Class 2: Password security. v2

Note Title

Main objective for today: work out how long it takes to crack different types of passwords

[Go over syllabus.]

[Go over discussion questions from the lab]

① Cryptographic hash functions

A **hash function** is a computational procedure that creates a fingerprint of some data (e.g. a file, or a password). The fingerprint, or **hash**, has a fixed length.

(also called **digest**, or **checksum**)

example given any number, we output the last 2 digits. e.g. '157447' becomes '47'. This is a hash function, transforming a number of any length into a 2-digit hash.

A **collision** is when 2 different inputs hash to the same value

exercise: give an example of a collision for the above hash function.

A <u>cryptographic hash function</u> is a hash function for which no-one knows how to efficiently create collisions. (For a decent-sized hash, random guessing is <u>not</u> efficient — it would take billions of years to find a collision.)

<u>Famous examples</u>

see below for explanation

- MD5 – creates a 128-bit hash
  - published 1992
  - has been cracked and is not secure – collisions are known! See wikipedia page.

- SHA-256 – creates a 256-bit hash
  - published 2001
  - some weaknesses known, but still considered secure. See wikipedia.

Demo: Use web search to find an online SHA-256 calculator. Compute the SHA-256 hash of "Dickinson". How about "dickinson"?

The length of the hash is usually measured in <u>bits</u> or <u>bytes</u>. <u>Bits</u> are the 1s and 0s actually stored in a computer.

A <u>byte</u> is (roughly) the same thing as a character (like "a", "6", or "5" or "?") and it takes 8 bits to represent one byte. So a SHA-256 hash is 256 bits or 32 bytes or 32 characters. (or 64 hexadecimal digits, but don't worry if you don't know what this is).

② What are cryptographic hash functions used for?

Many, many things. One application is to verify passwords without storing them:

- computer does not store your password
- instead, stores (e.g. SHA-256) hash of your password
- when you enter your password, the computer calculates the hash and compares with stored version.

③ Our attack model for password cracking + number of guesses needed

We assume a malicious hacker has obtained access to a system's file of password hashes. The hacker guesses passwords randomly, computing the hash of each one to see if it matches a hash on the system.

Number of guesses required, on average, is:

$$\text{avg num guesses} = \frac{1}{\text{num hashes} \times \text{prob of single correct guess}}$$

e.g. If password file has 50 hashes, and chance of correct guess is one-in-a-million, avg guesses needed is

$$1 \Big/ 50 \times \frac{1}{10^6} = \frac{10^6}{50} = 20{,}000.$$

Exercise: How many guesses are needed, on average, to crack one password from a file of 1000 hashes, if each guess succeeds with probability one-billionth?

Math help: 10^6 means $\underbrace{10 \times 10 \times \ldots \times 10}_{\text{six times}}$
or 1 with 6 zeroes after it
i.e. 1,000,000 i.e. 1 million.

(in typed text, sometimes written 10^6)

(4) Chance of success with a guess

Chance of success is just

$$\frac{1}{\text{num possibilities}}.$$

e.g. (a) For a password selected at random from a dictionary of 50,000 words, chance of success is $\frac{1}{50{,}000}$

(b) for a 4-digit PIN (i.e. a numeric password), number of possibilities is

$$10 \times 10 \times 10 \times 10 = 10^4 = 10{,}000$$

↗ num possibilities for each digit

Chance of success is $\frac{1}{10{,}000}$.

(c) For a 6-letter alphabetic password (all lowercase), num possibilities is

$$26 \times 26 \times \ldots \times 26 = 26^6 \approx 300 \text{ million}.$$
$$= 3 \times 10^8$$

Chance of success is about $1/3\times10^8$

(5) How long to crack?

A modern desktop computer can compute (very roughly) 100,000 hashes per second.

Average time to crack (in seconds) = $\dfrac{\text{avg num guesses needed}}{\text{num hashes per second}}$

Exercise: How long to crack a 6-letter lowercase password?

(6) If time, discussion about this 100,000 hashes per second number:

- specialized machinery can be much faster, e.g. 'DES cracker' from 1998 did 10^{11}/sec.
- used multiple machines is faster
- how does this compare with the numbers from our lab? If different, why?

(7) If time, discuss some interesting results from the optional reading.

Source & further reading: Some of these notes are based on Smith, §6.2–6.4. That is a good place for additional information.