Slinging the Hash (Function)

What are hash functions good for?

- Hashing values to insert into arrays: hash tables
- Converting long strings into shorter strings in an irreversible way: one-way function
  - Turn a 200 KB document into a 16 byte hash value
  - Turn *any* document into a 16 byte hash value
- One-way functions have lots of uses…
What is a one-way function?

- One-way functions convert inputs into outputs in such a way that
  - It’s “difficult” to find the original input given the output
  - It’s “difficult” to find any input that gives a particular output
- Commonly used one-way functions include
  - MD5
  - SHA-1 (Secure Hash Algorithm)
- Example:
  - \texttt{md5(“MD5 test”)=cc60dda980ccbf65540520703d91b27c}
  - “Difficult” to find any other string with the same MD5 hash
  - “Difficult” to get “MD5 test” from its hash value

How difficult is “difficult”?

- For a good hash function, we must use brute force to find corresponding values
- MD5 has 16 byte hash values (128 bits)
  - \(2^{128}\) different possible hash values
  - Must try, on average \(2^{127}\) different strings to find one with a given hash value!
    - Generates half of \(2^{128}\) different hash values
- This means that it’s darn near impossible to guess a string with a matching hash value
- Why is this so useful?
**Passwords in Unix**

- **How is your password stored?**
  - Stored in a file readable by any user!
  - How can we do this without giving away your password?
- **Solution: use hash values**
  - Store the hash of your password rather than the password itself
  - Each time you log in, compare hash(password) with the value stored in the file
  - If they match, you’re granted access
- **Nobody can figure out your password from the value stored online!**
  - Makes it difficult to impersonate you
  - There is a catch, though…

**Guessing Unix passwords**

- It’s very hard to find any password that corresponds to 0905a74d4251fc1b2ec794738a81f324
- However, we can try lots of different possibilities
  - All of the words in the dictionary
  - Names (people, places, etc.)
  - Combinations of the above
  - Slight modifications
    - “O” -> “0”
    - “I” -> “1”
    - “E” -> “3”
- Doing so would turn up my password: “zachary”
- We could even set up a list and check everyone’s password against the list
  - Doing this finds lots of matches!
  - Moral: choose a good password that isn’t a dictionary word
What else are hash functions good for?

- Suppose I want to prove I have a document at a certain time without revealing its contents
- Example: exam solutions
  - Students want to be sure the solutions exist before they take the exam
  - Professor doesn’t want to give out the solution in advance
- Example: patent
  - Patent should remain secret until granted
  - Interested parties may want proof that the patent really was filed on the claimed date
- This can be done with hash functions!

Proving you have a document...

- Step 1: hash the document
- Step 2: hand the resulting hash to anyone who wants it
  - Hash gives no clue as to content of document
  - When the document is revealed, the hash can be recomputed
    - If they match, the document must have existed when you first got the hash!
- This is used all the time to verify document creation date while not revealing content
Signing a document

- Suppose you have a multi-page contract (in electronic form): how can you ensure that each page is valid?
  - Sign each page in ink?
  - Sign only the last page?
- Different solution: generate a hash value, and sign it
  - If hash value is signed, document must be signed too
  - Hash value is much smaller and easier to store
- There are techniques for signing documents that we won’t cover in class (ask me during office hours if you’re interested)

Interested in computer security?

- There will be an undergrad class in computer security in Spring 2003
- Prerequisite is CMPS 111
- How can you take this class?
  - Take 12C in Spring 2002
  - Take 101 in Fall 2002
  - Take 111 in Winter 2003
  - Now, you’re ready for computer security
- Topics include
  - Encryption'
  - Authentication
  - Computer security
  - Viruses & malicious code