CipherVault

Using Fernet Symmetric Encryption

Group - 15

Department : [Computer Science & Information Technology & Computer Science & Technology]

UNDER THE GUIDANCE OF

- Prof. Dr. Subhalaxmi Chakraborty

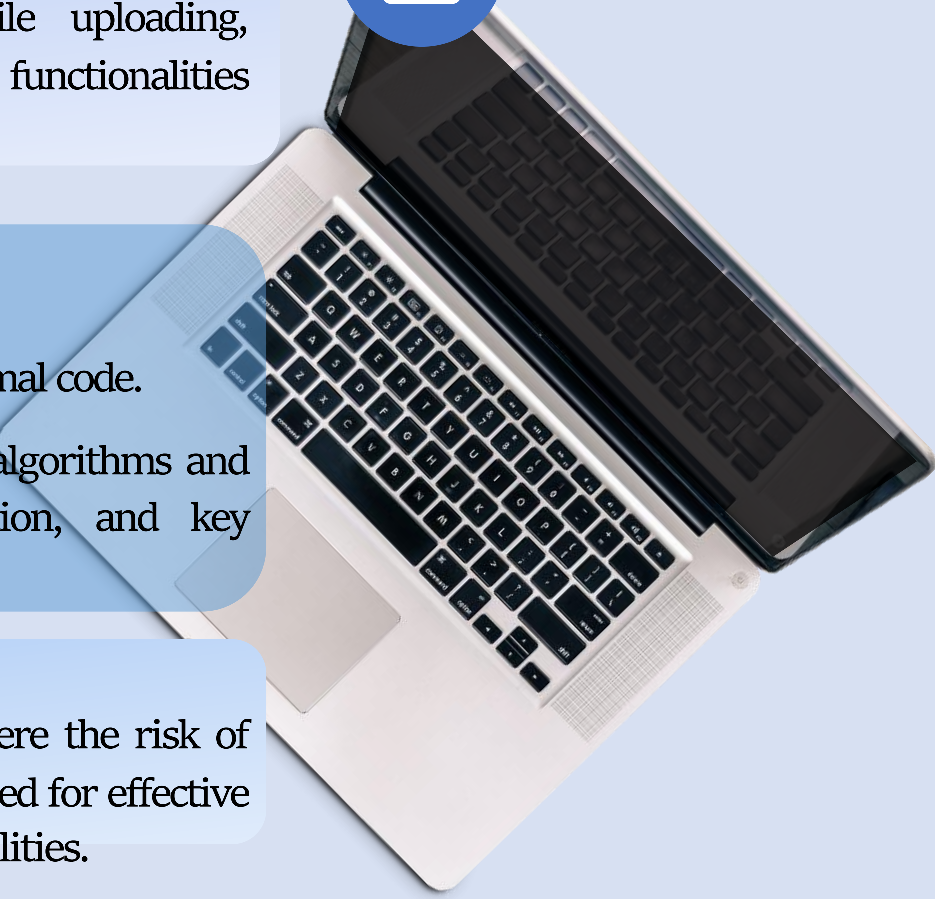
GROUP MEMBERS

- Shivsundar Bera
- Arindam Roy
- Soumyajit Patra
- Hriteesha Pramanik
- Sayandeep Mondal
- Swarnadeep Roy

PROJECT OVERVIEW

OBJECTIVE: The primary aim of this project is to develop a user-friendly web application that enables users to securely encrypt and decrypt files. By providing a seamless interface, the application aims to simplify the process of managing sensitive data while ensuring robust security measures.

SCOPE: The scope of the project encompasses various aspects, including file uploading, encryption/decryption operations, key management, and user interaction. The application's functionalities are designed to cater to both individual users and organizations with data security needs.



TECHNOLOGIES USED:

- Streamlit: A Python library used for building interactive web applications with minimal code.
- Cryptography library: A comprehensive library for implementing cryptographic algorithms and protocols in Python, providing essential functionalities for encryption, decryption, and key management.

IMPORTANCE: Data security is of paramount importance in today's digital age, where the risk of unauthorized access and data breaches is ever-present. This project addresses the critical need for effective encryption solutions to safeguard sensitive information from potential threats and vulnerabilities.

PROJECT STRUCTURE

1

PROJECT FILES

- **main.py:** Contains the code for the Streamlit web application.
- **style/request.css:** Local CSS file for custom styling.
- **Secret.key:** Stores the symmetric encryption key generated by the application.

2

KEY FUNCTIONS

Explanation of functions such as load_lottieurl(), generate_key(), encrypt(), decrypt(), etc.

3

DEPENDENCIES

- Python 3.x
- Streamlit
- Pillow (PIL)
- requests
- Cryptography

4

DETAILED DESCRIPTION

- **Main.py:** Importing libraries, defining functions, implementing web application logic.
- **CSS file:** Customizing visual appearance using CSS.
- **Secret.key:** Storing and managing encryption keys for data security.

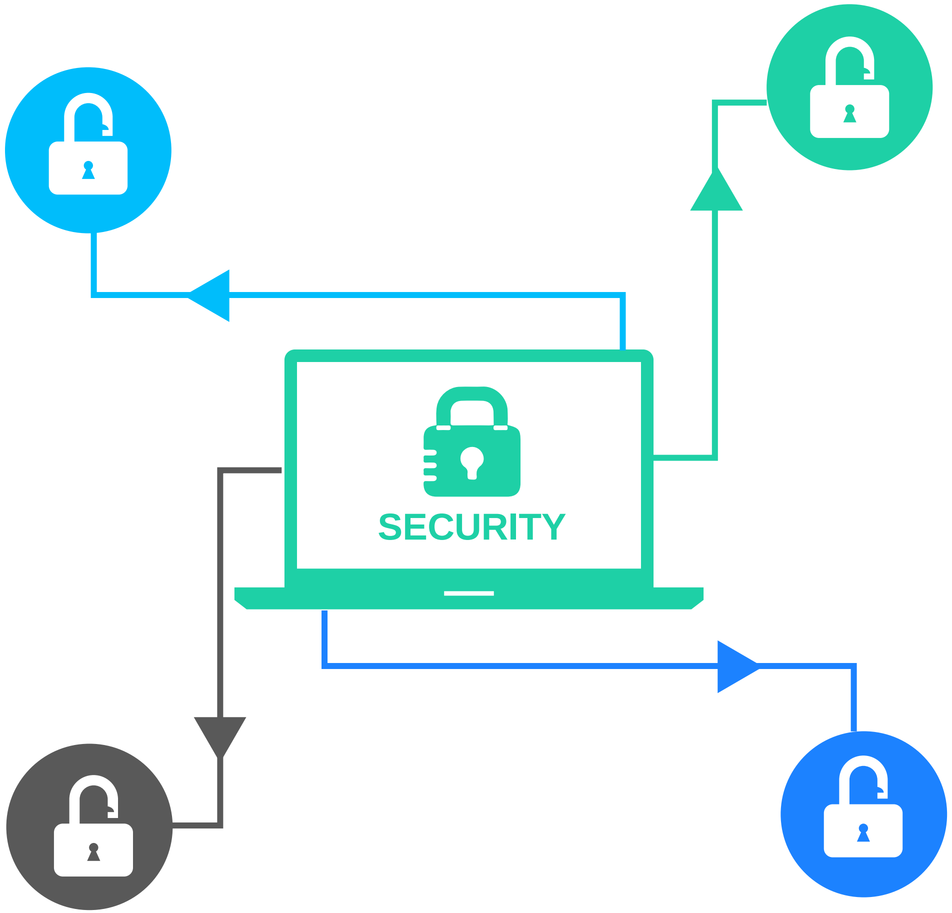
ENCRYPTION ALGORITHMS

AES (Advanced Encryption Standard)

AES (Advanced Encryption Standard) is a widely adopted symmetric encryption algorithm known for its security and efficiency. With support for key sizes of 128, 192, or 256 bits, it offers scalability to match different security requirements. AES operates through substitution-permutation network (SPN) structure, making it resistant to various cryptographic attacks. Its widespread adoption in securing communication channels and data at rest highlights its reliability. Overall, AES stands as a robust solution for ensuring confidentiality and integrity in modern information systems.

DES (Data Encryption Standard)

DES (Data Encryption Standard) is a symmetric encryption algorithm initially developed in the 1970s. It operates on 64-bit blocks of data with a 56-bit key. Despite being widely used in the past, DES is now considered insecure due to its small key size. It uses a Feistel network structure, where data undergoes multiple rounds of permutation and substitution. Due to vulnerabilities and advancements in cryptanalysis, DES has largely been replaced by more secure algorithms like AES.



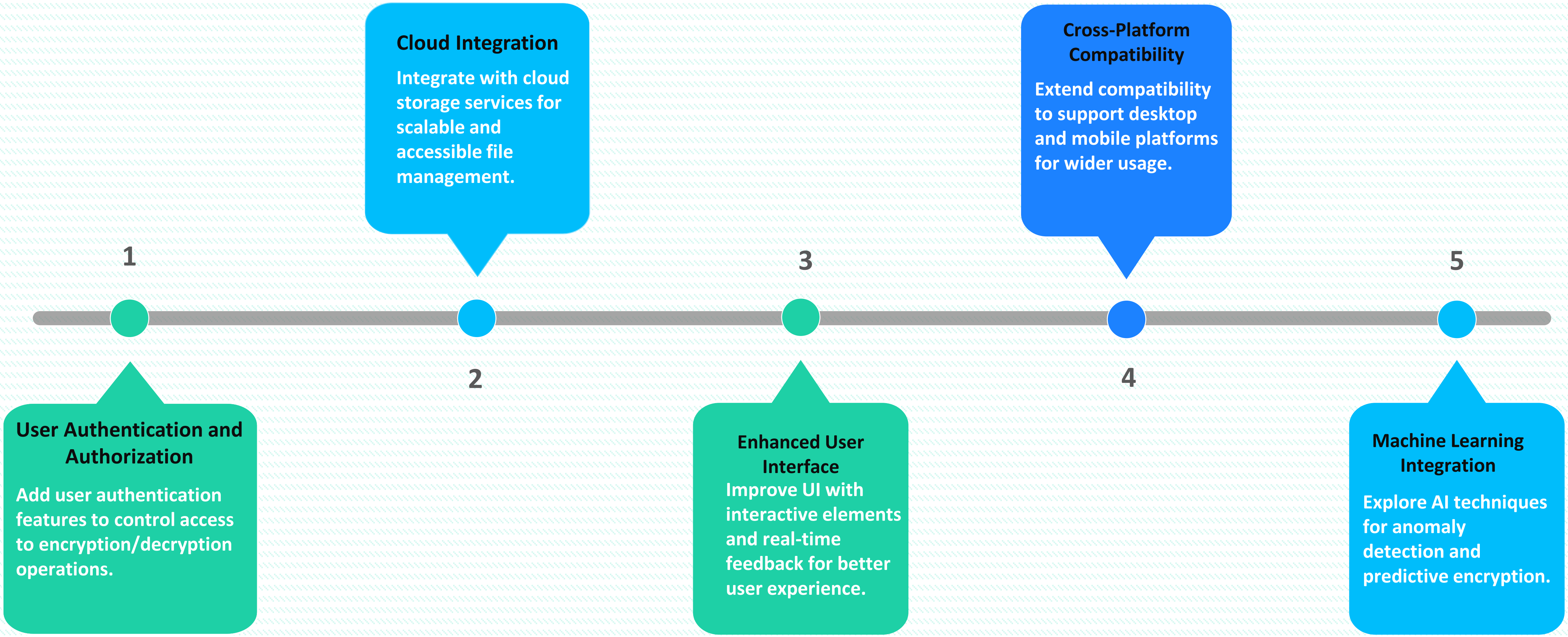
Triple DES (3DES)

Triple DES (3DES) is an encryption technique that enhances the security of the Data Encryption Standard (DES) by applying the algorithm three times consecutively. It involves encrypting data with one key, decrypting it with another, and then encrypting it again with a third key. Despite providing increased security compared to DES, 3DES is slower due to multiple encryption rounds and has been largely replaced by more efficient algorithms like AES in modern cryptographic systems. However, it is still occasionally used in legacy systems and where backward compatibility is necessary.

Blowfish

Blowfish is a symmetric encryption algorithm designed by Bruce Schneier in 1993. It operates on 64-bit blocks of data and supports variable key lengths up to 448 bits, making it adaptable to different security needs. Blowfish employs a Feistel network structure with 16 rounds of encryption. While not as widely used as AES, Blowfish remains popular for its simplicity, speed, and flexibility, particularly in applications where resource constraints are a concern.

Future Prospects



1

User Authentication and Authorization

Add user authentication features to control access to encryption/decryption operations.

2

Cloud Integration

Integrate with cloud storage services for scalable and accessible file management.

3

Enhanced User Interface

Improve UI with interactive elements and real-time feedback for better user experience.

4

Cross-Platform Compatibility

Extend compatibility to support desktop and mobile platforms for wider usage.

5

Machine Learning Integration

Explore AI techniques for anomaly detection and predictive encryption.

CONCLUSION

01 Recap Of Achievements

Successfully developed a secure file encryption and decryption web application.

02 Benefits

Enhanced data security, user-friendly interface, customizable features..

03 Key Takeaway

Importance of data encryption, role of technology in securing sensitive information.

THANK YOU