Practical 13

Problem statement : Write a program to implement Digital Signature Algorithms: DSA.

#include <stdio.h>

#include <openssl/dsa.h>

#include <openssl/err.h>

#include <openssl/rand.h>

void handleErrors(void) {

    ERR\_print\_errors\_fp(stderr);

    abort();

}

int main() {

    // Initialize OpenSSL

    OpenSSL\_add\_all\_algorithms();

    ERR\_load\_crypto\_strings();

    // Generate DSA parameters

    DSA \*dsa = DSA\_new();

    if (dsa == NULL) handleErrors();

    if (DSA\_generate\_parameters\_ex(dsa, 1024, NULL, 0, NULL, NULL, NULL) != 1) {

        handleErrors();

    }

    // Generate DSA key pair

    if (DSA\_generate\_key(dsa) != 1) {

        handleErrors();

    }

    // Message to be signed

    unsigned char \*message = (unsigned char \*)"This is a test message";

    unsigned int message\_len = strlen((char \*)message);

    // Sign the message

    unsigned char \*signature = (unsigned char \*)malloc(DSA\_size(dsa));

    unsigned int sig\_len;

    if (DSA\_sign(0, message, message\_len, signature, &sig\_len, dsa) != 1) {

        handleErrors();

    }

    printf("Message signed successfully.\n");

    // Verify the signature

    int verify\_status = DSA\_verify(0, message, message\_len, signature, sig\_len, dsa);

    if (verify\_status == 1) {

        printf("Signature verified successfully.\n");

    } else if (verify\_status == 0) {

        printf("Signature verification failed.\n");

    } else {

        handleErrors();

    }

    // Clean up

    DSA\_free(dsa);

    free(signature);

    EVP\_cleanup();

    ERR\_free\_strings();

    return 0;

}