Pi International Journal of Mathematical Sciences
© PIJMS | Volume 1 | Issue 2 | October 2025 | ISSN: 3107-9830 (Online)
Available online at: https://pijms.com/
DOI: https://doi.org/10.59828/pijms.v1i2.8

Graph-Based Unified Cryptographic system for Secure Communication and Financial Transactions

Surbhi Soni[1], Dr. Binny Gupta[2], Hemant Gena[3]

[1,3]Scholar, Department of Mathematics, Sri Khushaldas University, Pilibanga, Hanumangarh

[2]Assistant Professor, Department of Mathematics, Sri Khushaldas University, Pilibanga, Hanumangarh, Raj.

Email: Surbhisoni95@gmail.com[1] | binnykakkar29@gmail.com[2]

Received: 05 October 2025 | **Accepted:** 26 October 2025 | **Published:** 30 October 2025

ABSTRACT

In today's increasingly interconnected world, ensuring the protection of digital information transmitted over unprotected communication networks —whether wired or wireless—is paramount. Encryption algorithms are central to protecting sensitive data, and hybrid encryption methods, which combine different encryption techniques, provide a stronger level of security compared to conventional single-key encryption methods. Although many Unified encryption models exist, many fall short in addressing diverse security challenges. This paper proposes a novel Unified encryption scheme that combines the widely-used public key cryptosystem (RSA) algorithm, a newly introduced Secret key cryptography (SKC) algorithm, and a Graph-Based Encryption Approach. This combination aims to provide more robust data privacy, integrity, and security. We believe this enhanced model offers an effective solution for secure communication in the face of evolving cyber threats.

Keywords: Cryptography, Unified Encryption, RSA, Secret Key (SKC), Graph-Based Encryption, Data Security, Information Privacy

1. Introduction

Cryptography is often described as the art and science of secret writing, transforming readable information into an unreadable format using various mathematical and computational techniques. Its core purpose is to ensure the privacy, accuracy, and verification of data during transmission are maintained. This is accomplished using techniques such as encryption, digital signatures, hash algorithms, and message authentication protocols, which together safeguard data privacy and protect it from unauthorized access and tampering [1]-[4]. Encryption works by converting plaintext into ciphertext, a format that is unreadable to outsiders. The procedure of converting encrypted data back into its original, readable form is called decryption, and this process generally relies on secret keys. The correct decryption key is essential for the authorized recipient to retrieve the original message. Within the domain of cryptography, encryption algorithms are

broadly classified into two types: symmetric and asymmetric encryption. Symmetric encryption relies on a single key for both encrypting and decrypting, establishing a shared secret between parties. In contrast, asymmetric encryption utilizes a pair of keys for secure communication, utilizes a key pair—public and private—to solve the challenge of key distribution. problem typically seen in symmetric encryption [5], [6]. Public key cryptosystems, such as RSA, ensure security by using public keys for encryption and private keys for decryption, minimizing the need for secret exchanges between users. While the use of asymmetric and systems individually has proven effective, the need for more robust and efficient cryptographic systems has spurred the development of unified encryption frameworks. A unified encryption system brings together elements of both symmetric and asymmetric techniques into a single, integrated approach. This hybridization provides enhanced security by leveraging the strengths of each individual method while addressing their weaknesses. Moreover, integrating graph theory with cryptographic algorithms has opened new avenues for security innovation. Graph-based cryptography utilizes graph structures to model complex relationships between keys and data, enhancing both encryption strength and computational efficiency.

Graph theory in cryptography introduces a visual and mathematical approach to key exchange, data flow, and encryption key management. By applying graph-based techniques, cryptosystems can model data flows and optimize key distribution mechanisms, ensuring higher security and scalability in complex networks. This innovative approach not only strengthens encryption protocols but also addresses real-world challenges related to the secure transmission of sensitive data across public and private networks. As cyber threats continue to evolve, a unified approach to cryptography combined with graph-theoretic techniques is poised to offer stronger defenses against potential security breaches. Given the increasing frequency of cyber-attacks and malicious activities in both private and public sectors, securing sensitive data are more crucial than ever. Traditional encryption methods often struggle to meet the growing demands for both performance and security. Therefore, a unified cryptographic system that leverages both classical techniques and modern mathematical innovations, such as graph theory, offers a promising solution to achieve secure, efficient, and scalable data protection [12], [13]. By combining multiple cryptographic techniques and applying advanced algorithms, these systems promise to provide greater confidentiality and robustness against future cyber threats [16]-[25].

This paper proposes a unified cryptographic framework that incorporates traditional cryptographic methods with graph theory, creating a novel, high-performance solution for securing communications and ensuring the safe transmission of sensitive data across networks.

2. Devised Algorithm

2.1 RAS and SKC tools

Symmetric keys can be implemented in two primary methods: stream ciphers and block ciphers. Block ciphers work by Converting a fixed-size data block into encrypted text, ensuring that the length of the output remains consistent with the input. For a standard message, the table typically includes 26 characters along with a blank space. However, since the size and number of characters are larger in this case, we also

incorporate special symbols into the table's construction. Each element within the table compartments is then assigned a numerical value.

Table-1

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
V	W	X	Y	Z	$space$	!	@	#	$	%	^	&	*	(	)	=	+	?	−	:
22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42

2.1.1 Deriving the SKC credentials: The modular reciprocal refers to an integer z that satisfies the equation.

$$cz \equiv 1(modn) \quad (1)$$

The value of z must lie within the set (1, 2, …, (n-1)), which represents the range of integers modulo n. Reciprocal of "c" modulo "n" exists only if "c" and "n" are coprime, meaning their greatest common divisor (gcd) is 1.

2.1.2 Deriving the RSA keys: It is crucial to select large prime numbers so that it becomes difficult for someone to determine them. Calculate m = p q

Use Carmichael function λ(m), where $a^{\lambda(m)} \equiv 1mod(m)$ and $gcd\ (a,m)\ \equiv 1$ (2)

Choose an integer i such that i is coprime to λ(m) and $1 < i < \lambda(m)$ (3)

The pair of values (m, i) forms the public key.

Identify ‘e’ with using the enhanced Euclidean method such that i. $e \equiv 1\ (mod\ \lambda(m))$. (4)

The pair (m,e) marks up the private key [19]

2.1.3 Encryption

In the encryption procedure, the RSA technique requires the plaintext along with the values of i and m. The message is first converted into integers corresponding to the alphabet 1 to 42. Given the the original message P, expressed as a numerical value, and the encrypted message C. is computed using the formula (5)

$$C = P^i mod\ m \quad (5)$$

2.1.4 Decryption

Using the private key, the decryption process is essentially the inverse of encryption. In the RSA method, decryption follows this reverse procedure, the ciphertext along with the values of e and n is required. The original message or the sender's content can be recovered using the following formula:

$$P = C^e mod\ m \quad (6)$$

2.2 Mathematical Foundations of Key Generation: RSA, Symmetric Cryptography, and Graph Theory

In the RSA algorithm, a key assumption is that factoring large numbers is a difficult task. This means that if you're given a large number n, it's challenging to figure out which prime numbers multiply together to create it. This difficulty is what makes RSA one of the most widely used asymmetric encryption systems.

2.2.1 SKC

The process of generating a secret Key (SKC) is much simpler. The procedure starts with choosing two integers—one positive and one negative—using 43 as a constant modulus. The steps to generate the key are:

a. Pick two integers, one positive and one negative, referred to as n_1 and n_2.
b. Compute the modular inverse of n with respect to 43 (this becomes key 1, denoted as k_1).
c. Then, calculate the modular inverse of n_1 with respect to 43(this results in key 2, denoted as k_2).

2.2.2 RSA

The process of generating RSA credentials is simple and adaptable. You begin by selecting two large prime numbers, p and q, and next, compute m by multiplying p and q. Unlike other methods, there is no fixed modulus number in RSA. The steps for key generation are as follows:

a. Pick two substantial prime numbers, p and q, selected unpredictably.
b. Calculate m and use Carmichael function λ(m), where $a^{\lambda(m)} \equiv 1 mod(m)$ and $gcd\ (a,m)\ \equiv 1$
c. Select an integer i so that i is coprime to λ(m) and $2 \leq i \leq \lambda(m)$.
d. Determine e, where $1 < e < \lambda(m)$ such that:

$$i * e \equiv 1(mod\lambda(m))$$

e. The public key is made up of (m, i), while the private key comprises (m, e).

2.2.3 Graph theory

Encryption Process

Step 1: Start by converting the original message into an encrypted form.

Step 2: Translate each character or element of the message (denoted as *M*) into its corresponding numerical value using a predefined conversion table. This results in a new sequence (denoted as *S*), where each character is replaced by its numeric equivalent. The encrypted version of the message will be represented as a complete bipartite graph.

Step 3: Construct a graph (denoted as *G*) based on the sequence *S*. The graph will consist of two sets of vertices (representing different elements of *S*) and edges connecting them based on the sequence. Each edge will have a weight corresponding to the numeric value of the element it represents.

Step 4: Transmit the graph *G* to the intended recipient. This graph now encodes the original message in a secure form, making it difficult to decipher without the proper decryption key or method.

Decryption Process

To decrypt the message, the process is reversed:

Step 5: Upon receiving the encrypted graph *G*, the receiver organizes the vertices of the graph according to their edge weights. The vertices should be arranged in ascending or descending order based on the weight of the connecting edges.

Step 6: After sorting the vertices, a sequence is formed that corresponds to the order in which the elements should be decoded.

Step 7: The receiver then identifies the labels of the vertices in the sequence and, by following the order, recovers the original message. Each label corresponds to a character or element from the original message, allowing the encrypted text to be reconstructed.

3. Implementation

3.1 Algorithm An algorithm is a systematic series of actions that must be performed in a particular sequence to obtain the intended result. The steps for the unified security algorithm are outlined in the following sections. Sections 3.1.1, 3.1.2, and 3.1.3 explain the processes for SKC, RSA and Bipartite graph respectively. Section 3.1.4 clarifies the process of encryption and decryption.

3.1.1. Using SKC

The implementation of Secret Key (SKC) encryption is straightforward and involves. Two variables: a non-negative integer and a negative integer. The procedure for the encryption workflow in SKC is detailed as follows:

a. Select a random integer, which can be either positive or negative → n_1, n_2

b. Calculate the inverse of n_1, n_2 mod 43 → k_1, k_2 // cryptographic element creation in SKC

c. unencrypted data → M

d. Encrypted message → C = (M × n_1 × n_2) mod 43 // SKC Encryption

3.1.2. Using RSA

The RSA encryption method is moderately complex and relies on two substantial prime numbers. Several computational steps, such as GCD, modular inverses, Carmichael function λ(m), and congruences, are involved. The RSA encryption steps are outlined below from A_1 to A_8:

A_1. Choose two prime numbers, $m \rightarrow pq$ / *RSA* encryption setup

A_2. Compute $b\hat{}\lambda(m) \equiv 1\ mod(m)$ and verify $gcd(b,m) \equiv 1$ (Carmichael function λ(m))

A_3. Randomly select i such that gcd (i, λ(m)) → 1

A_4. Ensure e and λ(m) are coprime

A_5 Identify e such that i * e → 1 (mod λ(m))

A_6. Verify that e is the modular inverse of i

A_7. The encryption key is → (i, m)

A8. The decryption key is → (e, m)

You said:

3.1.3 Using Graph based Encryption

This section presents an encryption method designed for secure communication between two parties. The algorithm leverages the structure of a bipartite graph below are the steps involved in the encryption process:

1. **Define the Set of Primes**: Consider a set $\mathbf{P}_n$ of the first "N" prime numbers, where **N** is calculated as ⌈(42/k) + k⌉, with **2 < k < 21** and **k** being a fixed key based on the length of the word. Now, let's proceed with encrypting a message of length **S**.
2. **Message Length and Table Setup**: For a message of length k, create a table of dimensions $(N-k) \times k$. where the first number represents the number of rows, and the second represents the number of columns.
3. **Partition the Alphabets**: The alphabets of the message are divided as follows: The first k prime positions are allocated horizontally. The next positions, from $k+1\ to\ N$, are placed vertically.

4. **Labeling the Alphabets**: Each alphabet is labeled with a pair of integers C_iR_i, where:
   - R_i is the row index.
   - C_i is the column index.
5. **Labeling the Entries**: The entries of the table are labeled using the combination of row and column indices (denoted as C_iR_i). For basic encryption, the table typically consists of the 26 letters of the alphabet along with a space. However, since we are working with a larger set of characters in this case, we also introduce additional symbols and special characters into the table. This allows us to support a broader range of characters for a more complex encryption scheme. For example, if k=6 then N=13 and the table of dimension 7×6 is as follows:

	2	3	5	7	11	13
17	A	B	C	D	E	F
19	G	H	I	J	K	L
23	M	N	O	P	Q	R
29	S	T	U	V	W	X
31	Y	Z	space	!	@	#
37	$	%	^	&	*	(
41	)	=	+	?	-	:

By this table we find that the value of A is 217, N obtains value 323 and space receives 531 etc.

6 **Constructing the Path Graph**: Every item in a compartment is given a numerical value. The column number is defined by the initial element, and the row number is determined by the final element. For example, if the letter 'A' is labeled with number 217, then calculate the edge set of the graph, which will look like pairs (c_1, r_1), (c_2, r_2), ..., (c_n, r_n).

7 **Graph Structure**: Now, construct a bipartite graph using these edges set. In this graph, edges moves from c_i to r_i.

Next, assign random weights to each edge in decreasing order and send the labeled graph.

Decryption Algorithm:

The recipient receives the labeled bipartite graph and begins by sorting the edges by their weights in descending order. They then reorder the edges into pairs of **(**column, row**)** based on the weight sequence. Finally, use the table to extract the corresponding alphabet, thus decrypting the message.

4. Example

To gain a clearer understanding of the suggested hybrid method, the original message "S$T@A%R" was selected for experimentation. This example allows us to test how the method handles both letters and numbers, providing insights into its practical effectiveness.

4.1 key generation of SKC

We selected n_1=5 and its modular inverse, 26 as the experimental values for key generation. In this setup, the second number chosen is n_2= -8, and its modular inverse is 16. These two inverse values are securely

exchanged with the recipient for decryption purposes. The steps involved in the key generation process for SKC are outlined as follows:

a. Choose a random integer, n_1=5.

b. The modular inverse of 5 is 26 (verification: 5 × 26 mod 43 = 1), so Key1=k_1 = 26.

c. Select a random negative integer, n_1 = -8.

d. The modular inverse of -8 is determined to be 16, So Key 2=k_2=16. The encryption procedure using SKC is illustrated in Table 2.

Table 2. Encryption procedure of SKC

Plain Text	Numeric Value	C=(α×n_1×n_2) mod 43	Cipher Text
S	19	14	N
$	31	7	G
T	20	17	Q
@	29	1	A
A	1	3	C
%	32	10	J
R	18	11	K

4.2. RSA Key Generation

In the RSA key generation experiment, we chose the initial prime number as p = 2 and the subsequent prime number as q = 19. Using these primes, we calculated m and λ(m). Additionally, a random value for 'i' was chosen, along with its corresponding modular inverse 'e', based on modulo λ(m). The RSA key generation steps are as follows:

- p =2 ; q = 19; thus, m = 38 and λ(m) = 18.
- Choose ‘i’ = 7, and its modular inverse, 'e', is 13 (verified by 7 × 13 mod 18 = 1).
- The public key is $(i, m) = (7, 38)$, and the private key is $(e, m) = (13, 38)$.

4.3. RSA ciphering

Referring to Table 2, we derive the encrypted message. "NGQACJK" which corresponds to the integer values: 14,7,17,1,3,10,11. The first column lists the ciphertext, the second column provides the corresponding integer values, the third column applies RSA encryption, and the fourth column shows the remainder output. The final alphanumeric values are assigned based on the computed results. The encryption process, using both the RSA public and private keys, is described by the formula (p^i mod m), and the final output is shown in Table 3.

Table 3. RSA encryption workflow

Ciphered Text	numerical value	RSA encoding	Remainder Output	corresponding alphanumeric
N	14	$(14)^7 mod 38$	22	V
G	7	$(7)^7 mod 38$	7	G
Q	17	$(17)^7 mod 38$	5	E
A	1	$(1)^7 mod 38$	1	A
C	3	$(3)^7 mod 38$	21	U
J	10	$(10)^7 mod 38$	34	&
K	11	$(11)^7 mod 38$	11	K

4.4. Graph Encryption

M= VGEAU&K represent the message that needs to be encoded for secure transmission

Step1 In this example, $n = [(42/7) + 7]$, So $n = 13$. Take a set of first 13 primes. As,

P_{13} = {2,3,5,7,11,13,17,19,23,29,31,37,41}.

Step 2 Figure 1 displays a table. To construct this table, the dimensions are set as (n − k) × k, where the values are 6 × 7.

Table- 4 (figure 1)

	2	3	5	7	11	13	17
19	A	B	C	D	E	F	G
23	H	I	J	K	L	M	N
29	O	P	Q	R	S	T	U
31	V	W	X	Y	Z	space	!
37	@	#	$	%	^	&	*
41	(	)	=	+	?	-	:

Step3 Message M= VGEAU&K become $V = 231, G = 1719, E = 1119, A = 219, U = 1729, \& = 1337, K = 723$.

i.e. encrypted message $M *= 231\ 1719\ 1119\ 219\ 1729\ 1337\ 723$

Step4 Next, we create a path graph by assigning labels to the vertices, as illustrated in Figure 2.

Figure 2

Vertex set = (3,7,11,13,17,19,23,29,31,37)

Here, $V1 =$ {2,7,11,13,17} $and\ V2 =$ {19,23,29,31,37}

The set of vertices in the bipartite graph becomes.

Edge set = $G(V1, V2) = \{(2,31), (17,19), (11,19), (2,19), (17,29), (13,37), (7,23)\}$

The diagram is created as shown in Figure 3. Following this, random weights are assigned to the neighbouring edges of the bipartite graph in Figure 3, as illustrated in Figure 4. The labeled diagram in Figure 4 is then transmitted to the receiving authority.

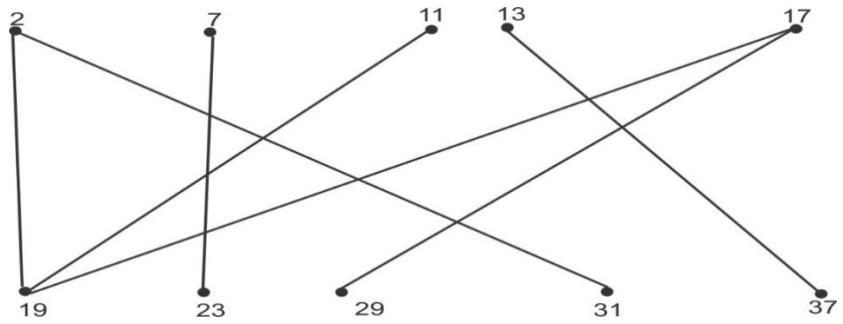

Figure 3

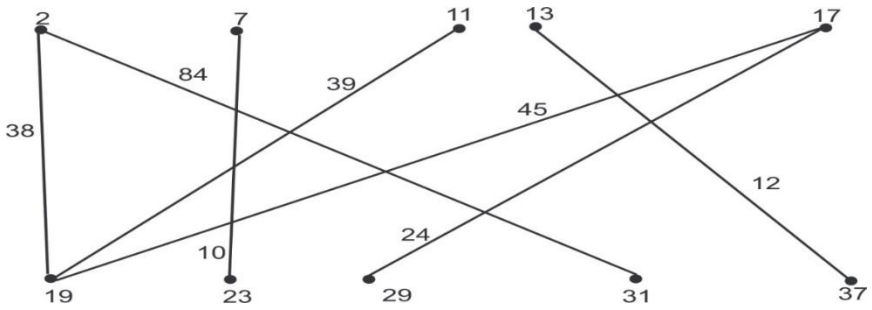

Figure 4

To decrypt, follow these steps:

Step 1: Begin by arranging the edge weights in descending order: W=84, 45, 39, 38, 24, 12, 10.

Step 2: Next, order the edges as follows: {(2,31), (17,19), (11,19), (2,19), (17,29), (13,37), (7,23)}. The related path diagram is presented in Figure 2.

Step 4: Finally, using the values derived from the table 4, we recover the alphabet VGEAU & K.

This process demonstrates how the encrypted graph is decrypted step-by-step, translating the encoded values back into the original message.

Decryption process of RSA and SKC

The decryption process for both RSA and SSK is demonstrated through the graph-driven decryption procedure. To recover the original message, we use both the SSK and RSA methods, as detailed in Tables 5 and 6.

In the initial stage of the procedure, Table 5 shows the steps for RSA decryption. The first column lists the ciphertext, followed by the corresponding integer values in the second column. The third column displays the RSA encryption results, while the fourth column displays the remainder values. In the fifth column, the resulting value is mapped to an alphanumeric character.

Next, in the second stage of decryption, Table 6 presents the application of the SSK algorithm. The first column in Table 6 contains the ciphertext from the previous stage. The second column shows the associated numerical values, followed by the SKC decryption procedure in the third column. The final column reveals the original message or plaintext.

The entire decryption procedure for this hybrid scheme, combining RSA, SKC and graph, is outlined across Tables 5 and 6. It utilizes the private key, represented as $c^e mod\ m$, and the inverse of n_1 and n_2 , which are referred to as k_1 and k_2 in the process. This approach illustrates how both cryptographic techniques work together to decrypt the message securely.

Table -5 Decryption procedure of RSA

Cipher Text	**Integer Value**	**RSA Decryption**	**Remainder value**	**Equivalent alphanumeric**
V	22	$(22)^{13} mod 38$	14	N
G	7	$(7)^{13} mod 38$	7	G
E	5	$(5)^{13} mod 38$	17	Q
A	1	$(1)^{13} mod 38$	1	A
U	21	$(21)^{13} mod 38$	3	C
&	34	$(34)^{13} mod 38$	10	J
K	11	$(11)^{13} mod 38$	11	K

Table -6 Decryption procedure of SKC

Ciphered Message	**Numerical Value**	**P=(β×k_1×k_2)mod43**	**Original Message**
N	14	19	S
G	7	31	$
Q	17	20	T
A	1	29	@
C	3	1	A
J	10	32	%
K	11	18	R

These three cryptographic methods work together in harmony to decrypt the encrypted graph. The graph-based decryption plays a crucial role in unravelling the encoded data, step by step, and translating it back into the original message. This hybrid approach showcases the power of combining RSA, SKC, and graph encryption to create a secure and efficient decryption process that ensures the integrity and confidentiality of the message.

Conclusion:

This research introduces a novel hybrid cryptosystem that combines RSA, symmetric key encryption, and bipartite graph-based encryption to enhance both security and performance. By leveraging the strengths of RSA for robust security and symmetric key algorithms for faster processing, this hybrid approach ensures efficient data encryption while maintaining high security. The addition of bipartite graph encryption further strengthens the system, providing an extra layer of protection by increasing the challenge for attackers to tamper with or forge encrypted data. Through experimental results, we demonstrate that this system offers a reasonable encryption and decryption time, outperforming traditional methods like 3DES and AES in both security and efficiency. Overall, this hybrid scheme offers flexibility and higher security without compromising on performance, making it a promising solution for modern cryptographic needs. By combining RSA, SSK, and graph encryption, our method addresses key management challenges and optimizes processing speed. This approach provides a secure, efficient, and scalable solution for encrypting data in various real-world applications, ensuring that both data protection and performance are prioritized.

References

[1]. Razaq, H. Alolaiyan, M. Ahmad et al., “A novel method for generation of strong substitution-boxes based on coset graphs and symmetric groups,” IEEE Access, vol. 8, pp. 75473–75490, 2020.

[2]. A. Razaq, M. Awais Yousaf, U. Shuaib, N. Siddiqui, A. Ullah, and A. Waheed, “A novel construction of substitution box involving coset diagram and a bijective map,” Security and Communication Networks, vol. 2017, p. 16, Article ID 5101934, 2017.

[3]. Shoukat, K. A. Bakar, and S. Ibrahim, “A Generic Hybrid Encryption System (HES),” Research Journal of Applied Sciences, Engineering and Technology, vol. 5, no. 9, 2013, doi: 10.19026/rjaset.5.4793.

[4]. A. Singh, M. Marwaha, B. Singh, and S. Singh, “Comparative Study of DES, 3DES, AES and RSA,” International journal of computers and technology, vol. 9, no. 3, Jul. 2013, doi: 10.24297/ijct.v9i3.3342.

[5]. A. Taha, K. Hosny, and D.-D. Salama, “An improved security schema for mobile cloud computing using hybrid cryptographic algorithms,” Far East Journal of Electronics and Communications, vol. 18, no. 4, pp. 521–546, Apr. 2018, doi: 10.17654/EC018040521.

[6]. B. R. Arunkumar, “Applications of Bipartite Graph in diverse fields including cloud computing,” International Journal of Modern Engineering Research, vol. 5, no. 7, p. 7, 2015. D. R. Stinson, Cryptography: Theory and Practice, Chapman and Hall/CRC, Boca Raton, FL, USA, 4th edition, 2018.

[7]. D. Sinha and A. Sethi, “Encryption using network and matrices through signed graphs,” International Journal of Computer Applications (0975-8887), vol. 138, no. 4, pp. 6–13, 2016.

[8]. F. Giacon, E. Kiltz, B. Poettering, "Hybrid Encryption in a Multi-user Setting," Springer, vol. 10769, pp. 159-189, 2018, doi: 10.1007/978-3-319-76578-5_6.

[9]. F. Yao and J. Su, “Hybrid Encryption Scheme for Hospital Financial Data Based on Noekeon Algorithm,” Sec. and Commun. Netw., vol. 2021, Jan. 2021, doi: 10.1155/2021/7578752.

[10]. G. A. Selim, “How to encrypt a graph,” International Journal of Parallel, Emergent and Distributed Systems, vol. 35, no. 6, pp. 668–681, 2020.

[11]. G. Viswanath, P. V. Krishna, "Hybrid encryption framework for securing big data storage in multi-cloud environment," Evolution Intelligent, vol. 14, pp. 691–698, 2021, doi: 10.1007/s12065-020-00404-w.

[12]. I. A. Shoukat, K. A. Bakar, and S. Ibrahim, “A Generic Hybrid Encryption System (HES),” Research Journal of Applied Sciences, Engineering and Technology, vol. 5, no. 9, 2013, doi: 10.19026/rjaset.5.4793.

[13]. J. F. W. Herschel and S. F. O. LaCroix, "A Collection of Examples of the Applications of the Calculus of Finite Differences," UK: Nabu Press, 2011.

[14]. J. Hu, J. Liang, and S. Dong, “A bipartite graph propagation approach for mobile advertising fraud detection,” Mobile Information Systems, vol. 2017, p. 12, Article ID 6412521, 2017.

[15]. K. R. Sajay, S. S. Babu, and Y. Vijayalakshmi, “Enhancing the security of cloud data using hybrid encryption algorithm,” J Ambient Intell Human Comput, Jul. 2019, doi: 10.1007/s12652-019-01403-1.

[16]. K. R. Nataraj and M. N. Praphul, “FPGA Implementation of Hybrid Cryptosystem,” International Journal of Emerging Science and Engineering (IJESE), vol. 1, no. 8, pp. 14–19, 2013.

[17]. M. Hamdi, J. Miri, and B. Moalla, “Hybrid encryption algorithm (HEA) based on chaotic system,” Soft Comput, vol. 25, no. 3, pp. 1847–1858, Feb. 2021, doi: 10.1007/s00500-020-05258-z.

[18]. M. Sujithra, G. Padmavathi, and S. Narayanan, “Mobile Device Data Security: A Cryptographic Approach by Outsourcing Mobile Data to Cloud,” Procedia Computer Science, vol. 47, pp. 480–485, Jan. 2015, doi: 10.1016/j.procs.2015.03.232.

[19]. M. Yamuna and A. Elakkiya, “Data transfer using fundamental circuits,” International Journal of Computer and Modern Technology, vol. 2, no. 01, 2015.

[20]. M. Yamuna and K. Karthika, “Data transfer using bipartite graphs,” International Journal of Advance Research in Science and Engineering, vol. 4, no. 02, pp. 128–131, 2015.

[21]. O. P. Akomolafe and M. O. Abodunrin, “A Hybrid Cryptographic Model for Data Storage in Mobile Cloud Computing,” International Journal of Computer Network and Information Security (IJCNIS), vol. 9, no. 6, pp. 53–60, Jun. 2017, doi: 10.5815/ijcnis.2017.06.06.

[22]. P. Kuppuswamy and S. Q. Y. Al-Khalidi, “Implementation of security through simple symmetric key algorithm based on modulo 37,” International Journal of Computers and Technology, vol. 3, no. 2, pp. 335–338, Oct. 2012, doi: 10.24297/ijct.v3i2c.2896.

[23]. P. Kedia and S. Agrawal, “Encryption using Venn-diagrams and graph,” International Journal of Advanced Computer Technology, vol. 4, no. 01, pp. 94–99, 2015.

[24]. P. L. K. Priyadarsini, “A survey on some applications of graph theory in cryptography,” Journal of Discrete Mathematical Sciences and Cryptography, vol. 18, no. 3, pp. 209–217, 2015.

[25]. R. Frucht and F. Harary, “On the corona of two graphs,” Aequationes Math, vol. 4, pp. 322–325, 1970.

[26]. R. L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” Communications of the ACM, vol. 21, no. 2, pp. 120–126, 1978.

[27]. R. Selvakumar and N. Gupta, “Fundamental circuits and cutsets used in cryptography,” Journal of Discrete Mathematical Sciences and Cryptography, vol. 15, no. 4-5, pp. 287–301, 2012.

[28]. S. Bojjagani, V. N. Sastry, C. M. Chen, S Kumari, M. K. Khan, "Systematic survey of mobile payments, protocols, and security infrastructure," Journal of Ambient Intelligence and Humanized Computing, 2021, doi: 10.1007/s12652-021-03316-4.

[29]. S. Manna, M. Prajapati, A. Sett, K. Banerjee, and S. Dutta, “Design and implementation of a two-layered hybrid cryptosystem,” in 2017 Third International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), Nov. 2017, pp. 327–331, doi: 10.1109/ICRCICN.2017.8234529.

[30]. S. Tariq, M. Khan, A. Alghafis, and M. Amin, “A novel hybrid encryption scheme based on chaotic Lorenz system and logarithmic key generation,” Multimed Tools Appl, vol. 79, no. 31, pp. 23507–23529, Aug. 2020, doi: 10.1007/s11042-020-09134-8.

[31]. T. S. Ali and R. Ali, “A Novel Medical Image Signcryption Scheme Using TLTS and Henon Chaotic Map,” IEEE Access, vol. 8, pp. 71974–71992, 2020, doi: 10.1109/ACCESS.2020.2987615.

[32]. Ü. Çavuşoğlu, S. Arslan, and S. Kurt, “A hybrid encryption approach based on symmetric encryption algorithm and chaotic encryption scheme,” 2018 International Conference on Computer Science and Engineering (UBMK), 2018, pp. 346–350, doi: 10.1109/UBMK.2018.8559733.

[33]. W. Gao, C. Lai, and Z. Shen, “A Hybrid Encryption Scheme Using RSA and Advanced Encryption Standard for Privacy-Preserving in the Cloud,” Future Internet, vol. 12, no. 7, pp. 120–130, Jun. 2020, doi: 10.3390/fi12070120.

Cite this Article:

*Surbhi Soni, Dr. Binny Gupta, Hemant Gena, Graph-Based Unified Cryptographic system for Secure Communication and Financial Transactions", **Pi International Journal of Mathematical Sciences**, **ISSN:** 3107-9830 (Online), Volume 1, Issue 2, pp. 31-43, October 2025.*

Journal URL: *https://pijms.com/*

DOI: *https://doi.org/10.59828/pijms.v1i2.8*