جامعة سيئون
SEIYUN UNIVERSITY

Seiyun University, Yemen

1001001
1010100

جامعة سيئون
كلية الحاسبات
COLLEGE OF COMPUTERS

Subject Name

SPECIAL TOPICS IN

INFORMATION SECURITY

Prepared by :

walaa Abdaruhman

Registration number

22202041073

information security

```
# -*- coding: utf-8 -*-
"""
Seiyun University - Information Security Department
Special Topics in Information Security - Lab Assignment 05
IoT and Cyber-Physical Systems Security

Student: Walaa Abdulrahman
Enrollment: 22202041073
"""

print("=" * 70)
print("🔧 INSTALLING REQUIRED LIBRARIES")
print("=" * 70)

!pip install pycryptodome -q

print("✅ PyCryptodome installed successfully!")
print()
```

```
======================================================================
🔧 INSTALLING REQUIRED LIBRARIES
======================================================================
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.3/2.3 MB 23.8 MB/s eta 0:00:00
✅ PyCryptodome installed successfully!
```

```
print("=" * 70)
print("📦 IMPORTING LIBRARIES")
print("=" * 70)

import random
import datetime
import time
import hashlib
import base64
from Crypto.Cipher import AES
from Crypto.Util.Padding import pad, unpad
from Crypto.Random import get_random_bytes

print("✅ All libraries imported successfully!")
print()

print("👩‍🎓 STUDENT INFORMATION:")
print("-" * 40)
print(f"Name: Walaa Abdulrahman")
print(f"Enrollment: 22202041073")
print(f"Date: {datetime.datetime.now().strftime('%Y-%m-%d')}")
print(f"Course: Special Topics in Information Security")
print(f"Assignment: Lab 05 - IoT and CPS Security")
print(f"Instructor: Prof. Ahmed Abuamer")
print(f"Due Date: 30 November 2025")
print("=" * 70)
print()
```

```
======================================================================
📦 IMPORTING LIBRARIES
======================================================================
✅ All libraries imported successfully!

👩‍🎓 STUDENT INFORMATION:
----------------------------------------
Name: Walaa Abdulrahman
Enrollment: 22202041073
Date: 2025-12-03
Course: Special Topics in Information Security
Assignment: Lab 05 - IoT and CPS Security
Instructor: Prof. Ahmed Abuamer
Due Date: 30 November 2025
======================================================================
```

```
print("=" * 70)
print("🔐 PART I: IoT DEVICE DATA ENCRYPTION - FUNCTIONS")
print("=" * 70)
print()

def generate_sensor_data():
    """Generate random sensor readings for IoT device"""
    temperature = round(random.uniform(22.0, 32.5), 2)
    humidity = round(random.uniform(45.0, 78.5), 2)
    timestamp = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
```

```
        return {
            'temperature': temperature,
            'humidity': humidity,
            'timestamp': timestamp,
            'device_id': 'IoT_Sensor_001'
        }

    def data_to_string(sensor_data):
        """Convert sensor data dictionary to string format"""
        return f"TEMP:{sensor_data['temperature']},HUM:{sensor_data['humidity']},TIME:{sensor_data['timestamp']},ID:{sensor_dat

    class IoTCryptoSystem:
        """AES-based encryption system for IoT devices"""

        def __init__(self):
            self.key = get_random_bytes(16)  # 128-bit key for AES
            print(f"🔑 Encryption Key Generated: {self.key.hex()}")

        def encrypt(self, plaintext):
            """Encrypt data using AES-CBC mode"""
            cipher = AES.new(self.key, AES.MODE_CBC)
            iv = cipher.iv
            plaintext_bytes = plaintext.encode('utf-8')
            ciphertext = cipher.encrypt(pad(plaintext_bytes, AES.block_size))
            return iv + ciphertext

        def decrypt(self, encrypted_data):
            """Decrypt AES-CBC encrypted data"""
            iv = encrypted_data[:16]
            ciphertext = encrypted_data[16:]
            cipher = AES.new(self.key, AES.MODE_CBC, iv)
            plaintext_bytes = unpad(cipher.decrypt(ciphertext), AES.block_size)
            return plaintext_bytes.decode('utf-8')

    print("✅ Part I functions defined successfully!")
    print()
```

```
======================================================================
🔐 PART I: IoT DEVICE DATA ENCRYPTION - FUNCTIONS
======================================================================

✅ Part I functions defined successfully!
```

```
    print("=" * 70)
    print("🚀 PART I: EXECUTING IoT DATA ENCRYPTION SIMULATION")
    print("=" * 70)
    print()

    def simulate_iot_encryption():
        """Complete simulation of IoT device data encryption"""

        print("[STEP 1] GENERATING SENSOR DATA")
        print("-" * 40)

        sensor_data = generate_sensor_data()
        print(f"🌡️ Temperature: {sensor_data['temperature']}°C")
        print(f"💧 Humidity: {sensor_data['humidity']}%")
        print(f"🕒 Timestamp: {sensor_data['timestamp']}")
        print(f"📱 Device ID: {sensor_data['device_id']}")
        print()

        print("[STEP 2] FORMATTING DATA")
        print("-" * 40)

        plaintext = data_to_string(sensor_data)
        print(f"📝 Data String: {plaintext}")
        print()

        print("[STEP 3] INITIALIZING ENCRYPTION SYSTEM")
        print("-" * 40)

        crypto = IoTCryptoSystem()
        print()

        print("[STEP 4] ENCRYPTING DATA")
        print("-" * 40)

        encrypted_data = crypto.encrypt(plaintext)
        print(f"🔒 Encrypted Data (Base64):")
        print(base64.b64encode(encrypted_data).decode()[:80] + "...")
        print(f"📏 Packet Size: {len(encrypted_data)} bytes")
        print()

        print("[STEP 5] SIMULATING TRANSMISSION")
        print("-" * 40)
        print("📡 Transmitting data to cloud server...")
        print("✅ Transmission successful!")
        print()

        print("[STEP 6] DECRYPTING DATA")
        print("-" * 40)

        decrypted_text = crypto.decrypt(encrypted_data)
        print(f"🔒 Decrypted Data: {decrypted_text}")
        print()

        print("[STEP 7] PARSING DECRYPTED DATA")
        print("-" * 40)

        parsed_data = {}
        parts = decrypted_text.split(',')
        for part in parts:
            if ':' in part:
                key, value = part.split(':', 1)
                parsed_data[key] = value

        print("📊 Parsed Data:")
        for key, value in parsed_data.items():
            print(f"  • {key}: {value}")
        print()

        print("[STEP 8] DATA INTEGRITY VERIFICATION")
        print("-" * 40)

        if (float(parsed_data.get('TEMP', 0)) == sensor_data['temperature'] and
            float(parsed_data.get('HUM', 0)) == sensor_data['humidity']):
            print("✅ VERIFICATION SUCCESSFUL - Data integrity confirmed!")
        else:
            print("❌ VERIFICATION FAILED - Data mismatch detected!")

        return sensor_data, encrypted_data, parsed_data

    # Execute simulation
    print("Starting Part I Simulation...")
    print("-" * 50)
    part1_data, part1_encrypted, part1_parsed = simulate_iot_encryption()
    print("✅ PART I COMPLETED SUCCESSFULLY!")
    print()
```

```
======================================================================
🚀 PART I: EXECUTING IoT DATA ENCRYPTION SIMULATION
======================================================================

Starting Part I Simulation...
--------------------------------------------------
[STEP 1] GENERATING SENSOR DATA
----------------------------------------
🌡 Temperature: 25.64°C
💧 Humidity: 54.03%
🕒 Timestamp: 2025-12-03 18:18:43
📱 Device ID: IoT_Sensor_001

[STEP 2] FORMATTING DATA
----------------------------------------
📝 Data String: TEMP:25.64,HUM:54.03,TIME:2025-12-03 18:18:43,ID:IoT_Sensor_001

[STEP 3] INITIALIZING ENCRYPTION SYSTEM
----------------------------------------
🔑 Encryption Key Generated: ab533c64805b6da9a77c0fab358c56f4

[STEP 4] ENCRYPTING DATA
----------------------------------------
🔒 Encrypted Data (Base64):
E8AszDMnW7IR/467V2/EwewH5V9P0qiGO+DfE1fiO6rOvs09rmYI8vZzpObjPhPisDmeDpylllGbdYMw...
📏 Packet Size: 80 bytes

[STEP 5] SIMULATING TRANSMISSION
----------------------------------------
📡 Transmitting data to cloud server...
✅ Transmission successful!

[STEP 6] DECRYPTING DATA
----------------------------------------
🔒 Decrypted Data: TEMP:25.64,HUM:54.03,TIME:2025-12-03 18:18:43,ID:IoT_Sensor_001
```

```
[STEP 7] PARSING DECRYPTED DATA
----------------------------------------
📊 Parsed Data:
  • TEMP: 25.64
  • HUM: 54.03
  • TIME: 2025-12-03 18:18:43
  • ID: IoT_Sensor_001

[STEP 8] DATA INTEGRITY VERIFICATION
----------------------------------------
✅ VERIFICATION SUCCESSFUL - Data integrity confirmed!
✅ PART I COMPLETED SUCCESSFULLY!
```

```
print("=" * 70)
print("🔄 PART II: IoT DEVICE LIFECYCLE - FUNCTIONS")
print("=" * 70)
print()

class IoTDeviceLifecycle:
    """Class to simulate IoT device security lifecycle"""

    def __init__(self, device_name="IoT_Device_001"):
        self.device_name = device_name
        self.firmware_version = "1.0.0"
        self.logs = []

    def log_event(self, stage, message):
        """Log lifecycle events with timestamp"""
        timestamp = datetime.datetime.now().strftime("%H:%M:%S")
        log_entry = f"[{timestamp}] [Stage {stage}] {message}"
        self.logs.append(log_entry)
        print(log_entry)
        return log_entry

    def threat_modeling(self):
        """Stage 1: Threat Modeling"""
        print("\n📋 STAGE 1: THREAT MODELING")
        print("-" * 30)

        self.log_event(1, "Starting threat modeling process")
        time.sleep(0.5)

        threats = [
            "Unauthorized physical access",
            "Network eavesdropping",
            "Firmware tampering",
            "Denial of Service (DoS) attacks"
        ]

        for i, threat in enumerate(threats, 1):
            self.log_event(1, f"Threat identified {i}: {threat}")
            time.sleep(0.3)

        self.log_event(1, "Threat modeling completed ✓")
        return True

    def secure_boot(self):
        """Stage 2: Secure Boot"""
        print("\n🔐 STAGE 2: SECURE BOOT")
        print("-" * 30)

        self.log_event(2, "Initializing secure boot process")
        time.sleep(0.5)

        # Simulate firmware hash verification
        firmware_hash = hashlib.sha256(b"secure_firmware_v1.0").hexdigest()
        stored_hash = hashlib.sha256(b"secure_firmware_v1.0").hexdigest()

        self.log_event(2, f"Calculated firmware hash: {firmware_hash[:16]}...")
        self.log_event(2, f"Stored reference hash: {stored_hash[:16]}...")

        if firmware_hash == stored_hash:
            self.log_event(2, "Firmware integrity verified ✓")
            return True
        else:
            self.log_event(2, "Firmware verification failed ✗")
            return False

    def key_injection(self):
        """Stage 3: Secure Key Injection"""
        print("\n🔑 STAGE 3: KEY INJECTION")
        print("-" * 30)
        self.log_event(3, "Starting secure key injection")
        time.sleep(0.5)

        keys = {
            "encryption_key": get_random_bytes(16).hex(),
            "authentication_key": get_random_bytes(32).hex(),
            "device_certificate": "CERT_" + hashlib.md5(b"device_cert").hexdigest()[:16].upper()
        }

        for key_name, key_value in keys.items():
            self.log_event(3, f"Injecting {key_name}: {key_value[:16]}...")
            time.sleep(0.2)

        self.log_event(3, "All keys injected securely ✓")
        return True

    def ota_update(self):
        """Stage 4: OTA Firmware Update"""
        print("\n📲 STAGE 4: OTA UPDATE CHECK")
        print("-" * 30)

        self.log_event(4, "Checking for OTA firmware updates")
        time.sleep(0.5)

        # Simulate update check
        if random.choice([True, False]):
            self.log_event(4, f"Update available: v{self.firmware_version} → v1.1.0")
            time.sleep(0.5)
            self.log_event(4, "Verifying update signature...")
            self.log_event(4, "Signature verification successful ✓")
            self.firmware_version = "1.1.0"
            self.log_event(4, f"Firmware updated to v{self.firmware_version} ✓")
        else:
            self.log_event(4, f"No updates available (Current: v{self.firmware_version})")

        return True

    def decommission(self):
        """Stage 5: Secure Decommissioning"""
        print("\n♻️  STAGE 5: DECOMMISSIONING")
        print("-" * 30)

        self.log_event(5, "Starting secure decommissioning process")
        time.sleep(0.5)

        self.log_event(5, "Wiping encryption keys...")
        time.sleep(0.3)

        self.log_event(5, "Wiping authentication keys...")
        time.sleep(0.3)

        self.log_event(5, "Performing factory reset...")
        time.sleep(0.5)

        self.log_event(5, "Device successfully decommissioned ✓")
        return True

    def run_full_lifecycle(self):
        """Execute all 5 stages of security lifecycle"""
        print(f"\n🚀 STARTING DEVICE LIFECYCLE: {self.device_name}")
        print("=" * 50)

        stages = [
            self.threat_modeling,
            self.secure_boot,
            self.key_injection,
            self.ota_update,
            self.decommission
        ]

        for i, stage_func in enumerate(stages, 1):
            if not stage_func():
                print(f"\n❌ Stopped at Stage {i}")
                return False

        print(f"\n✅ ALL 5 STAGES COMPLETED SUCCESSFULLY!")
        print(f"📊 Total Events Logged: {len(self.logs)}")
        print(f"📈 Final Firmware Version: v{self.firmware_version}")
        return True
print("✅ Part II class and functions defined successfully!")
print()
```

```
======================================================================
🔄 PART II: IoT DEVICE LIFECYCLE - FUNCTIONS
======================================================================

✅ Part II class and functions defined successfully!
```

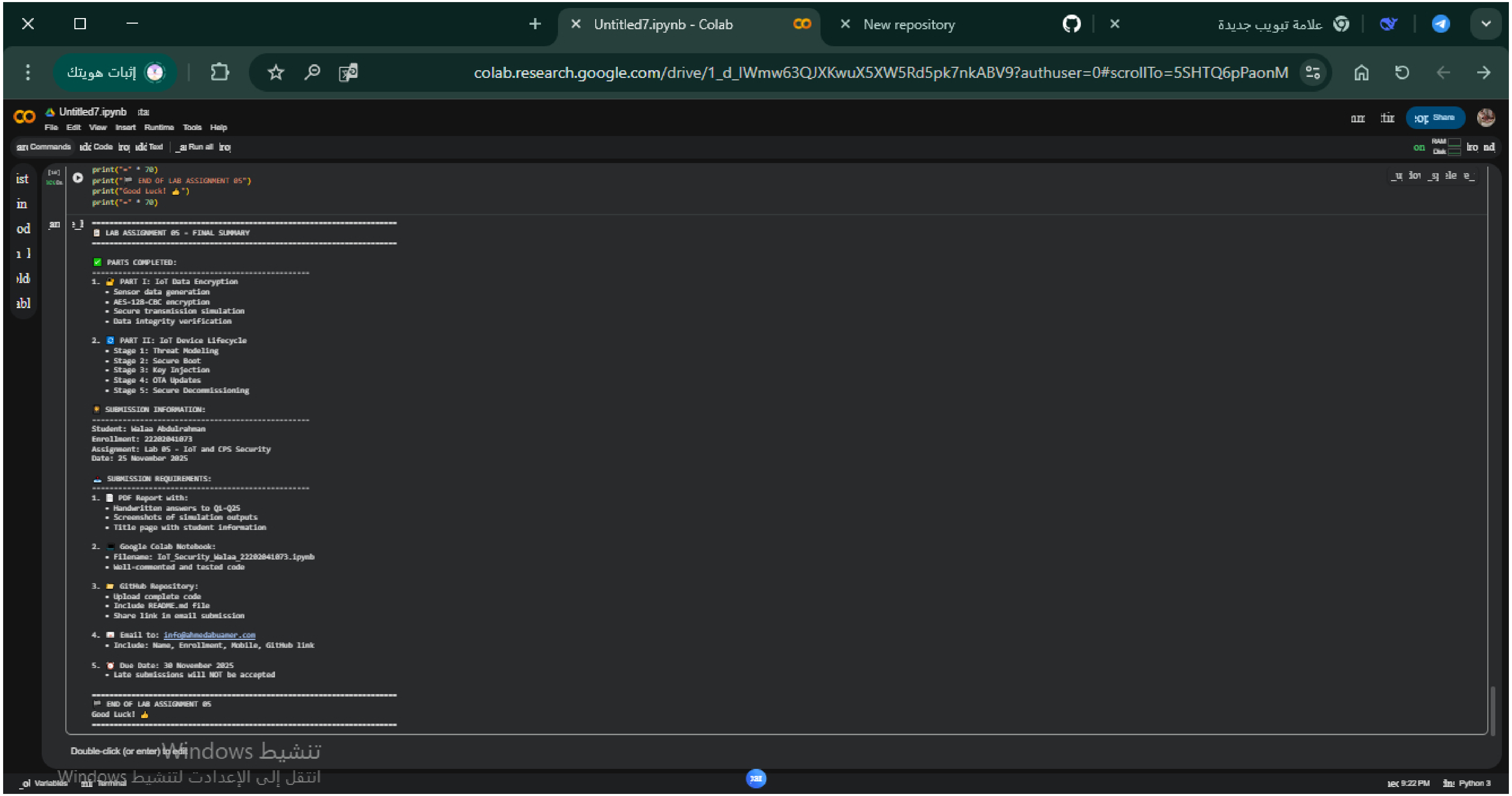
```
print("=" * 70)
print("🚀 PART II: EXECUTING IoT DEVICE LIFECYCLE SIMULATION")
print("=" * 70)
print()

# Create device instance
device = IoTDeviceLifecycle("Smart_IoT_Device_2025")

# Run full lifecycle
print("Starting IoT Device Lifecycle Simulation...")
print("-" * 50)
success = device.run_full_lifecycle()

print("\n" + "=" * 50)
if success:
    print("🎉 LIFECYCLE SIMULATION COMPLETED!")
else:
    print("⚠️  LIFECYCLE SIMULATION HAD ISSUES")

print("\n📜 COMPLETE EVENT LOG:")
print("-" * 50)
for log in device.logs:
    print(log)
print()
```

```
======================================================================
🚀 PART II: EXECUTING IoT DEVICE LIFECYCLE SIMULATION
======================================================================

Starting IoT Device Lifecycle Simulation...
--------------------------------------------------

🚀 STARTING DEVICE LIFECYCLE: Smart_IoT_Device_2025
==================================================

📋 STAGE 1: THREAT MODELING
------------------------------
[18:19:26] [Stage 1] Starting threat modeling process
[18:19:26] [Stage 1] Threat identified 1: Unauthorized physical access
[18:19:26] [Stage 1] Threat identified 2: Network eavesdropping
[18:19:27] [Stage 1] Threat identified 3: Firmware tampering
[18:19:27] [Stage 1] Threat identified 4: Denial of Service (DoS) attacks
[18:19:27] [Stage 1] Threat modeling completed ✓

🔐 STAGE 2: SECURE BOOT
------------------------------
[18:19:27] [Stage 2] Initializing secure boot process
[18:19:28] [Stage 2] Calculated firmware hash: de51ee04110845d7...
[18:19:28] [Stage 2] Stored reference hash: de51ee04110845d7...
[18:19:28] [Stage 2] Firmware integrity verified ✓

🔑 STAGE 3: KEY INJECTION
------------------------------
[18:19:28] [Stage 3] Starting secure key injection
[18:19:28] [Stage 3] Injecting encryption_key: f13d5f83f5e010a2...
[18:19:28] [Stage 3] Injecting authentication_key: 368da575e9cb4496...
[18:19:29] [Stage 3] Injecting device_certificate: CERT_2E33F0F85B1...
[18:19:29] [Stage 3] All keys injected securely ✓

📲 STAGE 4: OTA UPDATE CHECK
------------------------------
[18:19:29] [Stage 4] Checking for OTA firmware updates
[18:19:29] [Stage 4] No updates available (Current: v1.0.0)

♻️  STAGE 5: DECOMMISSIONING
------------------------------
[18:19:29] [Stage 5] Starting secure decommissioning process
[18:19:30] [Stage 5] Wiping encryption keys...
[18:19:30] [Stage 5] Wiping authentication keys...
[18:19:30] [Stage 5] Performing factory reset...
[18:19:31] [Stage 5] Device successfully decommissioned ✓

✅ ALL 5 STAGES COMPLETED SUCCESSFULLY!
📊 Total Events Logged: 22
📈 Final Firmware Version: v1.0.0

==================================================
🎉 LIFECYCLE SIMULATION COMPLETED!

📜 COMPLETE EVENT LOG:
--------------------------------------------------
[18:19:26] [Stage 1] Starting threat modeling process
[18:19:26] [Stage 1] Threat identified 1: Unauthorized physical access
```

```
print("=" * 70)
print("📋 LAB ASSIGNMENT 05 - FINAL SUMMARY")
print("=" * 70)
print()

print("✅ PARTS COMPLETED:")
print("-" * 50)
print("1. 🔐 PART I: IoT Data Encryption")
print("   • Sensor data generation")
print("   • AES-128-CBC encryption")
print("   • Secure transmission simulation")
print("   • Data integrity verification")
print()
print("2. 🔄 PART II: IoT Device Lifecycle")
print("   • Stage 1: Threat Modeling")
print("   • Stage 2: Secure Boot")
print("   • Stage 3: Key Injection")
print("   • Stage 4: OTA Updates")
print("   • Stage 5: Secure Decommissioning")
print()

print("👩‍🎓 SUBMISSION INFORMATION:")
print("-" * 50)
print("Student: Walaa Abdulrahman")
print("Enrollment: 22202041073")
print("Assignment: Lab 05 - IoT and CPS Security")
print("Date: 25 November 2025")
print()

print("📤 SUBMISSION REQUIREMENTS:")
print("-" * 50)
print("1. 📄 PDF Report with:")
print("   • Handwritten answers to Q1-Q25")
print("   • Screenshots of simulation outputs")
print("   • Title page with student information")
print()
print("2. 💻 Google Colab Notebook:")
print("   • Filename: IoT_Security_Walaa_22202041073.ipynb")
print("   • Well-commented and tested code")
print()
print("3. 📁 GitHub Repository:")
print("   • Upload complete code")
print("   • Include README.md file")
print("   • Share link in email submission")
print()
print("4. 📧 Email to: info@ahmedabuamer.com")
print("   • Include: Name, Enrollment, Mobile, GitHub link")
print()
print("5. ⏰ Due Date: 30 November 2025")
print("   • Late submissions will NOT be accepted")
print()

print("=" * 70)
print("🏁 END OF LAB ASSIGNMENT 05")
print("Good Luck! 👍")
print("=" * 70)
```

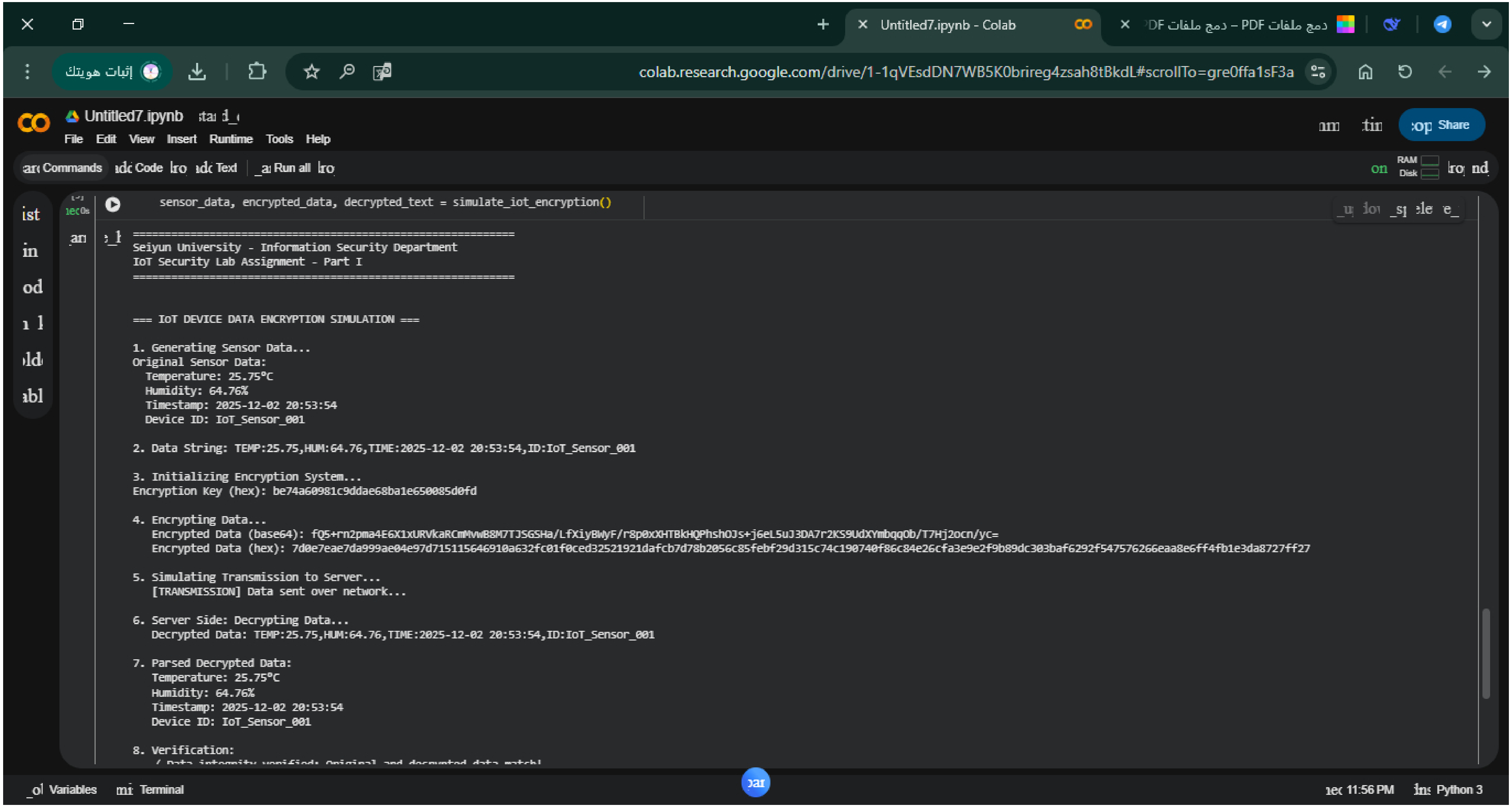
```
======================================================================
📋 LAB ASSIGNMENT 05 - FINAL SUMMARY
======================================================================

✅ PARTS COMPLETED:
--------------------------------------------------
1. 🔐 PART I: IoT Data Encryption
   • Sensor data generation
   • AES-128-CBC encryption
   • Secure transmission simulation
   • Data integrity verification

2. 🔄 PART II: IoT Device Lifecycle
   • Stage 1: Threat Modeling
   • Stage 2: Secure Boot
   • Stage 3: Key Injection
   • Stage 4: OTA Updates
   • Stage 5: Secure Decommissioning

👩‍🎓 SUBMISSION INFORMATION:
--------------------------------------------------
```

```
Student: Walaa Abdulrahman
Enrollment: 22202041073
Assignment: Lab 05 - IoT and CPS Security
Date: 25 November 2025

📤 SUBMISSION REQUIREMENTS:
--------------------------------------------------
1. 📄 PDF Report with:
   • Handwritten answers to Q1-Q25
   • Screenshots of simulation outputs
   • Title page with student information

2. 💻 Google Colab Notebook:
   • Filename: IoT_Security_Walaa_22202041073.ipynb
   • Well-commented and tested code

3. 📁 GitHub Repository:
   • Upload complete code
   • Include README.md file
   • Share link in email submission

4. 📧 Email to: info@ahmedabuamer.com
   • Include: Name, Enrollment, Mobile, GitHub link

5. ⏰ Due Date: 30 November 2025
   • Late submissions will NOT be accepted

======================================================================
🏁 END OF LAB ASSIGNMENT 05
Good Luck! 👍
======================================================================
```


Untitled7.ipynb - Colab
New repository
علامة تبويب جديدة
إثبات هويتك
colab.research.google.com/drive/1_d_lWmw63QJXKwuX5XW5Rd5pk7nkABV9?authuser=0#scrollTo=5SHTQ6pPaonM
Untitled7.ipynb
File Edit View Insert Runtime Tools Help
Commands
Code
Text
Run all
Share
RAM
Disk
Double-click (or enter) to edit
Variables
Terminal
9:22 PM
Python 3
تنشيط Windows
انتقل إلى الإعدادات لتنشيط Windows

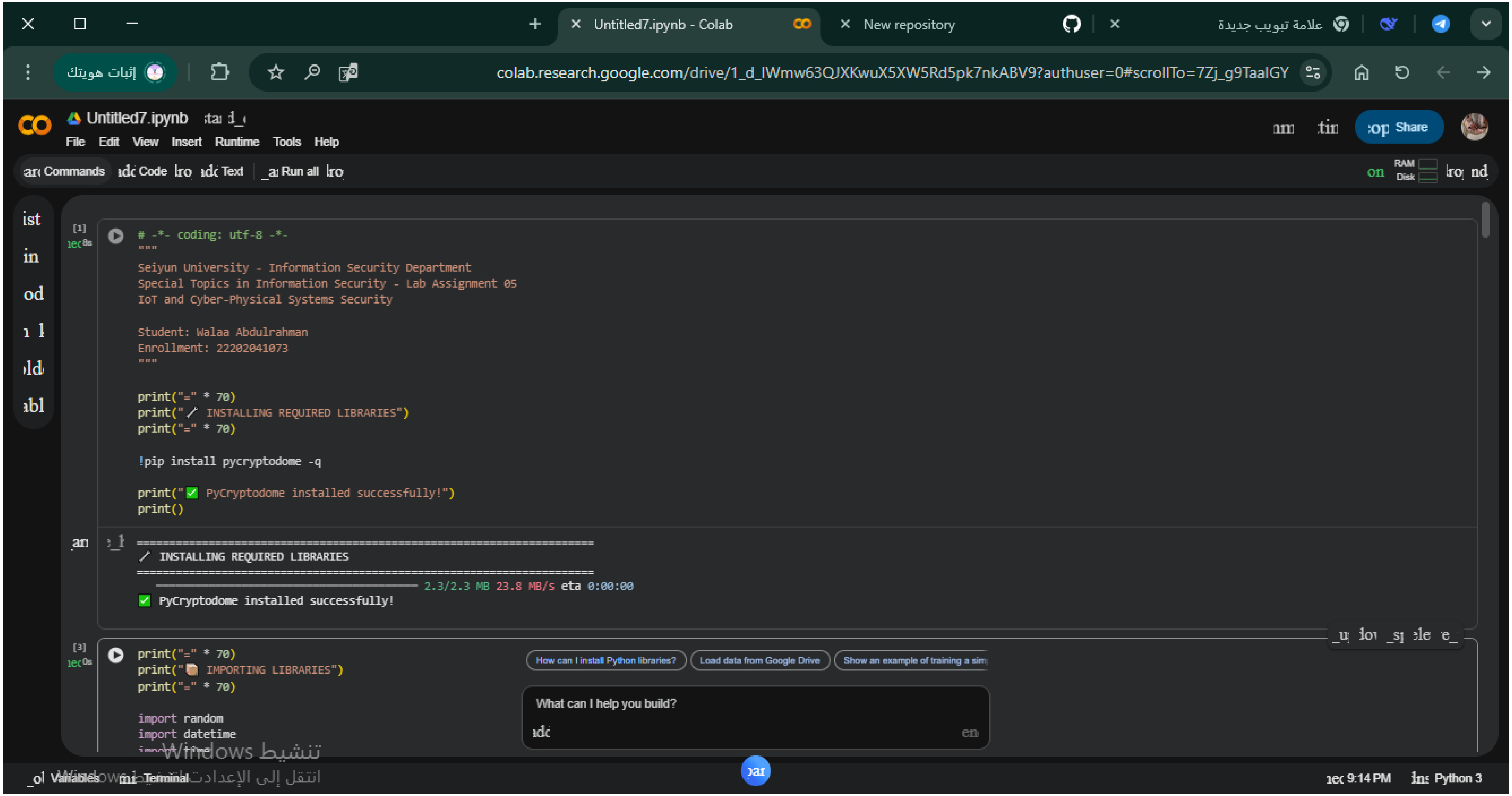
```
print("=" * 70)
print("🏁 END OF LAB ASSIGNMENT 05")
print("Good Luck! 👍")
print("=" * 70)
```

```
======================================================================
📋 LAB ASSIGNMENT 05 - FINAL SUMMARY
======================================================================

✅ PARTS COMPLETED:
--------------------------------------------------
1. 🔐 PART I: IoT Data Encryption
   • Sensor data generation
   • AES-128-CBC encryption
   • Secure transmission simulation
   • Data integrity verification

2. 🔄 PART II: IoT Device Lifecycle
   • Stage 1: Threat Modeling
   • Stage 2: Secure Boot
   • Stage 3: Key Injection
   • Stage 4: OTA Updates
   • Stage 5: Secure Decommissioning

🎓 SUBMISSION INFORMATION:
--------------------------------------------------
Student: Walaa Abdulrahman
Enrollment: 22202041073
Assignment: Lab 05 - IoT and CPS Security
Date: 25 November 2025

📤 SUBMISSION REQUIREMENTS:
--------------------------------------------------
1. 📄 PDF Report with:
   • Handwritten answers to Q1-Q25
   • Screenshots of simulation outputs
   • Title page with student information

2. 💻 Google Colab Notebook:
   • Filename: IoT_Security_Walaa_22202041073.ipynb
   • Well-commented and tested code

3. 📁 GitHub Repository:
   • Upload complete code
   • Include README.md file
   • Share link in email submission

4. 📧 Email to: info@ahmedabuamer.com
   • Include: Name, Enrollment, Mobile, GitHub link

5. ⏰ Due Date: 30 November 2025
   • Late submissions will NOT be accepted

======================================================================
🏁 END OF LAB ASSIGNMENT 05
Good Luck! 👍
======================================================================
```


Untitled7.ipynb - Colab
colab.research.google.com/drive/1-1qVEsdDN7WB5K0brireg4zsah8tBkdL#scrollTo=gre0ffa1sF3a
Untitled7.ipynb
File Edit View Insert Runtime Tools Help
Commands Code Text Run all
Share
RAM Disk
Variables Terminal
11:56 PM Python 3

```
sensor_data, encrypted_data, decrypted_text = simulate_iot_encryption()
```

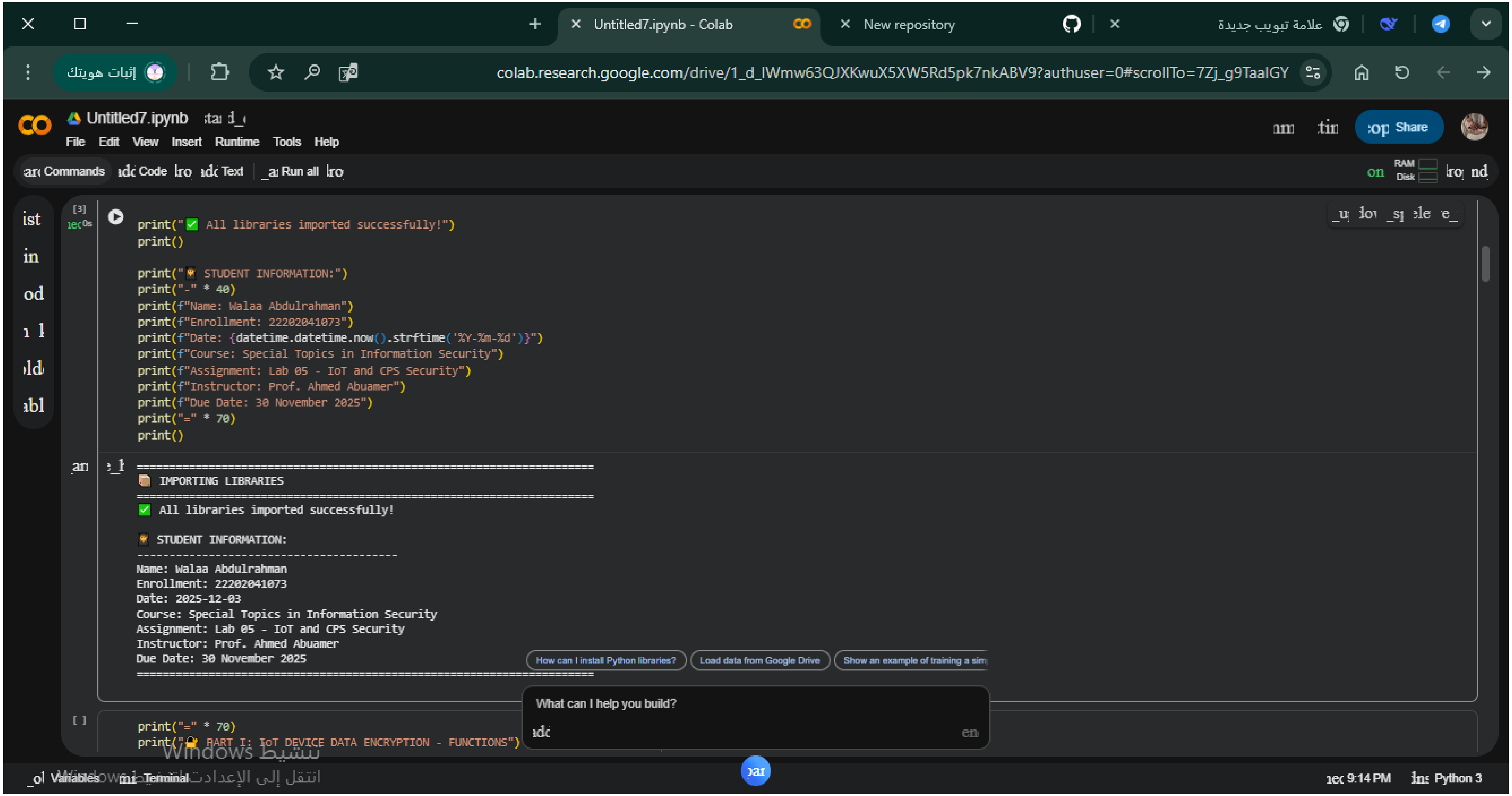
```
============================================================
Seiyun University - Information Security Department
IoT Security Lab Assignment - Part I
============================================================


=== IoT DEVICE DATA ENCRYPTION SIMULATION ===

1. Generating Sensor Data...
Original Sensor Data:
  Temperature: 25.75°C
  Humidity: 64.76%
  Timestamp: 2025-12-02 20:53:54
  Device ID: IoT_Sensor_001

2. Data String: TEMP:25.75,HUM:64.76,TIME:2025-12-02 20:53:54,ID:IoT_Sensor_001

3. Initializing Encryption System...
Encryption Key (hex): be74a60981c9ddae68ba1e650085d0fd

4. Encrypting Data...
   Encrypted Data (base64): fQ5+rn2pma4E6X1xURVkaRCmMvwB8M7TJSGSHa/LfXiyBWyF/r8p0xXHTBkHQPhshOJs+j6eL5uJ3DA7r2KS9UdXYmbqqOb/T7Hj2ocn/yc=
   Encrypted Data (hex): 7d0e7eae7da999ae04e97d715115646910a632fc01f0ced32521921dafcb7d78b2056c85febf29d315c74c190740f86c84e26cfa3e9e2f9b89dc303baf6292f547576266eaa8e6ff4fb1e3da8727ff27

5. Simulating Transmission to Server...
   [TRANSMISSION] Data sent over network...

6. Server Side: Decrypting Data...
   Decrypted Data: TEMP:25.75,HUM:64.76,TIME:2025-12-02 20:53:54,ID:IoT_Sensor_001

7. Parsed Decrypted Data:
   Temperature: 25.75°C
   Humidity: 64.76%
   Timestamp: 2025-12-02 20:53:54
   Device ID: IoT_Sensor_001

8. Verification:
   ✓ Data integrity verified: Original and decrypted data match!
```


Untitled7.ipynb - Colab
New repository
colab.research.google.com/drive/1_d_IWmw63QJXKwuX5XW5Rd5pk7nkABV9?authuser=0#scrollTo=7Zj_g9TaalGY
Untitled7.ipynb
File Edit View Insert Runtime Tools Help
Commands
Code
Text
Run all
Share
RAM
Disk
How can I install Python libraries?
Load data from Google Drive
Show an example of training a sim
What can I help you build?
Variables
Terminal
9:14 PM
Python 3
تنشيط Windows
انتقل إلى الإعدادات لتنشيط Windows

```
[1]
# -*- coding: utf-8 -*-
"""
Seiyun University - Information Security Department
Special Topics in Information Security - Lab Assignment 05
IoT and Cyber-Physical Systems Security

Student: Walaa Abdulrahman
Enrollment: 22202041073
"""

print("=" * 70)
print("🔧 INSTALLING REQUIRED LIBRARIES")
print("=" * 70)

!pip install pycryptodome -q

print("✅ PyCryptodome installed successfully!")
print()
```

```
======================================================================
🔧 INSTALLING REQUIRED LIBRARIES
======================================================================
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.3/2.3 MB 23.8 MB/s eta 0:00:00
✅ PyCryptodome installed successfully!
```

```
[3]
print("=" * 70)
print("📦 IMPORTING LIBRARIES")
print("=" * 70)

import random
import datetime
```

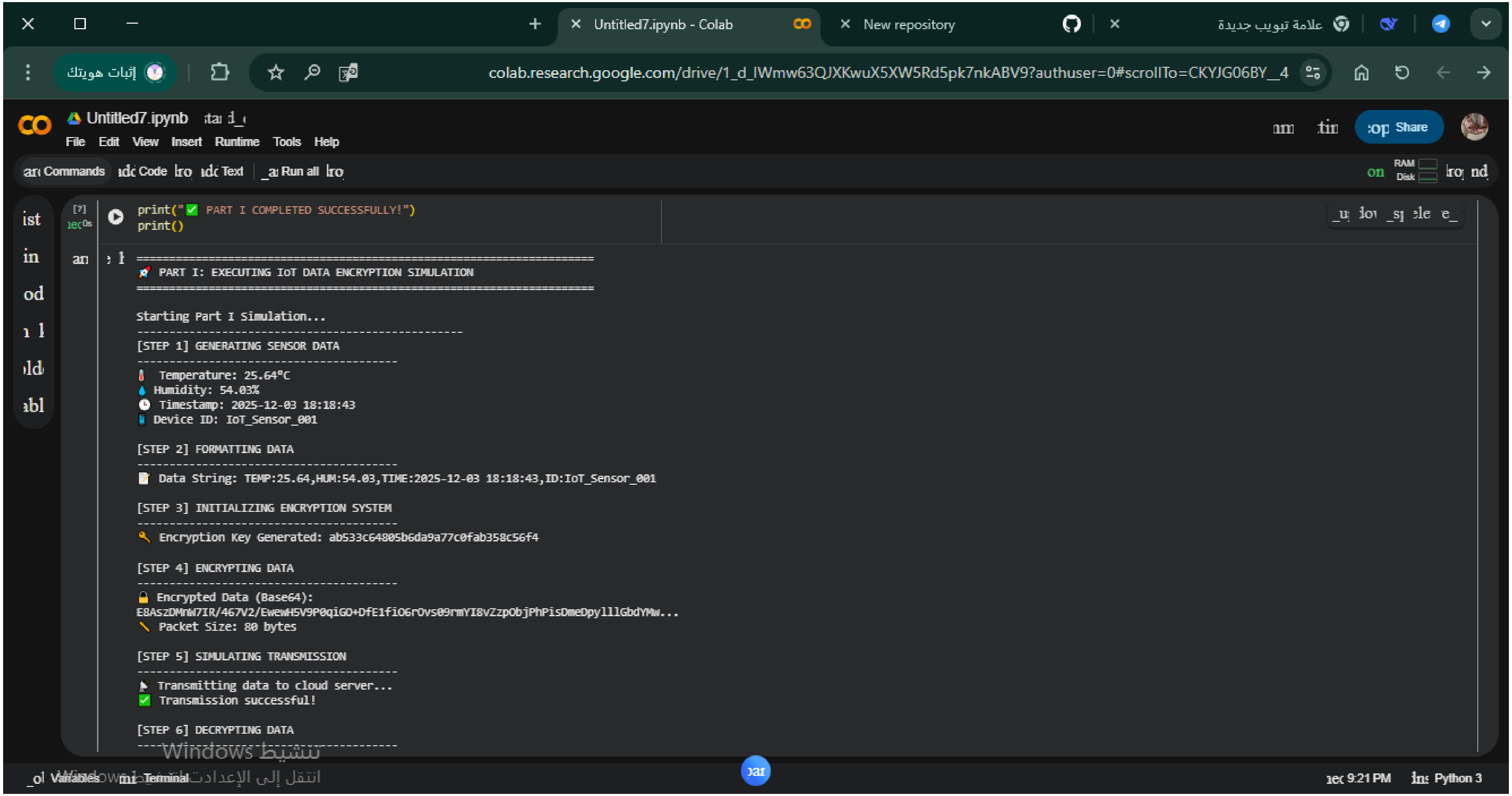


Untitled7.ipynb - Colab
New repository
colab.research.google.com/drive/1_d_lWmw63QJXKwuX5XW5Rd5pk7nkABV9?authuser=0#scrollTo=7Zj_g9TaalGY
Untitled7.ipynb
File Edit View Insert Runtime Tools Help
Commands Code Text Run all
Share
RAM
Disk
[3]
How can I install Python libraries?
Load data from Google Drive
Show an example of training a sim
What can I help you build?
Variables Terminal
9:14 PM
Python 3

```
print("✅ All libraries imported successfully!")
print()

print("🧑‍🎓 STUDENT INFORMATION:")
print("-" * 40)
print(f"Name: Walaa Abdulrahman")
print(f"Enrollment: 22202041073")
print(f"Date: {datetime.datetime.now().strftime('%Y-%m-%d')}")
print(f"Course: Special Topics in Information Security")
print(f"Assignment: Lab 05 - IoT and CPS Security")
print(f"Instructor: Prof. Ahmed Abuamer")
print(f"Due Date: 30 November 2025")
print("=" * 70)
print()
```

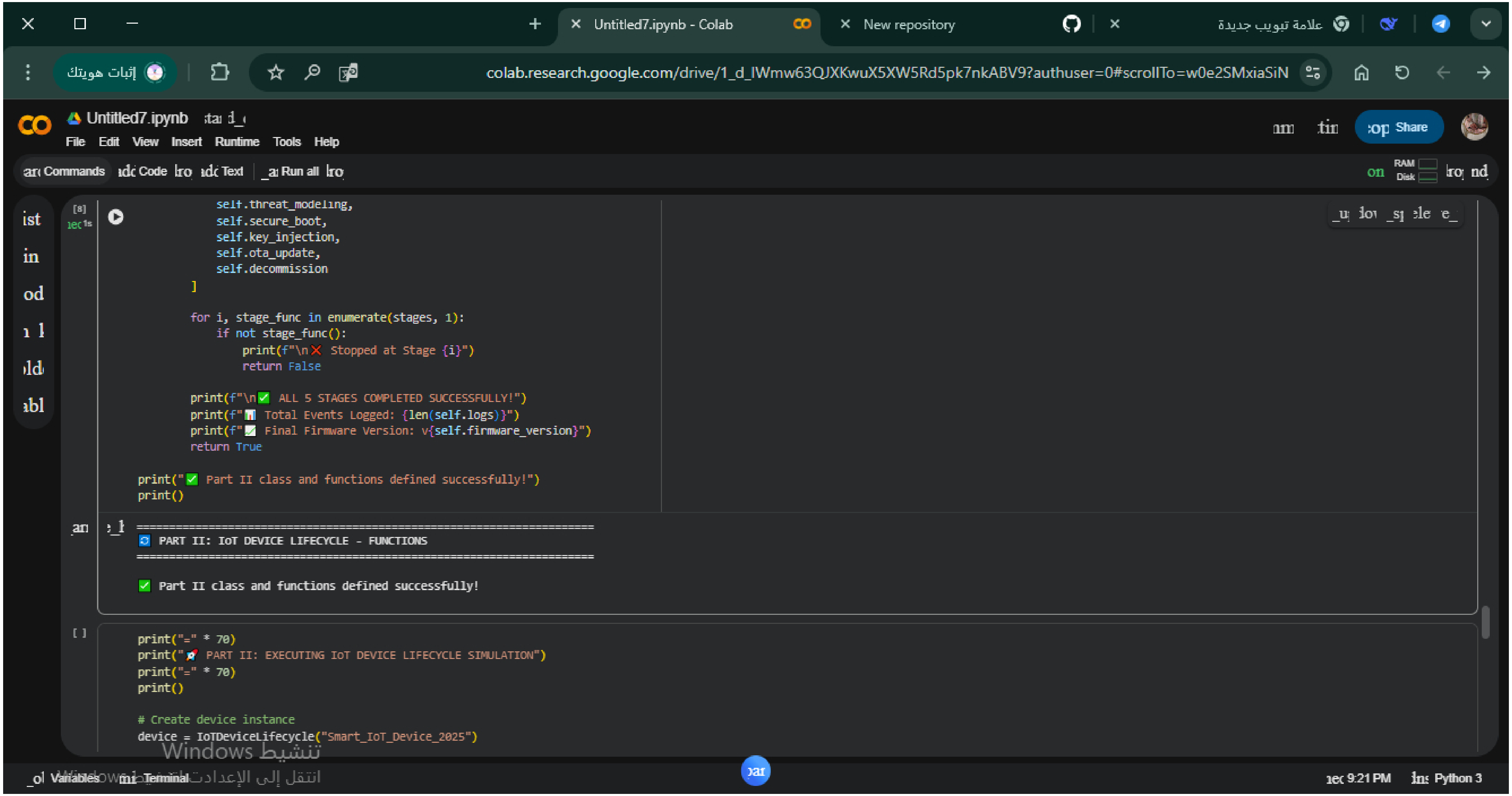
```
======================================================================
📦 IMPORTING LIBRARIES
======================================================================
✅ All libraries imported successfully!

🧑‍🎓 STUDENT INFORMATION:
----------------------------------------
Name: Walaa Abdulrahman
Enrollment: 22202041073
Date: 2025-12-03
Course: Special Topics in Information Security
Assignment: Lab 05 - IoT and CPS Security
Instructor: Prof. Ahmed Abuamer
Due Date: 30 November 2025
======================================================================
```

```
print("=" * 70)
print("🔐 PART I: IoT DEVICE DATA ENCRYPTION - FUNCTIONS")
```



```
print("✅ PART I COMPLETED SUCCESSFULLY!")
print()
```

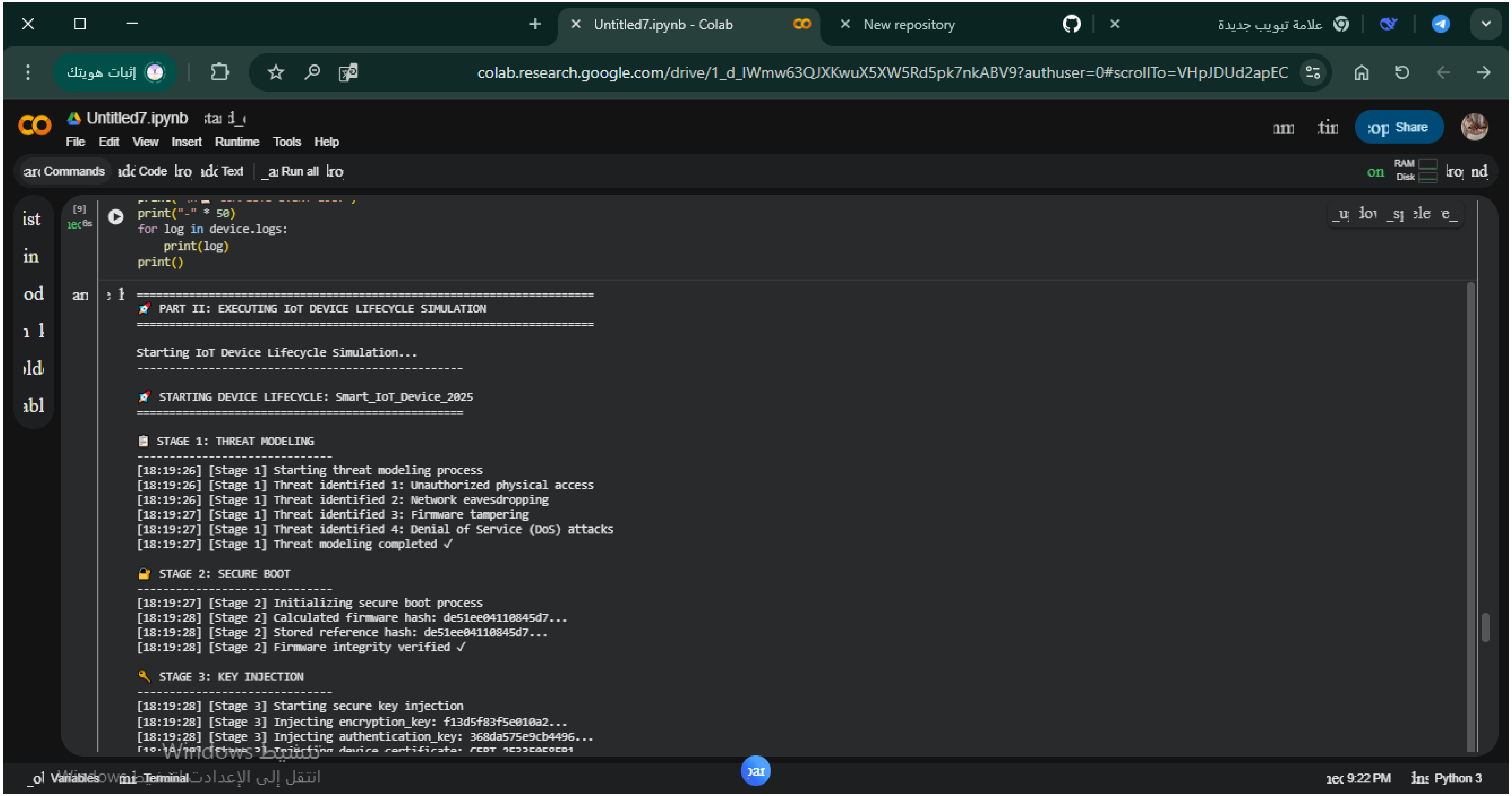
```
======================================================================
🚀 PART I: EXECUTING IoT DATA ENCRYPTION SIMULATION
======================================================================

Starting Part I Simulation...
--------------------------------------------------
[STEP 1] GENERATING SENSOR DATA
----------------------------------------
🌡 Temperature: 25.64°C
💧 Humidity: 54.03%
🕒 Timestamp: 2025-12-03 18:18:43
📱 Device ID: IoT_Sensor_001

[STEP 2] FORMATTING DATA
----------------------------------------
📝 Data String: TEMP:25.64,HUM:54.03,TIME:2025-12-03 18:18:43,ID:IoT_Sensor_001

[STEP 3] INITIALIZING ENCRYPTION SYSTEM
----------------------------------------
🔑 Encryption Key Generated: ab533c64805b6da9a77c0fab358c56f4

[STEP 4] ENCRYPTING DATA
----------------------------------------
🔒 Encrypted Data (Base64):
E8AszDMnW7IR/467V2/EwewH5V9P0qiGO+DfE1fiO6rOvs09rmYI8vZzpObjPhPisDmeDpylllGbdYMw...
📏 Packet Size: 80 bytes

[STEP 5] SIMULATING TRANSMISSION
----------------------------------------
📡 Transmitting data to cloud server...
✅ Transmission successful!

[STEP 6] DECRYPTING DATA
----------------------------------------
```



```
            self.threat_modeling,
            self.secure_boot,
            self.key_injection,
            self.ota_update,
            self.decommission
        ]

        for i, stage_func in enumerate(stages, 1):
            if not stage_func():
                print(f"\n❌ Stopped at Stage {i}")
                return False

        print(f"\n✅ ALL 5 STAGES COMPLETED SUCCESSFULLY!")
        print(f"📊 Total Events Logged: {len(self.logs)}")
        print(f"📈 Final Firmware Version: v{self.firmware_version}")
        return True

print("✅ Part II class and functions defined successfully!")
print()
```

```
======================================================================
🔄 PART II: IoT DEVICE LIFECYCLE - FUNCTIONS
======================================================================

✅ Part II class and functions defined successfully!
```

```
print("=" * 70)
print("🚀 PART II: EXECUTING IoT DEVICE LIFECYCLE SIMULATION")
print("=" * 70)
print()

# Create device instance
device = IoTDeviceLifecycle("Smart_IoT_Device_2025")
```



```
print("-" * 50)
for log in device.logs:
    print(log)
print()
```

```
======================================================================
🚀 PART II: EXECUTING IoT DEVICE LIFECYCLE SIMULATION
======================================================================

Starting IoT Device Lifecycle Simulation...
--------------------------------------------------

🚀 STARTING DEVICE LIFECYCLE: Smart_IoT_Device_2025
==================================================

📋 STAGE 1: THREAT MODELING
------------------------------
[18:19:26] [Stage 1] Starting threat modeling process
[18:19:26] [Stage 1] Threat identified 1: Unauthorized physical access
[18:19:26] [Stage 1] Threat identified 2: Network eavesdropping
[18:19:27] [Stage 1] Threat identified 3: Firmware tampering
[18:19:27] [Stage 1] Threat identified 4: Denial of Service (DoS) attacks
[18:19:27] [Stage 1] Threat modeling completed ✓

🔐 STAGE 2: SECURE BOOT
------------------------------
[18:19:27] [Stage 2] Initializing secure boot process
[18:19:28] [Stage 2] Calculated firmware hash: de51ee04110845d7...
[18:19:28] [Stage 2] Stored reference hash: de51ee04110845d7...
[18:19:28] [Stage 2] Firmware integrity verified ✓

🔑 STAGE 3: KEY INJECTION
------------------------------
[18:19:28] [Stage 3] Starting secure key injection
[18:19:28] [Stage 3] Injecting encryption_key: f13d5f83f5e010a2...
[18:19:28] [Stage 3] Injecting authentication_key: 368da575e9cb4496...
[18:19:28] [Stage 3] Injecting device certificate: CERT_2523F058F81
```

Lab Assignment No. 05

. IoT and Cyber-Physical Systems Security

Q1- Define the term IoT. Give two example of IoT devices?

A- IoT is a networks of Physical devices with sensors and Software that connect to the internet to exchange data Example: Samert home Cameras thermostats.

Q2 What is a Cyber-Physical system (CPS)?

A. A System ~~Focuses on device Conn~~ that integrates physical prosesses With digital Computing and Control, often used in critical infrastructure

Q3. What is the difference between IoT and CPS?

A3- IoT :- Focusses on device connectivity and data collection

CPS :- integrates Computing with physical Control processes.

Q4: What are botnets? Mention one Well-Known IoT bothet attack?

A4: Botnets are network of Compromised devices controlled remotely by an attacker. Example: The Mirai botnet attack (2016).

Q5: What is Secure Boot in embedded systems?

A5: A security mechanism that ensures only trusted, digitally signed Software loads during startup

Q6 Explain the Five Stages of the Embedded System Security Lifecycle With one example each?

A6: 1- Threat Modeling : Identify Potential risks (e.g analyzing smart Camer)

2- Secure Boot : Verify Software at startup (e.g ~~in a Secure hardware element~~ checking Firware signature)

3- Secure Key Injection: Store cryptographic Keys securely (e.g in a secure hardware element)

4: Secure OTA Updates: update device Software securely over the air

5- Secure Decommissioning: wipe data and Keys before device disposal

Q7. What are the main vulnerabilities in IoT devices? Give two real-world attack example?

A7: Vulnerbilities: Weak default passwords, unpatched Firmware, unencrypted communication.

Attacks Examples: Mirai botnet, Jeep Hack.

Q8. What does OTA updata mean in the IoT security lifecycle?

A8. The ability to securely updata device Software wirelessly, ensuring updata integrity before installation.

Q9 Defing Lightweight Cryptography. Why is it essential in IoT?

A9 Cryptography designed For divces with limited resources (CPU, memory power)

Importance: Provides strong Security with low resource consumption

Q10. Describe how botnets like Mirai Compromise IoT systems and how such attacks can be prevented?

A10- Compromise: Scan for devices using default credentials and infect them.

prevention:- change default passwords, regular Software updates, close unused ports.

Q11 Explain the role of secure firmware updates and hardware root of trust in IoT device Security?

A11- Secure Firmware Updates: Ensure Updates are authentic and untampered

Hardware Root of Trust: A secure hardware component that stores keys and performs crypto operations.

Q12. What is the Function of PKI in IoT Communication?

A12 PKI provides digital certificates for mutual authentication between device and server, ensuring data confidentiality and integrity.

Q24_ Compare Embedded System Security and Traditional Computer Security?

A 24_ Embedded System: Limited resources difficult updataes, prone to physical attacks.

Traditional Computers: More resources, easier updates, Stronger software protection.

Q25_ Evaluate the challenges of deploying lightweight cryptography on Constrained IoT device?

A 25_ Challenges: Limited memory. Power Consumption, balancing security and performance.

Solutions: Use dedicated algorithms like PRESENT, optimize implementation efficiency.

Q13- Define Edge Computing and its role in IoT Security?

A

A13- Processing data near its source (insteed of distant servers)
Security Role: Reduces data in transit, limiting exposure to attacks

Q14- Illustrate ICS/SADA System architecure and list common vulnerabilities?

A14- Simple Architecture: Sensors → PLCs → SCADA servers → Operator interface
Common Vulnerabilities: Weak access control, lacks of encryption, unpatched software.

Q15- Explain threat modeling and how it applies during the IoT p product design phase?

A15- A systematic process to identify and assess potential vulnerabilities
Application: Analyzing how the device handles data and finding weaknesses before manufacturing

Q16- Discuss the concept of a trusted execution environment (TEE) and how it enhances embedded security?

A16- An isolated area in the processor for exacuting sensitive Operations
Enhancement: protects data even if the main OS is compromised.

Q17- Describe the role of blockchain in improving IoT Security?

A17- Ensures data integrity through immutability. Used for decentralized authentication and preventing spoofing.

Q18- Explain how secure multi-party computation (MPC) can be used in IoT for privacy-perseving data sharing?

A18- Allows multiple devices to process shared data without revealing individual readings.

Q19- Write a short note on lightweight Cryptography aloriths suitable For IoT ?

A19- Algorithms like PRESENT, SPEck, SIMON (lightweight ciphers), PHOTON, SPONGENT (lightweight ciphers), ECC (efficient public-key crypto).

Q20- Discuss a real-world IoT security breach (e.g. Mirai, Stuxnet, Jeep Hack) ?

A20- Jeep Hack (2015):- Researchers exploited a vulnerability in the UConnect system, allowing remote Control of brakes, A/C, and enterainment.

Q 21- In a Smart City traffic System, how Can you ensure data authenticity? Which Cryptographic techniques world you chooes?

A21- Authenticity:- Use digital Signatures and mutual authentication
Techniques:- ECC For Signatures, AES For encryption, SHA-256 For hashing.

Q22- A Factory's SCADA System is Connected to the interent, how could an attacker exploit it? Suggest preventive measures?

A22- Exploit: Breach via weak prosswords or Software vulnerabilities
Prevention: Network segmentation, regular updates, traffic monitoring

Q23- Assume you are designing an IoT-based smart home lighting system identify threats and suggest three mitigations?

A23- Threats: Unauthorized Control, data eavesdropping Mitigations:
1- Confidentiality:- Encrypt communications
2- Integrity:- Digitally Sign Firmware.
3- Availability:- Protect against DDos attacks.