Secure Data Handling: Performance Analysis of AES, Walsh Transform, and Hybrid Encryption

Svetlana Grigoryan[1], Sergo Episkoposian[2]
[1,2]National Polytechnic University of Armenia, Yerevan, Armenia
[1]e-mail: svetlanagrigoryanmt840@polytechnic.am
svetlanagrigoryan65@gmail.com
[2]e-mail: sergo.episkoposyan@polytechnic.am

***Abstract*—This study examines the performance of three encryption algorithms: the Walsh Transform, the Advanced Encryption Standard (AES), and a novel hybrid approach that integrates both techniques. The cryptographic strength of each method is assessed using key metrics such as entropy, avalanche effect and chi-square distribution. The evaluation focuses on the degree of randomness in the encrypted output, the sensitivity of the encryption to minor changes in the input, and the uniformity of the ciphertext distribution. The findings indicate that the hybrid AES+Walsh algorithm offers enhanced security characteristics, making it a promising solution for applications requiring robust data protection.**

***Keywords*—AES, Walsh transform, Hadamard matrix, encryption algorithms, data security, hybrid encryption.**

I. Introduction

Cryptography combines scientific principles and practical techniques to protect information by transforming it into unreadable formats for unauthorized parties. Among modern cryptographic tools, the Walsh Transform and hybrid encryption methods that integrate Walsh functions with established algorithms have recently attracted attention due to their potential to enhance security metrics such as diffusion and randomness.

This study focuses on evaluating the performance of three encryption techniques: the Walsh Transform, the Advanced Encryption Standard (AES), and a novel hybrid AES+Walsh algorithm. AES, standardized by the U.S. National Institute of Standards and Technology in 2001 as a successor to the Data Encryption Standard (DES), is a widely adopted block cipher operating on 128-bit data blocks with key sizes of 128, 192, or 256 bits. Its proven security and efficiency make AES a relevant benchmark for assessing new cryptographic approaches.

In our early works [1-5], we have observed text encryption topics with Walsh transform, RSA encryption, and combined the methods to get a hybrid W-RSA encryption algorithm. RSA has a plaintext length limitation, making it unsuitable for large data encryption. In contrast, the AES algorithm does not have limitations for the plaintext. By investigating these algorithms, this research aims to explore how the integration of Walsh functions with AES can improve cryptographic strength, measured through entropy, avalanche effect, and statistical tests, thereby contributing to the development of more robust encryption schemes.

II. Walsh Functions and Hadamard Matrix

Before introducing the Walsh function [6-9], we need to establish what the Rademacher function is.

Definition 1. The Rademacher system is defined as

$$r_0(x) = \begin{cases} 1, x \in \left[0, \frac{1}{2}\right), \\ -1, x \in \left(\frac{1}{2}, 1\right], \end{cases} \quad r_0(x+1) = r_0(x),\ r_k(x) = r_0(2^k x), \qquad k = 1,2,\ldots,$$

i.e., to find the r_k Rademacher function, the interval $[0; 1)$ is split into 2^{k+1} equal subintervals, on each of which the $r_k(x)$ function takes +1 and -1 values successively.

The Walsh system is the collection of all finite products formed from Rademacher functions. Precisely stated:

Definition 2. $W_0(x) \equiv 1$. Let n be any natural number, represented as $n = \sum_{s=1}^{k} 2^{m_s}$, $m_1 > m_2 > \cdots > m_k$. The n-th Walsh function will be defined as follows:

$$W_n(x) = \prod_{s=1}^{n} r_{m_s}(x)$$

Properties of Walsh functions

1. *Orthogonality*: The Walsh functions $W_n(x)$ are orthogonal on the interval [0, 1], meaning that for all $m, n \in \mathbb{N}$ the following holds:

$$\int_0^1 w_m(x) w_n(x) dx = \begin{cases} 1, & when\ m = n \\ 0, & when\ m \neq n \end{cases}$$

2. *Unit Energy:* Each function $w_k(x)$ has the norm L_2 equal to 1:

$$\|w_k(x)\|_2 = \sqrt{\int_0^1 w_k^2(x) dx} = 1$$

3. *Completeness:* Any function $f(x) \in L_2[0,1)$ can be expanded in a Walsh series:

$$f(x) = \sum_{k=0}^{\infty} c_k w_k(x)$$

where the coefficients c_k are defined as:

$$c_k = \int_0^1 f(x) w_k(x) dx$$

We can get Walsh functions using the Hadamard Matrix [10].

Walsh functions can be constructed using Hadamard matrices. A Hadamard matrix is a square matrix with entries of +1 or -1, where all rows are mutually orthogonal—meaning any two different rows have a scalar product of zero. For order n (where n is a power of 2), a Hadamard matrix is constructed as:

$$H_n = \begin{bmatrix} H_{n/2} & H_{n/2} \\ H_{n/2} & -H_{n/2} \end{bmatrix},$$

where $H_1 = [1]$.

For example:

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \qquad H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}.$$

The rows of the Hadamard matrix, when arranged in proper order, constitute the Walsh functions. This gives us a straightforward, recursive approach to generating Walsh functions [10].

III. Walsh-Hadamard Transform

Given a text expressed as a data vector $x \in \mathbb{R}^N$, where $N = 2^n$, we apply the Walsh-Hadamard transform for encryption in the following manner:

1. Direct transformation:
$$y = H_n x.$$
2. Inverse transformation: to decipher the data, the transposed matrix is used:
$$x = \frac{1}{N} H_n^T y.$$

Let e represent the noise vector added to the encrypted data:

$$\tilde{y} = y + e$$

When reversed, the recovered data will be:

$$\tilde{x} = \frac{1}{N} H_n^T \tilde{y} = x + \frac{1}{N} H_n^T e$$

If the noise vector e is small in the L_2 norm, the reconstruction error will also be small:

$$\|\tilde{x} - x\|_2 = \frac{1}{N} \|H_n^T e\|_2 \leq \frac{1}{N} \|H_n\|_2 \|e\|_2 = \|e\|_2$$

IV. Walsh Functions in Cryptography

The Walsh transform offers a different approach from conventional encryption techniques—it doesn't rely on asymmetric keys. Instead, both the sender and receiver share the same key to perform the transformation and its reversal. This simplifies the encryption process and reduces computational demands compared to asymmetric methods. The encryption process begins by expressing the original data as a vector, then multiplying it by a Walsh matrix. The result is a set of Walsh coefficients, which serve as the encrypted output ([1,2]). Before applying the Walsh transform, some pre-processing steps are typically required, such as formatting, normalizing, or padding the data. The transform decomposes the input into coefficients that reflect the contribution of each Walsh basis function to the original data.

Let's take a closer look at the math behind the Walsh transform. Consider an original data set as a $x = [x_1, x_2, \ldots, x_n]$ vector where the dimension represents the size of the data.

The Walsh transform is applied by multiplying this vector by a Walsh matrix H of the same dimension. The result is a vector of coefficients $c = [c_1, c_2, \ldots, c_n]$, obtained as:

$$c = H * x.$$

These coefficients represent the transformed version of the original data, where each value indicates the contribution of a corresponding Walsh basis function.

During decryption, the original data is reconstructed using the selected significant coefficients.

Here's how that works:

Let's say we have a $c' = [c_1', c_2', \ldots, c_k']$, vector of significant coefficients where k represents how many significant coefficients, you're working with. To recover the original data, the inverse Walsh transform is applied using only the selected significant coefficients:

$$x' = H^T * c',$$

In this case, the H^T matrix represents the transposed version of the Walsh transform matrix.

V. Advanced Encryption Standard (AES)

The Advanced Encryption Standard (AES) [11,12] is a symmetric encryption algorithm that protects data by scrambling it into 128-bit blocks using a secret key of 128, 192, or 256 bits. Depending on the key size, AES runs through 10, 12, or 14 rounds of encryption, with each round designed to thoroughly mix and obscure the original data. It arranges each block into a 4×4 grid of bytes and performs a sequence of operations: SubBytes replaces each byte using a carefully designed lookup table, ShiftRows shifts the rows of the grid to spread changes across the block, MixColumns blends the bytes within each column using finite field mathematics, and AddRoundKey mixes in a portion of the expanded key using XOR. The final round omits the MixColumns step to ensure decryption can reverse the process cleanly. The round keys themselves are generated by expanding the original key through rotation, substitution, and XOR operations. Decryption works by applying the inverse operations in reverse order. AES's design is rooted in strong mathematical principles, achieving both confusion and diffusion to resist attacks. Its balance of security, speed, and efficiency has made it the global standard for encrypting everything from online communications and file storage to VPNs and messaging apps.

VI. Hybrid Algorithm

The hybrid encryption algorithm developed in this work combines the Walsh Transform with the Advanced Encryption Standard (AES) to establish a two-layered security mechanism. In this approach, the Walsh Transform acts as a preprocessing stage, applied to the original data before AES encryption. By leveraging orthogonal Walsh functions, the input is transformed into the Walsh domain, where its structure is obscured in a pattern of coefficients (Figure 1).

The process begins by converting the plaintext message into its ASCII-based numerical representation, forming an input vector $x = [x_1, x_2, \dots, x_n]$. To enable compatibility with the Walsh Transform, the vector is padded such that its length n becomes a power of two. The transformed data is then computed using the Walsh matrix H, where the encryption-stage coefficients are obtained via the equation:

$$c = H \cdot x.$$

Here, H is an orthogonal Walsh matrix of size $n \times n$, and c is the resulting vector of Walsh-domain coefficients. This transformation disperses the structure of the original data, making it resistant to pattern recognition or statistical analysis. The coefficients c are converted into a floating-point byte stream to preserve numerical precision, and this byte stream is encrypted using AES in Cipher Block Chaining (CBC) mode. AES utilizes a randomly generated 128-bit symmetric key and a 128-bit initialization vector (IV) to perform secure encryption.

During decryption, the process is reversed. AES decryption is applied first using the same key and IV to retrieve the encoded Walsh-domain values. These values are then processed using the inverse Walsh Transform. Given that Walsh matrices are orthogonal, the inverse transform is equivalent to the original transform, i.e., $H^{-1} = H$. Therefore, the original message is recovered by applying:

$$c = H \cdot c$$

The resulting vector is rounded to the nearest integers, any padding is removed, and the ASCII values are decoded to reconstruct the original plaintext message. This hybrid encryption scheme combines the structural obfuscation of the Walsh Transform with the cryptographic strength of AES, offering enhanced resistance against both brute-force and statistical attacks while maintaining efficient performance.

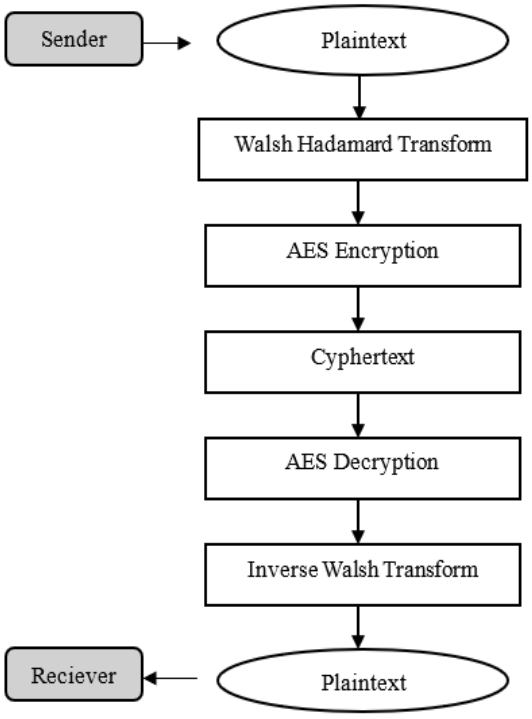

Figure 1. Diagram of Hybrid approach

To illustrate the process, consider the plaintext message "Chat." Each character is first converted into its ASCII code. This gives us the input vector:

$$x = [67,\ 104,\ 97, 116]$$

Since the vector length $n = 4$ is already a power of two, no padding is necessary. We then apply the Walsh transform using the 4×4 Walsh matrix H:

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

Multiplying the matrix by the input vector:

$$c = H \cdot x = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 67 \\ 104 \\ 97 \\ 116 \end{bmatrix} = \begin{bmatrix} 384 \\ -56 \\ -42 \\ -18 \end{bmatrix}$$

These transformed Walsh-domain coefficients $c = [384, -56, -42, -18]$ are converted into 32-bit floating-point format and serialized into bytes. Then, AES encryption is applied using CBC mode with a randomly generated key and IV. For example:

$AES\ Key\ (128 - bit)$:
$3f2a9c5d8b7e4f01d23456789abcdeff$

and

$Initialization\ Vector\ (IV)(128 - bit)$:
$00112233445566778899aabbccddeeff$

The encrypted output is a ciphertext in hexadecimal format, e.g.:

$Ciphertext = 7a3f1b9c5e2d8f4a9c7e0b12 \dots$

On the receiving side, the AES decryption using the same key and IV restores the byte stream of Walsh-domain coefficients:

$$c = [384,\ -56,\ -42,\ -18]$$

Applying the inverse Walsh Transform (same matrix H, since it is orthogonal):

$$x = H \cdot c = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 384 \\ -56 \\ -42 \\ -18 \end{bmatrix} = \begin{bmatrix} 67 \\ 104 \\ 97 \\ 116 \end{bmatrix}$$

These recovered values are converted back to characters:

$$[67,\ 104,\ 97,\ 116] \rightarrow "Chat"$$

The original message "Chat" is successfully restored.

VII. Cryptographic Security Metrics

When we want to figure out how well cryptographic algorithms actually work and whether we can trust them, we look at several different measurements. They help us understand how secure they are, how fast they perform, and how random their output really is.

One of the first things we check is something called entropy - basically, this tells us how unpredictable the

system is, which is crucial for good encryption. It is calculated using the Shannon entropy formula:

$$H = -\sum_{i=1}^{n} p_i \cdot \log_2(p_i)$$

where H is the entropy in bits, p_i is the probability of the i^{th} symbol, and n is the number of possible symbols.

Another key thing we look for when testing cryptographic algorithms is something called the Avalanche Effect. This basically means that even tiny changes to your input - like flipping just one bit - should create major, unpredictable changes in what comes out the other end.

In a perfect scenario, changing a single bit in your input should flip about half of the bits in your output. This kind of behavior is really important because it creates strong diffusion and stops attackers from spotting patterns they could exploit.

The Avalanche Effect is typically measured by:

$$Avalanche = \frac{Total\ Number\ of\ Output\ Bits}{Number\ of\ Changed\ Output\ Bits} \cdot 100\%$$

The Chi-Square Test is another way we check how random and evenly distributed the output from cryptographic systems really is - things like encrypted text or key streams. The essence of the method is to compare how often certain symbols (like bits or bytes) actually show up versus how often they should appear if everything was truly random.

The test is calculated using the formula:

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$$

where χ^2 is the Chi-Square statistic, O_i is the observed frequency of the i^{th} symbol, E_i is the expected frequency of the i^{th} symbol, n is the number of distinct symbols.

Table 1 shows how three different encryption methods stack up when they're all given the same test phrase: "Hello Armenia!"

First, we checked entropy. The hybrid AES+Walsh method came out on top with a score of 3.9448 bits per byte, which means it creates the most random-looking results. Regular AES wasn't far behind at 3.4772, while Walsh by itself only managed 2.9261—still decent, but not as impressive.

Next, we tested the avalanche effect. Both AES and the combined method performed really well here, changing about 30% of the bits (30.47% for AES and 30.86% for AES+Walsh). Walsh alone only managed to change 13.94% of the bits.

Finally, we ran a chi-square test to see how evenly distributed the scrambled bytes were. AES scored the best at 272.00, the hybrid method got 259.20, and Walsh alone came in at 240.00.

While Walsh adds some interesting mathematical complexity, AES on its own and the AES+Walsh combination both deliver much stronger security than using Walsh by itself. The hybrid approach gives you the best randomness, while AES provides the most balanced overall performance.

Table 1. Performance Comparison of Encryption Algorithms (For "*Hello Armenia!.*" plaintext.)

	Walsh	AES	AES+W
Entropy	2.9261 bits per byte	3.4772 bits per byte	3.9448 bits per byte
Avalanche Effect	13.94% bits changed	30.47% bits changed	30.86% bits changed
Chi-Square Test	240.00	272.00	259.20

VIII. Conclusion

When examining the test results collectively, it becomes evident that AES—whether applied independently or in combination with the Walsh Transform—offers significantly stronger security compared to using the Walsh Transform alone. Both AES and the hybrid AES+Walsh methods produce encrypted outputs that exhibit a high degree of randomness, a desirable property that enhances resistance to cryptanalysis. These methods also demonstrate strong sensitivity to minor alterations in the input: even a single-character change in the original message results in a substantially different ciphertext. This characteristic is essential for robust data protection.

Notably, the hybrid approach that integrates AES with the Walsh Transform appears to offer a synergistic advantage. It combines the established cryptographic strength of AES with the structural obfuscation introduced by the Walsh Transform, thereby enhancing overall encryption complexity and making unauthorized decryption substantially more difficult.

References

[1] S. A. Episkoposian, "Application of Walsh system in data encryption," *Tuijin Jishu/Journal of Propulsion Technology*, vol. 44, no. 3, pp. 494–502, 2023.

[2] S. A. Episkoposian, "Securing information with the Walsh transform," in *Proc. III Int. Sci. Pract. Conf.: Questions, Hypotheses, Answers – Science XXI Century*, Toronto, Canada, pp. 10–14, Sep. 12–13, 2023.

[3] S. A. Episkoposian and S. A. Grigoryan, "Hybrid cryptographic algorithm based on AES, RSA and Walsh transform," *Theor. Appl. Technol. Sci. Rev.*, vol. 3, no. 1, pp. 736–745, Feb. 2025.

[4] S. Episkoposian, G. Margarov, and S. Grigoryan, "A metric-based comparison of Walsh, RSA, and hybrid Walsh-RSA encryption techniques," *Bradleya*, vol. 43, no. 7, pt. 1, pp. 2–12, Jul. 2025. https://doi.org/10.61586/0PbJL

[5] A. S. Yepiskoposyan, "Improvement of information encryption with secret key generation using the Walsh–Hadamard transform," in *Proc. XI Int. Sci. Conf.*, Dortmund, Germany, pp. 67–72, 2024. https://doi.org/10.5281/zenodo.10564951

[6] N. J. Fine, "On the Walsh functions," *Trans. Amer. Math. Soc.*, vol. 65, no. 3, pp. 372–414, 1949. https://doi.org/10.1090/s0002-9947-1949-0032833-2

[7] R. B. Lackey and D. Meltzer, "A simplified definition of Walsh functions," *IEEE Trans. Comput.*, vol. C-20, no. 2, pp. 211–213, Feb. 1971. https://doi.org/10.1109/T-C.1971.223214.

[8] F. Schipp, W. R. Wade, and P. Simon, *Walsh Series: An Introduction to Dyadic Harmonic Analysis*. Budapest, Hungary: Akadémiai Kiadó, 1990.
[9] J. L. Walsh, “A closed set of normal orthogonal functions,” *American Journal of Mathematics*, vol. 55, pp. 5–24, 1923.
[10] H. O. Kunz, "On the equivalence between one-dimensional discrete Walsh–Hadamard and multidimensional discrete Fourier transforms," *IEEE Trans. Comput.*, vol. 28, no. 3, pp. 267–268, Mar. 1979.
[11] J. Daemen and V. Rijmen, *The Design of Rijndael: AES — The Advanced Encryption Standard*. Springer, 2002.
[12] National Institute of Standards and Technology, *Announcing the Advanced Encryption Standard (AES)*, FIPS PUB 197, U.S. Department of Commerce, 2001. https://doi.org/10.6028/NIST.FIPS.197-upd1