```cpp
#include "LFSR.hpp"
#include <SFML/System.hpp>
#include <SFML/Window.hpp>
#include <SFML/Graphics.hpp>
#include <iostream>
#include <string>
#include <bitset>
#include <cmath>

int encrpyt(int value, int tap);

int main(int argc, char *argv[])
{
        sf::Image image;
        if(!image.loadFromFile(argv[1])){
                return -1;
        }
        sf::Image after = image;
        LFSR lfsr(argv[3], atoi(argv[4]));
        sf::Color p;

        sf::Vector2u size = image.getSize();
        sf::RenderWindow window1(sf::VideoMode(size.x, size.y), "Original");
        sf::RenderWindow window2(sf::VideoMode(size.x, size.y), "Encrypted");

        int X = size.x;
        int Y = size.y;

        for (int x = 0; x < X; x++)
        {
                for (int y = 0; y < Y; y++)
                {
                        p = after.getPixel(x,y);
                        p.r = encrpyt(p.r, lfsr.generate(8));
                        p.g = encrpyt(p.g, lfsr.generate(8));
                        p.b = encrpyt(p.b, lfsr.generate(8));
                        after.setPixel(x, y, p);
                }
        }

        sf::Texture texture;
        texture.loadFromImage(image);

        sf::Texture texture2;
        texture2.loadFromImage(after);

        sf::Sprite sprite;
        sprite.setTexture(texture);

        sf::Sprite sprite2;
        sprite2.setTexture(texture2);

        while(window1.isOpen() && window2.isOpen())
        {
                sf::Event event;
                while(window1.pollEvent(event))
                {
                        if (event.type == sf::Event::Closed)
                        {
                                window1.close();
                        }
                }
                while(window2.pollEvent(event)){
                        if (event.type == sf::Event::Closed)
                        {
                                window2.close();
                        }
                }
                window1.clear(sf::Color::White);
                window1.draw(sprite);
                window1.display();
                window2.clear(sf::Color::White);
                window2.draw(sprite2);
                window2.display();
        }

        if (!after.saveToFile(argv[2]))
        {
                return -1;
        }
        return 0;

}

int encrpyt(int value, int tap)
{
        std::bitset<8>V(value);
        std::bitset<8>T(tap);
        std::string Value = V.to_string();
        std::string Tap = T.to_string();
        int j = 7;
        int x = 0;
        for (int i = 0; i < 8; i++)
        {
                if (Value[i] != Tap[i])
                {
                        x = x+pow(2,j);
                }
                j--;
        }
        return x;
}
```