DOI https://doi.org/10.24297/ijct.v24i.9606

A NEW ROBUST HOMOMORPHIC ENCRYPTION SCHEME BASED ON PAILLIER, RESIDUE NUMBER SYSTEM AND EL-GAMAL

1]Peter Awonnatemi Agbedemnab, 2]Abdul Somed Safianu and 3]Abdul-Mumin Selanwiah Salifu

School of Computing and Information Sciences,
C. K. Tedam University of Technology and Applied Sciences,
Navrongo, Ghana
pagbedemnab@cktutas.edu.gh, assafianu.stu@cktutas.edu.gh & asalifu@cktutas.edu.gh

Abstract The new focus of cryptographic research is on encryption schemes that can withstand cyber-attacks, with the arrival of cloud computing. The widely used public key encryption system designed by Taher El Gamal based on the discrete logarithm problem has been used in many sectors such as internet security, E-voting systems, and other applications for a long time. However, considering the potential data security threats in cloud computing, cryptologists are developing new and more robust cryptographic algorithms. To this end, a new robust homomorphic encryption scheme based on Paillier, Residue Number system (RNS), and El Gamal (PRE), is proposed in this paper., which is expected to be highly effective and resistant to cyber-attacks. The proposed scheme is composed a three-layer encryption and a three-layer decryption processes thereby, making it robust. It employs an existing RNS moduli set $\{2^n + 1, 2^n, 2^n - 1, 2^{n-1} - 1\}$, having passed it through the Paillier encryption process for forward conversion and then the El Gamal cryptosystem to encrpyt any data. The decryption process is a reversal of these processes starting from the El Gamal through a reverse conversion with the same moduli set using the Chinese Remainder Theorem (CRT). The simulation results shows that the proposed scheme outperforms similar existing schemes in terms of robustness and therefore, making it more secured which however, trades off with the time of execution in similar comparison.

Keywords: Cloud computing, Information Security, Chinese Remainder Theorem, Paillier, El Gamal, RNS.

1 Introduction

The world has to brace itself for the effects of cloud computing as it is finally here with its advantages and disadvantages. It is believed that cloud computing has revolutionize the computational speed and efficiency of modern-day computers which could likely lead to serious security bridges. Cloud computing is a model for enabling quick provisioning and release of customised and dependable computing resources through a shared network (Michael Kavis, 2014). Also, according to (Thabit, Can, Alhomdy, Al-Gaphari, & Jagtap, 2022), cloud computing is a technological framework where computing services encompassing storage, processing power, databases, networking, analytics, software, and intelligence are provided over the Internet. There is no contradiction that technological advancements, especially in electronic communications, have played a crucial role in shaping the modern age. As the importance of preserving data confidentiality, integrity, authenticity, and non-repudiation in its transmission and storage becomes increasingly vital, cryptography has emerged as a crucial discipline within the field of information technology (Koundinya & Gautham, 2021). Therefore, researchers continue toleverage on various encryption techniques such as Paillier, RSA, ECC, etc. with other techniques for securing data in the cloud (Hodowu, Korda, & Ansong, 2020).

Encryption is a process of ensuring that information is protected from unwanted and unauthorized access. The Residue Number System (RNS) represents numbers in residues by decomposing large integers as a vector of remainders of a given moduli such that the greatest common divisor (gcd) of any two pairwise moduli does not exceed one. The choice of the moduli set coupled with the certain inherent properties of RNS such as parallelism, fault tolerance, modularity in particular make RNS suitable for implementation in chaotic processes such as encryption. (Baagyere, Agbedemnab, Qin, Daabo, & Qin, 2020). For simplicity, given a set of relatively prime moduli $S = \{m_1, m_2, \ldots\ldots m_n\}$, such that the greatest common divisor (gcd) between any pair is one, i.e. $gcd(m_i, m_j) = 1$, for $i \neq j$. We can compute $M = \prod_{i=1}^{n}$, where S and M are the base and the dynamic range of the RNS respectively. Now any number X which is less than M has a unique representation within the dynamic range as a vector $X = x_1, x_2, \ldots\ldots x_n$, where $x_i = |X|_{m_i}$, and is referred to as the *forward conversion*. The *reverse conversion* is the process of getting back to the weighted binary or decimal number through a laborious process either using the Chinese Remainder Theorem (CRT) or the Mixed Radix

Conversion which are the traditional techniques for performing reverse conversion in RNS. The CRT is computed as (Agbedemnab, Baagyere, & Daabo, 2019);

$$\left|\sum_{i=1}^{N} m_i |x_i a_i|_{m_i}\right|_M \tag{i}$$

where,

$$M = \prod_{i=1}^{N} m_i;\ a_i = \frac{M}{m_i};\ |x_i.a_i|_{m_i} = 1$$

The Paillier cryptosystem is a partial homomorphic encryption scheme which was invented by Pascal Paillier in 1999 and allows two types of computation: addition of two ciphertexts and multiplication of a ciphertext by a plaintext number, (Suryavanshi, Alnajdi, Alhajeri, FahimAbdullah, & Christofides, 2023). The El Gamal cryptosystem on the other hand, is an asymmetric cryptosystem which uses public keys to encrypt messages/data and communicate between two parties. El Gamal is popular because of the difficulty of its discrete logarithm problem. That is, in a cyclic group that is even if we know g^a and g^k, it is extremely difficult to compute g^{ak}. This makes it a suitable choice for robust encryption schemes, (Chokparova & Urbas, 2022).

The increasing use of cloud computing services has resulted in the accumulation of enormous quantities of sensitive data in the cloud. Data security remains a notable issue since unauthorized data access could result in severe repercussions. With the emergence of cloud computing, there is a need for a more robust and efficient approach to data security in cloud computing (Lenstra & Verheul, 2001). The current state of homomorphic encryption systems, particularly the Paillier encryption system, is plagued by numerous limitations, such as unauthorized access which could lead to potential security breaches, as outlined by (Koundinya & Gautham, 2021). It has been proved by (Asiedu & Salifu, 2022) that, the Paillier Cryptosystem can be compromised through various mathematical attacks without the need to either address its security assumption related to Decisional Composite Residuosity (DCRA) or utilize its private key parameters. Meanwhile, research exploring the synergies between Paillier, RNS, and El-Gamal Cryptosystems to enhance data security in data transmission and storage, especially in the era of cloud computing, remains scarce. It appears little, or no work is done to unfold how Paillier, RNS, and El-Gamal Cryptosystems can be used to enhance data security, during data transmission and storage. This paper therefore, seeks to present an encryption system that can ensure data integrity, confidentiality, and authentication in cloud systems using Paillier and El Gamal crytosystems, and the RNS. The rest of the paper is structured as follows: Section 2 presents the struture and algorithmic flow of the proposed scheme with numerical illustrations in Section 3. The performance of the proposed scheme is evaluated based on standard metrics in Section 4 with comparisons to existing similar schemes and finally concludes in Section 5.

2 The Proposed Scheme

The proposed cryptosystem is a new robust three-layer encryption scheme which is a hybrid of the Paillier, RNS, and El Gamal (PRE) cryptosystems. The Paillier cryptosystem is used to encrypt the plaintext using the usual Paillier encryption process which is then converted into its residues through an RNS forward conversion process while the El Gamal encryption process is used to encrypt the resulting RNS residues. The cryptosystem was implemented using Python and on a 64-bit operating system, an x64-based processor, and an Intel(R) Core (TM) i5-4310M CPU @ 2.70GHz with 8.00 GB RAM. Next, is a demonstration of the steps involved in the implementation:
We begin with a psuedocode for the proposed scheme as shown in Algorithm 1.

The proposed cryptosystem is a hybrid of Paillier, RNS, and El Gamal cryptosystem and uses the RNS moduli set $\{2^n + 1, 2^n, 2^n\breve{}1, 2^{(n-1)}\breve{}1\}$,that was proposed by (Cao, Srikanthan, & Chang, 2005; Schoinianakis, 2020). The cryptosystem is divided into three parts: *key generation*, *encryption*, and *decryption* processes. The encryption process encodes and transforms plaintext into a Paillier cipher-text which is converted to its RNS equivalent residues, and it is further encrypted using the El Gamal cryptosystem. This helps thicken the encryption layer hence protecting the plaintext from various forms of attacks. On the other hand, the decryption process transforms and decodes an El Gamal cipher-text back into its plain form i.e. the residues that are converted back to the Paillier cipher-text using the CRT reverse conversion process and the final plain text is gotten by applying the Paillier decryption process.

The detailed processes are:

Algorithm 1 Psuedocode for the Proposed Cryptosystem

The proposed methods and strategies used to develop the encryption algorithm were implemented using the Paillier, RNS, and El Gamal algorithms.The moduli set used for the RNS representation is $\{2^n+1, 2^n, 2^n-1, 2^{n-1}-1\}$.
procedure EncodeData(Paillier Algorithm)
    Data was encoded first using the Paillier algorithm.
    The Paillier-encoded data was then encrypted again using the RNS algorithm, which simplified large numbers into small ones for faster processing.
    The RNS-encoded data was then encrypted again using the El Gamal algorithm, providing an added layer of security.
    The encrypted data will then be stored in the cloud or transmitted.
    When data is required, it will be decrypted using the reverse process.
    The El Gamal decryption is done first, followed by the RNS decryption (reverse conversion) and the Paillier decryption.
end procedure

I. *Key Generation*:

Generate two large prime numbers p and q such that $\gcd(pq, (p-1)(q-1)) = 1$.
Calculate $n = pq$ and $\lambda = \text{lcm}(p-1, q-1)$, where lcm denotes the least common multiple.
Choose a random integer g such that $g^\lambda \bmod n^2 = 1$.
Encode the plaintext message m as an integer value in the range $[0, n-1]$.
Choose a random integer r in the range $[1, n-1]$.
Calculate the values of μ and λ as follows:

$$\mu = \left(L(g^\lambda \bmod n^2)\right)^{-1} \tag{ii}$$

Choose n distinct even numbers.
Compute $Keys = (m_1, m_2, m_3, m_4) =$
$(2^n+1, 2^n, 2^n-1, 2^{n-1}-1)$ respectively.
Since the keys generated are co-prime, the dynamic range is given as $M = m_1 \cdot m_2 \cdot m_3 \cdot m_4$.
Hence $DR = [0, M-1]$.
Choose a cyclic group G of order q and a generator g of G.
Let x be a randomly chosen integer in the range $1 \le x \le q-1$.
The public key is (G, g, h) where $h = g^x \bmod q$, and the private key is x.

II. *Encryption Process*:

Compute the ciphertext c_1 as follows:

$$c_1 = g^m \cdot r^n \bmod n^2 \tag{iii}$$

The resulting ciphertext c_1 is further encrypted using RNS forward conversion.
$RNS = (|C_n|_{m_1}, |C_n|_{m_2}, |C_n|_{m_3}, |C_n|_{m_4})$.
Encrypt all the RNS residues $\in c_2$, using the usual Elgamal encryption process.
To encrypt the residues $r_1, r_2, r_3, r_4 \in c_2$, choose a random integer k in the range $1 \le k \le q-1$.
Compute the ciphertext as shown in eqn 2.7:

$$C_a = g^k \bmod q \tag{iv}$$

$$\begin{aligned} Enc(r_1) &= C_{r_1} = r_1 \cdot C_a^k \bmod q, \\ Enc(r_2) &= C_{r_2} = r_2 \cdot C_a^k \bmod q, \\ Enc(r_3) &= C_{r_3} = r_3 \cdot C_a^k \bmod q, \\ Enc(r_4) &= C_{r_4} = r_4 \cdot C_a^k \bmod q. \end{aligned}$$

The resulting ciphertext is
$([C_a, C_{r_1}], C_a, C_{r_2}], [C_a, C_{r_3}], [C_a, C_{r_4}])$, which is then transmitted.

III. *Decryption Process*:

To decrypt the ciphertext

$([C_a, C_{r_1}], [C_a, C_{r_2}], [C_a, C_{r_3}], [C_a, C_{r_4}])$, compute the plaintext as:

$$p_{r_1} = C_{r_1} \cdot (C_a^x)^{-1} \bmod q = r_1,$$
$$p_{r_2} = C_{r_2} \cdot (C_a^x)^{-1} \bmod q = r_2,$$
$$p_{r_3} = C_{r_3} \cdot (C_a^x)^{-1} \bmod q = r_3,$$
$$p_{r_4} = C_{r_4} \cdot (C_a^x)^{-1} \bmod q = r_4,$$

2: The resulting plaintext is $p_{r_n} = (r_1, r_2, r_3, r_4)$.
3: Use the reverse Chinese Remainder Theorem (CRT) to combine the residues $\in c_2$ into a single decimal X.
4: That is, compute X as:

$$(r_1 M_1 \cdot M_1^{-1} + r_2 M_2 \cdot M_2^{-1} + r_3 M_3 \cdot M_3^{-1} + r_4 M_4 \cdot M_4^{-1}) \mod M \quad (v)$$

5: The resulting plaintext message is $c_1 = X \mod M$. c_1 is further decrypted using the usual Paillier decryption process, which is given as:
6:

$$m = L(c^\lambda \mod n^2) \cdot \mu \mod n \quad (vi)$$

2.1 Structure of the Proposed Cryptosystem

Figure 1 points is a block diagram that shows the main components of the structure of the proposed cryptosystem. As has been indicated earlier, the whole structure is divided into three modules: Key generation, Encryption process,and Decryption process. The following blocks of psuedocodes presented as algorithms 2, 3 and 4 further describe each of the modules in detail.

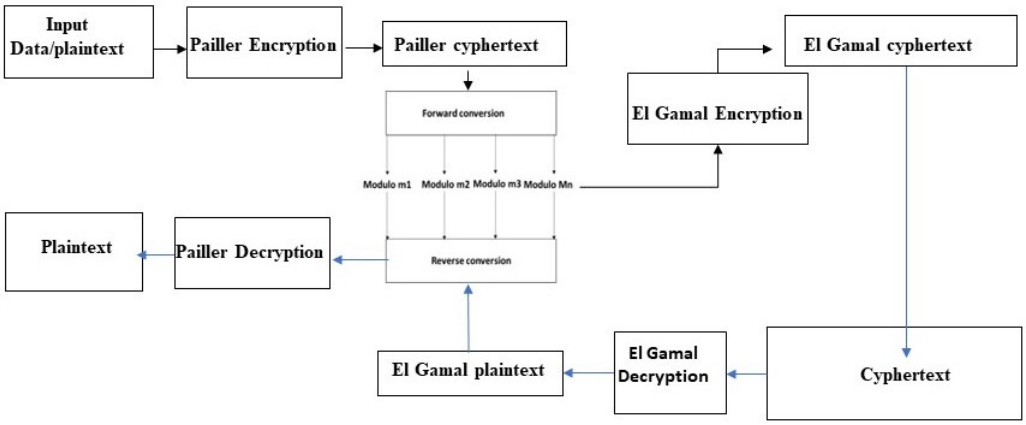

Figure 1: Proposed Cryptosystem

Algorithm 2 Key Generation

1: PK: Choose large prime numbers p and q such that the prime numbers are not equal
2: Compute $n = p \cdot q$
3: Compute $\lambda = \text{lcm}(p-1, q-1)$
4: Choose an integer g such that $g^\lambda \mod n^2 = 1$
5: Encode the plaintext message m as an integer value in the range $[0, n\breve{}1]$
6: Choose a random integer r in the range $[1, n\breve{}1]$
7: RK: Choose n distinct even numbers. Compute Keys $= \{m_1, m_2, m_3, m_4\} = \{2^n + 1, 2^n, 2^n\breve{}1, 2^{(n-1)}\breve{}1\}$: the DR $>$ the Paillier encrypted value
8: EK: Choose a cyclic group G of order q and a generator g of G
9: Generate Epubkey (G, g, h): $h = g^x \mod q$, and *Eprivtkey* is x, a randomly chosen integer in the range $1 \leq x \leq q-1$

Algorithm 3 Pallier-RNS-El Gamal Encryption

Require: Plaintext (P_{txt})
ASCII values $\leftarrow [P_{\text{txt}}]$
Compute C1 = PallierEnc(X_n) ▷ Encryption Process
PCiphertext: $mi \leftarrow$ C1
Get ($C1$): mi and choose $n \in M$; M is the dynamic range
Compute C2 = ($|C1|_{m_1}$, $|C1|_{m_2}$, $|C1|_{m_3}$, $|C1|_{m_4}$)
RCiphertext: $(r_1, r_2, r_3, r_4) \leftarrow$ C2
Gen Epubkey []: Random Prime Number generator
Get (RCiphertext, Epubkey) **for** $i = 1$ *to* n *in* $C2$ **do**
end
Gen random K
Compute $C_a = g^k \mod q$
Encrypt the residues in C2 as $C_{r_i} = r_i \cdot C_a \mod q$

ECiphertext: Unicode ($[(C_a, C_{r_1}), (C_a, C_{r_2}) \ldots (C_a, C_{r_n})]$)

Algorithm 4 Pallier-RNS-El Gamal Decryption

Require: Decode ECiphertext, Eprivtkey **for** $i = 1$ *to* n **do**
end
$Pr_n = C_{r_1} \cdot C_{x_a^{-1}} \mod q = r_n$

C2 RNS residues state: $C2 = (r_1, r_2, r_3, r_4)$
▷ RNS reverse Process
Get ($C2$, N)
Input: Compute CRT_inverse = $|M \cdot M^{-1}|_{mn} = 1$
Output: $C1 = (r_1 \cdot M1 \cdot M1^{-1} + r_2 \cdot M_2 \cdot {M_2}^{-1} + r_3 \cdot M_3 \cdot {M_3}^{-1} + r_4 \cdot M_4 \cdot M_4^{-1}) \% M$

Pallier Decryption process
Get ($C1$, P_{privtkey})
$X_n = \text{Dec}(C1)$
Input: ASCII values (X_n)
ASCII values $\leftarrow [(P_{\text{txt}})]$
Output: plaintext (P_{txt}) =0

3 Numerical Illustrations

Next, we present some numerical examples in order to demonstrate and validate the performance and efficiency and/or usability of the proposed scheme. For example, consider the moduli set $\{17, 16, 15, 7\}$ and $n = pq = 143$, then

I. Key Generation

a. Paillier Keys:Generate two large prime numbers p and q such that $\gcd(13 * 11, (13-1)(11-1)) = 1$. Calculate $n = pq = 143$,$n^2 = 20449$ and $\lambda = \text{lcm}(13-1, 11-1)$, Choose a random integer g such that $g^\lambda \bmod n^2 = 1$. Let $g = 37$, Encode the plaintext message m as an integer value in the range $[0, n-1]$. Let $m = 43$, Choose a random integer r in the range $[1, n-1]$; also, let $r = 35$.

b. RNS Keys: Choose $n = 4$ distinct even numbers. Compute $Keys = (m_1, m_2, m_3, m_4) = (2^4 + 1, 2^4, 2^4 - 1, 2^{4-1} - 1)$ respectively; which implies that $Keys = \{m_1, m_2, m_3, m_4\} = \{17, 16, 15, 7\}$. Since the keys generated are co-prime, the dynamic range is give as: $M = 17 \times 16 \times 15 \times 7$, therefore, the legitimate representation range is $[0, 28560 - 1]$

c. Elgamal Keys: Choose a cyclic group G of order $q = 13$ and a generator 2 of G. Let $x = 3$ be a randomly chosen integer in the $1 \leq 3 \leq 19 - 1$ range. The public key is $(19, 2, 8)$ where $h = 2^3 \bmod 19 = 8$, and the private key is 3.

II. Encryption process

a. Paillier Encryption process: Compute the ciphertext c_1 as follows:

$$c_1 = 37^{43} \cdot 35^{143} \bmod 143^2 c_1 = 10445 \quad (vii)$$

b. RNS forward conversion (Encryption) process: The resulting ciphertext $c_1 = 10445$ is further encrypted using the proposed RNS conversion. To encrypt a plaintext message c_1, first represent it as a pair of residues $RNS(|10445|_{17}, |10445|_{16}, |10445|_{15}, |10445|_7)$. Therefore the resulting second ciphertext is the residues computed $c_2 = (r_1, r_2, r_3, r_4) = (7, 13, 5, 1)$ which is further encrypted using the usual Elgamal encryption process.

c. Elgamal Encryption process: Encrypt all the RNS residues $\in c_2$,use the usual Elgamal encryption process.To encrypt the message $r_1, r_2, r_3, r_4 \in c_2$, choose a random integer 7 in the range $1 \leq 7 \leq 19 - 1$. Compute the ciphertext as:(ciphertextA) $C_a = 2^7 \bmod 19 = 14$,

$$\begin{aligned} Enc(r_1) &= C_{r_1} = 7 * 8^7 \bmod 19 = 18, \\ Enc(r_2) &= C_{r_2} = 13 * 8^7 \bmod 19 = 9, \\ Enc(r_3) &= C_{r_3} = 5 * 8^7 \bmod 19 = 2, \\ Enc(r_4) &= C_{r_4} = 1 * 8^7 \bmod 19 = 8. \end{aligned}$$

The resulting ciphertext is $[(14, 18), (14, 18)(14, 9)(14, 2)(14, 8)]$. which is then transmitted.

III. Decryption process

c. El gamal Decryption process:To decrypt the ciphertext $(C_a, C_{r_1}, C_{r_2}, C_{r_3}, C_{r_4}) = (14, 18, 9, 2, 8)$, compute:(plaintext) $p_{r_1} = 18 * (14^3)^{-1} \bmod 19$ where $(14^x)^{-1}$ denotes the modular multiplicative inverse of 14^3 modulo $19 = 12$.Hence,$p_{r_1} = 18 * 12 \bmod 19 = 7$,$p_{r_2} = 9 * 12 \bmod 19 = 13$,$p_{r_3} = 2 * 12 \bmod 19 = 5$,$p_{r_4} = 8 * 12 \bmod 19 = 1$,the resulting plaintext is $p_{r_n} = (7, 13, 5, 1)$.

b. RNS reverse conversion(Decryption) process:To convert the residues $\in c_2$ to the paillier ciphertext,Use the reverse Chinese Remainder Theorem (CRT) to combine the residues $\in c_2$ into a single decimal X:

$$X = (7 \cdot 1680 \cdot 11 + 13 \cdot 1785 \cdot 9 + 5 \cdot 1904 \cdot 14 + 1 \cdot 4080 \cdot 6) \mod 28560 \quad (viii)$$

where M_1^{-1},M_2^{-1},M_3^{-1} and M_4^{-1} are the modular multiplicative inverses of M_1 modulo m_1, M_2 modulo m_2, M_3 modulo m_3 and M_4 modulo m_4 respectively. The resulting plaintext message is $c_1 = X \mod M$. c_1 is further decrypted using the usual paillier decryption process.

a. Paillier Decryption process: Calculate the values of μ and λ as follows:

$$\mu = (L(37^{60} \bmod 143^2))^{-1} = 125 \quad (ix)$$

where, $L(\mu) = \frac{\mu - 1}{n}$

$$\lambda = \text{lcm}(13 - 1, 11 - 1) = 60 \quad (x)$$

,where $\varphi(143)$ is Euler's totient function. Calculate the plaintext message m as follows:

$$m = L(10445^{60} \bmod 143^2) \cdot 125 \bmod 143 \quad (xi)$$

where, $L(12156) = \frac{12156 - 1}{143} = 85$.

The decrypted message is $m = 43$ and the resulting plaintext message is '+'.

4 Performance Evaluation

The simulated results of the proposed scheme was evaluated using standard metrics such as the cipher uniqueness test, the runtime, key sensitivity, and throughput rate. In order to evaluate its contribution, it was also compared with similar best-known state-of-the-art cryptosystems. The analysis from tables 3, 4, and 6 shows that the proposed

cryptosystem has higher encryption and decryption runtimes, and lower throughput compared to that in (Asiedu & Salifu, 2022) and (Koundinya & Gautham, 2021). Though the proposed cryptosystem has a higher runtime than that in (Asiedu & Salifu, 2022) and (Koundinya & Gautham, 2021), the proposed cryptosystem has an additional layer of encryption which thickens the encryption cryptosystem. Thus the proposed cryptosystem uses three encryption layers whilst the cryptosystems by (Asiedu & Salifu, 2022) and (Koundinya & Gautham, 2021) use two encryption layers.

4.1 Cipher Uniqueness Test

Table 1: Cipher Uniqueness Text

ASCII	**Pallier cipher**	**Final Cipher text**
43	10445	[[(79531,412876),(79531,517657),(79531,377949),(79531,308095)]]
43	10445	[[(370100,26750),(370100,124203),(370100,168156),(370100,450968)]]
65	6014	[[(443818,610426),(443818,793134),(443818,793134),(443818,182708)]]
65	6014	[[(104485,72858),(104485,61186),(104485, 61186), (104485, 100625)]]

In Table 1, the unique cipher text is generated depending on the random value generated which, results in different cipher text, but the plaintext returned is always the same as the text that was encrypted. The encrypted messages are +, +, A, and A. It was observed that as the Paillier cipher text increases, only schemes with the RNS turn to give the right text since the RNS encrypted is always less than the selected prime number in the El Gamal encryption process.

4.2 Runtime Measurement

We analysed the time taken for our PRE scheme to produce an output. In the analysis, we used $n = 4, k = 7, P = 11, q = 13$ and simulated several times in order to record the average encrytion and decryption runtime. The results of this measure is presented in Table 2 as well as Fig. 2. The results reveal that, encryption and decryption times increase as the number of elements and/or the number of iterations increase. Thus, 'Hello' text with 5 letters and a single iteration requires 1.070000 microseconds for encryption and 1.300000 microseconds for decryption. This would change if the size of the text or the number of iterations changed. For instance, if the number of text increases to 96, the total encryption and decryption times for the Text are 2.000000 microseconds and 3.870000 microseconds respectively. This gives an average time of 100 times the single text. For example, if the number of iterations increases to 100, the total encryption and decryption times for the 'Hello' Text are 14.832 microseconds and 13.251700 microseconds respectively. This gives an average time of 100 times the single iteration. It is evident from Table 2 and Fig. 2, that the encryption runtime keeps increasing as the input length increases while it takes less time to decrypt the encrypted text back to the original text.

Table 2: Encryption and Decryption Runtime of the PRE system

Text Length	Text	**Encryption Time (ms)**	**Decryption Time (ms)**
11	Hello world	0.000518	0.000300
16	Zeinab is my mom	0.000724	0.000450
25	Jumping foxes vex bad dogs	0.001127	0.000689
29	God bless our homeland Ghana.	0.00132	0.0007188

The encryption runtime was juxtaposed with two scenarios where running Paillier with RNS (PR) only and Paillier with El Gamal (PE) only presented by (Asiedu & Salifu, 2022) and (Koundinya & Gautham, 2021) respectively. This is shown in Table 3.

From Table 3 and Fig. 3, it is clear that the three-layer encryption scheme is a bit slower for encryption because of the extra layer of security it possesses.The PE cryptosystem followed closely behind. It turned out that the PR scheme performed faster than the other. This can be as a result of the utilisation of residues i.e. smaller numbers. Similarly as

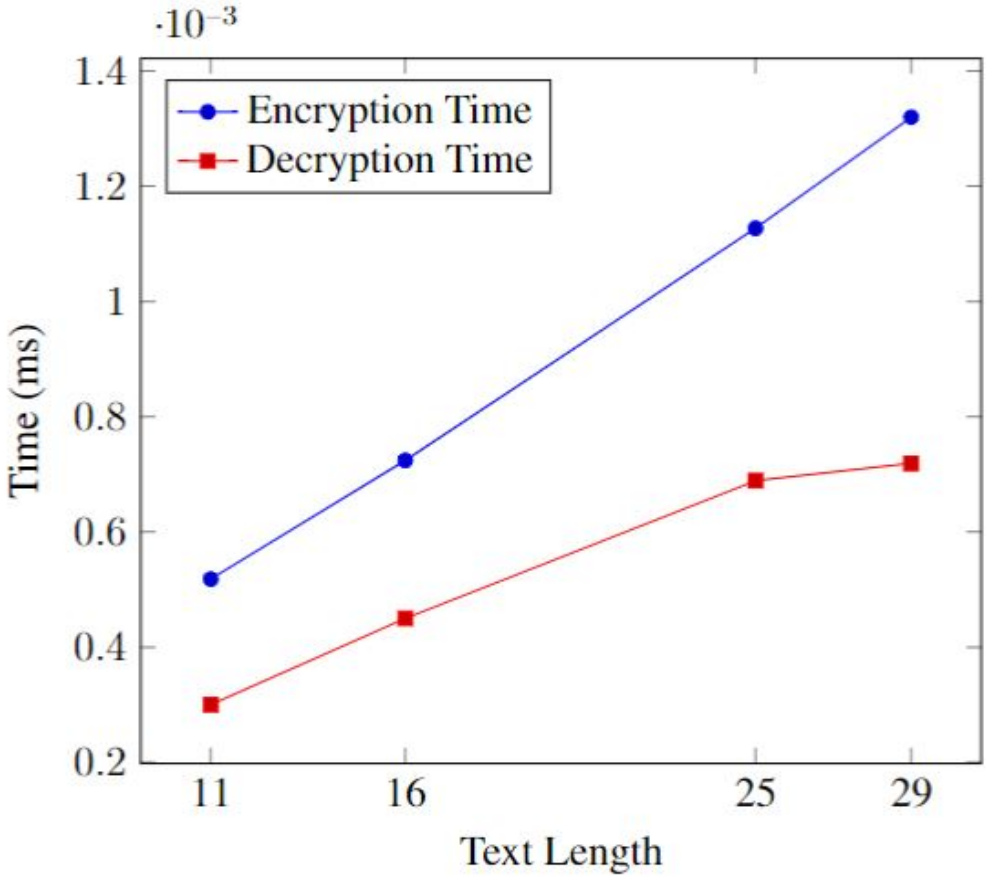

Figure 2: Runtimes

Table 3: Encryption Runtimes

Text-length	**Text**	**Cryptosystem Encryption time**		
		PRE	**PR**	**PE**
11	Hello world	0.000518	0.000078	0.000148
16	Zeinab is my mom	0.000724	0.000109	0.000207
25	Jumping foxes vex bad dogs	0.001127	0.000169	0.000335
29	God bless our homeland Ghana.	0.00132	0.0003417	0.0006518
34	Tomorrow is certainly not certain.	0.001517	0.000220	0.000722

shown in Table 4 and Fig. 4, the proposed PRE is slower to decrypt as a result of the extra layer. Notwithstanding, this extra layer makes the proposed PRE very robust.

4.3 Key sensitivity process

The sensitivity of the encryption system's key parameters was investigated by varying the "g" value in the Paillier public key and the "y" value in the ElGamal public key while keeping other parameters constant. The study used specific values for parameters such as $p, q, n, P, G, y, \lambda$, μ, N and, x. The results of these variations were observed and recorded in Table 5.

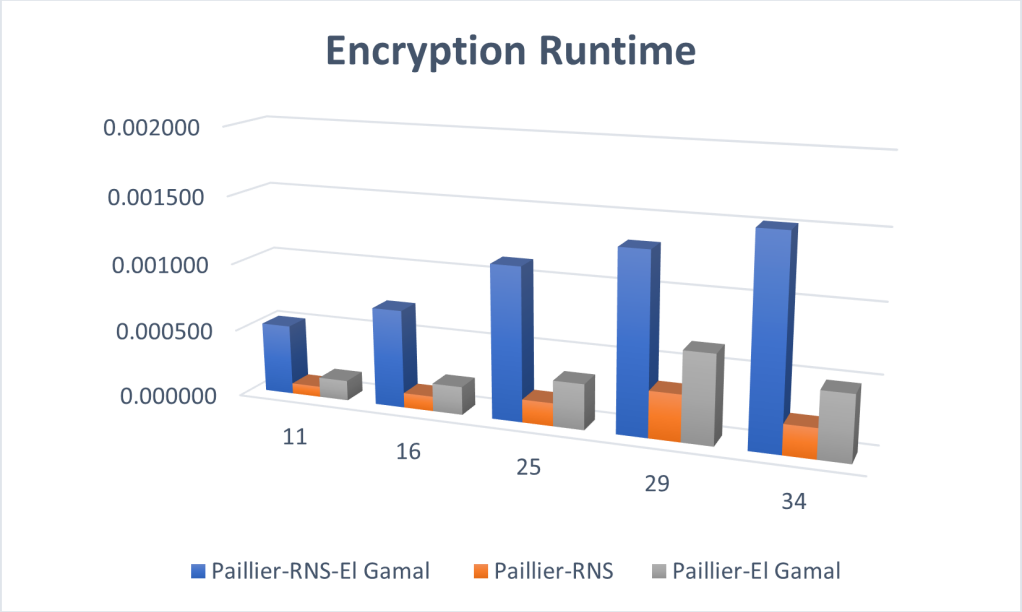

Figure 3: Encryption Runtimes

Table 4: Decryption Runtimes

Text-length	**Text**	**Cryptosystem Decryption time**		
		PRE	**PR**	**PE**
11	Hello world	0.000300	0.000144	0.000086
16	Zeinab is my mom	0.000450	0.000265	0.000114
25	Jumping foxes vex bad dogs	0.000689	0.000311	0.000222
29	God bless our homeland Ghana.	0.0007188	0.0003496	0.0001893
34	Tomorrow is certainly not certain.	0.000882	0.000454	0.000187

Table 5: Key Sensitivity Analysis

S no.	**Plaintext**	**Cryptosystem keys**		**Decrypted text**	**Results**
		$g = n + 1$	$y = pow(G, X, P)$		
1	43	144	393581	43	True
2	63	144	393581	63	True
3	65	144	393581	65	True
4	43	198	393580	12	False
5	63	198	393580	133	False
6	65	897	393580	66	False
7	43	144	7984580	68	False
8	63	144	7984580	39	False
9	65	144	7984580	119	False

4.4 Throughput Evaluation

Throughput, also known as data transfer rate, is calculated by dividing the total size of the encrypted plaintext by the time taken to complete the encryption process. A greater throughput value indicates improved performance.

$$\text{Throughput} = \frac{\text{Data Size}}{\text{Processing Time}}$$

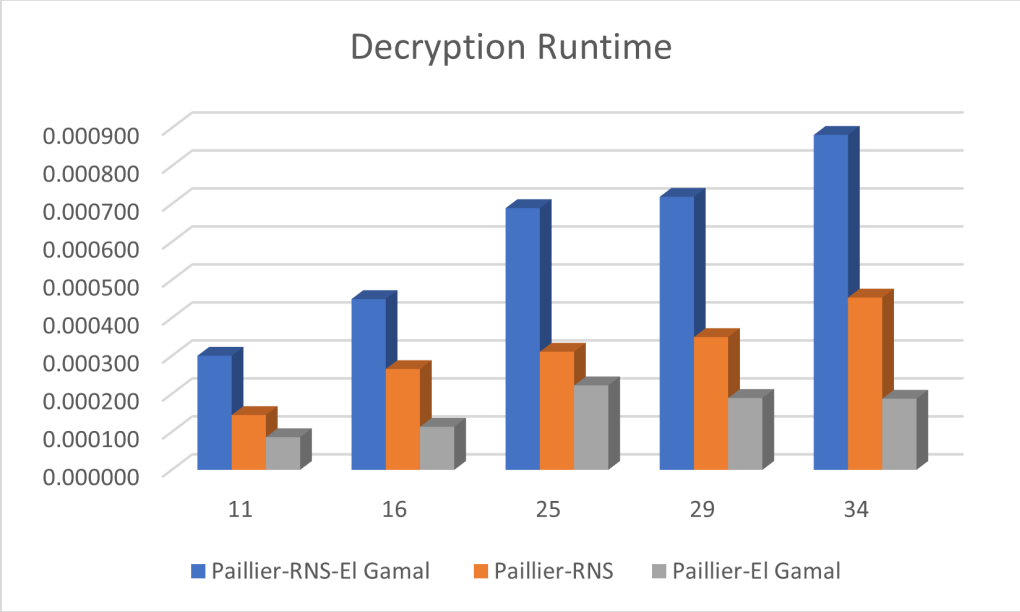

Figure 4: Decryption Runtimes

Where the data size is typically measured in bytes.

Table 6: Comparison of the throughput of the various Cryptosystems

Cryptosystem	**Text size (byte)**	**Encryption Time(s)**	**Throughput(bps)**
PRE	60, 69, 82,99	0.000551, 0.001221, 0.001453,0.002491	108893, 56511 , 56435, 39743
PR	60, 69, 82,99	0.000174, 0.000166, 0.000311,0.000685	345224 , 414663 , 263327 , 144610
PE	60, 69, 82,99	0.000223, 0.000413, 0.000629,0.001070	269058,166989, 130324 , 92541

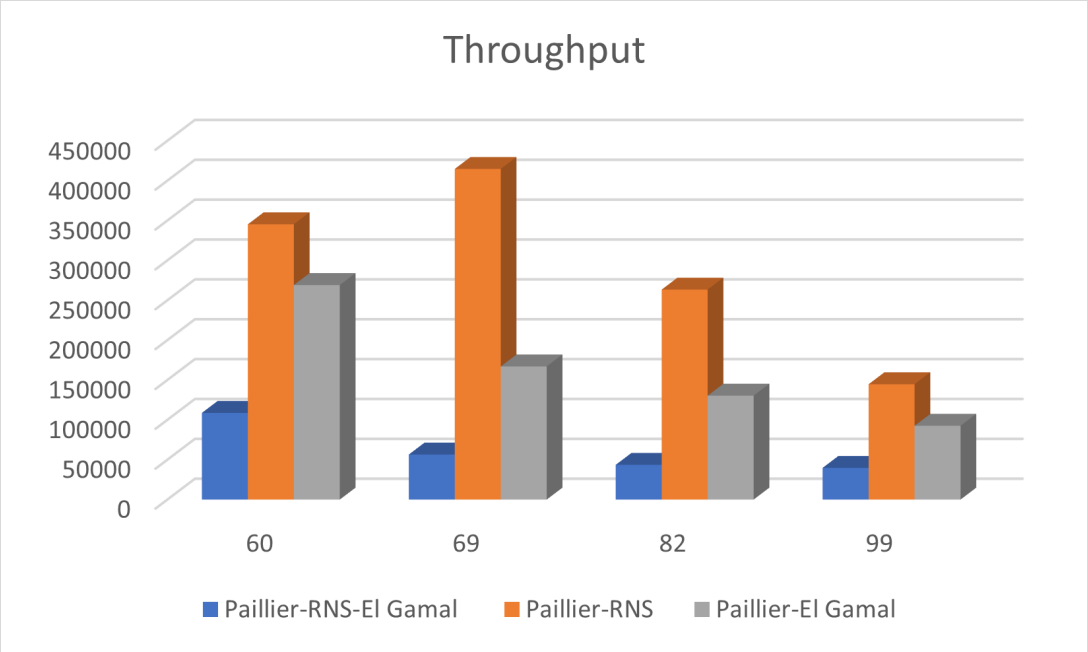

Figure 5: Comparison of the Throughput

5 Conclusions

This paper presented a new robust hybrid encryption scheme based on the Paillier, RNS, and El Gamal cryptosystems, a PRE cryptosystem. It has been demonstrated that the approach enhanced security against brute force and known plaintext attacks, overcoming some of the limitations in the conventional methods as a result of the additional layer. The proposed scheme should therefore, be a best choice in instances where robustness is preferred. The encryption and decryption runtime can be improved in future works.

Conflict of Interest
The authors declare no conflict of interest situation.

References

Agbedemnab, P. A., Baagyere, E. Y., & Daabo, M. I. (2019). A novel text encryption and decryption scheme using the genetic algorithm and residual numbers. In *Proceedings of 4th international conference on the* (Vol. 12, pp. 20–31). doi: 10.1109/AFRICON46755.2019.9133919

Asiedu, D., & Salifu, A.-M. (2022). Secured Paillier Homomorphic Encryption Scheme Based on the Residue Number System. *Int. J. Cryptogr. Inf. Secur.*, *12*(1), 1–13. doi: 10.5121/ijcis.2022.12101

Baagyere, E. Y., Agbedemnab, P. A., Qin, Z., Daabo, M. I., & Qin, Z. (2020). A multi-layered data encryption and decryption scheme based on genetic algorithm and residual numbers. *IEEE Access*, *8*, 100438–100447. doi: https://doi.org/10.1109/ACCESS.2020.2997838

Cao, B., Srikanthan, T., & Chang, C. (2005). Efficient reverse converters for four-moduli sets { 2n1, 2n, 2n+1, 2n+11} and {2n1, 2n, 2n+1, 2n11}. *IEE Proc. - Comput. Digit. Tech.*, *152*(5), 687. Retrieved from `https://digital-library.theiet.org/content/journals/10.1049/ip-cdt{_}20045155` doi: 10.1049/ip-cdt:20045155

Chokparova, Z., & Urbas, L. (2022). Application of multiplicative homomorphic encryption in process industries. In L. Montastruc & S. Negny (Eds.), *32nd european symposium on computer aided process engineering* (Vol. 51, p. 1267-1272). Elsevier. Retrieved from `https://www.sciencedirect.com/science/article/pii/B9780323958790502125` doi: https://doi.org/10.1016/B978-0-323-95879-0.50212-5

Hodowu, D. K. M., Korda, D. R., & Ansong, E. D. (2020). An enhancement of data security in cloud computing with an implementation of a two-level cryptographic technique, using aes and ecc algorithm. *Int. J. Eng. Res. Technol*, *9*, 639–650.

Koundinya, A. K., & Gautham, S. K. (2021). Two-Layer Encryption based on Paillier and ElGamal Cryptosystem for Privacy Violation. *Int. J. Wirel. Microw. Technol.*, *11*(3), 9–15. doi: 10.5815/ijwmt.2021.03.02

Lenstra, A. K., & Verheul, E. R. (2001). Selecting cryptographic key sizes. *Journal of cryptology*, *14*, 255–293. doi: https://doi.org/10.1007/s00145-001-0009-4

Michael Kavis, M. K. (2014). *Architecting the cloud design decisions for cloud computing service models.* Wiley Online Library. doi: https://doi.org/10.1002/9781118691779

Schoinianakis, D. (2020, sep). Residue arithmetic systems in cryptography: a survey on modern security applications. *J. Cryptogr. Eng.*, *10*(3), 249–267. Retrieved from `https://link.springer.com/10.1007/s13389-020-00231-w` doi: 10.1007/s13389-020-00231-w

Suryavanshi, A. V., Alnajdi, A., Alhajeri, M. S., FahimAbdullah, & Christofides, P. D. (2023). An encrypted mpc framework for security to cyber-attacks. In A. C. Kokossis, M. C. Georgiadis, & E. Pistikopoulos (Eds.), *33rd european symposium on computer aided process engineering* (Vol. 52, p. 1513-1518). Elsevier. Retrieved from `https://www.sciencedirect.com/science/article/pii/B9780443152740502419` doi: https://doi.org/10.1016/B978-0-443-15274-0.50241-9

Thabit, F., Can, O., Alhomdy, S., Al-Gaphari, G. H., & Jagtap, S. (2022). A novel effective lightweight homomorphic cryptographic algorithm for data security in cloud computing. *International Journal of intelligent networks*, *3*, 16–30. doi: https://doi.org/10.1016/j.ijin.2022.04.001

Author Biography

Peter Awonnatemi Agbedemnab received his B.Sc. degree in Computer Science, M.Sc. with research and, Ph.D. degrees in Computational Mathematics all from the University for Development Studies, Tamale, Ghana in 2012, 2015 and 2020 respectively. He is currently a Senior Lecturer and the Head for the Department of Information Systems and Technology in the School of Computing and Information Sciences at the C. K. Tedam University of Technology and Applied Sciences, Navrongo, Ghana. Dr. Agbedemnab current research interests include but not limited to theoretical computing, genetic algorithms, cryptography, and information security and technologies.

Abdul Somed Safianu holds a Bachelor of Science degree in Computer Science from the University for Development Studies, Tamale, Ghana which he obtained in 2020. He recently successfully graduated with an MPhil. Computer Science degree from the C. K. Tedam University of Technology and Applied Sciences, Navrongo, Ghana. He is currently working as an Associate IT Auditor at KPMG Ghana. His research interests include Cryptography, cyber security, AI, machine learning and deep learning.

Abdul-Mumin Selanwiah Salifu is an Associate Professor with the department of Information Systems and Technology at the C. K. Tedam University of Technology and Applied Sciences. Prof. Abdul-Mumin Selanwiah Salifu has conducted several research works in various areas including Wireless Network Security, Network Intrusion Systems, Cryptography, Data Communication and Computer Networks, Network Protocols, Residue Number Systems and its Applications, Database Applications, Satellite Communications and Wireless Sensor Networks.