Project Design Document – CS 331

Project Title: AES Encryption/Decryption Showcase

Team Member(s): Audrey Fitchett, Braedon Kennedy, Cesar Ochoa, Leo Haener-Pope, Rhett Edwards

Objective

This project aims to demonstrate the encryption and decryption of a message using AES encryption. This will show the different steps of AES encryption and highlight how AES usage increases the security within a cyber system and ultimately provides more safety for both online and offline authentication and authorization.

This project is both important and relevant to this course because it demonstrates the importance of cryptography within cyber systems. Encryption provides a crucial step for making it exponentially harder for malicious entities to use stolen data, as only those with special knowledge of the keys are able to decrypt information. This provides a massive amount of cyber security and safety for systems that utilize data that is important to keep hidden from others.

Ultimately, this project addresses the problem of if data is intercepted by malicious entities, they can just simply use that information to do whatever they want. By encrypting and decrypting with AES, if attackers intercept data, they are ultimately unable to use it because only those with specific knowledge of the keys are able to decrypt and see what the information actually is. This project will provide more information about the usage and importance of cryptosystems like AES.

Scope

This project is going to include a demonstration program showing the encryption

and decryption (with steps) of a message that the user inputs to the program. It will feature the key generation required for AES, along with the methods that AES utilizes such as ShiftBytes(), ShiftRows(), and MixColumns(). The user will be able to use the console to enter a message into the program, to which the program will send back information, including the encrypted, then decrypted message along with other information about what is occurring during the encryption/decryption processes.

The target audience for this project are people who study cybersecurity and computer science. This could be people who are trying to gain a better understanding of encryption and decryption processes or those who just want to be shown the utilization of AES encryption functions in practice. This also includes people who want to be shown how data is kept private in computer environments.

Things to keep in mind include that this project will be just a demonstration of the processes that occur in just the encryption, decryption, and key management of AES in a controlled, internal offline environment, meaning that certain aspects of real-world AES systems may not be fully included in this project. Aspects such as network transport handling or key vaults/infrastructure are not things we aim to fully address in this project and may not be included in our program we create. AES also can be used with different sizes of keys, and for our project we will utilize 256-bit key encryption/decryption. If we find we have the time, we can further implement the other 192-bit, and 128-bit AES encryption methods. Also, if we find time, we can expand our project through implementing more advanced and secure random number generation, and explore different key handling techniques/methods commonly used for AES.

Security Concepts Applied

This ultimately falls under Encryption and Key Management, through its encryption of data and keys which are only known between 2 communicating individuals. Users can give a message to be encrypted to our project, and the encrypted version of the message will be shown, and then the decrypted (real) message will be

shown to demonstrate the use of AES cryptography and the private key that is shared between 2 individuals.

Though not actually keeping the information private/confidential, this project demonstrates the usage of AES to create unreadable data to those who are not supposed to have specific access to it, keeping information private and only known by those who actually need to know it, meaning this can fall under Confidentiality in the CIA Triad. Though not directly falling under access control and authentication/authorization, this project also promotes these parts of cyber security by demonstrating how we can make these parts/environments of cyber security even safe through the use of cryptography.

Methodology

1. **System Design Overview**

- The user will be communicating with the program through the console and the program will output back into the console
- Symmetric block cipher
- Cryptographic keys of 256 bits (may implement 192 and 128)
- Organizes bytes into a grid in column major order
- Plaintext is xor'd with part of the key (byte)
- Each byte is then substituted
- Each row is shifted left by its row number
- Columns are then mixed by multiplying by a known matrix
- Round key is then added
- For 256 bit keys, this process is and will be repeated in our project for 14 rounds to ensure maximum security
- Along with main encrypt() and decrypt() functions, other functions such as shiftBytes() and mixColumns() will also be created, and any other helper function that we may need for encryption/decryption.
- Print statements, describing what certain parts of the program are doing will also be utilized to show users what is occurring during AES encryption/decryption.

2. **Data Collection & Preprocessing**

- We will primarily be collecting plaintext from the console
- If specifics for encryption are not provided then the program will generate the required information itself (such as a key)
- We may utilize a library of common words to generate random messages

3. **Implementation Details**

- We will be organizing our program on github
- Github projects will be utilized to track project progress and current issues
- Individuals may use their own coding environments but it likely be through Visual Studio Code
- The program will be implemented in Java
    - AesEncryptionApp (Main entry point)
    - AesEncryptor (encrypt, decrypt, generateKey, generateIv)
    - KeyManager(Class)
        - Generate, store, and retrieve keys and IV's
        - Maybe saving/loading from file
    - AesUtils (encodeBase)
        - Handle encoding/decoding (base64) byte conversion so that AesEncryptor is focused solely on encryption logic
    - EncryptionException (custom exception handling)

4. **Evaluation Metrics**

- To evaluate accuracy we will create some examples by hand and compare them to see if the program matches the encryption
- If we have enough time we may create program(s) that will attempt to decipher without the key
- Assert that plaintext before encryption and after decryption match
- Assert that all expected characters are supported
- Assert that encryption/decryption take a reasonable amount of time
- Assert that no memory leaks were discovered

Conclusion

To summarize: this project implements an AES encryption within a single device. Input is provided to the console and the output is sent back out to the console, data is not transmitted between devices. The 256 bit key will be the same to encipher and decipher. Possible things we may also implement: different size keys, a way to attempt to decipher without the key, other encryption/decryption methods. We hope that this project will demonstrate the importance of cryptography, specifically AES encryption/decryption, within computer environments and highlighting how confidentiality is maintained when communicating between different devices. This project will demonstrate the steps of AES encryption/decryption and help both us, and

the audience we present to, further extend our cybersecurity knowledge.

References

National Institute of Standards and Technology. “Advanced Encryption Standard (AES).” *Federal Information Processing Standards Publication*, US Department of Commerce, 26 May 2001, updated 9 May 2023. https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197