|  |
| --- |
| **C++ console-based application** |

**Question**

Understand the 3D file and implement a C++ console-based application to perform the Advanced Encryption Standard (AES) algorithm in C++ to encrypt and decrypt the given 3D file (.OBJ). The encryption and decryption process needs to be performed with multi-threading process using C++.

**Explanation**

AES (Advanced Encryption Standard), also known as Rijndael encryption method in cryptography, is a block encryption standard adopted by the federal government of the United States. This standard is used to replace the original DES, which has been widely used all over the world and has become one of the most popular symmetric key algorithms

This test code for AES encryption and decryption of a picture:

**Code**

#include <fstream>

typedef bitset<8> byte;

typedef bitset<32> word;

/\*\*

\* Converting an array of char characters into binary

\* Save it in a byte array

\*/

void charToByte(byte out[16], const char s[16])

{

for(int i=0; i<16; ++i)

for(int j=0; j<8; ++j)

out[i][j]= ((s[i]>>j) & 1);

}

/\*\*

\* Divide consecutive 128 bits into 16 groups and store them in a byte array

\*/

void divideToByte(byte out[16], bitset<128>& data)

{

bitset<128> temp;

for(int i=0; i<16; ++i)

{

temp = (data << 8\*i) >> 120;

out[i] = temp.to\_ulong();

}

}

/\*\*

\* Merge 16 byte s into 128 consecutive bits

\*/

bitset<128> mergeByte(byte in[16])

{

bitset<128> res;

res.reset(); //Set 0

bitset<128> temp;

for(int i=0; i<16; ++i)

{

temp = in[i].to\_ulong();

temp <<= 8\*(15-i);

res |= temp;

}

return res;

}

int main()

{

string keyStr = "abcdefghijklmnop";

byte key[16];

charToByte(key, keyStr.c\_str());

//Key expansion

word w[4\*(Nr+1)];

KeyExpansion(key, w);

bitset<128> data;

byte plain[16];

//Encrypt the file flower.jpg into cipher.txt

ifstream in;

ofstream out;

in.open("D://flower.jpg", ios::binary);

out.open("D://cipher.txt", ios::binary);

while(in.read((char\*)&data, sizeof(data)))

{

divideToByte(plain, data);

encrypt(plain, w);

data = mergeByte(plain);

out.write((char\*)&data, sizeof(data));

data.reset(); //Set 0

}

in.close();

out.close();

//Decrypt cipher.txt and write the picture flower1.jpg

in.open("D://cipher.txt", ios::binary);

out.open("D://flower1.jpg", ios::binary);

while(in.read((char\*)&data, sizeof(data))){

divideToByte(plain, data);

decrypt(plain, w);

data = mergeByte(plain);

out.write((char\*)&data, sizeof(data));

data.reset(); }

in.close();

out.close();

return 0;

}