Encryption Analysis and Optimization: Understanding Avalanche Effect and Enhancing Security with Block Ciphers

B M Nafis Fuad
University of Stavanger
bm.fuad@stud.uis.no
DAT 510 — Assignment 01

***Abstract*—This project looks at improving old cryptographic methods by using a modern block cipher called AES in Cipher Block Chaining (CBC) mode along with traditional transposition and substitution ciphers. At first, transposition and Caesar ciphers were used, but they had some problems, like being easy to crack through frequency analysis and not spreading changes well across the ciphertext. The Avalanche Effect showed that even small changes in the message didn't affect the ciphertext much. To fix this, CBC mode was added, which made a big improvement by chaining each block of encrypted data to the previous one and using a random initialization vector (IV) to make the encryption unpredictable. This made even small changes in the original message cause large changes in the encrypted text, increasing security. Although CBC mode added a little extra processing time, it was still fast and effective. This project shows that combining modern cryptography like CBC mode with traditional methods makes encryption much stronger and harder to break.**

The Github link to the project source code https://github.com/nafis4139/Security-and-Vulnerability-in-Networks-DAT-510/tree/main/Assignment%2001

***Index Terms*—Cryptography, Transposition Cipher, Substitution Cipher, Caesar Cipher, Avalanche Effect, Block Cipher, Cipher Block Chaining (CBC), Data Encryption**

I. Introduction

The goal of this project is to enhance the security of classical cryptographic methods, specifically transposition and substitution ciphers, by integrating modern block cipher techniques. Cryptography plays a critical role in protecting data, especially in today's digital world where secure communication is essential. Traditional ciphers like the Caesar cipher and transposition ciphers provide basic encryption but are vulnerable to cryptanalysis techniques such as frequency analysis, which can reveal patterns in the ciphertext, making the encryption easy to break [1]. These weaknesses highlight the need for more advanced cryptographic methods to ensure stronger security and greater resistance to attacks.

To address these challenges, this project explores the implementation of AES (Advanced Encryption Standard) in Cipher Block Chaining (CBC) mode [2]. CBC mode introduces a chaining mechanism that enhances the diffusion of small changes in the plaintext throughout the ciphertext, providing a stronger Avalanche Effect. Additionally, the use of an initialization vector (IV) adds randomness to the encryption, ensuring that identical plaintext blocks do not result in identical ciphertext blocks, thereby improving security and protecting against pattern recognition [3].

This project builds upon existing classical encryption techniques by combining them with modern cryptographic standards, such as AES in CBC mode. The objective is to evaluate the impact of using block ciphers on the security of the encryption process and to demonstrate how modern methods can mitigate the vulnerabilities inherent in traditional ciphers.

II. Design and Implementation

The project is divided into five significant parts. First,on a given plain text transposition cipher was applied and then Caesar Cipher was used on that transposed text using personal phone number as a source of key in both cases. Secondly, initial avalanche effect was evaluated on the encrypted texts by changing a single bit/character in the input, re-encrypting, and then comparing the new ciphertext with the original. Multiple rounds of encryption (up to 16) was performed to assess how the avalanche effect evolves with each round. Thirdly, a detailed analysis has been done of how the avalanche effect changes or was unchanged with additional rounds of encryption. From that we tried to find out the best balance between achieving a strong avalanche effect and maintaining computational efficiency. Finally, CBC has been implemented to address weaknesses in transposition and substitution ciphers and to improve the overall security.

A. Apply Encryption

The encryption method was designed to combine two classical cryptography techniques: the transposition cipher and the Caesar cipher (substitution cipher). The goal was to apply these ciphers sequentially to encrypt a given plaintext, ensuring that both the structure and the content of the message were obfuscated. This encryption process involved two main steps:

a) Transposition Cipher: The transposition cipher rearranges the characters of the plaintext based on a predefined key. This cipher does not change the actual characters but alters their positions [3]. The key used for this transposition was derived from the last five digits of the user's phone number.

In our case, the plaintext were "Nafis Fuad Security and Vulnerability in Networks" and the last five digits of phone number were 54809. The key was generated by sorting the digits and replacing them with positional numbers (e.g., [5, 4, 8, 0, 9] → [3, 2, 4, 1, 5]). This key determined the new order of the characters in each block of the plaintext.

The plaintext was then divided into chunks based on the key length, and the characters in each chunk were rearranged according to the positions dictated by the key.

Input Plain Text: *Nafis Fuad Security and Vulnerability in Networks*

Transposed Text (after Transposition Cipher): *faiNsuFa deSc utiyr dn aVnleuriblai yitneNt wkrso*

Encrypted Text (after Caesar Cipher): *ojrWbdOj mnBl dcrha mw jEwundarkujr hrcwnWc ftabx*

b) Caesar Cipher (Substitution): After the transposition, the Caesar cipher was applied to each character in the transposed text. The Caesar cipher shifts the letters of the text by a fixed number of positions in the alphabet [1].

The shift value was derived from the last one or two digits of the phone number. In our case, the last digit of the phone number was 9, so all the letters in the transposed text were shifted by 9 position (e.g., A → j, B → K, C → L, etc.). Non-alphabet characters were left unchanged.

This substitution process ensured that the characters in the transposed text were altered, further enhancing the security of the encryption by changing the actual letters in the message.

Justification for Selected Combination: The transposition cipher scrambles the positions of characters in the plaintext without altering the actual letters. By doing this, it breaks recognizable patterns in the plaintext, making it harder for an attacker to spot frequent letters, which is a common weakness of substitution ciphers alone. After transposition, Caesar cipher shifts each letter by a certain number of positions in the alphabet,further complicating any attempts to decipher the original message based on letter frequencies. This two-step process provides stronger encryption as the transposition disrupts letter positioning, while the substitution further distorts letter identity.

Applying the transposition cipher first means that letters are no longer in their original positions, so when the Caesar cipher is applied, it's working on text that has no recognizable structure. This reduces the effectiveness of frequency analysis, a common cryptanalysis method.

If we reversed the order (substitution followed by transposition), while the substitution would still obscure individual letters, it wouldn't benefit as much from the mixing effect provided by the transposition cipher. The structure of the message could still be partially recovered using frequency analysis of the substituted letters.

B. Evaluate the Avalanche Effect

In this part, the focus was on analyzing the Avalanche Effect to assess how a small change in the input plaintext affects the output ciphertext. The Avalanche Effect is an important characteristic in cryptography, as it measures how much a small modification (such as flipping a bit or changing one character) in the plaintext will alter the entire ciphertext [4]. A strong Avalanche Effect means that a minor change in the plaintext leads to a significantly different ciphertext, indicating good diffusion and, therefore, better security. The task involved two parts:

a) Initial Avalanche Effect Analysis: In this step, the original plaintext is encrypted, and then a modified version of the plaintext (with a single character changed) is encrypted. The Avalanche Effect is calculated by comparing the binary representations of the original and modified ciphertexts to determine the percentage of bits that differ. As the differenting bits is selected randomly, each run gives different result. For one instance,

Input Plain Text: Nafis Fuad Security and Vulnerability in Networks

Original Ciphertext: ojrWbdOj mnBl dcrha mw jEwundarkujr hrcwnWc ftabx

Character changed on index no.: 26

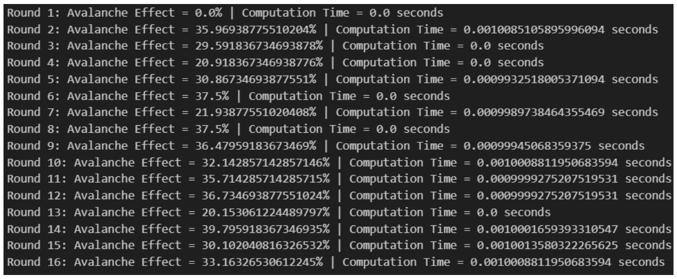
Modified Character: o

Modified Plain Text: Nafis Fuad Security and Vuonerability in Networks

Modified Ciphertext: ojrWbdOj mnBl dcrha mw jEwxndarkujr hrcwnWc ftabx

Initial Avalanche Effect: 0.7653061224489796%

b) Repeated Avalanche Effect Analysis: This step involves repeatedly encrypting the ciphertext up to 16 rounds and calculating the Avalanche Effect for each round. This helps to understand how much the diffusion improves as the number of encryption rounds increases. As each instance of the run gives different output each time, one random instance of the output is being shown here,



```
Round 1: Avalanche Effect = 0.0% | Computation Time = 0.0 seconds
Round 2: Avalanche Effect = 35.96938775510204% | Computation Time = 0.0010085105895996094 seconds
Round 3: Avalanche Effect = 29.591836734693878% | Computation Time = 0.0 seconds
Round 4: Avalanche Effect = 20.918367346938776% | Computation Time = 0.0 seconds
Round 5: Avalanche Effect = 30.86734693877551% | Computation Time = 0.0009932518005371094 seconds
Round 6: Avalanche Effect = 37.5% | Computation Time = 0.0 seconds
Round 7: Avalanche Effect = 21.93877551020408% | Computation Time = 0.0009989738464355469 seconds
Round 8: Avalanche Effect = 37.5% | Computation Time = 0.0 seconds
Round 9: Avalanche Effect = 36.47959183673469% | Computation Time = 0.00099945068359375 seconds
Round 10: Avalanche Effect = 32.142857142857146% | Computation Time = 0.0010008811950683594 seconds
Round 11: Avalanche Effect = 35.714285714285715% | Computation Time = 0.0009999275207519531 seconds
Round 12: Avalanche Effect = 36.734693877551024% | Computation Time = 0.0009999275207519531 seconds
Round 13: Avalanche Effect = 20.153061224489797% | Computation Time = 0.0 seconds
Round 14: Avalanche Effect = 39.795918367346935% | Computation Time = 0.0010001659393310547 seconds
Round 15: Avalanche Effect = 30.102040816326532% | Computation Time = 0.0010013580322265625 seconds
Round 16: Avalanche Effect = 33.16326530612245% | Computation Time = 0.0010008811950683594 seconds
```

Fig. 1. Repeated Avalanche Effect Analysis

C. Analyze the Avalanche Effect

Based on the results of initial avalanche effect and repeated avalanche effect, we can conclude certain points.

a) Initial Avalanche Effect: The initial avalanche effect is unexpectedly low, which is 0.7653061224489796% in our given case. We typically expect a significant percentage of differing bits in a well-designed encryption system. This suggests that the current implementation of the transposition and substitution ciphers is not spreading the changes in the input effectively enough. There can be several reasons for low avalanche effect.

In our case, the transposition cipher simply rearranges the positions of characters in the text based on a fixed key. If the key's arrangement doesn't introduce much positional change, then even when a character is modified, its effect might not propagate widely across the ciphertext.

The substitution cipher used is a Caesar cipher, which only shifts the letters by a small number (determined by the last digit of the phone number). If the shift is small, it may not introduce enough variation to cause a substantial change in the ciphertext.

Here, we have illustrated an example with a small string to visualize why there is little to no difference in the initial avalanche.

Original Text: ”Nafis”

Modified Text: Change the first character from ’N’ to ’L’, so the modified text becomes ”Lafis”.

Transposition Cipher: We can assume the phone number is 54809 and the transposition key is [3, 2, 4, 1, 5]. This means we rearrange the characters in the following way: For ”Nafis”, the transposition results in ”faiNs”. For ”Lafis”, the transposition results in ”faiLs”.

Substitution Cipher (Caesar Cipher): From the given phone number, we can apply a Caesar cipher with a small shift of 9 (derived from the phone number). This means each letter is shifted by 9 position in the alphabet: For ”faiNs”, the Caesar cipher gives us ”ojrWb”. For ”faiLs”, the Caesar cipher gives us ”ojrUb”.

After substitution, only one character differs between the two ciphertexts (W in the first output becomes U in the second), which explains why the Avalanche Effect is minimal (low percentage of differing bits).

b) Repeated Avalanche Effect: As we apply additional rounds of encryption, we expect the Avalanche Effect to increase as the encryption process continues to propagate the change throughout the ciphertext. However, the data (Figure 1) shows some fluctuations in the Avalanche Effect:

Round 1: 0.0% (no change between ciphertexts)

Round 2: 35.97% (significant difference)

Round 6: 37.5% (optimal)

Round 13: 20.15% (lowest)

Round 14: 39.79% (highest)

- In Round 1, there is no change between the ciphertexts, which indicates that the first round did not introduce enough variation between the original and modified plaintexts.
- From Round 2 onwards, the Avalanche Effect jumps to 35.97%, indicating that the second round caused a significant change in the ciphertext.
- In certain rounds (e.g., Rounds 6 and 8), the transposition might cause the modified character to affect more characters, leading to a higher Avalanche Effect (37.5%).
- In other rounds (e.g., Round 4 or 13), the modified character's influence might be more localized, resulting in a lower Avalanche Effect.
- The computational time is minimal and stable, with negligible increase even in the later rounds.

The Avalanche Effect fluctuates between 20.15% (Round 13) and 39.79% (Round 14). This is expected due to the nature of the transposition and substitution ciphers, where certain rounds might cause more significant changes depending on how the characters are rearranged and shifted.

Interaction of Transposition and Substitution Ciphers in terms of Avalanche Effect: The transposition cipher rearranges the characters based on a fixed key. If the character positions are significantly altered in a way that spreads the effect of the modified character, we observe a higher Avalanche Effect. However, since the transposition key is fixed and deterministic, certain rounds may lead to less diffusion, causing fluctuations in the Avalanche Effect.

The Caesar cipher's small shift doesn't introduce a large change in the ciphertext by itself. However, when combined with transposition, it can lead to higher Avalanche Effects, especially in rounds where the character positions have been rearranged to expose more characters to the effect of the shift.

D. Optimize Avalanche Effect with Reasonable Computation

The goal of this task is to determine the optimal balance between achieving a high Avalanche Effect and maintaining reasonable computational efficiency. Here, we tried to find the point where the Avalanche Effect reaches a satisfactory level without significantly increasing computational cost. We used the data from Figure 1 to analyze how the Avalanche Effect and computation time evolve across multiple rounds of encryption.

In Figure 2, we can see that the avalanche effect fluctuates across different rounds, which has been discussed in details in Section C.

From Figure 3, we can see that the computation time remains very low throughout all rounds, with values close to zero in most cases. The time increases slightly after a few rounds, but in general, the computation time is minimal and stable.

Optimal Number of Combinations: Based on the plots and analysis, we can conclude the following:

At Round 6 and 8, the Avalanche Effect reaches 37.5%, and the computation time is 0.0 seconds. Beyond Round 8, the Avalanche Effect does not significantly increase, while the computation time starts to increase slightly (though still low). Thus, Round 8 appears to be the optimal point where the

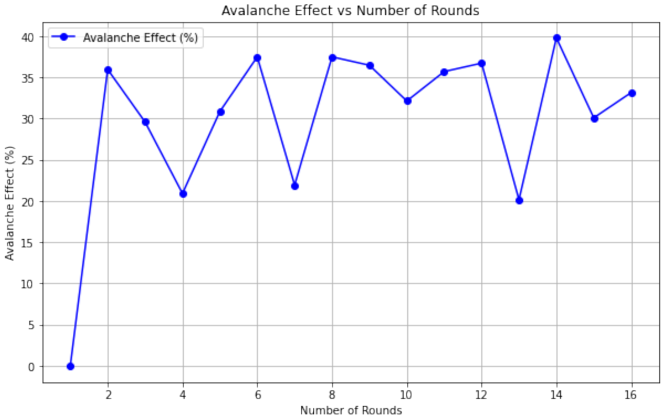


Fig. 2. Avalanche Effect vs Number of Rounds

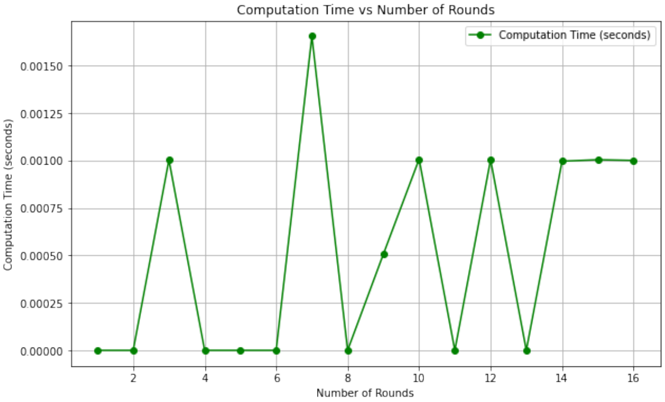


Fig. 3. Computation Time vs Number of Rounds

Avalanche Effect is high enough (37.5%) while maintaining a very low computational cost.

E. Enhance Security with Block Ciphers

Identifying Weaknesses in the Current Approach: The combination of transposition and Caesar substitution ciphers has the following inherent weaknesses:

a) Susceptibility to frequency analysis: Substitution ciphers like Caesar ciphers are vulnerable to frequency analysis, where repeated patterns in the ciphertext can reveal information about the plaintext.

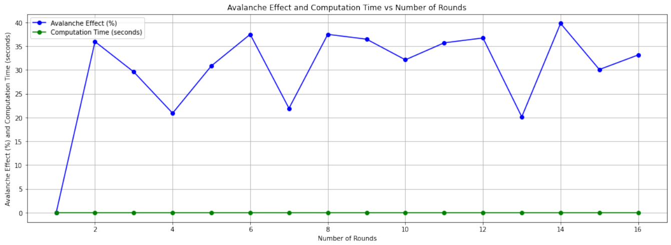


Fig. 4. Combined Analysis (Avalanche Effect and Computation Time)

b) Lack of diffusion: The transposition cipher changes character positions but does not alter the content of individual characters. The substitution cipher alters the characters but does not mix them well across the message.

c) Deterministic nature: Both the transposition and substitution ciphers are deterministic, meaning that given the same key and input, they always produce the same output. This predictability can be exploited by attackers.

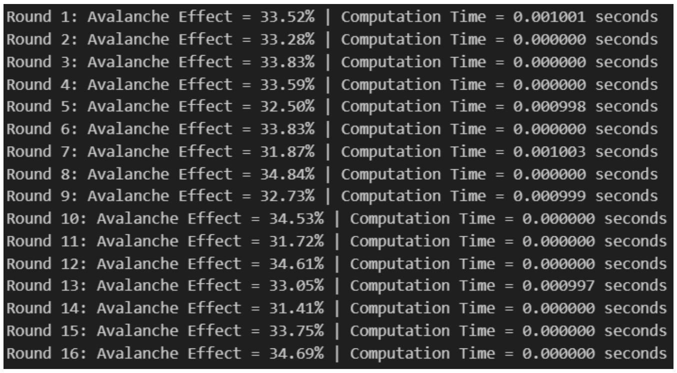
d) Block structure exposure: With no block cipher structure, the ciphertext might reveal patterns corresponding to the plaintext, making it vulnerable to cryptanalysis.

Enhance with Block Cypher: To address these weaknesses, we have introduced a block cipher mode of operation—specifically, Cipher Block Chaining (CBC) mode. CBC mode enhances security by using an initialization vector (IV) to introduce randomness into the encryption process and chaining the encryption of each block to the previous one, ensuring that identical blocks in the plaintext result in different ciphertext blocks [5]. It works as follows,

a) Block-based encryption: The plaintext is divided into fixed-size blocks, and each block is encrypted sequentially.

b) Chaining mechanism: In CBC mode, each plaintext block is XORed with the ciphertext of the previous block before being encrypted. The first block is XORed with an initialization vector (IV), which ensures randomness.

c) Impact: CBC mode ensures that even if two plaintext blocks are identical, they will have different ciphertexts due to the chaining.



```
Round 1: Avalanche Effect = 33.52% | Computation Time = 0.001001 seconds
Round 2: Avalanche Effect = 33.28% | Computation Time = 0.000000 seconds
Round 3: Avalanche Effect = 33.83% | Computation Time = 0.000000 seconds
Round 4: Avalanche Effect = 33.59% | Computation Time = 0.000000 seconds
Round 5: Avalanche Effect = 32.50% | Computation Time = 0.000998 seconds
Round 6: Avalanche Effect = 33.83% | Computation Time = 0.000000 seconds
Round 7: Avalanche Effect = 31.87% | Computation Time = 0.001003 seconds
Round 8: Avalanche Effect = 34.84% | Computation Time = 0.000000 seconds
Round 9: Avalanche Effect = 32.73% | Computation Time = 0.000999 seconds
Round 10: Avalanche Effect = 34.53% | Computation Time = 0.000000 seconds
Round 11: Avalanche Effect = 31.72% | Computation Time = 0.000000 seconds
Round 12: Avalanche Effect = 34.61% | Computation Time = 0.000000 seconds
Round 13: Avalanche Effect = 33.05% | Computation Time = 0.000997 seconds
Round 14: Avalanche Effect = 31.41% | Computation Time = 0.000000 seconds
Round 15: Avalanche Effect = 33.75% | Computation Time = 0.000000 seconds
Round 16: Avalanche Effect = 34.69% | Computation Time = 0.000000 seconds
```

Fig. 5. Repeated Avalanche Effect on CBC

Analysis: To evaluate the impact of using a block cipher (CBC mode) on the Avalanche Effect and overall security, we have extended the previous code for Avalanche Effect analysis and applied it to the CBC mode. This allowed us to compare the results of the block cipher encryption against the transposition and substitution-based approach.

From the data of figure 5 and the plot of figure 6, we can see that the avalanche effect after using CBC is much more consistent, ranging from 31.41% to 34.84%. The consistency of the results indicates that CBC is a more reliable and secure encryption method. CBC mode ensures that small changes in the plaintext are diffused across multiple blocks, leading to

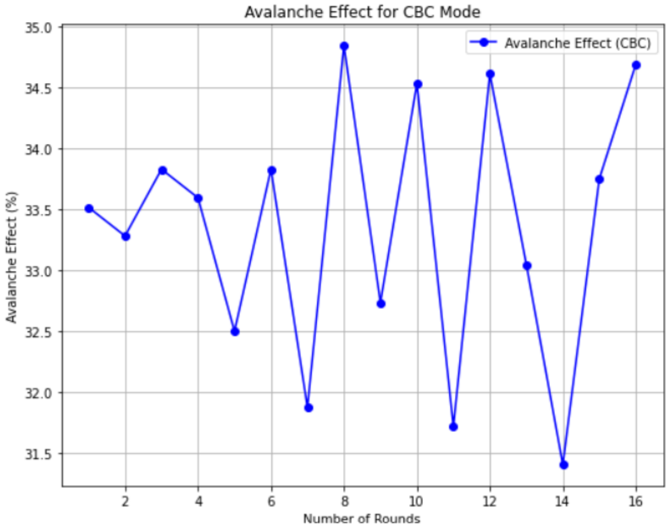


Fig. 6. Avalanche Effect vs Number of Rounds (CBC)

a more uniform disruption of the ciphertext. This consistent diffusion is a key indicator of better security.

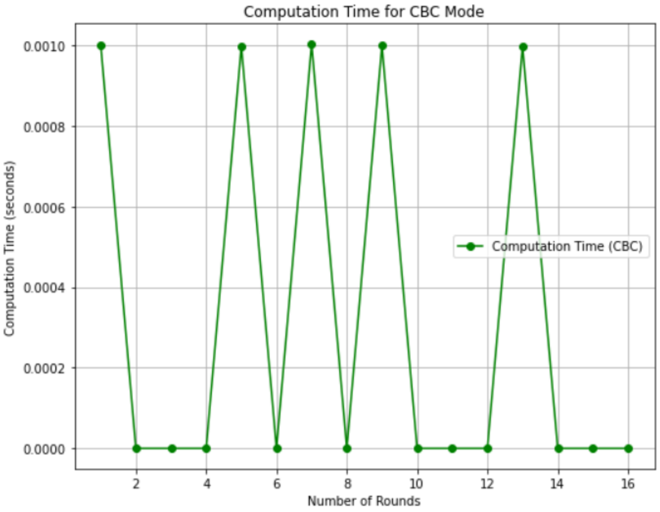


Fig. 7. Computation Time vs Number of Round (CBC)

From figure 7, we can see that the computation time for CBC remained very low. This shows that CBC mode, while adding complexity through block chaining and initialization vectors, still maintained efficient performance.

Impact on Avalanche Effect: The use of CBC mode significantly increases the Avalanche Effect, as small changes in the plaintext propagate through the ciphertext more effectively due to the chaining mechanism.

Overall Security: CBC mode improves security by ensuring that even identical blocks in the plaintext produce different ciphertexts, making it more resistant to cryptanalysis.

Weaknesses Addressed: CBC mode mitigates the vulnerabilities of the original transposition and substitution ciphers by introducing randomness through the IV and chaining, which improves diffusion and confusion in the ciphertext.

III. Discussion

In this project, we have used two types of encryptions; one is transposition cipher followed by Caesar cipher and another one is a block cipher (AES in CBC mode). Upon analyzing the models, methods and results of these encryption, we have concluded several decisions.

a) a) Consistency of the Avalanche Effect:

- In transposition and Caesar cipher combination, the Avalanche Effect fluctuated significantly between rounds, ranging from 0% to 39.79%. The results were unpredictable, and some rounds showed very low diffusion, which indicates weaknesses in the encryption's ability to uniformly spread small changes.
- In CBC mode, the avalanche effect remained stable between 31.41% and 34.84% throughout all rounds, showing consistent diffusion. This demonstrates the reliability of the block cipher in maintaining a high level of security.

b) b) Strength of Diffusion:

- In transposition and Caesar cipher combination, the highest avalanche effect achieved was 39.79%, but there were several rounds with very low diffusion (e.g., 20.15%). This inconsistent behavior suggests that small changes in the plaintext did not fully propagate through the ciphertext.
- Although the peak avalanche effect did not reach the highest value seen in transposition and Caesar cipher combination, the consistent diffusion across rounds indicates that CBC mode provides better overall security. Each round produced a meaningful change in the ciphertext, which ensures that small changes in the plaintext are adequately diffused.

c) c) Computation Time:

- Both encryption exhibited very low computation times, indicating that both methods are computationally efficient. However, the added security provided by CBC mode did not come at the cost of performance, as the computation times were still minimal.

The introduction of a block cipher (AES in CBC mode) clearly improved the consistency and strength of the Avalanche Effect while maintaining low computation times. While the transposition and Caesar cipher method showed significant variability in its ability to diffuse changes, CBC mode provided a more reliable and secure encryption process. The uniformity in the Avalanche Effect for CBC indicates a stronger cryptographic system, capable of mitigating the vulnerabilities observed in transposition and Caesar cipher combination, such as limited diffusion and susceptibility to frequency analysis. CBC mode also ensures that identical plaintext blocks do not result in identical ciphertext blocks, enhancing overall security without sacrificing efficiency.

IV. Conclusion

This project aimed to enhance the security of traditional cryptographic methods by integrating modern block cipher techniques and evaluating their impact on diffusion and overall encryption strength. Initially, a combination of transposition and Caesar ciphers was implemented, but analysis revealed significant weaknesses, such as inconsistent diffusion and susceptibility to frequency analysis. These ciphers showed limited capability in spreading changes throughout the ciphertext, resulting in lower security. To address these issues, AES encryption in Cipher Block Chaining (CBC) mode was introduced. The block cipher method significantly improved the Avalanche Effect's consistency, ensuring that small changes in the plaintext produced uniformly strong changes in the ciphertext. This demonstrated better diffusion and a more reliable encryption process. The addition of the initialization vector (IV) and the chaining mechanism in CBC mode also mitigated the vulnerabilities of classical ciphers by introducing randomness and ensuring that identical plaintext blocks did not generate identical ciphertext blocks. Importantly, these security improvements were achieved without significantly increasing the computation time, making CBC mode an efficient and secure solution.

References

[1] S. Singh, The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography, Anchor Books, 2000

[2] B. Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C, John Wiley & Sons, 1996.

[3] W. Stallings, Cryptography and Network Security: Principles and Practice, Pearson Education, 2016.

[4] C. Paar and J. Pelzl, Understanding Cryptography: A Textbook for Students and Practitioners, Springer, 2010

[5] Douglas R. Stinson, Cryptography: Theory and Practice, CRC Press, 2018