Batch Encryption with AES in CBC Mode

Samantha Berry

Marist College, 3399 North Road Poughkeepsie, NY 12601, USA
{samantha.berry1}@a

Abstract. This is a milestone for the final project in MSCS 630. It outlines research and work has been completed so far and future additions to the project.

1 Introduction

For this project I have decided to make a software that does encryption and decryption using AES 128-bit in CBC mode. The software is intended to encrypt and decrypt files in large batches, allowing the user to preform operations on groups of files in a directory instead of having to select plaintext sources individually I built this project of off my experience in developing AES 128-bit encryption from the labs. For this project I went on to further my understanding of AES by researching and implementing AES 128-bit decryption and AES CBC mode.

I chose to make a batch encryption software because I wanted to expand my abilities with AES and because I was interested in alternate uses for encryption. Many well-known examples of encryption in the modern world concern data in transit. Data is encrypted before entering an unsecure network and then are decrypted at the other end. I wanted to know about other uses for encryption in secure data storage.

2 Background

The AES (Advanced Encryption Standard) was established in 2001 by the U.S. National Institute of standards and Technology (NIST). It was described by Daemen and Rijmen in their 1999 submission to the NIST in response to their search for a secure cypher for government data. In the modern world, AES is commonly used for encryption by both government and private groups.

The exact version of AES used in my project is Cipher Block Chaining (BCB) The principles for CBC implementation in computer ciphers was invented in 1976 and described in a paper by Ehrsam, Meyer, Smith, and Tuchman. In CBC mode each subsequent block of plaintext is XORed to the ciphertext from the last round. Since there is no previous input for the first round, it is XORed to a randomly generated input vector.

3 Methodology

For this project I intent to create a program that encrypts and decrypts files using AES in CBC. So far, I have implemented the AES encryption and decryption and tested it to ensure that they work correctly. I have also implemented a method that converts utf-8 plan text to 128-bit hex strings and back. This method has also been tested to ensure it is working correctly. This method will be used in the final program to convert input from files into a format that can be read by my AES encryption method and return the output of decryption into a human-readable format.

Before the project is completed, I have several elements left to implement. The main features yet to be added include AES CBC mode and file reading and writing. Once all these elements have been completed I will be able to implement all of them in a main driver method for user interactivity.

4 Testing

Since the elements of the program have not been combined into a cohesive whole yet, each element is tested separately.

Testing the AES encryption/decryption is done using Junit. For AES encryption I feed in randomly generated 128-bit strings of hex for the key and plaintext and compare the output to a known correct cyphertext. Testing AES decryption is done using the same set of data, but the cyphertext is fed in and it is checked to the plaintext. For the testing data set, some of it I generated myself using the random.org hex generator, and some I gathered from my classmates' tests for lab 5.

Testing of String to Hex conversion is also done using Junit. Known plaintext strings are converted to hex, converted back to plaintext and are compared to the original string.

5 Conclusions

No conclusions can be drawn at the moment since the project is not complete.

References

1.
2.