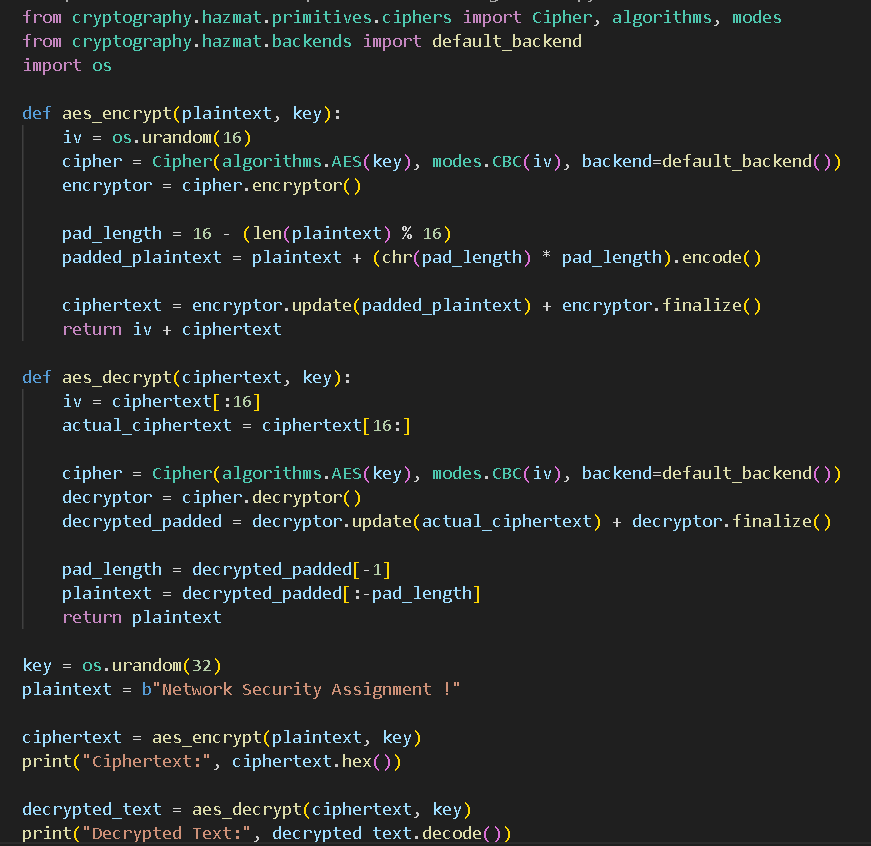
**Implementation of AES Encryption Algorithm**

|  |  |
| --- | --- |
| **Supervisor**  **Dr. Narendran Rajagopalan,**  **Associate Professor,**  **NITPY.** | **Assignment by,**  **Tella Praneeth (CS22B1055).** |

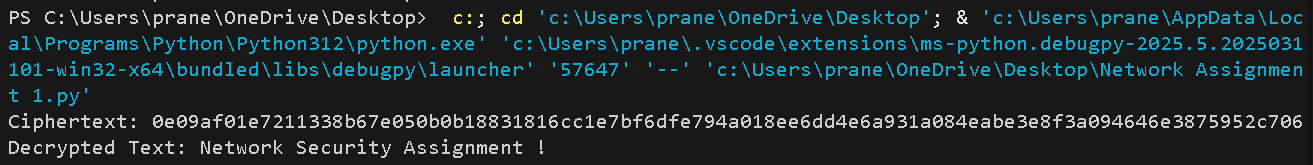
**Introduction**

Advanced Encryption Standard (AES) is a symmetric encryption algorithm widely used for securing data. It operates on fixed block sizes of 128 bits and supports key sizes of 128, 192, or 256 bits. In this assignment, we implement AES encryption and decryption using Python's cryptography library.

**Implementation:**



**Output:**



**Explanation:**

* Key Generation: We generate a 256-bit (32-byte) random key.
* Initialization Vector (IV): A 16-byte random IV is generated to ensure encryption uniqueness.
* Padding: AES requires input to be a multiple of 16 bytes. If needed, we append padding.
* Encryption: The plaintext is encrypted using AES in CBC mode.
* Decryption: The ciphertext is decrypted, and the padding is removed.
* Output Verification: The decrypted text matches the original plaintext, ensuring successful encryption and decryption.

**Conclusion:**

This implementation demonstrates AES encryption and decryption using Python. AES provides strong security and is widely used for data protection. The correct decryption output verifies the accuracy of our implementation.