving its security assumption of Decisional Composite Residuosity (DCRA) or using its private key parameters until Asiedu and Salifu (Asiedu & Salifu, 2020) proved that, the Paillier Cryptosystem can be broken under a series of mathematical attacks without solving its security assumption of Decisional Composite Residuosity (DCRA) or using its private key parameters.

This paper proposed a secured Paillier Cryptosystem using the inherent advantages of RNS to overcome those security challenges, making the scheme secure and robust for full utilization in the community of cryptography.

4. THE PROPOSED CRYPTOSYSTEM

Our proposed RNS based cryptosystem involves two stages of encryption and two stages of decryption. The first stage of the encryption process maintains the traditional Paillier encryption process and the second stage process is the encryption using the recommended moduli set by the RNS Forward converter. At the first stage of the decryption process, our proposed RNS based reverse converter is adopted and finally, the traditional Paillier decryption process will be used at the second stage of the decryption process. Figure 1 below demonstrate the proposed cryptosystem.

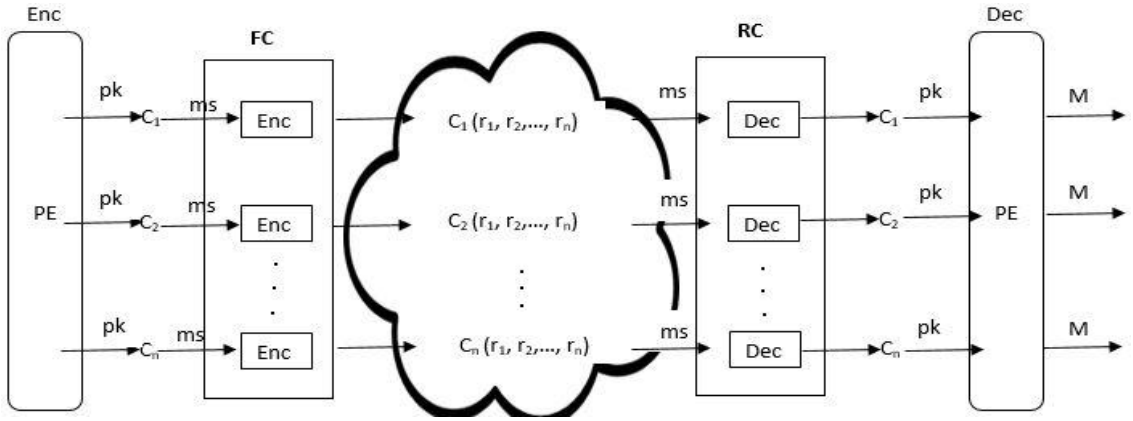

Figure 1. Proposed Cryptosystem

4.1. The Paillier Cryptosystem

Step 1: Choose two large prime numbers of equal length, p and q
Step 2: Calculate n = p * q
Step 3: Compute $\emptyset(n) = (p-1)(q-1), Note: \gcd(\emptyset(n), n) = 1$
Step 4: Compute $\lambda = lcm(p-1, q-1)$
Step 5: Choose $g \in Z_{n^2}$. Generator in most general form: $g = (1+\alpha.n)\beta^n$
Step 6: Compute $\mu = (L(g^\lambda \ mod \ n^2))^{-1} \ mod \ n, where \ L(u) = (u-1)/n$
Step 7: Public Key: (n, g)
Step 8: Private Key: (λ, μ)

Encryption:

Step 9: Choose a random integer $r \in Z_n^*, r < n$
Step 10: Plaintext $m \in Z_n, m < n$
Step 11: Ciphertext $c = g^m.r^n \ mod \ n^2, c \in Z_{n^2}$

Decryption:

Step 12: Key: (λ, μ)
Step 13: Compute $m = L(c^\lambda \ mod \ n).\mu \ mod \ n, m \in Z_n$

4.2. The Proposed Algorithm

Step 1: Key: $m = \{m_1, m_2, m_3, m_4\} = \{2^n+1, 2^n, 2^n-1, 2^{n-1}-1\} \ for \ n \ even. where \ m_1 = 2^n+1, m_2 = 2^n, m_3 = 2^n-1 \ and \ m_4 = 2^{n-1}-1.$

Encryption

Step 2: Get the first stage ciphertext $C = g^m.r^n \ mod \ n^2, c \in Z_{n^2}$ (In the usual Paillier encryption)
Step 3: Input $\{m_1, m_2, m_3, m_4\} = \{2^n+1, 2^n, 2^n-1, 2^{n-1}-1\}, the \ moduli \ set$
Step 4: Compute the second stage ciphertext $C_n, C_n\overrightarrow{RNS}$ $(|c_n|_{m_1}, |c_n|_{m_2}, |c_n|_{m_3}, |c_n|_{m_4})$.
$C_n = (r_1, r_2, r_3, r_4)$. RNS forward conversion process.

Decryption

Step 4: Key: $\{m_1, m_2, m_3, m_4\} = \{2^n+1, 2^n, 2^n-1, 2^{n-1}-1\}, the \ moduli \ set.$
Step 5: Input $C_n = (r_1, r_2, r_3, r_4)$, the resulting ciphers from Step 4.
Step 6: Compute the first stage decryption process using RNS reverse conversion below:
$X = m_1m_2m_3p + v$
$where,$
$s = (m_1|(r_2 - r_1)m^{-1}|_{m_2} + r_1)$
$v = m_1m_2(|((r_3 - s).(m_1m_2)^{-1})|_{m_3}) + s$
$p = |(r_4 - v)(m_1m_2m_3)^{-1}|_{m_4}$
X, the decimal equivalent, now becomes the ciphertext obtained from the Paillier encryption process.

Step 7: Second stage decryption, $m = L(x^{\lambda} \bmod n).\mu \bmod n, m \in Z_n$. In the usual Paillier decryption

Only the second stage encryption and first stage decryption processes are considered in the implementation below.

4.3. Stage-2 Encryption Scheme Implementation (Forward Conversion)

At stage-2, the cipher text from stage-1 (Paillier encryption method) has to be encrypted again using the forward converter with the proposed moduli set $\{2^n + 1, 2^n, 2^n - 1, 2^{n-1} - 1\}$ $for\ n\ even, where\ m_1 = 2^n + 1, m_2 = 2^n, m_3 = 2^n - 1, m_4 = 2^{n-1} - 1$. Each of the ciphertext from stage-1 (Paillier encryption method) would go through these channels as second layer encryption as follows:

$$C_1\overrightarrow{RNS}\ (|c_1|_{m_1}, |c_1|_{m_2}, |c_1|_{m_3}, |c_1|_{m_4})$$
$$C_2 = (r_1, r_2, r_3, r_4)$$
$$\vdots$$
$$C_n\overrightarrow{RNS}\ (|c_n|_{m_1}, |c_n|_{m_2}, |c_n|_{m_3}, |c_n|_{m_4})$$
$$C_n = (r_1, r_2, r_3, r_4)$$

Figure 2 below demonstrate the Stage-2 Encryption Scheme Implementation

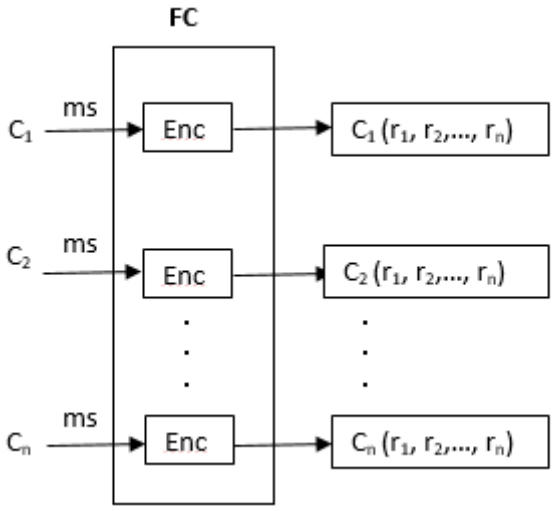

Figure 2. Stage-2 Encryption Scheme process

4.4. Stage-1 Decryption Process (Reverse Conversion)

The second stage deciphering process of the proposed cryptosystem is accomplished using the proposed moduli set $\{2^n + 1, 2^n, 2^n - 1, 2^{n-1} - 1\}$ $for\ n\ even, where\ m_1 = 2^n + 1, m_2 = 2^n, m_3 = 2^n - 1, m_4 = 2^{n-1} - 1$. The residues (ciphers) in stage-2 encryption stage (forward conversion) have to be converted to their stage-1 (Paillier encryption method) encryption cipher(s) (decimal equivalent). This is achieved by using the proposed new reverse converter as follows:

For two moduli set, we have:

$$X = m_1 p + r_1$$

$$where,$$
$$p = |(r_2 - r_1)m^{-1}|_{m_2}$$
X is the decimal equivalent.

For three moduli set, we have:

$$X = m_1 m_2 p + s$$
$$where,$$
$$s = (m_1|(r_2 - r_1)m^{-1}|_{m_2} + r_1)$$
$$p = |(r_3 - s)(m_1 m_2)^{-1}|_{m_3}$$
X is the decimal equivalent.

For four moduli set, we have:

$$X = m_1 m_2 m_3 p + v$$
$$where,$$
$$s = (m_1|(r_2 - r_1)m^{-1}|_{m_2} + r_1)$$
$$v = m_1 m_2(|((r_3 - s).(m_1 m_2)^{-1})|_{m_3}) + s$$
$$p = |(r_4 - v)(m_1 m_2 m_3)^{-1}|_{m_4}$$
X is the decimal equivalent.

For five moduli set, we have:

$$X = m_1 m_2 m_3 m_4 p + v$$
$$where,$$
$$s = (m_1|(r_2 - r_1)m^{-1}|_{m_2} + r_1)$$
$$q = m_1 m_2(|((r_3 - s).(m_1 m_2)^{-1})|_{m_3}) + s$$
$$v = m_1 m_2 m_3(|(r_4 - q).(m_1 m_2 m_3)^{-1}|_{m_4}) + q$$
$$p = |(r_5 - v)(m_1 m_2 m_3 m_4)^{-1}|_{m_5}$$
X is the decimal equivalent.

So the residues (ciphers) in Stage-2 encryption have to be converted back to stage-1 cipher using the reverse converter for the four moduli set above.

Figure 3 below demonstrate the Stage-1 Decryption implementation.

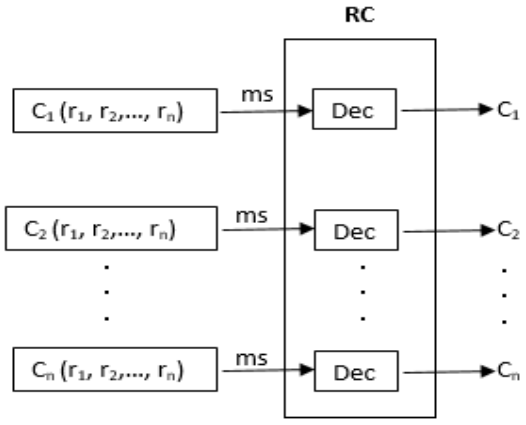

Figure 3. Stage-1 decryption process

Illustration with the Proposed Cryptosystem

Encryption Process (Stage - 2) Using the Moduli Set $\{2^n + 1, 2^n, 2^n - 1, 2^{n-1} - 1\}$.

Let p=13, q=11 then n=143, n^2 = 20449, g = 144 and r = 1. The messages 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10 using the stage-one encryption process (Paillier tradition encryption method) have 144, 287, 430, 573, 716, 859, 1002, 1145, 1288, and 1431 as their respective cipher text.

At stage-two, the cipher text from stage-one has to be encrypted again using the forward converter with the proposed moduli set $\{2^n + 1, 2^n, 2^n - 1, 2^{n-1} - 1\}$ $for\ n\ even$. Setting n=4, our proposed moduli set becomes $\{2^4 + 1, 2^4, 2^4 - 1, 2^{4-1} - 1\} = \{17, 16, 15, 7\}$. Each of the cipher text from stage-one (Paillier traditional encryption method) would go through these channels as second layer encryption as follows:

$$C_1 \overrightarrow{RNS}\ (|c_1|_{m_1}, |c_1|_{m_2}, |c_1|_{m_3}, |c_1|_{m_4})$$
$$C_2 = (r_1, r_2, r_3, r_4)$$
$$144\ \overrightarrow{RNS}\ (|144|_{17}, |144|_{16}, |144|_{15}, |144|_7)$$
$$C_2 = (8,0,9,4)$$
$$287\ \overrightarrow{RNS}\ (|287|_{17}, |287|_{16}, |287|_{15}, |287|_7)$$
$$C_2 = (15,15,2,7)$$
$$430\ \overrightarrow{RNS}\ (|430|_{17}, |430|_{16}, |430|_{15}, |430|_7)$$
$$C_2 = (5,14,10,3)$$
$$\vdots$$
$$1431\ \overrightarrow{RNS}\ (|1431|_{17}, |1431|_{16}, |1431|_{15}, |1431|_7)$$
$$C_2 = (3,7,6,3)$$

In summary, we have:

Table 1. Stage – 1 encryption ciphers

Message m	Stage-1 cipher text C_1	Stage-2 cipher text C_2
1	144	(8,0,9,4)
2	287	(15,15,2,0)
3	430	(5,14,10,3)
4	573	(15,13,3,6)
5	716	(2,12,11,2)
6	859	(9,11,4,5)
7	1002	(16,10,12,1)
8	1145	(6,9,5,4)
9	1288	(13,8,13,0)
10	1431	(3,7,6,3)

Now, C_2 are the secured cipher text which will be transmitted.

Decryption Process (Stage - 1) using the Proposed Reverse Converter for Four Moduli set in section 4.2.

The second stage deciphering process of the proposed cryptosystem is accomplished using the proposed moduli set $\{2^n + 1, 2^n, 2^n - 1, 2^{n-1} - 1\}$ $for\ n\ even$. Setting n=4, our proposed moduli set becomes $\{2^4 + 1, 2^4, 2^4 - 1, 2^{4-1} - 1\} = \{17, 16, 15, 7\}$. The residues (ciphers) in stage-

2 have to be converted to their stage-1 encryption cipher(s) (decimal equivalent). So the residues in table 1, column C_2 (Stage-2 cipher text) have to be converted back to stage-1 ciphers as represented in table 1, column C_2 using the proposed new reverse converter with four moduli set as follows:
For four moduli set, we have:

$$X = m_1 m_2 m_3 p + v$$
$$where,$$
$$s = (m_1 | (r_2 - r_1) m^{-1} |_{m_2} + r_1)$$
$$v = m_1 m_2 (| ((r_3 - s).(m_1 m_2)^{-1}) |_{m_3}) + s$$
$$p = |(r_4 - v)(m_1 m_2 m_3)^{-1}|_{m_4}$$

X is the decimal equivalent.

For $C_2 = (8,0,9,4)$, we have :

$m_1 = 17, m_2 = 16, m_3 = 15, m_4 = 7, r_1 = 8, r_2 = 0, r_3 = 9, r_4 = 4.$

$X = 17.16.15p + v.$

Where,

$$s = (17|(0 - 8)17^{-1}|_{16} + 8)$$
$$s = (17|-8.1|_{16} + 8)$$
$$s = (17.8 + 8)$$
$$s = 144$$
$$v = 17.16(|((9 - 144).(17.16)^{-1})|_{15}) + 144$$
$$v = 272(|(-135.8)|_{15}) + 144$$
$$v = 272(|-1080|_{15}) + 144$$
$$v = 272.0 + 144 = 144$$
$$p = |(4 - 144)(17.16.15)^{-1}|_7$$
$$p = |(-140)(4080)^{-1}|_7$$
$$p = |-140.6|_7 = 0$$

$X = 17.16.15(0) + 144.$

$X = 144$

:

For $C_2 = (3,7,6,3)$, we have:

$m_1 = 17, m_2 = 16, m_3 = 15, m_4 = 7, r_1 = 3, r_2 = 7, r_3 = 6, r_4 = 3.$

$X = 17.16.15p + v.$

Where,

$$s = (17|(7 - 3)17^{-1}|_{16} + 3)$$
$$s = (17|4.1|_{16} + 3)$$
$$s = (17.4 + 3)$$
$$s = 71$$
$$v = 17.16(|((6 - 71).(17.16)^{-1})|_{15}) + 71$$
$$v = 272(|(-65.8)|_{15}) + 71$$
$$v = 272(|-520|_{15}) + 71$$
$$v = 272.5 + 71 = 1431$$
$$p = |(3 - 1431)(17.16.15)^{-1}|_7$$
$$p = |(-1428)(4080)^{-1}|_7$$
$$p = |-1428.6|_7 = 0$$
$$p = |-8568|_7 = 0$$

$X = 17.16.15(0) + 1431.$

X = 1431

5. PERFORMANCE EVALUATION OF THE PROPOSED CRYPTOSYSTEM WITH THE MODULI SET $\{2^n + 1, 2^n, 2^n - 1, 2^{n-1} - 1\}$

5.1. Key Space Analysis

A key space, also known as a keysapce, is the collection of all valid, feasible, and different keys in a cryptosystem. The size of the key space is proportional to the security of a cryptosystem. Because an attacker will try to brute force the message with all conceivable key combinations, an intercepted communication with a wider keyspace will be more resistant to attackers' decoding efforts. The more permutations there are, the more secure the encryption scheme becomes. With a key of length n bits, there are 2^n possible keys. As n increases, this number climbs exponentially.

With the propose cryptosystem, the dynamic range, which is dependent on the moduli set, limits the valid choice of number representation in RNS. When the moduli set is chosen to have a tiny dynamic range, the algorithm is limited to only a few values that are qualified for stage-2 encryption, making it easy for attackers to crack the system. Hence, the proposed cryptosystem uses a moduli sets $\{2^n + 1, 2^n, 2^n - 1, 2^{n-1} - 1\}$ with dynamic range of 4n-bit, note that, n is the product of p and q (Paillier encryption process) which adapt 1024bits key space. Hence our key space is 4096bits ($2^{4n} \Rightarrow 2^{4096}$) which is greater than the Diffie and Hellman "brute force" attack of 56bits (2^{56}) possible keys combination of choices (Diffie & Hellman, n.d.). Because the moduli set is part of the private component of the classical Paillier cryptosystem, the key space in the proposed system is increased. Moreover, the proposed moduli set is sufficiently large enough and best fit design for multiplicative subgroup $Z^*_{n^2}$ for the cipher space of the stage-1 encryption ciphers.

Table 2. Key Space Analysis

Key Size n	Paillier Cryptosystem Key Space 2^n	Proposed Cryptosystem Key Space 2^{4n}
56	2^{56}	2^{224}
128	2^{128}	2^{512}
512	2^{512}	2^{2048}
1024	2^{1024}	2^{4096}
2048	2^{2048}	2^{8192}

For all of the described techniques, Table 2 displays the probability for brute force attacks, i.e., breaking the algorithm through trial and error to obtain the key by utilizing automated software to make a high number of consecutive guesses. The lower the probability of this attack, the larger the key space. To obtain a big key space, a good cryptography technique should have a large key size. The proposed cryptosystem requires exponentially more work ($2^{4n}\ key\ space$) to brute force attacks which is impractical as compare to the traditional Paillier cryptosystem ($2^n\ key\ space$).

5.2. Key Sensitivity Analysis

The goal of key sensitivity analysis is to see how sensitive an encryption method is to changes in initial conditions. It means that changing the encryption key will result in a completely different cipher. A good encryption algorithm must be sensitive to the key it uses. To put it another way,

changing just one bit of the key must result in an entirely different decoded message than the original.

RNS with a moduli set $\{m_1, m_2, m_3, \ldots, m_n\}$ has a dynamic range of $m_1 * m_2 * m_3 \ldots * m_n$. For instance, RNS (3, 5) can represent 15 unique values. Considering the moduli used, the interval [0, 14] is an absolute choice. There exist some ambiguities in any given fixed length number representation, which is demonstrated in the table 3 below:

Table 3. RNS Representation

	USING RNS(2,3) M = 2 x 3 = 6						
		A	**B**	**C**	**D**	**E**	**F**
1	**n**	0	1	2	3	4	5
	r	(0,0)	(1,1)	(2,0)	(0,1)	(1,0)	(2,1)
2	**n**	6	7	8	9	10	11
	r	(0,0)	(1,1)	(2,0)	(0,1)	(1,0)	(2,1)
3	**n**	12	13	14	15	16	17
	r	(0,0)	(1,1)	(2,0)	(0,1)	(1,0)	(2,1)
4	**n**	18	19	20	21	22	23
	r	(0,0)	(1,1)	(2,0)	(0,1)	(1,0)	(2,1)
5	**n**	24	25	26	27	28	29
	r	(0,0)	(1,1)	(2,0)	(0,1)	(1,0)	(2,1)
k	n_k	...	...	...	...	...	...
	r_k	...	...	...	...	...	...

From Table 3 above, the interval [0, 5] is an obvious choice for permissible number representation, which is in row one (1). From column A: 6, 12, 18, 24 . . . n_k as in the order stated, has equal residue representation as zero (0). From column B: 7, 13, 19, 25 . . .n_k as in the order stated, has equal residue representation as one (1). From column C: 8, 14, 20, 26 . . . n_k as in the order stated, has equal residue representation as two (2), likewise column D to F. What these means is that, without the knowledge of the moduli set (a private key component) in question or used, one has to guess from infinite numbers which have the same residue representation as the absolute choice of the required residues, which then becomes a very good feature for security purposes.

For instance, when (0,0) is transmitted as indicated in table 3, the attacker gets confused as to the exact number for that residue representation since we have infinite numbers for the same residue as demonstrated in Column A. This means that, a slight change in the parameters of the moduli set leads to a dramatic change in the resulting residues. For such a moduli sensitivity, RNS becomes a strong security parameter because an attacker is left with an unlimited number of guesses to get the correct number for such residue representation without the knowledge of the moduli set used which takes infinite years of computations.

Table 4. State of art Results Evaluation

Message (m)	Stage-one ciphers (C_1) Paillier Encryption	Stage-two ciphers (C_2) Proposed Algorithm
1	144	(8,0,9,4)
2	287	(15,15,2,0)
3	430	(5,14,10,3)
4	573	(12,13,3,6)
5	716	(2,12,11,2)
6	859	(9,11,4,5)
7	1002	(16,10,12,1)
8	1145	(6,9,5,4)
9	1288	(13,8,13,0)
10	1431	(3,7,6,3)

In the table 4 above, transmitting all the plaintexts (messages) are secured because the second level of encryption (proposed algorithm) will transform C_1 into C_2. This resolves the security vulnerabilities of the Paillier cryptosystem asserted in (Asiedu & Salifu, 2020), making the scheme fully functional in real world applications such as e-voting systems, e-cash systems and its related applications.

6. CONCLUSION

An improved Paillier cryptosystem have been implemented based on the Residue Number System. The improved cryptosystem of the tradition Paillier cryptosystem have two stages of encryption and decryption. The first stage is the traditional Paillier encryption process and the second stage is to pass the cypher text obtained from Paillier encryption process into moduli (forward conversion) to prevent the computation of the plaintext from the cipher text without solving its security assumption of Decisional Composite Residuosity or the knowledge of its private keys under mathematical attacks. The key length is also enhanced with the key space of 4096-bits ($2^{4n} \Rightarrow 2^{4096}$) as the moduli are part of the private key component of the proposed cryptosystem. The security of the cryptosystem is proportional to the length of the private key. This will help reduce the vulnerability to attacks like brute force. The key sensitivity analysis offers strong resistance to Brute-force and key sensitivity attacks.

7. FUTURE DIRECTIONS

- Further research is of interest in computational time and cost since the proposed cryptosystem focused on improving the security robustness of the Paillier Cryptosystem.
- Since RNS is demonstrating the very promising security robustness of Paillier Homomorphic Encryption Scheme, similar applications are recommended for other public-key cryptosystem exhibiting security threats.
- How homomorphic computation (additive or multiplicative) could be carried out after the stage-two encryption phase is recommended for future work.

REFERENCES

[1] Albugmi, A., Alassafi, M. O., Walters, R., & Wills, G. (2016). Data security in cloud computing. 5th International Conference on Future Generation Communication Technologies, FGCT 2016, August, 55–59. https://doi.org/10.1109/FGCT.2016.7605062

[2] Asiedu, D., & Salifu, A.-M. (2020). Security Evaluation of Pailiar Homormophic Encryption Scheme. Asian Journal of Research in Computer Science, 6(3), 12–17. https://doi.org/10.9734/ajrcos/2020/v6i330159

[3] Catalano, D., Gennaro, R., & Howgrave-Graham, N. (2001). The bit security of paillier's encryption scheme and its applications. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2045, 229–243. https://doi.org/10.1007/3-540-44987-6_15

[4] Damgård, I. B., Jurik, M. J., Brics, M. J. J., Damgård, I., & Jurik, M. (2003). A length-flexible threshold cryptosystem with applications. Springer. https://link.springer.com/chapter/10.1007/3-540-45067-X_30

[5] Damgård, I., Jurik, M., & Nielsen, J. B. (2010). A generalization of Paillier's public-key system with applications to electronic voting. International Journal of Information Security, 9(6), 371–385. https://doi.org/10.1007/S10207-010-0119-9

[6] Diffie, W., & Hellman, M. E. (n.d.). Rivest 2014 L14.1 Paper New Directions in Cryptography Invited PapDiffie, W., & Hellman, M. E. (n.d.). Rivest 2014 L14.1 Paper New Directions in Cryptography Invited Paper. 29–40.er. 29–40.

[7] El Makkaoui, K., Ezzati, A., Beni-Hssane, A., & Ouhmad, S. (2020). Fast Cloud–Paillier homomorphic schemes for protecting confidentiality of sensitive data in cloud computing. Journal of

Ambient Intelligence and Humanized Computing, 11(6), 2205–2214. https://doi.org/10.1007/S12652-019-01366-3

[8] Flores, I. (1969). Residue Arithmetic and Its Application to Computer Technology (Nicholas S. Szabo and Richard I. Tanaka). SIAM Review, 11(1). https://doi.org/10.1137/1011027

[9] Fontaine, C., & Galand, F. (2007). A survey of homomorphic encryption for nonspecialists. Eurasip Journal on Information Security, 2007. https://doi.org/10.1155/2007/13801

[10] Gupta, C. P., & Sharma, I. (2013). A fully homomorphic encryption scheme with symmetric keys with application to private data processing in clouds. 2013 4th International Conference on the Network of the Future, NoF 2013, August 2018. https://doi.org/10.1109/NOF.2013.6724526

[11] Gupta Iti Sharma Asso, C. P. (2013). Fully Homomorphic Encryption Scheme with Symmetric Keys. http://arxiv.org/abs/1310.2452

[12] Jiang, C., & Pang, Y. (2020). Encrypted images-based reversible data hiding in Paillier cryptosystem. Multimedia Tools and Applications, 79(1–2), 693–711. https://doi.org/10.1007/S11042-019-07874-W

[13] Jurik, M. J. (2003). Extensions to the paillier cryptosystem with applications to cryptological protocols. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.67.9647&rep=rep1&type=pdf

[14] Moore, C., O'Neill, M., … E. O.-… I. I., & 2014, undefined. (2014). Practical homomorphic encryption: A survey. Ieeexplore.Ieee.Org, 2792–2795. https://doi.org/10.1109/ISCAS.2014.6865753

[15] Moulay, D., Ouadghiri, E., Hassan, N., Ibtihal, M., & Driss, E. O. (2017). Homomorphic encryption as a service for outsourced images in mobile cloud computing environment. Igi-Global.Com. https://doi.org/10.4018/IJCAC.2017040103

[16] Paillier, P. (1999). Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. BT - Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding. 223–238. https://doi.org/10.1007/3-540-48910-X_16%0Ahttps://www.wikidata.org/entity/Q56287504

[17] Papisetty, S. (2017). Homomorphic Encryption: Working and Analytical Assessment: DGHV, HElib, Paillier, FHEW and HE in cloud security. https://www.diva-portal.org/smash/record.jsf?pid=diva2:1082551

[18] Pettersen, N., & Gjøsteen, K. (2016). Applications of Paillier s Cryptosystem. https://ntnuopen.ntnu.no/ntnu-xmlui/bitstream/handle/11250/2410268/15986_FULLTEXT.pdf

[19] Schoinianakis, D. (2020). Residue arithmetic systems in cryptography: a survey on modern security applications. Journal of Cryptographic Engineering, 10(3). https://doi.org/10.1007/s13389-020-00231-w

[20] Shihab Ahmed, H. A., & Zolkipli, M. F. (2016). Data Security Issues in Cloud Computing: Review. International Journal of Software Engineering and Computer Systems, 2(February), 58–65. https://doi.org/10.15282/ijsecs.2.2016.5.0016