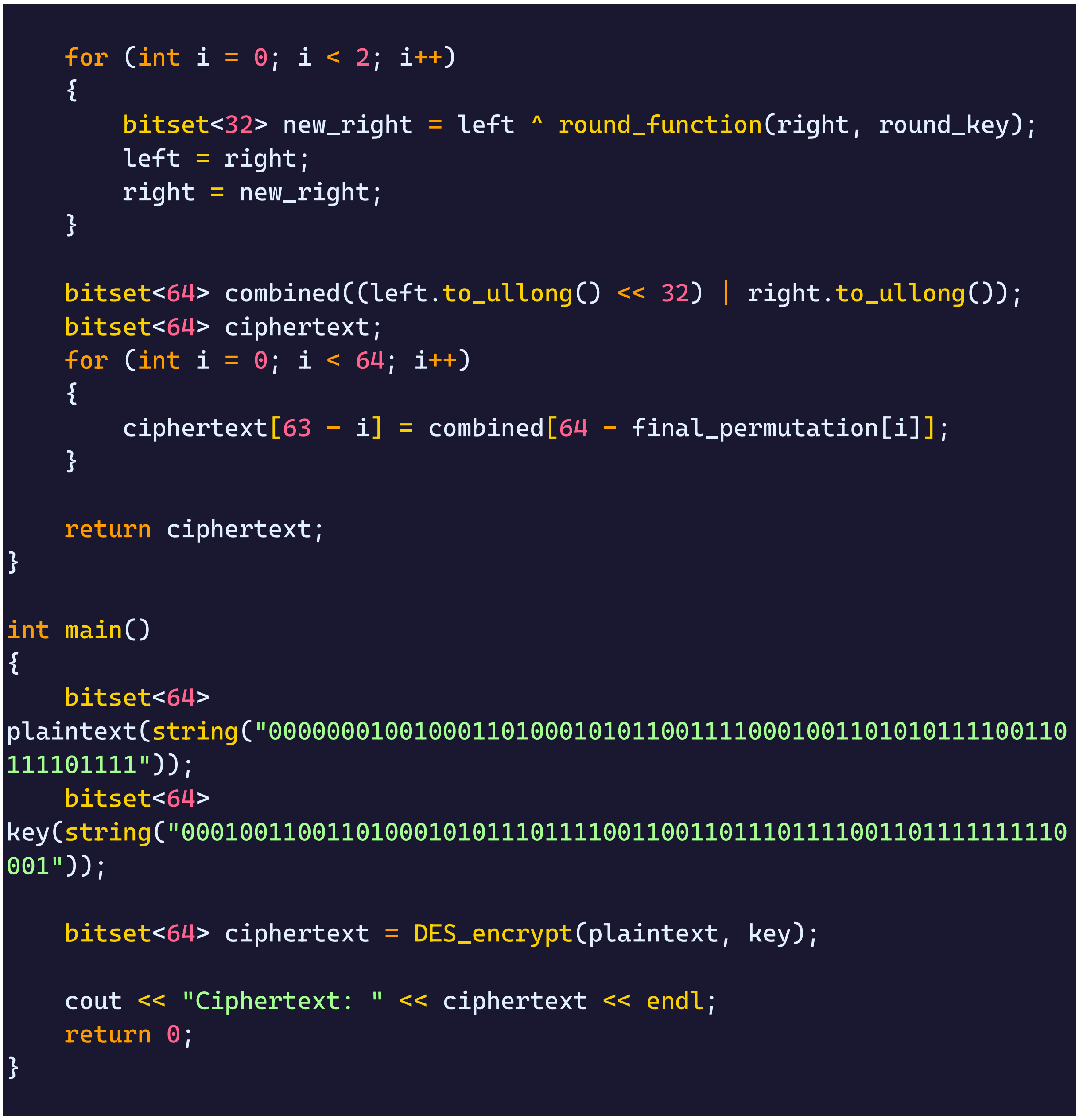
# Cryptography, Network and Security

## Assignment 5

Apply DES algorithm for practical applications

Code:

```
#include <iostream>
#include <bitset>
#include <vector>

using namespace std;

// Define the initial permutation table
int initial_permutation[64] = {
    58, 50, 42, 34, 26, 18, 10, 2, 60, 52, 44, 36, 28, 20, 12, 4,
    62, 54, 46, 38, 30, 22, 14, 6, 64, 56, 48, 40, 32, 24, 16, 8,
    57, 49, 41, 33, 25, 17, 9, 1, 59, 51, 43, 35, 27, 19, 11, 3,
    61, 53, 45, 37, 29, 21, 13, 5, 63, 55, 47, 39, 31, 23, 15, 7};

// Define the final permutation table
int final_permutation[64] = {
    40, 8, 48, 16, 56, 24, 64, 32, 39, 7, 47, 15, 55, 23, 63, 31,
    38, 6, 46, 14, 54, 22, 62, 30, 37, 5, 45, 13, 53, 21, 61, 29,
    36, 4, 44, 12, 52, 20, 60, 28, 35, 3, 43, 11, 51, 19, 59, 27,
    34, 2, 42, 10, 50, 18, 58, 26, 33, 1, 41, 9, 49, 17, 57, 25};

// Dummy function for round function and key schedule (simplified for
demonstration)
bitset<32> round_function(bitset<32> right, bitset<48> key)
{
    return right ^ bitset<32>(key.to_string().substr(0, 32));
}

// DES encryption function
bitset<64> DES_encrypt(bitset<64> plaintext, bitset<64> key)
{
    bitset<64> permuted_text;

    for (int i = 0; i < 64; i++)
    {
        permuted_text[63 - i] = plaintext[64 - initial_permutation[i]];
    }

    bitset<32> left = permuted_text.to_ullong() >> 32;
    bitset<32> right = permuted_text.to_ullong();

    bitset<48> round_key = bitset<48>(key.to_string().substr(0, 48));
```



```
    for (int i = 0; i < 2; i++)
    {
        bitset<32> new_right = left ^ round_function(right, round_key);
        left = right;
        right = new_right;
    }

    bitset<64> combined((left.to_ullong() << 32) | right.to_ullong());
    bitset<64> ciphertext;
    for (int i = 0; i < 64; i++)
    {
        ciphertext[63 - i] = combined[64 - final_permutation[i]];
    }

    return ciphertext;
}

int main()
{
    bitset<64>
plaintext(string("0000000100100011010001010110011110001001101010111100110
111101111"));
    bitset<64>
key(string("0001001100110100010101110111100110011011101111001101111111110
001"));

    bitset<64> ciphertext = DES_encrypt(plaintext, key);

    cout << "Ciphertext: " << ciphertext << endl;
    return 0;
}
```