Final Year B.Tech. (CSE) – VII [2024-25]

6CS451: Cryptography and Network Security Lab (C&NS Lab)

Date: 26/08/2024

Assignment 6

PRN: 21510042 **Name:** Omkar Rajesh Auti

---

1. Apply AES algorithm for practical applications

Ans:

The Advanced Encryption Standard (AES) is a widely used symmetric encryption algorithm that is both fast and secure. It is the standard encryption algorithm used by governments, financial institutions, and many other organizations. Unlike DES, which is now considered insecure, AES is robust and provides a high level of security.

Practical Application of AES Algorithm

We can use the **pycryptodome** library in Python to implement AES encryption and decryption. The AES algorithm can work with key sizes of 128, 192, or 256 bits, and it operates on 128-bit blocks. In this example, we'll use AES with a 256-bit key in Cipher Block Chaining (CBC) mode.

Python Code:

```
from Crypto.Cipher import AES
from Crypto.Util.Padding import pad, unpad
from Crypto.Random import get_random_bytes


def aes_encrypt(plain_text, key):
    """
    Encrypt the plain text using AES algorithm.

    Parameters:
```

```
    plain_text (str): The text to be encrypted.
    key (bytes): The encryption key (must be 16, 24, or 32
bytes long).

    Returns:
    bytes: The initialization vector (IV) and the encrypted
cipher text.
    """
    cipher = AES.new(key, AES.MODE_CBC)
    iv = cipher.iv  # Initialization vector
    padded_text = pad(plain_text.encode(), AES.block_size)
    encrypted_text = cipher.encrypt(padded_text)
    return iv, encrypted_text


def aes_decrypt(iv, cipher_text, key):
    """
    Decrypt the cipher text using AES algorithm.

    Parameters:
    iv (bytes): The initialization vector used during
encryption.
    cipher_text (bytes): The encrypted text to be decrypted.
    key (bytes): The decryption key (must be 16, 24, or 32
bytes long).

    Returns:
    str: The decrypted plain text.
    """
    cipher = AES.new(key, AES.MODE_CBC, iv)
    decrypted_text = unpad(cipher.decrypt(cipher_text),
AES.block_size)
    return decrypted_text.decode()


def main():
    """
    The main function to run the program.
    """
```

```
    print("\nAES Encryption and Decryption")

    # Generate a random 32-byte key for AES (256-bit)
    key = get_random_bytes(32)
    print(f"\nGenerated Key (in hexadecimal): {key.hex()}")

    # Input plaintext
    plain_text = input("\nEnter the plain text to encrypt:
")

    # Encrypt the plaintext
    iv, encrypted_text = aes_encrypt(plain_text, key)
    print(f"\nInitialization Vector (IV) (in hexadecimal):
{iv.hex()}")
    print(f"\nEncrypted Text (in hexadecimal):
{encrypted_text.hex()}")

    # Decrypt the ciphertext
    decrypted_text = aes_decrypt(iv, encrypted_text, key)
    print(f"\nDecrypted Text: {decrypted_text}")


if __name__ == "__main__":
    main()
```

Output:

```
PS C:\Users\omkar\OneDrive\Desktop\SEM7\CNS LAB> python -u "c:\Users\omkar\OneDrive\Desktop\SEM7\CNS LAB\Assignment 6\aes.py"

AES Encryption and Decryption

Generated Key (in hexadecimal): 9386a2e13110e424f7db8286de5303d18067a35f3e34352d97d93a062c7eb7d1

Enter the plain text to encrypt: Keep it secret!

Initialization Vector (IV) (in hexadecimal): d1802431c456d8fbe65149f3f6cf8f26

Encrypted Text (in hexadecimal): 9a46a061646697d48d93625d2a604423

Decrypted Text: Keep it secret!
PS C:\Users\omkar\OneDrive\Desktop\SEM7\CNS LAB>
```

Practical Applications of AES:

- **File Encryption**: Encrypting sensitive files before storing them on disk.

- **Secure Communication:** Ensuring that data sent over the network remains confidential.
- **Data Protection in Applications:** Encrypting user data, such as passwords, to protect them from unauthorized access.

AES is widely adopted due to its strength and efficiency, and it remains the standard for securing digital data across various industries.