Lab 0: The Crypto Workstation Setup

Course: CSC 4575: Applied Cryptography & Network Security **Estimated Time:** 45–60 Minutes
Prerequisites: Reliable Internet connection, ~30GB free disk space.

1. Overview & Objectives

In this course, we do not just *talk* about encryption; we build, break, and analyze it. To do this safely and effectively, you need a **clean, isolated Linux environment** capable of:

- **Raw Socket Manipulation:** Creating VPN tunnels (requires kernel access).
- **Packet Sniffing:** Analyzing traffic with Wireshark/Tcpdump.
- **Post-Quantum Compiling:** Building experimental libraries (LibOQS) from source.

Why a Local VM? While Google Colab is great for math, it cannot simulate network interfaces or modify kernel routes. We will use **Ubuntu Server 24.04 LTS (Headless)** accessed via **VS Code Remote-SSH**. This mimics a professional cloud/devops workflow.

---

2. Phase 1: Download Required Software

A. The Operating System (All Users)

Download the **Ubuntu Server 24.04 LTS** ISO file.

- **Link:** Get Ubuntu Server
- *Note:* It is roughly 2.0 GB.

B. The Hypervisor (Choose Your Path)

Path A: Windows Users & Intel-based Macs

You will use **VirtualBox**. It is free, mature, and widely supported.

- **Download:** VirtualBox 7.x
- **Windows User Warning:** You *may* need to enable "Virtualization Technology" (VT-x/AMD-V) in your computer's BIOS if the VM fails to start.

Path B: Apple Silicon Macs (M1 / M2 / M3 Chips)

VirtualBox **does not work** reliably on Apple Silicon. You must use **UTM**.

- **Download:** UTM for Mac
- **Architecture:** UTM runs the *ARM64* version of Ubuntu natively. It is blazing fast.

---

3. Phase 2: Virtual Machine Configuration

Configuration Specs (Minimum)

Regardless of your hypervisor, use these settings:

- **CPU:** 2 Cores
- **RAM:** 4096 MB (4 GB)
- **Disk Size:** 25 GB (Dynamically allocated is fine)
- **Network:** Bridged Adapter (Preferred) or NAT.

Path A: VirtualBox Setup (Windows/Intel Mac)

1. Open VirtualBox -> **New**.
2. **Name:** `CSC4575-Workstation`.
3. **ISO Image:** Select the Ubuntu 24.04 ISO you downloaded.
4. **Unattended Install:** *Skip* this if possible (check "Skip Unattended Installation") to have full control.
5. **Hardware:** Set 4GB RAM / 2 CPUs.
6. **Hard Disk:** Create a Virtual Hard Disk (25GB).
7. **Finish** and **Start**.

Path B: UTM Setup (Apple Silicon)

1. Open UTM -> **Create a New Virtual Machine**.
2. Select **Virtualize** (Not Emulate).
3. Select **Linux**.
4. **Boot Image:** Browse to your Ubuntu Server ISO.
5. **Hardware:** 4GB RAM, 2 Cores.
6. **Storage:** 25GB.
7. **Save** and run.

---

4. Phase 3: Installing Ubuntu

1. **Boot:** Follow the on-screen prompts (English -> Continue without updating -> Done).
2. **Network:** It should auto-detect an IP address (DHCP).
3. **Storage:** Use "Use an entire disk" (This is the *virtual* disk, not your real hard drive).
4. **Profile Setup:**

- **Your Name:** Student Name
- **Server Name:** `crypto-box`
- **Username:** `student` (or your preference)
- **Password:** Pick something memorable.

5. **SSH Setup (CRITICAL):**

- When asked "Install OpenSSH server?", check ☑ **Install OpenSSH server**.
- *Do not skip this. We need SSH for VS Code.*

6. **Featured Server Snaps:** Select None.
7. **Reboot:** When finished, select "Reboot Now". (If using VirtualBox, you may need to manually remove the ISO from the Optical Drive menu if it loops back to the installer).

---

5. Phase 4: The "Golden Script" Setup

Once you log in to your new VM (black screen, white text), you need to provision it with the course tools.

Step 1: Download the Setup Script In your VM terminal, type the following command carefully to download the script directly:

```
wget https://raw.githubusercontent.com/[YOUR_REPO]/setup_vm.sh
```

(Note: If the instructor has not hosted this yet, use `nano setup_vm.sh`, paste the script contents provided in the syllabus/LMS, and save with Ctrl+O, Enter, Ctrl+X).

Step 2: Make it Executable

```
chmod +x setup_vm.sh
```

Step 3: Run the Script

```
./setup_vm.sh
```

- Enter your password when prompted.
- **Wait:** This process takes 10–20 minutes. It compiles quantum-safe cryptographic libraries from source code.

Step 4: Verify When the script finishes, you should see `SETUP COMPLETE!`. Run this command to check if the Python environment works:

```
~/csc4575/venv/bin/python3 -c "import oqs; print('Post-Quantum Ready!')"
```

If it prints `Post-Quantum Ready!`, you are successful.

---

6. Phase 5: Connecting VS Code (The "Pro" Workflow)

We will not write code inside the clumsy VM window. We will use VS Code on your main computer to edit files *inside* the VM.

1. **On Your VM:** Type `ip addr` (or `ifconfig`). Look for your IP address (e.g., `192.168.1.50` or `10.0.2.15`).
2. **On Your Host (Windows/Mac):**

- Open **Visual Studio Code**.
- Install the Extension: **Remote - SSH** (by Microsoft).
- Click the blue "><" icon in the bottom-left corner.
- Select **Connect to Host...** -> **Add New SSH Host**.
- Type: `ssh student@<YOUR_VM_IP_ADDRESS>`
- Select the config file (default is fine).
- Click "Connect" (bottom right).
- Enter your VM password.

3. **Success:** The green bar in the bottom left should now say `SSH: <IP>`.
4. **Open Folder:** Go to File -> Open Folder... -> select `/home/student/csc4575`.

You are now ready for Lab 1.