iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** IX **Month of publication:** September 2024

DOI: https://doi.org/10.22214/ijraset.2024.64392

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue IX Sep 2024- Available at www.ijraset.com

Analysis of Image Encryption using Triple DES

Anton George[1], Arya Dubal[2]
Fourth Year Computer Engineering, Thadomal Shahani Engineering College, University of Mumbai

Abstract: ***Keeping your essential data secure in today's world is very essential. There are different types of cyber-attacks which attackers can use to exploit your personal or confidential data for their own needs. Cryptography and its algorithms in cybersecurity has been an important part in providing secure methods to hide or encrypt information or sensitive data which only the person who was intended to be send can access it. This paper presents the triple Data Encryption Standard (DES) approach for secure image encryption from malicious users. This paper provides an efficient, easy and secure approach for encryption of images which is essential in today's scenario.***

Keywords: ***Data Encryption Standard (DES), Triple Data Encryption Standard (3DES), Message Digest Method 5 (MD5), American Standard Code for Information Interchange (ASCII)***

I. INTRODUCTION

Nowadays, automation, computation and communication over the network among users is in immense use among all sectors like software development, healthcare, banking, education, manufacturing and even in small scale industries. So, there would be some private communication or data in every sector which should be strongly encrypted, message be sent to the intended user only and protection from malicious users. For example, in banking, all the user data and passwords should be safely encrypted so that it cannot be hacked. Similarly for healthcare, patient health records should be kept secure as they should not be tampered for ill use.

Cryptography comes into play for securing such private and sensitive data. It uses special techniques and algorithms to encrypt the data or message into some code and sent to the intended receiver only who can decode and interpret it. DES is such a technique which is a symmetric key block cipher which encrypts the message or data for secure transfer over the network.

II. DES

DES stands for Data Encryption Standard. This is a symmetric-key block cipher. This accepts a key of length 56 bits, which turns out to be its salient feature. Encryption is done by this algorithm by converting the data into blocks of size 64 bits each, meaning, each 64 bits of the `data goes to the DES as input.

One of the notable features of the DES algorithm is its Feistel structure, named after cryptographer Horst Feistel. This structure involves multiple rounds of permutation and substitution, which enhances the security of the encryption process. Each round consists of key-dependent permutation and substitution functions applied to the data, increasing the complexity of deciphering the encrypted message without the corresponding key. While the Feistel structure contributes to DES's resilience against cryptographic attacks, it also underscores the algorithm's adaptability to different key lengths and block sizes, facilitating its integration into various cryptographic protocols and systems.

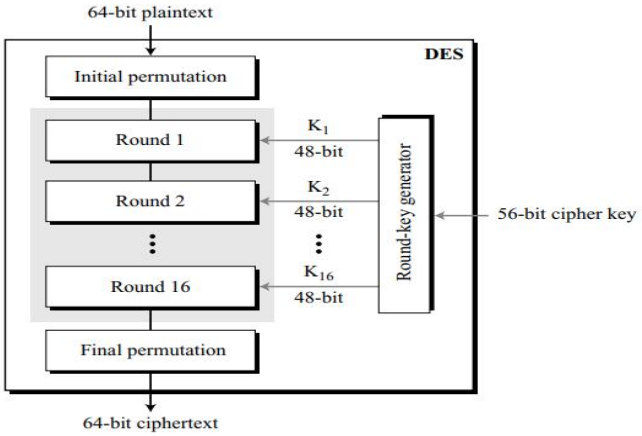


Fig. 1 DES Structure

DES initial and final permutations and 16 Feistel rounds. For each round, a new unique round key is generated from the original 56-bit key. However, normally a 64-bit key is provided initially, wherein the process of parity drop occurs which converts it to a 56-bit length. This parity drop process involves dropping every 8th bit in the 64-bit key and then permutating the bits according to the permutation table, which results in the 56-bit key which can be further processed by the round-key generator. The next step in the round-key generation is to divide the 56-bit key into two 28-bit halves respectively. Each part is shifted left circularly. The number of bits to be shifted depends upon the current round. For rounds 1,2,9 and 16, one bit is shifted and for the rest of the rounds, two bits are shifted. After this process is completed, another compression permutation takes place that compresses the 56-bit segment to 48-bit round key. Thus, we get the 48-bit key for a single particular round.

Each Feistel round takes input (L_{i-1} and R_{i-1}) from the previous round (or initial permutation box) and creates the 64-bit output (L_i and R_i) which goes in the next round. Each round has two elements, the swapper and the mixer. The swapper is invertible, which swaps the left half of the input with the right half. The mixer is invertible because it makes use of the XOR operation. All non-invertible elements are collected inside the function f(R_{i-1}, K_i).

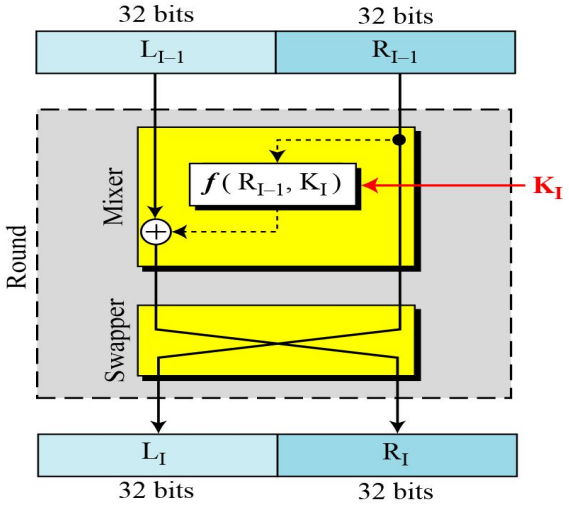


Fig. 1 DES Encryption

The DES function consists of four sections:

1) An expansion P-box
2) A whitener
3) A group of S-boxes
4) A straight P-Box

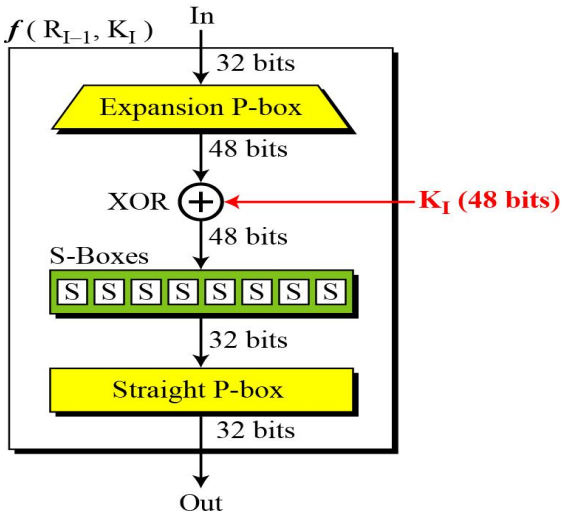


Fig. 3 DES Function

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue IX Sep 2024- Available at www.ijraset.com

III. TRIPLE DES

Triple DES or commonly known as 3DES is a type of DES encryption which is more secure than a simple DES or double DES Encryption. It uses three stages of encryption and decryption but in 3DES using two keys, it only uses 2 keys say K1 and K2 where K1 is repeated and in 3DES using three keys, it uses 3 keys K1, K2, K3 are used for 3 stages of the encryption.

A. Triple DES Encryption

This is the process of triple DES Encryption using 3 keys. So basically, the plain text which is to be encrypted initially goes through encryption using K1 key in the first stage, then the encrypted text goes through decryption using k2 key in the second stage and then finally again through encryption using the K3 key in the third stage. In this way, we obtain the final cipher text after 3 levels of encryption, decryption and then again, an encryption. To make triple DES compatible with single DES, the middle stage uses decryption in the encryption side and encryption in the decryption side. Triple DES makes it difficult for hackers and attackers to decrypt the cipher text as they have to go through multiple levels of encryption and decryption. For decryption in 3DES, all we have to do is reverse all the operations done during encryption of plain text. So, the cipher text which is encrypted initially goes through decryption using K1 key in the first stage, then the decrypted text goes through encryption using K2 in the second stage and finally a decryption again using key K3 in the third stage. After completion of all these steps we would get the initial plain text from the cipher text.

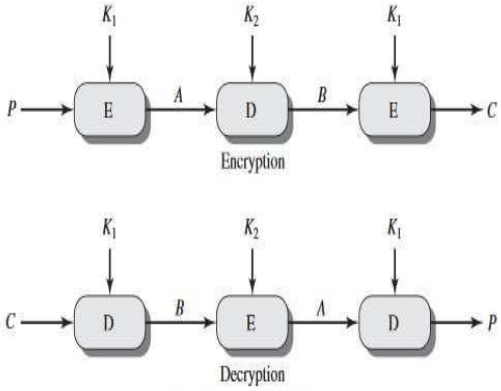


Fig. 4 Triple DES using 2 Keys

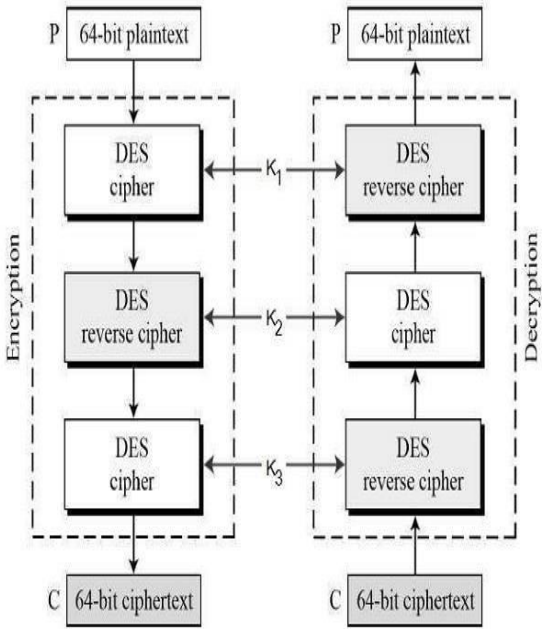


Fig. 5 Triple DES using 3 Keys

IV. IMAGE ENCRYPTION

Image Encryption is a process of converting an image into unreadable format by encrypting it using a secret key. Only a person who has the secret key can decrypt it back to the original image.

Some of the advantages of image encryption are:

1) Confidentiality: It protects the sensitive data on an image from unauthorized access.
2) Integrity: It also ensures that the image has not been overwritten or tampered with during its transmission from sender to receiver.
3) Authentication: Also verifies the sender's identity.

The disadvantages of image encryption are:

- Encryption and decryption processes can be very lengthy, especially if the images are large.
- The secret key should be securely stored and should only be shared between the sender and the intended receiver.

V. IMAGE ENCRYPTION USING TRIPLE DES

The python code implements a file encryption/decryption program using Triple DES (3DES) with the PyCryptodome library. It allows the user to select an operation (encryption or decryption), enter a file path, and provide a key.

A. *Image Encryption process in Python*

The python code implements a file encryption/decryption program using Triple DES (3DES) with the PyCryptodome library. It allows the user to select an operation (encryption or decryption), enter a file path, and provide a key.

Given below is the detailed explanation of the flow of code for image encryption:

1) *Imports:*

- DES3 from Crypto.Cipher: This is an important functionality to be imported from PyCryptodome library for the use of 3DES algorithm.
- MD5 from hashlib: Imports a function to create a message digest (hash) of the key.

2) *User Input:*

- The program asks the user if they want to encrypt or decrypt a file.
- The user inputs the specific location of the file on their device.
- Finally, a secret key is entered by the user, which is essential for both encryption and decryption in the image encryption process.

3) *Key Processing:*

- Now we use MD5 algorithm to convert the secret key entered by user into a hash of fixed length. The hashed key is now more secure and compatible with the encryption process. The encoded hash key is actually a 16-bit ASCII key produced with MD5 operation.
- The key parity of the hash key obtained from MD5 process is adjusted using the function DES3.adjust_key_parity() to produce the final triple DES key which meets the requirements of the 3DES algorithm.

4) *Encryption/Decryption Process:*

- So, then a cipher is created using final triple DES key generated in the above step integrated with MOD_EAX which provides confidentiality and authentication.
- A nonce is also used with it for generating random/ pseudo random number which is used for authentication protocol.

5) *File Processing:*

- The program opens the specified file.
- It reads the entire file into a single chunk of data.
- Depending on the chosen operation (encryption or decryption), the cipher processes the file content.

6) Output:

- The program opens the file again but allows writing to it.
- The encrypted/decrypted data is then written back to the original file, replacing the original content.
- Finally, a message indicates successful completion.

B. Source Code

```
from Crypto.Cipher import DES3
from hashlib import md5
while True:
    print('Choose operation to be done:\n\t1- Encryption\n\t2- Decryption')
    operation = input('Your Choice: ')
    if operation not in ['1', '2']:
        break
    file_path = input('File path: ')
    key = input('TDES key: ')
    hash_key = md5(key.encode('ascii')).digest()

    tdes_key = DES3.adjust_key_parity(hash_key)
    cipher = DES3.new(tdes_key, DES3.MODE_EAX, nonce=b'0')

    with open(file_path, 'rb') as input_file:
        file_bytes = input_file.read()
        if operation == '1':
            new_file_bytes = cipher.encrypt(file_bytes)
        else:
            new_file_bytes = cipher.decrypt(file_bytes)

    with open(file_path, 'wb') as output_file:
        output_file.write(new_file_bytes)
        print('Operation Done!')
        break
```

C. Code Output

Firstly, we need to have an image of any format (Ex. JPEG, PNG etc.) in our local machine.
Then we run the python code and give the file path of the image taken for encryption along with the secret triple DES key for encryption.

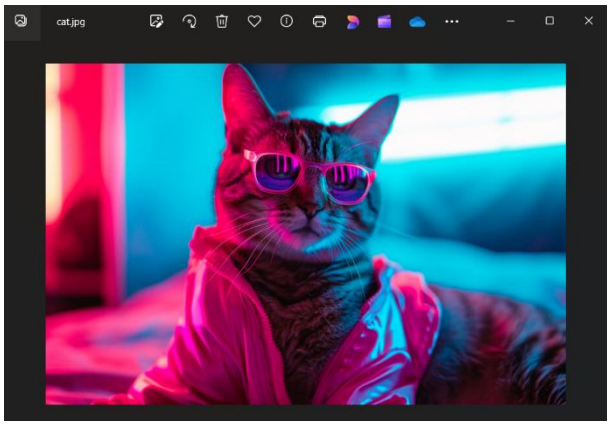


Fig. 6 Sample Image for Encryption

```
PS C:\Users\ANTON GEORGE\Downloads\Python_DES3-main\Python_DES3-main> python main.py
Choose operation to be done:
        1- Encryption
        2- Decryption
Your Choice: 1
File path: C:\Users\ANTON GEORGE\Desktop\cat.jpg
TDES key: 1234
Operation Done!
```

Fig. 7 Implementation of Encryption Process

Now, if we try to open the sample image, it would be encrypted and the system does not support its file format so the image is not displayed. It is because an image is displayed when its bits are stored in the correct format, but after the encryption process, the bits of the image are scrambled due to which the image is not shown.

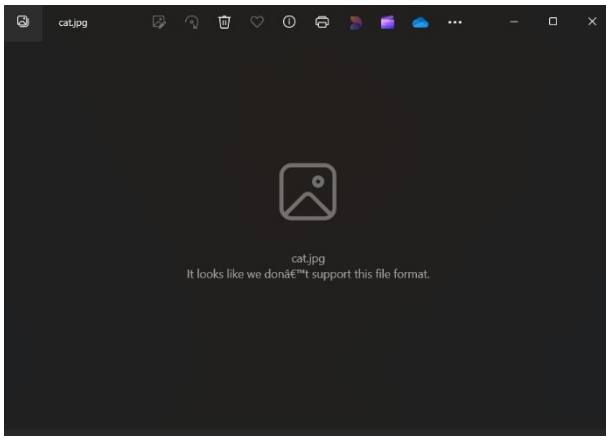


Fig. 8 Image after Encryption

Now to decrypt the image, we need to run the same python code again, enter the file path of the image and provide the same secret triple DES key which was entered during encryption.

```
PS C:\Users\ANTON GEORGE\Downloads\Python_DES3-main\Python_DES3-main> python main.py
Choose operation to be done:
        1- Encryption
        2- Decryption
Your Choice: 2
File path: C:\Users\ANTON GEORGE\Desktop\cat.jpg
TDES key: 1234
Operation Done!
```

Fig. 8 Implementation of Decryption process

VI. CONCLUSIONS

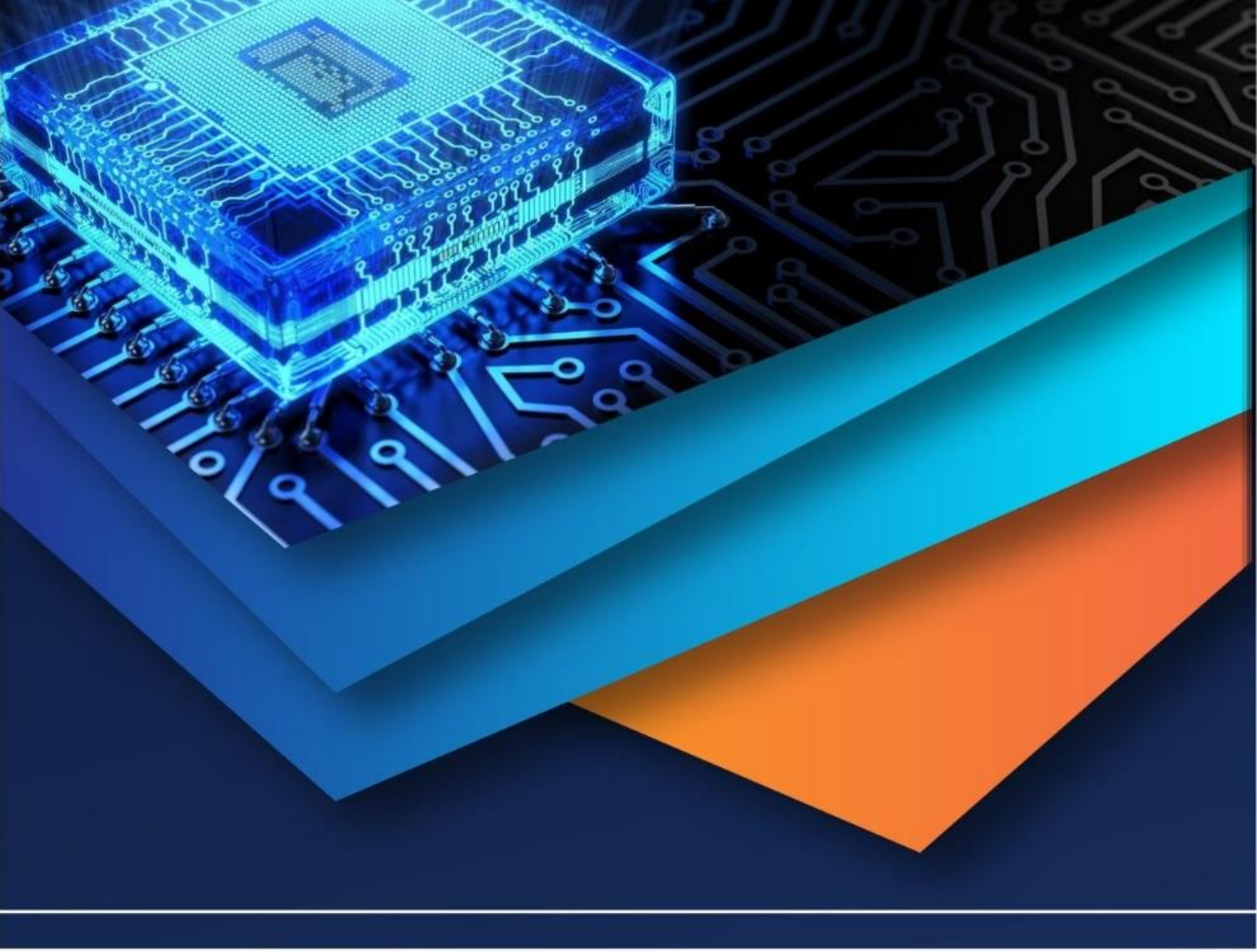
This paper conclusively demonstrates that the Triple Data Encryption Standard (3DES) algorithm is a highly effective method for securing image data against unauthorized access, providing robust protection through its triple-layered encryption technique. Despite its complex encryption process, 3DES maintains a balance between security and efficiency, making it practical for real-world applications. Its compatibility with current technological infrastructures ensures a smooth transition to more secure data handling practices.

The algorithm allows for seamless integration into existing systems, ensuring the confidentiality and integrity of sensitive information. As cyber threats evolve, the foundational security principles of 3DES will continue to be essential for the development of future encryption technologies.

REFERENCES

[1] Srivatsava, J. and Sheeja, R., 2020. Implementation of triple des algorithm in data hiding and image encryption techniques. Int J Adv Sci Technol, 29(3), pp.10549-10559.

[2] Vadlamudi, D., Kumar, R.J. and Sai, C.N., 2022, July. Image Encryption using Reverse Data Hiding Algorithm with Triple DES. In 2022 International Conference on Inventive Computation Technologies (ICICT) (pp. 36-41). IEEE.
[3] Peram, S.R., Neeraj, M. and Kumar, B.A., 2022. Analysis of image security by triple DES. Materials Today: Proceedings, 64, pp.808-813.
[4] Rao, M.G., Cenitta, D., Arjunan, R.V. and Babu, D.V., 2024, May. Securing Image using Triple Data Encryption Standard. In 2024 3rd International Conference on Artificial Intelligence For Internet of Things (AIIoT) (pp. 1-6). IEEE.
[5] Mohammad, O.F., Rahim, M.S.M., Zeebaree, S.R.M. and Ahmed, F.Y., 2017. A survey and analysis of the image encryption methods. International Journal of Applied Engineering Research, 12(23), pp.13265-13280.
[6] El-Zoghdy, S.F., Nada, Y.A. and Abdo, A.A., 2011. How good is the DES algorithm in image ciphering. International Journal of Advanced Networking and Applications, 2(5), pp.796-803.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY