

# IMAGE ENCRYPTION TECHNIQUES AND COMPARATIVE ANALYSIS

Shaolin Kataria<sup>1</sup>, Elio Jordan Lopes<sup>2</sup>, Bevis Mathew Niravil<sup>3</sup>, Shashank Keshav<sup>4</sup>

<sup>1</sup>BTECH IT Undergrad, School of Information Technology & Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, India

<sup>2,3,4</sup>BTECH CSE Undergrad, School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, India

\*\*\*

**Abstract** - Encryption is important tool for ensuring the security of multimedia data. All kinds of multimedia files circulate and travel across the web after going through scrambling. It is difficult for anyone to obtain the keys due to the file's encryption. In order to shield the data by anyone other than the person who's meant to receive it, Encryption techniques are adopted. Other than this person, the key to decrypt the data should not be in possession of any other individual. This is done by transforming the key into such a shape that can not be understood by any aggressor. Information encryption mainly involves substituting the information. This piece of information can be anything ranging from text snippets to images to even sounds. The key objective of encrypting such data is to veil it in a manner such that it cannot be recognized or detected making it less vulnerable amid the transmission. In order to return back to the specified end goal in mind which is the initial information that was encrypted, the receiver of the encrypted file inverses the process of encryption, which is known as decryption of data. The topic that we're now tackling for this project work is to learn, comprehend, and become familiar with the significant cryptography algorithms that are connected to data protection, especially the documents which are transferred between two parties back and forth over the web. For this project, we will be studying different encryption and decryption algorithms like DES, AES and RSA algorithms. We'll examine the working of these encryption techniques, as well as how they function and are executed on various sorts of data, and how they produce the necessary results in various time periods.

**Key Words:** Encryption, Decryption, Cryptography, Algorithms, DES, AES, RSA

## 1. INTRODUCTION

Algorithms used in encryption/ decryption can be compared using attributes or parameters like key length, Algorithm complexity (encryption/decryption, key setup) and best methods of attack like steps and time required to execute an attack. For research purposes and the academia, and students, there is a need to develop an application that can help them study and compare and learn and understand the working of different algorithm using the same file(data size) considering parameters most importantly time for comparing the techniques. So

we decided to develop and simple tool to help one understand the working of these algorithms and compare them based on certain parameters that we have provided in our detailed analysis. This application would also allow us to verify if a particular file has been encrypted or decrypted by conducting an integrity check to check if the file hashes work. Encryption and decryption are vital because they allow you to protect data that you don't want others to see, such as emails, chat history, tax information, credit card numbers, and other sensitive data. There is a need to create awareness about the various techniques used for the encryption process as there is a lot of tampering with data taking place during this times leading to data breaches and leaks.

### 1.1 Scope

Cryptography plays a very important role when different users exchange data and information. Users that exchange media files are vulnerable to various security concerns. This includes data leakage and data theft. Usually multimedia contents take up a lot of space. Encryption technique used in the exchange of information should be time efficient. In this work we consider three encryption techniques: AES, DES and RSA over image encryption. Cryptography or encryption, commonly known as "secret writing," is a technology that seeks disguising content so only the recipients for which the data is meant have accessibility to it. Active and passive attacks in a network channel might jeopardize the privacy of data which can also be subject to modification. Encrypting the data ensures its privacy in such cases. Cryptography basically comprises two things which are Plain Text & Cipher Text. The information intended for the receiver is referred to as the Plain Text whereas, CipherText is the encrypted version of the Plain Text. The interconversion between plaintext and ciphertext is done thanks to Encryption and Decryption algorithms.

There are two main classifications of Encryption and Decryption algorithms namely Symmetric Key Algorithm & Asymmetric Key Algorithm. In this paper, two out of the three encryption and decryption techniques we will be using are Symmetric Key Algorithms while the remaining one is an Asymmetric Key Algorithm. In traditional end-to-end encryption techniques, only the receiver can verify and validate the integrity of the file received in the exchange. This created many complications and problems, one of which is that it would be rendered ineffective if the

network architecture is ever under any kind of attack and fails to deliver data or information safely. Secondly, these algorithms were not enough for a secure transfer and exchange of information on their own. A majority of internet traffic consists of spam emails or other useless data of which the attackers are seldom caught. As a result, we believe there is a strong necessity for secure solutions that apply policies and procedures at each hop of the packet's journey through all the networks to ensure security. If the verification and validation of the data or information can be administered by the network architecture itself, counter measures can be taken within the network for any such attacks and it won't fall on the receiver alone. It could allow threats and attacks to be prevented faster and more effectively, increasing the probability of capturing the attackers.

In order to shield the data by anyone other than the person who's meant to receive it, Encryption techniques are adopted. Other than this person, the key to decrypt the data should not be in possession of any other individual. This is done by transforming the key into such a shape that can not be understood by any aggressor. Information encryption mainly involves substituting the information. This piece of information can be anything ranging from text snippets to images to even sounds. The key objective of encrypting such data is to veil it in a manner such that it cannot be recognized or detected making it less vulnerable amid the transmission. In order to return back to the specified end goal in mind which is the initial information that was encrypted, the receiver of the encrypted file inverses the process of encryption, which is known as decryption of data.

The process of encryption can be determined by

$$\rightarrow C = E(P,K)$$

where,

P = Original data

E = Encryption Algorithm

K = Encryption Key

C= Cipher text, which is communicated through a network and can be subject to a threat.

The process of encryption can be determined by

$$\rightarrow P = D(C, K)$$

where,

C = Cipher message

D= Decryption Algorithm

K= Decryption Key

P= Recovered data

## 1.2 Novelty

We implemented a single application which would include all 3 algorithms together. From only one software application one can use any algorithm (RSA,DES and RSA)

to encrypt the file file. Moreover we also did an extensive analysis of these algorithms at last giving us the exact details and in-depth knowledge of these encryption techniques

## 2. LITERATURE REVIEW

For our project, we did a literature survey of 20 latest journal papers relevant to our project topic. One paper did a Performance comparison of encryption algorithms where the authors evaluated the performance using two metrics: computing time and the memory usage. They achieved a result which showed that the RSA algorithm took the longest time to finish making it the slowest algorithm in terms of computational time and the fastest was found to be the Blowfish algorithm followed by AES and 3DES [1]. Another paper also did a detailed study of encryption algorithms and found the AES algorithm to be the most efficient out of the four considering constraints like throughput, time, avalanche effect and speed [2]. One paper did such a comparison of algorithms that enforce security in the domain of Cloud Computing. The authors performed a survey on research papers of the same domain and presented a comparison of some encryption techniques. Furthermore, they also suggested some management techniques for the security of cloud computing. The authors found that the performance of DES algorithm was poor when compared to other algorithms and thus has the lowest security [3]. In another paper, the combination of three types of encryption has been put forth to understand the merits of of each one. The research aims at a conclusive high security system. The chosen algorithms to build this hybrid structure were AES, RSA and HMAC. The overall encryption they achieved was meteoric with feeble hardware requirement with maximum security [4]. One of the journal publications proposed the creation and design of a crypto processor, which is a particularly unique microprocessor designed for the executing encryption algorithms. Storage devices, network routers, embedded devices, security gateways using the IPSec and SSL standards, and other security applications could all benefit from the specially introduced crypto processor [5]. One paper presented a detailed literature survey of many well-known Cryptography techniques such as AES, DES, 3DES and RSA, surveyed on the prevalent techniques pertaining to encryption. Various different cryptographic techniques were used to maintain security for any data transmissions in a network were used [6]. In one of the papers, various cryptographic algorithms based on bit-level shuffling and switching, chaotic mapping, compressed sensing, and DNA coding were presented. The comparison of several cryptographic algorithms was proposed in this study. was discussed that has been carried out on key space and speed. The key space of the mentioned techniques was found to be large enough to resist any attack and reliable for providing security. Also, Because of its exceptional properties, DNA Encryption outperformed when evaluated due to its computational rate.[7]. In one of the journal papers, a new altered and better version of AES Encryption algorithm was proposed to implement a safe

and symmetric image cryptography technique. To include a bit stream generator for image cryptography, the AES Encryption algorithm has been enhanced to solve the issue of textured regions that plagues other cryptographic techniques. The outcomes of the analysis done and execution were reported in detail. A comparison with typical cryptographic algorithms too revealed the altered algorithm's advantage.[8]. In another paper the author discussed how AES and DES algorithms work in ATMs. This article compared the AES and DES encryption algorithm, demonstrating how well the DES approach is employed in ATMs and how AES is much more reliable than DES. Therefore the authors suggest that in every ATM, AES algorithm should be put to use instead of DES. Here, the ATM will use a PIN which will act as a secret key. This PIN will be used to get hold of the PIN from respective bank account numbers. As a consequence, a natural PIN was created, to which an offset may be applied before the completed PIN was provided by the consumer. The offset serves no encryption purpose; it was only introduced to allow customers to select their own Passcode. [9]. In one paper the author discussed the security of information in the cloud. Propagated enrolling proved to be a very efficient way for alliances. People are saving their personal and basic data to fogs, and maintaining that information safely has become a critical concern. Numerous algorithms exist for information security like DES, AES. These are symmetric cryptographic methods. Each packet has a range of hash values. However, there is a drawback to hashing: once the data is intermingled, it cannot be reassembled in the same fashion it used to be. This constraint of hashing was overcome by the use of Symmetric Key algorithms [10]. One paper presented the simultaneous functioning of symmetric and asymmetric algorithms when it comes to encryption. They deduced that symmetric were significantly faster in performance [11]. It was determined that the AES approach employs the least amount of encryption time while the Asymmetric encryption approach uses much more encryption time.[12]. Another research proposed a method for lowering the computation complexity of multi-media cryptography while retaining compressed video characteristics. The authors' experiment suggests how a Secure Wavelet Transform and chaotic arithmetic coding, which are two blocks occupied for compression, meant for purposes like coding of videos, can be used for video encryption. The proposed systems were used to provide research findings for selected cryptography techniques [13]. Cloud computing Techniques and fundamental concepts of multimedia cloud computing were studied in detail. Proposed work of the Security method of RSA and DES, characteristics of multimedia cloud computing were listed and an analysis of the RSA Algorithm & DES Algorithm was done. A mixed approach was employed to encrypt lengthy information, wherein the messages were encoded using symmetric techniques (AES, DES, etc.) and the key was sent using asymmetric algorithms (RSA)[14]. Another of the journal publications performed a research of the available studies on cryptography algorithms. Each methodology was discovered to be distinct in its own right, suited for a

multitude of scenarios and with its own set of benefits and drawbacks. As per literature and research survey, the AES cryptography technique is the most efficient whether it is overhead, throughput, or length of key, block size and rounds.[15]. The researchers of one of the studies offered a new double tier cryptography technology that allows for minimal visual resemblance and great protection without being heavily disadvantaged in terms of speed-encryption ratio. The achieved an encryption ratio for PN and RSA technique rounded off to around sixteen percent, in contrast with the figures produced by the RSA technique of around fifty percent. For both the approaches, the authors found the decryption to be the same, which is 36%. The advantage of this method is because it has a modest consequence in terms of encryption and decryption time.[16]. A review of various cryptographic algorithms used to encrypt images was done in another one of the research papers. They specifically focused on AES and DES encryption algorithms. At the end of the paper they were able to derive the weakness and strengths of each algorithm. They evaluated the time taken and type of keys used in these encryption techniques. Moreover, they also analyzed which algorithm requires more processing [17]. RSA and ElGamal family's public key sizes were discussed. Both ensembles were taken into account, the classic and the elliptic. The usefulness of the generated systems was discussed. They found out the practicality of matching AES security using public keys [18]. The authors of one of the journal papers implemented and observed Speed and storage parameters are used to achieve multi level cryptography utilizing the Data Encryption Standard (DES) and multi-prime RSA, a tweaked and upgraded form of the RSA Cryptosystem. The parameters were represented against their average values, by making use of distinct primes and the representation of the results was done graphically and in tabular form, to provide further surety of conclusions. The paper also lists the benefits and justifications for employing this method [19]. In one of the papers, the authors presented a non-invasive, cheap and successful method for injecting defects into an ARM9 broad-purpose CPU by reducing the input power. They thoroughly define both the stress paradigm and the mistakes introduced during calculation, with respect of the resulting rate and corruption trends on the results obtained. Using the techniques and methods known in open literature, the authors of this paper validated and verified the effectiveness of the fault model that was proposed which involved leading with practical attacks and then implementing AES and RSA cryptography algorithms [20].

### 3. METHODOLOGY

Cryptography techniques have numerous variants, with unique topology and provides secure data communication via network interfaces, as well as ensuring authentication and privacy. The corresponding OSI layers PC programme must implement all of these end-to-end encryption and decryption procedures.

A single key is used for both encrypting and decrypting in the case of a **Secret Key** also referred to as a **Symmetric Key**.

**Public Key (Asymmetric Key)** In this case, the keys used are distinct, the public one being used for encryption while the private one takes care of decryption..

**AES algorithm**

The AES algorithm handles any combination of data(128 bits) using 128, 192, and 256 bit encryption. It goes through 10 rounds, 12 rounds and 14 rounds for 128, 192, 256 bit keys respectively. It breaks down 128 bit length data to be broken down into operational chunks, as these are organized into an array of bytes in a matrix of 44 dimensions. It starts with the Round Key stage. Before arriving at the final round, it traverses 9 main rounds in four alterations, performed in different phases namely, subbytes, mix columns and round key.

Techniques used for Decryption are vast and deft. Some of those techniques perform substitution of bytes in an inverse manner, and the shifting of rows and columns in an inverse manner.

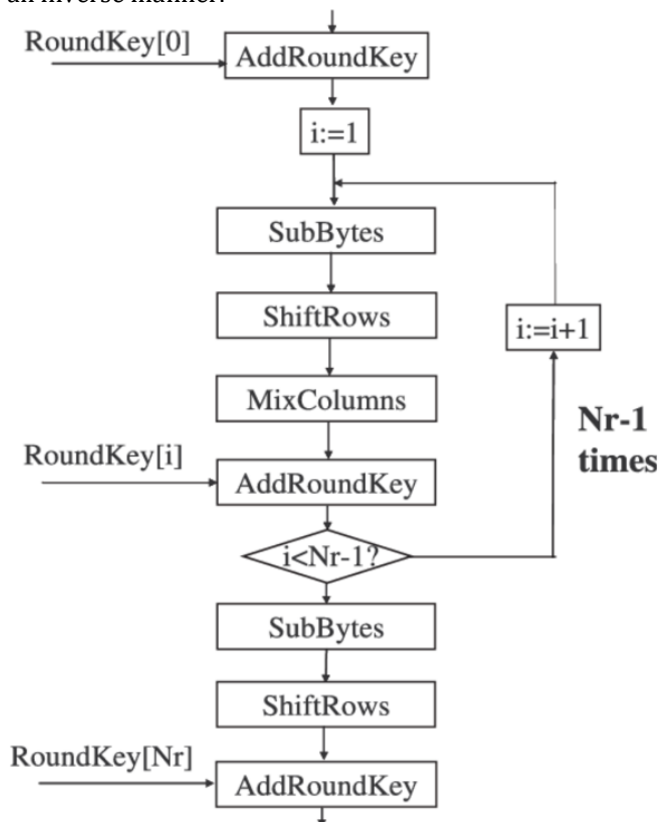


Fig -1: Low-Level Diagram of the AES Algorithm

**DES Algorithm**

DES is arguably the most well known and widely available architecture for cryptography. It has a 56 bit length, 64 bit block length. Usually prone to assault when a weak key is used. It commenced with a 64 bit digit key, but was limited to 56 bit keys by the NSA. Therefore, it discards 8 bit data

from the 64 bit key length and make uses of the 56 bit key to encrypt data in blocks of 64 bit

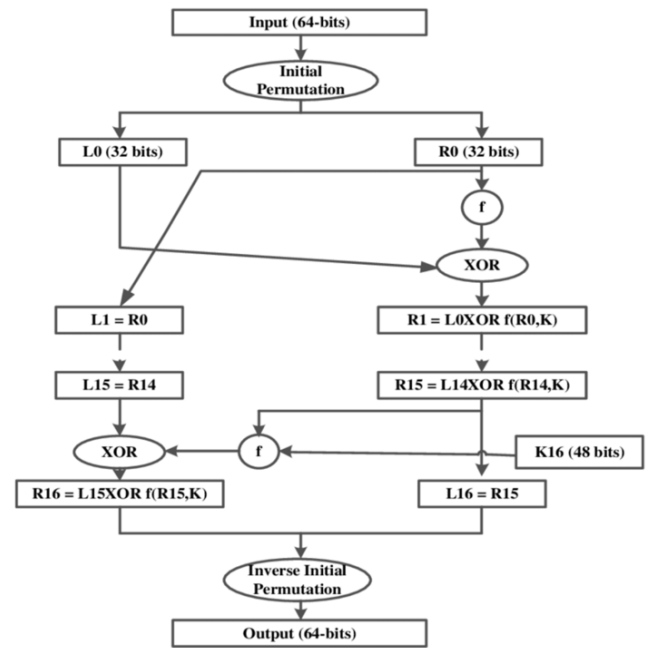


Fig -2: Low-Level Diagram of the DES Algorithm

**RSA Algorithm**

RSA is a cryptographic algorithm that is asymmetric. It stands out when compared to its contemporaries for a lot of factors. The factors include exchanging of keys, marks given by the computer, encryption of blocks, based on variable-sized or independent alike. It creates people using two indivisible numbers and private keys. The sizes range is 1024-4096 bits. This is how encryption and decryption is achieved. The encryption of the intended message that the sender wishes to send is done using the public key possessed by the receiver. The data decryption of the encrypted data sent by the sender is done using the private key possessed by the receiver.

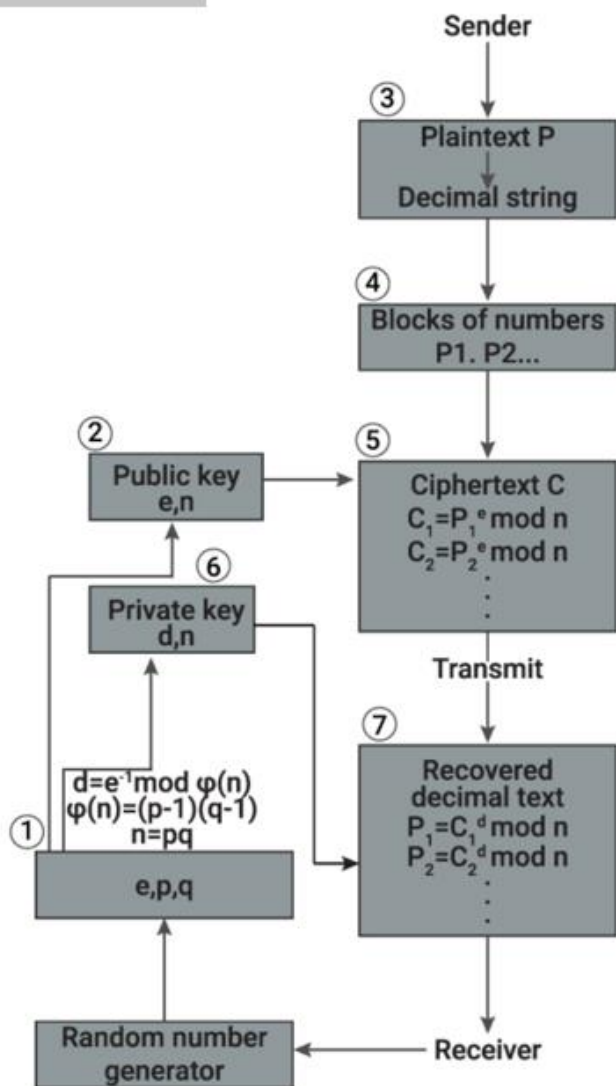


Fig -3: Low-Level Diagram of the RSA

### 3.1 Test-bed

#### Algorithm for Encryption Tool

**Step 1:** Execute program

**Step 2:** Select an encryption algorithm

**Step 3:** Add File/Directory for Encryption

**Step 4:** File/Directory will be successfully encryption

**Step 5:** Browse the encrypted file to decrypt it

**Step 6:** Click on Decrypt to Decrypt the image in destination folder

**Step 7:** Do an Integrity Check of Hash Value. By selecting the location for original file and decrypted file

**Step 8:** Integrity will be Verified!

We developed a simple Graphical User Interface with an easy to understand User Experience that would

more. Qt has a set of multi-platform C++ libraries that provide high-level APIs for interfacing with

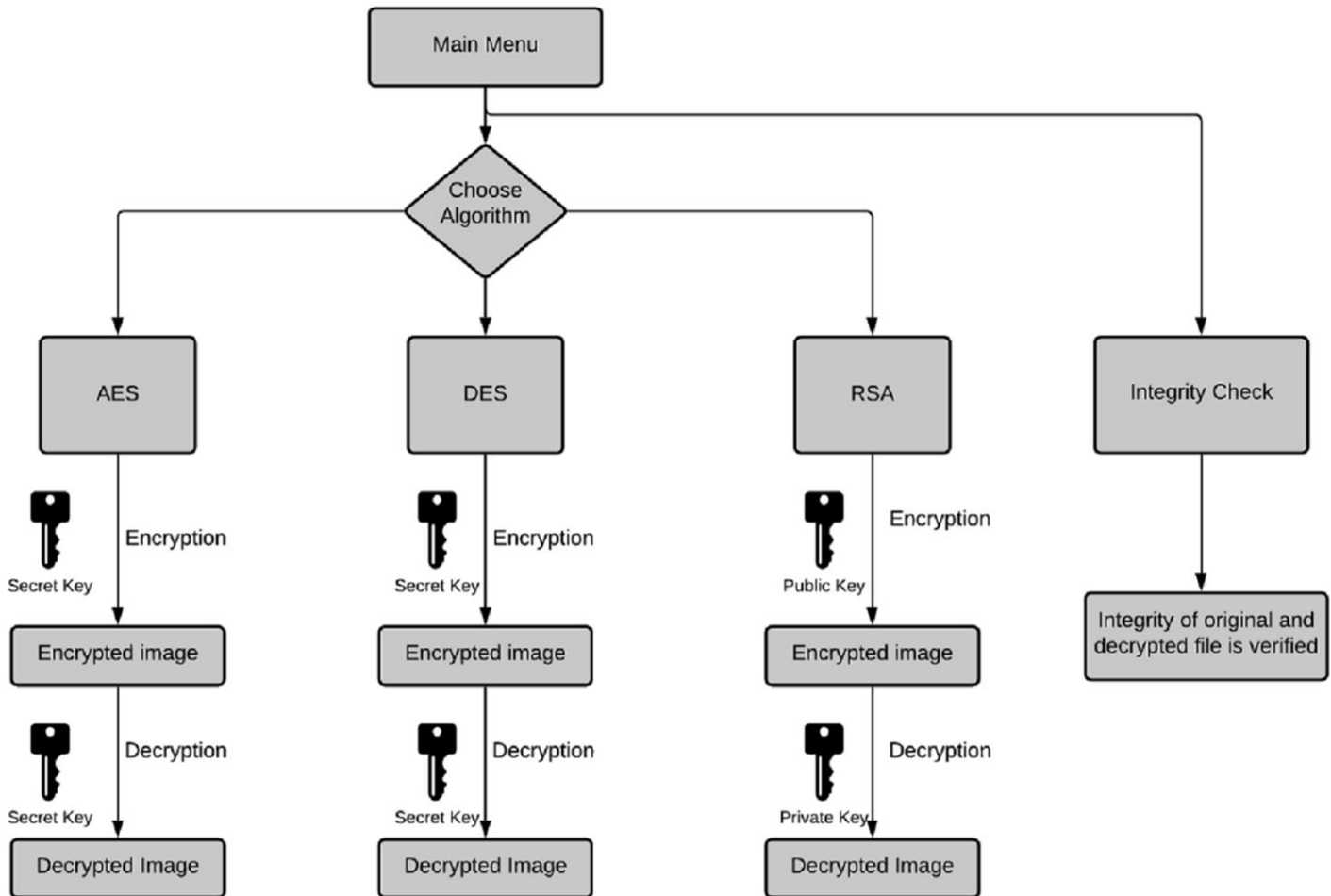


Fig -4: High Level Diagram of the Architecture

assist a user in studying and conducting encryption and decryption of images and also Verify the integrity of the image after the encryption and decryption. We used Qt for python which is a project providing an official set of Python bindings which one can use for building simple and effective UI for projects. It has two main components:

- PySide6, to use Qt6 APIs in Python applications, and
- ShibToken, a binding generator tool, to expose C++ projects to Python environment

It features a list of comprehensive and easy to understand modules and provides support for building projects involving security, machine learning applications, CRUD applications and much

modern desktop. Services that adhere to location, and multimedia connections like Bluetooth and NFC are covered as well for a traditional user interface design are just a few of the features available. PyQt5 is a collection of Python bindings for Qt 5. There are about 35 extensions in total.

It comes with over 35 extension modules that make Python a viable alternative to C++ for application development on all supported platforms, including

### 3.2.1. Python Cryptography Toolkit (pycrypto)

This package includes secure hash methods like SHA256 and RIPEMD160, as well as several encryption schemes (AES, DES, RSA, and ElGamal). The package is set up in such a way that adding other modules is trivial. This section is virtually finished, and the software interface will not change in an

incompatible way in the future; all that is left is to fix any bugs that may develop.

Clients and servers can encrypt data and authenticate one another, while daemons can encrypt important information for added protection.

### 3.2.2. Fast Base64 implementation

Seeks to provide a rapid base64 implementation with encoding using base64

### 3.2.3. OAEP PKCS#1 (RSA)

The PKCS#1 cypher uses RSA and OAEP padding and is asymmetric. The name of the protocol given in RFC8017, RSAES-OAEP, is capable of encrypting communications up to the length of the RSA modulus. Using the recipient's public key, we can encrypt data (assumed to be stored locally in the public.pem file).

### 3.2.4. Crypto.Cipher package

The purpose of this package is to withhold the integrity of the data.

The different types of encryption algorithms follow:

**1. Symmetric ciphers** are those in which all parties decrypt and encrypt data using the same key. Symmetric ciphers are typically robust and can handle an abundance of data.

**2.** The characteristic of asymmetric ciphers is that the parties involved use unique keys. The sender has its public key unconcealed and that is used for data encryption, whereas the receiver has a concealed private key used for data decryption. Ciphers of Asymmetric nature are notorious for their sluggishness and limited payload capacity. PKCS#1 OAEP is an example (RSA).

**3. Hybrid ciphers:** : The fusion of two sorts of ciphers mentioned above can be beneficial. A symmetric cypher encrypts the real data while an asymmetric cypher secures a short-lived symmetric key (under that key).

To build a cypher object, use the new() method from the appropriate cypher module (e.g. Crypto.Cipher.AES.new()).

The argument that comes first usually constitutes the cryptographic key; its length varies depending on the encryption. To encrypt data, we utilize the plaintext to call the cypher object's encrypt() function.

This technique will return the ciphertext. With the ciphertext, you invoke the decrypt() method of the cypher object to decode data. The plaintext is returned by the procedure. Both functions can accept OUTPUT parameters.

There are two types of symmetric ciphers:

● **Stream ciphers:** These are referred to as natural ciphers since the processing of encrypted data is done in chunks.

● **Block ciphers** are ciphers that can only handle a certain quantity of data at a given time. AES is the most extensively used block encryption (16 bytes). In general, a block cypher can only be used when it's paired with an encryption mode that allows for a configurable amount of data to be encrypted. A block cypher can be converted to a stream cypher using several modes, such as CTR.

Ciphers that merely provide confidentiality without any sort of authentication are widely regarded as bad. Instead, primitives for symmetric encryption and authentication have been defined (MAC).

## 3.3 Graphical User Interface

The QtGui module uses QtCore with GUI functionality.

● A single colour QPixmap is a programme that allows you to map bits of (1-bit depth)

● Device-independent colour maps with QColorGroup and QColorMap.

● QColor converts device-dependent pixel values to QColor.

● QHBoxLayout: Horizontally aligns widgets.

● QImage is a hardware-independent image representation.

● QImageReader is a multi-format image reader that can read images from files as well as other sources.

● QImageWriter is a format-agnostic photo-saving interface for files and other media.

● QDialog: A simple dialogue for asking the user for a single value.

● QListWidget: Manages information about a model's selected items.

● QProgressBar: A progress bar that can be horizontal or vertical.

● QScreen is a Qt for Linux base class for screen drivers.

### 3.3.1 Functions involved

**add\_file(self), remove\_file(self)** To add and remove files from the user interfaces, are used. Jpeg and png are acceptable image formats. The Encryption feature is enabled once the image has been uploaded. The Encryption boolean is set to false when a file is removed.

**selection\_changed(self)** is a toggle switch that allows you to choose between encryption and decryption.

After executing the `encrypt(self)` function, pick the Encrypt Button to open a popup to choose/create a file that will be our encrypted file. It creates an RSA public key after requesting a public key for encryption. The picture can be decrypted using the recipient's private key once it has been encrypted. Then the pairing of public and private key needs to be done. The bit key length to be assigned is 4096 and that estimates to 512 bytes.

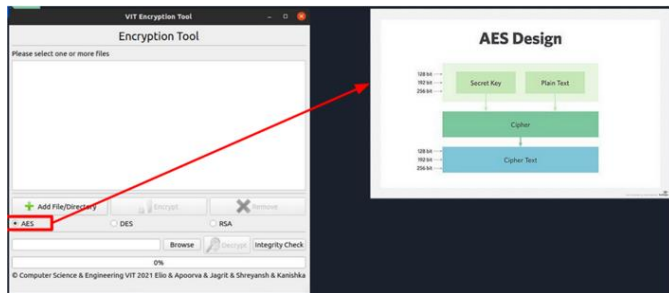
AES and DES, for example, are symmetric algorithms that use a 56-bit key to conduct encryption.

`verify(self)` performs an integrity check by comparing the file hashes of the encrypted image with the decrypted image. If the check succeeds, a success message is returned; otherwise, the check fails.

Python time is what we utilize.

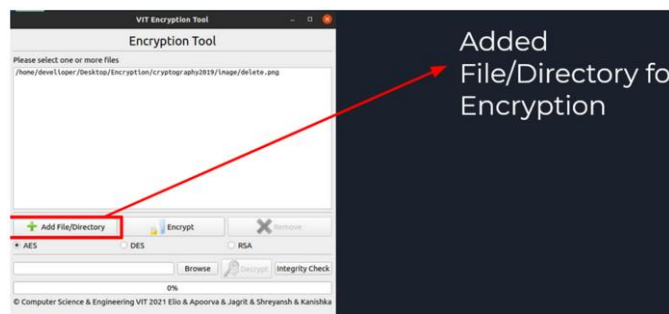
The `time()` function returns the number of seconds since the beginning of time.

#### 4. RESULTS AND DISCUSSIONS



**Fig -5:** Toggle between encryption techniques using the interface

On the user interface the user can toggle between different modes of Encryption and Decryption. In the above figure Fig.5, we have enabled the AES encryption.

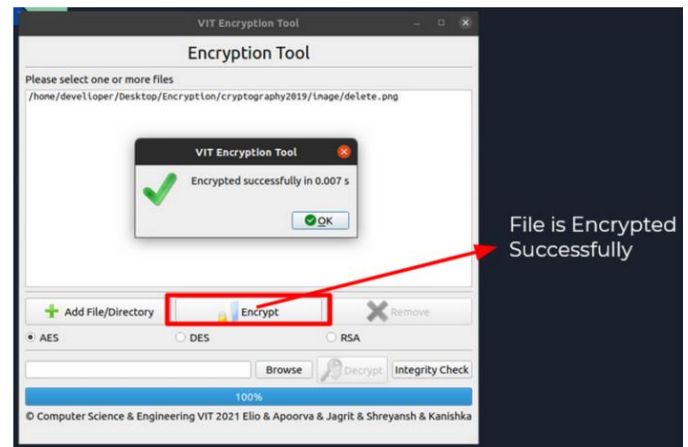


**Fig -6:** Added file appears in the Queue

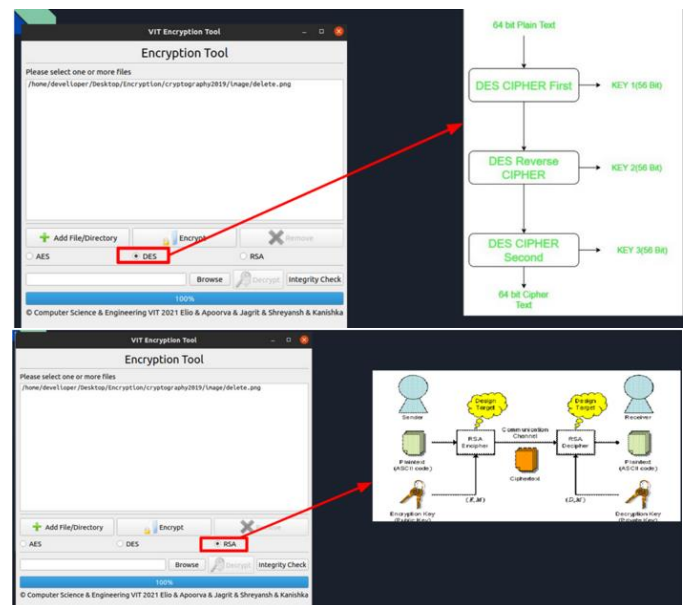
Files are added in the User Interface and get queued up in the File Queue as shown in the Fig. 6.

So the time for the encryption is calculated using python's packages and the GUI returns those results and the files are stored as binary in the compiled folder. The same file can be used to perform decryption followed by integrity checks.

We can use the selection buttons to do experiments with RSA and DES encryptions. In the similar way we get the encrypted binary files stored on the compiled directory, which can be later used for decryption.

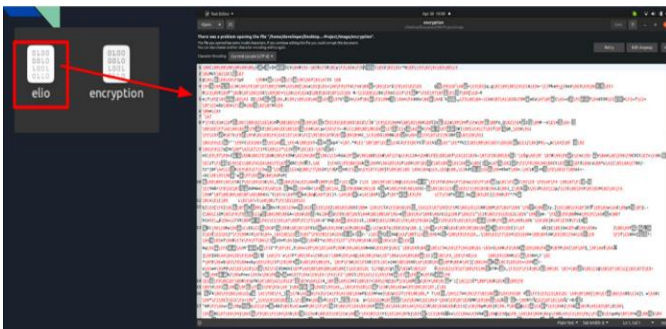


**Fig -7:** Result of the encrypted image along with the time taken

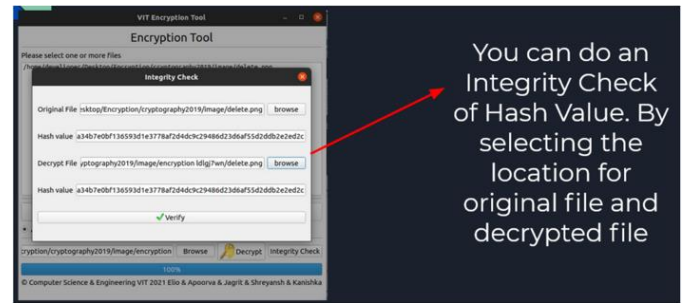


**Fig -8:** Similar experiments can be performed for DES and RSA algorithms.

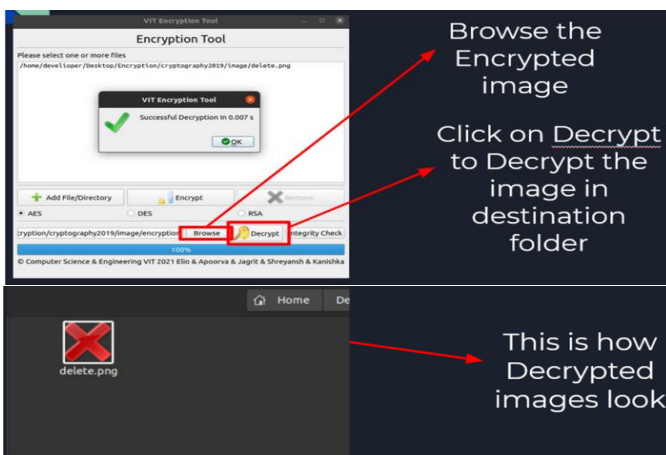




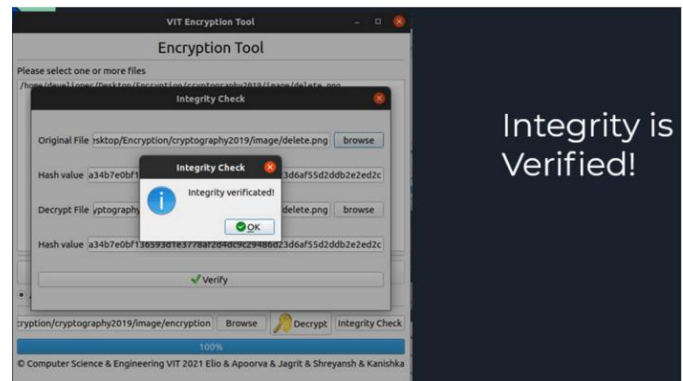
**Fig -9:** Encrypted Files are stored inside a compiled directory



**Fig -11:** Decryption is done on the compiled binary file and the decrypted file appears in a separated directory



**Fig -10:** Decryption is done on the compiled binary file and the decrypted file appears in a separated directory

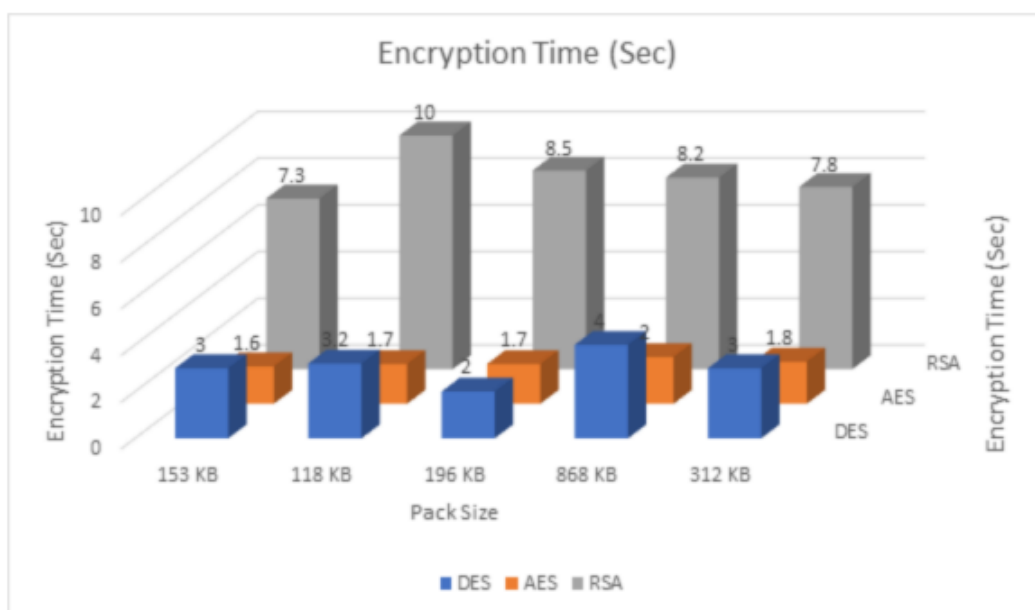


**Fig -12:** Integrity check pass result

In the Fig-12, we do an integrity check to determine if the decrypted image hash matches with the hash of original image. Python has hashing packages that lets you get the hash of every image. If the hash image check passes, it returns a successful output, else it displays the output as “failed”,

Factors	RSA	DES	AES
Created By	Ron Rivest, Adi Shamir, and Leonard Adleman in 1978	IBM in 1975	Vincent Rijmen, Joan Daemen in 2001
Key Length	Depends on the number of bits on modulus n where $n = p \cdot q$	56 bits	128,192 or 256 bits
Rounds	1	16	10- 128 bits key, 12 - 192 bits key, 14- 256 bits key
Block Size	Variable	64 bits	128 bits
Cipher Type	Asymmetric Block Cipher	Symmetric Block Cipher	Symmetric Block Cipher
Speed	Slowest	Slow	Fast
Security	Least Secure	Not Secure Enough	Excellent Security
Memory Utilisation	Requires most memory space	Requires Least Memory Space	Requires Moderate Memory Space
Power Consumption	High	Low	Low
Computational Power	Faster	Moderate	Faster

**Table -1:** Comparative Analysis of DES v/s AES v/s RSA based on Key Length, Speed, Rounds, Block Size, Cipher Type, Security, Power Consumption, Computational Power and Memory Utilization



**Chart -1:** Comparison based on Encryption Time



Chart -2: Comparison based on Decryption Time

Table -2: Comparative Analysis of DES v/s AES v/s RSA based on Encryption Time, Decryption Time and Buffer Size

S. No.	Algo	Pack Size (KB)	Encrypt Time (Sec)	Decrypt Time (Sec)	Buff Size
1	DES	153	3.0	1	157
	AES		1.6	1.1	152
	RSA		7.3	4.9	222
2	DES	118	3.2	1.2	121
	AES		1.7	1.2	110
	RSA		10.0	5.0	188
3	DES	196	2.0	1.4	201
	AES		1.7	1.24	200
	RSA		8.5	5.9	257
4	DES	868	4.0	1.8	888
	AES		2.0	1.2	889
	RSA		8.2	5.1	934
5	DES	312	3.0	1.6	319
	AES		1.8	1.3	300
	RSA		7.8	5.1	416

## 5. CONCLUSIONS

Using the AES, DES, and RSA algorithms, we were able to securely encrypt files. We considered adding integrity checks to our programme while implementing the project. This software's integrity check feature allows the user to check themselves and compare the encrypted and decrypted files using a hash value. Furthermore, throughout our comparative investigation, we discovered a distinct pattern: the AES algorithm proves to be superior in terms of encrypting and decryption time. The AES algorithm takes roughly an order of magnitude less time than the RSA algorithm. In terms of security, AES is regarded as the finest algorithm once again. This is because the output of each round in AES algorithm is involved in the input of another round, making AES one of the hardest algorithm to hack.

### 5.1 Security Analysis

Along with encryption and decryption time it is important for the algorithm to be secure. The better the algorithms, the better the level of security it has. Here is the analysis of the security level of various algorithms.

1. **AES:** Because it employs variable length key bits, AES provides a high level of security. Working like RSA modulo arithmetic procedures, although it can be inverted theoretically. The security of an algorithm is measured by time taken to crack and the cost taken by the attacker to find the key to encryption. Various attacks were tried on AES for instance square attack, differential attack and key attack but none of them were able to decode AES.

2. **DES:** 56 bit key length of DES is a major problem in terms of security for DES. Only attack that is possible on DES is brute force attack with a parallel machine of nodes about 2000 combined together, which is capable of key search at the rate of 50 million keys/sec. Further one can perform cryptanalysis by exploiting the characteristics of DES. The weak S boxes in the algorithm provide a possible means of cryptanalytic attack.

3. **RSA:** RSA's security is based on a huge set of numbers and the composite number n's root modulus. In order to ascertain the security level, get a constant m such that  $C \text{ equals } m \pmod{n}$ , in which (n,e) is available publicly and C is the cypher, which can be decrypted using standard methods. If the hacker or attacker obtains a single exponent d which is a secret exponent using the available public key denoted by (n,e). When little quantities of time are factored in, however, it takes a lengthy time, proving that RSA is a strong algorithm.

## 6. FUTURE WORKS

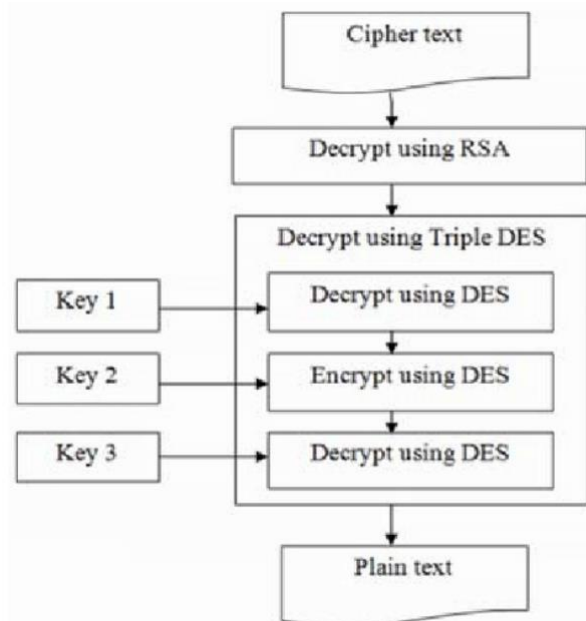


Fig -13: Hybrid algorithms using DES, AES, RSA

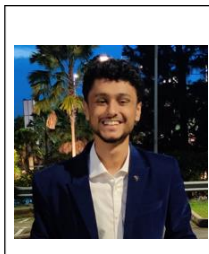
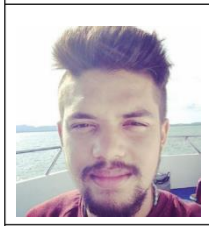

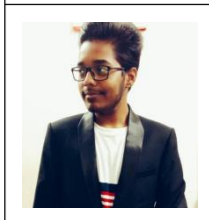
We could in the future also use a hybrid of these algorithms to encrypt and decrypt the data. For instance here (as shown above) we could use a hybrid of RSA and triple DES for decryption of cipher text.

## REFERENCES

- [1] Daniel Commey, Selorm Griffith Klogo and James Dzisi Gadze. Performance comparison of 3DES, AES, Blowfish and RSA for Dataset Classification and Encryption in Cloud Data Storage. International Journal of Computer Applications 177(40):17-22, February 2020.
- [2] Singh, Gurpreet & Kinger, Supriya. (2013). A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security. International Journal of Computer Applications. 67.33-38. 10.5120/11507-7224.
- [3] Chittibabu, Priya. (2019). A Comparative Analysis Of DES, AES and RSA Crypt Algorithms For Network Security in Cloud Computing. 6. 574-82. 10.1729/Journal.19997.
- [4] Harba, E. S. I. (2017) Secure Data Encryption Through a Combination of AES, RSA and HMAC. Engineering, Technology & Applied Science Research Vol. 7, No. 4, 1781-1785
- [5] Naji, A. & Zaidan, A. & Bahaa, Bilal & Hameed, Shihab & Khalifa, Othman. (2003). Novel Approach for Secure Cover File of Hidden Data in the Unused Area within EXE File Using Computation between Cryptography and Steganography.
- [6] Singh, Gurpreet & Kinger, Supriya. (2013). A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security. International Journal of Computer Applications. 67.33-38. 10.5120/11507-7224.
- [7] Jaryal, S. (2017). Comparative Analysis of Various Image Encryption Techniques.

- [8] Zeghid, Medien & Machhout, Mohsen & Khriji, Lazhar & Baganne, Adel & Tourki, Rached. (2007). A Modified AES Based Algorithm for Image Encryption. *World Academy of Science, Engineering and Technology*. 1. 745-750.
- [9] Lin, Ching-yun (2006) *Multimedia Security System*. EE E6886: Topics in Signal Processing.
- [10] Rednic, Emanuil & Toma, Andrei (2009) *Security Management in a Multimedia System Vol 4. No. 2 JAQM*.
- [11] A. Hamza and B. Kumar, "A Review Paper on DES, AES, RSA Encryption Standards," 2020 9th International Conference System Modeling and Advancement in Research Trends (SMART), 2020, pp. 333-338, doi: 10.1109/SMART50582.2020.9336800.
- [12] Mahajan, Prerna & Sachdeva, Abhishek (2013) A Study of Encryption Algorithms AES, DES and RSA for Security. *Global Journal of Computer Science and Technology Network, Web & Security*, Volume 13 Issue 15.
- [13] A. Pande, P. Mohapatra and J. Zambreno, "Securing Multimedia Content Using Joint Compression and Encryption," in *IEEE MultiMedia*, vol. 20, no. 4, pp. 50-61, Oct.-Dec. 2013, doi: 10.1109/MMUL.2012.29.
- [14] Guleria, S., & Vatta, S. (2013). TO ENHANCE MULTIMEDIA SECURITY IN CLOUD COMPUTING ENVIRONMENT USING CROSSBREED ALGORITHM.
- [15] Sivakumar R, Balakumar B and Pandeewaran V (2018) A Study of Encryption Algorithms (DES, 3DES and AES) for Information Security. *International Research Journal of Engineering and Technology (IRJET) Volume: 05 Issue: 04*
- [16] Chadha, Aman & Mallik, Sushmit & Chadha, Ankit & Johar, Ravdeep & Roja, M.. (2015). Dual-Layer Video Encryption using RSA Algorithm. *International Journal of Computer Applications*. 116. 33-40. 10.5120/20302-2341.
- [17] S. Srilaya and S. Velampalli, "Performance Evaluation for DES and AES Algorithms- A Comprehensive Overview," 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2018, pp. 1264-1270, doi: 10.1109/RTEICT42901.2018.9012536.
- [18] Lenstra A.K. (2001) Unbelievable Security Matching AES Security Using Public Key Systems. In: Boyd C. (eds) *Advances in Cryptology — ASIACRYPT 2001*. ASIACRYPT 2001. *Lecture Notes in Computer Science*, vol 2248. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/3-540-45682-1\\_5](https://doi.org/10.1007/3-540-45682-1_5)
- [19] Kaur, Surinder & Bharadwaj, Pooja & Mankotia, Shivani. (2017). Study of Multi-Level Cryptography Algorithms: Multi-Prime RSA and DES. *International Journal of Computer Network and Information Security*. 9. 22-29. 10.5815/ijcnis.2017.09.03.
- [20] Barenghi, Alessandro & Bertoni, G.M. & Breveglieri, Luca & Pelliccioli, Mauro & Pelosi, Gerardo. (2010). Low voltage fault attacks to AES. 7-12. 10.1109/HST.2010.5513121.

**BIOGRAPHIES**

	<p><b>Shaolin Kataria</b> is a Bachelor of Technology undergraduate in Information Technology (B.Tech IT) from Vellore Institute of Technology, Vellore, India, to be completed in April 2022. He has interned at NUS Singapore and HPE in the past. He is currently working in the domain of IoT and robotics as an intern in different start-ups like Artenal and Wimera. His research interest includes IoT, robotics, applications of machine learning, cyber security and applications of Blockchain. He has been felicitated with the GD Naidu Young Scientist Award by his university along with 3 special achiever awards and 1 achiever award. He has experience in technical blogging, counselling with societies/ clubs in his university along with an ed tech start up Corporate Gurukul.</p>
	<p><b>Elio Jordan Lopes</b> is a Bachelor of Technology undergraduate in Computer Science from Vellore Institute of Technology. He is a Blockchain developer at Footium, graduate of India's Celo Fellowship by Celo Foundation in 2021. He is currently working as a product developer at the Celo Camp accelerator. His research interests include Blockchain, DAOs web and mobile application development, Decentralised Finance, machine learning and augmented reality and game development. Awards received: Winner Chainlink Virtual Spring hackathon, ETHGlobal Hackmoney and ETHInida ETHOdyssey. Actively participating in Blockchain education and developer advocacy.</p>
	<p><b>Bevis Mathew Niravil</b> is an undergraduate Computer Science student at Vellore Institute of Technology. He is an aspiring data science student. His deep interests lie in Math and Object oriented programming. He is aiming to pursue his graduate studies in the field of data science and research.</p>
	<p><b>Shashank Keshav</b> is a Bachelor of Technology undergraduate in Computer Science from Vellore Institute of Technology. He is an incoming System Development Engineering intern at Amazon 2022. He is a competitive programming enthusiast . His research interests include Blockchain, Web Development, Decentralised Finance and machine learning.</p>