1

Ahmed_Crypto Tool Assignement

introduction

The Ahmed_Crypto Tool is a Python-based cryptographic tool designed to handle a variety of cryptographic operations including encryption, decryption, hashing, digital signatures, and signature verification. The tool supports both symmetric and asymmetric encryption algorithms, such as AES, RSA, and RC4, along with a suite of hashing algorithms like MD5, SHA-1, and SHA-256. This tool is designed for both novice and advanced users, allowing them to secure data and verify the integrity of messages.

Key Features

1. File upload and text input.
2. AES encryption and decryption.
3. RSA encryption and digital signatures.
4. RC4 stream cipher encryption.
5. Supports hash generation using MD5, SHA-1, and SHA-256.
6. Blind and regular digital signature creation.
7. Signature verification.

Algorithms

1 AES Algorithm

AES is a symmetric block cipher that encrypts data in fixed-size blocks (128 bits) using a secret key (128, 192, or 256 bits).

Library — cryptography.hazmat for encryption/decryption

2 RSA Algorithm

RSA is an asymmetric cryptographic algorithm used for secure data transmission. It uses a pair of keys: a public key for encryption and a private key for decryption.

Library — cryptography.hazmat and PyCryptodome.

3 RC4 Algorithm

RC4 is a stream cipher that encrypts data one byte at a time using a variable-length key.

Library — PyCryptodome.

4 Hash Functions

Ensure data integrity by generating a unique hash value for any given input text | MD5 - SHA128 - Sha256

Library — hashlib

5 Digital Signatures

Signing and verifying messages to authenticate the sender and ensure data integrity.

Library — cryptography.hazmat

2

Functional Overview

Symmetric Encryption (AES)

Purpose — Encrypt and decrypt data using the AES algorithm.

Mode — CBC (Cipher Block Chaining) with a fixed IV (Initialization Vector).

key — A 16-byte symmetric key for both encryption and decryption.

Usage — Encrypt sensitive data, then decrypt it using the same key.

RSA Key Pair Generation

Purpose — Generate a pair of public and private RSA keys for encryption and signing.

Algorithm — RSA (2048-bit key size).

Usage — The public key is used for encryption and signature verification, while the private key is used for decryption and signing.

RC4 Stream Cipher Encryption

Purpose — Encrypt data using the RC4 stream cipher.

key — 16-byte key used for both encryption and decryption.

Usage — Suitable for fast encryption tasks where speed is more critical than security.

Digital Signature Generation and Verification

Purpose — Generate and verify digital signatures to ensure data integrity and authenticity.

Algorithm — RSA with SHA-256 hashing.

Usage — Sign a message or file to prove authenticity and verify its integrity using the public key.

Hashing Algorithms

Purpose — Generate a hash value of a text input.

Functions — MD5, SHA-1, SHA-256.

Usage — Hash messages or files to check their integrity.

Blind Signature

Purpose — A technique that allows a signature to be generated without revealing the content of the message.

Usage — Often used in privacy-preserving applications.