2025,Volume 2 , Issue 7 , PSIJK (58-64)

Pure Sciences International Journal of Kerbala

A Secure Hybrid Symmetric Cryptosystem Combining Huffman Coding, Affine Transformation, and Cartesian Graphs

Ihsan Mezher Rasheed [a*], Kawthar Abdulabbas Hassoon [a]

[a] *Department of Mathematics,, College of Education for Pure Sciences, University of Kerbala , Karbala, Iraq*

P A P E R I N F O

Received: 05.08.2025
Accepted: 16.09.2025
Published: 30.09.2025

***Keywords*:**
Symmetric cryptography, Huffman coding, affine transformation, cartesian product, graphs, encryption, decryption.

A B S T R A C T

We present in this research a new data symmetric cryptography model that combines three techniques: Huffman coding for compression, affine transformation for value obfuscation, and the Cartesian product of graphs to create a complex encrypted graph. The suggested model demonstrates enhanced cryptographic strength and adaptability across various document types. It also provides a high level of security due to to owing to the structural complexity of the generated graph and the multi-layered encryption method.

DOI: 10.53851/psijk.v2.i7. 58-64

Table 1. Nomenclature and Definitions Used in the Study

NOMENCLATURE			
k_i	length of words of plaintext M	Z_{127}	Indices to the ring of integers modulo 127
M	plaintext	P_j	Plaintext block i (matrix form)
A	Invertible transformation matrix (2×2)	x	the padding character
B	Offset matrix (2×2)	V(graph)	represents the vertex set of a graph.
G	Graph used in the Cartesian product	**Subscripts**	
H	A secret keyword-generating Graph that is used in the Cartesian product graph.	i	Indices (words)
C_j	Ciphertext block j	j	Indices (blocks)
$\|H\|$	The length of H	n	Indices total words in the plaintext.

1. INTRODUCTION

The exponential expansion of digital communication and the growing complexity of cyber-attacks have rendered traditional encryption methods insufficient. Consequently, modern research increasingly favors hybrid encryption models that combine multiple layers of protection, aiming to enhance complexity and security without significantly impacting performance. Traditional schemes such as Affinity ciphers, Huffman-based techniques, and graph-theoretic methods are examples of traditional systems that only partially answer the cryptography problem. None of them offer a thorough balance of transformation, compression, algebraic processing, and structural complexity. Our work suggests a hybrid symmetric cryptosystem that combines Cartesian product graphs, affine matrix transformations (executed through matrix multiplication), and Huffman coding in order to close this gap. Although it uses matrix multiplication for affine operations, this combination improves computational and structural security, provides

*Corresponding Author Institutional Email: ihsanmrasheed@uokerbala.edu.iq (Ihsan Mezher Rasheed)

greater resistance to statistical and brute-force attacks, and it is still computationally feasible for contemporary communication systems and resource-constrained settings.

2 .PREVIOUS WORKS

One foundational approach is presented by Beena Kittur et al. (Kittur et al., 2022), who proposed a symmetric encryption model based on Huffman coding combined with an affine transformation, achieving initial value obfuscation. In another direction, Marwah Ali Hussein Ali et al. (Hussein Ali et al., 2023) introduced a cryptographic model based on the Cartesian Product of Graphs (CPG), which enhanced structural encryption by mapping plaintext into graph-based representations. Furthermore, Hassan Rashed Yassein and Hadeel Hadi Abo-Alsood (Abo-Alsood & Yassein, 2021) introduced an enhanced NTRU encryption that boosts efficiency and security. To improve performance and defend against lattice-based assaults, their system makes use of sophisticated matrix operations. This connection is especially pertinent to our work because our proposed hybrid cryptosystem extends the reach of matrix-based cryptography into a multi-layered encryption framework by utilizing matrix multiplications, specifically through affine transformations, in conjunction with Huffman coding and Cartesian product graphs. Additional studies, such as that by Khalid Bekkaoui et al. (Bekkaoui et al., 2020), applied Hamiltonian cycles and divide-and-conquer strategies to graph-based encryption, offering a modular and scalable model for lightweight cryptographic applications. Similarly, P.A.S.D. Perera and G.S. Wijesiri (Perera & Wijesiri, 2021) explored encryption based on matrix graph transformations and key mixing, providing another dimension of cryptographic diffusion. Notably, Ahmed A. Omran et al. (Omran et al., 2024) investigated even sum edge domination in graphs, a concept that can inform the development of graph-based encryption schemes.

3. ESSENTIAL FACTS

Table 2. Standard ASCII Table: Complete Sequential Display(127–0)

Dec	Char	Dec	Char
0	NUL	...	...
1	SOH	32	SP (␣)
2	STX	33	!
...	...	...	...
65	A	97	a
...	...	...	...
122	z	125	}
123	{	126	~
124	|	127	DEL

Note: For character representation in this work, we will only use the first 127 characters (from decimal 0 to 126) of the conventional ASCII encoding. The last ASCII character (decimal value 127), which stands for the DEL (Delete) control character, will not be taken into account.

4. PROPOSED CRYPTOSYSTEM

Building upon these foundational works, this paper proposes a novel symmetric encryption model that unifies three independent security principles into a layered structure:

- Huffman coding for statistical compression and frequency masking,
- Affine matrix transformation for mathematical diffusion and value distortion,
- Cartesian product graph encoding for structural complexity and topological obfuscation.

The synergy of these layers results in a model that is computationally secure, structurally ambiguous, and highly adaptable. A comprehensive procedural description, mathematical foundation, and encryption/decryption example (using the word "mathematics") are presented.

The model's security is analyzed and compared to previous schemes to highlight its robustness and adaptability in modern applications such as secure graph communications, embedded systems, and the Internet of Things IoT.

4.1 Encryption Procedure

Shared Secret Keys (k_i, Leaf Levels, A, B, H_i)

PUBLIC PARAMETERS:

- In this scheme, each word in the plaintext is individually encrypted and transmitted along with its corresponding length (denoted by k_i)
- Leaf Levels: Depths of leaf nodes in the Huffman tree

Step 1: using English characters represented in ASCII Table 2., that are represented by numbers. The user can select a plaintext word or an English sentence made up of a few words, with 126 letters available for selection.

Step 2: The length of the plain text (k_i) has been added to each of these quantities individually, Modulo 127.

Step 3: A pseudo-Huffman tree is constructed where values derived from (the numerical output of the preceding encryption layer + k_i mod 127) are treated as frequency weights for symbolic nodes. This ensures that the Huffman tree's structure remains ambiguous to an unauthorized party, as its construction is based on a secret, intermediate encrypted value rather than the plaintext ASCII.

Step 4: Decimal equivalents are created using the prefix codes of the letters that were found to be leaves in the pseudo-Huffman tree.

Step 5: The sender transmits this ciphertext to the recipient by adding (each leaf's level to the decimal equivalent that was determined in step 3) modulo 127.

Step 6: The ciphertext resulting from the previous phase is segmented into multiple 2×2 matrices P_j. In cases where the total number of characters is not divisible by four (the matrix block size), padding is applied to fill the remaining positions in the final matrix. For consistency, the padding character is chosen as 'the letter x', and its corresponding ASCII value according to Table 2. is used during encryption.

Step 7: Using a Matrix Affine Transformation, the sender encrypts the ciphertext that was obtained in step 6 once more.
Matrix Affine Transformation is defined by the following equation:

$$C_j \equiv \left(A \times P_j + B\right) mod\ 127$$

Where:
A: Invertible transformation matrix (2×2).
B: Offset matrix (2×2).
P_i: is the matrix representation of the plaintext or the output from the previous encryption level (step 6).
.
Step 8: The numerical values obtained after encryption are first mapped back to their corresponding characters using the standard ASCII table. These characters are then used to construct the graph structure. Specifically, the encrypted characters are grouped into a primary set G, which represents the first component of the Cartesian product. A secondary set H, which is derived deterministically from a randomly chosen secret keyword, to form a secure foundation for the construction of the Cartesian product graph, by the following steps:

- Each character in the key corresponds to a unique vertex.
- An edge is drawn from each character to the next in sequence.
- This creates a linear path reflecting the exact order of the key characters.

Step 9: Computing the ciphertext as a Cartesian Product of Graphs (CPG) of two graphs, by the following steps:

1. The Cartesian product graph (CPG) of two simple graphs G and H is denoted as GH where: $V(GH) = \{(u, v): u \in V(G) \text{ and } v \in V(H)\}$
2. In this graph, the vertices (u, v) and (u', v') are adjacent if and only if one of the following conditions is satisfied:

   - **Condition 1**: u = u′ and v is adjacent to v' in H.
   - **Condition 2**: $v = v'$ and u is adjacent to u' in G.

Pseudocode Encryption Procedure
Input: Plaintext
Output: Final Ciphertext
Step 1: Convert charters to ASCII
Step 2: Add length of plain text (k_i) mod 127 to each value
Step 3: Build Pseudo-Huffman Tree using encrypted values as weights
Step 4: Convert binary prefix codes to decimals
Step5: Add leaf level to its decimal mod 127
Sept 6: if Length divisible by 4
Segment into 2x2 matrices
else Pad with 'x'
Segment into 2x2 matrices
end if
Step 7: Apply Affine Transform $C_j \equiv A \times P_j + B$ mod 127 to each matrix
Step 8: Map numbers to chars (G) and Construct Graph H from secret keyword
Step 9: Compute Cartesian Product Graph $G * H$ and get the final Ciphertext $GH = G * H$
End.

4.2 Decryption Procedure

To obtain plain text, execute these steps:

1. Extract the G and H sets from the Cartesian product graph.

2. Apply the inverse affine transformation.

3. Subtract leaf levels from numerical values to obtain the original Huffman decimal codes.

4. Convert decimal numbers into binary prefix codes.

5. Restore the original text with the Huffman tree.

5. EXAMPLE

5.1. Encryption Procedure

Step 1: Convert Plaintext to ASCII Values.

Each character in the input plaintext is translated to its matching ASCII code using Table 2.

Plaintext = Mathematics ASCII values:

$[77, 97, 116, 104, 101, 109, 97, 116, 105, 99, 115]$.

Step 2: Add Plaintext Length Modulo 127

length $k = 11$. Add k to each ASCII value (mod 127).

Modified values are:

$[88, 108, 0, 115, 112, 120, 108, 0, 116, 110, 126]$.

Step 3: Build a pseudo-Huffman tree from the values above.

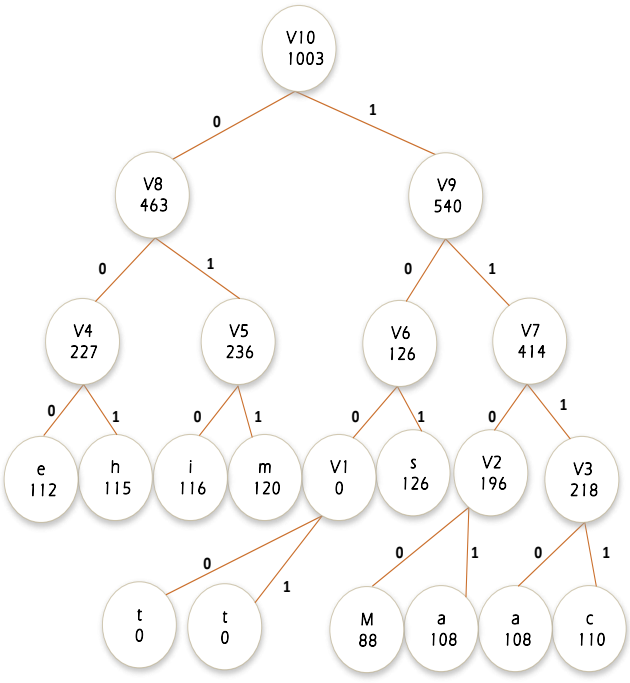

Figure 1. Convert Plaintext to ASCII Values

Table 3. Summarized steps (4 and 5)

Character	Huffman Code	Decimal	Leaf level	Encrypted values
M	1100	12	4	16
A	1101	13	4	17
T	1000	8	4	12
h	001	1	3	4
e	000	0	3	3
m	011	3	3	6
a	1110	14	4	18
t	1001	9	4	13
i	010	2	3	5
C	1111	15	4	19
S	010	5	3	8

Step 6: Form 2×2 Matrices

$$P_1 = \begin{pmatrix} M & a \\ t & h \end{pmatrix} \equiv \begin{pmatrix} 16 & 17 \\ 12 & 4 \end{pmatrix}$$

$$P_2 = \begin{pmatrix} e & m \\ a & t \end{pmatrix} \equiv \begin{pmatrix} 3 & 6 \\ 18 & 13 \end{pmatrix}$$

and $P_3 = \begin{pmatrix} i & c \\ s & x \end{pmatrix} \equiv \begin{pmatrix} 5 & 19 \\ 8 & 120 \end{pmatrix}$

Step 7: Apply Affine Matrix Transformation

Let $A = \begin{pmatrix} 3 & 3 \\ 1 & 2 \end{pmatrix}, B = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$

Where $C_i \equiv (A \times P_i + B)\ mod\ 127$

$$C_1 \equiv (A \times P_1 + B)\ mod\ 127$$
$$\equiv \begin{pmatrix} 16 & 17 \\ 12 & 4 \end{pmatrix} \times \begin{pmatrix} 3 & 3 \\ 1 & 2 \end{pmatrix} + \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \text{ mod } 127$$
$$\equiv \begin{pmatrix} 66 & 83 \\ 41 & 45 \end{pmatrix} \text{ mod } 127$$

$$C_2 \equiv \begin{pmatrix} 3 & 6 \\ 18 & 13 \end{pmatrix} \times \begin{pmatrix} 3 & 3 \\ 1 & 2 \end{pmatrix} + \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$
$$\equiv \begin{pmatrix} 16 & 22 \\ 36 & 81 \end{pmatrix} \text{ mod } 127$$

$$C_3 \equiv \begin{pmatrix} 5 & 19 \\ 8 & 120 \end{pmatrix} \times \begin{pmatrix} 3 & 3 \\ 1 & 2 \end{pmatrix} + \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

$$\equiv \begin{pmatrix} 35 & 54 \\ 145 & 265 \end{pmatrix} \bmod 127$$

$$\equiv \begin{pmatrix} 35 & 54 \\ 18 & 11 \end{pmatrix} \bmod 127$$

Therefore

$$C_1 \equiv \begin{pmatrix} B & S \\ ) & - \end{pmatrix}$$

$$C_2 \equiv \begin{pmatrix} DLE & SYN \\ \$ & Q \end{pmatrix}$$

$$C_3 \equiv \begin{pmatrix} \# & 6 \\ v & VT \end{pmatrix}$$

Mathematics → G =BS)DLESYN$Q#vVT

Steps (8 and **9)**

The encryption key H selected is **All2**.

Then, using the Cartesian product of graphs to generate a complicated encrypted graph to send it to the recipient as follows:

$$GH = G * H =$$

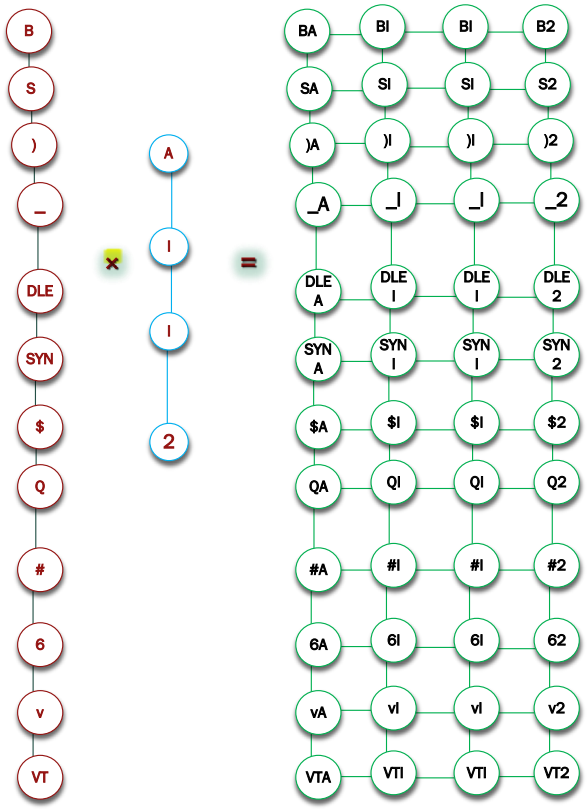

Figure 2. using the Cartesian product of graphs to generate a complicated encrypted graph to send it to the recipient

5.2. Decryption Procedure

Step 1: Retrieve the Cartesian Product GH Graph and decompose it into G and H and extract vertex values: G

Step 2: Convert data to numerical values, then recombine them into a 2×2 matrix from the flat list.

Step 3: Apply Inverse Affine Transformation

$$P_i \equiv (C_i - B) \times A^{-1} \, mod \, 127$$

Such

$$P_1 \equiv (C_1 - B) \times A^{-1} \, mod \, 127$$

$$\equiv \left(\begin{pmatrix} 66 & 83 \\ 41 & 45 \end{pmatrix} - \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \right) \times 3^{-1} \begin{pmatrix} 2 & -3 \\ -1 & 3 \end{pmatrix} \bmod 127$$

$$\equiv 85 \times \begin{pmatrix} 65 & 82 \\ 40 & 44 \end{pmatrix} \times \begin{pmatrix} 2 & -3 \\ -1 & 3 \end{pmatrix} \bmod 127$$

$$\equiv 85 \begin{pmatrix} 48 & 51 \\ 36 & 12 \end{pmatrix} \bmod 127$$

$$\equiv \begin{pmatrix} 16 & 17 \\ 12 & 4 \end{pmatrix} \bmod 127$$

Continuing the calculations to identify the remaining components.

Step 5: Subtract Leaf Levels

Step 6: Convert Decimal to Huffman Binary

Step 7: Decode Using Huffman Tree

Plaintext = Mathematics

6. SECURITY ANALYSIS

6.1. Resistance to Attacks

The proposed encryption model employs a multi-layered security architecture incorporating Huffman compression, affine matrix transformation, and Cartesian graph structures, which collectively strengthen its resistance to various types of cryptographic attacks:

- **Brute-force attacks** Brute-force attacks are computationally infeasible due to the exponentially large key space introduced by variable-length Huffman leaf levels, invertible affine matrix pairs, and complex graph combinations.
- **Statistical attacks** are mitigated through the Huffman compression step, which removes predictable frequency patterns commonly exploited by attackers.
- **Structural analysis attacks** are complicated by the final encoding into a Cartesian Product Graph (CPG), which effectively obfuscates

direct relationships between ciphertext and plaintext.

6.2. Cryptanalysis Perspective

Assuming the plaintext consists of n words $W_1, W_2,..., W_n$, each encrypted independently, the brute-force key space complexity is calculated as follows:

- Each word W_i has a length k_i, where $1 \leq k_i \leq 126$ (based on the usable ASCII range 0–126).
- The number of possible length combinations for n words is 126^n (since each k_i has 126 as a maximum possible value). This term (126^n) represents only the complexity of guessing the words' lengths, not the actual plaintext.
- In the affine transformation stage, we consider all possible combinations of transformation matrix A and offset matrix B:
  - For B: 127^4 possible 2×2 matrices (each element in $\boldsymbol{Z_{127}}$)
  - For A: The total number of invertible 2×2 matrix structures over $\boldsymbol{Z_{127}}$ (Morrison, 2006) is:

$$(127^2 - 1)(127^2 - 127)$$

Furthermore, the final encryption layer, which employs the Cartesian product of graphs, provides its own key space. If the number of vertices in the second graph H is $|H|$, then the Cartesian key space becomes: $\boldsymbol{126^{|H|}}$

6.3. Assumptions Clarification

Independence of Components: The calculation assumes that the encryption of each word and the transformations applied are independent of one another. This means that the choice of key for one component does not affect the others, allowing for a multiplicative combination of their respective key spaces.

- Choice of n: The variable n represents the number of words in the plaintext. The assumption is that this number can vary, and thus, the total key space scales exponentially with n.
- Choice of |H|: Similarly, |H| refers to the number of vertices in graph H. The assumption is that this can also vary, which impacts the Cartesian key space contribution.

Consequently, the total brute-force key space is the product of these components:

$$\begin{aligned} Total\ Key\ Space &= (Key\ Space\ for\ k_i) \\ &\times (Key\ Space\ for\ A\ and\ B) \\ &\times (Key\ Space\ for\ H) \end{aligned}$$

$$\begin{aligned} Total\ Key\ Space &= 126^n \times ((127^2 - 1)(127^2 \\ &- 127) \times 127^4 \times 126^{|H|} \end{aligned}$$

Such complexity is well beyond the capabilities of current high-performance computing systems.

6.4. Comparative Security Analysis

To strengthen the evaluation, we consider the following metrics:

Resistance to Specific Attacks: Analyzing how each model fares against known attack types, such as brute-force, statistical, and structural attacks.

Structural Complexity: Assessing the depth of structural security provided by each model, particularly in the context of data encoding and transformation.

- **Beena et al. (Affine + Huffman):** This model demonstrates effective compression and initial obfuscation. However, it lacks structural complexity. By introducing a topological encryption layer using CPGs, our model significantly enhances structural security.
- **Marwah et al. (CPG-only):** Their work emphasizes graph-based structural encoding but omits value transformation and frequency masking.
- Our hybrid model brings together all three security dimensions.
- **AES (Advanced Encryption Standard):** Although AES continues to be the gold standard in symmetric cryptography, our model offers specific advantages for scenarios involving structured data, such as encrypted graphs or network communications. Furthermore, it is flexible enough to support future integration with public-key systems or hardware-based enhancements.

7. CONCLUSION

This paper introduced a novel hybrid symmetric encryption scheme combining Huffman coding, affine matrix transformation, and the Cartesian product of graphs. The proposed method delivers high structural and computational security by blending data compression, algebraic transformation, and graph theory. The system proves resilient against brute-force and statistical attacks, and it holds promise for adaptable use in secure communications, structured data protection, and

lightweight cryptographic systems in constrained environments like IoT. Future research can explore how using the tensor product of graphs might improve the security of graph-based cryptographic systems, along with adding additional security mechanisms, and exploring extensions to public-key frameworks or hardware-level integration.

REFERENCES

Abo-Alsood, H. H., & Yassein, H. R. (2021). Design of an alternative NTRU Encryption with High Secure and Efficient. *International Journal of Mathematics and Computer Science*, *16*(4), 1469–1477.

Bekkaoui, K., Ziti, S., & Omary, F. (2020). A robust scheme to improving security of data using graph theory. *International Journal of Advanced Computer Science and Applications*, *11*(5).

Hussein Ali, M. A., Omran, A. A., & Ajeena, R. K. K. (2023). The cartesian product graph for encryption schemes. *AIP Conference Proceedings*, *2591*(1), 1–9.

Kittur, B., Kumari, D. C., Kulkarni, S. G., & KM, M. (2022). Super-Encryption Method using Affine Transform Via Trees. *International Journal of Mathematics Trends and Technology-IJMTT*, *68*(6), 190–194.

Morrison, K. E. (2006). Integer sequences and matrices over finite fields. *Journal of Integer Sequences*, *9*(06.2.1), 1–28.

Omran, A. A., Rasheed, I. M., Obaid, S. A., & Talib, R. H. (2024). Even Sum Edge Domination in Graph. *Pure Sciences International Journal of Kerbala*, *1*(4), 59–63.

Perera, P., & Wijesiri, G. S. (2021). Encryption and decryption algorithms in symmetric key cryptography using graph theory. *Psychology and Education Journal*, *58*(1), 3420–3427.