AES and DES Encryption Web Application

1. Project Title

AES and DES-Based Secure Text Encryption Web Application

---

2. Project Overview

This project is a simple web-based application that demonstrates how data can be protected using **AES (Advanced Encryption Standard)** and **DES (Data Encryption Standard)** encryption. The system allows users to enter plain text, encrypt it securely using AES or DES, and then decrypt it back to its original form.

The main goal of the project is to show a practical implementation of symmetric encryption in the field of **Information Security**, focusing on simplicity, correctness, and clear understanding rather than complexity.

---

3. Objectives of the Project

- Demonstrate how AES and DES encryption works in practice.
- Protect text data from unauthorized access.
- Show the process of encryption and decryption using the same secret key.
- Build a simple and clear web interface for user interaction.
- Apply theoretical security concepts in a real working system.

---

4. Why AES and DES?

AES (Advanced Encryption Standard) and DES (Data Encryption Standard) are widely used symmetric encryption algorithms.

Reasons for using both: - AES: Strong security, fast, widely adopted, supports multiple key sizes. - DES: Older standard, simpler, demonstrates the evolution of encryption standards. - Comparing AES and DES helps understand **security vs simplicity trade-offs**.

In this project, **AES-128** is used for strong security, while **DES with a derived 8-byte key** is included for educational comparison.

---

5. System Architecture

The system is divided into the following components:

1. **encrypt.py** – Handles AES encryption logic.

2. **decrypt.py** – Handles AES decryption logic.
3. **encrypt_des.py** – Handles DES encryption logic.
4. **decrypt_des.py** – Handles DES decryption logic.
5. **app.py** – Flask backend that connects encryption logic with the web interface.
6. **index.html** – Frontend web page for user interaction.

---

6. Technologies Used

- **Programming Language:** Python
- **Web Framework:** Flask
- **Encryption Library:** PyCryptodome
- **Frontend:** HTML + Bootstrap
- **Encryption Algorithms:** AES and DES (Symmetric Encryption)

---

7. AES Encryption Process

1. The user enters plain text in the web interface.
2. The text is sent to the Flask backend.
3. The encryption function uses:
4. AES algorithm
5. Secret key
6. Secure encryption mode
7. The output is encrypted binary data.
8. The encrypted data is encoded using **Base64** to make it readable and transferable.
9. The encrypted text is displayed to the user.

---

8. AES Decryption Process

1. The user enters the encrypted Base64 text.
2. The backend decodes it back to binary data.
3. The same AES secret key is used.
4. The original plain text is restored.
5. The decrypted text is displayed.

---

9. DES Encryption Process

1. The user enters plain text.
2. A password is optionally used (default provided).
3. A **key is derived** from the password (first 8 bytes of SHA-256 hash).
4. DES in CBC mode is used to encrypt the text.
5. Initialization Vector (IV) is generated and concatenated with ciphertext.
6. Output is Base64 encoded for safe display and transfer.

---

10. DES Decryption Process

1. User provides the Base64 encrypted text.
2. Backend decodes Base64 to retrieve IV and ciphertext.
3. The same password/key is used.
4. DES decrypts the ciphertext using CBC mode and IV.
5. Plain text is restored and displayed.

---

11. Role of Base64 Encoding

Base64 is not encryption. It is used to: - Convert binary encrypted data into readable text. - Allow encrypted data to be safely displayed and transferred in web applications.

AES and DES provide security, while Base64 provides compatibility.

---

12. Security Discussion

- AES and DES are symmetric encryption algorithms, meaning the same key is used for encryption and decryption.
- If the secret key/password is kept secure, the data remains protected.
- AES is strong against brute-force attacks; DES is weaker but included for educational purposes.
- The project demonstrates confidentiality, a core principle of information security.

---

13. Conclusion

This project successfully demonstrates how AES and DES encryption can be implemented in a real web application. It combines theoretical security concepts with practical coding, making it a strong example of applying cryptography in information security systems.

The system is simple, secure, and effective for educational purposes and project discussion.