***SAVEETHA SCHOOL OF ENGINEERING***

***SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCE***

**EXP NO 9 Demonstrate Diffie-Hellman Key Exchange algorithm.**

**AIM**

To Demonstrate Diffie-Hellman Key Exchange algorithm in C language.

**PROCEDURE**

* Generates Diffie-Hellman parameters, including a prime number and a generator.
* Generates public and private keys for both parties.
* Computes the shared secret key using the other party's public key and its own private key.
* Prints the shared secret key.

**PROGRAM**

#include <stdio.h>

#include <stdint.h>

// Function to perform modular exponentiation

uint64\_t mod\_exp(uint64\_t base, uint64\_t exp, uint64\_t mod) {

uint64\_t result = 1;

base = base % mod;

while (exp > 0) {

if (exp % 2 == 1)

result = (result \* base) % mod;

exp = exp >> 1;

base = (base \* base) % mod;

}

return result;

}

// Function to perform Diffie-Hellman key exchange

void diffie\_hellman(uint64\_t p, uint64\_t g, uint64\_t a, uint64\_t b) {

uint64\_t A = mod\_exp(g, a, p); // Alice's public key

uint64\_t B = mod\_exp(g, b, p); // Bob's public key

uint64\_t shared\_secret\_A = mod\_exp(B, a, p); // Shared secret computed by Alice

uint64\_t shared\_secret\_B = mod\_exp(A, b, p); // Shared secret computed by Bob

printf("Alice's public key (A): %llu\n", A);

printf("Bob's public key (B): %llu\n", B);

printf("Shared secret computed by Alice: %llu\n", shared\_secret\_A);

printf("Shared secret computed by Bob: %llu\n", shared\_secret\_B);

}

int main() {

// Public parameters (prime number and generator)

uint64\_t p = 23; // Prime number

uint64\_t g = 5; // Generator

// Private keys for Alice and Bob

uint64\_t a = 6; // Alice's private key

uint64\_t b = 15; // Bob's private key

// Perform Diffie-Hellman key exchange

diffie\_hellman(p, g, a, b);

return 0;

}

**OUTPUT**

