Lab 8: Password Cracking

National Chiao Tung University
Chun-Jen Tsai
11/30/2018

Lab 8: Password Cracking

- In this lab, you will design a circuit to guess an 8-digit password scrambled with the MD5 hashing algorithm
  - The password is composed of eight decimal digits coded in ASCII codes
  - The MD5 hash code of the password will be given to you
  - Your circuit must crack it, and display the original password and the time it takes for you to crack the password on the LCD module
- The deadline of the lab is on 12/11

Introduction to Password System

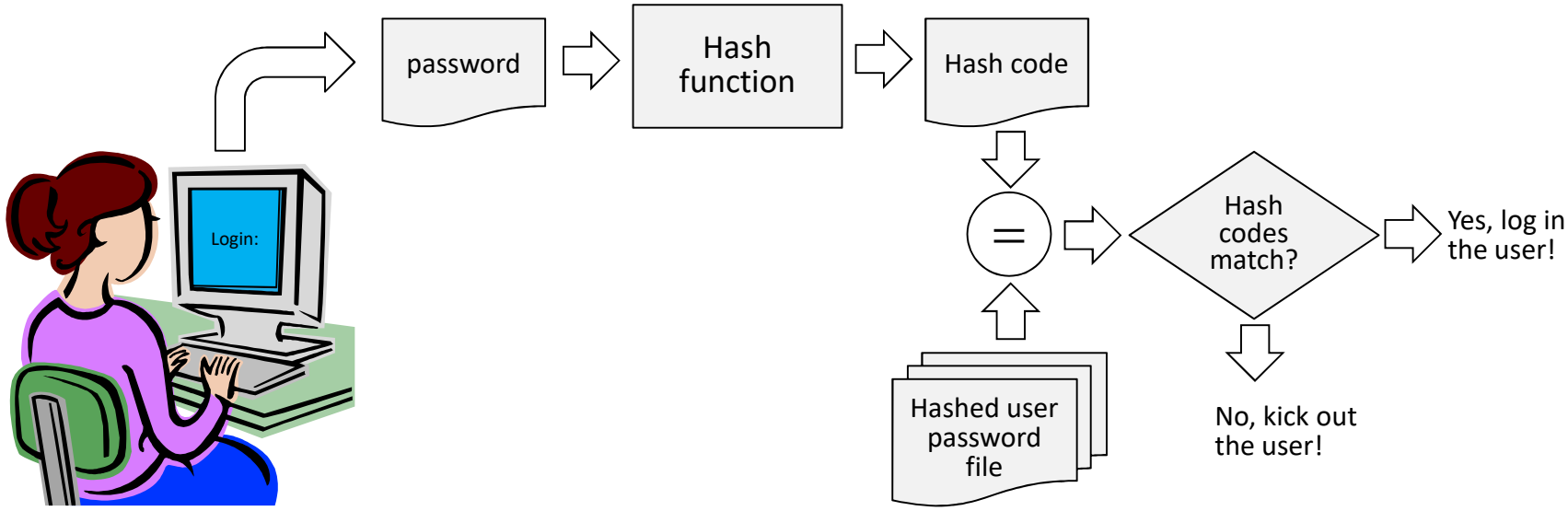
- The passwords of a login system are stored in a user account file in “encrypted” format
  - The encryption algorithm for passwords is not reversible
  - You cannot decrypt the encrypted password and restore the original password
  - For Linux, the password file is under /etc/shadow

```
user1:$6$6155bfdd22808014a1e2ccd198IN3zshkbyWjrrYVmrd.cM/xx
7YF2/yNaw4v9xJuYUq2QkskRd6CRKb0.G8m1mFLWCr4v.:17221:0:99999
:7:::
user2:$6$7fbf8a8b90bcbb2ba650cc8b0714b739ByB5lL23WwxWEE790j
rs8jVPmKcXqzO19yW2NWn2L3LK/ZX/x0j0eHDwp0SlM90:17444:0:99999
: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:



The MD5 Hash Function

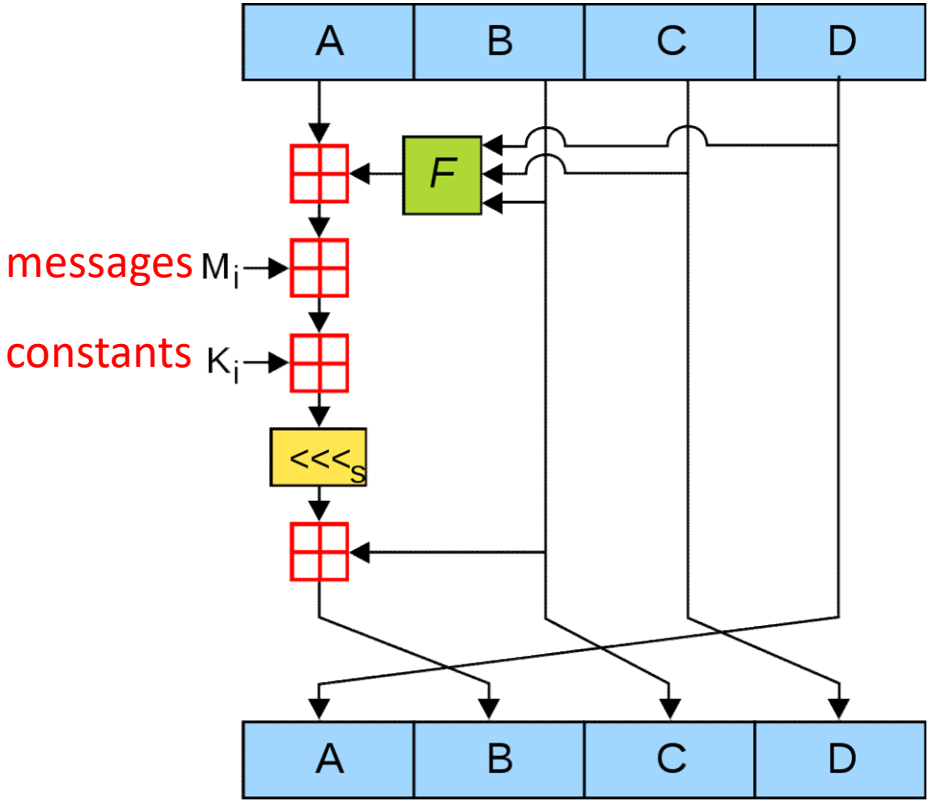
- Message Digest 5 (MD5) is a popular hash function that convert any file into a 128-bit hash code
- MD5 was developed by Ronald Rivest in 1991, and became a standard known as IETF RFC-1321
- There are many applications for MD5
  - Compute a checksum of a file to make sure that it is not modified
  - Scramble passwords so that they can be distributed securely
- MD5 has serious vulnerability and is considered an insecure hash function (see RFC-6151, 2011)

The Algorithm of MD5 (1/2)

- MD5 processes a variable-length message into a fixed-length output of 128 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
- MD5 divide the has code of 128-bit into four 32-bit words, A, B, C, and D; and perform complex XOR, AND, OR, NOT, modular, and rotation operations using the 512-bit message blocks as the input

The Algorithm of MD5 (2/2)

- One MD5 operation†:



$$F_1(B, C, D) = (B \,\&\, C) \mid (\sim B \,\&\, D)$$
$$F_2(B, C, D) = (B \,\&\, D) \mid (C \,\&\, \sim D)$$
$$F_3(B, C, D) = B \wedge C \wedge D$$
$$F_4(B, C, D) = C \wedge (B \mid \sim D)$$

⊞ means addition modulo 2^{32}

$<<<_s$ means left rotate

† https://en.wikipedia.org/wiki/MD5#Algorithm

The Sample C-Model for Lab 8

- In this lab, a sample C model for the MD5 algorithm is available on the E3 website
  - The `md5()` function only computes the MD5 hash code of a message that has less than 55 characters
  - Our password is composed of 8 numbers in ASCII code
- A brute-force cracker code to guess an MD5 password is shown as follows:

```
uint8_t pattern[9], hash[16];
uint8_t passwd_hash[16] =
    { 0xE8, 0xCD, 0x09, 0x53, 0xAB, 0xDF, 0xDE, 0x43,
      0x3D, 0xFE, 0xC7, 0xFA, 0xA7, 0x0D, 0xF7, 0xF6 };

for (idx = 0; idx < 100000000; idx++) {
    sprintf(pattern, "%08d", idx);
    md5(pattern, 8, hash);
    if (!strncmp(hash, passwd_hash, 16)) break;
}

if (idx < 100000000) printf("The password is %s.\n", pattern);
```

What You have to Do for Lab 8

- You must rewrite the md5() function and the cracker code using Verilog and implement it on the Arty board
- In your circuit, the password hash code shall be declared as follows:

```
reg [0:127] passwd_hash = 128'hE8CD0953ABDFDE433DFEC7FAA70DF7F6;
```

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

```
Passwd: xxxxxxxx
Time: yyyyyyy ms
```

xxxxxxxx is the 8-digit password

yyyyyyy is the computation time in milliseconds (in decimal number)

  - Note: it takes an Intel i7-4770 PC 27 seconds to crack it!

Comments on Parallel Computation

- In order to crack the code as fast as possible, you should try to instantiate multiple copies of `md5()` circuit blocks and compute the hash code in parallel

- For example, if you have 10 instances of `md5()`, each circuit only have to compute 10,000,000 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