Program 32 digital signature

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <openssl/dsa.h>

#include <openssl/sha.h>

void generate\_dsa\_signature(const char \*message, DSA \*dsa\_key) {

// Hash the message using SHA-256

unsigned char hash[SHA256\_DIGEST\_LENGTH];

SHA256((const unsigned char \*)message, strlen(message), hash);

// Generate random value k for each signature

BIGNUM \*k = BN\_new();

BN\_rand\_range(k, dsa\_key->q);

// Calculate r = (g^k mod p) mod q

BIGNUM \*r = BN\_new();

BN\_mod\_exp(r, dsa\_key->g, k, dsa\_key->p, dsa\_key->ctx);

BN\_mod(r, r, dsa\_key->q, dsa\_key->ctx);

// Calculate s = (k^-1 \* (H(m) + x\*r)) mod q

BIGNUM \*s = BN\_new();

BIGNUM \*x = dsa\_key->priv\_key;

BIGNUM \*h = BN\_bin2bn(hash, SHA256\_DIGEST\_LENGTH, NULL);

BIGNUM \*temp1 = BN\_new();

BIGNUM \*temp2 = BN\_new();

BN\_mod\_mul(temp1, x, r, dsa\_key->q, dsa\_key->ctx);

BN\_mod\_add(temp2, h, temp1, dsa\_key->q, dsa\_key->ctx);

BN\_mod\_inverse(temp1, k, dsa\_key->q, dsa\_key->ctx);

BN\_mod\_mul(s, temp1, temp2, dsa\_key->q, dsa\_key->ctx);

// Print the generated signature

printf("Generated DSA Signature:\n");

printf("r: %s\n", BN\_bn2hex(r));

printf("s: %s\n", BN\_bn2hex(s));

// Free memory

BN\_free(k);

BN\_free(r);

BN\_free(s);

BN\_free(h);

BN\_free(temp1);

BN\_free(temp2);

}

int main() {

// Initialize DSA parameters

DSA \*dsa\_key = DSA\_generate\_parameters(1024, NULL, 0, NULL, NULL, NULL, NULL);

DSA\_generate\_key(dsa\_key);

// Message to be signed

const char \*message = "Hello, DSA!";

// Generate DSA signature for the message

generate\_dsa\_signature(message, dsa\_key);

// Free DSA key

DSA\_free(dsa\_key);

return 0;

}

Output:

