Lab 9: Password Cracking

Lan-Da Van and Chun-Jen Tsai

Department of Computer Science

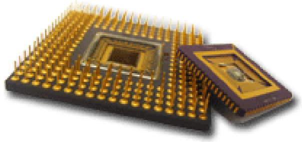
National Yang Ming Chiao Tung University

Taiwan, R.O.C.

Fall, 2024

Lab 9: Password Cracking

- In this lab, you will design a circuit to guess a 9-digit password scrambled with the SHA-256 hashing algorithm.
  - The password is composed of 9 decimal digits coded in ASCII codes.
  - The SHA-256 hash code of the password will be given to you.
  - The circuit must crack the password and show it on the LCD module. The time taken to crack the password must also be displayed on the LCD module.
- The lab file submission deadline is on 11/18 by 6:00pm.

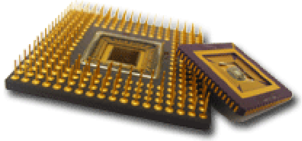


Introduction to Password System

- The passwords of a login system are stored in a user account file in “encrypted” format.
  - The encryption algorithm for passwords is not reversible.
  - You cannot decrypt the encrypted password and restore the original password.
  - For Linux, the password file is under /etc/shadow.

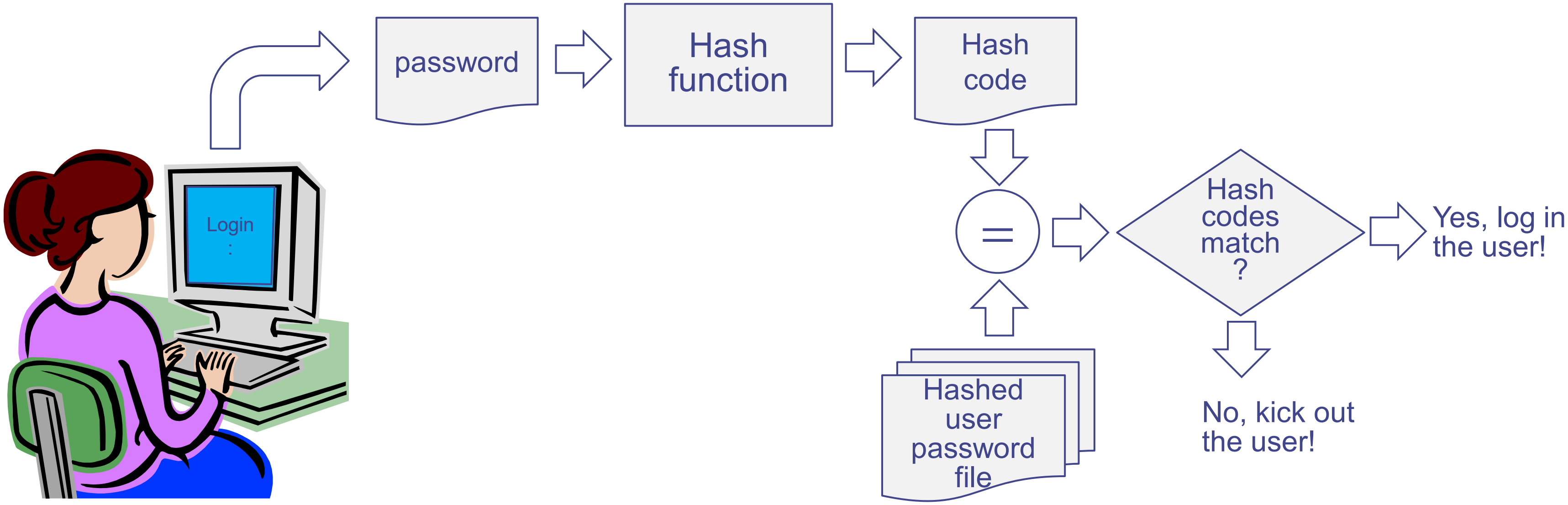
```
user1:$6$6155bfdd22808014a1e2ccd198IN3zshkbyWjrrYVmrd.cM/xx
7YF2/yNaw4v9xJuYUq2QkskRd6CRKb0.G8m1mFLWCr4v.:17221:0:99999
:7:::
user2:$6$7fbf8a8b90bcbb2ba650cc8b0714b739ByB51L23WwxWEE790j
rs8jVPmKcXqzO19yW2NWn2L3LK/ZX/x0j0eHDwp0S1M90:17444:0:99999
:7:::
```

The hash code of user2’s password!



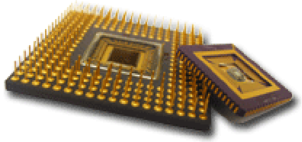
Hash Functions for Passwords

- There are many one-way hash functions for passwords: MD5, Blowfish, SHA-256, and SHA-512.
- Ideally, two different passwords will be transformed into two different hash codes by the hash functions:



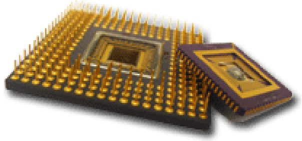
SHA256 Hash Function

- SHA256 is a popular hash function that converts any file into a 256-bit hash code.
- There are many applications for SHA256.
  - Data integrity protection
  - Digital signature verification
  - Password hashing
  - SSL handshake (HTTPS)
  - Block chain (Bitcoins)
- SHA-256 is currently considered a secure hash function that is resistant to collision attacks.



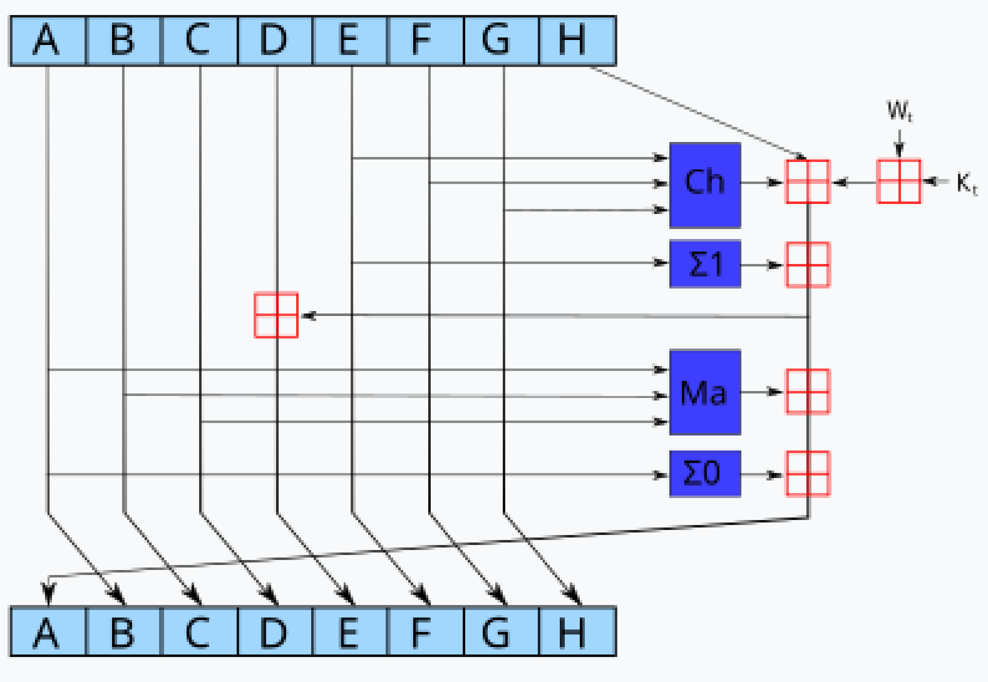
Algorithm of SHA256 (1/2)

- SHA256 processes a variable-length message into a fixed-length output of 256 bits.
- The input message is broken up into chunks of 512-bit blocks (sixteen 32-bit words); the message is padded so that its length is divisible by 512.
- The last 8 bytes of the last 512-bit block contains the bit length of the original message.
- SHA256 divides the hash code of 256-bit into eight 32-bit words; and performs complex XOR, AND, OR, NOT, choice, majority and rotation operations using the 512-bit message blocks as the input.



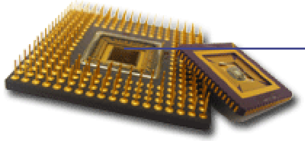
Algorithm of SHA256 (2/2)

- One SHA256 operation†:



$$\mathrm{Ch}(E,F,G) = (E \land F) \oplus (\neg E \land G)$$
$$\mathrm{Ma}(A,B,C) = (A \land B) \oplus (A \land C) \oplus (B \land C)$$
$$\Sigma_0(A) = (A \ggg 2) \oplus (A \ggg 13) \oplus (A \ggg 22)$$
$$\Sigma_1(E) = (E \ggg 6) \oplus (E \ggg 11) \oplus (E \ggg 25)$$

⊞ means addition modulo 2^{32}.



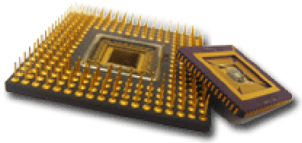
† https://en.wikipedia.org/wiki/SHA-2

Comments on Parallel Computation

- In order to crack the code as fast as possible, you should try to instantiate multiple copies of SHA256 cracking unit and compute the hash code in parallel.

- For example, if you have 10 instances of SHA256 cracking unit , each circuit only needs to compute 100,000,000 hash codes.
  - As soon as one of the circuits finds a match, the cracking operations can be terminated.

- Your grade will be evaluated based on the cracking speed of your circuit.



What You have to Do for Lab 9

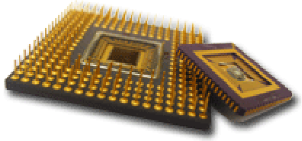
- You must write an SHA256 cracking circuit using Verilog and implement it on the Arty board.
- In your circuit, the password hash code shall be declared as follows:

```
reg [255:0] passwd_hash = 256'hf120bb5698d520c5691b6d603a00bfd662d13bf177a04
571f9d10c0745dfa2a5
```

- Once the user presses BTN3, your circuit will crack the password and show it on the LCD module.

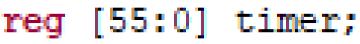
P	w	d	:	x	x	x	x	x	x	x	x	x			
T	:	t	t	t	t	t	t	t	t	t	t	t	t	t	t

  - x : nine digit of password
  - t : total clock cycle you use to crack the password (HEX)
  - Note: it might takes modern PC 16.7 minutes to crack it!



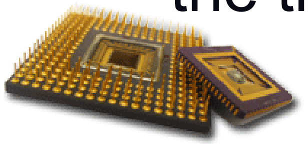
Timer Regulation

- You will need to design a timer as follow:
  - Your timer should add 1 every 100Mz clock cycle.
  - As soon as your circuit detects BTN3 logic signal is high, you should start your timer immediately.
  - You can only stop your timer after you finish cracking the password.
- You should show your timing information on LCD.
  - Show timer [55:0] with hexadecimal on LCD display.



```
reg [55:0] timer;
```

  - For example, If you spend 1000 cycles cracking the password, you should show: 00000000003E8 on LCD .
  - Lock your timer if you reach the maximum number !
- If your timer or LCD display does not meet the above requirements, you will be ineligible to participate in the timing ranking.



Lab 9 Grading

- Functional Correctness (3~5 hidden testcases) – 50%
  - TA will put the hidden testcases into your “passwd_hash” registers to test your design correctness.
- Timing Check – 20%
  - If all the testcases result in WNS > 0 in your design, you will pass this part.
  - If you fail any testcases, you will lose these points.
- Speed Ranking – 20%
  - The faster you crack, the higher score you will get. The ranked result will be divided into at most 5 level, the point you get will depend on which level you are.
  - However, if your timer or LCD display format does not meet the requirements, you will lose all the points in this part.
  - If you fail any testcases, you will lose these points.
- Question – 10%