## Secure Information Management Platform

Submitted in partial fulfilment of the requirements for the award of degree of

### BACHELOR OF COMPUTER APPLICATION



**Submitted to:**

### Mr Amit Kumar

**Submitted by:**

**Aryansh Sharma**

**18021020182**

**SCHOOL OF COMPUTING SCIENCE & ENGINEERING**

### Galgotias University, Greater Noida

**May 2021**

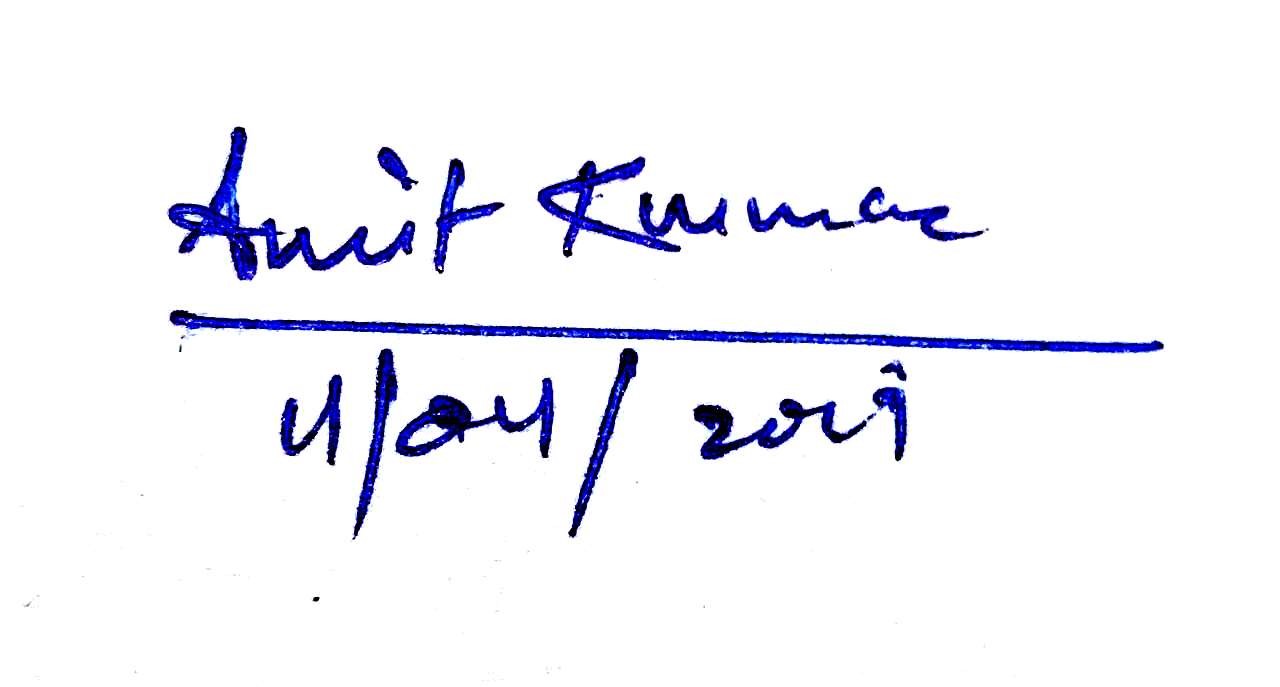


SCHOOL OF COMPUTING AND SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Certified that this project report “SECURE INFORMATION MANAGEMENT PLATFORM” is the bonafide work of “ARYANSH SHARMA”

(18021040182) who carried out the project work under my supervision.



SIGNATURE OF SUPERVISOR

SIGNATURE OF DEAN

DECLARATION

I the undersigned solemnly declare that the project report **Secure Information Management Platform** is based on my own work carried out during the course of our study under the supervision of Mr Amit Kumar.

I assert the statements made and conclusions drawn are an outcome of my research work. I further certify that

1. The work contained in the report is original and has been done by me under the general supervision of my supervisor.

1. The work has not been submitted to any other Institution for any other degree/diploma/certificate in this university or any other University of India o abroad.

1. We have followed the guidelines provided by the university in writing the report.

1. Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and giving their details in the references.

**Aryansh Sharma**

**18021040182**

## ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank them

I owe my deep gratitude to our project **Guide Mr Amit Kumar**, who took keen interest on our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.

I respect and thank Our **Program Chair Mr Shiv Kumar Verma**, for providing me an opportunity to do the project work in and giving us all support and guidance which made me complete the project duly. I am extremely thankful to him for providing such a nice support and guidance, although he had busy schedule managing the University affairs.

I am thankful to and fortunate enough to get constant encouragement, support from my **Team Member Naved Anjum and Anubhav Mishra.** We worked as a team in successfully completing our project work.

**Aryansh Sharma**

# ABSTRACT

Automate Business workflow using Secure Information Management Platform it integrates the security and efficiency on one single platform. The huge expansion in the data generation makes the flow of business and resources slow for both corporates and clients. This Paper Proposes a reliable, secure and efficient solution to manage data and organize it on your existing Operating System. It serves as a Utility Software to Optimize File Directories and Sort Different File Types based on their Extension to Separate Folders to facilitate Ease of Access and work flow Efficiency.

The Internet makes it possible to exchange information between people, corporate and Business Enterprises and makes information security more and more important. Some of data might be transmitted through insecure channel from sender to receiver. Different techniques and methods have been using by private and public sectors to protect sensitive data from intruders because of the security of electronic data is crucial issue. However, there is often Data Breaches and Sniffing of Data Packets between Active Hosts and Server Nodes. Therefore, based on the characteristics and application requirements of the Security Standard Provided by The National Institute of Standards and Technology(NIST), this paper optimizes the Advanced Encryption Standard (AES) and builds the platform that provide File Encryption and reliability to the End User.

These two utility Software provide an efficient way to manage and transfer files from one user to another, it makes the business process workflow fast and secure.

# Table of Contents

**Declaration............................................................................................................ i**

**Acknowledgements ................................................................................................ ii**

**Abstract ................................................................................................................ iii**

**Table of Contents .................................................................................................. iv**

**List of Figures ...................................................................................................... v**

**List of Tables ...................................................................................................... vi**

**List of Acronyms................................................................................................... vii**

**List of Graphs….................................................................................................... viii**

**Chapters:**

1. **Introduction ................................................................................................ 1**
2. **Algorithms and Libraries .......................................................................... 3**
3. **Feasibility Study.......................................................................................... 4**
4. **Frontend Design GUI Framework ............................................................ 5**

1. **Algorithm Workflow and Pseudocode ..................................................... 19**

1. **Performance Evaluation of AES................................................................ 31**

1. **Conclusion & Future Scope........................................................................ 33**

**References**

**Annexure- I: Attendance Report**

**Annexure- II: Progress Report by Guide**

**Annexure-III: Plagiarism Report through Turntin**

**Annexure-IV: Research Paper (Published/Communicated)**

# List of Figures

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Title** | **Page No.** |
| 1 | Random Folder with different file types | 5 |
| 2 | SIMP – File Management GUI | 6 |
| 3 | Browse path selection | 7 |
| 4 | Success message console log | 8 |
| 5 | Input of random folder with multiple different file types | 9 |
| 6 | Output of random folder after File Management | 9 |
| 7 | SIMP- File Encryption GUI | 10 |
| 8 | Input File for Encryption | 11 |
| 9 | File Encryption Process Successful | 12 |
| 10 | Output Encrypted File | 13 |
| 11 | Input Encrypted File for decryption | 14 |
| 12 | File Decryption Process Successful | 15 |
| 13 | Output decrypted file in original format | 16 |
| 14 | Decryption using Bad Secret Key Input | 17 |
| 15 | File format not supported Error | 18 |
| 16 | Workflow of SIMP- File Management | 19 |
| 17 | Encryption Flowchart | 23 |
| 18 | Decryption Flowchart | 28 |

# List of Tables

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Title** | **Page No.** |
| 1 | AES Parameters | 24 |
| 2 | AES Key Size with generation time | 31 |
| 3 | AES File Size Encryption and Decryption time | 31 |

# List of Acronyms

|  |  |
| --- | --- |
| **Acronyms** | **Full Form** |
| SIMP | Secure Information Management Platform |
| NIST | National Institute of Standards and Technology |
| AES | Advanced Encryption Standard |
| GUI | Graphical User Interface |
| DLL | Dynamic Link Library |
| Tk | Tkinter Library |
| DES | Data Encryption Standard |
| ARK | Add Round Key |

# List of Graphs

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Title** | **Page No.** |
| 1 | AES File Size Encryption Time | 32 |
| 2 | AES File Size Decryption Time | 32 |

### 1. INTRODUCTION

Secure Information Management Platform perform two major tasks first one is it sort different files based on their extension to distinct folders in accordance to use user can easily access these files. Secondary it performs AES- Advance Encryption Standard to Encrypt files and uses to symmetric key to perform encryption and decryption on the files.

It provides GUI interface that gives an amazing User Experience integrate it with the existing Operating System and organize your system to increase efficiency and Advanced Encryption Standard provide a surety of security which no one can break. Business Firms use this to securely transfer their confidential data from one system to another.

It provides an interactive Interface for the best user experience user can easily navigate to different files, browse them and modify them easily and encryption provides a way to securely store and transfer of those files from one user to another.

No Existing Operating System provide file sorting based on their extension for a clean and clear organised system, as compared to windows and Linux it works as a utility software to increase efficiency and the encryption module gives the user a sense of safe and secure storage and sharing of information from one user to another, Internet communication is playing the important role to transfer large amount of data in various fields. Some of data might be transmitted through insecure channel from sender to receiver. Different techniques and methods have been using by private and public sectors to protect sensitive data from intruders because of the security of electronic data is crucial issue. Cryptography is one of the most significant and popular techniques to secure the data from attackers by using two vital processes that is Encryption and Decryption. Encryption is the process of encoding data to prevent it from intruders to read the original data easily.

Our Encrypt module based on AES provide the confidentiality, integrity, nonrepudiation and authentication, especially in a corporate environment where confidential data are transferred frequently.

### 2. ALGORITHMS AND LIBRARIES

#### Python – Algorithm

The algorithms used in this software are written in Python for sorting of different file types and the Encryption Standard is provided by National Institute of Standards and Technology.

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

#### Tkinter Framework for Graphical User Interface

The software uses a rich Graphical User Interface to provide best User Experience to the end users, Tkinter Framework is used to achieve this smooth and fast interactive platform that defines frontend designing components and integrate it with the backend Algorithms.

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

### 3. FEASIBILITY STUDY

#### 1. Technical Capability

Python is easy to use for Algorithm and for fast scripting purpose and National

Institute of Standards and Technology provides a fully tried and trusted Encryption Standard for secure transmission of information.

#### 2. Budget

Python is an open source language so there is no cost as such included in the making of this software and most of the development IDE and tools are Open Source.

#### 3. Legality

User Agreement of file access is required for the process of this software however as far as companies are concerned they can formulate a Term and Condition before installation of the software.

#### 4. Risk

There is no major risk involved with this project, the encryption algorithms would be highly sophisticated and efficient file management will be achieved.

#### 5. Operational Feasibility

Business Process Automation is the major problem corporates are facing now a day as the growth in the Data is exponential and managing it is a complicated task, this software would outcome those challenges.

#### 6. Available Alternative Systems

The system that are available for sorting is paid and the user needs to use different programs for sorting and encryption, however we made the two systems available in a single program and its open source to use.

### 4. FRONTEND DESIGN GUI FRAMEWORK

Most of the time, tkinter is all we really need, but a number of additional modules are available as well. The Tk interface is located in a binary module named \_tkinter. This module contains the low-level interface to Tk, and should never be used directly by application programmers. It is usually a shared library (or DLL), but might in some cases be statically linked with the Python interpreter.

#### The SIMP – File Management GUI is shown below with sample input and output

The Sorting process is as follow:

1. Select the path of the folder containing multiple files of different types
2. The Sweep button arranges the file based on their extension to separated folders and the output is shown in the terminal interface of the program. This Can be further analyse by the GUI Shown below which illustrate the Steps of GUI function:

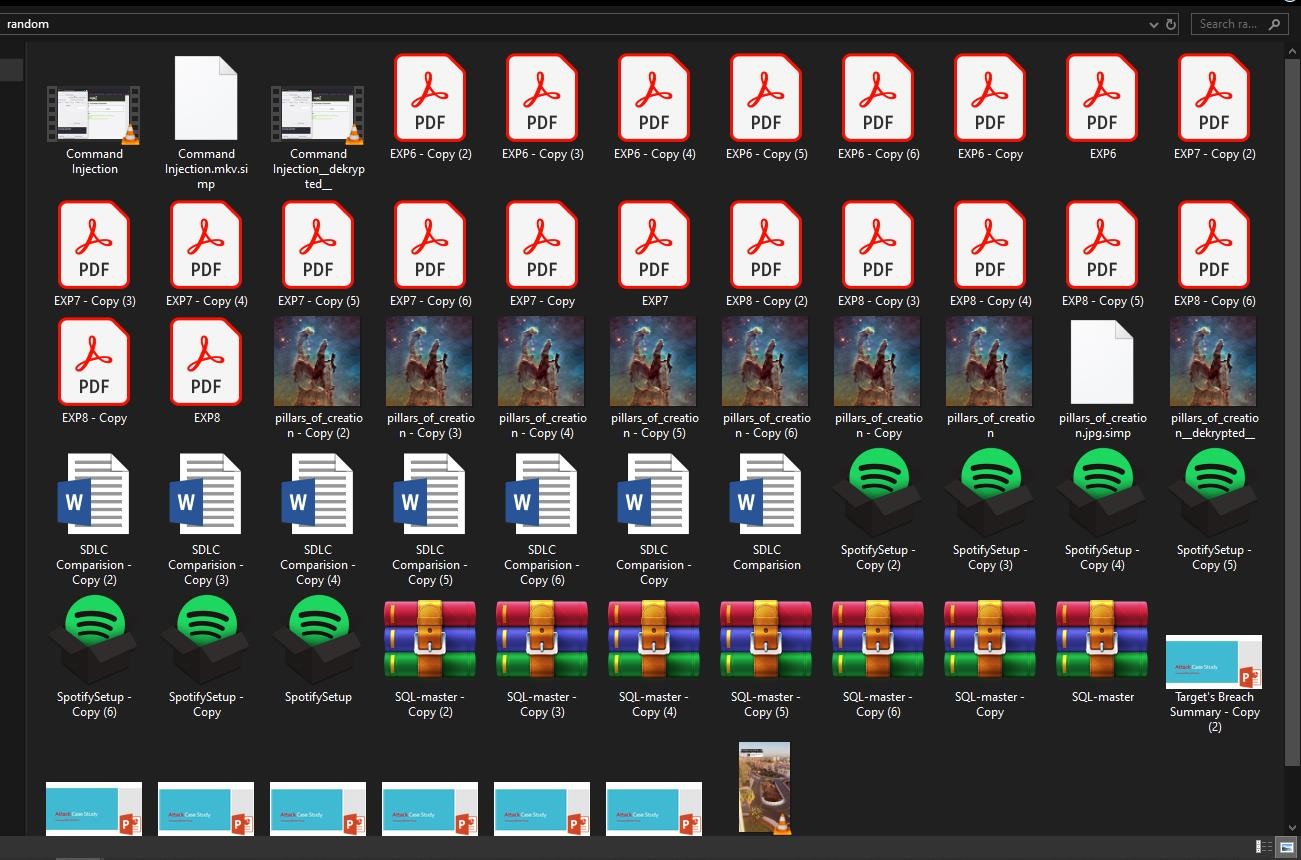


Figure 1. Random Folder with different file types

1. A folder named random is taken to demonstrate which contains multiple files of different types.
2. SIMP- File Management started, the initial Program has three buttons
   1. Browse: To Open and select the folder location.
   2. Sweep: To Run the Program and arrange the files into different folders based on their file types.
   3. Encrypt Launcher: To Open the SIMP – File Encryption Program.

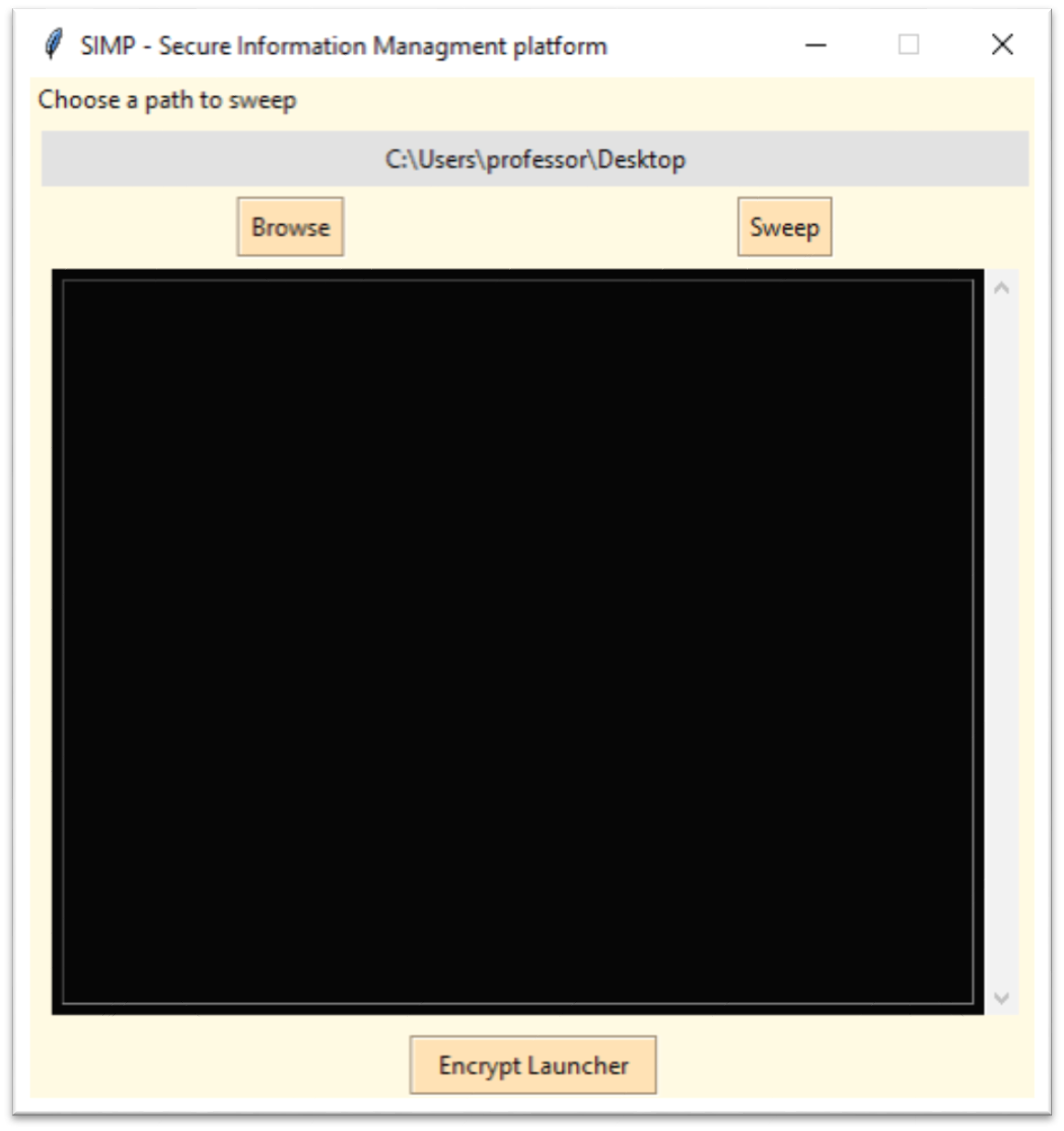


Figure 2. SIMP – File Management GUI

1. Click the Browse Button to Select the Folder Path.

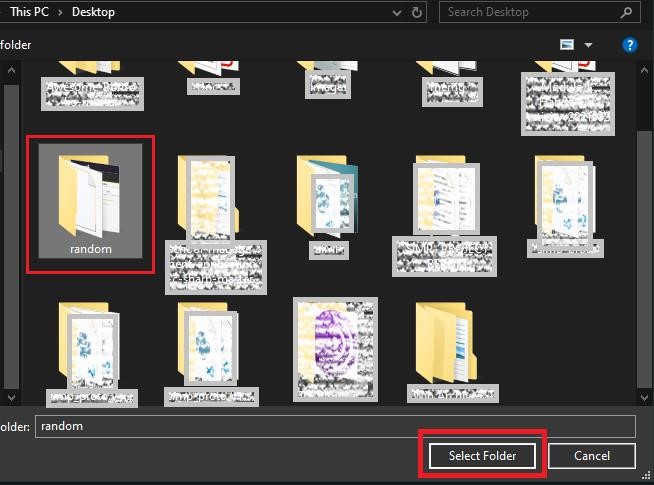


Figure 3. Browse path selection

1. Now the Folder is Folder is selected and we can run the sweep function to sort the program and arrange the files based on their file types in to distinct folders.

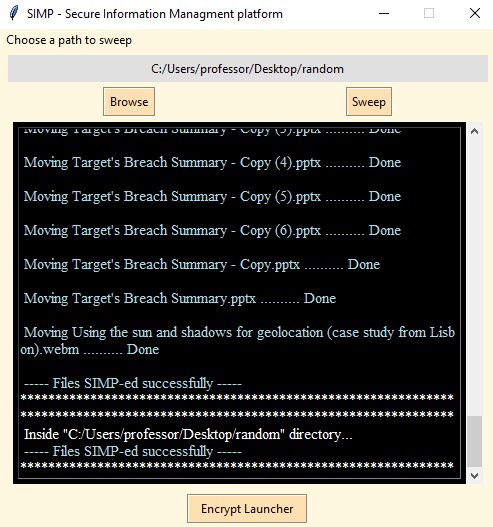


Figure 4. The Success message is printed in the console log

1. All the files are arranged and the final output is shown below with comparison to the previous one.

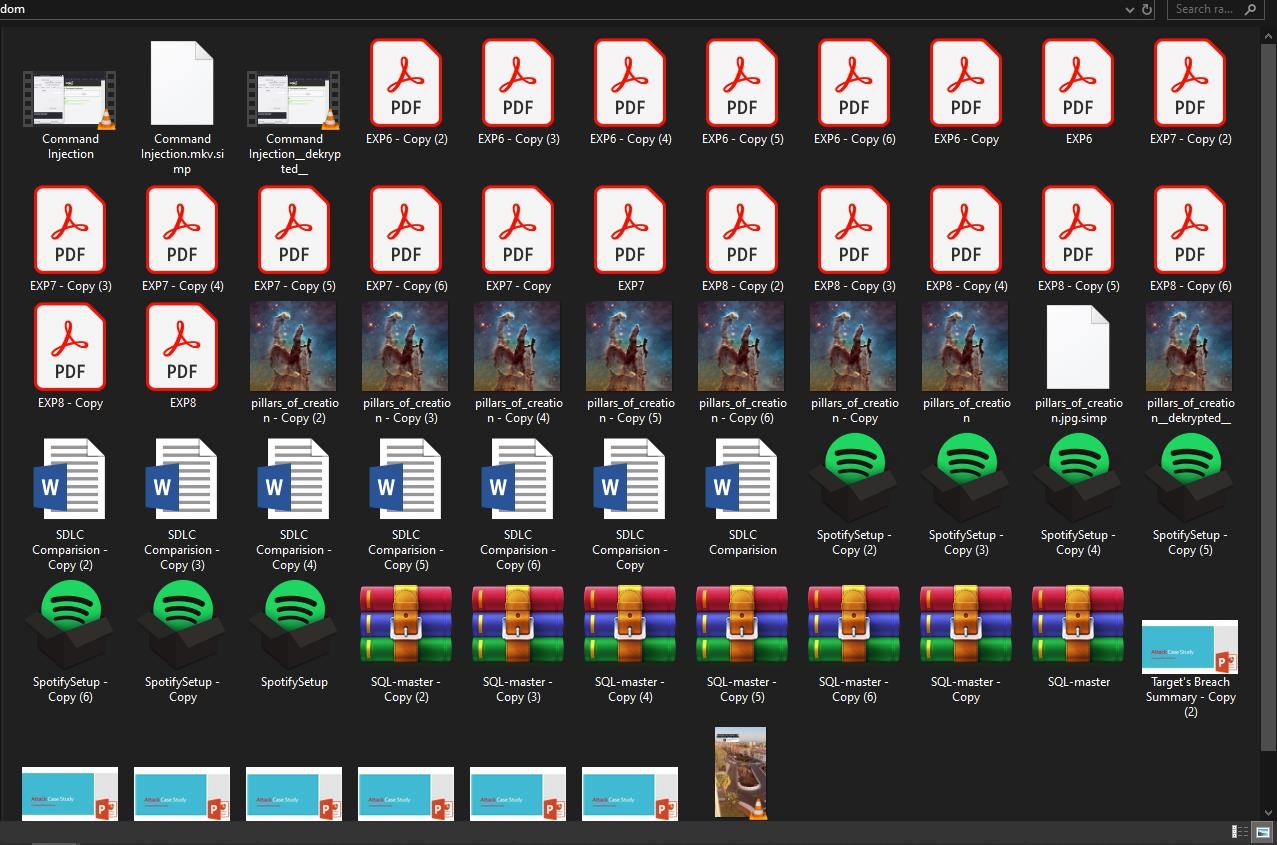


Figure 5. **Input** of random folder with multiple different file types

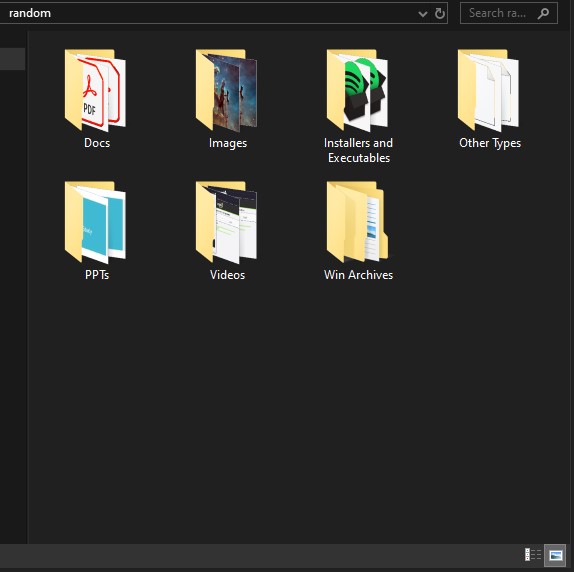


Figure 6. **Output** of random folder after File Management

**The SIMP – File Encryption GUI is shown below with sample input and output.**

**Encryption Process**

1. SIMP – File Encryption Program initial start GUI.

2.

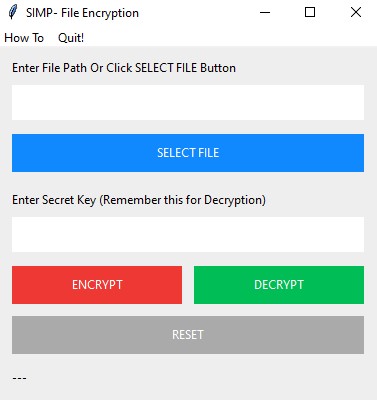


Figure 7. SIMP- File Encryption GUI

The Initial GUI Contain these elements:

* 1. **Select File:** To Open and select a file to Encrypt.
  2. **Secret Key Input:** The Key value to encrypt and use for decryption in the later process. The same symmetric key would be use for encryption and decryption.
  3. **Encrypt:** To Start the encryption process from normal file selected in the path
  4. **Decrypt:** To Decrypt the Encrypted file containing (. simp) extension.
  5. Status Bar: To Reflect the Progress of Encryption, Decryption or Error in path selection.

1. For Encryption Process we need to first select a path of the file and provide it as input.

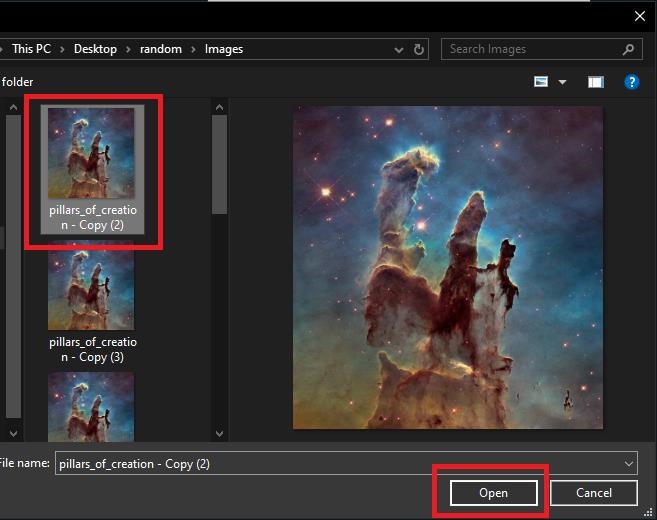


Figure 8. **Input** File for Encryption (pillars\_of\_creation.jpeg)

1. The Selected file path is updated and the Secret key input is provided as

“Apple” in this example. Click the Encrypt Button and the File Encryption Process is successfully done and the Status is Printed in the Status Bar below.

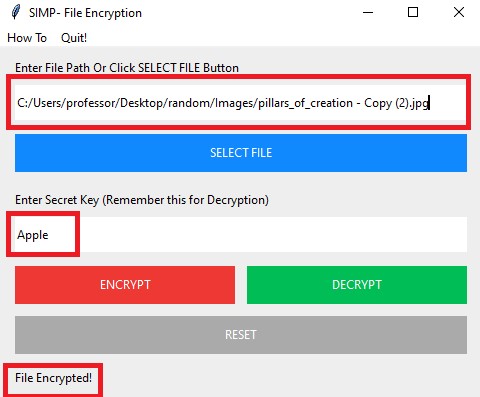
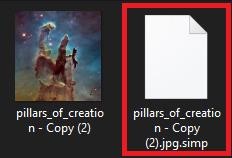


Figure 9. **File Encryption Process Successful** (pillars\_of\_creation.jpeg)

1. The Output can be located in the same path of original file but in encrypted format and unreadable by any app, the encrypted file can be identified by the .simp extension in the encrypted file name.



##### Figure 10. Output Encrypted File

**Decryption Process**

1. The Decryption Process starts with the encrypted file. First the encrypted file path is selected in the same SIMP- File Encryption GUI.

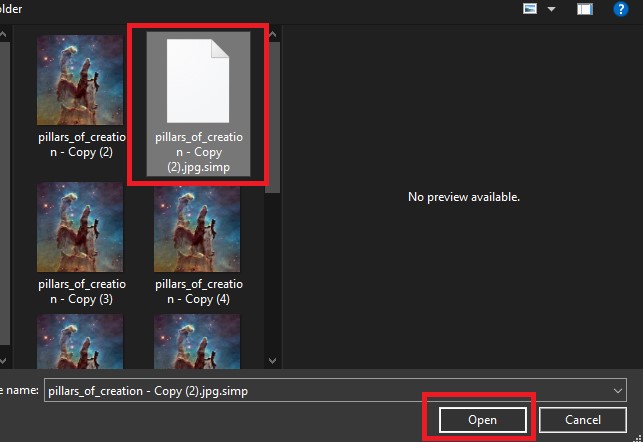


Figure 11. **Input Encrypted File for decryption**

1. The Selected file path is updated and the Secret key input is provided as “Apple” in this example. Click the Decrypt Button and the File Decryption Process is successfully done and the Status is Printed in the Status Bar below.

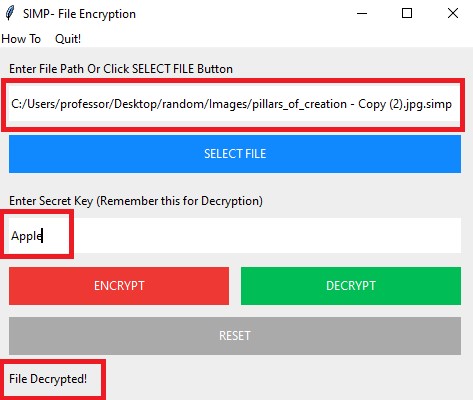


Figure 12. **File Decryption Process Successful**

1. The Output can be located in the same path of encrypted file but in decrypted format and readable by apps and Operating System as a Picture, the decrypted file can be identified by the \_dekrypted\_ in the decrypted file name.

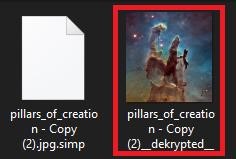
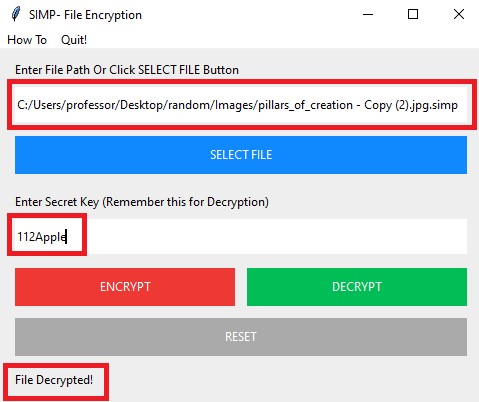


Figure 13. **Output decrypted file in original format** **Decryption Process with Wrong Secret Key Input**

1. If the any non-authorize user try to decrypt the file with wrong input as the original secret key is “Apple” but the non-authorize user provided “112Apple” Which is a wrong decryption key and hit Decrypt the status bar display the File Decrypted Message but the file is not decrypted and a corrupted output is generated.



##### Figure 14. Decryption using Bad Secret Key Input

2. Corrupted Output of Bad Secret key the file generated is same as the correct secret key input but its not readable by apps as the result shows below in figure 12 the Windows Photo Viewer is not recognizing the file as a valid image.

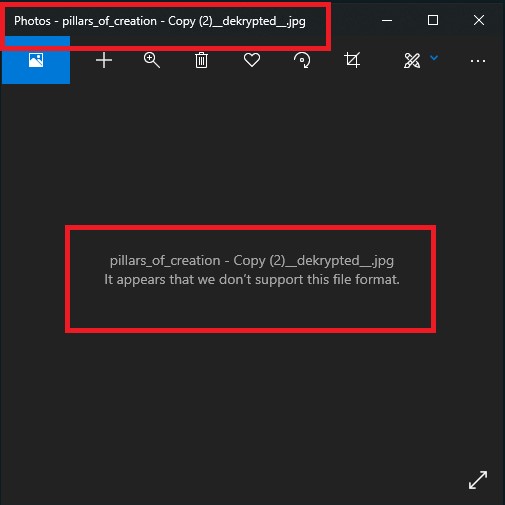
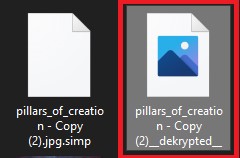


Figure 15. **File format not supported Error**

### 5. ALGORITHMS WORKFLOW AND PSEUDOCODE

#### 1. Sorting Files based on their Extension

The File Sorting Algorithm is based on the bunch of if else conditional statement for each file in the folder it starts over a loop under the loop there are many if else conditional statements based on different file type and for each true statement block the algorithm creates a new directory with the name associated to that file type and move the original file to that folder. It makes the files organized and easy to access in the scenario of a bigger

picture with a lot of files it’s a very useful tool to provide efficiency to the end user, this feature is not providing by any Operating System out there such as

Microsoft’s Windows or Linux. It is a utility Software works on both the OS platforms.

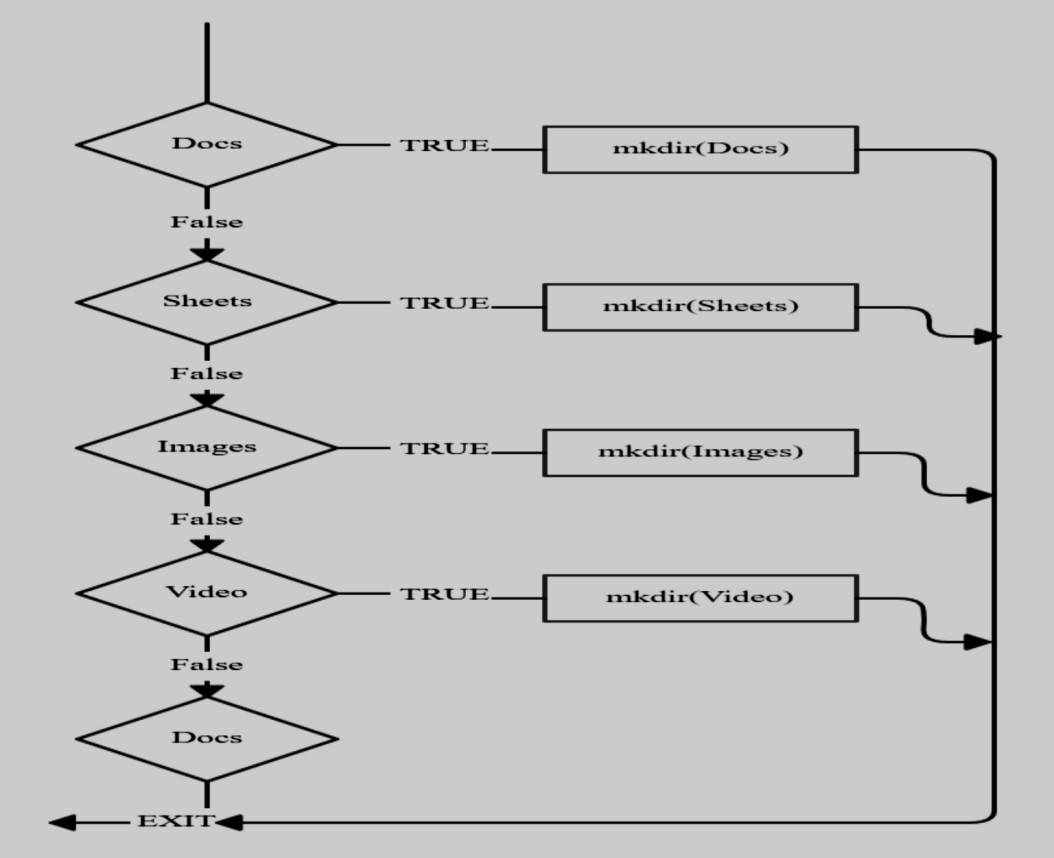


Figure 16. Workflow of SIMP- File Management

#### Pseudocode of SIMP- File Management

Initialize PATH variable depending on the platform

**Begin** if ‘win’ is sys.platform

os = win elif ‘lin’ or ‘mac’ in sys.platform os = ‘unix’ **Get Folder Path**

**Try :**

For: all files in current path

If : ext = = Docs

Mdkir (‘docs’)

Move(file, Docs)

If : ext = = Sheets

Mdkir (‘Sheets’)

Move(file, Sheets) If : ext = = PPTs

Mdkir (‘PPTs’)

Move(file, PPTs)

If : ext = = Text Docs

Mdkir (‘Text Docs’)

Move(file, Text Docs)

If : ext = = Images

Mdkir (‘Images’)

Move(file, Images)

If : ext = = Music

Mdkir (‘Music’)

Move(file, Music)

If : ext = = Video

Mdkir (‘Video’)

Move(file, Video)

If : ext = = Web

Mdkir (‘Web’)

Move(file, Web)

If : ext = = Coding

Mdkir (‘Coding’)

Move(file, Coding)

If : ext = = Python Script

Mdkir (‘Python Script’)

Move(file, Python Script)

**Except: Exception**

Error

**End**

2. AES- Advanced Encryption Standard for File Encryption

Internet communication is playing the important role to transfer large amount of data in various fields. Some of data might be transmitted through insecure channel from sender to receiver. Different techniques and methods have been using by private and public sectors to protect sensitive data from intruders because of the security of electronic data is crucial issue. Cryptography is one of the most significant and popular techniques to secure the data from attackers by using two vital processes that is Encryption and Decryption. Encryption is the process of encoding data to prevent it from intruders to read the original data easily.

This stage has the ability to convert the original data (Plaintext) into unreadable format known as Cipher text. The next process that has to carry out by the authorized person is Decryption. Decryption is contrary of encryption. It is the process to convert cipher text into plain text without missing any words in the original text. To perform these process cryptography relies on mathematical calculations along with some substitutions and permutations with or without a key.

Modern cryptography provide the confidentiality, integrity, nonrepudiation and authentication. These days, there are a number of algorithms have been available to encrypt and decrypt sensitive data which are typically divided into three types. Frist one is symmetric cryptography that is the same key is used for encryption and decryption data. Second one is Asymmetric cryptographic. This types of cryptography relies on two different keys for encryption and decryption. Finally, cryptographic hash function using no key instead key it is mixed the data. The symmetric key is much more effective and faster than Asymmetric. Some of the common symmetric algorithms is Advance Encryption Standard (AES), Blowfish, Simplified Data Encryption Standard (S-DES) and 3DES. The main purpose of this paper will provide a detail information about Advanced Encryption Standard (AES) algorithm for encryption and decryption data then make a comparison between AES and DES algorithm to show some idea why replacing DES to AES algorithm.

Advanced Encryption Standard (AES) algorithm is one on the most common and widely symmetric block cipher algorithm used in worldwide. This algorithm has an own particular structure to encrypt and decrypt sensitive data and is applied in hardware and software all over the world. It is extremely difficult to hackers to get the real data when encrypting by AES algorithm. Till date is not any evidence to crake this algorithm. AES has the ability to deal with three different key sizes such as AES 128, 192 and 256 bit and each of this ciphers has 128 bit block size.

**Encryption Process:**

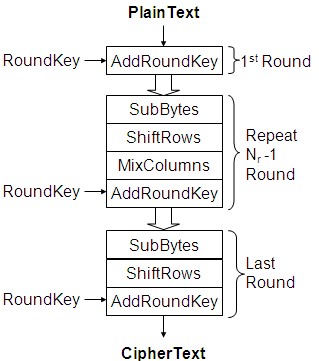


Figure 17. Encryption Flowchart

Secure Information Management Platform uses the same Encryption process shown in Figure 17 to Encrypt the Files and save it in the encrypted format with a .simp extension a key is set by the user at the encryption process after the file is selected.

Each round (except the last one) is a uniform and parallel composition of 4 steps

* SubBytes (byte-by-byte substitution using an S-box)
* ShiftRows (a permutation, which cyclically shifts the last three rows in the State)
* MixColumns (substitution that uses Galois Fields, corps de Galois, GF(28) arithmetic)
* AddRound key (bit-by- bit XOR with an expanded key)

**AES Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Key Length**  **(nk words)** | **Block Size**  **(nb words)** | **Number of**  **Rounds**  **(nr)** |
| **AES- 128** | 4 | 4 | 10 |
| **AES-192** | 6 | 4 | 12 |
| **AES-256** | 8 | 4 | 14 |

1 word = 32 bit

Table 1. AES Parameters

**AES Keys**

With 128 bit: 2128 = 3.4x 1038 possible keys – A PC that tries 255 keys per second needs **149.000 billion years to break AES.**

#### Pseudocode of SIMP- File Encryption Process

Cipher(byte in [4\*nb], word w [Nb \* (Nr + 1) ] ) **begin** byte state [4, Nb] state = in

AddRoundKey ( state , w [0, Nb- 1 ]) for round = 1 step 1 to nr-1

SubByte(state)

ShiftRows(state)

MixColumns(state)

AddRoundKey (state, w[round \*Nb, (round + 1 ) \*Nb-1]) **end for**

SubByte(state)

ShiftRows(state)

AddRoundKey (state, w [Nr \* Nb, (Nr +1 )\* Nb -1]) out = state **end**

#### Pseudocode – round key generation key expansion

KeyExpansion(byte key [4\* Nk], word w [Nb\* (Nr + 1 )], Nk )

**Begin** word temp

i = 0 while ( i < Nk) w[i] = word (key [4 \* i], key [4 \* i + 1 ], key [4 \* i + 2], key [4 \* i + 3 ]) i = i + 1 end while i = Nk while( i < Nb \* (Nr +1 ) temp = w [i -1 ] if ( i mod Nk = 0) temp = SubWord(RotWord(temp)) xor Rcon [i /Nk] else if ( Nk > 6 and i mod Nk = 4 ) temp = SubWord (temp)

end if w[i] = w [i – Nk ] xor temp i = i + 1 end while **end**

#### Cipher Sequence

The Encryption sequence is as follows:

1. Add Round Key (ARK)
2. Byte Sub, Shift Row, Mix Column, Add Round Key 1 Round
3. More rounds like #2
4. Byte Sub, Shift Row, Add Round Key (Skipped Mix Column ) Since Mix Column is the only step that has interaction among bytes, elimination of this step is the motivation behind excluding this sequence as a round.

This Sequence is not symmetrical, which allows the steps of each round to be done in different order (i.e. Byte Sub step could be done after Shift Row step). The

Reason behind having step #1 before the round is that the step come after it can be deciphered without knowing the key. So an initial key must be added to increase confusion to the cipher so that an outside user cannot recognize the pattern of the four steps in each round and in turn decipher the message.

**Decryption Process:**

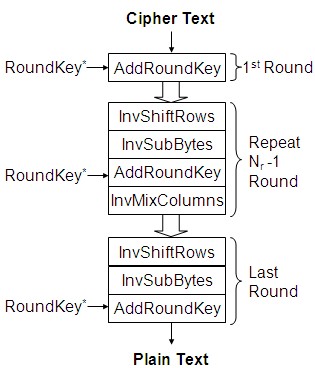


Figure 18. Decryption Flowchart

The decryption process uses the same key that was used for encryption of the file and the same SIMP Encryption tool is used to Decrypt the File. The User select the Encrypted file and provide the key which was set earlier to encrypt the file, and the file will be decrypted using the decryption process shown above.

#### Pseudocode – Decryption – reverse operation

InvCipher (byte in [4\*Nb], byte out [4 \* Nb], word w [Nb\* (Nr+ 1 )])

**Begin** byte state[4, Nb] state = in

AddRoundKey (state , w [Nr\* Nb, (Nr + 1 )\* Nb -1 ]) for round = Nr -1 step -1 down to 1

InvShiftRows(state)

InvSubBytes(state)

AddRoundKey (state, w [round \*Nb , (round + 1 ) \* Nb -1 ]) InvMixColumns(state) end for

InvShiftRows(state)

InvSubBytes(state)

AddRoundKey(state, w [0, Nb-1]) out = state **end**

To decrypt an AES-encrypted ciphertext, it is necessary to undo each stage of the encryption operation in the reverse order in which they were applied. The three stage of decryption are as follows:

 Inverse Final Round AddRoundKey

ShiftRows

SubBytes

 Inverse Main Round AddRoundKey

MixColumns

ShiftRows

SubBytes

 Inverse Initial Round

AddRoundKey

Of the four operations in AES encryption, only the AddRoundKey operation is its own inverse (since it is an exclusive-or). To undo AddRoundKey, it is only necessary to expand the entire AES key schedule (identically to encryption) and then use the appropriate key in the exclusive-or.

### 6. Performance Evaluation of AES

We have tested the performance of AES algorithms by using parameters such as encryption time, decryption time and key generation time.

Encryption time is the time required by any encryption function to convert plaintext into cipher text. Decryption time is the time required to convert again cipher text into plain text. Similarly, key generation time is the time taken by key generation function to generate keys. All these functions generate different times according to the size of text files and key length in any algorithm. Table 2 shows the AES key size with generation Time.

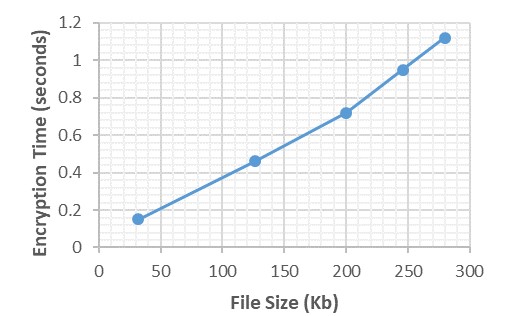
|  |  |  |
| --- | --- | --- |
| **Algorithm** | **Key Size (bits)** | **Generation Time**  **(milliseconds)** |
| AES | 128 | 75ms |

Table 2. AES Key size with generation time

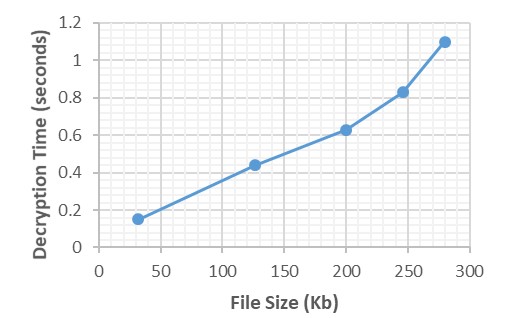
Table 3 shows the encryption and decryption time of AES with different file size. Performance result shows that when we increase the input file size the encryption and decryption time is also increased.

|  |  |  |
| --- | --- | --- |
| **File Size (kilo bytes)** | **Encryption Time**  **(in seconds)** | **Decryption Time**  **(in seconds)** |
| 32 | 0.15 | 0.15 |
| 126 | 0.46 | 0.44 |
| 200 | 0.72 | 0.63 |
| 246 | 0.95 | 0.83 |
| 280 | 1.12 | 1.10 |

Table 3. AES File Size Encryption and Decryption time



Graph 1. AES File Size Encryption time



Graph 2. AES File Size Decryption time

### 7. CONCUSION & FUTURE SCOPE

Optimize File Directories and Sort Different File Types based on their Extension to Separate Folders to facilitate Ease of Access and work flow Efficiency. the Advanced Encryption Standard (AES) and builds the platform that provide File Encryption and reliability to the End User.

One of the primary advantages of AES is its ubiquity. Since it is defined as the standard used by the US government, it is supported by most vendors. Also, it is relatively fast in both hardware and software.

The three possible key lengths supported by AES allow users to pick a trade-off between speed and security. Increased key length increases the execution time of both encryption and decryption. At this time, all three key lengths are considered secure and the best known attacks against AES reduce effective key length by at most three bits.

AES uses a single S-Box for all bytes in all rounds. In contrast, DES uses eight distinct S-Boxes, which increases implementation requirements.

These two utility Software provide an efficient way to manage and transfer files from one user to another, it makes the business process workflow fast and secure.

It provides GUI interface that gives an amazing User Experience integrate it with the existing Operating System and organize your system to increase efficiency and Advanced Encryption Standard provide a surety of security which no one can break. Business Firms use this to securely transfer their confidential data from one system to another.

The future scope is to add a md5 message digest algorithm which is widely used hash function producing a 128- bit hash value for File Integrity and Authentication of the original message or file source.

### REFERENCES

1. Abdullah, A. M., & Aziz, R. H. H. (2016, June). New Approaches to Encrypt and Decrypt Data in Image using Cryptography and Steganography Algorithm., International Journal of Computer Applications, Vol. 143, No.4 (pp. 11-17).
2. Singh, G. (2013). A study of encryption algorithms (RSA, DES, 3DES and AES) for information security. International Journal of Computer Applications, 67(19).
3. Gaj, K., & Chodowiec, P. (2001, April). Fast implementation and fair comparison of the final candidates for Advanced Encryption Standard using Field Programmable

Gate Arrays. In Cryptographers’ Track at the RSA Conference (pp. 84-99). Springer Berlin Heidelberg.

1. Stallings, W. (2006). Cryptography and network security: principles and practices. Pearson Education India.
2. Yenuguvanilanka, J., & Elkeelany, O. (2008,April). Performance evaluation of hardware models of Advanced Encryption Standard (AES) algorithm. In Southeastcon, 2008. IEEE (pp. 222-225).
3. Lu, C. C., & Tseng, S. Y. (2002). Integrated design of AES (Advanced Encryption Standard) encrypter and decrypter. In Application-Specific Systems, Architectures and Processors, 2002. Proceedings. The IEEE International Conference on (pp. 277-285).
4. Mohamed, A. A., & Madian, A. H. (2010, December). A Modified Rijndael Algorithm and it's Implementation using FPGA. In Electronics, Circuits, and Systems (ICECS), 2010 17th IEEE International Conference on (pp. 335-338).
5. Pramstaller, N., Gurkaynak, F. K., Haene, S., Kaeslin, H., Felber, N., & Fichtner, W. (2004, September). Towards an AES crypto-chip resistant to differential power analysis. In Solid-State Circuits Conference, 2004. ESSCIRC 2004.
6. Cryptography: A Comparative Analysis for Modern Techniques January 2017International Journal of Advanced Computer Science and Applications 8(6)
7. Youtube,com, medium.com, github.com, stackoverflow.com