Abstract— This paper will discuss the implementation of a project that uses AES (Advanced Encryption Standard) with the purpose of encrypting and decrypting images so that they may be passed securely.

1. Introduction

In today's day and age, sensitive images are constantly being sent back and forth between parties. The need of keeping these sensitive images secure is the motivation behind this paper. The AES algorithm is currently used by the U.S. government to protect classified information. Because we use AES, the sender and receiver of the image must have shared the key beforehand. By using AES, the image is encrypted so that an interceptor of the image cannot discern what the image may be. [1]

With AES if the key is kept a secret it is extremely difficult to decrypt the image.

Section 2 will touch on works related to this project. Later, in section 3 we discuss the implementation of the project. In section 4, we discuss tests and data used. Finally, in section 5, we discuss the results of the tests.

2. Background and Related Work

Many other researchers have already implemented successful implementations of AES image encryption, and documented their findings in papers. However, most of these papers simply describe how AES works, and not how their image encryption algorithm is implemented.

3. Methodology

Currently, the project has successfully implemented the encryption and decryption of the AES algorithm on 128-bit blocks. As of the writing of this milestone, the function that converts the image to matrices is in progress. The project is being developed in the IntelliJ IDE.

The sender of the image will be able to enter the plain image and key value that was agreed on by both parties. The web app then begins to encrypt and rebuild a scrambled image.

The planned method of image encryption is to convert each pixel's RGB value to hexadecimal form and build a matrix to store many them. Next, we encrypt each matrix using AES and build a scrambled image using the resulting ciphertext containing new RGB values. The web app then returns the cipher image as a downloadable image. The sender then sends the downloaded image to the receiver.

When the receiver gets the image, they can enter the cipher image and key into the web app to decrypt the image and download the plain image.

4. Experiments

The tests performed so far include ensuring the state is correctly encrypted, and then successfully decrypted. Other tests that are meant to be added later include encrypting and decrypting images, ensuring the web app can handle incorrect user

inputs, using images of different sizes, and ensuring the image content cannot be discerned.

5. Discussion

So far, the project has been going well, and the tests are successful. Moving forward, I may try using parallel processing to speed up the encryption and decryption of the images.

6. Conclusion

Overall, the project is a success so far, although it is far from done. Some other features that would be helpful is adding a ability to change key lengths, a more attractive UI, an ability to hide messages in the images, and the ability to encrypt other files like PDFs.

7. References

[1]. P. Radhadevi, P. Kalpana, “Secure Image Encryption using AES”, P. RADHADEVI* et al, Volume: 1 Issue: 2, ISSN: 2319-1163, page 115-117.