CSCI 663G Group Project Proposal

Group D Members:

Keita Sakurai

Nirak Choun

Seanglong Lim

Our project repository on GitHub.

Project design for 2. AES:

Directory structure

```
├── aes.py
├── aes_utils.py
├── tests
      └── test_aes.py
      └── test_aes_system.py
```

aes.py: Main module containing the AES implementation.
We will implement AES encryption with a round key in the encrypt function. We will also implement AES decryption with a round key in the decrypt function. Specifying aes_cipher(Key), ciphertext, and decrypted_text.

aes_utils.py: Utility module containing helper functions for key expansion, substitution, andother AES operations.
We wil create key_expansion and sub_bytes functions which performe to help. for key expansion, substitution, and other AES operations.

tests/: Directory containing unit test cases and system test cases.

Unit Test Cases: Create unit tests for the AES class and utility functions using a testing framework like unittest. Create the following four functions and test each one:

```
def test_key_expansion:
def test_sub_bytes:
def test_aes_encryption:
def test_aes_decryption:
```

System Test Code: Create a system test to verify the overall functionality of the AES implementation.

Project design for 4. The Diffie–Hellman key exchange:

Directory structure

```
diffie_hellman_key_exchange/
├── diffie_helleman.py
├── tests/
      └── test_diffie_hellman.py
```

diffie_hellman.py: Create class DiffieHellman and then, define the following four functions which is to initialize variables, to generate private key, public key, and shared secret key, respectively.

```
def __init__(self, ...):
def generate_private_key:
def generate_public_key:
def generate_shared_secret:
```

tests/test_diffie_hellman.py: Create class TestDiffieHellman and define test key exchange function to test key exchange.

Whole System Test Code Structure (main.py or another entry point): Create main.py and try Example of using DiffieHellman for key exchange.