Problem 1

Description (2 pts)

Since the merger of National Chiao Tung University and National Yang Ming University into National Yang Ming Chiao Tung University, there have been several changes in cybersecurity policies, particularly in the 2FA authentication of NYCU Portal and VPN settings. As a student of NYCU, you have keenly noticed these changes and wish to contribute to improving campus cybersecurity. Please complete the following two sub-questions

Requirements

1. As discussed in the class on March 7, the Two-Factor Authentication (2FA) method has specific weakness. Please describe the weakness and suggest possible solutions to address them.
2. How can the Yangming and Guangfu campuses design a VPN to securely share research resources?

Problem 1-1:

雙因子認證可能存在著一些弱點，包含SIM卡交換攻擊、裝置遺失...等，其中SIM卡交換攻擊是指攻擊者冒充受害者，向電信公司申請重新發行SIM卡並將受害者的門號綁定到自己的設備上；一旦新卡成功入手，所有簡訊驗證碼就會直接傳送到攻擊者的手機。另外，裝置遺失則是會導致攻擊者可以有直接取得您在裝置中獲得的驗證碼。如此一來，雙因子認證機制將受到這些威脅。然而，面對這些潛在的威脅，我們將提出一些可能解法:

(1) 為了降低門號被轉移後OTP遭攔截的風險，建議減少對簡訊驗證的依賴，改用行動驗證App，如:Google Authenticator、Microsoft Authenticator…等不受電信業者管控的多因素驗證方式，確保認證過程更安全。

(2) 在電信業者端，建議在遵守個資法等法規基準的前提下，強化SIM卡申辦與換發時的身分驗證程序。可透過更嚴謹的證件檢驗、通話或額外OTP驗證等多重機制，確保只有通過正當身分審核的人才能完成申請，防範攻擊者輕易冒名換卡取得門號。

(3) 針對裝置遺失而被突破雙因子認證這件事，我們除了要為手機設定生物辨識，並且將通知訊息內容點選不顯示於鎖屏狀態時，藉此保護自己訊息不會輕易地外流。至於，能否順利將遺失裝置取回則是端看個人運氣了。

Simple Examples:

小明劍魔是一個實況主，平常習慣使用簡訊驗證作為銀行帳戶的雙因子認證，然而身為實況主的他近期因為各種魔改影片爆紅，某日攻擊者A假冒小明劍魔的身分聲稱需要換發SIM卡，並憑藉著高超的偽裝技術以及對著電信公司怒吼我SIM卡壞掉了，詢問你們怎麼辦，你們還不快回答我！Looking my eyes! Tell me why why?之後成功換發了新的SIM卡，導致小明劍魔手機內的簡訊驗證碼全數被攔截。幸好，小明劍魔本身就有在手機上安裝了Google Authenticator作為備用認證方式，因此當攻擊者嘗試登入小明劍魔的銀行帳戶時，因無法提供來自Google Authenticator產生的動態密碼而受阻。事後小明劍魔對著螢幕後面的粉絲們大喊，你們以為雙因子認證就萬無一失了嗎，天天有人模仿我說我要換發新的SIM卡，要是只用簡訊做認證但沒有載行動驗證App作備用我要怎麼搞，回答我！

Problem 1-2:

我認為可以在陽明與光復校區之間部署一條使用IPsec的Site-to-Site VPN，並採用AES-256-GCM進行加密，再搭配SHA-2進行完整性驗證，以確保欲傳輸的資料在雙校區之間的傳輸是安全且穩定的。另外為了更有效地提高安全等級，我們也能使用憑證認證來替VPN Gateway簽發金鑰，避免預共享金鑰的管理風險。至於在網路架構上，我們可以將各自的內部網路分段，並透過防火牆設定存取控制清單，只允許特定VLAN或IP網段(白名單機制)進行互通。這樣可以也可以有效防止兩邊所有內網都毫無限制地相互開放，同時也能根據不同需求針對性地授權。若校方希望進一步強化使用者端的連線管控，也可以在雙校區部署認證伺服器（如LDAP），並與VPN Gateway串接多因子認證機制，讓每次登入都必須經過額外驗證，降低攻擊者只憑盜用帳密就能冒名登入的風險。

Simple Examples:

阿志是一位在光復校區進行醫學影像分析的資科工所的研究生，他需要從陽明校區的醫學院中取得一批患者腦部醫學影像資料進行分析。若兩個校區之間已經部署了採用 **IPsec Site-to-Site VPN**的網路連線，並透過AES-256-GCM與SHA-2提供高等級的加密和完整性驗證，這位研究生只需在光復校區內部登入相關系統(如: NAS)，就能像在陽明校區一樣直接並安全地存取那些MRI影像。若是有人想要攔截或竊聽中途

傳輸的MRI影像檔，會因為整個傳輸過程已加密並具備完整性驗證機制，使得攻擊者最終只能得到無法解讀的加密訊息，也難以竄改任何數據。

Problem 2:

本次實驗示範了如何使用字典攻擊來嘗試破解SHA-1雜湊值，首先，我們先將字串利用hashlib.sha1轉成雜湊值，再跟目標雜湊進行比對，若兩兩相同，則代表破解成功，一開始我們成功破解了Easy、Medium兩個，然而在添加salt的雜湊卻遇到了一些困難，後來我們依循著定義發現: 由於在未加salt時，只要password.txt裡包含對應的原始密碼，程式便能將其與目標雜湊配對，但是呢在加salt的case中，我們必須先將salt進行雜湊值的破解，接著，我們先是把salt和字典裡的密碼組合計算新的雜湊，再來比對目標雜湊值是否吻合即可。而在過程中，我們發現若是添加的salt較為常見或是簡易，又或者是字典檔案包含更多組合如10^10種組合以上...等，攻擊者仍有機會找出真正的密碼。換言之，在實際應用時，我們應加強salt的複雜度，或是採用更複雜的加密機制，以有效提高密碼安全性。**執行結果如下，我們可發現Easy Hash對應的是scorption (花費302次嘗試)，而Medium Hash對應到的是wh00sh (花費939438次嘗試)，至於我們先破解出的salt是redbull (花費2785次嘗試)，而Leet hacker hash則是對應到puppy(花費2854次嘗試)。**

```
xe c:/Users/蘇柏叡/Desktop/313553024/problem2/main.py
Hash: 884950a05fe822dddee8030304783e21cdc2b246
Password: scorpion
Took 302 attempts to crack message.

Hash: 9b467cbabe4b44ce7f34332acc1aa7305d4ac2ba
Password: wh00sh
Took 939438 attempts to crack message.

Salt found: redbull
Salt took 2785 attempts to find.

Hash: 9d6b628c1f81b4795c0266c0f12123c1e09a7ad3
Password: puppy
Took 2854 attempts to crack message.
```

Problem 3:

我們首先利用個人學生證號碼**(313553024)**來生成初始雜湊值，並且我們將這個雜湊值作為前一區塊的雜湊值，並且進行比對與檢核是否能在不使用nonce的情況下就已經符合某個特定目標字首；若無法符合則再使用nonce進行Brute-Force Searching，以求使新的區塊雜湊符合這個目標字首。至於整個流程我們會透過SHA-256演算法來計算雜湊。

因此，若是我們想要取得以指定字串開頭的結果，則必須不斷嘗試改變nonce，**直到某一回合成功符合目標時，即代表挖礦成功**，我們會於Log file中記錄下成功的雜湊值與nonce；反之，若是在我們已經將nonce遍歷至最大的範圍，卻依然找不到符合條件的雜湊值，則表示該回合挖礦失敗。因此，不論該回合挖礦成功或者失敗，我們本程式都將會針對挖礦結果於Log file中進行描述。如下圖所示，我們能發現原題目設定至少挖礦出6碼，而我們在本次實驗中，我們將長度改設定為8。此時，我們將發現在Round2結束時，已經正確預測出下一回合的數字就是3，因此，在Round3時是出現without nonce的訊息，另外我們在結束第七個回合往第八個回合挖礦時，總工耗費逾107分鐘，最終因為遍歷至nonce最大範圍卻找不到符合條件之雜湊值，因此在Round8時出現了EROR的標籤，換言之，在本次挖礦最大長度設定為8的實驗中，我們成功了7個回合，並且在第二個回合時就已經成功預測出學號第三碼。

logger.log

```
2025/03/26 23:09:13 [INFO] [preImage] 11d4750ccb6ea747c8697d696f93169cc4bc4bccc43bfa16eaf50ac658cdf0f9
2025/03/26 23:09:13 [INFO] [Round 1 with nonce 00000013] 391ee0e7e8cc79b2ed16335b52c72897dfbb0078f9a8dc0a6dad27537675ce6d
2025/03/26 23:09:13 [INFO] [Round 2 with nonce 00000343] 3138660fc2d0fa0873c8265b475633ae81dbfd96b512d8ee72b7e5cc7fc5c35b
2025/03/26 23:09:13 [INFO] [Round 3 without nonce] 3138660fc2d0fa0873c8265b475633ae81dbfd96b512d8ee72b7e5cc7fc5c35b
2025/03/26 23:09:13 [INFO] [Round 4 with nonce 00009b8a] 3135b56dd23cbcbabc04e5c834d8bc719778095007a77a5ef63b8060b3f48154
2025/03/26 23:09:13 [INFO] [Round 5 with nonce 0000477e] 31355c46103f63ec283d44c71f77e0603f53f42f2d2f031194c1a81707063652
2025/03/26 23:09:22 [INFO] [Round 6 with nonce 006f8e7f] 3135534960829f352b7c95e6ebd95fc64975d445515479a2754ff7dafe668b58
2025/03/26 23:12:59 [INFO] [Round 7 with nonce 09ad3b94] 3135530e0c6b7b169374978a47eb070c905dbf5dbb87d9c63fd0c0781f5922e5
2025/03/27 02:00:12 [EROR] [Round 8] not found with running out of nonce
```