DES Cipher Implementation - Readme

Objective

The objective of this project is to implement the Data Encryption Standard (DES) algorithm from scratch, without relying on any external libraries. The implementation includes all elements of the 16 rounds of DES, such as the F-box, 32-bit exchanges, and the generation of sub-keys for each round. The program is designed to encrypt and decrypt a 64-bit plaintext using DES, and it verifies the correctness of the implementation using at least three pairs of `<plaintext, ciphertext>`.

Implementation Details

Key Components

1. **Initial Permutation (IP)**: The plaintext is permuted using the initial permutation table before the rounds begin.
2. **Key Generation**: The 56-bit key is permuted and divided into two halves. Sub-keys are generated for each round using circular shifts and compression.
3. **Round Function (F-box)**: The F-box applies expansion, XOR with the round key, substitution using S-boxes, and permutation to the right half of the data.
4. **16 Rounds of DES**: The algorithm performs 16 rounds of encryption and decryption, with the output of each round being stored for verification.
5. **Final Permutation (FP)**: After the 16 rounds, the final permutation is applied to produce the ciphertext or decrypted plaintext.

Verification

The program verifies the correctness of the implementation by:

1. **Decryption Verification**: Ensuring that the ciphertext, when decrypted, yields the original plaintext.
2. **Round Output Verification**:
    - Verifying that the output of the 1st encryption round matches the output of the 15th decryption round.
    - Verifying that the output of the 14th encryption round matches the output of the 2nd decryption round.

Test Cases

Three pairs of `<plaintext, ciphertext>` are used for testing:

1. `{'plaintext': '0123456789ABCDEF', 'key': '133457799BBCDFF1'}`

2. `{'plaintext': 'FEDCBA9876543210', 'key': 'AABBCCDDEEFF0011'}`
3. `{'plaintext': '0000000000000000', 'key': 'FFFFFFFFFFFFFFFF'}`

Output

The program outputs the following:

1. **Encryption Round Outputs**: The output of each encryption round is stored and printed.
2. **Decryption Round Outputs**: The output of each decryption round is stored and printed.
3. **Final Ciphertext**: The final encrypted output for each pair.
4. **Final Decrypted Plaintext**: The final decrypted output for each pair.
5. **Verification Results**: The program verifies the correctness of the round outputs and prints the results.

How to Run the Program

1. **Prerequisites**: Ensure you have Python installed on your system.
2. **Run the Script**: Execute the script `code.py` using Python:

```
python code.py
```

3. **Output Files**: The program generates two output files:
    - `encryption.txt`: Contains the encryption round outputs and final ciphertext for each pair.
    - `decryption.txt`: Contains the decryption round outputs and final decrypted plaintext for each pair.

Code Structure

- **Tables & Constants**: The script defines all the necessary tables and constants for DES, including the initial permutation table, key permutation table, S-boxes, and final permutation table.
- **Helper Functions**: Functions for circular shifts, binary-to-hex conversion, hex-to-binary conversion, and the round function (F-box) are implemented.
- **Main Logic**: The main logic handles the encryption and decryption processes, including the generation of round keys, application of the round function, and storage of round outputs.
- **Verification**: The program verifies the correctness of the implementation by comparing the outputs of specific encryption and decryption rounds.

Example Output

Encryption

```
=== Pair 1 Encryption ===
After initial permutation: 14A7D67818CA18AD
Round key: 1B02EFFC7072
Output in Round 1 is 5A78E3945A78E394
...
Final encrypted output is 85E813540F0AB405
```

Decryption

```
=== Pair 1 Decryption ===
Round key: 1B02EFFC7072
Output in Round 1 is 5A78E3945A78E394
...
Final output is 0123456789ABCDEF
```

Verification

```
=== VERIFICATION OF ROUND OUTPUTS ===
Pair 1: Encryption Round 1 equals Decryption Round 15: 5A78E3945A78E394
Pair 1: Encryption Round 14 equals Decryption Round 2: 5A78E3945A78E394
```

Conclusion

This project successfully implements the DES algorithm from scratch, including all components of the 16 rounds. The implementation is verified using three test cases, ensuring that the ciphertext can be decrypted back to the original plaintext and that the round outputs match as expected. This project demonstrates a deep understanding of the DES algorithm and its internal workings.