Image Encryption using AES and RSA

Jigyas Sharma
EECS
University of Kansas
Lawrence, Kansas
engineer@ku.edu

Aiden Schmelzle
EECS
University of Kansas
Lawrence, Kansas
schmelzle@ku.edu

***Abstract*—Given the state of our world, the importance of data and privacy is not only a necessity but has become imperative. The field of security has made significant strides in creating secure encryption methods. AES and RSA have been two encryption methods that have stood the test of time in symmetric-key cryptography and public-key cryptography. This project explores the use of these algorithms as well as how the foundational ideas can be applied to the concept of encryption of images. Furthermore, the project sheds light on some challenges and implications along the project before moving onto possible directions for future work in the area of image encryption.**

***Index Terms*—image encryption, AES, RSA, security, cryptography**

I. Introduction

The amount of data leaks and attacks on the digital infrastructure have lead to having over 5 billion records vulnerable in just 2023. The need of privacy in our world constantly increases as the demand for private data increases. The novelty of being able to attack over the internet is that the adversary can attack in jurisdictions where they are not located, this enables the adversaries to attack more openly and be more risk taking as the consequences from doing these attacks are near zero. The field of cyber-security has made significant strides in protecting data that is plaintext, however, the other mediums of data have not seen significant protection. Through the course of this project, we will try to understand the working of two of the most popular encryption algorithms in symmetric-key cryptography and public-key cryptography. Symmetric-key cryptography is a field in cryptography where a singular key is established for encryption as well as decryption. The problem with symmetric-key encryption is that it requires a pre-established secret. However, symmetric-key cryptography has been employed in various fields like Financial services, Military and Government due to its strong security guarantee and fast encryption and decryption speed. Essentially, symmetric-key cryptography is employed in fields where performance needs to be optimal and the data size is large. AES is one of the most widely used symmetric-key cryptographic algorithm. AES was standardised as the symmetric-key method by United States National Institute of Standards and Technology in 2001. During the course of this project, we look at functioning and methodology of AES and different modes in AES to employ the algorithm in encryption and decryption of images. ****INSERT RSA Introduction****.

II. Background and Motivation

In the current state of digital infrastructure, the dominance of media in providing entertainment and general communication of ideas has prevailed. The constant use of images and other forms of media to communicate ideas, provide entertainment, and creation communities proves the dominance of media in the present. However, the widespread use of media presents significant challenges in the field of security. This project is motivated by the need of this urgent bridging between protection and media. The project focuses to explore the adaptability and efficiency of AES and RSA in encryption of images. Through this idea, project aims to contribute to the ever evolving field of cyber-security.

A. Symmetric-key Cryptography and AES

Historically, Symmetric-key cryptography was first developed as Caesar ciphers, where alphabets were shifted from their original place to some value left or right to make sure that if adversaries were to get hold of the message, the message would not convey any information without the knowledge of the shift. Throughout the human timeline, there have been multiple methods such as, Vigenere Cipher during the rennaisaince, Enigma during the world war, and the development of telegraph as a secure communication channel. These methods have laid the foundational work for ideology behind symmetric-key cryptography in the digital age. The Data Encryption Standard(DES) which was developed by IBM in 1970s was the first symmetric-key encryption method that was later adopted as the standard by the United States government. However, by the late 1990s, DES was deemed insecure due to its short key sizes which made it prone to brute-force attacks. This led to the development of the Advanced Encryption Standard in the 1990s which essentially replaced the DES in 2001. AES fixed some of the problems that DES had regarding security. Firstly, AES increased the key size from DES to having a variety of 128 bytes, 192 bytes and 256 bytes, on a root level, this makes it exponentially harder to conduct brute-force attacks. Secondly, AES is more resistant to cryptanalysis attacks since the secret keys have a more complex structure. Thirdly, AES improved the efficiency on the hardware, thus improving the performance of the implementation. Lastly, AES solved the major issue with DES being prone to some plaintext attacks which were not possible on AES.

B. Public-key Cryptography and RSA

This is placeholder text that needs to be filled by Aiden

III. AES Encryption and Methodology in Image Encryption

AES Encryption works in fixed size block of 128 bits. AES has multiple modes of operation, however, due to images having different resolutions and the world advancing and creating larger resolutions for clearer view, the project was more aligned with the Chain Block Cipher(CBC) mode of operation. Chain Block Cipher in AES works by XORing the current input block with the previous cipher block. This also adds a layer of security which will become more apparent after understanding the implementation of AES in image encryption.

A. Working of AES

AES employs multiple steps to encrypting and has some key components namely the S-box and the round key, . This section provides an overview of how the Chain Block Cipher operation mode in AES encrypts and decrypts given data. The first step in AES is to establish a secret key, this key ranges from a size of 128, 192 or 256 bytes. After the encryption algorithm receives the secret key, the algorithm utilizes a key expansion method to produce a round key. The round key is generated from the secret key for each round of encryption. Given a plain-text, AES decomposes the plaintext into multiple 128 bit blocks, the last few blocks also go through a padding process which ensure that each block consists of 128 bits. Now, the algorithm has multiple 128 bit blocks to encrypt, each of these blocks go through a multiple steps of encryption.

1) Each byte block and the associated round key for the block are passed through an XOR gate. This is the AddRoundKey step.
2) Each of the byte in the block is replaced with another value from the S-box. This is the SubBytes step and the S-box is a static lookup table that is derived from the multiplicative inverse over the Galois Field.
3) Each of the bytes go through a cyclical left shift. This is the ShiftRows step and the left shift that occurs is dynamic and random.
4) Each of the 4 bytes in column are then combined. This is the MixColumn step and it ensures further diffusion.

The above defined steps are repeated for each 128 bit block. In the last step, the MixColumn step is omitted. This is done to make sure that during decryption efficiency can be guaranteed, however, this does not lead to weakening of security. Furthermore, the decryption process is works in reverse to the encryption method by inverting all the methods with the use of the secret key to produce the round key and other components.

B. AES in Image Encryption

Now, that AES has clear defined steps and how to implement them, the project implements the method to encrypt a given image using AES. Since, the project uses CBC operation mode, it is impossible to encrypt images without converting them to plaintext. The project chose to go with the Chain Block Cipher Method as the importance of the project was availability and real life use, other modes of operations would either limit the resolution of the images that could be encrypted or degrade the image quality when decrypting. This presented the project with a challenge of being able to degrade images into plaintext and then recreating the images without data loss. The project used some of the python image processing libraries. The idea behind being that if we can process images as numpy arrays, we can pass that as the plaintext to be able to encrypt the data. Furthermore, when we recreate the data we need to save some image meta data to recreate the image from the decrypted numpy arrays. The python libraries used for the project are cv2 to process the image and numpy to be able to create an array from the processed image data. The pseudo steps that take place for encryption are defined below:

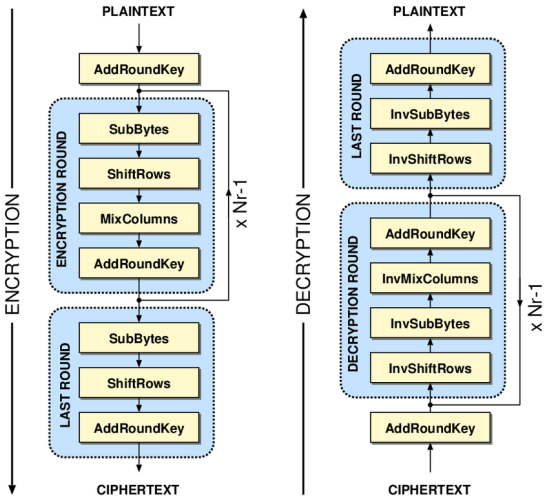

Fig. 1. Steps to CBC Encryption [1]

1) The user uploads the image which is processed by cv2 by using the imread() function.
2) The processed cv2 image is then converted to a numpy array.
3) The numpy array is then passed through tobytes() function.
4) The image metadata such as color channels and original shape is saved using shape function from Numpy.
5) The original image is then discarded from the encryption server.
6) Each of the 128 bit block goes through a function pad() to add null bits to blocks that are not 128 bits.
7) The key and initialization vector is produced to begin the encryption process.
8) The image data that was converted to bytes goes through the AES encryption process.
9) The script produces 4 files: Key, Initialization Vector, Image metadata and Encrypted Data.
10) The Image metadata, Initialization Vector and Encrypted data is stored on a database.

11) The user is provided with a key and an identification number.

During the decryption process, the user provides the identification number which helps retrieve files from the database and the key is provided to be able to go through the decryption process with the associated files. Considering the database has no security and the attackers can retrieve the data from the server, the server only stores the encrypted data which without a key is computationally infeasible to recreate.

C. Benchmarks of the framework

The framework was tested in an isolated manner where the logic of encryption and decryption were tested independently. The testing dataset consisted of 623 images with the average resolution of the images being 1920x1080 and the peak resolution in the dataset being 7680x4320. The encryption framework had an average time to create the required files in 0.2 seconds and the worst performance of encryption at 0.5 seconds. The decryption framework was tested on the files that were created from the encryption process. The decryption process took an average of 0.4 seconds to decrypt data and recreate the image with the worst time recorded being 0.8 seconds.

D. Challenges and Limitations of the framework

The framework in controlled testing environments performs extremely well, however, it has some challenges and limitations. Firstly, the key from the encryption server to user should be transferred over a secure channel. These channels foundationally employ the ideology of public-key cryptography which makes the key being compromised from to a Man-in-the-Middle attack. The framework benchmarks were performed on a variety of different devices, the benchmarks mentioned above are the average of all the devices. The framework is heavily reliant on the hardware of the device that encrypts and decrypts the data as the implementation currently runs locally. The framework loses efficiency and increases time in encryption and decryption benchmarks on CPUs older than 9th generation Intel CPUs. Lastly, the framework starts to lose image quality beyond the 3840x2160 resolution. The data from an 7680x4320 resolution image is only reproduced to 2560x1440. This behaviour could essentially be caused as the secret key is used to create round key, however, when the image data is so large, the round key expansion causes data to lose quality over a certain chain size.

IV. RSA Encryption

RSA is an asymmetric-key encryption algorithm that relies on the mathematical properties of large prime numbers. It is often used for secure key exchange and digital signatures. In image encryption, RSA can be used to encrypt the AES key used for pixel encryption. This ensures that the key required to decrypt the image remains secure.

V. Conclusion and Future Work

In this paper, we have explored the use of AES and RSA encryption algorithms for image encryption. These encryption techniques provide a strong layer of security for sensitive visual data. In the future, further research can be conducted to optimize the performance of these algorithms for real-time image encryption applications. Additionally, exploring other cryptographic methods for image encryption can be an interesting avenue for future work.

Acknowledgment

The authors would like to acknowledge the support of Dr. Wang throughout the semester and along the project for the foundational ideas in Data Privacy and Security.

References

[1] I. Nti, E. Gymfi, and O. Nyarko-Boateng, "Implementation of advanced encryption standard algorithm with key length of 256 bits for preventing data loss in an organization," *International Journal of Advancements in Technology*, vol. 08, 01 2017.