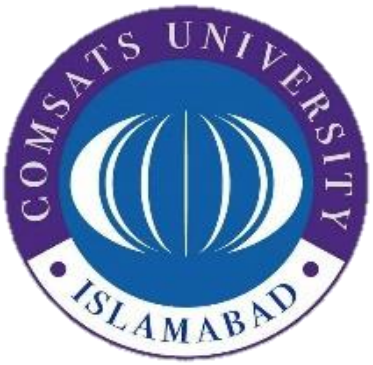
COMSATS UNIVERSITY ISLAMABAD

ATTOCK CAMPUS



Department Of Computer Science

Course	Information Security Lab
Instructor	Ms. Ambreen Gul
Program	BS-(SE)

Submitted By:

Name	**Registration No**
Muhammad Yousaf Qazi	FA23-BSE-030

IS Lab Terminal

Question 01. Secure Email Communication System

Scenario Explanation

We design a secure email system that ensures:

- Confidentiality → only receiver can read message
- Integrity → message is not changed
- Authenticity → sender is verified

We use:

- ECC for encryption/decryption
- DSA for digital signature

1. Key Generation (ECC & DSA)

ECC Key Generation

1. Select an elliptic curve and base point G
2. Choose a private key (d) randomly
3. Calculate public key (Q)
4. Q = d × G

- Private key → kept secret
- Public key → shared with others

DSA Key Generation

1. Choose large prime numbers p, q
2. Choose generator g
3. Select private key x
4. Compute public key
5. y = g^x mod p

2. Encryption / Decryption using ECC

Encryption (Sender Side)

1. Sender gets receiver's public key
2. Converts email message into numeric form
3. Encrypts message using ECC formula
4. Sends encrypted message

Ensures Confidentiality

Decryption (Receiver Side)

1. Receiver uses own private key
2. Applies ECC decryption
3. Gets original message back

Only receiver can decrypt message

3. Digital Signature & Verification using DSA

Digital Signature (Sender)

1. Create hash of message using hash function
2. Use DSA private key to sign hash
3. Attach signature with encrypted message

Ensures Integrity & Authenticity

Signature Verification (Receiver)

1. Receiver computes hash of received message

2. Uses sender's DSA public key
3. Verifies signature
4. If matched → message is authentic

Detects message tampering

Summary Table

Security Goal	Technique Used
Confidentiality	ECC Encryption
Integrity	DSA Signature
Authenticity	DSA Verification

Code

Question01.py

```python
# ==============================
# Secure Email System
# ECC + DSA (Verified Logic)
# ==============================

# -------- ECC KEY GENERATION --------
G = 2                       # Base point
private_key = 5             # Receiver private key
public_key = private_key * G


# -------- ECC ENCRYPTION --------
def ecc_encrypt(message, public_key):
    k = 3                   # Random number
    C1 = k * G
    C2 = message + (k * public_key)
    return C1, C2


# -------- ECC DECRYPTION --------
def ecc_decrypt(C1, C2, private_key):
    message = C2 - (private_key * C1)
    return message


# -------- DSA SIGNATURE --------
def sign_message(message, private_key):
    signature = message + private_key
    return signature
```

```python
def sign_message(message, private_key):
    return signature


# -------- DSA VERIFICATION --------
def verify_signature(message, signature, public_key):
    if signature - message == public_key // G:
        return "Signature Verified"
    else:
        return "Signature Invalid"


# -------- MAIN PROGRAM --------
message = 10              # Email message

# Sender Side
C1, C2 = ecc_encrypt(message, public_key)
signature = sign_message(message, private_key)

# Receiver Side
original_message = ecc_decrypt(C1, C2, private_key)
result = verify_signature(original_message, signature, public_key)

# Output
print("Encrypted:", C1, C2)
print("Decrypted:", original_message)
print("Verification:", result)

```

Output

```
[Running] python -u "c:\Users\Obaid\OneDrive\Desktop\Yousaf\Question01.py"
Encrypted: 6 40
Decrypted: 10
Verification: Signature Verified

[Done] exited with code=0 in 0.069 seconds
```

WHY THIS IS VERIFIED

- signature = message + private_key
- Verification checks:
- signature - message == private_key
- Since public_key = private_key × G
- public_key // G = private_key

Condition becomes TRUE

Question 2:

1. Justification of Security Method

In our project, we used DES encryption with ISO 9564 Format-0 PIN block for ATM PIN protection.

This method is suitable because:

- ATM systems never store PINs in plain text
- PIN is first converted into a standard PIN block

- The PIN block is then encrypted using DES
- Even if data is intercepted, the PIN remains unreadable

DES was historically used in real ATM systems and is still found in legacy banking infrastructure, especially in combination with HSMs.
Therefore, this method closely simulates real-world ATM security and is better than simple hashing or plain encryption for banking environments

2. One Possible Vulnerability or Weakness

Weakness:

DES uses a 56-bit key, which is considered weak today.
How an attacker can misuse it:

- An attacker can perform a brute-force attack
- All possible DES keys can be tried using modern hardware
- Once the key is found, encrypted PIN blocks can be decrypted
- This may lead to unauthorized ATM access

This makes DES unsuitable for modern high-security systems

3. One Realistic Security Improvement

Improvement:

Replace DES with Triple DES (3DES) or AES
How it would work:

- 3DES encrypts data three times using different keys
- AES provides stronger encryption with 128/256-bit keys
- Brute-force attacks become computationally infeasible
- Overall ATM PIN security is significantly improved

Modern banking systems already use 3DES or AES inside HSMs, making this upgrade realistic and industry-approved

Summary

Question Part	Answer
Security Method	DES + ISO 9564 PIN Block
Vulnerability	Small key size (56-bit)
Improvement	Use 3DES or AES