10.64837/GSA.13.5.4

Global and Stochastic Analysis
Vol. 13 No. 5 (2026)
Received: 22nd June 2026 Revised: 21st August 2026 Accepted: 11th September 2026

AN ALGORITHMIC APPROACH TO SECRET SHARING USING UNIT RINGS AND HYPERGRAPH-BASED ACCESS STRUCTURES

BHAVYA K, VASUDEVA, AND RAGHUNANDAN K R

Abstract. This paper proposes a hybrid encryption and decryption scheme that combines Unit-Sum Hypergraphs over finite rings with the RSA cryptosystem. The Unit-Sum Hypergraph is used to define an access structure in which a set of participants is authorized when the corresponding set of vertices forms a hyperedge. A random session seed is generated for each encryption session and protected using RSA. The recovered session seed, together with the neighborhood structure and vertex degrees of the authorized vertices, is used to generate participant-dependent encryption keys. These keys are applied to the plaintext to produce the ciphertext, while a verification tag is generated to detect inconsistencies during decryption. The proposed scheme combines RSA-based session-seed protection with the mathematical structure of Unit-Sum Hypergraphs to provide an access-structure-based encryption mechanism. Experimental evaluation is performed for different participant sizes and message sizes to study the computational performance of the proposed scheme. The results demonstrate successful encryption and decryption and show that the scheme remains computationally feasible as the participant population and message size increase. The proposed framework provides a foundation for additional mathematical constraints into the access structure for future development of multi-level authorization mechanisms.

1. Introduction

Cryptography has played a vital role in secure communication since ancient times, when military officers and diplomats relied on secret codes to protect confidential information. In the modern digital era, the rapid growth of the Internet, cloud computing, mobile communication, and online transactions has significantly increased the demand for robust information security. As communication technologies continue to evolve, so do the techniques employed by adversaries to compromise sensitive information, making the development of advanced cryptographic methods increasingly important.

Cryptography is the science of transforming plaintext into an unintelligible form, known as ciphertext, so that only authorized users possessing the appropriate key can recover the original message [1, 10]. Various mathematical concepts have been employed in cryptography [2, 3, 4]. In recent years, graph theory has emerged as a mathematical framework for designing cryptographic systems. Thamizisai and

2020 *Mathematics Subject Classification.* Primary 05C25; Secondary 05C30, 16P10, 94A62.
Key words and phrases. Secret sharing, Unit-Sum Hypergraph, Access structure, Encryption, Decryption,Algorithm.

Corresponding author: Vasudeva Acharya (Email address: vasudeva.acharya@nitte.edu.in)

Mary Jeya Jothi [11] proposed a graph-labeling-based encryption and decryption technique using Maximization of Even Multiplicative Labeling (MEML) for secure communication. Samuvel et al. [12] proposed the use of the corona product of graphs to assign numerical values to vertices based on Pythagorean numerology. Baizhu et al. [13] proposed an encryption algorithm based on corona graphs and bipartite graphs. Gomathi et al. [14] studied an ASCII-based data encryption system that uses a coloring-based tripartite graph as a key for secure data exchange. Obaidullah Wardak et al. [15] investigated the use of Cayley graphs for graph-based encryption and decryption. Ranasinghe [16] used snake vertex labeling for symmetric encryption of plaintext. Manal Hatif Kadhim [17] proposed a triple vertex cycle graph and an encryption scheme based on the proposed graph structure.

Secret sharing (SS) is a fundamental cryptographic technique in which a dealer distributes a secret among a set of participants such that only authorized subsets of participants are able to reconstruct the secret, while unauthorized subsets cannot reconstruct it. Since its introduction through the threshold-based constructions of Shamir [5] and Blakley [6], secret sharing has become an important technique in secure multiparty computation, distributed cryptography, and access control. Subsequently, the theory was extended beyond threshold schemes to general access structures. The works of Benaloh and Leichter [7], Ito et al. [9], and Brickell and Davenport [8] contributed to the development of secret-sharing schemes for general access structures.

Today, cryptography addresses a wide range of security requirements, including confidentiality, integrity, authentication, digital signatures, identity verification, and secure electronic transactions. However, many graph-based cryptographic approaches employ predefined graph structures and relatively static key-generation mechanisms, which may limit key diversity and adaptability. In particular, the potential of Unit-Sum Hypergraphs over finite rings for defining access structures and generating session-dependent encryption keys has received limited attention.

This paper proposes a novel **Hypergraph-Based Encryption Scheme** using Unit-Sum Hypergraphs over finite rings. A random session seed is protected using RSA, and participant-dependent encryption keys are subsequently generated from the neighborhood structure of the Unit-Sum Hypergraph. The proposed approach combines RSA-based public-key cryptography with hypergraph-based key generation to provide session-dependent encryption keys and an access-structure-based encryption mechanism. The computational performance of the proposed scheme is evaluated with respect to participant size and message size.

2. Preliminaries

This section provides the fundamental mathematical concepts,cryptographic techniques required for the proposed scheme, including Unit-Sum Hypergraphs, access structures and RSA cryptosystem.

2.1. Unit-Sum Hypergraph.

Definition 2.1. Let $U(\mathbb{Z}_n)$ denote the set of units of $\mathbb{Z}_n$. The *Unit-Sum Hypergraph* on $\mathbb{Z}_n$ is the hypergraph

$$H_U(\mathbb{Z}_n) = (V, E),$$

where

$$V = \mathbb{Z}_n$$

and

$$E = \{A \subseteq \mathbb{Z}_n : \sigma(A) \in U(\mathbb{Z}_n)\}. \tag{2.1}$$

For every subset $A \subseteq \mathbb{Z}_n$, the subset sum modulo n is defined by

$$\sigma(A) = \left(\sum_{x \in A} x\right) \bmod n. \tag{2.2}$$

The elements of E are called the *hyperedges* of $H_U(\mathbb{Z}_n)$.

Definition 2.2. [18] For $x \in V$, the neighborhood of x in the associated Unit-Sum graph G_U is defined as

$$N_G(x) = \{z \in V : x + z \in U(\mathbb{Z}_n)\}. \tag{2.3}$$

The degree of a vertex x is defined by

$$d_G(x) = |N_G(x)|. \tag{2.4}$$

2.2. Access Structure. Let

$$P = \{P_0, P_1, \ldots, P_{n-1}\}$$

be the set of participants, where participant P_i corresponds to the vertex $i \in \mathbb{Z}_n$. Define the correspondence mapping

$$\phi : P \to V$$

by

$$\phi(P_i) = i.$$

Definition 2.3. The *access structure* induced by the Unit-Sum Hypergraph

$$H_U(\mathbb{Z}_n)$$

is the family

$$\Gamma = \left\{A \subseteq P : \left(\sum_{P_i \in A} i\right) \bmod n \in U(\mathbb{Z}_n)\right\}. \tag{2.5}$$

Each set $A \in \Gamma$ is called an *authorized set* and is permitted to access the secret.

Theorem 2.4. *A set of participants is authorized to access the secret if and only if its corresponding set of vertices is a hyperedge of the Unit-Sum Hypergraph $H_U(\mathbb{Z}_n)$.*

Proof 1. Let $A \subseteq P$ be a set of participants and let

$$B = \phi(A) \subseteq V$$

be the corresponding set of vertices.

Suppose first that $A \in \Gamma$. By the definition of the access structure,

$$\left(\sum_{P_i \in A} i \right) \bmod n \in U(\mathbb{Z}_n).$$

Since $B = \phi(A)$ contains exactly the vertex labels corresponding to the participants in A, we have

$$\left(\sum_{x \in B} x \right) \bmod n \in U(\mathbb{Z}_n).$$

Hence, by the definition of the Unit-Sum Hypergraph,

$$B \in E.$$

Conversely, suppose that $B = \phi(A)$ is a hyperedge of $H_U(\mathbb{Z}_n)$. Then, by the definition of E,

$$\left(\sum_{x \in B} x \right) \bmod n \in U(\mathbb{Z}_n).$$

Since B is the corresponding vertex set of A, it follows that

$$\left(\sum_{P_i \in A} i \right) \bmod n \in U(\mathbb{Z}_n).$$

Therefore, by the definition of Γ,

$$A \in \Gamma.$$

Thus,

$$\boxed{A \in \Gamma \iff \phi(A) \in E}.$$

Hence, a set of participants is authorized if and only if its corresponding set of vertices is a hyperedge of the Unit-Sum Hypergraph $H_U(\mathbb{Z}_n)$.

Any subset in Γ^c cannot reconstruct the secret and obtains no information about it.

Example 2.5. Consider the ring $\mathbb{Z}_5 = \{0, 1, 2, 3, 4\}$. Since 5 is prime,

$$U(\mathbb{Z}_5) = \{1, 2, 3, 4\}.$$

Let

$$P = \{P_0, P_1, P_2, P_3, P_4\}$$

be the set of participants, where participant P_i corresponds to the vertex $i \in \mathbb{Z}_5$.

A subset of participants is authorized if the modular sum of the corresponding vertices is a unit element modulo 5.

Some hyperedges of the Unit-Sum Hypergraph are

$$\begin{aligned} e_1 &= \{0,1\}, && 0+1 \equiv 1 \pmod 5, \\ e_2 &= \{0,1,2,3\}, && 0+1+2+3 \equiv 1 \pmod 5, \\ e_3 &= \{1,2,3\}, && 1+2+3 \equiv 1 \pmod 5. \end{aligned}$$

Hence, the corresponding authorized sets of participants are

$$\Gamma \supseteq \{\{P_0, P_1\}, \{P_0, P_1, P_2, P_3\}, \{P_1, P_2, P_3\}\}.$$

Now consider the subsets

$$\{1,4\}, \qquad \{1,2,3,4\}.$$

Their modular sums are

$$1+4 \equiv 0 \pmod 5, \qquad 1+2+3+4 \equiv 0 \pmod 5.$$

Since

$$0 \notin U(\mathbb{Z}_5),$$

these subsets are not hyperedges of $H_U(\mathbb{Z}_5)$. Therefore, the corresponding participant sets

$$\{P_1, P_4\}, \qquad \{P_1, P_2, P_3, P_4\}$$

are unauthorized sets and belong to Γ^c.

Thus, every hyperedge of the Unit-Sum Hypergraph corresponds to an authorized set of participants, whereas every non-hyperedge corresponds to an unauthorized set.

2.3. RSA Cryptosystem. The security of RSA [10] is based on the computational difficulty of factoring large composite integers.

In the proposed Hypergraph-Based Encryption Scheme, RSA is utilized to securely establish a random seed between the communicating parties. This random seed is then used to derive dynamic encryption keys from the Unit Sum Hypergraph structure, thereby combining the security of public-key cryptography with the efficiency of hypergraph-based symmetric encryption.

3. Proposed Methods

3.1. RSA-Based Hypergraph Encryption-Decryption Scheme. In this section, we propose a hypergraph-based encryption scheme using an RSA-generated random session seed and a Unit-Sum Hypergraph over the finite ring $\mathbb{Z}_n$. Initially, a random session seed is generated and securely transmitted using RSA. The recovered session seed is subsequently used to derive participant-dependent encryption keys from the Unit-Sum Hypergraph.

The corresponding decryption algorithm recovers the shared random session seed using the RSA private key and reconstructs the same participant-dependent keys from the Unit-Sum Hypergraph. These keys are then used to recover the original plaintext from the ciphertext. The complete encryption and decryption procedures are described in the following subsections.

3.1.1. *Encryption.* The encryption procedure consists of the following steps.

1) **Plaintext Encoding**
   The plaintext message is converted into its UTF-8 byte representation, resulting in the integer sequence
   $$M = (m_1, m_2, \ldots, m_L),$$
   where L denotes the plaintext length in bytes.
2) **Hypergraph Construction**
   Let n denote the order of the Unit-Sum Hypergraph. Construct the Unit-Sum Hypergraph
   $$H_U(\mathbb{Z}_n) = (V, E),$$
   where the hyperedge set E is defined in Equation (2.1).
   For UTF-8 byte-oriented encryption, the modulus is chosen such that
   $$n \geq 256.$$
3) **Access Structure**
   Select an authorized participant set from the access structure defined in Equation (2.5). Let
   $$A = \{v_1, v_2, \ldots, v_s\} \in \Gamma,$$
   where $s = |A|$ denotes the number of authorized participants.
4) **Random Session Seed Generation using RSA**
   (a) The receiver generates an RSA key pair by selecting two large prime numbers p and q.
   (b) Compute
   $$N = pq, \qquad \phi(N) = (p-1)(q-1).$$
   (c) Select a public exponent e satisfying
   $$\gcd(e, \phi(N)) = 1.$$
   (d) Compute the private exponent
   $$d \equiv e^{-1} \pmod{\phi(N)}.$$
   (e) The sender selects a random session seed
   $$r \in \mathbb{Z}_N.$$
   (f) Encrypt the session seed using the receiver's public key:
   $$R = r^e \pmod{N}.$$
   (g) The encrypted session seed R is transmitted together with the ciphertext.
5) **Hypergraph Key Generation**
   For every authorized vertex $v_i \in A$,
   (a) Compute the neighborhood
   $$N_G(v_i) = \{u \in V : (u + v_i) \bmod n \in U(\mathbb{Z}_n)\}.$$

(b) Compute the neighborhood sum

$$w(v_i) = \left(\sum_{x \in N_G(v_i)} x \right) \bmod n. \tag{3.1}$$

(c) Compute the vertex degree

$$d_G(v_i) = |N_G(v_i)|.$$

(d) Generate the participant-dependent encryption key

$$k_i = (r + w(v_i) + d_G(v_i)) \bmod n. \tag{3.2}$$

6) **Key Sequence Generation**

Using the s participant keys, generate a cyclic key sequence of length L as

$$\kappa_j = k_{((j-1) \bmod s)+1}, \qquad j = 1, 2, \ldots, L. \tag{3.3}$$

Thus, the participant keys are repeated cyclically until all plaintext symbols are assigned a key.

7) **Block Formation**

The plaintext sequence is divided into consecutive blocks of size s only for organizational purposes. If the final block contains fewer than s symbols, the cyclic key sequence continues without interruption. Hence, the encryption operation is applied uniformly to all L plaintext symbols.

8) **Block Encryption**

For each plaintext symbol m_j, compute

$$c_j = (m_j + \kappa_j) \bmod n, \qquad j = 1, 2, \ldots, L. \tag{3.4}$$

The complete ciphertext sequence is

$$C = (c_1, c_2, \ldots, c_L).$$

9) **Authentication Tag**

Compute the verification tag

$$T = \left(\sum_{j=1}^{L} c_j + \sum_{j=1}^{L} \kappa_j \right) \bmod n. \tag{3.5}$$

10) **Ciphertext Transmission**

The sender transmits

$$(R, C, T)$$

to the receiver.

3.1.2. *Decryption.* The decryption procedure is carried out as follows.

1) **Recover the Random Session Seed**
   Using the RSA private key (d, N), compute
$$r = R^d \pmod{N}.$$
2) **Construct the Hypergraph**
   Construct the same Unit-Sum Hypergraph $H_U(\mathbb{Z}_n)$ as defined in the hypergraph construction step of the encryption procedure.
3) **Verify the Authorized Participant Set**
   Verify that the supplied participant set
$$A = \{v_1, v_2, \ldots, v_s\}$$
   belongs to the access structure Γ. If
$$A \notin \Gamma,$$
   the decryption procedure is rejected.
4) **Regenerate the Hypergraph Keys**
   For every authorized vertex $v_i \in A$, compute the neighborhood
$$N_G(v_i) = \{u \in V : (u + v_i) \bmod n \in U(\mathbb{Z}_n)\}.$$
   The corresponding neighborhood sum is computed as
$$w(v_i) = \left(\sum_{x \in N_G(v_i)} x \right) \bmod n.$$
   The degree of v_i is
$$d_G(v_i) = |N_G(v_i)|.$$
   Using the recovered session seed r, regenerate the participant-dependent key
$$k_i = (r + w(v_i) + d_G(v_i)) \bmod n. \tag{3.6}$$
5) **Generate the Key Sequence**
   Using the regenerated participant keys, construct the same cyclic key sequence used during encryption:
$$\kappa_j = k_{((j-1) \bmod s)+1}, \qquad j = 1, 2, \ldots, L. \tag{3.7}$$
6) **Recover the Plaintext**
   For each ciphertext symbol c_j, recover the corresponding plaintext symbol using
$$m_j = (c_j - \kappa_j) \bmod n, \qquad j = 1, 2, \ldots, L. \tag{3.8}$$
   The recovered plaintext sequence is
$$M = (m_1, m_2, \ldots, m_L).$$

7) **Verify the Authentication Tag**
Compute the verification value

$$T' = \left(\sum_{j=1}^{L} c_j + \sum_{j=1}^{L} \kappa_j \right) \bmod n. \tag{3.9}$$

If

$$T' = T,$$

the recovered plaintext is accepted; otherwise, the decryption procedure returns *Reject*.

3.2. Algorithm for the above Proposed Mechanism.

Algorithm 1 RSA-based Unit-Sum Hypergraph Encryption

Input:

- Plaintext message $M = (m_1, m_2, \ldots, m_t)$ of length L;
- Number of participants L, where $n \leq L$;
- RSA public key (e, N).

Output: (R, C, T),

where R is the RSA-encrypted random value, C is the ciphertext, and T is the authentication tag.

1: Convert the plaintext into its integer representation

$$M = (m_1, m_2, \ldots, m_n).$$

2: Let

$$\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$$

be the set of vertices corresponding to the n participants.

3: Compute the set of units

$$U(\mathbb{Z}_n) = \{x \in \mathbb{Z}_n : \gcd(x, n) = 1\}.$$

4: Construct the Unit-Sum Hypergraph

$$H_U(\mathbb{Z}_n) = (V, E),$$

where,

$$V = \mathbb{Z}_n, E = \left\{ A \subseteq V : \sum_{x \in A} x \pmod n \in U(\mathbb{Z}_n) \right\}.$$

5: Define the access structure as

$$\Gamma = E_\Gamma,$$

where $E_\Gamma = \left\{ A \subseteq V : \sum_{x \in A} x \pmod n = y \in U(\mathbb{Z}_n) \right\}$

6: Select an authorized hyperedge

$$A = \{v_1, v_2, \ldots, v_s\} \in \Gamma.$$

7: **for** each $v_i \in A$ **do**

8: Compute the neighbourhood

$$N(v_i) = \{u \in V : \ u + v_i \in U(\mathbb{Z}_n)\}.$$

Compute the neighbourhood sum

$$w(v_i) = \sum_{x \in N(v_i)} x \pmod{n}.$$

Compute the vertex degree

$$d(v_i) = |N(v_i)|.$$

For a random integer $r \in \mathbb{Z}_N$. Encrypt using RSA: $R = r^e \pmod{N}$.
Generate the hypergraph key

$$k_i = (r + w(v_i) + d(v_i)) \pmod{n}.$$

end for
Define the key sequence for the selected authorized participants by

$$\kappa_j = k_{((j-1) \bmod s)+1}, \qquad j = 1, 2, \ldots, t.$$

for $j = 1, 2, \ldots, t$ **do**
Compute the ciphertext symbol

$$c_j = (m_j + \kappa_j) \pmod{n}.$$

end for
Compute the authentication tag

$$T = \left(\sum_{j=1}^{t} c_j + \sum_{j=1}^{t} \kappa_j \right) \pmod{n}.$$

Form the ciphertext

$$C = (c_1, c_2, \ldots, c_t).$$

return

$$(R, C, T).$$

Algorithm 2 RSA-based Unit-Sum Hypergraph Decryption

Input:

- Ciphertext $C = (c_1, c_2, \ldots, c_t)$;
- RSA-encrypted random value R;
- Authentication tag T;
- RSA private key (d, N);
- Authorized hyperedge

$$A = \{v_1, v_2, \ldots, v_s\} \in E_\Gamma.$$

Output:

$$M = (m_1, m_2, \ldots, m_t),$$

or REJECT if authentication fails.

Construct the Unit-Sum Hypergraph

$$H_U(\mathbb{Z}_n) = (V, E), \qquad V = \mathbb{Z}_n.$$

Verify that

$$A = \{v_1, v_2, \ldots, v_s\} \in E_\Gamma.$$

If $A \notin E_\Gamma$, return REJECT.

Only the participants corresponding to the vertices in A participate in the key-generation process.

Recover the random value r using the RSA private key:

$$r = R^d \pmod{N}.$$

for each $v_i \in A$ **do**

Compute the neighbourhood

$$N(v_i) = \{u \in V : \ u + v_i \in U(\mathbb{Z}_n)\}.$$

Compute the neighbourhood sum

$$w(v_i) = \sum_{x \in N(v_i)} x \pmod{n}.$$

Compute the vertex degree

$$d(v_i) = |N(v_i)|.$$

Regenerate the hypergraph key

$$k_i = (r + w(v_i) + d(v_i)) \pmod{n}.$$

end for

Generate the key sequence

$$\kappa_j = k_{((j-1) \bmod s)+1}, \qquad j = 1, 2, \ldots, t.$$

for $j = 1, 2, \ldots, t$ **do**

Recover the plaintext symbol

$$m_j = (c_j - \kappa_j) \pmod{n}.$$

end for

Verify the authentication tag:

$$T' = \left(\sum_{j=1}^{t} c_j + \sum_{j=1}^{t} \kappa_j \right) \pmod{n}.$$

if $T' = T$ **then**

Accept the recovered plaintext

$$M = (m_1, m_2, \ldots, m_t).$$

else

Return REJECT.

end if

return M.

Example 3.1. To demonstrate the operation of the proposed RSA-based Unit-Sum Hypergraph encryption scheme, an experimental example was performed using the plaintext message "Hello world". The authorized participant group was selected as $A = \{1, 2, 4\}$, and a 256-bit RSA key pair was generated using two CPU cores. The plaintext was converted into its UTF-8 integer representation before encryption.

For the selected authorized group, the sum of the participant indices is

$$1 + 2 + 4 = 7,$$

and since

$$\gcd(7, 256) = 1,$$

the selected group satisfies the authorization condition in $U(\mathbb{Z}_{256})$.

The encryption procedure generated the RSA-encrypted session value

$$R = 175481302989093397794560892704385427611 3$$
$$811175409596638619754357587024792849 4,$$

the ciphertext sequence

$$C = [192, 93, 100, 228, 103, 24, 239, 103, 106, 228, 92],$$

and the authentication tag

$$T = 140.$$

Thus, the transmitted ciphertext is represented by the tuple

$$(R, C, T).$$

At the receiver side, the RSA private key was used to recover the session seed. The same Unit-Sum Hypergraph and authorized participant group were then used to regenerate the participant-dependent keys. The ciphertext was subsequently decrypted according to

$$m_j = (c_j - \kappa_j) \pmod{256}.$$

The recovered plaintext was

$$\text{Decrypted Message} = \text{"Hello world"}.$$

The recovered message was identical to the original plaintext, and the authentication verification was successful. Hence,

$$\text{Decrypted Message} = \text{"Hello world"}, \qquad \text{Match Successful} = \text{True}.$$

This example demonstrates the successful execution of the proposed encryption, decryption, and authentication procedures for the selected authorized participant group.

4. Security Analysis

4.1. Participant-Size Scalability. To evaluate the scalability of the proposed scheme with respect to the number of participants, experiments were conducted for participant sizes $n \in \{64, 128, 256, 512, 1024, 2048, 4096\}$. A fixed 1 KB message and the authorized participant group $A = \{1, 2, 4\}$ were used throughout the experiment. The corresponding execution times are presented in Table 1.

The results indicate that the participant-dependent preprocessing cost increases as the number of participants grows. In particular, vertex-data computation increases from approximately 0.66 ms for $n = 64$ to 1.71 ms for $n = 4096$, making it the dominant hypergraph-related computational component for larger participant sets. This behavior is consistent with the theoretically derived $O(ng/P)$ complexity, where g denotes the number of operations associated with vertex-data computation and P represents the number of parallel workers. The unit-set construction time also increases with n, although its contribution to the overall execution time remains comparatively small.

TABLE 1. Execution time (s) for different participant sizes

n	Unit Set	Vertex Data	RSA Enc.	RSA Dec.	Total Enc.	Total Dec.
64	0.0039	0.6581	0.0444	0.1442	0.956 ± 0.420	0.933 ± 0.362
128	0.0071	0.7834	0.0055	0.1096	0.817 ± 0.322	1.033 ± 0.389
256	0.0141	0.7290	0.0056	0.1021	0.799 ± 0.214	0.936 ± 0.340
512	0.0387	0.9032	0.0055	0.1104	0.922 ± 0.270	1.032 ± 0.287
1024	0.1338	0.9560	0.0055	0.1017	1.012 ± 0.172	1.114 ± 0.147
2048	0.1963	1.1085	0.0055	0.1084	1.696 ± 0.846	1.937 ± 1.855
4096	0.4152	1.7146	0.0056	0.1053	2.265 ± 0.295	2.598 ± 1.651

4.2. Message-Size Scalability. The scalability of the proposed scheme with respect to message size was evaluated separately using $n = 256$. Message sizes ranging from 1 KB to 1 MB were considered, and 30 independent repetitions were performed for each message size. The mean execution time, standard deviation, and corresponding 95% confidence interval were calculated for both encryption and decryption.

The experimental results are summarized in Table 2. The encryption time increases from 0.923 ms for a 1 KB message to 106.370 ms for a 1 MB message. Similarly, the decryption time increases from 0.884 ms to 100.069 ms over the same range. The approximately linear growth in execution time with respect to message size indicates that message processing is the dominant size-dependent component. This observation provides empirical support for the theoretically derived $O(L)$ complexity, where L denotes the message length.

4.3. Component-Wise Complexity. To identify the sources of computational overhead, the total execution time was decomposed into unit-set construction, vertex-data computation, RSA encryption, RSA decryption, and the overall encryption and decryption times. The component-wise results are given in Table 3.

The decomposition shows that vertex-data computation is the major participant dependent component. Its execution time increases from 0.6581 ms at $n = 64$ to

TABLE 2. Execution time (ms) for different message sizes

	Encryption		Decryption	
Size	Mean ± SD	95% CI	Mean ± SD	95% CI
1 KB	0.923 ± 0.190	0.852–0.994	0.884 ± 0.182	0.816–0.952
4 KB	1.179 ± 0.900	0.843–1.515	1.132 ± 0.122	1.087–1.178
16 KB	2.212 ± 0.248	2.120–2.305	2.514 ± 0.670	2.264–2.765
64 KB	6.558 ± 0.408	6.406–6.710	7.616 ± 1.960	6.885–8.348
256 KB	25.370 ± 2.254	24.528–26.211	26.156 ± 4.616	24.433–27.880
1 MB	106.370 ± 7.630	103.521–109.219	100.069 ± 3.961	98.590–101.548

1.7146 ms at $n = 4096$. The unit-set construction time also increases with the participant size, from 0.0039 ms to 0.4152 ms. In contrast, the RSA encryption and decryption components remain nearly constant across different values of n. This stability is expected because the RSA parameters are fixed at 256 bits and therefore the cost of the modular exponentiation does not depend directly on the number of participants.

TABLE 3. Component-wise execution time (ms) for different participant sizes

n	Unit Set	Vertex Data	RSA Enc.	RSA Dec.	Total Enc.	Total Dec.
64	0.0039	0.6581	0.0444	0.1442	0.9563 ± 0.4195	0.9330 ± 0.3624
128	0.0071	0.7834	0.0055	0.1096	0.8170 ± 0.3224	1.0331 ± 0.3888
256	0.0141	0.7290	0.0056	0.1021	0.7985 ± 0.2142	0.9355 ± 0.3399
512	0.0387	0.9032	0.0055	0.1104	0.9219 ± 0.2704	1.0320 ± 0.2867
1024	0.1338	0.9560	0.0055	0.1017	1.0116 ± 0.1716	1.1140 ± 0.1470
2048	0.1963	1.1085	0.0055	0.1084	1.6956 ± 0.8457	1.9371 ± 1.8549
4096	0.4152	1.7146	0.0056	0.1053	2.2654 ± 0.2953	2.5980 ± 1.6512

4.4. Experimental Validation of Computational Complexity. To empirically validate the theoretical computational complexity of the proposed encryption and decryption algorithms, experiments were performed for participant sizes $n \in \{64, 128, 256, 512, 1024, 2048, 4096\}$. For each configuration, 30 independent repetitions were conducted using a fixed 1 KB message, authorized participant group $A = \{1, 2, 4\}$, 256-bit RSA parameters, and two parallel workers. The mean execution time, standard deviation, and 95% confidence interval were determined from the repeated measurements.

The experimental results confirm the expected dependence of the computational cost on the participant size. In particular, vertex-data computation represents the principal participant-dependent component and exhibits an increasing trend as n grows. This behavior is consistent with the theoretically obtained $O(ng/P)$ complexity under parallel execution. The unit-set construction also increases with n, in agreement with its theoretical $O(n \log n)$ complexity. On the other hand, the

RSA encryption and decryption costs remain approximately stable because the RSA key size is fixed.

A separate message-size experiment was conducted for $n = 256$, with message sizes ranging from 1 KB to 1 MB. The encryption time increased from 0.923 ms to 106.370 ms, while the decryption time increased from 0.884 ms to 100.069 ms. The observed increase is approximately proportional to the message size, supporting the theoretical $O(L)$ message-processing component, where L denotes the length of the input message.

Overall, the experimental results are consistent with the theoretical complexity analysis. The participant-size experiment demonstrates that the principal scalability cost arises from hypergraph-related vertex-data computation, whereas the message-size experiment confirms that the dominant cost for large messages is the linear message-processing component. These results demonstrate that the proposed scheme remains computationally feasible as both the participant population and message size increase.

5. Security Analysis

The security of the proposed scheme relies on the combination of RSA-based session-seed protection and dynamic key generation using the Unit Sum Hypergraph. A fresh random seed is generated for each encryption session, producing a distinct session key and thereby reducing the risk of key-reuse and known-plaintext attacks. Since the seed is securely exchanged using RSA, an adversary without the private key cannot easily recover it. Furthermore, the hypergraph-based key generation introduces structural complexity and improves key diversity, making brute-force and key-recovery attacks more difficult. Thus, the proposed scheme provides enhanced confidentiality through session-dependent dynamic encryption keys.

6. Conclusion

The main concept of this work is the use of a Unit-Sum Hypergraph to define an access structure for secure communication. A set of participants is authorized when the corresponding set of vertices forms a hyperedge of the Unit-Sum Hypergraph. Thus, the hypergraph provides a mathematical framework for distinguishing authorized participant sets from unauthorized sets. In this work, a hybrid encryption and decryption scheme based on Unit-Sum Hypergraphs and the RSA cryptosystem has been proposed. The proposed approach integrates the mathematical structure of the Unit-Sum Hypergraph with RSA-based protection of a randomly generated session seed.

The experimental results demonstrate the successful encryption and decryption of messages using the proposed scheme. An important direction for future work is to strengthen the access structure by introducing additional constraints in addition to the unit-sum condition. For example, an authorized set could be required to satisfy the hyperedge condition together with an additional number-theoretic or structural constraint, such as a prescribed coprimality condition or another property of the participant labels. In general, such a construction can be represented

as

$$\Gamma' = \{A \in \Gamma : C(A) = \text{true}\},$$

where $C(A)$ represents an additional constraint imposed on the authorized set. Multiple constraints could also be combined to construct a multi-level access structure. Such extensions may provide greater flexibility in defining authorization policies and increase the structural diversity of the resulting key-generation mechanism. The security implications of these additional constraints should be formally analyzed in future work.

Overall, the integration of Unit-Sum Hypergraph structures with RSA provides a promising hybrid framework for dynamic and access-structure-based encryption. The proposed construction demonstrates how algebraic and graph-theoretic structures can be combined with public-key cryptographic techniques to develop session-dependent encryption mechanisms for secure communication.

References

1. Rosen, K. H.: Elementary number theory, Pearson Education, London (2011).
2. Raghunandan, K. R., Shetty, R. and Aithal, G.: Key generation and security analysis of text cryptography using cubic power of Pell's equation, *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)* (2017) 1496–1500. DOI: 10.1109/ICICICT1.2017.8342791.
3. Aithal, G. and Shetty, S.: Image encryption scheme in public key cryptography based on cubic Pells quadratic case, *Indonesian Journal of Electrical Engineering and Computer Science* **20**(1) (2020) 385.
4. Raghunandan, K. R., Aithal, G., Shetty, S. and Khanna, B.: Key generation using generalized Pell's equation in public key cryptography based on the prime fake modulus principle to image encryption and its security analysis, *Cybernetics and Information Technologies* **20**(3) (2020) 86–101.
5. Shamir, A.: How to share a secret, *Communications of the ACM* **22**(11) (1979) 612–613.
6. Blakley, G. R.: Safeguarding cryptographic keys, *Proceedings of the National Computer Conference* **48** (1979) 313–317.
7. Benaloh, J. and Leichter, J.: Generalized secret sharing and monotone functions, *CRYPTO* (1990) 27–35.
8. Brickell, E. F. and Davenport, D. M.: On the classification of ideal secret sharing schemes, *Journal of Cryptology* **4**(2) (1991) 123–134.
9. Itô, M., Saito, A. and Nishizeki, T.: Secret sharing scheme realizing general access structure, *Electronics and Communications in Japan* **72**(9) (1989) 56–64.
10. Rivest, R., Shamir, A. and Adleman, L.: RSA (cryptosystem), *Arithmetic Algorithms and Applications* **19** (1978).
11. Jothi, R. M. J. and others: Encryption and decryption techniques in graph labeling: A novel approach to secure communication, *2025 IEEE 2nd International Conference for Women in Computing (InCoWoCo)* (2025) 1–5.
12. Samuvel, J. B., Srinitish, R. and Karthikeyan, B. C.: Encryption and decryption using corona product and pythagorean numerology in graph theory, *AIP Conference Proceedings* **3436**(1) (2026) 030004.
13. Ni, B., Qazi, R., Rehman, S. U. and Farid, G.: Some graph-based encryption schemes, *Journal of Mathematics* **2021**(1) (2021) 6614172.
14. Gomathi, D. and Nagarajan, S.: An ASCII value based data encryption using coloring tripartite graph, *Contemporary Mathematics* (2025) 2113–2130.
15. Wardak, O., Kumar, S. and Sinha, D.: Encryption and decryption scheme through signed Cayley graph, *Journal of Discrete Mathematical Sciences and Cryptography* **28**(7) (2025) 2811–2825.

16. Ranasinghe, P. G. R. S., Bandara, R. M. V. V. and Athapaththtu, A. M. S. L. K.: Symmetric encryption using snake graphs and supermagic covering, *Journal of the National Science Foundation of Sri Lanka* **52**(4) (2025).
17. Kadhim, M. H. and Ajeena, R. K. K.: Triple vertex of cycle graph for polyalphabetic encryption scheme, *Journal of Discrete Mathematical Sciences and Cryptography* **26** (2023) 1183–1188.
18. Ashrafi, N., Maimani, H. R., Pournaki, M. R. and Yassemi, S.: Unit graphs associated with rings, *Communications in Algebra* **38**(8) (2010) 2851–2871.

Department of Mathematics Nitte (Deemed to be University), NMAM Institute of Technology Nitte, Karkala, Karnataka, India – 574110
Email address: bhavya@nitte.edu.in

Department of Information Science Engineering Nitte (Deemed to be University), NMAM Institute of Technology Nitte,, Karkala, Karnataka, India – 574110
Email address: vasudeva.acharya@nitte.edu.in

Department of Computer Science Engineering Nitte (Deemed to be University), NMAM Institute of Technology Nitte,, Karkala, Karnataka, India – 574110
Email address: raghunandan@nitte.edu.in