Student Name: Pragati Shankar	USN: 1SI19CS090	Batch No: B2	Date: 27-12-2022

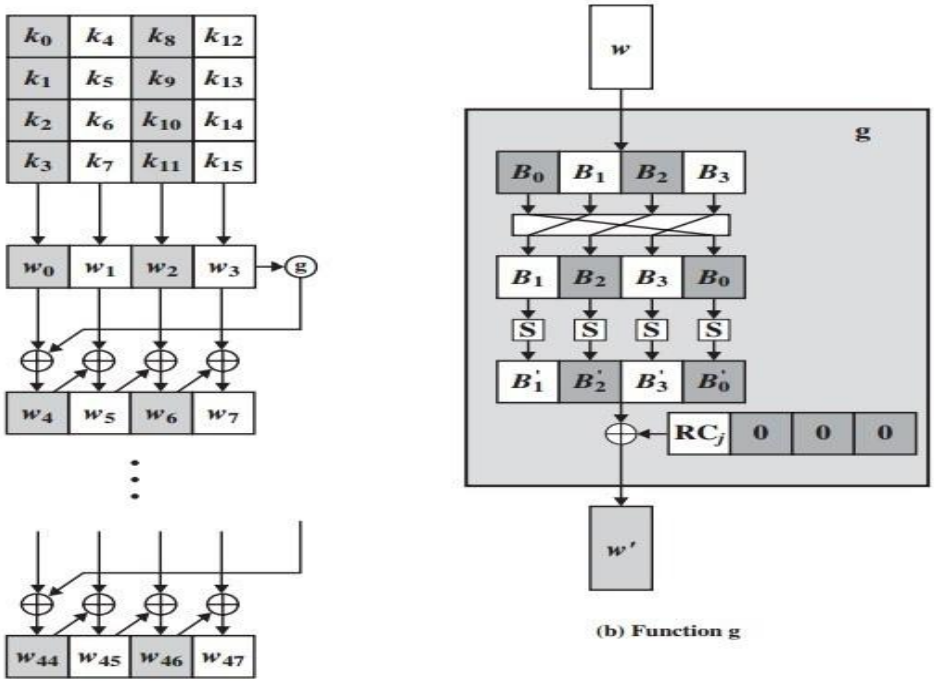
Evaluation:

Write Up (10 marks)	Clarity in concepts (10 marks)	Implementation and execution of the algorithms (10 marks)	Viva (05 marks)	Total (35 marks)

Sl.No	Name of the Faculty In-Charge	Signature
1.	H K Vedamurthy	
2.	Gururaj S P	

Question No: 7. Consider the 128 bits initial key and expand it to 10 different keys each of size 128 bits using AES key expansion technique.

Algorithm:



(a) Overall algorithm

(b) Function g

Figure 1.1 AES Key Expansion

CODE:-

```
#include <bits/stdc++.h>
using namespace std;
unsigned long long sbox[16][16] = {
	{0x63, 0x7c, 0x77, 0x7b, 0xf2, 0x6b, 0x6f, 0xc5, 0x30, 0x01, 0x67, 0x2b, 0xfe, 0xd7, 0xab, 0x76},
	{0xca, 0x82, 0xc9, 0x7d, 0xfa, 0x59, 0x47, 0xf0, 0xad, 0xd4, 0xa2, 0xaf, 0x9c, 0xa4, 0x72, 0xc0},
	{0xb7, 0xfd, 0x93, 0x26, 0x36, 0x3f, 0xf7, 0xcc, 0x34, 0xa5, 0xe5, 0xf1, 0x71, 0xd8, 0x31, 0x15},
	{0x04, 0xc7, 0x23, 0xc3, 0x18, 0x96, 0x05, 0x9a, 0x07, 0x12, 0x80, 0xe2, 0xeb, 0x27, 0xb2, 0x75},
	{0x09, 0x83, 0x2c, 0x1a, 0x1b, 0x6e, 0x5a, 0xa0, 0x52, 0x3b, 0xd6, 0xb3, 0x29, 0xe3, 0x2f, 0x84},
	{0x53, 0xd1, 0x00, 0xed, 0x20, 0xfc, 0xb1, 0x5b, 0x6a, 0xcb, 0xbe, 0x39, 0x4a, 0x4c, 0x58, 0xcf},
	{0xd0, 0xef, 0xaa, 0xfb, 0x43, 0x4d, 0x33, 0x85, 0x45, 0xf9, 0x02, 0x7f, 0x50, 0x3c, 0x9f, 0xa8},
	{0x51, 0xa3, 0x40, 0x8f, 0x92, 0x9d, 0x38, 0xf5, 0xbc, 0xb6, 0xda, 0x21, 0x10, 0xff, 0xf3, 0xd2},
	{0xcd, 0x0c, 0x13, 0xec, 0x5f, 0x97, 0x44, 0x17, 0xc4, 0xa7, 0x7e, 0x3d, 0x64, 0x5d, 0x19, 0x73},
	{0x60, 0x81, 0x4f, 0xdc, 0x22, 0x2a, 0x90, 0x88, 0x46, 0xee, 0xb8, 0x14, 0xde, 0x5e, 0x0b, 0xdb},
	{0xe0, 0x32, 0x3a, 0x0a, 0x49, 0x06, 0x24, 0x5c, 0xc2, 0xd3, 0xac, 0x62, 0x91, 0x95, 0xe4, 0x79},
	{0xe7, 0xc8, 0x37, 0x6d, 0x8d, 0xd5, 0x4e, 0xa9, 0x6c, 0x56, 0xf4, 0xea, 0x65, 0x7a, 0xae, 0x08},
	{0xba, 0x78, 0x25, 0x2e, 0x1c, 0xa6, 0xb4, 0xc6, 0xe8, 0xdd, 0x74, 0x1f, 0x4b, 0xbd, 0x8b, 0x8a},
	{0x70, 0x3e, 0xb5, 0x66, 0x48, 0x03, 0xf6, 0x0e, 0x61, 0x35, 0x57, 0xb9, 0x86, 0xc1, 0x1d, 0x9e},
	{0xe1, 0xf8, 0x98, 0x11, 0x69, 0xd9, 0x8e, 0x94, 0x9b, 0x1e, 0x87, 0xe9, 0xce, 0x55, 0x28, 0xdf},
	{0x8c, 0xa1, 0x89, 0x0d, 0xbf, 0xe6, 0x42, 0x68, 0x41, 0x99, 0x2d, 0x0f, 0xb0, 0x54, 0xbb, 0x16}
};
unsigned long long Rcon[10] = {
	0x01000000, 0x02000000, 0x04000000, 0x08000000, 0x10000000, 0x20000000, 0x40000000, 0x80000000,
0x1b000000, 0x36000000
};
string w[44];

string rotLeft(string word)
{
	return word.substr(8) + word.substr(0,8);
}
string SBoxFun(string word)
{
	string res = "";
	for(int i=0; i<4; i++){
		string byte = word.substr(i*8, 8);
		int row = bitset<4>( byte.substr(0,4) ).to_ulong();
		int col = bitset<4>( byte.substr(4,4) ).to_ulong();
		res += bitset<8>(sbox[row][col]).to_string();
	}
	return res;
}
string XOR(string x, string y){
	string res = "";
	for(int i=0; i<x.length(); i++)
	{
		res += (x[i] == y[i]) ? "0" : "1";
	}
	return res;
```

```
}
int main()
{
	unsigned long long hexkey1, hexkey2;
	cout << "\nEnter first 64-bit key in hexadecimal(16-digits) : " ;
	cin >> hex >> hexkey1;
	cout << "\nEnter next 64-bit key in hexadecimal(16-digits) : " ;
	cin >> hex >> hexkey2;
	string key = bitset<64>(hexkey1).to_string() + bitset<64>(hexkey2).to_string();
	cout << "Binary key (k) \t: " << key << endl;
	cout << "keyLen : " << key.length() << endl;
	for(int i=0; i<4; i++){
		w[i] = key.substr(i*32,32);
	}
	for(int i=4; i<44; i++)
	{
		string first = w[i-4];
		string second = w[i-1];
		if(i % 4 == 0)
		{
			second = rotLeft(second);
			second = SBoxFun(second);
			string tmp = bitset<32>(Rcon[i/4]).to_string();
			second = XOR(second, tmp);
		}
		w[i] = XOR(first, second);
	}
	string keys[11] = {""};
	for(int i=0; i<44; i++)
	{
		keys[i/4] += w[i];
	}
	for(int i=0; i<11; i++)
	{
		cout << "Key " << i << ": ";
		for(int j=0; j<16;j++)
		{
			cout << setfill('0') << setw(2)<<hex<<
			bitset<8>(keys[i].substr(j*8,8)).to_ulong() <<" ";
		}
		cout <<endl;
	}
	return 0;
}
```

Output Screenshot:

```
user@linux-OptiPlex-5090: ~/1SI19CS090
user@linux-OptiPlex-5090:~/1SI19CS090$ gedit aes1.cpp
user@linux-OptiPlex-5090:~/1SI19CS090$ g++ aes1.cpp
user@linux-OptiPlex-5090:~/1SI19CS090$ ./a.out

Enter first 64-bit key in hexadecimal(16-digits) : cc00ccfff0aaf0aa

Enter next 64-bit key in hexadecimal(16-digits) : 1fe22472901bb2a3
Binary key (k)  : 1100110000000000110011001111111111110000101010101111000010101010
1000011111111000100010010001110010100100000001101110110010101000 11
keyLen : 128
Key 0: cc 00 cc ff f0 aa f0 aa 1f e2 24 72 90 1b b2 a3
Key 1: 61 37 c6 9f 91 9d 36 35 8e 7f 12 47 1e 64 a0 e4
Key 2: 26 d7 af ed b7 4a 99 d8 39 35 8b 9f 27 51 2b 7b
Key 3: ff 26 8e 21 48 6c 17 f9 71 59 9c 66 56 08 b7 1d
Key 4: df 8f 2a 90 97 e3 3d 69 e6 ba a1 0f b0 b2 16 12
Key 5: c8 c8 e3 77 5f 2b de 1e b9 91 7f 11 09 23 69 03
Key 6: ae 31 98 76 f1 1a 46 68 48 8b 39 79 41 a8 50 7a
Key 7: ec 62 42 f5 1d 78 04 9d 55 f3 3d e4 14 5b 6d 9e
Key 8: ce 5e 49 0f d3 26 4d 92 86 d5 70 76 92 8e 1d e8
Key 9: e1 fa d2 40 32 dc 9f d2 b4 09 ef a4 26 87 f2 4c
Key a: 10 5a 32 b7 22 86 ad 65 96 8f 42 c1 b0 08 b0 8d
user@linux-OptiPlex-5090:~/1SI19CS090$
```