JTAM (Jurnal Teori dan Aplikasi Matematika)
http://journal.ummat.ac.id/index.php/jtam
p-ISSN 2597-7512 | e-ISSN 2614-1175
Vol. 10, No. 1, January 2026, pp. 84-94

Application on Hypergraph in Vigenere Chiper

Okta Endri Asari[1], Dafik[1], Robiatul Adawiyah[1*], Arika Indah Kristiana[1], Rafiantika Megahnia Prihandini[1]
[1]Department of Mathematics Education, Jember University, Indonesia
robiatul@unej.ac.id

Article History:
Received : 04-09-2025
Revised : 13-11-2025
Accepted : 14-11-2025
Online : 01-01-2026

Keywords:
Hypergraph;
Symmetric Encryption;
Vigenere Chiper.

ABSTRACT

Message protection remains a major focus in the field of cryptography. This study proposes a new development on the Caesar cipher algorithm by utilizing hypergraph as a keystream generation source. The research designs a super (a,d)-hyperedge antimagic total labeling method applied to three hypergraph structures (Volcano, Semi Parachute, and Comb) to generate the keystream. Security is evaluated using four mechanisms: brute force analysis, processing time, ciphertext character distribution, and ciphertext bit size. The findings prove that the hypergraph based approach is robust against brute force attacks, improve memory and time efficiency. Quantitatively, the Comb hypergraph demonstrates the best efficiency, achieving an encryption time of 0.0030 seconds for 512 bytes and superior storage efficiency (e.g., 136 bytes for 16 bytes), outperforming the Semi Parachute and Volcano structures. The main contributions include the hypergraph labeling-based keystream generation algorithm, dynamic block key construction, and a Vigenere protocol that is more adaptive to storage constraints and computationally efficient..

https://doi.org/10.31764/jtam.v10i1.34572

This is an open access article under the CC–BY-SA license

A. INTRODUCTION

In the digital era, information protection has become a very urgent need along with the increasing volume of data exchanged over communication networks (Badawi et al., 2020) (Djumadin, 2023). Encryption, the process of converting original data (plaintext) into a coded form (ciphertext), is the most effective method used to secure this informations, which have been widely adopted in various sectors, including military, healthcare, and banking (Alemami et al., 2023). Encryption cannot be understood without permission or a specific key (Baagyere et al., 2020). The main purpose is to prevent unauthorized access to confidential data, and this entire method relies heavily on the structure and strength of the encryption key used (Ardiansyah et al., 2023).

In general, the encryption process consists of two main stages, namely the encryption process itself and the decryption process. In the encryption stage, the original data is converted into a form that is not directly meaningful (ciphertext), while in the decryption stage, the ciphertext is returned to its original plaintext form so that it can be read again (Ardiansyah et al., 2023; Pabokory et al., 2016). This method relies heavily on the structure and strength of the encryption key used. One critical approach in developing robust keys is the utilization of keystreams, which are sequences of keys generated systematically to perform sequential data

encryption. Keystreams must exhibit high levels of randomness and non-linearity to effectively counter modern cryptanalysis techniques. With their flexible structure and complex connectivity, hypergraphs offer a promising mathematical foundation for generating key variations that are inherently non-predictable (Ding et al., 2020).

While previous studies have successfully employed graph labeling to construct encryption key structures, the application of hypergraph labeling specifically in generating cryptographic keystreams remains largely unexplored, constituting a significant research gap (Dafik et al., 2025). Our novelty addresses this by leveraging the unique properties of hypergraphs. Specifically, the use of antimagic labelling on a hypergraph structure can induce a high degree of uniqueness and randomness in label distribution, resulting in a more secure keystream compared to traditional graph-based methods (Prihandoko et al., 2022).

In the world of cryptography, there are two basic approaches to encryption systems, namely symmetric encryption and asymmetric encryption. Symmetric encryption uses the same key for both encryption and decryption, while asymmetric encryption uses two different keys, a public key and a private key (Moosavi et al., 2017). Symmetric encryption is faster in terms of execution, but requires a secure way for key distribution. On the other hand, asymmetric encryption simplifies key distribution but has a higher computational burden.

This paper proposes a new method to generate keystream through hypergraph labelling, specifically super (a,d)-hyperedge antimagic total labelling applied to certain graph shapes such as volcano graph, semi-parachute graph, and comb product of graph. This labelling aims to create a unique value for each edge based on the combination of the labels of the vertices and the edge itself (NourEldeen et al., 2025). These values are then used as the key source in the vigenere cipher system.

The Vigenere cipher was chosen in this study for two main reasons. First, it is one of the classic methods in cryptography that is simple and easy to understand, yet still relevant in the modern security context (Ginting et al., 2017; Soofi et al., 2016). Second, vigenere cipher is still used in various security applications, such as digital data protection (Achmad et al., 2020), encrypted communication (Gautam et al., 2018; Saraswat et al., 2016), digital image security (Gerhana et al., 2016), and digital signature systems. Our proposed method aims to resolve this core vulnerability. We introduce a technique utilizing super (a,d)-hyperedge antimagic total labelling applied to three specific hypergraph structures (Volcano, Semi Parachute, and Comb). The resulting unique edge weights are systematically used as the dynamic, non-linear key source in the modified Vigenere encryption protocol. The main contribution of this research is the development of hypergraph labeling that not only extends graph labelling theory, but also provides a foundation for forming dynamic block encryption systems (Arat kotzer). This labelling allows the use of different keys for each block of text, thus strengthening the security of the system against frequency and repetitive pattern attacks. By utilizing the hypergraph structure and antimagic labelling, the resulting keystream has non-linear characteristics and is difficult to predict (Wang et al., 2023).

In addition, this research produced a complete algorithm for: (1) forming a hypergraph labeling-based keystream, (2) determining the initial key for each block in the ciphertext, and (3) designing an encryption protocol that modifies the Vigenere cipher algorithm. The protocol is evaluated in terms of security, memory efficiency, and flexibility in block length adjustment.

Simulation results show that the constructed encryption system has the advantage of producing unrecognizable ciphertexts, while being efficient in memory usage.

In general, this research strengthens the connection between advanced graph theory (hypergraph) and practical applications in modern cryptography (Meenakshi et al., 2025). With a robust combinatorial approach and systematic algorithmic implementation, this contribution is expected to open new opportunities in the development of encryption systems based on mathematical structure labelling (Ali et al., 2024; Saif et al., 2021). This is in line with the increasing trend of the need for adaptive and complex structure-based security systems in the era of big data and increasingly advanced digital communications (Ramasamy et al., 2019).

B. METHODS

This study is classified as Applied Research, utilizing a dual methodology. First, a model that utilizes hypergraph labeling to generate stream keys is developed. The subsequent phase is computational experimental research, which involves simulating the encryption process and empirically evaluating the model's security and efficiency. The strength of the modified caesar cipher encryption is evaluated based on four security analysis approaches, namely brute force attack, encryption time analysis, encrypted character distribution, and encrypted bit size analysis. The primary Research Instrument is the Source Code and Simulation Program developed specifically for this study. Here are the research steps:

1. Hypergraph Model Development and Keystream Generation
   a. Hypergraph Construction: The study utilizes three distinct hypergraph structures and the initial parameters (V, E, H) for Volcano, Semi Parachute, and Comb structures.
   b. Labeling Function Definition: Apply the super $(a, d) -$ hyperedge antimagic total labeling function, specifying the exact formulas for a and d .
   c. Keystream Derivation: The distinct hyperedge weights $w(e)$ resulting from the labeling function are the source of the dynamic key.

2. Encryption Protocol Integration: The generated keystream is integrated into the Vigenère cipher to perform polyalphabetic encryption.
3. Security and Efficiency Evaluation: The final system's performance is rigorously assessed using four scientific test mechanisms. For each mechanism, specific criteria and quantitative indicators are established to validate the findings.

The hypergraph based encryption key generation process starts by designing the hypergraph structures used in this study, namely volcano $\mathcal{V}_{6,8}$, semi parachute $\mathcal{SP}_{6,4}$ dan comb $\mathcal{CB}_{6,6}$. The next stage is hypergraph labelling by applying the super (a,d)-hyperedge antimagic total labelling method, as shown in Figure 1. The resulting label represents the characteristics of each hypergraph. Based on the hypergraph that has been labelled, a constructing block is constructed that illustrates the structure of the hypergraph systematically. The results of this visualization are then used to determine the sequence of numbers that form the basis for assigning the initial digit value i and block length b, which will be used in the key generation process. These two parameters are used to generate the initial key, which acts as the main input in the algorithm, as described in Algorithm 1. After this function is executed, a dynamic stream

key is generated, unique for each hypergraph structure and labelling result used. This stream key is used as the main key in the message encryption and decryption process using the vigenere chiper, so that data security can be significantly improved, as shown in Figure 1.

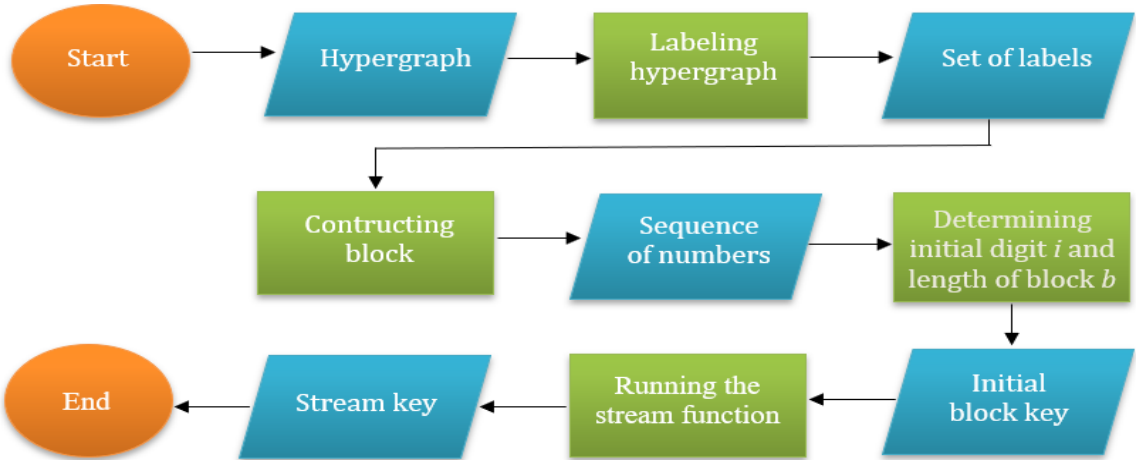


Figure 1. Design of Keystream Generation Model Using Hypergraph Structure

C. RESULT AND DISCUSSION

This section presents the main findings of the study, focusing on the development and implementation of algorithms that integrate hypergraph structures into the Vigenère cipher. The discussion begins with the construction of hypergraph-based models and their corresponding labeling schemes, followed by the design of an algorithm that adapts these models to the encryption and decryption processes of the cipher. Furthermore, the section highlights the potential advantages of employing hypergraphs—such as increased structural complexity and flexibility—in enhancing the security features of the Vigenère cipher. Illustrative examples and computational experiments are provided to demonstrate the applicability of the proposed approach and to analyze its performance in practical settings.

Algorithm 1. Keystream Agorithm using Hypergraph

Input: text (plaintext)
Output: keystream k_i

1 **START**
2 **INPUT** a plaintext
3 **Define** m
4 **Define** n
5 **Check** parity:
  IF $m \bmod 2 == 1$ **and** $n \bmod 2 == 0$ OR
  $n \bmod 2 == 0$ and n $\bmod 2 == 1$, **THEN** continue
  ELSE return to Step 2 with adjusted text

6 **Take** formulas for super (a,d)-hyperedge antimagic total labeling
7 **Compute** label sequences
8 **Combine** all hyperedge labels in sequence
  Name the sequence as k_i
9 **Let** t←length of k_i
10 **END**

Algorithm 1 generates a sequence of numbers that is used as the source of the keystream. To illustrate this algorithm, an example of super (a,d)-hyperedge antimagic total labeling with volcano $\mathcal{V}_{6,8}$, semi parachute $\mathcal{SP}_{6,4}$ dan Comb $\mathcal{CB}_{6,6}$structures is used. The labelling is illustrated in Figure 2, which shows the labels on vertices and hyperedges, including up to their total weights. The resulting keystream sequence consists of several blocks, namely: The Volcano structure consists of 2 blocks: Block $1 = \{ 348, 350, 352 \}$, Block $2 = \{354, 356, 358, 360, 362, 364, 366, 368\}$, The Semi Parachute structure consists of 2 blocks: Block $1 = \{327, 329, 331, 333\}$, Block $2 = \{335, 337, 339, 341, 343, 345, 347\}$, The Comb structure consists of 3 blocks: Block $1 = \{320, 324, 328, 332, 336, 340\}$, Block $2 = \{322, 330, 338\}$, Block 3$= \{326, 334\}$.

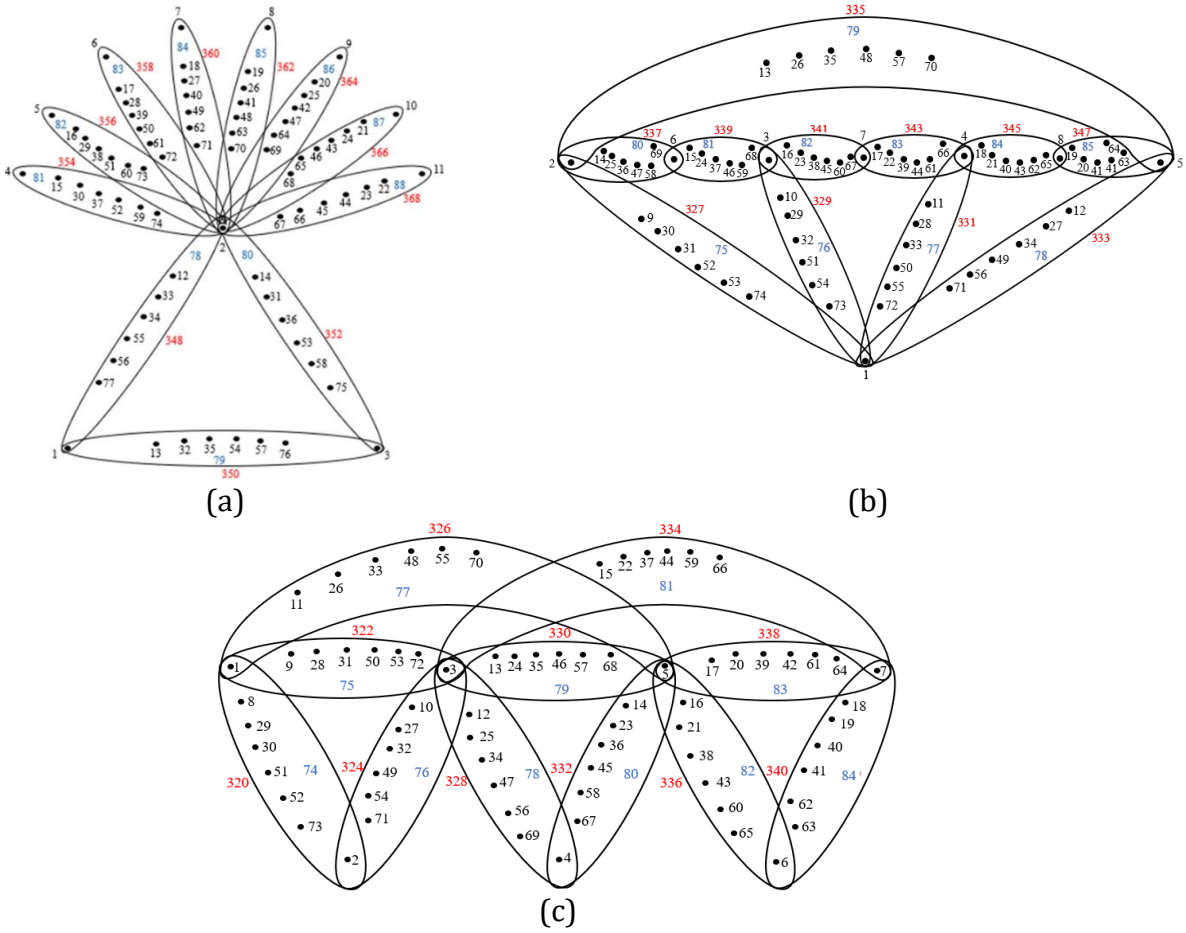


Figure 2. (a)Volcano Hypergraph $V_{6,8}$; (b) Semi Parachute Hypergraph $SP_{6,4}$; and (c)Comb hypergraph $CB_{6,6}$

1. Design and Setup

This research uses MATLAB R2013a software to run the algorithm simulation. Tests were conducted on a system with hardware specifications such as 4 GB RAM, AMD Athlon Gold 3150U processor with a speed of 3.3 GHz, and AMD Radeon Graphics graphics card. To measure the performance of the developed algorithm, a number of datasets with varying plaintext lengths were used, namely 16 bits, 32 bits, 64 bits, 128 bits, 256 bits, and 512 bits. Performance evaluation is done by considering the computation duration and message size after encryption.

The test results are then compared for each type of hypergraph volcano $\mathcal{V}_{6,8}$, semi parachute $\mathcal{SP}_{6,4}$, and comb $\mathcal{CB}_{6,6}$ with the parameters $d = 2$ as the basis for comparison.

2. Keystream Formation Process through Hypergraph Structure

In this discussion, we present an illustration of the keystream generation and encryption steps that utilize a combination of volcano hypergraph, semi parachute hypergraph, and comb hypergraph, which are then combined with the vigenere cipher symmetric encryption algorithm. Table 1, Table 2, Table 3 illustrates the encryption process using the plaintext "PENGHARGAAN" as an example. The first step in the process is to convert each character of the plaintext P_i into a numerical representation x_i, based on a particular mapping such as an ASCII table or a special index in the alphabet. After that, the keystream k_i is generated from the combined structure of the three hypergraphs through applying the super (a,d)-hyperedge antimagic total labeling. The size and configuration of the hypergraph are automatically adjusted according to the length of the plaintext. The security of the system is enhanced by encrypting the keystream using a vigenere cipher that applies a private key. The encryption process is done by summing the plaintext value x_i and the value of the encrypted keystream, followed by a mod 94 operation to keep the result within the character space of the alphabet used. The final ciphertext C_i is then obtained by converting the numeric values back into characters according to the defined alphabet.

Table 4, Table 5, Table 6 presents the decryption steps that reverse the encryption process. Each character in the ciphertext $C_i$is first translated into a numerical index x_i. The previously encrypted keystream is then decrypted using the private key to recover the value k_i. The plaintext value Pi is calculated by subtracting x_i with k_i, then performing the mod 94. The resulting numeric value is then converted back to the original character according to the alphabet used.

Table 1. Encryption on Volcano Hypergraph $\mathcal{V}_{6,8}$

Plaintext	**P**	**e**	**n**	**g**	**h**	**a**	**r**	**g**	**a**	**a**	**n**
x_i	15	30	39	32	33	26	43	32	26	26	39
k_i	348	350	352	354	356	358	360	362	364	366	368
$(x_i + k_i) mod 94$	81	4	15	10	13	8	27	18	14	16	31
Chipertext	>	*E*	*P*	K	N	*I*	*b*	*S*	O	Q	f

Table 2. Encryption Process on Semi Parachute Hypergraph $\mathcal{SP}_{6,4}$

Plaintext	**P**	**e**	**n**	**g**	**h**	**a**	**r**	**g**	**a**	**a**	**n**
x_i	15	30	39	32	33	26	43	32	26	26	39
k_i	327	329	331	333	335	337	339	341	343	345	347
$(x_i + k_i) mod 94$	58	75	86	81	84	81	92	87	83	85	4
Chipertext	6	=	[	>	/	>	~	]	?	{	*E*

Table 3. Encryption Process on Comb Hypergraph $\mathcal{CB}_{6,6}$

Plaintext	**P**	**e**	**n**	**g**	**h**	**a**	**r**	**g**	**a**	**a**	**n**
x_i	15	30	39	32	33	26	43	32	26	26	39
k_i	320	324	328	332	336	340	332	330	338	326	334
$(x_i + k_i) mod 94$	53	72	85	82	87	84	0	80	82	70	91
Chipertext	1	−	{	.	]	/	*A*	,	.	(	′

Table 4. Descryption Process on Volcano Hypergraph $\mathcal{V}_{6,8}$

Chipertext	>	***E***	***P***	**K**	**N**	***I***	***b***	***S***	**O**	**Q**	**f**
x_i	81	4	15	10	13	8	27	18	14	16	31
k_i	348	350	352	354	356	358	360	362	364	366	368
$(x_i - k_i) mod 94$	15	30	39	32	33	26	43	32	26	26	39
Plaintext	P	e	n	g	h	a	r	g	a	a	n

Table 5. Descryption Process on Semi Parachute Hypergraph $\mathcal{SP}_{6,4}$

Chipertext	**6**	=	[	>	/	>	~	]	**?**	**{**	***E***
x_i	58	75	86	81	84	81	92	87	83	85	4
k_i	327	329	331	333	335	337	339	341	343	345	347
$(x_i - k_i) mod 94$	15	30	39	32	33	26	43	32	26	26	39
Plaintext	P	e	n	g	h	a	r	g	a	a	n

Table 6. Descryption Process on Comb Hypergraph $\mathcal{CB}_{6,6}$

Chipertext	**1**	–	{	.	]	/	***A***	,	.	**(**	′
x_i	53	70	81	76	79	74	91	82	80	82	1
k_i	320	324	328	332	336	340	332	330	338	326	334
$(x_i - k_i) mod 94$	53	72	85	82	87	84	94	80	82	70	91
Plaintext	P	e	n	g	h	a	r	g	a	a	n

3. Mathematical Design of Key Flow Based on Hypergraph Structure

Let $H = (V, \mathcal{E})$ be a volcano hypergraph where the vertex set V consist $\{x\} \cup \{x_1, x_2\} \cup \{y_1, y_2, \dots, y_n\}$, and the hyperedge set $\mathcal{E}$ is defined as:

$$\mathcal{E} = \{e_{1,i} = \{x, x_i, x_{i+1}\} | 1 \le i \le 3\} \cup \{e_{2,i} = \{x, y_i, y_{i+1}\} | 1 \le i \le n\}$$

Let $H = (V, E)$ be a semi parachute hypergraph where the vertex set V consist $\{x\} \cup \{x_1, x_2, \dots, x_n\} \cup \{u_1, u_2, \dots, u_n\}$, and the yperedge set $\mathcal{E}$ is defined as:

$$\mathcal{E} = \{e_{1,i} = \{x, x_i, x_{i+1}\} | 1 \le i \le n\} \cup \{e_{2,i} = \{x_i, u_i, x_{i+1}, u_{i+1}\} | 1 \le i \le 2n - 1\}$$

Let $H = (V, \mathcal{E})$ be a comb hypergraph where the vertex set V consist $\{x_1, x_2, \dots, x_n\} \cup \{y_1, y_2, \dots, y_n\}$

$$\mathcal{E} = \{e_{1,i} = \{y_i, x_i, y_{i+1}, x_{i+1}\} | 1 \le i \le n\} \cup \left\{e_{2,i} = \{y_i, y_{i+1}\} | 1 \le i \le \frac{n}{2}\right\}$$
$$\cup \left\{e_{3,i} = \{y_i, y_{i+2}\} | 1 \le i \le \frac{n}{2} - 1\right\}$$

We define a lebeling function $H: V \cup \mathcal{E} \to \mathbb{Z}^+$ satisfying a super (a,d)-hyperedge antimagiclabeling condition, i.e.,

$$\forall e \in w(e) = \sum_{v \in e} H(v) + H(e) = a + (i-1)d, \text{we distinct } w(e)$$

The result weight sequence $\{w(e_1), w(e_2), \dots, w(e_n)\}$ is then converted into a keystream k_i, where each weight is mapped to a benary segment k_i, concatenated to form the final stream used in encryption (Algorithm 2) or descryption (Algorithm 3).

Algorithm 2. Encryption Algorithm with Vigenere Chiper

	Input: Plain text T, Keystream K, Symmetric key (Privat key) K_s
	Output: Cipher text C
1	Initialize alphabet map A and convert T to index vector V using A
2	Generate raw keystream K_raw using hypergraph-based function
3	Encrypt keystream K_raw using Vigenère cipher with key K_s:
4	**foreach** index i in V do
	Compute key index k_i from encrypted keystream K
	Apply modular arithmetic and structured permutation based on k_i
	Append transformed index to cipher vector C
5	**end**
6	Map cipher indices in C back to characters using A:

Algorithm 3. Descryption Algorithm with Vigenere Chiper

	Input: Cipher text C, Encrypted keystream K, Symmetric key (Privat key) K_s
	Output: Plain text T
1	Initialize alphabet map A and convert C to index vector C_idx using A
2	Generate raw keystream K_raw using hypergraph-based function
3	Descrypt keystream K_raw using Vigenère cipher with key K_s
4	foreach index i in C_idx do
	Compute key index k_i from encrypted keystream K
	Apply modular arithmetic and structured permutation based on k_i
	Append transformed index to cipher vector V
5	**end**
6	Map cipher indices in V back to characters using A:

4. Attack Analysis

In this paper, we analyse the time complexity and space complexity of brute force attacks on three types of hypergraphs, namely volcano hypergraph, $(327,2)$-semi parachute hypergraph, and $(320,2)$-comb hypergraph, which are applied in symmetric encryption technique using vigenere cipher method through total antimagic super (a,d)-hyperedge labelling. The time complexity can be seen in Table 7 our results show that the comb hypergraph performs best compared to the semi parachute hypergraph and volcano hypergraph in terms of encryption time, producing the ciphertext faster. This suggests that comb hypergraph is a more efficient choice in terms of processing time.

Table 7. Time Comparison for Encryption (seconds)

Encryption text lenght	16 bytes	32 bytes	64 bytes	128 bytes	256 bytes	512 bytes
Volcano hypergraph $\mathcal{V}_{6,8}$	**0,0572**	**0,0160**	**0,0341**	**0,0666**	**0,1564**	**0,2440**
Semi parachute hypergraph $\mathcal{SP}_{6,4}$	**0,0020**	**0,0002**	**0,0011**	**0,0011**	**0,0020**	**0,0042**
Comb hypergraph $\mathcal{CB}_{6,6}$	**0,0018**	**0,0002**	**0,0005**	**0,0007**	**0,0013**	**0,0030**

The space complexity is shown in Table 8 our analysis reveals that volcano hypergraph and comb hypergraph produce smaller byte sizes compared to semi parachute hypergraph, thus both are superior in terms of storage efficiency.

Table 8. Size Bytes Comparison for Encryption (seconds)

Encryption text lenght	16 bytes	32 bytes	64 bytes	128 bytes	256 bytes	512 bytes
Volcano $\mathcal{V}_{6,8}$	**136**	**264**	**520**	**1032**	**2056**	**4104**
Semi parachute hypergraph$\mathcal{SP}_{6,4}$	**208**	**400**	**784**	**1552**	**3088**	**6160**
Comb hypergraph $\mathcal{CB}_{6,6}$	**136**	**264**	**520**	**1032**	**2056**	**4104**

D. CONCLUSION AND SUGGESTIONS

This study successfully developed and validated a dynamic keystream generation method using super (a, d)-hyperedge antimagic total labeling applied to three hypergraph structures, significantly enhancing the security of the Vigenère cipher. Scientifically, this approach provides a novel foundation for generating cryptographically secure keys by leveraging the complex combinatorial properties of hypergraphs, directly overcoming the key predictability and pattern repetition vulnerabilities inherent in polyalphabetic ciphers. Quantitatively, the Comb hypergraph demonstrated the best performance, achieving minimal encryption time (e.g., 0.0030 seconds for 512 bytes) and producing a low ciphertext frequency variance ($\sigma^2 \approx 0.0001$), thus validating its robustness against frequency analysis and its high efficiency. Based on these findings, we propose an open problem for future research, which is to explore other hypergraph types and their applicability in more robust and adaptive text encryption schemes.

ACKNOWLEDGEMENT

We would like to thank the assistance and support provided by LP2M University of Jember, which has provided encouragement and motivation during the implementation of this research.

REFERENCES

A. Gerhana, Y., Entik, I., Syarifudin, U., & R. Zulmi, M. (2016). Design of digital image application using vigenere cipher algorithm. In *2016 4th International Conference on Cyber and IT Service Management* (pp. 1-5). IEEE. https://doi.org/10.1109/CITSM.2016.7577571

Achmad, A. D., Dewi, A. A., Purwanto, M. R., Nguyen, P. T., & Sujono, I. (2020). Implementation of vigenere cipher as cryptographic algorithm in securing text data transmission. *Journal of Critical Reviews*, *7*(1), 76–79. https://doi.org/10.22159/jcr.07.01.15

Alemami, Y., Mohamed, M. A., & Atiewi, S. (2023). Advanced approach for encryption using advanced encryption standard with chaotic map. *International Journal of Electrical and Computer Engineering*, *13*(2), 1708–1723. https://doi.org/10.11591/ijece.v13i2.pp1708-1723

Ali, N., Sadiqa, A., Shahzad, M. A., Imran Qureshi, M., Siddiqui, H. M. A., Abdallah, S. A. O., & Abd El-Gawaad, N. S. (2024). Secure communication in the digital age: a new paradigm with graph-based encryption algorithms. *Frontiers in Computer Science*, *6*(October). https://doi.org/10.3389/fcomp.2024.1454094

Ardiansyah, F. D., Damayanti, A., Putri, C. A. M., Rany, A. F. D., Biroso, S. J., & Tahir, M. (2023). Implementasi kriptografi Caesar Chiper pada aplikasi enkripsi dan dekripsi. *Jurnal Ilmiah Sistem Informasi dan Ilmu Komputer, 3*(1), 105–112. https://doi.org/10.55606/juisik.v3i1.438

Baagyere, E. Y., Agbedemnab, P. A. N., Qin, Z., Daabo, M. I., & Qin, Z. (2020). A Multi-Layered Data Encryption and Decryption Scheme Based on Genetic Algorithm and Residual Numbers. *IEEE Access*, *8*, 100438–100447. https://doi.org/10.1109/ACCESS.2020.2997838

Badawi, A. Al, Hoang, L., Mun, C. F., Laine, K., & Aung, K. M. M. (2020). PrivFT: Private and Fast Text Classification with Homomorphic Encryption. *IEEE Access*, *8*, 226544–226556. https://doi.org/10.1109/ACCESS.2020.3045465

Dafik, Venkatraman, S., Sathyanarayanan, G., & Baihaki, R. I. (2025). Enhancing Text Encryption and Secret Document Watermarking through Hyperladder Graph-Based Keystream Construction on Asymmetric Cryptography Technology. *Statistics, Optimization & Information Computing, 14*(1), 247-263. https://doi.org/10.19139/soic-2310-5070-2310

Ding, Y., Tan, F., Qin, Z., Cao, M., & Choo, K. K. R. (2021). DeepKeyGen: A deep learning-based stream cipher generator for medical image encryption and decryption. *IEEE Transactions on Neural Networks and Learning Systems, 33*(6), 4915–4929. https://doi.org/10.1109/TNNLS.2021.3062754

Djumadin, Z. (2023). Privacy protection in the big data era: A review of personal data protection policies. *Jurnal Restorasi: Hukum dan Politik, 1*(02), 72–78. https://seaninstitute.or.id/bersinar/index.php/restorasi/article/view/61

Gautam, D., Agrawal, C., Sharma, P., Mehta, M., & Saini, P. (2018, May). An enhanced cipher technique using vigenere and modified caesar cipher. In *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)* (pp. 1-9). IEEE.

Meenakshi, A., Mythreyi, O., Mrsic, L., Kalampakas, A., & Samanta, S. (2025). A Fuzzy Hypergraph-Based Framework for Secure Encryption and Decryption of Sensitive Messages. *Mathematics*, *13*(7), 1–20. https://doi.org/10.3390/math13071049

Moosavi, S. R., Nigussie, E., Levorato, M., Virtanen, S., & Isoaho, J. (2018). Low-latency approach for secure ECG feature based cryptographic key generation. *IEEE Access, 6*, 428–442. https://doi.org/10.1109/ACCESS.2017.2766523

Nasution, S. D., Ginting, G. L., Syahrizal, M., & Rahim, R. (2017). Data security using Vigenere cipher and Goldbach codes algorithm. *International Journal of Engineering Research & Technology (IJERT), 6*(1), 360–363. https://doi.org/10.17577/IJERTV6IS010245

NourEldeen, N. M., Badr, E., Hagag, I. M., & Shabana, H. (2025). Graph-Based Approach to the Radio Assignment Problem in Wireless Communication Networks with Applications in Cryptography. *European Journal of Pure and Applied Mathematics*, *18*(1). https://doi.org/10.29020/nybg.ejpam.v18i1.5614

Pabokory, F. N., Astuti, I. F., & Kridalaksana, A. H. (2016). Implementasi Kriptografi Pengamanan Data Pada Pesan Teks, Isi File Dokumen, Dan File Dokumen Menggunakan Algoritma Advanced Encryption Standard. *Informatika Mulawarman : Jurnal Ilmiah Ilmu Komputer*, *10*(1), 20. https://doi.org/10.30872/jim.v10i1.23

Prihandoko, A. C., Dafik, & Agustin, I. H. (2022). Stream-keys generation based on graph labeling for strengthening Vigenere encryption. *International Journal of Electrical and Computer Engineering*, *12*(4), 3960–3969. https://doi.org/10.11591/ijece.v12i4.pp3960-3969

Ramasamy, P., Ranganathan, V., Kadry, S., Damaševičius, R., & Blažauskas, T. (2019). An image encryption scheme based on block scrambling, modified zigzag transformation and key generation using enhanced Logistic–Tent map. *Entropy, 21*(7), 656. https://doi.org/10.3390/e21070656

Saif, H. A., Gharib, G. M. I., & AL-Mousa, M. R. (2021). a Mathematical Proposed Model for Public Key Encryption Algorithms in Cybersecurity. *Advances in Mathematics: Scientific Journal*, *10*(9), 3063–3092. https://doi.org/10.37418/amsj.10.9.1

Saraswat, A., Khatri, C., Sudhakar, Thakral, P., & Biswas, P. (2016). An Extended Hybridization of Vigenere and Caesar Cipher Techniques for Secure Communication. *Procedia Computer Science*, *92*, 355–360. https://doi.org/10.1016/j.procs.2016.07.390

Soofi, A. A., Riaz, I., & Rasheed, U. (2016). An Enhanced Vigenere Cipher For Data Security. *International Journal of Scientific & Technology Research*, *5*(3), 141–145.

Wang, J., Lan, S., Li, X., Lu, M., Guo, J., Zhang, C., & Liu, B. (2023). Research on the Method of Hypergraph Construction of Information Systems Based on Set Pair Distance Measurement. *Electronics (Switzerland)*, *12*(20), 1–16. https://doi.org/10.3390/electronics12204375