Final Report on

Encryption Using AES

Submitted to
SPL I Evaluation Committee
Bachelor of Science in Software Engineering
Institute of Information Technology
Noakhali Science and Technology University

Submitted By
Md. Altaf Hossain
altaf2516@student.nstu.edu.bd

Supervised By
Nazmun Nahar
nazmun@nstu.edu.bd

Submission Date:

10 September, 2023

Final Report on **Encryption using AES**

By

Md. Altaf Hossain
MUH2125034M
Year – 2 Term -1
altaf2516@student.nstu.edu.bd

Approved By

Nazmun Nahar
Lecturer
Institute of Information Technology
Noakhali Science and Technology University

Project Description

Introduction:

AES, or Advanced Encryption Standard, is a widely used and highly secure encryption algorithm that plays a fundamental role in ensuring the confidentiality and security of data in various digital applications. It was established as a standard by the U.S. National Institute of Standards and Technology (NIST) in 2001 and has since become the de facto encryption standard globally. AES is symmetric encryption, meaning the same key is used for both encryption and decryption. Here's an introduction to AES:

Symmetric Encryption: AES is a symmetric-key encryption algorithm, which means the same secret key is used for both encrypting and decrypting data. This key must be kept secret, as anyone with access to it can decrypt the data.

Block Cipher: AES operates on blocks of data, with a fixed block size of 128 bits (16 bytes). It processes these blocks sequentially, which is why it's often referred to as a block cipher.

Key Lengths: AES supports key lengths of 128, 192, and 256 bits. Longer keys generally provide stronger encryption, but they also require more computational resources.

Substitution-Permutation Network: AES uses a series of mathematical operations to transform each block of data. These operations include substitution (replacing each byte with another byte) and permutation (rearranging the bytes). This process is repeated multiple times (rounds) to ensure security.

Multiple Rounds: AES employs a varying number of rounds for encryption and decryption, depending on the key length. It uses 10 rounds for a 128-bit key, 12 rounds for a 192-bit key, and 14 rounds for a 256-bit key. Each round adds another layer of encryption and confusion to the data.

Security: AES is designed to be highly secure and has withstood extensive cryptanalysis. Its security is based on the strength of its key schedule, substitution-permutation network, and the complexity of its mathematical operations.

Efficiency: AES is not only secure but also efficient in terms of both encryption and decryption. This makes it suitable for real-time applications and resource-constrained

devices.

Wide Adoption: AES has become the standard encryption choice for many organizations, governments, and industries worldwide. It is used in a variety of applications, including securing internet communications (e.g., HTTPS), protecting files and data at rest, and safeguarding sensitive information in databases.

Standardization: AES's status as a NIST standard has contributed to its widespread adoption and interoperability.

Targeted Customer/ User:

Any Technical and non-Technical person who want to transfer files using encryption

Features and Description:

Description:

AES is a robust and widely trusted encryption algorithm that plays a crucial role in ensuring the confidentiality and security of data in various digital applications. It is a fundamental building block of modern cryptography.

Key Features:

AES Encrypt:

Description: Use AES for data encryption

Benefits: Sensitive file can be easily secured

AES Decrypt:

Description: Use AES for data decryption

Benefits: Only authorized people can access the file

User-Friendly Interface:

Description: Intuitive and straightforward interface for users of all backgrounds.

Benefits: Reduces the learning curve, making the project accessible to a wide range of users

Model:

In this project **'Waterfall model'** will be followed.
As it is a small project and the almost every requirement is defined and there is no scopeof changing the requirements so I decided to follow waterfall model throughout the project.

Resources:

- Book-java the complete reference
- GitHub

Language:

- Java

IDE:

- Vs code

User Guideline

Getting Started:

Launch the application by executing the main program.
You will be presented with a command-line interface.

AES Encrypt:
Type enc to encrypt files using the AES encryption algorithm.
Enter the 16 character key.
Provide the input file path and the output file path.
The tool will perform the operation and display a confirmation message.

AES Decrypt:
Type dec to decrypt files using the AES decryption algorithm.
Enter the 16 character key.
Provide the input file path and the output file path.
The tool will perform the operation and provide feedback.

Additional Notes:
Terminating the Application:
Type exit to terminate the application gracefully.

```
PS D:\SPL1_final> cd "d:\SPL1_final\" ; if ($?) { javac Main.java } ; if ($?) { java Main }
For AES encryption enter the command enc
For AES decryption enter the command dec
To terminate enter the command exit
Enter your choice: enc
Enter the 16 character key: altafhossain2002
Enter the file path to read data: D:\altaf\A.pdf
Enter the file path to write encrypted data: D:\altaf\k.pdf

Encryption Successful

For AES encryption enter the command enc
For AES decryption enter the command dec
To terminate enter the command exit
Enter your choice: dec
Enter the 16 character key: altafhossain2002
Enter the file path to read encrypted data: D:\altaf\k.pdf.enc
Enter the file path to write decrypted data: D:\altaf\new.pdf

Decryption Successful

For AES encryption enter the command enc
For AES decryption enter the command dec
To terminate enter the command exit
Enter your choice: exit
PS D:\SPL1_final>
```

Source Code Documentation

Description	Name	Location	Return Type	Access modifier
To encrypt using AES encryption algorithm	encrypt()	Class: AESEncrypt	void	public
To decrypt using decryption AES algorithm	decrypt ()	Class: AESDecrypt	void	public
To Expand key	expandKey()	Class: KEYExpand	Int[]	public
	main()	Class:Main	void	public

Challenges

- Key Management
- Integration of code
- Testing and Debugging

Future Work

- Adding a Key exchange Algorithm
- Making it web based application

Conclusion:

AES has proven to be a robust and dependable encryption algorithm, and its continued use is crucial in maintaining data confidentiality and integrity in an increasingly digital and interconnected world. As of my last knowledge update in September 2021, AES remains a cornerstone of modern cryptographic practices, though it's essential to stay updated with any advancements or changes in encryption standards and best practices.