# Quiz 2

(Deadline March 21, 2025)

## 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?

**Evaluation**

1. Briefly describe the answer: **1 point each sub-question**.
2. Give a simple example: **0.1 point extra credit** each sub-problem.

## Problem 2

**Description** (1.5 pts)

You are one of the world's top hackers. One day, you receive three messages and decide to break them by using a brute-force method to crack their SHA-1 hashes.

### Requirements

1. **Input:** Write a program that breaks SHA-1 hashes in a **brute force** manner using this password list:
   - Password list URL: `https://raw.githubusercontent.com/danielmiessler/SecLists/master/Passwords/Common-Credentials/10-million-password-list-top-1000000.txt`
   - **There is no need to include the password list in the uploaded files.**
2. **Output Format:** For each hash, your program must output:

```
Hash: db3ae03df555104cd021c6308d5d11cfa40aac41
Password: hotmom
Took 30568 attempts to crack message.
```

3. **Target Hashes:** Break these SHA-1 hashes:
   a) Easy hash:
   `884950a05fe822dddee8030304783e21cdc2b246`
   b) Medium hash:
   `9b467cbabe4b44ce7f34332acc1aa7305d4ac2ba`
   c) Leet hacker hash:
   `9d6b628c1f81b4795c0266c0f12123c1e09a7ad3`
   *Hint: Use this salt term:* *`dfc3e4f0b9b5fb047e9be9fb89016f290d2abb06`*
   *Format: sha1(salt + password)*
4. **Documentation:** Briefly describe your program (how to run your program and what you do in the program), and take a **screenshot of the results** and include them in your PDF file.

### Evaluation

- Successful decryption of messages: **1 point for all**
- Comprehensive description: **0.5 points**
- The absence of a description or screenshots will result in a score of **0 points** for this problem.

## Problem 3

### Description (6.5 pts)

The blockchain technology has gained significant traction in recent years, with mining emerging as a fundamental component of this technology. In this task, you are required to demonstrate advanced mining skills by utilizing your student ID as a seed to mine a personalized ”Student ID Block.”

### Requirements

1. Initialization
   (a) **Use your student ID** as the seed and hash it to generate the initial preimage. For demonstration purposes, I use `123456789` as the sample student ID.

   ```
   preImage := sha256("123456789")
   ```

   (Your student ID should be treated as a string when hashing.)

   (b) Determine your **starting block** based on the `preImage`.

   - If the `preImage` is `0xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx`, your starting block is the **first block** (The first digit of the `preImage` does **not** match the first digit of the student ID).
   - If the `preImage` is `1axxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx`, your starting block is the **second block** (The first digit of the `preImage` matches the first digit of the student ID, but the second digit does not match the second digit of the student ID).
   - If the `preImage` is `12bxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx`, your starting block is the **third block** (The first and second digits of the `preImage` match the first and second digits of the student ID, but the third digit does not match the third digit of the student ID).

2. Mining
   (a) For each block (round), you must combine **the previous block's hash value** with a **nonce value** and compute the hash repeatedly until you find a valid target hash for that block (round).

   - **First block (round):** The target hash should start with `1xxxxxxxx`....
   - **Second block (round):** The target hash should start with `12xxxxxxx`...

   (b) The nonce starts at `0x00000000` and increases incrementally. Each candidate hash for the block is calculated as follows:

   ```
   blockHash := sha256(previousBlockHash + nonce)

   // e.g.
   hash := sha256("
   ↪ xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx" + "
   ↪ yyyyyyyy")
   ```

The nonce value ranges from `0x00000000` to `0xffffffff`.

(Both prepreviousBlockHash and nonce are hex digits represented as string type.)

3. Log Record

   You must output the process as a log file named `logger.log`. The log should follow this format:

```
2025/03/10 13:20:00 [INFO] [preImage] 1xxxxxxxxx...
2025/03/10 13:20:00 [INFO] [Round 1 without nonce] 1xxxxxxxxx...
2025/03/10 13:20:10 [INFO] [Round 2 with nonce deadbeef] 12xxxxxxxx...
2025/03/10 13:20:10 [INFO] [Round 3 with nonce badc0de5] 123xxxxxxx...
2025/03/10 13:21:10 [INFO] [Round 4 with nonce c0ffee69] 1234xxxxxx...
2025/03/10 13:25:45 [INFO] [Round 5 with nonce d15ea5e5] 12345xxxxx...
2025/03/10 13:30:00 [INFO] [Round 6 with nonce ba5eba11] 1234567xxx...
2025/03/10 13:31:00 [INFO] [Round 7 without nonce] 1234567xxx...
2025/03/10 13:40:00 [EROR] [Round 8] not found with running out of nonce
```

4. Briefly describe your program (how to run your program and what you do in the program). **This problem is no need to take the screenshot, but it is required to upload the log file.**

### Evaluation

- Successfully find the first 6 digits of your student ID in the hash: **1 point per digit**.
- For each additional matching digit, earn **0.1 point extra credit** per digit.
- If you exhaust the entire nonce range without finding the first 6 digits of the target hash, you will still receive **6 points**.
- Description: **0.5 points**.
- If the submission ZIP file does not include the log file, you will **receive 0 points** for this problem.
- **Warning:** If your log record is falsified, you will **receive 0 points for this assignment and an additional 5 points will be deducted from your semester score**.

## Reference

### Problem 3's Log Format

- For go:

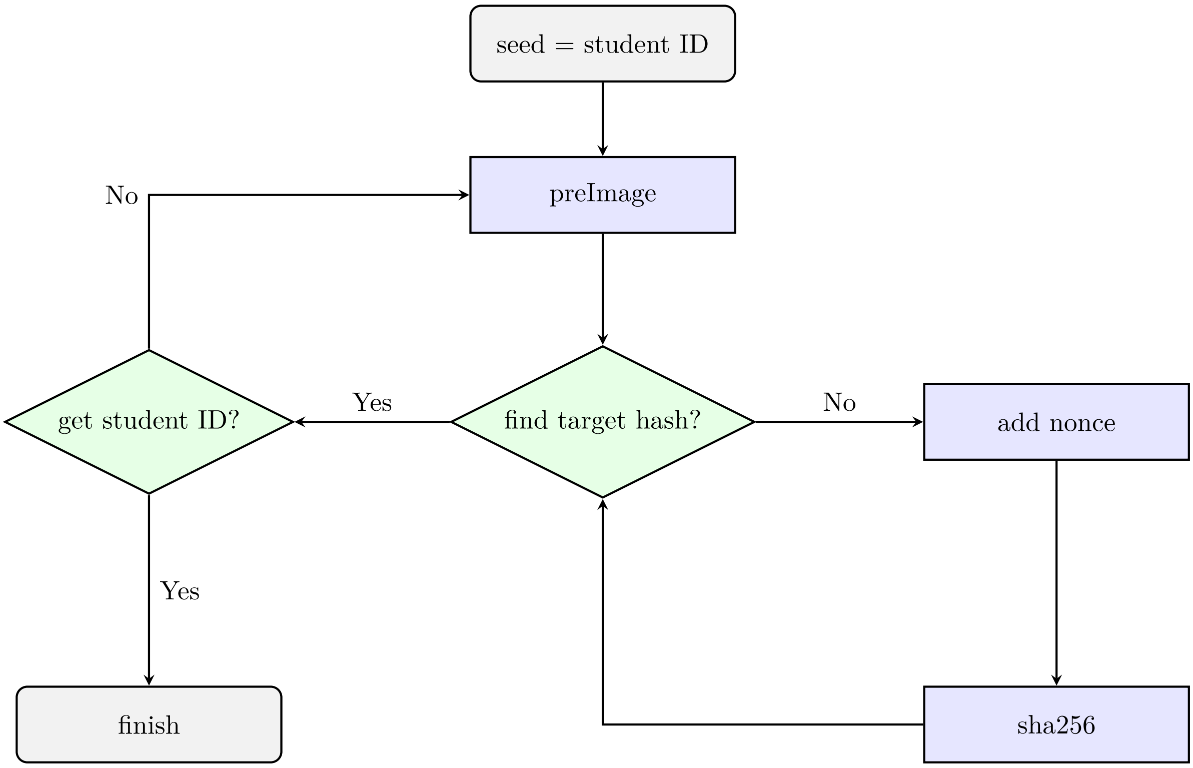
  logger-go: `https://github.com/Alonza0314/logger-go#logger-in-file`

- For python:

  Do it yourself!

  (Caution: the error tag is `[EROR]`, not `[ERROR]`.)

### Problem 3's Diagram



## Submission Guidelines

1. Upload a **single ZIP file** named `<student_id>.zip` to the new e3 platform, with the following structure:

    - `<student_id>.zip`
        - `<student_id>.pdf`
        - `problem2`
            - `main.go` or `main.py`
            - (additional codes...)
        - `problem3`
            - `main.go` or `main.py`
            - `logger.log`
            - (additional codes...)

    (You may include any additional code files, as long as they follow this file structure.)

    (**Incorrect file structure or incorrect file names will result in a 2 points deduction.**)

2. Present your solutions in **numerical order**, clearly labeling each problem.

## Grading Policy

1. This quiz comprises three problems:

    - Problem 1: 2 points (1 point for each sub-question, 0.1 point extra credit for each example)
    - Problem 2: 1.5 points (1 points for cracking all messages, 0.5 points for description)
    - Problem 3: 6.5 points (6 points for mining, 0.5 points for description)

2. **Late Submission Penalty:** A penalty of **0.5 points per day** will be applied for late submissions, up to a maximum of 20 days. Beyond 20 days, late submissions will be assigned zero.

3. All quizzes are **mandatory**. Failure to submit any quiz will result in an automatic failing grade for the course.