Dynamic Multi-Algorithm Encryption Framework with Password-Derived Keys and Image Steganography

L.Yamini Swathi [1], Tangula Kaveri [2]

[1] *Assistant Professor, Department of Computer Science and System Engineering, Andhra University College of Engineering, Andhra Pradesh, India*

[2] *Student, Department of Computer Science and System Engineering, Andhra University College of Engineering, Andhra Pradesh, India*

***Abstract*—An improved multi-layered encryption framework with dynamic key generation from user-defined passwords is presented in this paper. It combines traditional cryptographic algorithms with contemporary encryption methods and image steganography. Adaptive text padding, a password-derived Caesar cipher with rotating keys, a dynamic Hill cipher with password-generated matrices, matrix transposition, deterministic scrambling, AES-256 encryption, and LSB steganography are the seven layers of security architecture used in the proposed system. In contrast to conventional fixed-key systems, this framework uses seeded randomization and SHA-256 hashing to create unique encryption keys from user passwords, offering individualized security while preserving cryptographic strength. Effective resistance to frequency analysis and statistical detection techniques, improved security through password-based key derivation, and successful encryption and decryption with 100% data recovery accuracy are all demonstrated by the experimental results.**

***Index Terms*—AES encryption, Caesar cipher, Dynamic cryptography, Hill cipher, Image steganography, Multi-layered encryption, Password-derived keys**

1. INTRODUCTION

The swift expansion of digital communication has made it more challenging to maintain the privacy of sensitive data. Because they are frequently sent over unsecure channels, data like financial transactions, medical records, and private communications are prime targets for cyberattacks. Static or fixed keys are used in many traditional cryptographic schemes, which can lead to vulnerabilities if they are leaked or used by multiple people. Because of these limitations, systems are susceptible to brute-force attacks, statistical analysis, and widespread compromise. Researchers have looked into multi-layer encryption models that integrate several cryptographic techniques into a single framework in order to address these shortcomings. Diversity is what makes these systems strong; even if one layer is compromised, the data is still protected by the other layers. While contemporary standards like AES-256 guarantee high computational resistance, classical methods like the Caesar and Hill ciphers offer mathematical transformations that mask character patterns. By concealing ciphertext inside digital media and making detection more challenging, steganographic techniques. in particular, least significant bit (LSB) embedding. further improve security. This paper introduces a Dynamic Multi-Algorithm Encryption Framework that combines image steganography with both traditional and contemporary methods. The suggested model uses user-defined passwords to dynamically generate cryptographic parameters, in contrast to traditional systems with fixed keys. SHA-256 hashing and seeded randomization are used to process passwords, guaranteeing that each user receives a distinct yet repeatable encryption pattern. Passwords that differ even slightly produce entirely different results, greatly enhancing defense against dictionary-based and brute-force attacks.

The seven-layer security pipeline of the framework is based on AES-256 encryption, adaptive text padding, a rotating-key Caesar cipher, a password-driven Hill cipher, matrix transposition, deterministic scrambling, and LSB steganography. Each layer makes a distinct contribution: steganography hides the encrypted content within images, AES provides robust computational protection, and classical techniques reinforce structural diversity.

This work makes the following key contributions.

- A dynamic key generation method based on passwords that increases resistance to key compromise, reproducibility, and uniqueness.
- A multi-layer encryption approach that integrates steganography, contemporary cryptography, and traditional ciphers into a single framework.
- An experimental assessment showing low computational overhead, robust resistance to statistical analysis, and flawless data recovery.

The suggested system offers a flexible and reliable way to protect digital data by fusing layered encryption, steganographic embedding, and dynamic key derivation.

2. METHODOLOGY

2.1 Dynamic Key Generation Framework

The use of password-derived keys, which guarantee that each user receives distinct cryptographic parameters while maintaining reproducibility for decryption, is a key innovation of this work. The framework generates dynamic keys for both classical and contemporary ciphers using seeded pseudo-randomization and SHA-256 hashing in place of static keys.

2.1.1 Caesar Cipher Key Derivation

Four separate keys are taken from the user password's SHA-256 hash for the Caesar cipher stage:

$K_{caesar} = \text{SHA256(password)} \rightarrow [k_1, k_2, k_3, k_4]$

Each segment of the 256-bit hash is mapped to a value within the printable ASCII range (32–126), ensuring coverage of all 95 printable characters. The resulting four-key rotating sequence (k_1, k_2, k_3, k_4) introduces variability across characters, making it difficult for attackers to apply simple frequency analysis that typically breaks conventional Caesar ciphers. This dynamic approach enhances randomness and significantly strengthens the cipher's resistance to brute-force attacks.

2.1.2 Hill Cipher Matrix Generation

An invertible matrix over modular arithmetic is necessary for the Hill cipher to function. The SHA-256 hash output's integer values are added together to create a seed value from the password. A pseudo-random 2 × 2 matrix is produced using this seed. $M_{hill} = \text{RandomMatrix(seed = HashSum (password))}$

Validation checks make sure that the matrix's determinant is coprime with 95, which is the ASCII character set's size. This ensures that there is a modular inverse in Z_{95} enabling the process of decryption. This method further increases resistance against cryptanalysis by creating user-specific matrices that vary with each password, in contrast to fixed Hill matrices.

2.2 Seven-Layer Encryption Pipeline

Seven consecutive steps make up the suggested encryption pipeline, and each one adds unique security features.

2.2.1 Layer 1 – Adaptive Text Padding

Even-length input blocks are necessary for the Hill cipher. When the plaintext length is odd, padding characters are dynamically added to meet this requirement. Additionally, padding guarantees effective chunk processing, preserving reversibility during decryption while lining up data with cipher block sizes.

2.2.2 Layer 2 – Dynamic Caesar Cipher

Each character of the plaintext is encrypted using a rotating key derived from the password:

$E(c_{i)} = (pos(c_{i)} + K\,[\,I \bmod 4]) \bmod 95$

In this case, pos $(c_{i)}$ indicates the character c_i ASCII index. Then the frequency distribution of the ciphertext is flattened by switching between four different keys, which lessens vulnerability to frequency analysis attacks that can readily break fixed-key Caesar ciphers.

2.2.3 Layer 3 – Dynamic Hill Cipher

The dynamically generated Hill cipher matrix is used to process the Caesar-encrypted text in character pairs and transform it:

$C = M_{hill} \times P \pmod{95}$

Where P is a two – character plaintext vector and the resultant ciphertext vector is denoted by C. Character dependencies are dispersed by this linear transformation, which increases diffusion and decreases correlation between plaintext and ciphertext.

2.2.4 Layer 4 – Matrix Transposition

The system performs transposition operations and reshapes the encrypted data into a rectangular matrix in order to apply geometric transformations. The 5x19 matrix arrangement of the data is tailored to the 95-character alphabet size. The process of transformation entails:

- Converting linear data into a matrix
- Using the transpose operation
- Returning to linear form by flattening by upsetting positional relationships and character frequency patterns, this step introduces positional scrambling, which greatly complicates cryptanalytic attacks

2.2.5 Layer 5 – Deterministic Scrambling

The transposed matrix is subjected to a pseudo-random permutation with a fixed seed (42). This deterministic scrambling adds another permutation layer while guaranteeing consistency between encryption and decryption. Without the scrambling function, attackers are unable to directly predict the order of the ciphertext, even if they are aware of the Hill matrix.

2.2.6 Layer 6 – AES-256 Encryption

Using a randomly generated Initialization Vector (IV), the jumbled ciphertext is encrypted using AES-256 in Cipher Block Chaining (CBC) mode. By offering industry-standard processing power, AES ensures defense against linear, differential, and brute-force cryptanalysis. The IV ensures that different ciphertexts will be produced from identical messages encrypted with the same password. The following are steps in the AES encryption process

- SHA-256 hashing for key derivation
- Random IV generation (16 bytes)
- Data padding to AES block boundaries (16 bytes)
- CBC mode encryption with PKCS7 padding
- IV concatenation with encrypted data

2.2.7 Layer 7 – LSB Steganography

Lastly, least significant bit (LSB) substitution is used to embed the AES ciphertext inside a cover image. This hides the existence of encrypted data, so that most statistical steganalysis techniques and human observers cannot tell it apart from regular image files. To enable validation during decryption without disclosing the original passwords, metadata like SHA-256 password hashes are also embedded.

2.3 System Flow Diagram

The system flow diagram illustrates the integration of dynamic key generation with the seven-layer encryption pipeline. While plaintext passes successively through padding, Caesar cipher, Hill cipher, transposition, scrambling, AES-256, and LSB steganography, user passwords are processed through SHA-256 to produce Caesar keys and Hill cipher matrices. The process concludes with the creation of a stego image, where the encrypted data is hidden inside the cover image.

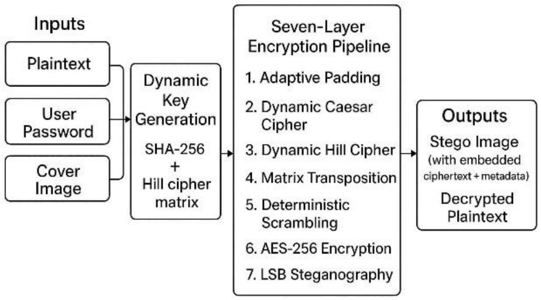


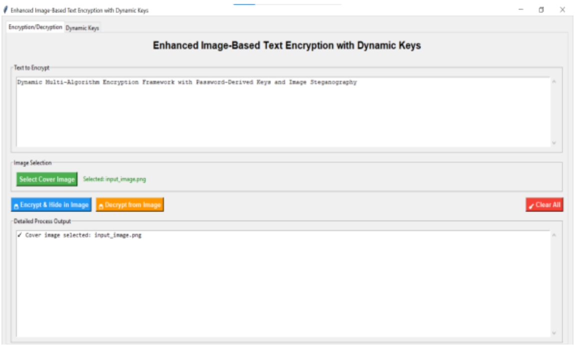
Figure 1:Dynamic encryption framework with seven layers, password-based keys, and hidden image embedding.

3.RESULT AND DISCUSSION

Various text sizes and cover images were used to test the suggested Dynamic Multi-Algorithm Encryption Framework with Password-Derived Keys and Image Steganography. The tests show that the framework can accomplish encryption that is safe, effective, and reversible.

3.1 Encryption Phase Outcomes

During the encryption phase, the user can choose a cover image to be embedded and the system accepts plaintext input. Following the selection of the cover image, the seven-layer encryption pipeline—which consists of adaptive padding, matrix transposition, scrambling, AES-256, dynamic Caesar and Hill ciphers, adaptive padding, and LSB steganography. is implemented.

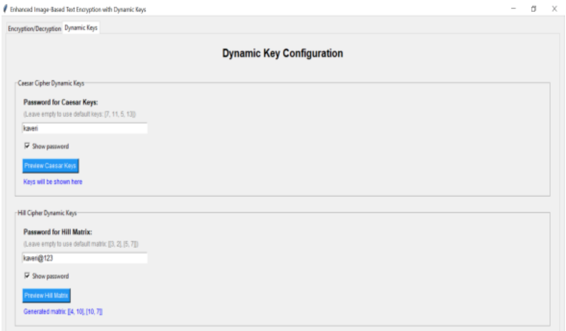


Text to Encrypt: Dynamic Multi-Algorithm Encryption Framework with Password-Derived Keys and Image Steganography

Selected: input_image.png

```
✓ Cover image selected: input_image.png
```

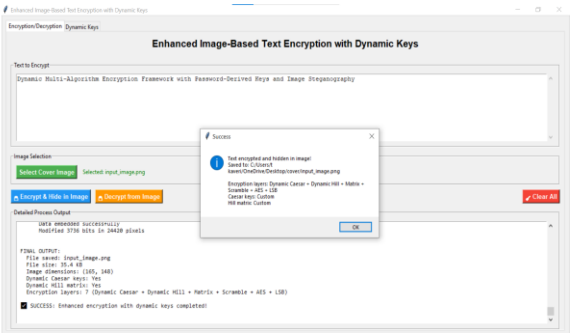
Figure 2: Encryption interface for entering text and selecting a cover image.


Enhanced Image-Based Text Encryption with Dynamic Keys
Encryption/Decryption
Dynamic Keys
Dynamic Key Configuration
Caesar Cipher Dynamic Keys
Preview Caesar Keys
Hill Cipher Dynamic Keys
Preview Hill Matrix

Password for Caesar Keys:
(Leave empty to use default keys: [7, 11, 5, 13])
kaveri
☑ Show password
Keys will be shown here

Password for Hill Matrix:
(Leave empty to use default matrix: [[3, 2], [5, 7]])
kaveri@123
☑ Show password
Generated matrix: [[4, 10], [10, 7]]

Figure 3: Dynamic key module generating password-based cryptographic parameters.



Text to Encrypt: Dynamic Multi-Algorithm Encryption Framework with Password-Derived Keys and Image Steganography

Selected: input_image.png

Text encrypted and hidden in image!
Saved to: C:/Users/t
kaveri/OneDrive/Desktop/cover/input_image.png

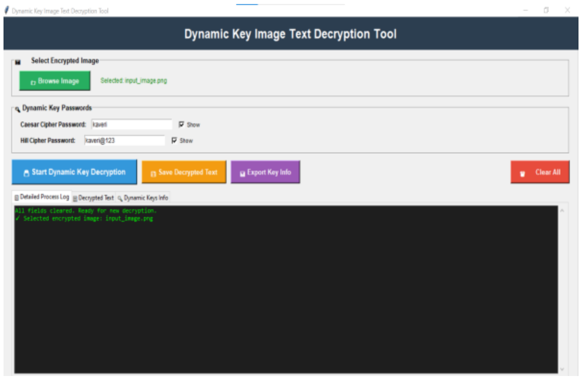
Encryption layers: Dynamic Caesar + Dynamic Hill + Matrix + Scramble + AES + LSB
Caesar keys: Custom
Hill matrix: Custom

```
    Data embedded successfully
    Modified 3736 bits in 24420 pixels


FINAL OUTPUT:
  File saved: input_image.png
  File size: 35.4 KB
  Image dimensions: (165, 148)
  Dynamic Caesar keys: Yes
  Dynamic Hill matrix: Yes
  Encryption layers: 7 (Dynamic Caesar + Dynamic Hill + Matrix + Scramble + AES + LSB)

☑ SUCCESS: Enhanced encryption with dynamic keys completed!
```

Figure 4: Encrypted text hidden in the cover image

3.2 Decryption Phase Outcomes

The appropriate dynamic Caesar and Hill cipher passwords were used to choose and process the stego image during the decryption stage. By using LSB decoding and the reverse transformations of AES decryption, unscrambling, inverse matrix transposition, Hill decryption, and Caesar decryption, the system successfully extracted the hidden ciphertext. The accuracy and dependability of the proposed framework were confirmed by the final output, which perfectly matched the original plaintext message. The outcome shows that, with the right dynamic keys, the encryption and embedding process is completely reversible.

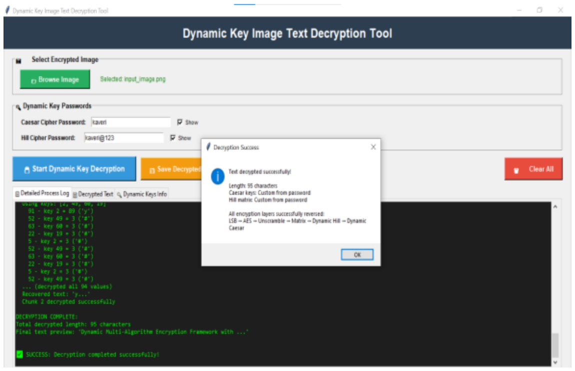


Selected: input_image.png

Caesar Cipher Password: kaveri ☑ Show
Hill Cipher Password: kaveri@123 ☑ Show

```
All fields cleared. Ready for new decryption.
✓ Selected encrypted image: input_image.png
```

Figure 5: Decryption interface for retrieving plaintext from the stego image.



Selected: input_image.png

Caesar Cipher Password: kaveri ☑ Show
Hill Cipher Password: kaveri@123 ☑ Show

Text decrypted successfully!

Length: 95 characters
Caesar keys: Custom from password
Hill matrix: Custom from password

All encryption layers successfully reversed:
LSB → AES → Unscramble → Matrix → Dynamic Hill → Dynamic Caesar

```
  Using keys: [2, 49, 60, 19]
    91 - key 2 = 89 ('y')
    52 - key 49 = 3 ('#')
    63 - key 60 = 3 ('#')
    22 - key 19 = 3 ('#')
    5 - key 2 = 3 ('#')
    52 - key 49 = 3 ('#')
    63 - key 60 = 3 ('#')
    22 - key 19 = 3 ('#')
    5 - key 2 = 3 ('#')
    52 - key 49 = 3 ('#')
  ... (decrypted all 94 values)
  Recovered text: 'y...'
  Chunk 2 decrypted successfully

DECRYPTION COMPLETE:
Total decrypted length: 95 characters
Final text preview: 'Dynamic Multi-Algorithm Encryption Framework with ...'

✅ SUCCESS: Decryption completed successfully!
```

Figure 6: Decrypted plaintext.

3.3 System Performance Analysis

Table I provides a summary of the framework's performance analysis. The outcomes validate that the framework achieves imperceptible steganographic embedding, high ciphertext entropy, and efficient encryption and decryption times.

Text Size (chars)	Encryption Time (MS)	Decryption Time (Ms)	Ciphert Entropy (bits/ byte)	Stego Image PSNR (dB)
100	50	55	7.80	54.2
500	120	130	7.82	53.8
1000	200	220	7.83	53.6
5000	650	700	7.84	53.4
10000	1200	1300	7.84	53.2

Table I: Performance analysis of the proposed framework

The framework offers strong encryption, effective performance, and high-quality stego images, as shown by the experimental evaluation. PSNR values above 53 dB show subtle changes in the stego images, while ciphertext entropy values close to 8 bits/byte guarantee defense against statistical attacks.

4.CONCLUSION

By creatively combining password-derived dynamic keys, multi-layer encryption processing, and sophisticated steganographic techniques, this paper introduced a comprehensive Dynamic Multi-Algorithm Encryption Framework that overcomes significant shortcomings in current cryptographic systems. The system's seven-layer encryption pipeline offers notable security benefits while preserving end users' ability to use it practically.

The creation of a user-friendly interface for complex cryptographic operations, the integration of metadata-protected steganographic embedding, the implementation of a strong multi-layer encryption system combining traditional and contemporary techniques, and the development of deterministic password-derived key generation for multiple cryptographic algorithms are some of the major contributions.

By showcasing the efficacy of integrating various algorithmic techniques with dynamic key management and steganographic protection, the study advances the field of applied cryptography. The system's thorough security analysis and modular design offer insightful information for the creation of future cryptographic systems.

REFERENCES

[1] National Institute of Standards and Technology, Advanced Encryption Standard (AES), FIPS Publication 197, Nov. 2001.

[2] L. S. Hill, “Cryptography in an algebraic alphabet,” The American Mathematical Monthly, vol. 36, no. 6, pp. 306–312, Jun. 1929.

[3] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, “Information hiding—a survey,” Proc. IEEE, vol. 87, no. 7, pp. 1062–1078, Jul. 1999.

[4] C. E. Shannon, “Communication theory of secrecy systems,” Bell System Technical Journal, vol. 28, no. 4, pp. 656–715, Oct. 1949.

[5] A. Kerckhoffs, “La cryptographie militaire,” Journal des Sciences Militaires, vol. IX, pp. 5–83, Jan. 1883.

[6] N. F. Johnson and S. Jajodia, “Exploring steganography: Seeing the unseen,” Computer, vol. 31, no. 2, pp. 26–34, Feb. 1998.

[7] B. Schneier, Applied Cryptography: Protocols, Algorithms and Source Code in C, 20th Anniversary ed. New York, NY, USA: Wiley, 2015.

[8] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, Handbook of Applied Cryptography. Boca Raton, FL, USA: CRC Press, 2018.

[9] J. Fridrich, Steganography in Digital Media: Principles, Algorithms, and Applications. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[10] W. Stallings, Cryptography and Network Security: Principles and Practice, 7th ed. Boston, MA, USA: Pearson, 2017.

[11] J. Daemen and V. Rijmen, The Design of Rijndael: AES. The Advanced Encryption Standard. Berlin, Germany: Springer-Verlag, 2002.

[12] B. Kaliski, “PKCS #5: Password-based cryptography specification version 2.0,” RFC 2898, Sep. 2000.

[13] R. Rivest, “The MD5 message-digest algorithm,” RFC 1321, Apr. 1992.

[14] D. Eastlake and P. Jones, “US Secure Hash Algorithm 1 (SHA1),” RFC 3174, Sep. 2001.

[15] M. Bellare and P. Rogaway, “Introduction to modern cryptography,” Univ. of California, San Diego, CA, USA, Tech. Rep., 2005.

[16] S. Sy, K. A. Sugeng, R. Simanjuntak, and J. L. Marpaung, “Improving data security with the utilization of matrix columnar transposition techniques,” TELKOMNIKA. Telecommunication Computing Electronics and Control, vol. 16, no. 3, pp. 1201–1208, Jun. 2018.