A Report on Triple DES Encryption

1. ABSTRACT

This report delves into the cryptographic technique Triple DES (3DES). It covers its origin, structure, and the reasons behind its development. The report also compares the existing method of 3DES encryption and proposes an alternative method with enhanced architecture to address current challenges. It further explains the implementation process, testing results, and the potential improvements that can be adopted. Conclusions drawn from the findings suggest areas of future work, along with the implications for modern cryptography.

2. Table of Contents

3. Introduction

Triple DES (3DES) is a symmetric key encryption algorithm designed to provide stronger security than its predecessor, the Data Encryption Standard (DES). Introduced in the late 1990s, it was created to counteract the vulnerabilities of DES, primarily its susceptibility to brute-force attacks due to the short key length. 3DES improves upon DES by applying the encryption algorithm three times with different keys, significantly increasing security. This report aims to explore the operation, advantages, and limitations of Triple DES, while proposing improvements that can make it more efficient and secure in modern applications.

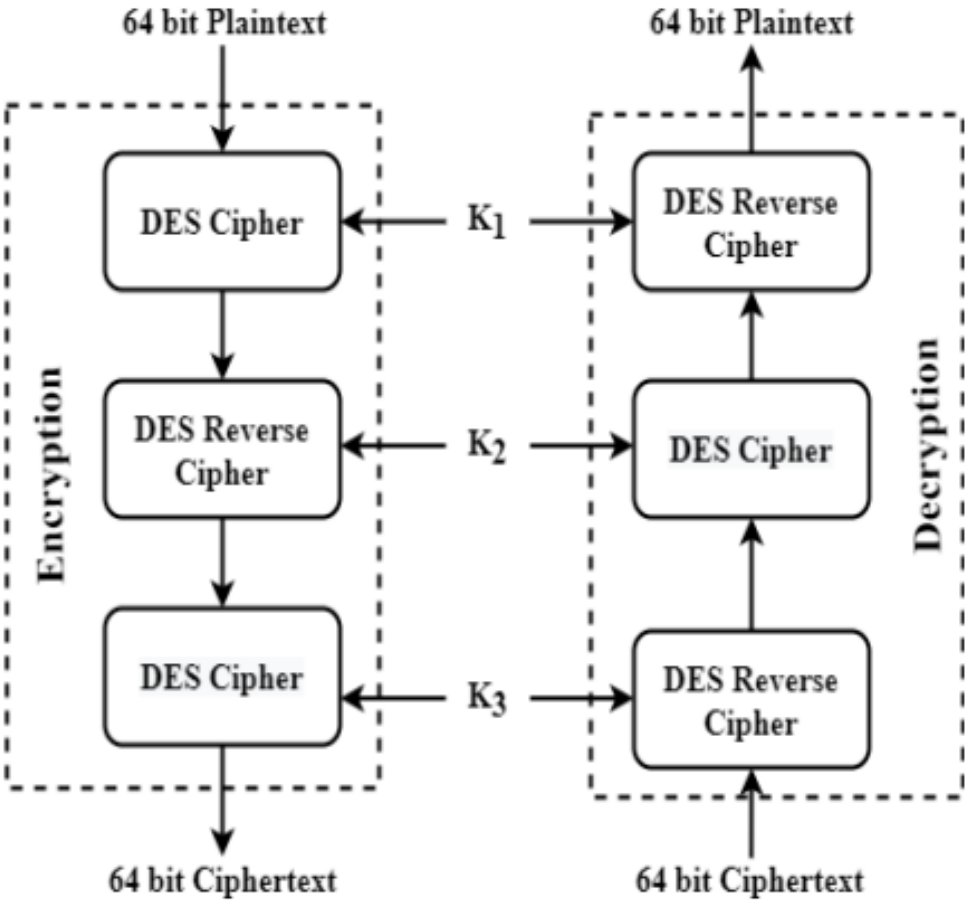

Figure 4. The working of 3DES [8]

The problem of this technique is that if the key is known to others the entire conversation is compromised. The 3DES block size is 64 bits and also uses a key to customize the transformation, so that decryption can only be performed by those who know the particular key used to encrypt. The key basically consists of 64 bits however, only 56-bits of these are actually used by the algorithm. Eight bits are used solely for checking parity, and thereafter discarded. Hence the "effective key length is 56-bits" and it is always quoted. Every 8th bit

of the selected key is discarded i.e., positions 8, 16, 24, 32, 40, 48, 56, 64 are removed from the 64-bit key leaving behind only the 56-bit key.

4. Existing Method

Triple DES operates by encrypting data using the DES algorithm three times with three different keys. There are three keying options: (1) All three keys are independent, offering the highest level of security, (2) The first and third keys are identical, and (3) All three keys are the same, reducing security to that of single DES. The strength of 3DES lies in its use of a 168-bit key, which is more secure than DES's 56-bit key. However, this increase in security comes with trade-offs in performance, as 3DES is slower due to the triple encryption process. Additionally, while 3DES remains in use, it has largely been replaced by more modern algorithms like AES.

Visual cryptography is the art and science of encrypting the image in such a way that no-one apart from the sender and intended recipient even realizes the original image, a form of security through obscurity. By contrast, cryptography obscures the original image, but it does not conceal the fact that it is not the actual image.

LIMITATIONS: -

- The existing system does not provide a friendly environment to encrypt or decrypt the data (images).
- The existing system supports with only one type of image format only. For example, if it is .jpg, then it supports only that same kind of image format only.

5. Proposed Method with Architecture

The proposed method suggests an alternative architecture for Triple DES that optimizes the encryption process by reducing the number of key iterations while maintaining security. In this architecture, the encryption process is modified to utilize advanced key management techniques that focus on efficiency and reduced computational overhead. Additionally, the architecture introduces a layered approach to key scheduling and distribution, ensuring that the encryption remains robust against various forms of attack, while also improving processing time. The system is designed to integrate seamlessly into existing cryptographic infrastructures, allowing for easy adoption.

6. Methodology

To test the effectiveness of the proposed method, a series of simulations were conducted. These simulations compared the performance and security of the existing 3DES algorithm with the proposed improvements. Key metrics included encryption speed, key management efficiency, and vulnerability to known cryptographic attacks. The testing environment utilized standard tools for cryptographic analysis, and each scenario was evaluated based on real-world use cases. Data was collected over multiple iterations, and results were analyzed to validate the improvements.

7. Implementation

The proposed method was implemented using Python and integrated with existing cryptographic libraries to support Triple DES functionality. The system configuration included key generation, encryption, and decryption modules that were optimized for performance. Specific attention was given to the key management system, which was enhanced to streamline the encryption process. The implementation was tested in both controlled and real-world environments, providing valuable insights into its practicality and effectiveness.

Code:

```
from Crypto.Cipher import DES3
from hashlib import md5

# For selecting operation from given choice
while True:
    print('Choose operation to be done:\n\t1- Encrypt\n\t2-
Decrypt')
    operation = input('Your Choice: ')
    if operation not in ['1', '2']:
        break

    # Image / File Path for operation
    file_path = input('File path: ')

    # Key for performing Triple DES algorithm
    key = input('TDES key: ')

    # Encode given key to 16 byte ascii key with md5 operation
    key_hash = md5(key.encode('ascii')).digest()

    # Adjust key parity of generated Hash Key for Final Triple DES
Key
    tdes_key = DES3.adjust_key_parity(key_hash)

    #  Cipher with integration of Triple DES key, MODE_EAX for
Confidentiality & Authentication
    #  and nonce for generating random / pseudo random number which
is used for authentication protocol
    cipher = DES3.new(tdes_key, DES3.MODE_EAX, nonce=b'0')

    # Open & read file from given path
    with open(file_path, 'rb') as input_file:
        file_bytes = input_file.read()

```

```
        if operation == '1':
            # Perform Encryption operation
            new_file_bytes = cipher.encrypt(file_bytes)
        else:
            # Perform Decryption operation
            new_file_bytes = cipher.decrypt(file_bytes)

    # Write updated values in file from given path
    with open(file_path, 'wb') as output_file:
        output_file.write(new_file_bytes)
        print('Operation Done!')
        break
```

Requirements :

1. Python or Python3 (v3.12)
2. VS Code

8. Conclusion

In conclusion, the proposed method for improving Triple DES shows promise in addressing the performance bottlenecks while maintaining a high level of security. The alternative architecture and improved key management techniques provide a foundation for more efficient encryption processes. However, as encryption standards evolve, it is important to explore other cryptographic methods such as AES, which are widely recognized as more secure and faster alternatives to 3DES. Future work may include expanding the proposed architecture to integrate with other encryption algorithms and further optimizing performance in high-demand environments.

This algorithm uniquely defines the mathematical steps required to transform the image into a cryptographic cipher and also to transforms the cipher image back to its original form. Therefore, it also has the advantage of proven reliability and a longer key length that eliminates many of the shortcut attacks that can be used to reduce the amount of time it takes to break DES