LRSA-Hybrid Encryption Method Using Linear Cipher and RSA Algorithm to Conceal the Text Messages

Rundan Zheng[1], Chai Wen Chuah*[2], Janaka Alawatugoda*[3]
Guangdong University of Science & Technology, Dongguang, Guangzhou, China[1,2]
Research & Innovation Centers Division, Rabdan Academy Abu Dhabi, UAE[3]
Institute for Integrated and Intelligent Systems, Griffith University, Nathan, Queensland, Australia[3]

***Abstract*—Computer science and telecommunications technologies have been experiencing rapid advancements in recent years to protect sensitive data or information from potential harm, misuse, or destruction. By enhancing data security through various methodologies and algorithms, data can be better protected against attacks that may compromise its confidentiality, particularly in the case of text messages. Linear cipher is one of the earliest forms of cryptographic systems which operates by shifting letters that may not provide the highest level of security but adds a layer of complexity to the initial encryption process. Rivest-Shamir-Adleman algorithm represents a more advanced and rigorous approach to encryption that resistant to more sophisticated attacks. The Rivest-Shamir-Adleman algorithm utilizes the mathematical properties of large prime numbers to establish a secure communication channel. The combination of both algorithms or hybrid algorithms employed for data security, the security of text messages is significantly improved, ensuring the confidentiality of the text messages during its transmission. Hence, this research proposes two types of hybrid algorithms, namely Gradatim LRSA and Optimized LRSA, which ensure the confidentiality of the text message using encryption and decryption processes. The results also show that the Optimized LRSA performs with less computation compared to the Gradatim LRSA.**

***Keywords*—Confidentiality; data encryption; hybrid encryption; linear cipher; RSA algorithm; Gradatim LRSA; Optimized LRSA**

I. Introduction

Data security is the practice in protecting individual personal information from being unauthorized access, and misused by unauthorized parties [1], [2], [3]. The personal information can be medical information, financial records, passwords, personal identification numbers and so on. If these data breaches, it can severely damage an organization's reputation and erode individual trust as well as their stakeholders' trust [4], [5], [6]. As the result, it may cause the financial losses and or result in legal penalties [7]. Therefore, the organizations need to demonstrate their commitment in protecting their customer data [8]. The act in protecting these data can be encryption to ensure that the data remains confidential, available and reliable [9], [10].

Linear cipher is a historical cipher. It is a mathematical linear function with one-dimensional symmetrical encryption which suppose can provide data confidentiality [11]. The plaintext is in a linear relationship with the corresponding ciphertext making easily identifiable patterns. Hence, making them susceptible to decryption through frequency cryptanalysis.

To date, the most widely used public key cryptosystem is the Rivest-Shamir-Adleman (RSA) algorithm, which was proposed by Ron Rivest, Adi Shamir, and Leonard Adleman in 1976 [12]. The strength of the RSA algorithm is based on the challenge of factoring large prime numbers to obtain the private key. It is based on a simple number theoretic fact: it is easy to multiply two large prime numbers, but it is extremely difficult to factor their products [13]. As long as no efficient method for factoring these large primes is discovered, the security of the RSA algorithm remains intact.

The linear cipher is vulnerable to statistical attacks. Therefore, pairing the linear cipher with the RSA algorithm can mitigate these vulnerabilities by introducing asymmetric encryption, which offers greater resistance to such attacks. Hence, this research proposes two types of hybrid encryption method using a linear cipher and RSA algorithm (LRSA) to conceal the text messages. We denoted it as Gradatim LRSA and Optimized LRSA. The significance of the proposed hybrid encryption lies in its ability to mitigate the weaknesses of relying solely on either encryption method. Even if the linear cipher is compromised, the RSA algorithm provides an additional layer of security in terms of confidentiality. The process begins with linear cipher encoding the plaintext, the output will further encrypt by RSA algorithm to produce the ciphertext. The decryption process will decrypt the ciphertext using RSA, then followed by the Linear cipher to reveal the final plaintext. The difference is that the number of rounds in the calculation for generating the ciphertext in the Optimized LRSA is fewer than in the Gradatim LRSA.

The remainder of this paper is organized as follows: Section II presents the literature review. The research design is shown in Section III. Section IV introduces the proposed Gradatim LRSA and Optimized LRSA. Section V demonstrates the characters in numeric form. The simulation results are shown in Section VI. Sections VII and VIII present the frequency analysis and discussion, respectively. Finally, Section IX concludes the paper.

II. Literature Review

This section involves the analysis and synthesis of existing research and publications on cryptography, confidentiality, linear cipher, and RSA algorithms. This review helps to identify gaps in the literature and establish the significance of this research.

A. Cryptography

Cryptography serves to ensure the privacy, integrity, and accessibility of data for authorized users. Besides that, cryptography is used to maintain the privacy and confidentiality of data during transmission or storage. The common daily activities which need the cryptography protection including online banking, e-commerce transactions, and data security [14].

Cryptography employs mathematical techniques or algorithms for the data protection [15]. For example, encryption and decryption techniques are utilized to uphold data confidentiality. Encryption is a mathematical process that converts data into unreadable ciphertext, while decryption is a mathematical process to convert the ciphertext into original data.

B. Confidentiality

Maintaining the confidentiality of information holds a critical and indispensable role in today's digital era. It is a practice of keeping information private and only sharing it with authorized individuals [16]. Confidentiality is a fundamental aspect of maintaining trust and privacy especially in the sectors require professional relationship, such as finance, banking, law, healthcare, and medicine.

Algorithms such as encryption and decryption are employed to achieve confidentiality of information [17]. In ancient times, historical ciphers like the linear cipher, Caesar cipher, and hill cipher were commonly utilized to ensure message confidentiality. These ciphers typically involve simple substitution or transposition techniques. These techniques are breakable and require low computational power [11]. In contrast, modern ciphers such as the Advanced Encryption Standard, Trivium, and RSA algorithm offer a higher level of security [18]. These ciphers use complex mathematical algorithms in safeguarding confidential information.

C. Linear Cipher

Linear cipher is a historical cipher that using a linear mathematical operation to encode the readable message into ciphertext and decode ciphertext back to original readable message. The linear cipher encoding process as shown in Eq. (1) [11]. The secret keys are represented as a and b. "M" represents the set space of message (m). The output is the ciphertext c which is within the set space of M. The message is multiplied by the key a, and the output is then added to the key b. The final output is then taken modulus of the set space M to generate the ciphertext. Noted that the key a must coprime with M.

$$c = a * m + b \ mod \ M, \ where \ c, m \in M \tag{1}$$

The linear cipher decoding process is shown in Eq. (2) [11]. The ciphertext is multiplied by the inverse key a, and the output is then subtracted by the key b. Therefore, the key a must be relatively prime with M. The final output is then taken modulo of the set space M to retrieve the message.

$$m = a^{-1}(c - b) \ mod \ M, \ where \ c, m \in M \tag{2}$$

For example, suppose a message 'B' has a numeric value is 1. The linear cipher secret keys are $a = 5$ and $b = 10$. The message space M is equal to 27. Once the mesasge is encoded using Eq. (1), the ciphertext value is calculated $c = (5 * 1 + 10)$ mod 26 = 15. For the decoded process using Eq. (2), the message value is calculated $m = 5^{-1}(15 - 10)$ mod $27 = 11(15 - 10)$ mod 27 = 1.

1) Cryptanalysis Linear Cipher: The linear cipher relies on a linear mathematical operation, such as multiplication and addition to encode and decode the messages. Cryptanalysis of a linear cipher involves breaking the encryption of a message that has been encoded using a linear transformation.

In order to cryptanalyze the linear cipher, one can discover the weaknesses of the linear cipher by examining potential vulnerabilities within the key generation process or how the ciphertext is produced [19]. Once the values of a and b are compromised, we are able to decrypt the entire message.

Next, one may analyze ciphertext patterns, such that frequency analysis or pattern recognition. Frequency analysis involves analyzing the frequency of characters in the ciphertext and comparing it to the frequency of characters in the language of the message [11]. This may help determine the possible mapping of characters in the ciphertext to characters in the message. Then, we can form linear equations and determine the key used for encryption. Once the secret keys are known, the ciphertext can be decoded.

D. Rivest-Shamir-Adleman Algorithm

RSA algorithm is an asymmetric encryption scheme in the field of cryptography [12]. It was invented in 1977 by Rivest, Shamir, and Adleman. The algorithm uses a pair of keys which known as public key and private key. The public key is used for encryption and the private key is used for decryption. The security of the encryption relies on the difficulty of factoring large numbers. That numbers, we denoted it as p and q.

To generate the keys, one must choose two large prime numbers, p and q. Next, calculate their product, which is used as the modulus for encryption. We denote it as n. The public key is derived from the modulus n and an exponent e. The exponent, e, is chosen so that it is relatively prime to $\varphi(n)$. The private key is calculated using the Extended Euclidean algorithm to find the private exponent, d, which is the modular multiplicative inverse of e modulo $\varphi(n)$.

The RSA encryption process is shown in Eq. (3) [12]. The message is raised to the power of the public exponent, e, and the result is taken modulo n.

$$c = m^e \ mod \ n, \tag{3}$$

The RSA decryption process is shown in Eq. (4) [12]. The ciphertext is raised to the power of the private exponent, d, and the result is taken modulo n.

$$m = c^d \ mod \ n, \tag{4}$$

For example, assume a message's numeric value is 2. The RSA two prime numbers are $p = 11$ and $q = 13$. The value of $n, \varphi(n)$ are calculated as follows: $n = 11 * 13 = 143$, $\varphi(n) = 120$. Let's choose public key e = 7. To perform message encryption using Eq. (3), the ciphertext value is calculated as $c = m^e mod n = 2^7$ mod 143 = 128. For decryption of the ciphertext $c = 128$, the private key d is needed, where $d = e^{-1}$ mod $\varphi(n) = 7^{-1}$ mod 120 = 103. Using Eq. (4), the message value is calculated $m = c^d$ mod $n = 128^{103}$ mod 143 = 2.

E. Number Systems

Number systems are the system represent the numbers. For instance, decimal number system, binary number system and hexadecimal system [20]. The decimal number system or base-10 system is the counting method that commonly used daily counting scheme. It relies on ten symbols (digits), namely the digits or numbers from 0 to 9. For instance, number 10 (2 digits), 100 (3 digits, 1000 (4 digits) and so on.

Binary number system is a calculation system that operates with a base of 2. Each individual unit in the system is referred to as a bit, which stands for binary digit. This number system represents values using two distinct symbols: 0 and 1. For example, the decimal number 4 is expressed as 100 three-digit binary or 0100 in four-digit binary or 00100 in five-digit binary. Noted, these leading 0s in binary do not alter the value. There is way to convert binary to decimal. For example, a binary of 10011. The conversion of binary to decimal is as follows. $1*2^4+0*2^3+0*2^2+1*2^1+1*2^0 = 16+0+0+2+1 = 19$. However, a calculation needs to be performed for the reverse conversion from decimal to binary. For instance, for the decimal number 19, the digit is divided by 2, and the remainder is recorded as follows:

- $\frac{19}{2} = 9$, remainder 1
- $\frac{9}{2} = 4$, remainder 1
- $\frac{4}{2} = 2$, remainder 0
- $\frac{2}{2} = 1$, remainder 0
- $\frac{1}{2} = 0$, remainder 1

Hence,the final binary value is 10011.

In contrast, the hexadecimal number system utilizes 16 unique symbols to represent numeric values, which include the ten Arabic numerals (0-9) and six letters (A-F). In the hexadecimal system, each position corresponds to a value from 0 to 15, where A represents 10, B represents 11, C represents 12, D represents 13, E represents 14, and F represents 15. For example, the decimal number 18 is expressed as 12 in hexadecimal. The conversion of hexadecimal to decimal, for instance, a hexadecimal of $6D$. The D is 13. The number 6 is calculated as $6 * 16_1 + 13 * 16_0 = 96$. Hence, the final result is $96 + 13 = 109$. The reverse conversion, from decimal to hexadecimal, is done by dividing the number by 16. For example, when converting the decimal number 109, the number is divided by 16, and the remainder is recorded as follows:

- $\frac{109}{16} = 6$, remainder 13
- $\frac{6}{16} = 0$, remainder 6

The digit 13 is D, hence, the hexadecimal for 109 is equivalent to $6D$.

III. Research Design

Fig. 1 shows the research design. There are four major processes which are proposing the LRSA models, then simulation the encryption and decryption of the proposed LRSAs with the the short messages, long message and result analysis. Firstly, we proposed two types of LRSA which is Gradatim LRSA and Optimized LRSA. Next, the experiment of encryption and decryption the short message using the proposed LRSAs to ensure the LRSAs is work in practice. By using LRSAs to encrypt the message, we can get the ciphertext. We also may obtain the message from the ciphertext by using the decryption process of LRSAs. Secondly, the distribution of the long message characters is counted. Next, the message is encrypted using LRSAs, which resulting the ciphertext. One will compare distributed histogram with the ciphertext.

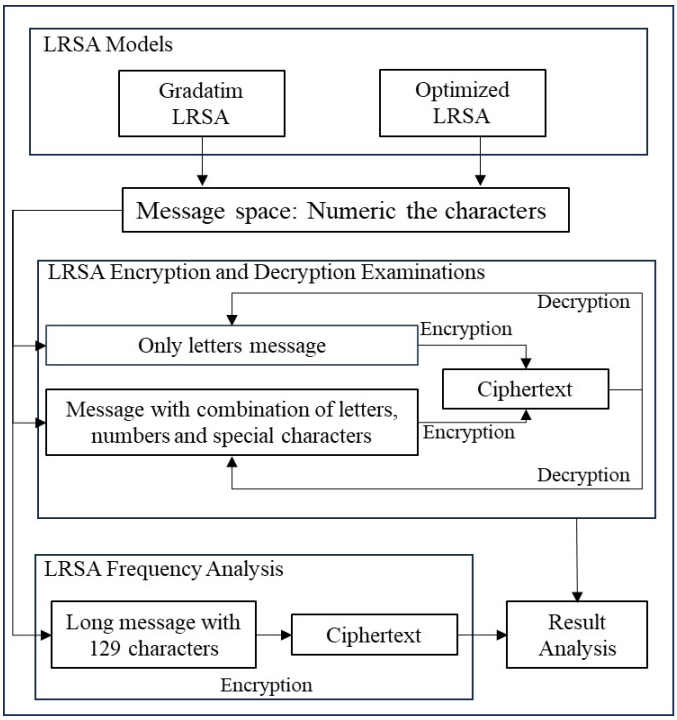

Fig. 1. Research design.

IV. Proposed LRSA

In this section, we present two designs of LRSA, known as Gradatim LRSA and Optimized LRSA. The Gradatim LRSA performs encryption and decryption character by character. The

Optimized LRSA has customized encryption and decryption at the stage of the RSA algorithm. The proposed hybrid encryption and decryption using the linear cipher and RSA algorithm to address the vulnerability of the linear cipher. This hybrid design allows for the implementation of more complex and robust security solutions.

A. Gradatim LRSA

The proposed hybrid encryption method uses both symmetric and asymmetric algorithms. The symmetric algorithm is a linear cipher, while the asymmetric algorithm is the RSA algorithm. We denote it as Gradatim LRSA, which means that the encryption and decryption processes are performed in a gradual and orderly manner.

The proposed hybrid Gradatim encryption as shown in Algorithm 1. By combining the linear cipher with the RSA algorithm, the message is first encoded using the linear cipher and then the resulting ciphertext is further encrypted using the RSA algorithm. This double encryption process prevents the detection of statistical patterns in the message.

Algorithm 1 Gradatim LRSA - Encryption Process

Require: Input: Message, m_{ml}.
1: Chooses the message space M.
2: Chooses the message m with the length of ml.
3: Selects secret keys, a and b, a must be coprime with M.
4: Selects two positive prime numbers, p and q, subjected p, $q > 2$.
5: Calculates $n = p * q$.
6: Calculates $\varphi(n)$.
7: Chooses a positive integer e, where e is coprime with $\varphi(n)$.
8: **for** $i \leftarrow 1$ to ml **do**
9: Calculates ciphertext, $c_i = (a * m_i + b \ mod \ M)^e$ mod n.
10: **end for**
11: **Output:** Ciphertext, c_{ml}.

The proposed hybrid Gradatim decryption as shown in Algorithm 2. The ciphertex is first decrypted using the RSA algorithm and then the resulting output is further decoded using the linear cipher.

Algorithm 2 Gradatim LRSA - Decryption Process

Require: Input: Ciphertext, c_{ml}.
1: Given cipertext, c_{ml}.
2: With existing secret keys, a and b.
3: Calculates private key, $d = e^{-1} \ mod \ \varphi(n)$, gcd(d, e) = 1.
4: **for** $i \leftarrow 1$ to ml **do**
5: Calculates message, $m_i = a^{-1} * ({c_i}^d \text{ mod } n - b) \text{ mod } M$.
6: **end for**
7: **Output:** Message, m_{ml}.

B. Optimized LRSA

The proposed a hybrid Optimized LRSA that uses a symmetric linear cipher and an asymmetric RSA algorithm. The modification occurs at the stage of the RSA algorithm, as shown in Algorithm 3. The message is encoded using the linear cipher. The resulting output is transformed into binary and concatenated. If the length of the message is not an even number, the string is padded with seven zeros. Noted it is the character 'A' as shown in Table I. For every 14 bits, the data is further encrypted with the RSA algorithm.

Algorithm 3 Optimized LRSA - Encryption Process

Require: Input: Message, m_{ml}.
1: Chooses the message space M.
2: Chooses the message m with the length of ml.
3: Selects secret keys, a and b, a must be coprime with M.
4: Selects two positive prime numbers, p and q, subjected p, $q > 2$.
5: Calculates $n = p * q$.
6: Calculates $\varphi(n)$.
7: Chooses a positive integer e, where e is coprime with $\varphi(n)$.
8: **for** $i \leftarrow 1$ to ml **do**
9: Calculates intermediate ciphertext, $k_i = (a * m_i + b) \text{ mod } M$.
10: **end for**
11: **for** $i \leftarrow 1$ to ml **do**
12: Converts intermediate ciphertext, k_i into binary.
13: **end for**
14: **if** ml is even number **then**
15: **for** $i \leftarrow 2$ to ml **do**
16: Concatenates intermediate ciphertext, such that $k_i||k_{i+1}$, we denoted it as K_i.
17: Convert K_i into decimal number, we denoted it as D_i.
18: Calculates ciphertext, $c_i = (D_i)^e$ mod n.
19: **end for**
20: **Output:** Ciphertext, $c_{\frac{ml}{2}}$.
21: **else**
22: Creates a temporary binary in such a way that k_{ml+1} is 0000000. The decimal value is 0. The alphabet is A.
23: **for** $i \leftarrow 2$ to $ml + 1$ **do**
24: Concatenates intermediate ciphertext, such that $k_i||k_{i+1}$, we denoted it as K_i.
25: Convert K_i into decimal number, we denoted it as D_i.
26: Calculates ciphertext, $c_i = (D_i)^e$ mod n.
27: **end for**
28: **Output:** Ciphertext, $c_{\frac{ml+1}{2}}$.
29: **end if**

The proposed hybrid Optimized LRSA decryption is shown in Algorithm 4. The ciphertext is first decrypted using the RSA algorithm. The resulting output is transformed into binary and divided into 7 bits per character. If the length of the message is an odd number, the last 7 bits are discarded. Then, the bit values are converted into decimal; these decimal values are further decoded using the linear cipher.

V. Numeric Character

The character variable with letter or symbol or numeric values is converted into a predefined set of rules numerical value as shown in Table I. For example the alphabet 'A' is equal to 0, 'B' is equal to 1, 'C' is equal to 2, and so on.

Algorithm 4 Optimized LRSA - Decryption Process

Require: Input: Ciphertext, $c_{\frac{ml}{2}}$ or $c_{\frac{ml+1}{2}}$.
With existing secret keys, a and b.
Calculates private key, $d = e^{-1} \mod \varphi(n)$, gcd(d, e) = 1.
if Ciphertext is $c_{\frac{ml}{2}}$ **then**
  for $i \leftarrow 1$ to $\frac{ml}{2}$ **do**
    Calculates intermediate ciphertext, $D_i = {c_i}^d \mod n$.
    Converts D_i into 14 bits binary, we denoted it as K_i.
    Splits the K_i into two 7 bits binary, such that k_{i*2-1} and k_{i*2}.
    Converts k_{i*2-1} and k_{i*2} into decimal number, we denoted it as D'_{i*2-1} and D'_{i*2} respectively.
    Calculates the message, $m_{i*2-1} = a^{-1} * (D'_{i*2-1} - b) \mod M$.
    Calculates the message, $m_{i*2} = a^{-1} * (D'_{i*2} - b) \mod M$.
  end for
Output: Message, m_{ml}.
else
  for $i \leftarrow 1$ to $\frac{ml+1}{2}$ **do**
    Calculates intermediate ciphertext, $D_i = {c_i}^d \mod n$.
    Converts D_i into 14 bits binary, we denoted it as K_i.
    Splits the K_i into two 7 bits binary, such that k_{i*2-1} and k_{i*2}.
    Converts k_{i*2-1} and k_{i*2} into decimal number, we denoted it as D'_{i*2-1} and D'_{i*2} respectively.
    Calculates the message, $m_{i*2-1} = a^{-1} * (D'_{i*2-1} - b) \mod M$.
    Calculates the message, $m_{i*2} = a^{-1} * (D'_{i*2} - b) \mod M$.
  end for
  Discarded m_{ml+1}.
Output: Message, m_{ml}.
end if

We have 26 upper and lower case letters each. There are ten numbers, nice special characters as well as space. For space, we denoted it as as '△'. Noted that this table can be enlarge if needed, which means the set space of message M is not fix.

VI. Simulation Result

This section provides a comprehensive overview of the simulation results obtained from the encryption and decryption processes conducted for the proposed LRSA (Gradatim LRSA and Optimized LRSA). The simulation involved processing messages of both short and long lengths to assess how the LRSA algorithm performs in encryption and decryption.

A. Gradatim LRSA - Encryption and Decryption

Section IV-A presents the Gradatim LRSA algorithm. Here, we show two examinations of encryption and decryption calculations. Both case studies have prime numbers for RSA, with $p = 31$, $q = 3$ and the public exponent e equal to 7. The private exponent d is the modular multiplicative inverse of e mod $(31 - 1)(3 - 1)$, resulting in 43.

TABLE I. Numeric the Alphabets, Numbers and Special Characters

A	0	**B**	1	**C**	2	**D**	3	**E**	4	**F**	5
G	6	**H**	7	**I**	8	**J**	9	**K**	10	**L**	11
M	12	**N**	13	**O**	14	**P**	15	**Q**	16	**R**	17
S	18	**T**	19	**U**	20	**V**	21	**W**	22	**X**	23
Y	24	**Z**	25	**a**	26	**b**	27	**c**	28	**d**	29
e	30	**f**	31	**g**	32	**h**	33	**i**	34	**j**	35
k	36	**l**	37	**m**	38	**n**	39	**o**	40	**p**	41
q	42	**r**	43	**s**	44	**t**	45	**u**	46	**v**	47
w	48	**x**	49	**y**	50	**z**	51	**0**	52	**1**	53
2	54	**3**	55	**4**	56	**5**	57	**6**	58	**7**	59
8	60	**9**	61	**'**	62	**?**	63	**!**	64	**.**	65
△	66	**,**	67	**;**	68	**@**	69	**"**	70	**:**	71

1) Experiment - 1: Assume the message is only capital letter, such that "COMPUTE". Based on Table I, we know the capital letter is from 0 until 25, therefore, the set space, M is equal to 26. Lets, secret keys of linear cipher are $a = 3$ and $b = 10$.

Table II shows the encryption of the text message "COMPUTE" using Gradatim LRSA. The ciphertext is in numeric form, which is "70 0 80 48 9 54", or in hexadecimal "46 00 50 30 09 36 34".

TABLE II. Gradatim LRSA Encryption "COMPUTE"

Message	C	O	M	P	U	T	E
Numeric, m	2	14	12	15	20	19	4
$m' = (3 * m + 10)$ **mod** 26	16	0	20	3	18	15	22
$(m')^7$**mod** 93	70	0	80	48	9	54	52
Ciphertext	70	0	80	48	9	54	52
Hexadecimal	46	00	50	30	09	36	34

Table III shows the decryption of the corresponding ciphertext "70 0 80 48 9 54 52" using Gradatim LRSA. The ciphertext is decrypted using RSA algorithm together with private key d, followed by linear cipher, $(3^{-1} * c - 10)$ mod 26. It is noted that the inverse of 3 mod 26 is based on Extended Euclidean algorithm, resulting in 9.

TABLE III. Gradatim LRSA Decryption Corresponding Ciphertext of "COMPUTE"

Ciphertext	70	0	80	48	9	54	52
$c' = c^{43}$**mod** 93	16	0	20	3	18	15	22
$m = 9(c' - 10)$ **mod** 26	2	14	12	15	20	19	4
Message	C	O	M	P	U	T	E

Table III shows the decryption of the corresponding ciphertext "70 0 80 48 9 54 52" using LRSA. The ciphertext is decrypted using RSA algorithm together with private key d,

followed by linear cipher, $(3^{-1} * c - 10) \bmod 26$. It is noted that the inverse of 3 mod 26 is based on Extended Euclidean algorithm, resulting in 9.

2) Experiment - 2: Assume the message is like a six character of password, such that "C0!fe@". The password is the combination of upper case letter, lower case letter, number and special character as shown in Table I. Therefore, the set space, M is equal to 72. Lets, secret keys of linear cipher are $a = 5$ and $b = 10$.

Table IV shows the encryption of the text message "C0!fe@" using Gradatim LRSA. The ciphertext is in numeric form, which is "80 60 75 42 70 67", or in hexadecimal "50 3C 4B 2A 46 43".

TABLE IV. GRADATIM LRSA ENCRYPTION "C0!FE@"

Message	C	0	!	f	e	@
Numberic, m	2	52	64	31	30	69
$m' = 5 * m + 10$ **mod** 72	20	54	42	21	16	67
$(m')^7$ **mod** 93	80	60	75	42	70	67
Ciphertext	80	60	75	42	70	67
Hexadecimal	50	3C	4B	2A	46	43

Table V shows the decryption of the corresponding ciphertext "80 60 75 42 70 67" using Gradatim LRSA. The ciphertext is decrypted using the RSA algorithm together with private key d, followed by linear cipher, $(5^{-1} * c - 10) \bmod 72$. It is noted that the inverse of 5 mod 72 is based on Extended Euclidean algorithm, resulting in 29.

TABLE V. GRADATIM LRSA DECRYPTION CORRESPONDING CIPHERTEXT OF "C0!FE@"

Ciphertext	80	60	75	42	70	67
$c' = c^{43}$ **mod** 93	20	54	42	21	16	67
$m = 29(c' - 10)$ **mod** 72	2	54	42	31	30	69
Message	C	0	!	f	e	@

Table V shows the decryption of the corresponding ciphertext "80 60 75 42 70 67" using Gradatim LRSA. The ciphertext is decrypted using the RSA algorithm together with private key d, followed by linear cipher, $(5^{-1} * c - 10) \bmod 72$. It is noted that the inverse of 5 mod 72 is based on Extended Euclidean algorithm, resulting in 29.

B. Optimized LRSA - Encryption and Decryption

Section IV-B presents the Optimized LRSA algorithm. Here, we show two examinations of encryption and decryption calculations with the word of "COMPUTE" and "C0!fe@" respectively. Both case studies have prime numbers for RSA, with $p = 127$, $q = 131$ and the public exponent e equal to 19. The private exponent d is the modular multiplicative inverse of $e \bmod (127 - 1)(131 - 1)$, resulting in 7759.

1) Experiment - 1: The experiment message is "COMPUTE". Based on Table I, we know the capital letter is from 0 until 25, therefore, the set space, M is equal to 26. The secret keys of linear cipher are $a = 3$ and $b = 10$.

Table VI shows the encryption of the text message "COMPUTE" using Optimized LRSA. The ciphertext is in numeric form, which is "6541 7480 11085 4721", or in hexadecimal '198D 1D38 2B4D 1271".

Table VII shows the decryption of the corresponding ciphertext "6541 7480 11085 4721" using Optimized LRSA. The ciphertext is decrypted using the RSA algorithm with the private key (d), which is 7759. The decimal output is converted into binary and further divided into 7 bits per character. Since the number of characters is odd, the last 7 bits are discarded. Next, the binary data is converted into decimal and decoded using a linear cipher.

2) Experiment - 2: The experiment 2, the word is a combination of upper case letter, lower case letter, number and special character, which is "C0!fe@". Therefore, the set space, M is equal to 72 as shown in Table I. Lets, secret keys of linear cipher are $a = 5$ and $b = 10$.

Table VIII shows the encryption of the text message "C0!fe@" using Optimized LRSA. The ciphertext is in numeric form, which is "15535 9394 1328", or in hexadecimal "3CAF 24B2 0530".

Table IX shows the decryption of the corresponding ciphertext "15535 9394 1328" using Optimized LRSA. The ciphertext is decrypted using the RSA algorithm along with the private key (d), which is 7759. The decimal output is then converted into binary and further divided into 7 bits per character. Since the number of characters is even, all bits are utilized. Next, the binary data is converted back into decimal and encoded using a linear cipher, such that $(5^{-1} * c - 10) \bmod 72$. It should be noted that the inverse of 5 modulo 72, calculated using the Extended Euclidean algorithm, is 29.

Table IX shows the decryption of the corresponding ciphertext "80 60 75 42 70 67" using LRSA. The ciphertext is decrypted using RSA algorithm together with private key d, followed by linear cipher, such that $5^{-1} * c - 10 \bmod 72$.

VII. FREQUENCY ANALYSIS

Frequency analysis is a technique used in cryptanalysis to determine the frequency of characters in an encrypted text. By analyzing the frequencies of letters in a message, cryptanalysts can make educated guesses about the substitution cipher being used to encrypt the text. The linear cipher is vulnerable to frequency analysis. By determining which characters occur most frequently in encrypted text, one may deduce the original message and crack the code.

For the frequency experiment, the chosen text message contains 129 characters. The message is: 'The linear cipher relies on a linear mathematical operation, such as multiplication and addition to scramble and decode messages.' The distribution of character is shown in Fig. 2. The highest distribution is of the space symbol ($\triangle$), which appears 18 times, followed by character 'a', which appears 14 times. The lowest distribution is of the characters 'T', 'b', 'g', comma symbol, and the full stop symbol, each of which appears once.

Lets examine if the text message is encoded using linear cipher. The set space, M is equal to 72 and the secret keys of linear cipher are $a = 5$ and $b = 10$.

TABLE VI. OPTIMIZED LRSA ENCRYPTION "COMPUTE"

Message, m	C	O	M	P	U	T	E
Numeric, m	2	14	12	15	20	19	4
$k = (3 * m + 10)$ **mod** 26	16	0	20	3	18	15	22
Binary, k	0010000	0000000	0010100	0000011	0010010	0001111	0010110
Concatenation, K	00100000000000		00101000000011		00100100001111		0010110<u>0000000</u>*
Decimal, D	2048		2563		2304		2816
$c = (D)^{19}$**mod** 16637	6541		7480		11085		4721
c **in hexadecimal**	198D		1D38		2B4D		1271

Notes: The underline binary is the concatenation for single character.

TABLE VII. OPTIMIZED LRSA DECRYPTION CORRESPONDING CIPHERTEXT OF "COMPUTE"

Ciphertext, c'	6541		7480		11085		4721
$D = c'^{7759}$**mod** 16637	2048		2563		2304		2816
Binary, K	00100000000000		00101000000011		00100100001111		0010110<u>0000000</u>*
Splitting, k	0010000	0000000	0010100	0000011	0010010	0001111	0010110
c'	16	0	20	3	18	15	22
$m = 9(c' - 10)$ **mod** 26	2	14	12	15	20	19	4
Message, m	C	O	M	P	U	T	E

Notes: The underline binary is the leftover binary, it is discarded.

TABLE VIII. OPTIMIZED LRSA ENCRYPTION "C0!FE@"

Message, m	C	0	!	f	e	@
Numberic, m	2	52	64	31	30	69
$k = 5 * m + 10$ **mod** 72	20	54	42	21	16	67
Binary, k	0010100	0110110	0101010	0010101	0010000	1000011
Concatenation, K	00101000110110		01010100010101		00100001000011	
Decimal, D	2614		5397		2115	
$c = (D)^{19}$ **mod** 16637	15535		9394		1328	
c **in hexadecimal**	3CAF		24B2		0530	

TABLE IX. OPTIMIZED LRSA DECRYPTION CORRESPONDING CIPHERTEXT OF "C0!FE@"

Ciphertext, c'	15535		9394		1328	
$D = c'^{7759}$ **mod** 16637	2614		5397		2115	
Binary, K	00101000110110		01010100010101		00100001000011	
Splitting, k	0010100	0110110	0101010	0010101	0010000	1000011
c'	20	54	42	21	16	67
$m = 29(c' - 10)$ **mod** 72	2	54	42	31	30	69
Message, m	C	0	!	f	e	@

The ciphertext is: 'hfQ0zk9Q;J0Gk:fQJ0JQzkQO0△90 ;0zk9Q;J04;TfQ4;TkG;z0△:QJ;Tk△950OQGf0;O04YzTk:zkG; Tk△ 90;9L0;LLkTk△ 90T△ 0OGJ;9BzQ0;9L0LQG△ LQ04Q OO;aQOv'. The distribution of character ciphertext is shown in Fig. 3. The highest distribution is of the zero, which appears 18 times, followed by semicolon symbol, which appears 14 times. One can conclude that the zero is the ciphertext for the character space symbol (△), while zero is the ciphertext for the character 'a'. Once, we know the ciphertext with the corresponding character, we can form the linear equations and find the secrect key a and b.

Next, we encrypt the text message using Gradatim LRSA and Optimized LRSA respectively. For both simulations, the set space M is equal to 72 and the secret keys of linear cipher are $a = 5$ and $b = 10$. The different only the RSA algorithm parameters.

For the Gradatim LRSA, the RSA two prime numbers are $p = 31$, $q = 3$ and the public exponent e equal to 7. The ciphertext is in hexadecimal form:

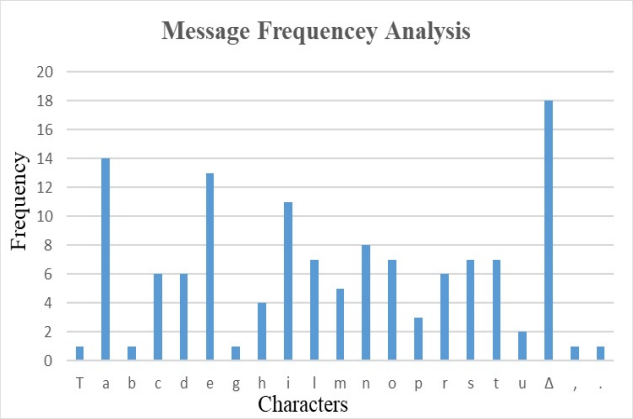

Fig. 2. Message frequency analysis.

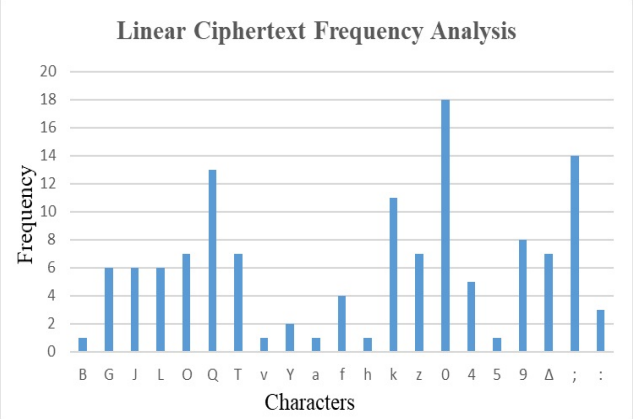

Fig. 3. Linear ciphertext frequency analysis.

‘421F464912243D464448490624291F4648494846122446324
94E3D49444912243D464448493844071F4638440724064412
494E2946484407244E3D39493203061F49443249380312422
4291224064407244E3D49443D2C49442C2C2407244E3D490
74E4932064844380112464944 3D2C492C46064E2C46493846
3232441A463208’.

For the Optimized LRSA, the RSA two prime numbers are $p = 127$, $q = 131$ and the public exponent e equal to 19. The ciphertext is in hexadecimal form: ‘037D03B33D7F0B202271252B261F05B319C20A913D7F2A
F111CB23E738CA3D7F0B2022713F6D15CF05D330130C45
09962A13385FZBCE15CF04D0222A18341DE9033638240B8
A22D4261F3D7F09960C452731033600A1033635AA32F704
D032A90F1706FC2C4507E4364736D6303F1DC02386137B2
3863F6D2AF1382433E212D0’.

Based on the ciphertext generated using Gradatim LRSA and Optimized LRSA, one may not be able to form the frequency distribution as the output ciphertext exceeds the numeric values presented in Table I. For example, when encrypting the character ‘o’ using Gradatim LRSA, the ciphertext value is 78. Therefore, we provide the ciphertext in hexadecimal form. This also indicates that both Gradatim LRSA and Optimized LRSA are not vulnerable to frequency attacks.

VIII. Discussion

Algorithms for encryption and decryption may protect data confidentiality. Encryption scrambles the message into unreadable ciphertext, while decryption is the reverse process of encryption. The proposed Gradatim LRSA and Optimized LRSA include both algorithms for encryption and decryption, as shown in Algorithms 1, 2, 3, and 4, respectively. The experiments in Section III demonstrate that both models effectively perform encryption and decryption.

Based on Table IV and Table VIII, the number of calculations for the second layer of the RSA algorithm shows that the Optimized LRSA is less than the Gradatim LRSA. If the length of the message is ml, the Gradatim LRSA must perform ml operations for the first layer of encryption (linear cipher) and ml operations for the second layer of encryption (RSA algorithm). However, for the Optimized LRSA, the number of calculations is only ml for the first layer (linear cipher) and $\frac{ml}{2}$ for the second layer (RSA algorithm). However, if the ml is a odd number, the Optimized LRSA will perform $\frac{ml+1}{2}$ operations for the second layer encryption. This shows that Optimized LRSA executes faster compared to Gradatim LRSA.

The linear cipher employs a linear mathematical operation to encode a message into unreadable ciphertext, with the frequency of characters in the message remaining the same in the ciphertext but with different characters. Based on the analysis in Section VII, showing that linear cipher is vulnerable to frequency analysis attacks. Hence, the proposed the hybrid encryption method using the linear cipher and RSA algorithm can resist the frequency attacks.

IX. Conclusion and Future Work

In conclusion, this research proposed a secure hybrid encryption method method that combines the strength of a linear cipher with the robust security of the RSA algorithm making it significantly more difficult for attackers to break the encryption. We name it as LRSA. We proposed models: Gradatim LRSA and Optimized LRSA. Optimized LRSA executes fewer rounds compared to Gradatim LRSA. However, both LRSAs can encrypt the message using secret keys from a linear cipher and public keys from the RSA algorithm. The encrypted output is the ciphertext. The LRSAs are also capable of decrypting the ciphertext and recovering the original message without compromising its security.

Existing linear cipher is vulnerable to the frequency attacks where the attackers are able to decipher the messages based on the frequency of occurrence of the characters in the ciphertext. But, the proposed LRSAs are resistance to the frequency attacks. As the proposed LSRAs add an addition layer of complexity to the encryption process, which is RSA algorithm. Hence, LRSAs provide a comprehensive and effective encryption solution that prioritizes the security for protecting sensitive information across different applications. The hybrid encryption method ensure the confidentiality of the messages only available for the authorize parties.

As for the nature of the linear cipher, which only allows for the encryption and decryption of English characters, this presents a major limitation of the proposed design. Therefore,

as future work, one plan is to explore hybrid encryption and decryption methods that can accommodate a broader range of languages, such as Mandarin, Jawi, and Japanese. This work may create a more universal encryption solution that meets the needs of multilingual users.

ACKNOWLEDGMENT

The authors would like to thank Guangdong University of Science & Technology, China, Rabdan Academy, United Arab Emirates.

REFERENCES

[1] C. Paar, J. Pelzl and T. Güneysu, *Introduction to cryptography and data security*, In Understanding Cryptography: From Established Symmetric and Asymmetric Ciphers to Post-Quantum Algorithms, pp. 1-35, 2024.

[2] R. Verma, A. Kumari, A. Anand and V.S.S. Yadavalli, *Revisiting shift cipher technique for amplified data security*, Journal of Computational and Cognitive Engineering, 3(1), pp. 8-14, 2024.

[3] S. Yalamati, *Data Privacy, Compliance, and Security in Cloud Computing for Finance*, In Practical Applications of Data Processing, Algorithms, and Modeling, pp 127-144, 2024.

[4] J. Pool, S. Akhlaghpour, F. Fatehi and A. Burton-Jones, *A systematic analysis of failures in protecting personal health data: a scoping review*, International Journal of Information Management, 74, pp. 102719, 2024.

[5] D. O. Ogundipe, *The impact of big data on healthcare product development: A theoretical and analytical review*, International Medical Science Research Journal, 4(3), pp. 341-360, 2024.

[6] S. Agarwal, P. Ghosh, T. Ruan and Y. Zhang, *Transient Customer Response to Data Breaches of Their Information*, Management Science, 2024.

[7] X. Wang, J. Yan, T. P. Munyon and T. R. Crook, *Breached But Not Broken: How Attributional Information Shapes Shareholder Reactions to Firms Following Data Breaches*, Corporate Reputation Review, pp. 1-22, 2024.

[8] Y. Guo, C. Wang and X. Chen, *Functional or financial remedies? The effectiveness of recovery strategies after a data breach*, Journal of Enterprise Information Management, 37(1), pp. 148-169, 2024.

[9] S. Chakraborty, C. Jackson, M. Frazier and K. Clark, *A Study on Password Protection and Encryption in the era of Cyber Attacks*, In SoutheastCon 2024, IEEE, pp. 1-5, 2024.

[10] V. Sasikala and B. S. CH, *Data Leakage Detection and Prevention Using Ciphertext-Policy Attribute Based Encryption Algorithm*, In 2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO), IEEE, pp. 1460-465, 2024.

[11] N. Ferguson, B. Schneier and T. Kohno, *Cryptography Engineering: Design Principles and Practical Applications*, Wiley, 2010.

[12] R. Das and G. Goldsztein, *Mathematics Behind the RSA Algorithm*, Journal of Student Research, 12 (1), 2023.

[13] K. Balasubramanian and M. P. Pitchai, *A survey of fermat factorization algorithms for factoring RSA composite numbers*, Multidisciplinary Science Journal, 6, 2024.

[14] S. Tanwar, R. Balavenu, H. H. Ramesha, M. Tiwari, K. K. Ramachandran and D. K. J. B. Sain, *Applied Cryptography in Banking and Financial Services for Data Protection*, Computer Science Engineering and Emerging Technologies: Proceedings of ICCS, 59, 2024.

[15] A. Desianty and M. I. Imelda, *Systematic Literature Review: Cybersecurity by Utilizing Cryptography Using the Data Encryption Standard (DES) Algorithm*, Jurnal Teknik Informatika, 17(1), pp.30-39, 2024.

[16] J. Schwenk, *Cryptography: Confidentiality. In Guide to Internet Cryptography: Security Protocols and Real-World Attack Implications*, Cham: Springer International Publishing, pp.13-41, 2022.

[17] H. B. Wolfe, *Cryptography: Protecting confidentiality, integrity and availability of data*, In Internet and Intranet Security Management: Risks and Solutions, pp.141-162, 2000.

[18] D. Mahto, D. A. Khan and D. K. Yadav,*Security analysis of elliptic curve cryptography and RSA*, In Proceedings of the world congress on engineering, 1, pp.419-422, 2016.

[19] W. Stallings, *Cryptography and network security:principles and practices*, Pearson Education India, 2006.

[20] B. J. LaMeres, *Number Systems*, In Introduction to Logic Circuits & Logic Design with Verilog, pp. 7-41, 2023.