CSA5122-CRYPTOGRAPHY FOR NETWORK AND SECURITY

LAB PROGRAMS EXECUTION

11.ECC ALGORITHM

# Elliptic Curve Cryptography - ECDH Key Exchange over small finite field

class ECC:

def \_\_init\_\_(self, a, b, p, G, n):

self.a = a # curve coefficient a

self.b = b # curve coefficient b

self.p = p # prime field (mod p)

self.G = G # generator point

self.n = n # order of the base point G

def inverse\_mod(self, k):

# Compute modular inverse using Fermat's little theorem

return pow(k, -1, self.p)

def is\_on\_curve(self, P):

if P is None:

return True

x, y = P

return (y\*\*2) % self.p == (x\*\*3 + self.a \* x + self.b) % self.p

def point\_add(self, P, Q):

if P is None: return Q

if Q is None: return P

x1, y1 = P

x2, y2 = Q

if x1 == x2 and y1 != y2:

return None # P + (-P) = 0

if P == Q:

# Point doubling

m = ((3 \* x1\*\*2 + self.a) \* self.inverse\_mod(2 \* y1)) % self.p

else:

# Point addition

m = ((y2 - y1) \* self.inverse\_mod(x2 - x1)) % self.p

x3 = (m\*\*2 - x1 - x2) % self.p

y3 = (m \* (x1 - x3) - y1) % self.p

return (x3, y3)

def scalar\_mult(self, k, P):

result = None

addend = P

while k:

if k & 1:

result = self.point\_add(result, addend)

addend = self.point\_add(addend, addend)

k >>= 1

return result

# Define elliptic curve: y^2 = x^3 + ax + b over F\_p

a = 2

b = 2

p = 17 # prime modulus

G = (5, 1) # base point on curve

n = 19 # assume the order of G is 19

# Initialize curve

curve = ECC(a, b, p, G, n)

# --- Alice's key pair ---

a\_priv = 7 # Alice's private key

A\_pub = curve.scalar\_mult(a\_priv, G) # Alice's public key

# --- Bob's key pair ---

b\_priv = 11 # Bob's private key

B\_pub = curve.scalar\_mult(b\_priv, G) # Bob's public key

# --- Shared Secret ---

shared\_A = curve.scalar\_mult(a\_priv, B\_pub)

shared\_B = curve.scalar\_mult(b\_priv, A\_pub)

# --- Output ---

print("Alice's Public Key: ", A\_pub)

print("Bob's Public Key: ", B\_pub)

print("Shared Secret (A): ", shared\_A)

print("Shared Secret (B): ", shared\_B)

print("Match:", shared\_A == shared\_B)

