**DECRYPTION CODE**

**(Because of different format code may appear different)**

#include <iostream>

#include <fstream>

#include <string>

#include <vector>

const std::string &SALT1 = "LM::TB::BB";

const std::string &SALT2 = "\_:/\_77";

const std::string &SALT3 = "line=boostedC++";

const std::string &BASE64\_CODES = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/";

std::string DecryptB64(std::string s);

std::string base64\_decode(const std::string &s);

int main(int argc, char \*argv[])

{

if (argc != 3)

return std::cout << "Program needs TWO arguments, input and output!" << std::endl, 2;

std::string in(argv[1]), out(argv[2]);

std::ifstream fi(in);

if (!fi)

return std::cout << "Cannot read input file '" << in << "'" << std::endl, 3;

std::string data;

fi >> data; // input into data

if (!fi)

return std::cout << "Input file '" << in << "' corrupted!" << std::endl, 4;

data = DecryptB64(data);

std::ofstream fo(out);

if (!fo)

return std::cout << "Cannot write to output file '" << out << "'" << std::endl, 5;

fo << data;

std::cout << "Decoding was successful" << std::endl;

return 0;

}

std::string base64\_decode(const std::string &s)

{

std::string ret;

std::vector<int> vec(256, -1);

for (int i = 0; i < 64; i++)

vec[BASE64\_CODES[i]] = i;

int val = 0, bits = -8;

for (const auto &c : s)

{

if (vec[c] == -1) break;

val = (val << 6) + vec[c];

bits += 6;

if (bits >= 0)

{

ret.push\_back(char((val >> bits) & 0xFF));

bits -= 8;

}

}

return ret;

}

std::string DecryptB64(std::string s)

{

s = s.erase(7, 1);

s = s.erase(1, 2);

s = base64\_decode(s);

s = s.substr(SALT2.length() + SALT3.length());

s = s.substr(0, s.length() - SALT1.length());

s = base64\_decode(s);

s = s.substr(0, s.length() - SALT1.length());

s = s.erase(7, SALT3.length());

s = base64\_decode(s);

s = s.substr(SALT1.length());

s = s.substr(0, s.length() - SALT2.length() - SALT3.length());

return s;

}