1. To implement signature scheme named digital signature standard (Euclidean Algorithm) using C language

**Program:**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <stdbool.h>

typedef struct {

unsigned int p, q, g, x, y;

} DSA\_KEY;

unsigned int euclidean\_algorithm(unsigned int a, unsigned int b) {

int t;

while (b != 0) {

t = b;

b = a % b;

a = t;

}

return a;

}

unsigned int modular\_inverse(unsigned int a, unsigned int b) {

int x = 0, y = 1, u = 1, v = 0, gcd = b, m, n, q, r;

while (a != 0) {

q = gcd / a;

r = gcd % a;

m = x - u \* q;

n = y - v \* q;

gcd = a;

a = r;

x = u;

y = v;

u = m;

v = n;

}

return y < 0 ? y + b : y;

}

void dsa\_sign(unsigned int \*r, unsigned int \*s, unsigned int h, unsigned int k, DSA\_KEY key) {

\*r = (key.g % key.p) % key.q;

\*s = (modular\_inverse(k, key.q) \* (h + key.x \* \*r)) % key.q;

}

bool dsa\_verify(unsigned int r, unsigned int s, unsigned int h, DSA\_KEY key) {

if (r < 1 || r > key.q - 1 || s < 1 || s > key.q - 1) {

return false;

}

unsigned int w = modular\_inverse(s, key.q);

unsigned int u1 = (h \* w) % key.q;

unsigned int u2 = (r \* w) % key.q;

unsigned int v = ((key.g % key.p) % key.q \* u1 + (key.y % key.p) % key.q \* u2) % key.q;

return v == r;

}

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

DSA\_KEY key = { 61, 53, 8, 31, 33 };

unsigned int h = 1234, k = 5678, r, s;

dsa\_sign(&r, &s, h, k, key);

printf("Signature: (r = %u, s = %u)\n", r, s);

if (dsa\_verify(r, s, h, key)) {

printf("Signature verified\n");

} else {

printf("Signature verification failed\n");

}

return 0;

}

