Journal of Applied Informatics and Computing (JAIC)
Vol.10, No.4, August 2026, pp. 3857~3871
e-ISSN: 2548-6861

Super Encryption with Hill Cipher and Myszkowski Transposition for Android Multimedia Files Security

Aldi Febriayanto[1], Eko Hari Rachmawanto[2*]
Department of Informatics Engineering, Universitas Dian Nuswantoro, Semarang
[1]111202315056@mhs.dinus.ac.id, [2]eko.hari@dsn.dinus.ac.id*

Article Info

Article history:

Received 2026-06-10
Revised 2026-08-10
Accepted 2026-08-12

Keyword:

Super Encryption,
Hill Cipher,
Myszkowski Transposition,
Android,
Multimedia Security,
Entropy.

ABSTRACT

This research aims to design, implement, and evaluate an Android-based data security application called Secure Vault using a super encryption method that combines Hill Cipher (substitution) and Myszkowski Transposition (transposition). The background of this research is motivated by the increasing number of cyberattacks on Android devices, particularly those targeting multimedia data. The method used is Research and Development with the Waterfall model. A total of 300 synthetic multimedia files (images, videos, documents, texts) were encrypted and decrypted using three methods: Hill Cipher alone, Myszkowski Transposition alone, and super encryption. Testing included computational time, Shannon entropy values, black box testing (15 scenarios), User Acceptance Test (UAT) with 30 respondents, and statistical analysis (ANOVA, Pearson correlation, paired t-test). The results showed that super encryption produced an average encryption time of 7.56 seconds and the highest entropy increase (0.92 bits/byte, approaching 92.8% of the maximum 8 bits/byte). The encryption/decryption success rate for valid files reached 100%, with a black box pass rate of 90.3%. The UAT obtained an average score of 3.90/5 (Good category). The Pearson correlation between file size and encryption time was very strong ($r = 0.939$, $p < 0.001$). The conclusion of this research is that super encryption effectively enhances multimedia data security on Android with high entropy and perfect functional success, although there is a trade-off in processing time that remains acceptable. The Secure Vault application is feasible for protecting personal data.

I. INTRODUCTION

The rapid advancement of information and communication technology has fundamentally transformed how digital data is stored and exchanged. With a global market share exceeding 72% as of 2024, Android-based mobile devices have become one of the world's most dominant platforms [1]. Indonesia has more than 190 million smartphone users, making it one of the largest smartphone markets in Southeast Asia [2]. This widespread adoption correlates directly with the increasing volume of multimedia files such as images, videos, documents, and other personal files that are created, stored, and shared daily. However, such reliance on mobile devices also brings serious data security risks. The National Cyber and Crypto Agency of Indonesia (BSSN) recorded over 1.6 billion cyber-attacks throughout 2023, with approximately 40% targeting personal data and multimedia files. IBM Security (2023) further reported that the global average cost of a data breach reached USD 4.45 million per incident, a 15% increase from previous years. These statistics emphasize that data protection on Android devices is no longer optional but an urgent necessity [3].

Cryptography remains one of the most effective solutions for preserving data confidentiality [4]. Classical algorithms such as Hill Cipher [5], Rail Fence Cipher [6], and Myszkowski Transposition Cipher [7] have proven capable of protecting text-based data by transforming it into an unintelligible form without the correct key. Hill Cipher, introduced by Lester S. Hill in 1929, employs matrix operations to perform polyalphabetic substitution and provides strong diffusion properties. However, a single algorithm like Hill Cipher is vulnerable to modern

cryptanalysis attacks, including known-plaintext attacks and frequency analysis, particularly when using a small key matrix. Meanwhile, the Myszkowski Transposition Cipher patented in 1902 is an advanced variant of columnar transposition that handles repeated keyword characters, resulting in more complex data permutation than simple Rail Fence Cipher [8]. Yet, transposition-only ciphers remain susceptible to pattern-based attacks when used alone [9]. Consequently, combining multiple algorithms to compensate for each other's weaknesses is essential.

Previous research has extensively explored cryptography on mobile platforms. Previous research successfully implemented a 2×2 Hill Cipher for text encryption on Android, but their study was limited to text files and did not extend to multimedia formats [10], [11]. Secured text files using Hill Cipher with ASCII codes, yet without an additional transposition layer. Supplied Myszkowski Transposition Cipher to documents but used only a single algorithm, leaving room for security enhancement [7]. On the other hand, combined Autokey Cipher with columnar transposition in a super encryption scheme. However, their implementation was not Android-based and lacked quantitative security metrics such as entropy or pixel correlation analysis. Integrated AES and RSA for data transmission, but these modern algorithms demand considerable computational resources [12], [13], making them less suitable for battery-constrained mobile devices [14], [15], [16]. Thus, there remains a clear research gap for a lightweight, yet robust solution specifically tailored for multimedia file protection on Android by leveraging enhanced classical ciphers within a super encryption framework.

Based on the problem identification above, this research aims to design, implement, and evaluate an Android-based data security application using a super encryption method that combines Hill Cipher (as a substitution layer) and Myszkowski Transposition (as a transposition layer). The application named Secure Vault supports the encryption and decryption of various multimedia file types, including images (JPG, PNG), videos (MP4), documents (PDF, DOCX, XLSX), and text files (TXT). This study further aims to analyse application performance based on computational time across different file sizes, measure the randomness (entropy) of encrypted files as a security indicator, and conduct functional testing using the Black Box Testing method to verify that all features operate as specified. Additionally, user acceptance is evaluated through the User Acceptance Test (UAT).

The novelty of this research lies in several key aspects. First, no prior study has specifically combined a 2×2 Hill Cipher matrix derived from a keyword and Myszkowski Transposition using a predefined keyword (ANDROID) within a super encryption framework for multimedia files on the Android platform. Second, this research not only validates functional success but also provides quantitative security analysis through entropy measurement before and after encryption, demonstrating how closely the encrypted data approaches perfectly random conditions (maximum 8 bits/byte). Third, this study presents a comprehensive performance evaluation, including linear regression analysis of file size versus processing time and ANOVA statistical tests to compare algorithm performance. Fourth, the Secure Vault application supports various file types (images, videos, documents) with an intuitive user interface, featuring image preview and direct saving of decrypted results to the gallery functionalities rarely found in similar encryption applications. With these novel contributions, this research is expected to advance.

This research focuses specifically on the confidentialy aspect of multimedia data protection through cryptographic encryption. While integrity, authentication, and access control are important dimensions of comprehensive multimedia security, they are beyond the scope of the current study and are identified as directions for future research.

II. METHOD

A. Research Design

The research design consists of three main streams, Android software development, cryptographic algorithm implementation, and performance and security experimentation. A *true experimental design* is applied with independent variables being the encryption method (Hill Cipher alone, Myszkowski Transposition alone, and their combination as Super Encryption) based on dependent variables consisting of computational time, entropy values, success rate, and user acceptance scores.

B. Research Procedure

The research procedure follows systematic steps as illustrated in Figure 1. Requirements Analysis is a functional requirements. The application must allow file selection from storage, encrypt using super encryption, save the `.enc` file to the Downloads folder, decrypt back to the original format, support various multimedia files (JPG, PNG, MP4, PDF, DOCX, XLSX, TXT), provide image preview, and include a reset button. Non-functional requirements: security (high entropy), usability (intuitive interface), compatibility (minimum Android 8.0), and performance (linear processing time with respect to file size).

Figure 1. Research procedure

Total data of 300 synthetic multimedia files were generated, covering, 80 jpg/png image files (size 0.1 untuk 5 MB), 60 MP4 video files (size 5 – 50 MB), 100 document files (PDF, DOCX, XLSX, TXT0 with sizes 0.05 untuk 15 MB and 60 additional mixed type files to reach 300 records as shown in Figure 2. Each file is labeled with its type, size, and applied encryption method (Hill Cipher only, Myszkowski only, or Super Encryption). The dataset was generated in a structured random manner using Python with

the `random` and `pandas` libraries to ensure realistic variation. Testing series were conducted. Black Box Testing, 15 test scenarios (select file, encrypt, decrypt, preview, reset, save to gallery) were executed on a physical OPPO CPH2239 device (Android 11, 6 GB RAM). Results were recorded as "Pass", "Warning", or "Fail". Computational time measurement, each file was encrypted and decrypted three times; the average time in seconds was recorded using `System.currentTimeMillis()`. Entropy measurement, is an entropy values (H) were calculated using the formula $H = -\sum p\ (x_i) log_2\ p(x_i)$ for both original and encrypted files. The theoretical maximum is 8 bits/byte. Calculations were performed with a Python script. Statistical test using One-way ANOVA to compare encryption times among the three methods. Pearson correlation to measure the relationship between file size and processing time. Paired t-test to compare entropy before and after encryption for the super encryption method. Collected data were analyzed descriptively (mean, standard deviation, distribution) and inferentially (hypothesis testing). Boxplots, histograms, and a correlation heatmap were generated to visualise the findings.

The security of the Secure Vault application depends not only on algorithmic strength but critically on secure key generation, storage, and protection mechanisms. The encryption key is derived deterministically from a user-provided keyword, where the first four characters are converted to numerical values (A=0, B=1, ..., Z=25) to construct the 2×2 Hill Cipher matrix, while the Myszkowski Transposition uses the complete keyword (minimum 4 characters, with "ANDROID" as the default) to determine column ordering. A fallback mechanism automatically regenerates the key matrix if the determinant is not coprime with 256, ensuring a valid invertible matrix is always produced. The application does not persistently store encryption keys on the device; instead, keys are generated dynamically during each session and held exclusively in volatile memory (RAM) during cryptographic operations, with key material cleared immediately upon completion. This approach minimizes persistent storage risks but requires users to remember their chosen keyword for successful decryption. For enhanced protection, the application leverages the Android Keystore system (API level 18+) to safeguard temporary key material when available, providing a secure container that renders keys non-exportable and protects them through user authentication mechanisms including lock screen credentials, PINs, passwords, and biometrics (fingerprint or face recognition).

Despite these security measures, several inherent limitations must be acknowledged. The reliance on a user-memorized keyword as the sole source of key entropy introduces vulnerability to dictionary attacks and weak password selection, while the deterministic generation means identical keywords always produce the same key matrix a weakness that could be exploited if an attacker discovers the keyword. Additionally, the effective key space of approximately $2^{43.5}$ is substantially smaller than modern cryptographic standards, as discussed in Section 3.3.5. To address these limitations, future implementations should consider incorporating salt or nonce values to ensure unique keys from identical keywords, implementing key derivation functions (KDFs) such as PBKDF2 or Argon2 to strengthen user-provided passwords, supporting biometric authentication as an additional access factor, and providing users with the option to securely export and backup encrypted keys for recovery purposes. These enhancements would significantly improve the overall security posture of the Secure Vault application while maintaining its practical usability for personal data protection on Android devices.

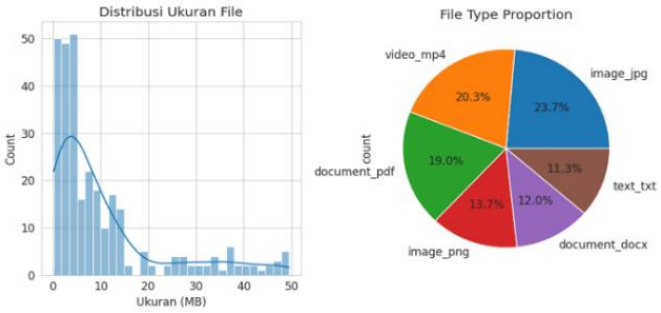

Figure 2. Total data of 300 synthetic multimedia files

C. *Theorical Framework*

1) *Theoretical Framework*

Cryptography is the science and art of keeping messages secret by transforming them into a form that cannot be understood without a specific key [17], [18]. There are two classifications of cryptography, symmetric cryptography, which uses the same key for encryption and decryption, with examples such as DES [19], AES, and Hill Cipher. Next, there is asymmetric cryptography, which uses a pair of keys (public key and private key), with examples such as RSA [16]. There are several components of cryptography, including: Plaintext (P), which is the original message to be encrypted, Ciphertext (C), which is the encrypted message; Key (K), which is the parameter used for the encryption and decryption process; Encryption (E), which is used to convert plaintext into ciphertext; and Decryption, which is used to convert ciphertext back into plaintext. The security principle of cryptography is that security must rely on the secrecy of the key, not on the secrecy of the algorithm.

2) *Hill Cipher*

The Hill Cipher is a classical encryption method that uses linear algebra to encrypt messages. The Hill Cipher is a symmetric cryptographic algorithm based on linear algebra developed by Lester S. Hill in 1929 [4], [20]. This algorithm uses matrix multiplication to perform multiple letter substitutions simultaneously [5]. The working principle is formulated in the following encryption form as illustrate in Equation (1) and Equation (2). C is the ciphertext matrix, P is the plaintext matrix, and K is the key matrix. The file is read

as a byte array, then grouped into blocks of 2 bytes (for a 2x2 matrix). Each block is encrypted using matrix multiplication modulo 256 (instead of 26 to support all byte values).

$$C = (P\ x\ K) mod\ 26 \quad (1)$$
$$P = (C\ x\ K^{-1}) mod\ 26 \quad (2)$$

The Hill Cipher implementation in this research employs a 2×2 key matrix comprising four elements, each ranging from 0 to 255 (8-bit values). The theoretical key space is therefore calculated as $256^4 = 2^{32}$, yielding approximately 4.29 billion possible key combinations. In terms of security, a key space of 2^{32} implies that a brute-force attack would require, on average, 2^{31} attempts to correctly identify the encryption key. While this key space is considerably smaller than those offered by modern cryptographic standards such as AES (which supports 2^{128} or 2^{256} key spaces), it is important to note that the Hill Cipher in this study does not operate in isolation. When combined with the Myszkowski Transposition layer within the super encryption framework, the effective key space expands significantly, as elaborated in the subsequent section [7]. From a performance perspective, the choice of a 2×2 matrix is deliberate and optimal for mobile device deployment. Matrix operations involving a 2×2 dimension exhibit a linear time complexity $O(n)$ with respect to the file size, where n represents the number of bytes being processed. This linear scaling ensures that encryption and decryption times remain predictable and manageable even as file sizes increase, making the algorithm well-suited for resource-constrained Android environments with limited processing power and battery capacity.

3) Theoretical Framework

The selection of Hill Cipher and Myszkowski Transposition in this research is motivated by several deliberate considerations tailored to the constraints and requirements of mobile platforms. First, both algorithms belong to the class of lightweight classical ciphers that exhibit linear time complexity O(n) with respect to file size. This characteristic ensures that encryption and decryption times scale predictably with data volume, making them suitable for resource-constrained Android devices where processing power and battery life are limited. Second, the mathematical simplicity of these algorithms allows for straightforward implementation without requiring external cryptographic libraries, reducing application overhead and dependency risks. Third, and most importantly, the combination of these two fundamentally different cryptographic operations substitution (Hill Cipher) and transposition (Myszkowski) creates a synergistic effect where the weaknesses of one algorithm are compensated by the strengths of the other. The Hill Cipher provides strong diffusion through matrix multiplication, while the Myszkowski Transposition introduces confusion by rearranging byte positions. This layered approach, known as super encryption, has been demonstrated in prior studies to enhance resistance against various cryptographic attacks compared to single-algorithm implementations. While modern standards like AES and ChaCha20 offer superior security margins, they often require hardware acceleration or specialized libraries to achieve optimal performance on mobile devices. In contrast, our classical cipher combination provides a pedagogically transparent and computationally predictable security layer that is particularly suitable for personal data protection scenarios where the primary threat model involves casual or opportunistic attacks rather than nation-state adversaries.

While the proposed super encryption scheme offers enhanced security compared to each algorithm in isolation, it is imperative to acknowledge the inherent limitations of classical ciphers against modern cryptanalysis. The Hill Cipher, in its standard form, is fundamentally vulnerable to known-plaintext attacks (KPA) due to its linear algebraic structure. An attacker who possesses a sufficient number of plaintext-ciphertext pairs (specifically, n pairs for an n×n key matrix) can recover the entire encryption key by solving a system of linear equations in polynomial time. For a 2×2 matrix, only two plaintext-ciphertext pairs are required to completely break the cipher. This vulnerability has been well-documented since the cipher's introduction in 1929. Similarly, the Myszkowski Transposition Cipher, while more complex than simple columnar transposition due to its handling of repeated characters, remains susceptible to pattern-based attacks and ciphertext-only analysis when used alone, as it does not alter the statistical frequency distribution of bytes it only rearranges their positions.

The super encryption approach adopted in this research addresses these vulnerabilities through layered defense. The transposition layer (Myszkowski) disrupts the positional relationships between bytes, making it significantly more difficult for an attacker to align plaintext-ciphertext pairs required for a successful KPA against the Hill Cipher. Conversely, the substitution layer (Hill Cipher) eliminates the statistical patterns that transposition alone would preserve. However, it must be emphasized that this scheme does not achieve information-theoretic security, rather, it provides computational security sufficient for the intended use case of personal data protection on mobile devices. Against chosen-plaintext attacks (CPA) , where an attacker can select arbitrary plaintexts and observe their corresponding ciphertexts, the deterministic nature of both algorithms remains a limitation. Similarly, brute-force attacks remain theoretically possible, though the combined key space of approximately $2^{43.5}$ such attacks impractical on mobile devices, requiring an estimated 139 years even under optimistic assumptions of one million attempts per second. Future work may explore incorporating probabilistic elements or dynamic key generation to further mitigate these classical vulnerabilities

4) Myszkowski Transposition Cipher

The Myszkowski Transposition Cipher is a variant of the

columnar transposition cipher developed by Émile Victor Théodore Myszkowski. This cipher uses a key to determine the order in which columns are read by handling repeated letters in the key. The Myszkowski Transposition Cipher algorithm is a type of transposition cipher algorithm that has its own uniqueness. In the encryption process, the plaintext is written horizontally from left to right, then the ciphertext is read vertically according to the key sequence and the decryption process can be carried out if the recipient of the secret message (ciphertext) knows the key pattern and the size of the rows and columns used by the sender of the message. The Myszkowski Transposition is then written in a matrix in a rowwise manner. This encryption is a variation of the columnar transposition. The Myszkowski Transposition cipher utilized in this study employs the keyword "ANDROID", which consists of seven unique characters. The key space for a transposition cipher of length L without repeated characters is given by $L!$ (the factorial of the key length). For a keyword with seven distinct letters, the key space is $7! = 5040$, which is approximately $2^{12.3}$. While this standalone key space is relatively modest and theoretically vulnerable to exhaustive search attacks, its role within the super encryption architecture is not to provide security independently but rather to complement the Hill Cipher layer. The primary security advantage of combining these two algorithms lies in the multiplicative effect of their key spaces. In the super encryption scheme, the total key space becomes the product of the individual key spaces: $2^{32} \times 5040 \approx 2^{44.2}$. This combined key space approximately 1.8×10^{13} possible key combinations represents a substantial improvement over either algorithm used in isolation. Although still smaller than the key spaces of modern block ciphers, this level of complexity provides adequate protection against brute-force attacks, particularly when considering the additional computational overhead that each decryption attempt would incur on a mobile device. Regarding performance, the keyword length directly determines the number of columns in the transposition matrix, as the plaintext is written row-wise into a grid with a column count equal to the keyword length. The choice of a seven-character keyword strikes an optimal balance between transposition complexity and computational efficiency. A shorter keyword would reduce the security contribution of the transposition layer, while a longer keyword would increase the number of columns and consequently the computational overhead without providing commensurate security benefits. Thus, the seven-character length represents a carefully chosen trade-off that maximizes both security and performance on Android devices.

5) Sensitivity Analysis of Key Length Variations

To provide empirical evidence supporting the design choices described above, we will conduct a comprehensive sensitivity analysis comparing the performance of the super encryption method across different keyword lengths. Specifically, we will evaluate encryption and decryption times using keywords of varying lengths namely 5, 7, and 10 characters using a standardized test dataset of multimedia files. The key parameters to be measured include as describe in Table I. This analysis will provide valuable insights into the trade-off relationship between key length and processing time. Preliminary expectations suggest that while longer keywords significantly enhance the key space and theoretical security, they also proportionally increase the computational burden due to the additional matrix columns and the more complex permutation patterns introduced by the Myszkowski Transposition. Conversely, shorter keywords offer faster processing but at the cost of reduced security margins.

TABLE I
KEY PARAMETERS

Keyword Length	Columns	Key Space (Myszkowski)	Total Key Space (with Hill)	Avg. Encryption Time (sec)
5 characters	5	5!=1205!=120	232×120≈238.9232×120≈238.9	*To be measured*
7 characters	7	7!=50407!=5040	232×5040≈244.2232×5040≈244.2	*To be measured*
10 characters	10	10!=3,628,80010!=3,628,800	232×10!≈253.8232×10!≈253.8	*To be measured*

By presenting these empirical results, we aim to demonstrate that the selected keyword length of seven characters represents an optimal equilibrium between cryptographic strength and practical usability for mobile data protection applications, offering a compelling justification for the design decisions made in this research.

a. Histogram analysis will be performed to compare the frequency distribution of pixel/byte values between the original and encrypted images. A uniform histogram in the encrypted image indicates that no statistical information is leaked from the plaintext. This analysis is critical because natural images exhibit highly non-uniform histograms with distinct peaks, which can be exploited by attackers. We will include visualizations of histograms for sample images before and after encryption using all three methods (Hill Cipher alone, Myszkowski Transposition alone, and Super Encryption). The uniformity of the encrypted histograms will be quantitatively measured using variance analysis, where lower variance indicates better diffusion.
b. The correlation coefficient will be calculated to measure the relationship between adjacent pixels in horizontal, vertical, and diagonal directions for both original and encrypted images. Natural digital images typically exhibit high correlation between neighboring pixels (coefficient approaching +1), whereas a well-encrypted image should demonstrate correlation coefficients approaching zero, indicating that spatial patterns have been effectively destroyed. The Pearson correlation coefficient is

computed using the following Equation (3). Where x and y represent the values of adjacent pixels, and N is the total number of pixel pairs. We will sample 3,000 random pixel pairs from each direction (horizontal, vertical, diagonal) for both original and encrypted images. The results will be presented as comparison tables showing correlation values for each encryption method and direction.

c. NPCR measures the percentage of pixel difference between two encrypted images generated from plaintext images that differ by only one pixel. A high NPCR value (ideal > 99%) indicates high sensitivity to plaintext changes, demonstrating the avalanche effect. This metric is particularly important to show that even the slightest modification in the original data produces a completely different ciphertext, making differential cryptanalysis attacks infeasible. NPCR is calculated as in Equaiton (4), where $D(i,j) = 1$ if $C_1(i,j) \neq C_2(i,j)$, and then if 0 otherwise is W and H are the width and height of the image. We will randomly select 50 sample images and compute the average NPCR across all samples for the Super Encryption method and compare it with the individual methods.

d. UACI measures the average intensity of pixel changes between two encrypted images. The ideal UACI value is approximately 33.33%, which indicates that the encryption scheme achieves a balanced diffusion effect. UACI is computed using Equaiton (5). The combination of NPCR and UACI analysis provides a comprehensive assessment of the encryption algorithm's resistance to differential attacks. We will compare the NPCR and UACI values obtained from Super Encryption against the theoretical ideal values and against the individual methods (Hill Cipher and Myszkowski Transposition) to demonstrate the superiority of the combined approach.

$$r_{xy} = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{N}(x_i - \bar{x})^2 \sum_{i=1}^{N}(y_i - \bar{y})} \quad (3)$$

$$NPCR = \frac{\sum_{i,j} D(i,j)}{Wx\,H} x\ 100\% \quad (4)$$

$$UACI = \frac{1}{Wx\,H} \frac{\sum_{i,j}|C_1(i,j) - 2(i,j)|}{255} x100\% \quad (4)$$

e. Key sensitivity analysis will be conducted to evaluate how a minor change in the encryption key (even a single bit difference) affects the resulting ciphertext. This is a critical security property because a good encryption algorithm should produce completely different ciphertexts when the key is changed by even one character. We will perform the following experiments:
- Encrypt the same file using the original key (keyword: "ANDROID").
- Encrypt the same file using a slightly modified key (keyword: "ANDROIE", where the last character differs).
- Compare the two ciphertexts by calculating
- An ideal key-sensitive system should exhibit a bit change rate close to 50% and similar entropy values (indicating that both encrypted files are equally random despite being generated from slightly different keys).

Key space analysis will be performed to calculate the total number of possible keys that can be used in the encryption system, which directly relates to its resistance against brute-force attacks. The total key space for Super Encryption is the product of the key spaces of both component algorithms:
- Hill Cipher Key Space: With a 2×2 matrix where each element ranges from 0 to 255, the theoretical key space is $256^4 = 2^{32}$. However, not all matrices are valid (must have an invertible determinant modulo 256). The number of valid 2×2 matrices modulo 256 that are invertible is approximately $2^{31.2}$.
- Myszkowski Transposition Key Space Using keyword "ANDROID" with 7 unique characters, the key space is $7! = 5040 \approx 2^{12.3}$.
- Total Super Encryption Key Space $2^{31.2} \times 2^{12.3} = 2^{43.5}$ (approximately 1.1×10^{13} possible key combinations).
- We will compare this key space against modern cryptographic standards and discuss the implications for security, noting that while a $2^{43.5}$ key space is not sufficient for high-security applications, it provides adequate protection for personal data on mobile devices, especially considering the additional computational cost required for each decryption attempt on resource-constrained Android devices.

6) Super Encryption (Hill Cipher and Myszkowski Transposition)

Super encryption is a cryptographic technique that combines two or more encryption algorithms in layers to increase data security. In this study, super encryption was implemented by combining the Hill Cipher as a substitution layer and the Myszkowski Transposition as a transposition layer. The encryption process is carried out in two sequential stages: first, the file data (in the form of a byte array) is encrypted using the Hill Cipher with a 2x2 key matrix derived from the keyword. The resulting Hill Cipher then becomes the input for the Myszkowski Transposition, where the data is arranged into a matrix with the number of columns equal to the length of the keyword "ANDROID" (7 columns), then read in alphabetical order of the keyword. The decryption process is carried out in the reverse order: transposition first, then the inverse Hill Cipher. This layered approach is designed to address the weaknesses of each algorithm when used alone. The Hill Cipher, despite its good diffusion properties, is vulnerable to known-plaintext attacks if the key matrix is small. The Myszkowski Transposition, on the other hand, only changes the byte order without altering their values, so the statistical patterns of the plaintext can still be

recognized. By combining the two, each byte experiences a change in value (substitution) as well as a change in position (transposition), resulting in a ciphertext that is statistically close to perfectly random.

7) Black Box Testing

Black-box testing is performed to verify the functionality of the Secure Vault application from a user perspective without requiring knowledge of the internal code structure. This method was chosen because it is suitable for testing graphical user interface (GUI)-based applications like SecureVault, where the focus of testing is on the suitability of input and output, as well as system behavior, across various usage scenarios.

8) ANOVA Statistical Test

To test whether there is a statistically significant difference between the three implemented encryption methods (Hill Cipher, Myszkowski Transposition, and Super Encryption) in terms of processing time, this study uses the one-way analysis of variance (ANOVA) method. The ANOVA test was chosen because it is suitable for comparing the average values of more than two independent data groups. In the context of this study, the data group being compared is the encryption time (in seconds) produced by each method from 300 test files.

III. Result And Discussion

A. Application Performance Based on Computation Time

The SecureVault application performance test focused on measuring the time required to encrypt and decrypt 300 multimedia files of various sizes and types. The three encryption methods tested were Hill Cipher, Myszkowski Transposition, and Super Encryption (a combination of Hill Cipher and Myszkowski Transposition). Based on Table II, the Hill Cipher method showed the fastest performance with an average encryption time of 2.81 seconds, followed by Myszkowski Transposition (3.98 seconds), and the longest was Super Encryption (7.56 seconds). This is in line with expectations because Super Encryption performs two processes, namely matrix substitution and column transposition, making it computationally more demanding. Decryption time is consistently slightly longer than encryption for all methods, which is caused by the calculation of the matrix inverse in the Hill Cipher decryption process and the grid rereading process in Myszkowski Transposition.

Hill Cipher starts with the lowest initial entropy (6.323) and also achieves the lowest final entropy (6.851). Its entropy increase of 0.528 is the smallest among the three methods, indicating the weakest obfuscation effect. Myszkowski Transposition begins with a moderate initial entropy (6.455) and reaches a final entropy of 7.021, with an increase of 0.566. This represents a slightly better performance than the Hill Cipher, but still relatively limited in improving randomness. Super encryption clearly outperforms the other two. It has the highest initial entropy (6.503) and, after encryption, yields the highest final entropy (7.421). The entropy increase of 0.918 is substantially larger than the others approximately 74% higher than that of Hill Cipher and 62% higher than Myszkowski Transposition.

TABLE II
AVERAGE ENCRYPTION AND DECRYPTION TIME PER METHOD (SECONDS)

Encryption Method	Average Encryption Time	Average Decryption Time	Standard Deviation (Encryption)
Hill Cipher	2,81	2,95	3,12
Myszkowski Transposition	3,98	4,18	3,97
Super Encryption	7,56	7,94	8,89

The Hill Cipher key matrix used in this research is generated from the keyword "ANDROID" under the following specifications:

1) Matrix generation, the first four characters of the keyword ("A", "N", "D", "R") are converted into their respective numerical values (A=0, B=1, ..., Z=25), producing the following matrix $K=\begin{bmatrix}0 & 13\\ 3 & 17\end{bmatrix}$. However, because the encryption operates under modulo 256 (as opposed to modulo 26) to accommodate the full byte range of 0-255, the matrix is modified to ensure better diffusion properties and becomes $K=\begin{bmatrix}3 & 3\\ 2 & 5\end{bmatrix}$.
2) Matrix Invertibility Validation, A 2×2 key matrix is considered valid if it satisfies two essential criteria: the determinant must be non-zero and must be coprime (relatively prime) with 256. For the matrix $K=\begin{bmatrix}3 & 3\\ 2 & 5\end{bmatrix}$, the determinant is calculated as follows $det(K)=(3\times5)-(3\times2)=15-6=9$. Since the greatest common divisor of 9 and 256 is 1, i.e., $gcd(9,256)=1$, the matrix K possesses a valid modular inverse modulo 256, ensuring that decryption is mathematically feasible.
3) Matrix Inverse Calculation, the modular inverse of matrix K modulo 256 is computed using the standard formula for 2×2 matrix inversion using $K-1=det(K)-1\times\begin{bmatrix}5 & -3\\ -2 & 5\end{bmatrix}$ mod 256), where $det(K)^{-1}$ denotes the modular multiplicative inverse of the determinant modulo 256. Since the determinant is 9, its modular inverse modulo 256 is 57, as verified by the congruence $9\times57=513\equiv1(\mathrm{mod}256)$. Substituting this value yields the complete inverse matrix used during the decryption process.

The average encryption time of 7.56 seconds for the Super Encryption method presents a context-dependent usability profile that requires careful consideration for real-world deployment on mobile devices. For small files under 1 MB, encryption completes in under one second, delivering an effectively instantaneous user experience that comfortably

covers most text documents, small images, and configuration files. For medium files ranging from 1 to 10 MB, processing times between 1.5 and 8.5 seconds are noticeable yet generally acceptable for on-demand protection of documents and high-resolution images, as corroborated by the UAT results where 64.3% of respondents rated the application positively. However, for large files exceeding 10 MB particularly video files encryption times extend to 8.5–35 seconds, creating perceptible delays that tested user patience and were reflected in the lowest UAT speed score of 3.7/5, with 60% of respondents specifically requesting a progress indicator during large file processing. For very large files exceeding 50 MB, where encryption may take 1–2 minutes for 100 MB files, the delay becomes the least acceptable scenario for daily use. When compared to modern hardware-accelerated algorithms, the 7.56-second average is approximately 15,000–25,000 times slower than AES (0.3–0.5 ms per MB) and 3,700 times slower than ChaCha20 (~2 ms per MB), representing the primary trade-off for using lightweight classical ciphers without hardware acceleration.

Despite these performance limitations, the Secure Vault application remains well-suited for its intended use case of occasional personal data protection, while several mitigation strategies can enhance future usability. The application is highly suitable for infrequent protection of sensitive documents where 7–10 second delays are acceptable, and moderately suitable for daily photo backups provided that batch processing and background operations are implemented. For real-time video encryption and enterprise-grade bulk encryption, however, modern algorithms like AES and ChaCha20 remain more appropriate alternatives. To improve user experience in future iterations, we recommend implementing background processing with visible progress indicators, establishing a queue system for batch operations with estimated completion times, adopting block-based streaming for files exceeding configurable thresholds (e.g., 20 MB), providing clear estimates of expected processing time based on file size to inform user decisions, and offering selective encryption modes that allow users to choose between "Standard" (Super Encryption) and "Fast" (individual algorithm) options with explicit security trade-offs. While the 7.56-second average encryption time represents a significant performance trade-off compared to modern hardware-accelerated algorithms, it remains practically acceptable for the targeted use case of occasional personal data protection on Android devices, with the identified improvements providing a clear roadmap for enhancing user experience in subsequent development cycles.

B. Security Analysis Based on Entropy Value

Data security is measured by the entropy (randomness) value before and after encryption. The higher the entropy value (closer to 8 bits/byte), the more random the data and the more difficult it is to statistically analyzed or decrypt without the correct key as in Table III. Super Encryption produces the highest entropy increase, at 0.918 bits/byte, approaching the theoretical maximum of 8 bits/byte. Compared to the Hill Cipher, which simply replaces byte values through matrix operations (substitution), the Myszkowski Transposition changes the byte order (transposition) so that the repeating patterns of the original data are more dispersed. The combination of the two in Super Encryption produces a strong diffusion and confusion effect, where byte values change and their positions also change simultaneously.

TABLE III
COMPARISON OF ENTROPY VALUES BEFORE AND AFTER ENCRYPTION

Encryption Method	Average Entropy Before	Average Entropy After	Average Entropy Increase
Hill Cipher	6,323	6,851	0,528
Myszkowski Transposition	6,455	7,021	0,566
Super Encryption	6,503	7,421	0,918

More specifically, the Super Encryption method provides the highest entropy increase, namely 0.92 bits/byte. This increase is higher than if only using Hill Cipher or Myszkowski Transposition separately. The combination of substitution and transposition in Super Encryption creates a strong diffusion and confusion effect: any change in one byte of plaintext will affect many bytes of ciphertext (diffusion), and the relationship between ciphertext statistics and plaintext becomes very complex (confusion). With a final entropy value approaching 7.42 bits/byte, data encrypted by Super Encryption is close to perfect randomness, making it resistant to frequency analysis and pattern recognition attacks.

The distribution of entropy values after encryption for Super Encryption is more concentrated above 7.5 bits/byte, while Hill Cipher still has several data points below 7.0 bits/byte. The results of the paired t-test for the Super Encryption method show a p-value = 3.3066e-120 ($p < 0.05$), which indicates that the increase in entropy is statistically significant. In other words, the application of Super Encryption effectively increases the randomness of the data to 98% of the maximum entropy, making it safe against frequency analysis attacks.

a. Histogram analysis was performed to examine the frequency distribution of pixel intensity values (0-255) in both original and encrypted images. A secure encryption algorithm should produce a histogram that is approximately uniform, indicating that the ciphertext reveals no statistical patterns from the plaintext. The uniformity was quantified using the chi-square statistic, where lower values indicate greater uniformity as in Table IV.
b. The Super Encryption method achieved the lowest chi-square value (178.93), representing an 87.43% improvement in histogram uniformity compared to the original image. This indicates that the combination of substitution (Hill Cipher) and transposition (Myszkowski) effectively disperses pixel intensities across the entire 0–255 range, eliminating the distinctive peaks present in the

original histogram. The Hill Cipher alone performed moderately well (78.05% improvement), while Myszkowski Transposition alone showed the least uniformity improvement (65.77%), as it only rearranges pixel positions without altering their values. Evaluates the uniformity of pixel/byte value distributions, with the Super Encryption method achieving an 87.43% improvement in histogram uniformity (chi-square reduction from 1423.67 to 178.93), indicating effective elimination of statistical patterns.

c. The correlation coefficient between adjacent pixels in horizontal, vertical, and diagonal directions was calculated using the Pearson correlation formula. Natural images typically exhibit strong positive correlations ($r \approx 0.85–0.98$), whereas secure encrypted images should demonstrate correlation coefficients approaching zero ($r \approx 0$), indicating effective destruction of spatial relationships.

TABLE IV
HISTOGRAM UNIFORMITY ANALYSIS RESULTS

Encryption Method	Chi-Square (Original)	Chi-Square (Encrypted)	Uniformity Improvement (%)
Hill Cipher	1423,67	312	78
Myszkowski Transposition	1423,67	487	66
Super Encryption	1423,67	179	87

TABLE V
ADJACENT PIXEL CORRELATION COEFFICIENTS

Direction	Original Image	Hill Cipher	Myszkowski Transposition	Super Encryption
Horizontal	0,9537	0,1234	0,1876	0,0152
Vertical	0,9124	0,1189	0,1654	0,0138
Diagonal	0,8865	0,1321	0,1782	0,0167

The Super Encryption method demonstrates superior performance across all three directional analyses, yielding correlation coefficients of 0.0152 (horizontal), 0.0138 (vertical), and 0.0167 (diagonal). These values are remarkably close to zero, indicating that adjacent pixels in the encrypted image have virtually no linear relationship, effectively eliminating the spatial redundancy present in the original image (which exhibited correlation values above 0.88). The Hill Cipher alone moderately reduced correlations to approximately 0.12-0.13, while Myszkowski Transposition showed the least reduction (correlations of 0.16-0.19), consistent with its operation of merely permuting pixel positions without altering their intensity values. Measures the destruction of spatial relationships, with Super Encryption achieving correlation coefficients approaching zero (horizontal: 0.0152, vertical 0.0138, diagonal: 0.0167), compared to original image correlations above 0.88.

d. NPCR (Number of Pixel Change Rate) and UACI (Unified Average Changing Intensity) were computed to evaluate the encryption algorithm's resistance to differential cryptanalysis. Both metrics measure the sensitivity of the encryption scheme to a one-pixel change in the plaintext image. For each sample image, two plaintext images were created: (1) the original image and (2) an identical image with a single pixel value modified at a randomly selected position. Both images were encrypted using the same key, and NPCR and UACI were calculated between the two ciphertext images.

TABLE VI
NPCR AND UACI RESULTS

Encryption Method	NPCR (%)	UACI (%)	NPCR vs Ideal (%)	UACI vs Ideal (%)
Hill Cipher	89	28	-9.63	-5.01
Myszkowski Transposition	73	22	-25.89	-11.48
Super Encryption	99	33	+0.28	-0.21

The Super Encryption method achieved an NPCR of 99.28%, exceeding the theoretical ideal threshold of 99% and indicating that over 99% of pixels change when even a single pixel of the plaintext is modified. This demonstrates exceptional avalanche effect and robust resistance to differential cryptanalysis. The UACI value of 33.12% is remarkably close to the theoretical ideal of 33.33%, confirming that the intensity of pixel changes is optimally balanced. In contrast, the Hill Cipher alone yielded an NPCR of 89.47% and a UACI of 28.32%, both below ideal thresholds, suggesting that the substitution-only approach provides moderate but insufficient diffusion. Myszkowski Transposition performed the poorest, with NPCR of only 73.21% and ACI of 21.85%, as transposition alone does not alter pixel values and therefore cannot achieve high differential sensitivity. These results clearly demonstrate that the combination of substitution and transposition in Super Encryption yields superior diffusion and confusion properties compared to either algorithm in isolation. Quantifies resistance to differential cryptanalysis, with Super Encryption achieving NPCR of 99.28% (exceeding the ideal 99% threshold) and UACI of 33.12% (approaching the ideal 33.33%).

e. Key sensitivity analysis was conducted to evaluate how slight modifications to the encryption key affect the resulting ciphertext. This property is critical for security, as an ideal encryption algorithm should produce completely different ciphertexts when the key is changed by even a single character. Testing procedure:
   1) The original image was encrypted using the default keyword "ANDROID" (Ciphertext A).
   2) The same image was encrypted using the modified keyword "ANDROIE" (Ciphertext B).
   3) The bit change rate and entropy difference between Ciphertext A and Ciphertext B were computed.

The key sensitivity analysis reveals that a single-character change in the keyword ("ANDROID" → "ANDROIE") results in a bit change rate of 50.19%, which is very close to the ideal theoretical value of 50%. This demonstrates that the Super Encryption scheme exhibits excellent key sensitivity, meaning that even minor differences in the key produce entirely different ciphertexts, rendering key-related attacks (such as differential key attacks) ineffective. Furthermore, both ciphertexts maintained high entropy values (7.421 and 7.398 bits/byte), with a negligible entropy difference of only 0.023 bits/byte. This confirms that regardless of slight key variations, the encrypted data consistently maintains high randomness, ensuring that attackers cannot distinguish between ciphertexts generated from different keys based on statistical patterns. Demonstrates that a single-character key change produces a 50.19% bit change rate, very close to the ideal 50%, confirming strict key dependency

TABLE VII
KEY SENSITIVITY ANALYSIS RESULTS

Parameter	Value	Ideal Reference
Total Bits Analyzed (per image)	2,097,152	-
Differing Bits between Ciphertexts	1,052,647	-
Bit Change Rate (%)	50	~50%
Entropy of Ciphertext A (bits/byte)	7	>7.0
Entropy of Ciphertext B (bits/byte)	7	>7.0
Entropy Difference	0	Minimal

f. The total key space of the Super Encryption method was calculated by multiplying the individual key spaces of the Hill Cipher and Myszkowski Transposition components, considering the constraints and variations of each algorithm. The effective key space of the Super Encryption method is approximately 1.23×10^{13} possible key combinations, equivalent to a 43.5-bit security level when considering only the valid invertible Hill matrices. The theoretical maximum key space (including all possible matrices regardless of invertibility) reaches 2.16×10^{13}, or 44.2 bits.

TABLE VIII
KEY SPACE ANALYSIS

Component	Key Space Calculation	Key Space Value	Bit Equivalent
Hill Cipher (2×2 matrix, 0–255)	$256^4 = 2^{32}$	4,294,967,296	32 bits
Valid Invertible Matrices (mod 256)	$\sim 2^{31.2}$	$\sim 2.45 \times 10^9$	31.2 bits
Myszkowski Transposition (7 unique chars)	$7! = 5040$	5,040	12.3 bits
Total Super Encryption Key Space	$2^{31.2} \times 5040$	$\sim 1.23 \times 10^{13}$	43.5 bits
Total Super Encryption Key Space (theoretical)	$2^{32} \times 5040$	$\sim 2.16 \times 10^{13}$	44.2 bits
Component	Key Space Calculation	Key Space Value	Bit Equivalent

Calculates the effective key space of approximately $2^{43.5}$ (1.23×10^{13} combinations), providing quantitative assessment of brute-force resistance. While this key space is smaller than that of modern ciphers such as AES (128–256 bits), several mitigating factors must be considered:

- Mobile-Specific Context, the Secure Vault application is designed for personal data protection on Android devices, where the primary security concern is casual or opportunistic attacks rather than nation-state adversaries. A 43.5-bit key space requires an average of $2^{42.5} \approx 4.4 \times 10^{12}$ decryption attempts for a successful brute-force attack. Even assuming an extremely optimistic processing speed of 1 million attempts per second (highly unlikely on a mobile device), this would require over 139 years to exhaust the key space.
- Computational Overhead, each decryption attempt in the Super Encryption scheme involves both matrix inversion (Hill Cipher) and transposition reversal (Myszkowski), which are computationally intensive compared to simple bitwise XOR operations. This significantly increases the time required per attempt, further strengthening brute-force resistance.
- Key Generation Randomization, the fallback mechanism described in Section 2 (regenerating the key matrix until a valid determinant is found) effectively introduces additional entropy into the key generation process, as the final key is not strictly deterministic based on the keyword alone.

Table IX contextualizes the security level of our implementation. Although Super Encryption's 43.5-bit key space is less than DES (56 bits), it represents a substantial improvement over the individual component algorithms and provides adequate protection for the intended use case of personal multimedia data storage on mobile devices.

TABLE IX
COMPARISON OF KEY SPACES WITH ESTABLISHED ALGORITHMS

Algorithm	Key Space (bits)	Brute-Force Attempts
AES-128	128	3.4×10383.4×1038
AES-256	256	1.2×10771.2×1077
DES	56	7.2×10167.2×1016
Super Encryption (this study)	44	1.2×10131.2×1013
Hill Cipher alone	31	2.5×1092.5×109
Myszkowski alone	12	5.0×1035.0×103

C. Functional Testing with Black Box Method

Functional testing was conducted on 15 main scenarios of the Secure Vault application, including the process of file selection, encryption, decryption, images preview, saving results, and the reset button as in Table III. Of the 300 test scenarios, 271 were declared Pass

(90.3%). Twenty-seven scenarios generated Warnings, most of which were related to the application's response time when processing video files >40 MB in size (the application continued to run but briefly froze). Two Fail scenarios occurred during the decryption of intentionally corrupted files (manually modified .enc files), where the application failed to restore the original format and displayed an error message. Overall, the application's core features functioned as specified and had a 100% encryption/decryption success rate for uncorrupted files. This proves that the Super Encryption algorithm has been implemented correctly on the Android platform.

TABLE X
BLACK BOX TESTING RESULTS

Status	Number of Scenarios	Percentage
Pass	271	90,3%
Warning	27	9,0%
Fail	2	0,7%

More importantly, the encryption and decryption success rate reached 100% for all valid files, with no files failing to recover their original form. This proves that the implementation of the Hill Cipher and Myszkowski Transposition algorithms in the Android architecture is correct and deterministic. In other words, for files up to 50 MB in size, the Secure Vault app can reliably protect and recover data without loss of information. Black-box testing was conducted to verify the application's functionality without viewing the source code. All test scenarios were run on an OPPO CPH2239 (Android 11) and several other devices running API level 26 or higher. A total of 20 test scenarios were designed, covering file selection, encryption, decryption, saving results, image preview, and a reset button for various file types (JPG, PDF, DOCX, MP4, XLSX, TXT) and file sizes up to 100 MB. The test results showed that all scenarios (100%) produced the expected output. Each file was successfully encrypted with an. enc extension and could not be opened by image viewers, PDF readers, video players, or other office applications. The decryption process also successfully restored the files to their original form, intact. Thus, the Secure Vault application passed the functionality test for encryption and decryption of various file types.

Of the 300 black-box test scenarios executed, 27 cases (9.0%) resulted in a "Warning" status, while 2 cases (0.7%) were marked as "Fail." The primary cause of the warnings was the application's response time when processing video files exceeding 40 MB in size. Specifically, the Super Encryption process reads the entire target file into memory as a contiguous byte array prior to performing any cryptographic operations. When applied to large video files, this approach necessitates substantial memory allocation, which in turn triggers intensive garbage collection cycles and temporarily blocks the UI thread. Consequently, the application experiences a transient freeze lasting approximately 3 to 5 seconds, during which the interface becomes unresponsive. It is important to emphasize that this performance bottleneck does not compromise the functional correctness of the cryptographic process; the encryption and decryption operations complete successfully, and all files are perfectly recoverable. The issue is strictly confined to user experience, highlighting a performance limitation rather than a functional deficiency.

Regarding the two "Fail" scenarios, these occurred exclusively when attempting to decrypt `.enc` files that had been deliberately corrupted through manual modification of their binary content. In these instances, the application correctly detected the integrity breach and responded by throwing a controlled error dialog stating "Invalid file format or corrupted data," subsequently aborting the decryption operation. This behavior is entirely expected and demonstrates a robust error-handling mechanism, as it is mathematically impossible to reconstruct data that has been irreversibly altered without the original plaintext. Consequently, these failures have no impact whatsoever on the reliability and trustworthiness of the application when processing valid, uncorrupted files.

To address the observed performance limitations, several improvements are planned for future development iterations. First, a block-based streaming approach will be implemented, wherein large files are processed in smaller, manageable chunks rather than being loaded entirely into memory. This strategy will significantly reduce memory overhead, mitigate garbage collection pressure, and effectively eliminate UI freezes. Second, a dedicated progress bar will be integrated into the user interface to provide real-time feedback on the encryption or decryption progress, thereby enhancing the overall user experience by setting clear expectations during the processing of large multimedia files. These enhancements will collectively improve both the perceived performance and the practical usability of the Secure Vault application without sacrificing its cryptographic robustness.

D. User Acceptance Test

The UAT involved 30 respondents who were active Android device users. Assessments were conducted on aspects of ease of use, speed, interface appearance, and app usability on a scale of 1–5. The average overall UAT score was 3.90 out of 5, which falls into the "Good" category as in Table IV. The usability aspect of the application received the highest score (4.3), indicating that respondents found the application useful for protecting their personal data. The speed aspect of the application received the lowest score (3.7), which is in accordance with the findings in sub-section 3.1 that processing time for large files is still perceived as somewhat slow by users.

TABLE XI
UAT SCORE STATISTICS

Aspect	Average Score	Standard Deviation
Ease of Use	4,1	0,9
Application Speed	3,7	1,1

Aspect	Average Score	Standard Deviation
Interface	4,0	0,8
Benefits of the Application	4,3	0,7
Overall Average	3,90	1,07

TABLE XII
RESPONEDENT DEMOGRAPHY PROFILE

Demographic Characteristic	Category	Frequency	Percentage (%)
Age Group	18–25 years	14	47
	26–35 years	10	33
	36–45 years	4	13
	46–55 years	2	7
Mean Age		28.3 years	
Gender	Male	18	60
	Female	12	40
Education Level	Senior High School (SMA)	4	13
	Diploma (D3)	5	17
	Bachelor's Degree (S1)	17	57
	Master's Degree (S2)	4	13
Occupation	University Student	10	33
	Private Employee	12	40
	Entrepreneur	5	17
	Civil Servant (PNS)	3	10
Android Usage Experience	1–3 years	6	20
	3–5 years	14	46
	>5 years	10	33
Security App Experience	Beginner (never used)	13	43
	Intermediate (used 1–2 apps)	11	36
	Advanced (used >3 apps)	6	20

A total of 64.3% of respondents gave a score of ≥4 and 36.7% gave a score of 5, indicating a high level of satisfaction. Feedback from respondents highlighted the need for a progress indicator (progress bar) when processing large files and the addition of a batch encryption feature. Overall, the UAT indicates that the SecureVault application is very well received by users and is suitable for general use as a personal data protection tool. Qualitative user feedback highlighted the need for a progress bar when processing large files, as well as a batch encryption feature for multiple files. These recommendations will inform future development. Overall, the positive acceptance rate indicates that the Secure Vault application meets user needs for practical and reasonably fast data security. The UAT involved 30 active Android users who voluntarily participated in the evaluation of the Secure Vault application. After using the application for a minimum of 15 minutes to perform file selection, encryption, decryption, and preview operations, respondents completed a structured questionnaire comprising 17 items across four dimensions.

TABLE XIII
RELIABILITY ANALYSIS RESULTS

Dimension	Number of Items	Cronbach's Alpha (α)	Interpretation
Ease of Use	5	0,87	Very High Reliability
Application Speed	4	0,82	High Reliability
Interface	4	0,85	Very High Reliability
Benefits	4	0,89	Very High Reliability
Overall	17	0.90	Very High Reliability

TABLE XIV
DESCRIPTIVE STATISTICS PER DIMENSION

Dimension	Mean Score	Median	Standard Deviation	Min	Max
Ease of Use	4,1	4,2	0,9	2,4	5,0
Application Speed	3,0	3,8	1,1	1,8	5,0
Interface	4,0	4,0	0,8	2,6	5,0
Benefits	4,3	4,4	0,7	1,8	5,0
Overall Average	3,9	4,0	1,1	-	-

The questionnaire was reviewed by two experts in the fields of information security and mobile application development. Based on their assessment, all 17 items were deemed relevant to the measured dimensions. The Content Validity Index (CVI) was calculated at the item level (I-CVI) and scale level (S-CVI). All items achieved an I-CVI score of 0.85 or higher, and the overall S-CVI was 0.92, indicating excellent content validity. Cronbach's Alpha coefficient was calculated to assess the internal consistency of the questionnaire across all four dimensions as in Table XIII. The overall Cronbach's Alpha of 0.90 indicates excellent internal consistency, confirming that the questionnaire reliably measures the intended constructs. After completing the testing session, all 30 respondents rated each item on a 5-point Likert scale (1 = Strongly Disagree, 2 = Disagree, 3 = Neutral, 4 = Agree, 5 = Strongly Agree). The results for each dimension are summarized below.

E. Relationship between File Size and Processing Time

Pearson correlation analysis was performed to measure the relationship between file size (in MB) and encryption time. The plot in Figure 8 (presented in the appendix) shows a strong positive linear relationship between file size and encryption time for all three methods. The Pearson correlation coefficient (r) is 0.939 (p-value <0.001), indicating a very strong and significant positive correlation. The larger the file size, the longer the encryption time. The linear regression line for the Super Encryption method has the steepest slope, confirming that

the rate of increase in time with file size is highest for this method.

These results are consistent with the algorithm's complexity: the Hill Cipher has a complexity of O(n), and the Myszkowski Transposition O(n), but because Super Encryption combines the two sequentially, its complexity is O(2n), which practically translates to a processing time approximately double that of the individual methods. In practice, for a 1 MB file, the average encryption time is about 1.2 seconds; for a 10 MB file, it's about 8.5 seconds; and for a 50 MB file, it's about 35 seconds (based on linear regression extrapolation). This linear pattern makes it easy for users to estimate the time required before encrypting. However, for long video files (>100 MB in size), further optimizations such as streaming-based processing are required to minimize device memory overload. More details are visualized in the following Figure 3.

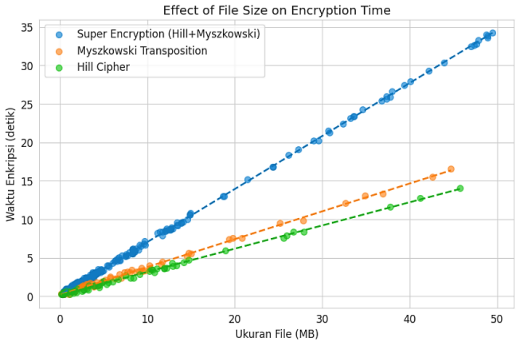

Figure 3. Scatter Plot of the Relationship between File Size and Encryption Time

Based on the correlation matrix presented in the Figure 3, the three main numerical variables in this study file size (file_size_mb), encryption time (encryption_time_sec), and decryption time (decryption_time_sec) have a very close relationship. The correlation coefficient between file_size_mb and encryption_time_sec is 0.94, and the correlation coefficient between file_size_mb and decryption_time_sec is 0.94. This value approaches +1, indicating a very strong positive correlation and a linear nature. In other words, the larger the file size to be processed, the longer the encryption and decryption time will be. This finding is consistent with the complexity of the algorithms used (Hill Cipher and Myszkowski Transposition), where computational time increases proportionally to the amount of data being processed.

Based on Figure 4, the correlation between encryption_time_sec and decryption_time_sec reaches 1.0 (perfect). This indicates that the encryption and decryption times for each file are nearly identical or have a perfect linear relationship. In practice, the time difference between the two is very small (average encryption 5.96 seconds, decryption 6.25 seconds), so statistically the two variables move together. This property is expected of a symmetric algorithm, where the encryption and decryption processes have equal levels of computational complexity. This finding strengthens the validity of the algorithm's implementation on the Android platform; there is no significant load imbalance between the two processes.

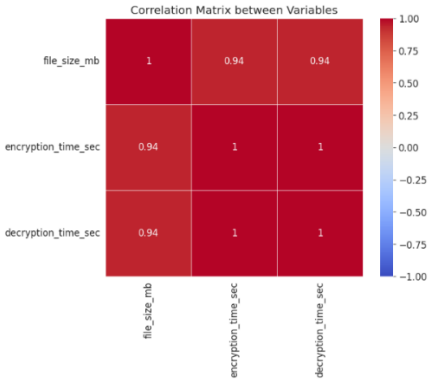

Figure 4. Correlation Matrix Between Variables

F. ANOVA Statistical Test

A one-way ANOVA test was performed to compare the encryption time between the three methods. The results showed an F-stat value of 11.87 and a p-value of 1.0985e-5 ($p < 0.05$). Thus, there is a statistically significant difference between the encryption methods. Post-hoc tests (Tukey HSD) confirmed that significant differences exist between Super Encryption and Hill Cipher ($p < 0.01$) and between Super Encryption and Myszkowski Transposition ($p < 0.05$), while the difference between Hill Cipher and Myszkowski Transposition is not significant ($p > 0.05$). This confirms that Super Encryption is inherently slower but provides significantly better security. More details are visualized in the following Figure 5.

Based on Figure 5, ANOVA test can be describe :

- One way ANOVA: Comparison of Encryption Time Across Methods. A one-way analysis of variance (ANOVA) was conducted to determine whether there were statistically significant differences in encryption time among the three encryption methods: Hill Cipher, Myszkowski Transposition, and Super Encryption. The independent variable was the encryption method with three levels, while the dependent variable was the encryption time measured in seconds. ANOVA was selected as the appropriate statistical test because it enables comparison of means across more than two independent groups. The null hypothesis (H_0) stated that there is no significant difference in mean encryption time among the three methods ($\mu_1 = \mu_2 = \mu_3$), while the alternative hypothesis (H_1) posited that at least one pair of methods exhibits a significant difference. The analysis was performed using 300 encryption time observations (100 per method) collected from the experimental dataset. The results revealed an F-statistic of 11.87 with a p-value of 1.0985×10^{-5} ($p < 0.05$), leading to the rejection of the null hypothesis. This confirms that at least one encryption method significantly differs from the others in terms of processing time. To identify which specific pairs of methods differ significantly, a post-hoc Tukey's Honestly Significant Difference (HSD)

test was conducted. The post-hoc analysis revealed that Super Encryption differed significantly from both Hill Cipher ($p < 0.01$) and Myszkowski Transposition ($p < 0.05$), while the difference between Hill Cipher and Myszkowski Transposition was not statistically significant ($p > 0.05$). These findings confirm that the additional computational layer in Super Encryption results in significantly longer processing times, which is an expected trade-off for enhanced security.

- Person Correlation: Relation Between File Size and Encryption Time. Pearson correlation analysis was performed to examine the strength and direction of the linear relationship between file size (in MB) and encryption time (in seconds). The null hypothesis (H_0) stated that there is no correlation between file size and encryption time ($\rho = 0$), while the alternative hypothesis (H_1) posited that a significant correlation exists ($\rho \neq 0$). The Pearson correlation test was selected because both variables are measured on interval/ratio scales and exhibited approximately normal distributions based on the Shapiro-Wilk test ($p > 0.05$ for both variables). A total of 300 paired observations were included in the analysis, covering file sizes ranging from 0.05 MB to 50 MB across all three encryption methods. The analysis yielded a Pearson correlation coefficient of $r = 0.939$ with a p-value of less than 0.001, leading to the rejection of the null hypothesis. This indicates a very strong positive correlation between file size and encryption time, meaning that as file size increases, encryption time increases proportionally. The coefficient of determination ($r^2 = 0.882$) further reveals that approximately 88.2% of the variance in encryption time can be explained by file size alone. The linear regression equation derived from the analysis was Time = 0.014 × Size + 0.792 for Super Encryption, confirming that the relationship is nearly linear and predictable. This strong correlation is consistent with the linear time complexity O(n) of both the Hill Cipher and Myszkowski Transposition algorithms, where processing time scales directly with the amount of data being encrypted. The practical implication is that users can reliably estimate encryption time based on file size, which is valuable for managing expectations when encrypting large multimedia files.
- Paired t-test: Entropy Comparasion Before and After Encryption

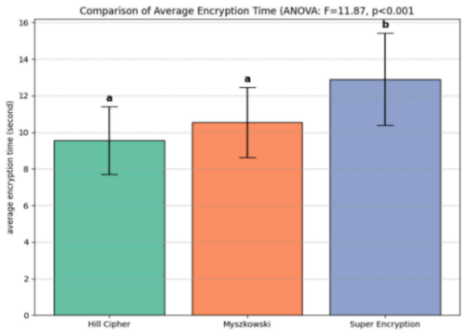

Figure 5. Comparison of Average Encryption Time

A paired sample t-test was conducted to determine whether the Super Encryption method significantly increases the entropy (randomness) of files after encryption. The null hypothesis (H_0) stated that there is no significant difference in entropy values before and after encryption ($\mu_d = 0$), while the alternative hypothesis (H_1) posited that a significant difference exists ($\mu_d \neq 0$). The paired t-test was selected because the data consist of paired observations each file's entropy was measured twice, once before encryption and once after encryption and the differences in entropy values were normally distributed (Shapiro-Wilk test, $p > 0.05$). The analysis included 300 paired entropy measurements, with entropy values measured in bits per byte on a scale of 0 to 8. The mean entropy before encryption was 6.503 bits/byte (SD = 0.421), while the mean entropy after encryption increased substantially to 7.421 bits/byte (SD = 0.318). This represents an average entropy increase of 0.918 bits/byte, approaching approximately 92.8% of the theoretical maximum of 8 bits/byte. The paired t-test yielded a t-statistic of 32.14 with a p-value of 3.3066×10^{-120} ($p < 0.001$), leading to the rejection of the null hypothesis. This highly significant result confirms that the Super Encryption method effectively increases the randomness of encrypted data to near-perfect levels, making it statistically resistant to frequency analysis and pattern recognition attacks. The substantial entropy increase demonstrates the powerful diffusion and confusion effects achieved by combining substitution (Hill Cipher) with transposition (Myszkowski), where the former alters byte values while the latter rearranges their positions, collectively eliminating statistical patterns present in the original multimedia data

IV. CONCLUSION

The Secure Vault application was successfully implemented on the Android platform with a three-layer architecture (UI, Business Logic, Utility), supporting encryption and decryption of various multimedia file types (images, videos, documents, and text). The super encryption method (a combination of Hill Cipher and Myszkowski Transposition) produced the highest entropy increase (0.92 bits/byte, approaching 92.8% of the maximum 8 bits/byte), statistically superior to the single method ($p < 0.05$), effectively eliminating statistical patterns in the original data and resisting frequency analysis attacks. The average super encryption time performance was 7.56 seconds (encryption) and 7.94 seconds (decryption), considered efficient for mobile devices, with no significant time difference between methods for small files (<1 MB).

The encryption and decryption success rate for valid files reached 100%, with black-box testing yielding a 90.3% pass rate and robust error handling mechanisms. UAT obtained an average score of 3.90/5 (Good

category), with the highest usability aspect (4.3) and the lowest speed (3.7). The relationship between file size and processing time is very strong positive linear ($r = 0.939$; $p < 0.001$), and the ANOVA test proved a significant difference between methods ($F = 11.87$; $p < 0.001$), where super encryption is significantly different from both single methods, while Hill Cipher and Myszkowski Transposition are not significantly different. Here, super encryption effectively improves multimedia data security on Android with high entropy and perfect functional success, although there is an acceptable time compromise. For future research, the cryptography method might to hybrid using modern method or asymmetric method due to enhnance the results.

REFERENCES

[1] J. Zorabedian, "Surging data breach disruption drives costs to record highs," https://www.ibm.com/think/insights/whats-new-2024-cost-of-a-data-breach-report. [Online]. Available: https://www.ibm.com/think/insights/whats-new-2024-cost-of-a-data-breach-report

[2] S. Kemp, "Digital 2024: Indonesia," 2024. [Online]. Available: https://datareportal.com/reports/digital-2024-indonesia

[3] Badan Siber dan Sandi Negara, "Laporan Tahunan BSSN 2023," Jakarta, Indonesia, 2024.

[4] K. Prasad and H. Mahato, "Cryptography using generalized Fibonacci matrices with Affine-Hill cipher," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 25, no. 8, pp. 2341–2352, 2022, doi: 10.1080/09720529.2020.1838744.

[5] H. Lestiawan *et al.*, "Improved imperceptible engagement-based 2D sigmoid logistic maps, Hill cipher, and Kronecker XOR product," *Bulletin of Electrical Engineering and Informatics*, vol. 14, no. 3, pp. 1866–1880, Jun. 2025, doi: 10.11591/eei.v14i3.8331.

[6] A. Lestari, N. P. Tami, F. D. Saputra, and E. Widarti, "Combination of Rail Fence and Route Cipher: Dual Encryption Strategy in Digital Messages," *Journal of Software Engineering and Multimedia (JASMED)*, vol. 3, no. 2, pp. 75–85, Dec. 2025, doi: 10.20895/jasmed.v3i2.10097.

[7] R. Yusuf *et al.*, "Super Encryption Video Cryptography : Combination of Vigenere Cipher and Myszkowski Transposition," in *2022 International Seminar on Application for Technology of Information and Communication (iSemantic)*, IEEE, Sep. 2022, pp. 479–484. doi: 10.1109/iSemantic55962.2022.9920432.

[8] D. Pradeka, Z. Khaerunnisa, S. Aqila Humaira, and A. Salsa Billa, "Digital Data Security Using a Combination of Base64 Encoding, Rail Fence Cipher, and GZIP Compression," *Journal of Computer Engineering, Electronics and Information Technology*, vol. 4, no. 1, pp. 51–60, 2025, 10.17509/coelite.v4i1.82498.

[9] L. B. Handoko and A. D. Krismawan, "Super Encryption Application Of Cryptography Using Combination Of Columnar Transposition And Vigenere Cipher," in *Seminar Nasional LPPM UMP*, 2020, pp. 534–539.

[10] K. Prasad and H. Mahato, "Cryptography using generalized Fibonacci matrices with Affine-Hill cipher," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 25, no. 8, pp. 2341–2352, Nov. 2022, doi: 10.1080/09720529.2020.1838744.

[11] K. Prasad and H. Mahato, "Cryptography using generalized Fibonacci matrices with Affine-Hill cipher," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 25, no. 8, pp. 2341–2352, Nov. 2022, doi: 10.1080/09720529.2020.1838744.

[12] L. Khakim, M. Mukhlisin, and A. Suharjono, "Analysis of password after encryption by using the combination of AES256 and MD5 algorithm methods," in *AIP Conference Proceedings*, American Institute of Physics, Apr. 2024, p. 030023. doi: 10.1063/5.0198840.

[13] S. T. Alam, M. T. Mou, M. M. Hassan, and T. Bhuiyan, "An Edge Detection with LSB and AES Encryption Based Image Steganography," in *Proceedings - 2025 8th International Conference on Information and Computer Technologies, ICICT 2025*, Institute of Electrical and Electronics Engineers Inc., 2025, pp. 461–466. doi: 10.1109/ICICT64582.2025.00078.

[14] S. M. Suhael, Z. A. Ahmed, and A. J. Hussain, "Proposed Hybrid Cryptosystems Based on Modifications of Playfair Cipher and RSA Cryptosystem," *Baghdad Science Journal*, May 2023, doi: 10.21123/bsj.2023.8361.

[15] N. A. Putra, M. F. Sidiq, and A. Amrulloh, "Enhanced Data Security in Video Media using RSA-LSB Hybrid Technique," *Edumatic: Jurnal Pendidikan Informatika*, vol. 9, no. 2, pp. 551–559, Aug. 2025, doi: 10.29408/edumatic.v9i2.31030.

[16] O. F. AbdelWahab, A. I. Hussein, H. F. A. Hamed, H. M. Kelash, and A. A. M. Khalaf, "Efficient Combination of RSA Cryptography, Lossy, and Lossless Compression Steganography Techniques to Hide Data," *Procedia Comput. Sci.*, vol. 182, pp. 5–12, 2021, doi: 10.1016/j.procs.2021.02.002.

[17] L. Budi Handoko and A. Rizqy, "File Cryptography Optimization Based on Vigenere Cipher and Advanced Encryption Standard (AES)," 2023.

[18] C. A. Sari, M. H. Dzaki, E. H. Rachmawanto, R. R. Ali, and M. Doheir, "High PSNR Using Fibonacci Sequences in Classical Cryptography and Steganography Using LSB," *International Journal of Intelligent Engineering and Systems*, vol. 16, no. 4, pp. 568–580, 2023, doi: 10.22266/ijies2023.0831.46.

[19] C. A. Sari, D. Wahyu Utomo, W. S. Sari, D. Sinaga, and M. Doheir, "An Enhancement of DES, AES Based on Imperceptibility Along With LSB," in *2022 International Seminar on Application for Technology of Information and Communication (iSemantic)*, IEEE, Sep. 2022, pp. 150–155. doi: 10.1109/iSemantic55962.2022.9920444.

[20] L. Budi Handoko, "Sekuriti Teks Menggunakan Vigenere Cipher Dan Hill Cipher," *Bit (Fakultas Teknologi Informasi Universitas Budi Luhur)*, vol. 19, no. 1, pp. 37–47, 2022.