CSA5122-CRYPTOGRAPHY FOR NETWORK AND SECURITY

LAB PROGRAMS EXECUTION

12.DIFFIE HELLMAN ALGORITHM

# Simple Diffie-Hellman Key Exchange in Python

def power(base, exponent, mod):

result = 1

base %= mod

while exponent > 0:

if exponent % 2 == 1:

result = (result \* base) % mod

exponent = exponent // 2

base = (base \* base) % mod

return result

# Publicly known variables

p = 23 # A prime number

g = 5 # A primitive root modulo p

# Alice's private key

a = 6

# Bob's private key

b = 15

# Alice computes public value

A = power(g, a, p)

# Bob computes public value

B = power(g, b, p)

# Exchange A and B

# Alice computes shared secret

shared\_secret\_alice = power(B, a, p)

# Bob computes shared secret

shared\_secret\_bob = power(A, b, p)

print(f"Public parameters: p = {p}, g = {g}")

print(f"Alice's public key: {A}")

print(f"Bob's public key: {B}")

print(f"Alice's shared secret: {shared\_secret\_alice}")

print(f"Bob's shared secret: {shared\_secret\_bob}")

