Antonino Tan-Marcello
MSCS 630
File Encryption on Storage Devices
Project Milestone
Due: 4/5/2017

Abstract

This paper illustrates the implementation of the Advanced Encryption Standard, more specifically as outlined by this paper, a Java program designed to securely encrypt and decrypt computer files. The program has shown to successfully encrypt several types of files, making the data within the files undecipherable and decoding them virtually impossible without the proper decryption procedure. When prompted, the program will also successfully decrypt the files back to their original state, as if they were never altered.

Introduction

Sensitive information must always be protected in one form or another. Securing information digitally, however, can actually be the most reliable, especially when AES comes into the picture. It is frequently seen in action movies where a computer chip/hard drive holds top secret information or programs that everyone wants to get their hands on. Most of the time, once it's in possession, whatever data was on that device is already exposed to that individual. However, if the data on that storage device has been encrypted, then it would be nearly impossible to get that decipher that data. The same idea can be applied to people who carry important information on their computers and storage devices. This leads up to the motivation behind this project, to directly encrypt computer files in order to protect information.

Background

A study conducted by Khati et al. titled Full Disk Encryption: Bridging Theory and Practice, have faced an issue of full disk encryption and storage limits. These researchers wanted to solve the issue of running out of storage space during the encryption process due to the additional storage of initialization vectors (IV) and/or message authentication codes (MAC). These researchers have proposed the notion of a diversifier, which does not use any additional storage space while allowing sectors of a hard drive to be encrypted. Khati et al. have shown that with a 2-bit diversifier, they can decrease the number of input/output operators. Perhaps a way Khati et al. could consider is to eliminate the authentication process, where a password is written to each file or each hard disk sector.

Methodology

The approach taken for this paper and project is to create a simple, user friendly encryption/decryption program for individuals who would like to encrypt specific files. Using Java as well as implementing AES at 128-bits, the goal is to efficiently build previously stated program. It was necessary to initially research published research papers, understand how AES works as well as getting familiar with designing a simple, user-friendly application.

Experiments

After completing a very early version of the program, it was able to correctly encrypt and decrypt various types of files using AES at 128-bits. Files used to test the program's encryption capabilities included several photo files varying in size as well as Word documents containing text. The resulting encrypted files were used to test the program's decryption ability, in order to see if the encrypted files can be transformed back to their original state.

Discussion

After performing some test cases, it is clear that the AES implementation towards the files has shown an obvious effect on their contents. For example, when the program encrypted a picture file, the photo no longer represents an actual picture when opened. Instead, when the file is opened, a text editor containing a many meaningless characters appear, leaving no trace of the original photo. Next after decrypting the encrypted picture file, the file opens and displays the same exact original photo, as if it was untouched. The same expected results occurred when encrypting a word document. When encrypted, the file opens showing undecipherable characters. Once decrypted, the original document and its contents are shown.

Conclusion

From the results obtained from the program so far, AES can be utilized to successfully secure computer files, as well as turning those files back to their original state. An application similar to the one applied to this project can be used to help people secure any type of document or personal data one may not want others to have access to. From our results, it is clear that encrypted files through AES can completely hide any data, and in order to get that information back, a decryption procedure with the correct key must be applied to those encrypted files. Otherwise, retrieving the encrypted information is virtually impossible.

References

1. Bossi, Simone, and Andrea Visconti. "What users should know about Full Disk Encryption based on LUKS." *International Conference on Cryptology and Network Security*. Springer International Publishing, 2015.
2. Chakraborty, Debrup, Cuauhtemoc Mancillas López, and Palash Sarkar. "Disk encryption: do we need to preserve length?." *Journal of Cryptographic Engineering* (2015): 1-21.
3. Khati, Louiza, Nicky Mouha, and Damien Vergnaud. "Full Disk Encryption: Bridging Theory and Practice." *Cryptographers' Track at the RSA Conference*. Springer, Cham, 2017.