International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 23s, 2025
https://www.theaspd.com/ijes.php

Computation Of RSA Algorithm Using Gaussian Graceful Labeling

S. Kavitha[1], G. Jayalalitha[2]
[1*]Research Scholar, Department of Mathematics, Vels Institute of Science, Technology and Advanced Studies (VISTAS), Pallavaram, Chennai, Tamil Nadu, India-600117. kavithasundaram55@gmail.com
[2] Professor, Department of Mathematics, Vels Institute of Science, Technology and Advanced Studies (VISTAS), Pallavaram, Chennai, Tamil Nadu, India-600117. g.jayalalithamaths.sbs@velsuniv.ac.in

ABSTRACT
This study introduces a novel encryption mechanism leveraging the Gaussian Graceful Labeling technique, designed to enhance both the security and efficiency of cryptographic systems. The method builds upon the RSA cryptosystem, integrating Gaussian Graceful Labeling to generate robust and secure keys. To further optimize performance, the approach incorporates a forward and backward ciphertext method, which applies the key in a dual-directional process for encryption and decryption. In this system, plaintext is encrypted using a key derived from Gaussian Graceful Labeling, a technique rooted in graph theory and Gaussian integers, offering a high degree of cryptographic complexity. The same key is then used in reverse to decrypt the message. The forward and backward mechanism provides an additional layer of efficiency, reducing computational overhead while maintaining strong security guarantees. This dual-directional process, coupled with the intricate properties of Gaussian integers, ensures a cryptosystem that is both computationally efficient and resistant to modern cryptographic attacks.

1. INTRODUCTION

Cryptography is used to safeguard the secrets of the digital world. In a period not defined by digital connectivity and information exchange, the need to protect sensitive data and communications has never been more critical. But personal messages, financial transactions, government intelligence transactions, the transmission of confidential information through networks poses significant risks[1-3]. This is where cryptography emerges as the steadfast guardian of our digital realm.

The science of secure communication is known as cryptography, which comes from the Greek words "kryptos," which referring concealed, and "graphein," which referring writing. It is an ancient art that dates back thousands of years, when ancient civilizations employed secret codes and ciphers to safeguard their messages from prying eyes. However, in today's complex digital landscape, cryptography has evolved into an advanced and indispensable field of study, making the online interactions, financial systems, and sensitive communications secure against adversaries [4-7]. The fundamental principles of cryptography are non-disclosure, confidentiality, integrity, authentication, and non-repudiation. The information is kept secret from unauthorized eyes by maintaining confidentiality, which ensures that only authorized parties can access and decrypt it. Integrity checks the data's consistency to make confident it has not been tampered with or changed during transmission. By confirming the parties' identities, authentication helps stop impersonation and unauthorized access. Finally, non-repudiation guarantees that neither the sender nor the recipient can dispute the receipt of a message.

These objectives are accomplished using a variety of cryptographic techniques, including traditional ones like symmetric and asymmetric key encryption and more recent ones like public-key communications, digital signatures, and secure hash functions. The complexity and strength of these techniques count on the sensitivity of the data and the level of security required. In this digital age, cryptography acting as a vital role in securing online banking, e-commerce communication, confidential business exchanges, and the privacy of individuals. It also underpins the functioning of block chain technology, which has revolutionized industries such as finance, supply chain, and healthcare. Governments rely on cryptography to secure sensitive information, protect national welfare, and ensure the safety of critical infrastructure. Despite its formidable

protection, cryptography is not invincible. As technology advances, the attackers seeking different methods to breach cryptographic systems[8-14].

This ongoing battle between cryptographers and adversaries results in a constant evolution of encryption techniques and security measures. In this context, understanding the principles and applications of cryptography is essential for anyone involved in the digital realm, be it a cyber security professional, a software developer, or an informed individual seeking to protect their personal data. This comprehensive field of study continues to be a powerful force behind the progress and security of the digital world, instilling trust and reliability in our ever-connected society. As we look into deeper into the world of cryptography, we unravel the secrets that safeguard our digital infrastructure, ensuring a safer and more secure future for generations to come.

Cryptography is used to practice secure communication when other parties are present. With mathematical algorithms, data is encrypted and decrypted. Encrypting a message and decrypting it by the recipient are the two primary steps in the cryptography process. The encrypted process involves encrypting plaintext to produce cipher text, while the decrypted process involves decrypting ciphertext to recover the original plaintext. Understanding these key concepts is essential to understanding how cryptography works.

In the ever-evolving landscape of digital communication, cryptography stands as an unyielding bastion, protecting our most sensitive data from prying eyes. From ancient civilizations' secret codes to the modern marvels like RSA encryption [15], the art and science of cryptography have come a long way. This journey has been marked by relentless innovation, pushing the boundaries of mathematical complexities and computational abilities[16].

This exploration has highlighted the fundamental principles of cryptography – privacy, reliability, authentication, and non-repudiation. It has delved keen on the historical significance of ciphers, emphasizing their educational value while acknowledging their limitations in the face of contemporary threats. The RSA algorithm, with its elegant asymmetry, exemplifies the pinnacle of modern cryptographic techniques, ensuring secure communication in an interconnected world [17,18].

To decrypt a message, similar method and secret key as in the encryption method must be used. By ensuring that only those who are authorized and possess the key can decode and obtain the original plaintext, this enhances the security of the communication or data being sent.

The processes of encryption and decryption are essential to cryptography, which is the study of secure communication when other people are around. Since a new key is generated for every session under forward secrecy, it is challenging for an attacker to decrypt previous messages even if they manage to get their hands on the current key. Under backward secrecy, communications that are encrypted now or in the future cannot be decrypted using compromised keys from the past[19,20]. These procedures must be followed in order to guarantee the integrity and confidentiality of sensitive data.

II. LITERATURE SURVEY

FORWARD AND BACKWARD CIPHERS

Forward and backward ciphers are simple encryption techniques that engage rearranging the letters of a message in either a forward or backward direction to conceal its meaning. While these ciphers are straightforward and easy to understand, they offer minimal security and are not suitable for protecting sensitive information in modern cryptographic applications. Nonetheless, they can serve as educational tools or recreational puzzles.

Forward Cipher

The forward cipher, also known as a Caesar cipher, is among the most basic and well-established encryption methods available. It involves shifting each letter in plaintext down the alphabet by a fixed number of positions. A shift of three, for example, turns "A" into "D," "B" into "E," and so on. This process is continued until the end of the alphabet, when it returns to "A."

Example:

Plaintext: HELLO

Shift: 3
Ciphertext: KHOOR

Backward Cipher

The backward cipher, as the name suggests, is the reverse of the forward cipher. Instead of shifting letters forward, it shifts them backward in the alphabet.

Example:
Plaintext: WORLD
Shift: 1
Ciphertext: VNQKC

It's important to note that forward and backward ciphers offer minimal security as they are susceptible to brute force attacks due to their limited number of possible keys (26 in the English alphabet). Additionally, they lack the complexity needed to withstand modern cryptanalysis techniques. As such, these ciphers are mainly used for educational purposes, to teach the basic principles of encryption and foster an understanding of more sophisticated cryptographic techniques. In real-world scenarios, more robust encryption methods such as AES, RSA, or elliptic curve cryptography are employed to ensure secure communication and data protection.

We will use a straightforward substitution cipher to develop an algorithm for processing cipher text both forward and backward. A predefined key is used to replace every letter in the plaintext with another letter in the cipher text in a substitution cipher. Using the same key, the backward process will decrypt the cipher text reverse into the original plaintext after the forward process has encrypted the plaintext to create it.

Here is the algorithm:

1. Define the Key:

Choose a secret key, which is a mapping of every letter in the plaintext alphabet to an equivalent communication in the cipher text alphabet. The key should be reversible, meaning every letter in the cipher text should map back to the original letter in the plaintext during decryption.

2. Forward Process (Encryption):

Input: Plaintext
Key Output: Cipher text

a. Initialize an blank string to gather the secret communication text.
b. Loop from beginning to end each character in the plaintext.
c. If the character is a letter:
- Use the Key to find the equivalent letter in the cipher text alphabet.
- Append the cipher text letter to the cipher text string.

d. If the character is not a letter (e.g., space, punctuation), simply append it to the cipher text string unchanged.
e. Return the cipher text string.

3. Backward Process (Decryption):

Input: Cipher text
Key Output: Plaintext

a. Initialize an blank string to store the decrypted plaintext.
b. Loop through every character in the cipher text.
c. If the character is a letter:
- Use the Key to find the parallel letter in the plaintext alphabet.
- Append the plaintext letter to the decrypted plaintext string.

d. If the character is not a letter (e.g., space, punctuation), simply append it to the decrypted plaintext string unchanged.
e. Return the decrypted plaintext string.

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 23s, 2025
https://www.theaspd.com/ijes.php

GAUSSIAN GRACEFUL LABELING

Gaussian graceful labeling is a concept in graph theory, specifically in the study of graph labeling. A Gaussian graceful labeling of a graph is an project of distinct Gaussian integers to the vertices of the graph such that the set of absolute differences of the edge labels (also known as the edge differences). In other words, if we label the vertices with integers 0 to $(n-1)z$ (where n is equal to the number of vertices), then the set of edge differences should be $\{z, 2z, 3z, \ldots, (n-1)z\}$.

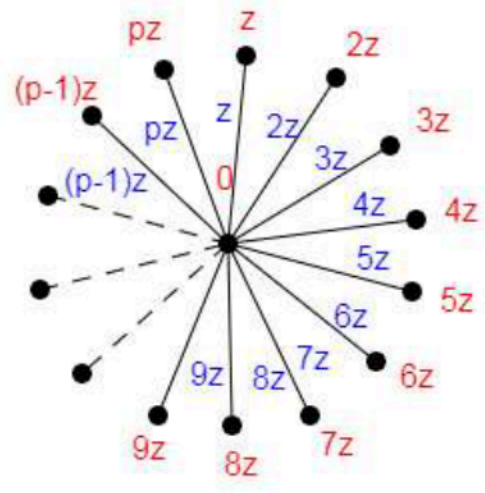

Fig.1 Gaussian graceful Labeling of a Star Graph

A Gaussian graceful labeling is a variation of graceful labeling where the edge differences are in arithmetic progression, meaning they form a sequence with a common difference. Specifically, in a Gaussian graceful labeling, the set of edge differences forms a Gaussian sequence or an arithmetic sequence with a constant second difference. Formally, a graph G= (n,m), a Gaussian graceful labeling is an assignment of labels to the vertices, say $L(v)$ for each vertex v, such that the set of differences |L(u) - L(v)|, for each edge (u, v), forms an arithmetic progression. Gaussian graceful labeling is an area of active research in graph theory, and it has connections with other mathematical concepts, such as number theory, combinatory, and coding theory. Many mathematicians and researchers have explored different types of graphs and their potential Gaussian graceful labelings.

RSA WORKING ALGORITHM

One well-known instance of asymmetric cryptography is the RSA algorithm, which is characterized by the use of both public and private keys. With the public key being freely shared and the private key being kept private, this dual-key system allows for secure communication.

RSA algorithm performance in action:

1. Usually when browsing the web, a client sends its public key to a server to start a conversation.
2. Data encrypted with the client's public key is transmitted by the server.
3. The client uses its private key to decrypt the data after receiving it.

Because it is asymmetric, only the authorized client will be able to decrypt the data that is transmitted, even in the event that an eavesdropper manages to obtain the client's public key.

The difficulty of factoring large integers is the fundamental component of RSA and the key to its security. The private key is generated using the same two large prime numbers that make up the public key. These prime numbers are multiplied together. As a result, the large composite number's impracticality is what keeps the private key secret. As a result, the security of RSA encryption increases proportionally with key size, with exponential encryption strength being increased by doubling or tripling the key length.

RSA keys usually have a length of 2048 or 1024 bits, with the latter being more vulnerable to future cracking as computing power increases. Although there have been predictions of upcoming vulnerabilities in 1024-bit keys, cracking them has proven to be an unbeatable obstacle thus far.

To send secret messages between the manager (M) and the employees (E) using the known integers N, p, and q, we can use a cryptographic technique called the RSA algorithm A popular asymmetric encryption technique that enables safe communication between parties who share public and secret keys is the RSA algorithm.

Here's how the RSA algorithm works for sending secret messages:

1. **Key Generation**
- The manager (M), who keeps them a secret, creates two large prime numbers, p and q.
- The manager determines and maintains the confidentiality of the modulus $N = p * q$.
- The Euler's Totient function, $\emptyset(N) = (p-1) * (q-1)$ is then calculated by the manager.

2. **Public and Secret key Pair**
- M chooses a public exponent (e) such that $e \in (1, \emptyset(N)$ and $\gcd(e, \emptyset(N)) = 1$
- M shares the public key (N, e) with all employees.

3. **Employee's Secret key**
- For each employee (E), M needs to calculate the corresponding secret key (d) secretly.
- The inverse of e modulo $\emptyset(N)$ is represented by d, meaning that

$d * e \equiv 1\ (mod\ \varphi(N))$.

4. **Encryption (Manager to Employee)**
- To send a secret message (m) to an employee (E), the manager (M) uses E's public key (N, e) to encrypt the message.
- The encrypted message (C) is calculated as: $C \equiv m^e\ (mod\ N)$.

5. **Decryption (Employee to Manager)**
- The employee (E) uses their secret key (d) to decrypt the received encrypted message (C).
- The decrypted message (M) is computed as: $m \equiv C^d (mod\ N)$.

6. Keep in mind that the security of RSA is derived from the difficulty of factoring large integers $(p\ and\ q)$ into their prime factors. As long as p and q are kept secret and are big enough, the RSA encryption is believed to be safe.

III. RESULTS

An additional layer of encryption can be added by using the RSA algorithm in conjunction with a combination of forward and backward ciphers, which will make it more difficult for unauthorized parties to decode the message. But it is essential to remember that the encryption's security is still needy on the strength of each individual cipher and the confidentiality of the keys used.

1. **Key Generation:**
- Two big prime numbers, p and q, are created by the manager (M), who keeps them a secret.
- The manager computes and maintains the secret module

$N = p * q\ and\ D = p - q$.
- The manager then calculates $\emptyset(N) = (p-1) * (q-1)$ to determine Euler's totient function.

2. **Public and Secret key Pair:**
- Manager chooses a public exponent (e) such that $\mathrm{e} \in (1, \emptyset(\mathrm{N})$ and $\gcd(\mathrm{e}, \emptyset(\mathrm{N})) = 1$
- Manager shares the public key (N, D, e) with all employees.

3. **Employee's Secret key:**
- For each employee (E), Manager needs to calculate the corresponding secret key (d) secretly.
- The modular multiplicative inverse of $e\ modulo\ \emptyset(N)\ is\ d * e \equiv 1\ (mod\ \emptyset(N))$.

4. **Encryption (Manager to Employee):**

- The manager (M) encrypts a secret message (m) that is sent to an employee (E) using the employee's public key $(\mathrm{N}, \mathrm{D}, \mathrm{e})$.
- The formula for computing the encrypted message (C) is

$\mathrm{C} \equiv ((\mathrm{m} + \mathrm{D})(\mathrm{mod}26))^{\mathrm{e}}(\mathrm{mod}\ \mathrm{N})$.

5. Decryption (Employee to Manager):

- The employee (E) utilizes their secret key (d) to unlock the encrypted communication that was received (C).
- The decrypted message (M) is computed as

$\mathrm{m} \equiv (\mathrm{C}^{\mathrm{d}}(\mathrm{mod}\ \mathrm{N}) - \mathrm{D})(\mathrm{mod}\ 26)$

The forward-backward cipher and RSA encryption add another level of complexity to the message, building it trickier for attackers to decode the original content. It's important to remember, though, that the power of the RSA algorithm and the confidentiality of the secret keys used for encryption and decryption remain the main factors determining the encryption's security.

It's fundamental to keep in mind that RSA's security depends on how hard it is to factor large integers like p and q into their prime factors. The RSA encryption is thought to be secure as long as p and q are kept secret and are sufficiently large.

Now let's examine the workings of the RSA algorithm:

Public Key Generation:

The first step in creating a public key with the RSA algorithm is to choose two prime numbers, P and Q. Using $P = 53\ and\ Q = 59$ as our examples, let's begin.

We then calculate the modulus for our public key, N, by taking the product of these prime numbers. $N = P * Q = 53 * 59 = 3127$ in this instance.

We now have to choose a small exponent, usually represented by the letter 'e', which is an integer that has no factors in common with the totient of N $(\emptyset(N))$. To keep things simple, let's assume $e = 3$.

N and e are the parts of our public key.

Secret Key Generation:

To generate the secret key, we first need to calculate the totient of N, denoted as $\emptyset(N)$. The formula for $\emptyset(N)$ is given by $(P-1)(Q-1)$.

Thus, $\emptyset(N) = (53-1)(59-1) = 3016$.

Now, we choose an integer value for 'k' and compute the secret key, denoted as *'d'*, using the formula: $d = (k * \emptyset(N) + 1) / e$.

Let's assume k = 2 for this example. Substituting the values, we get:

$d = (2 * 3016 + 1) / 3 = 6033 / 3 = 2011$.

So, the value of d for k = 2 is 2011, which serves as the secret key in our RSA encryption scheme.

Now we are ready with our – Public Key ($n = 3127, D = -6\ and\ e = 3$) and Secret key ($d = 2011$).

Now we will encrypt "HI":

First, we convert the letters to numbers using a standard mapping, where H corresponds to 8 (m1) and I corresponds to 9 (m2).

$H = 8 = m_1\ and\ I = 9 = m_2$

Thus Encrypted Data

$C_1 \equiv \left((m_1 + D)(mod26)\right)^{e}(mod\ N)$.

$= ((8-6)(mod\ 26))^3(mod\ 3127) = 2^3(mod\ 3127) = 8$

Encrypted Data

$C_2 \equiv ((m_2 + D)(mod26))^{e}(mod\ N)$.

$= ((9-6)(mod\ 26))^3(mod\ 3127) = 3^3(mod\ 3127) = 27$

Now we will decrypt $8\ and\ 27$

$m_1 \equiv \left({C_1}^{d}(mod\ N) - D\right)(mod26)$

$= ((8^{2011})(mod3127) + 6)(mod26) = (2+6)(mod\ 26) = 8$

$$m_2 \equiv \left({C_2}^d \, (mod\ N) - D\right)(mod 26)$$
$$= ((27^{2011})(mod 3127) + 6)(mod 26)$$
$$= (3 + 6)(mod\ 26) = 9$$
$$8 = H\ and\ I = 9\ i.e."HI".$$

IV. CONCLUSION

The integration of forward and backward ciphers with RSA encryption underscores the relentless pursuit of heightened security. By combining these methods, we created intricate layers of protection, making it exceedingly challenging for malicious actors to breach our communications. The meticulous processes of key generation, encryption, and decryption served as the backbone of this security framework, relying on the numerical complexity of prime numbers and modular arithmetic.

As we traversed deeper into the digital age, understanding and appreciating the nuances of cryptography become paramount. It is not merely the realm of cyber security professionals or software developers but knowledge essential for every individual navigating the digital sphere. In this symphony of algorithms and keys, the power to safeguard information lies not just in the hands of experts but in the collective understanding and responsible use of cryptographic techniques.

In the grand tapestry of our interconnected world, cryptography weaves the threads of trust and security. Through its application and continuous evolution, we carve a path toward a safer, more secure future, where privacy and integrity are not just ideals but tangible realities in our digital interactions. The journey of cryptography continues, promising a future where our secrets remain hidden, our data remains intact, and our digital realm remains secure.

REFERENCES

[1] R.L. Rivest, A. Shamir, and L.M. Adleman, "A Method for Obtaining Digital Signature and Public-key Cryptosystems," Communications of the ACM, Vol. 21, No. 2, pp. 120–126, 1978.

[2] Xin Zhou and Xiaofei Tang, "Research and implementation of RSA algorithm for encryption and decryption," Proceedings of 2011 6th International Forum on Strategic Technology, Harbin, Heilongjiang, 2011, pp. 1118-1121, doi: 10.1109/IFOST.2011.6021216.

[3] Ming-Der Shieh, Jun-Hong Chen, A new modular exponentiation architecture for efficient design of RSA cryptosystem, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2008.

[4] Diffie, Whitfield, and Martin E. Hellman. "New directions in cryptography." Democratizing Cryptography: The Work of Whitfield Diffie and Martin Hellman. 2022. 365-390.

[5] Aggarwal, D., & Maurer, U. (2009). Breaking RSA generically is equivalent to factoring. In Advances in Cryptology-EUROCRYPT 2009: 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings 28 (pp. 36-53). Springer Berlin Heidelberg.

[6] Levine, J., and Brawley, J.V. Some cryptographic applications of permutation polynomials. Cryptologia 1 (Jan. 1977), 76-92.

[7] Blömer, Johannes, and Alexander May. "A generalized Wiener attack on RSA." International Workshop on Public Key Cryptography. Springer Berlin Heidelberg, 2004.

[8] Williams, Henry. "A modification of the RSA public-key encryption procedure (Corresp.)." IEEE Transactions on Information Theory 26.6 (1980): 726-729.

[9] Galla, Lavanya K., Venkata SreeKrishna Koganti, and Nagarjuna Nuthalapati. "Implementation of RSA." 2016 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT). IEEE, 2016.

[10] Pfitzmann, Birgit, and Andreas Pfitzmann. "How to break the direct RSA-implementation of mixes." Advances in Cryptology—EUROCRYPT'89: Workshop on the Theory and Application of Cryptographic Techniques Houthalen, Belgium, April 10–13, 1989 Proceedings 8. Springer Berlin Heidelberg, 1990.

[11] Boneh, Dan, and Matthew Franklin. "Efficient generation of shared RSA keys." Advances in Cryptology—CRYPTO'97: 17th Annual International Cryptology Conference Santa Barbara, California, USA August 17–21, 1997 Proceedings 17. Springer Berlin Heidelberg, 1997.

[12] Zhou, Xin, and Xiaofei Tang. "Research and implementation of RSA algorithm for encryption and decryption." Proceedings of 2011 6th international forum on strategic technology. Vol. 2. IEEE, 2011.

[13] Boneh, Dan, and Hovav Shacham. "Fast variants of RSA." CryptoBytes 5.1(2002): 1-9.

[14] Milanov, Evgeny. "The RSA algorithm." RSA laboratories (2009): 1-11.

[15] Wiener, Michael J. "Cryptanalysis of short RSA secret exponents." IEEE Transactions on Information theory 36.3 (1990): 553-558.
[16] Boneh, Dan. "Twenty years of attacks on the RSA cryptosystem." Notices of the AMS 46.2 (1999): 203-213.
[17] Fiat, Amos. "Batch rsa." Advances in Cryptology—CRYPTO'89 Proceedings 9 (1990): 175-185.
[18] Gennaro, Rosario, et al. "Robust and efficient sharing of RSA functions." Advances in Cryptology—CRYPTO'96: 16th Annual International Cryptology Conference Santa Barbara, California, USA August 18–22, 1996 Proceedings 16. Springer Berlin Heidelberg, 1996.
[19] Gennaro, Rosario, Hugo Krawczyk, and Tal Rabin. "RSA-based undeniable signatures." Advances in Cryptology—CRYPTO'97: 17th Annual International Cryptology Conference Santa Barbara, California, USA August 17–21, 1997 Proceedings 17. Springer Berlin Heidelberg, 1997.
[20] Rabin, Tal. "A simplified approach to threshold and proactive RSA." Advances in Cryptology—CRYPTO'98: 18th Annual International Cryptology Conference Santa Barbara, California, USA August 23–27, 1998 Proceedings 18. Springer Berlin Heidelberg, 1998.