Task 10 Report

In this task we use the library called crypto to gather the necessary tools to Encrypt our password with the DES algorithm. We apply the algorithm 25 times and add ed salt to make random numbers generated to ensure every encryption is unique and secure. The program then checks if a given password matches the encrypted one and generates 10 sample encrypted passwords for demonstration.

A screen shot of a computer program

AI-generated content may be incorrect.

#include <cryptopp/des.h>

#include <cryptopp/modes.h>

#include <cryptopp/filters.h>

#include <cryptopp/osrng.h>

#include <cryptopp/base64.h>

#include <iostream>

#include <string>

using namespace CryptoPP;

using namespace std;

string generateSalt()

{

    AutoSeededRandomPool rng;

    byte salt[2];

    rng.GenerateBlock(salt, sizeof(salt));

    return string((char \*)salt, sizeof(salt));

}

string encryptPassword(const string &password, const string &salt, const SecByteBlock &key)

{

    string encrypted = password + salt;

    for (int i = 0; i < 25; ++i)

    {

        ECB\_Mode<DES>::Encryption desEncryptor;

        desEncryptor.SetKey(key, key.size());

        string temp;

        StringSource ss(encrypted, true,

                        new StreamTransformationFilter(desEncryptor, new StringSink(temp)));

        encrypted = temp;

    }

    string encoded;

    StringSource ss(encrypted, true, new Base64Encoder(new StringSink(encoded)));

    return encoded;

}

int main()

{

    SecByteBlock key(DES::DEFAULT\_KEYLENGTH);

    AutoSeededRandomPool rng;

    rng.GenerateBlock(key, key.size());

    for (int i = 0; i < 10; ++i)

    {

        string password = "Password" + to\_string(i + 1);

        string salt = generateSalt();

        string encryptedPassword = encryptPassword(password, salt, key);

        cout << "User " << (i + 1) << ":\n";

        cout << "  Original Password: " << password << endl;

        cout << "  Salt (hex): ";

        for (unsigned char ch : salt)

        {

            printf("%02X", ch);

        }

        cout << endl;

        cout << "Encrypted Password (Base64): " << encryptedPassword << endl

             << endl;

    }

    return 0;

}