Implementing one time pad (vernam cipher) in C++

```cpp
#include<iostream>

#include<string>
using namespace std;


long mod(int a, int b)
{
  return (a % b + b) % b;
}

string encrypt(string key, string m)
{
    string result = "";

    // traverse text
    for (int i=0;i<m.length();i++)
    {
        // apply transformation to each character
        result += char(mod(int(m[i]-65+key[i]-65), 26) +65);
    }

    // Return the resulting string
    return result;
}

string decrypt(string key, string m)
{
    string result = "";

    // traverse text
    for (int i=0;i<m.length();i++)
    {
      result += char(mod(int((m[i]-65)-(key[i]-65)), 26) +65);
    }

    // Return the resulting string
    return result;
}

int main(){
    string m;
    cout<<"Enter the message"<<'\n';
    cin>>m;
    string key;
    cout<<"Enter the key"<<'\n';
    cin>>key;
    string cipher = encrypt(key, m);
    cout<<"Encrypted message: "<<cipher<<'\n';

    cout<<"Decrypted message: "<<decrypt(key, cipher)<<'\n';
     system(“pause”);

    return 0;
}
```

Output

```
Enter the message Hello

Enter the key 22 25 10 9 5

Encrypted message: IEIJQ
Decrypted message: HELLO
```

OUTPUT

```
Enter the message WAFAA

Enter the key 22 25 10 9 5


Encrypted message: HLSSM
Decrypted message: WAFAA
```