# NETWORK SECURITY ASSIGNMENT -2 REPORT

-Sreekar Reddy(2021318)
-Satwik Rampelli(2021276)

We have written a code that implements the DES encryption and decryption algorithm.

```
string hex2bin(string s)
```

This function converts a hexadecimal string “s” into its binary equivalent. It uses an unordered map to map each hexadecimal digit to its corresponding binary representation. It iterates over each character in the string “s” and appends the binary representation of characters in the resulting string.

```
string bin2hex(string s)
```

This function does the opposite of hex2bin(string s), i.e it takes binary string “s” as input and converts it to its corresponding hexadecimal representation. It also uses an unordered map to map each 4bit substring to a hexadecimal digit.It iterates over the input string and extracts each 4bit substring from the input string and maps it to the corresponding hexadecimal digit and appends all the resulting hexadecimal digits in to single hexadecimal string.

```
int bin2dec(string binary)
```

This function converts the binary string into its corresponding decimal value. It iterates over the binary string in the input binary string, starting from the least significant bit. It calculates the bit’s decimal value by multiplying the bit with its corresponding power of 2.

```
string dec2bin(int num)
```

This function converts the decimal number into its binary representation. It uses a bitset library to directly convert the decimal number into a binary string of a fixed size of 4.

```
string permute(string k, vector<int> arr, int n)
```

This function is used to rearrange the bits of the input string according to the permutation specified in the vector. It iterators over each indice in the vector and generates a new string by selecting the bits from the input string at those indices.

```
string shift_left(string k, int nth_shifts)
```

This function does the left circular shift on the input string by nth_shifts. It uses rotate function from the library algorithm and shifts the bits towards left by one position x nth_shifts times.

```
string xor_strings(string a, string b)
```

The function performs a bitwise xor operation on the strings “a” and “b”. It iterates over each bit in both the strings and performa xor operation on the corresponding bits  and appends the result to a string and gives it as an output.

```
vector<string> encrypt(string pt, vector<string>& rkb, string type)
```

The function performs the encryption using the DES algorithm. It takes plain text , vector of keys for each and a type to indicate whether its encryption or decryption . In this function , first initial permutation is applied to the plaintext according to the initial permutation table  and then it is split into 2 halves (left and right : 32 bits each).

For each of the 16 rounds , the right half is expanded  from 32 bits to 48 bits using the expansion permutation and  xor operation is performed between the expanded 48 bit right  half and 48 bit round key and then substitution is done using the s-boxes. Then permutation is applied and then right and left halves are swaped.

Final permutation is applied to the combined left and right halves at the end to produce the cipher text.

```
void des(vector<string>& pt, vector<string>& ck)
```

The function performs the DES encyption and decryption for the plaint text and its corresponding cipher key pairs. It iterates over each pair and converts the cipher key from hexadecimal to bianry and applies parity bit drop table , then generating the round keys.

It encrypts the plaint text by using the generated keys for each round and stores the result of each round. It also decrypts the cipher texts using the reversed keys for each round and verifies the results with the stored results.

```
int main()
```

The function takes the user input for the plaintext and cipher key pairs ; checks whether the given pair has 16 hexadecimal characters. Then it calls des() function to perform DES encryption and decryption.