Title: Cryptography Algorithms Implementation (AES & RSA)

Author: Sulaiman

Organization: CodeC Technologies — Cyber Security Internship

Abstract:

This project demonstrates basic implementations of symmetric and asymmetric cryptography using AES (EAX mode) and RSA (PKCS1_OAEP) in Python. The aim is to show fundamental operations: encryption, decryption, key generation, and usage examples to secure small files and messages. These implementation-level demos are designed for educational purposes and to illustrate cryptographic concepts that are widely used in real systems.

Tools Used:

- Python 3

- PyCryptodome library (Crypto)

- Text editor / GitHub

Working Steps:

1. AES (symmetric)

- The AES demo reads a file in binary, encrypts using AES EAX mode (nonce, tag, ciphertext) and writes out a .enc file.

- Decryption reads nonce, tag and ciphertext and verifies integrity before restoring the original file.

- Key is passed as hex for demonstration; in production a secure key derivation & storage (KMS) is required.

2. RSA (asymmetric)

- The RSA demo generates a 2048-bit keypair (private.pem & public.pem).

- Encryption uses the public key with PKCS1_OAEP padding; decryption uses the private key.

- This demonstrates secure message exchange where only the private key holder can read the message.

Conclusion:

These demos provide clear, testable examples of cryptographic building blocks. They are educational and suitable for documentation and interview discussion. For real-world deployment, use secure key storage, strong randomness sources, and vetted libraries with proper configuration.

Date: <23-10-2025>