Cryptographic Hash Function

Stefan Lucks

Bauhaus-Universität Weimar

Summer 2019

Before taking this course, you

must pass an introduction to Cryptography, first.

This could, e.g., be “Introduction to Modern Cryptography” (Bachelor/Master), or some crypto course from your previous Bachelors’ program, . . .

- You must not do both the “Introduction to Modern Cryptography” and this course, in the same semester.
- You are not admitted to “Introduction to Modern Cryptography”, but you can do this this course, if you did study cryptography before.
- You must not do this course, but you are admitted to “Introduction to Modern Cryptography”, if you did not study cryptography before.
- Feel free to submit a copy of your transcripts of records from previous studies to my secretary, if you are not sure what counts as introduction to cryptography. I’ll immediately check to where you are admitted. Before the exam, we’ll require such a copy, anyway.

Some Information for You

- I'll switch between using the beamer and the blackboard.
- When I am using the blackboard, you should make notes.
- Information on the WWW:
  - Slides
  - Homework problems, and reading tasks
  - Code-example
  - Links to download further information and tools.

Motivation

- **Cryptographic hash functions** are denoted the
  - “Workhorses”,
  - “Swiss army knife”, and
  - “Duct tape”

  of cryptography.
- Since 2004: Many hash functions used in practice under attack.
- 2007–2012: Competition for a new hash standard
  (Our Submission Skein has been one of five finalists)

- We will also deal with Password-Hashing.
  Passwords are “everywhere”, to identify humans to systems, to unlock cryptographic keys, . . .
- March 2014 – June 2015: Password Hash Competition (PHC)
  (Special recognition for our Submission Catena submission)

Learning Goals

Learning goals:

- Methods from theoretical and practical cryptography
- Cryptanalysis (i.e., attacks)
- Security proofs
- Understanding of the

state of current research
in an important sub-field of cryptography.

1: Introduction

meat mincing machine (“Fleischwolf”)

What is / What Does a Hash Function?

- A HF H maps a potentially long message M to small (n-bit) outputs $H(M)$.
- Consider $H(M)$ as a "fingerprint" to identify M.
- Collisions are inputs $\mathbf{x} \neq \mathbf{y}$ with $h(\mathbf{x}) = h(\mathbf{y})$.

Collisions are **bad**. If it is feasible to find a collision, the hash function is insecure.

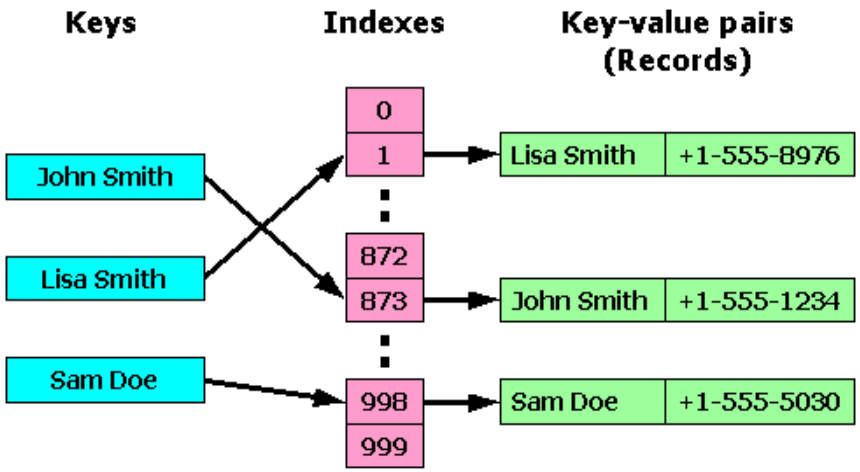
- Beyond security, you want a hash function to be **fast**. (Except when you are doing password hashing . . .)

Example:

$h : (\{0,1\}^c)^* \to \{0,1\}^c$, $h(x_1, \ldots, x_n) = x_1 \oplus \cdots \oplus x_n$.

1.1: Hash Functions for Hash Tables

- Hash table: data structure to implement an associative array
- Example: search in a phone book



- Advantages: fast access (O(1) if all goes well), small storage
- Disadvantages: collisions slow down the access

Division Method

- The table has p entries from 0 to $p-1$ (p prime).
- Write message $\mathbf{x}$ as a string of n m-bit values: $\mathbf{x} = (\mathbf{x_1}, \ldots, \mathbf{x_n})$ with $x_i \in \{0,1\}^m$ for $1 \leq i \leq n$
- Compute $h_1, \ldots, h_n$:

$$\begin{aligned} h_1 &= \mathbf{x_1} \\ h_2 &= (2 * h_1 + \mathbf{x_2}) \bmod p \\ &\vdots \\ h_i &= (2 * h_{i-1} + \mathbf{x_i}) \bmod p \end{aligned}$$

and set finally

$$h(\mathbf{x}) = h(\mathbf{x_1}, \ldots, \mathbf{x_n}) = h_n.$$

1.2: Cryptographic Hash Functions

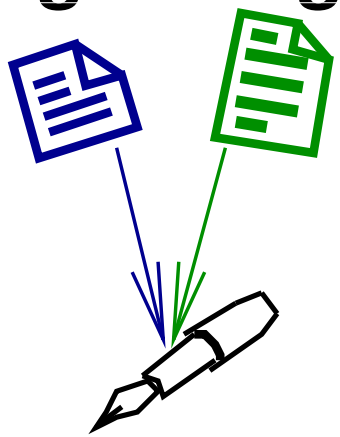
Example use cases

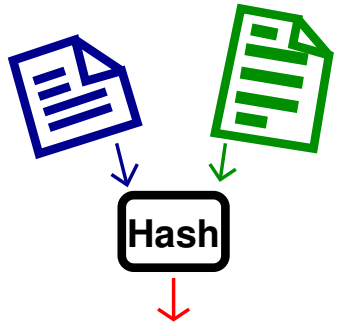
- Discover unauthorized manipulations of the message

  When first initialized, Open Source Tripwire scans the file system as directed by the administrator and stores information on each file scanned in a database. At a later date the same files are scanned and the results compared against the stored values in the database. Changes are reported to the user. Cryptographic hashes are employed to detect changes in a file without storing the entire contents of the file in the database.

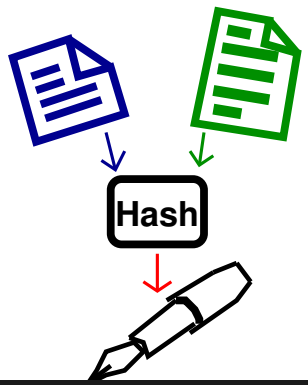
- Or in the context of digital signatures ($\rightarrow$ next slide)
- Or for “Commitments” ($\rightarrow$ blackboard).

Digital Signatures (Hash-then-Sign)





- Sign message of any length:
- Typically in two steps:
  1. Apply hash function H to compute a short n-bit fingerprint of message
  2. Apply signature scheme for n-bit messages.



Classical security requirements

Consider a hash function $h : \mathcal{X} \to \mathcal{Y}$:

Collision Resistance:

Infeasible to find a pair $\mathbf{x} \neq \mathbf{y}$ with $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ and $h(\mathbf{x}) = h(\mathbf{y})$

Second-preimage Resistance:

Given a random $\mathbf{x} \in_R \mathcal{X}$, infeasible to find $\mathbf{y} \in \mathcal{X}$ with $h(\mathbf{x}) = h(\mathbf{y})$.

(Pure) Preimage Resistance:

Definition 1: Given a random $\mathbf{z} \in_R \mathcal{Y}$, infeasible to find $\mathbf{x} \in \mathcal{X}$ with $h(\mathbf{x}) = \mathbf{z}$

Definition 2: Given $z = h(\mathbf{x})$, where $\mathbf{x} \in_R \mathcal{X}$ is randomly chosen; infeasible to find any $\mathbf{x}' \in_R \mathcal{X}$ with $h(\mathbf{x}') = z$, regardless of $\mathbf{x} = \mathbf{x}'$ or not.

Also important (a few slides ahead):

"Behave like a random oracle"

Example

Attacking the cryptographically weak division-method hash function

Consider h, based on the division method with prime $p = 7$:

1. Find $\mathbf{x}$ with $h(\mathbf{x}) = 1$.
2. Given $\mathbf{x} = (0011001100\,1100)$, find $h(\mathbf{x})$.
3. Find any $\mathbf{x}' \neq \mathbf{x}$ with $h(\mathbf{x}) = h(\mathbf{x}')$.
4. Find $\mathbf{x}$ with $h(\mathbf{x}) = 0$.
5. Find $\mathbf{x}$ with $h(\mathbf{x}) = 5$.

6. Homework: Repeat steps 1–5 for $p = 2857$.

Relationship between security definitions and example attacks

See blackboard for details

Relationship:

- Collision-resistant $\Rightarrow$ second-preimage-resistant.
- But not (second-preimage-resistant $\Rightarrow$ collision resistant).

Generic: Let $h : \{0,1\}^* \rightarrow \{0,1\}^n$ be an n-bit HF.

- Preimages and second preimages in time 2^n.
- Collisions in time $2^{n/2}$ (!)
  (note that the attack given here also requires memory of $2^{n/2}$; later we will see that an attack in roughly the same time with $O(1)$ units of memory is possible)

1.3: Random Oracles

The Problem:

- Collision, second and pure preimage resistance sometimes insufficient
- The exact properties a hash function should have are difficult to model/describe . . .

But we would need to define them **to prove the security of complex cryptosystems** using a hash function and other primitives.

An apparent Solution:

Assume the hash function behaves like an “ideal hash function”.

Random oracle = Ideal hash function

. . . which we do not have in practice

- For all $X \in \{0,1\}^*$: choose random $H(X) := Y \in \{0,1\}^n$.
- In other words:
  - If $X \neq X'$, then $H(X)$ and $H(X')$ are two independent uniformly distributed random variables.
  - If $X = X'$, then $H(X)$ is a uniformly distributed random variable and $H(X') = H(X)$.
- We cannot implement such a function. (**Why not?**)
- But we can perform "lazy sampling":
  - **Define** List of pairs $(X, H(X))$, initially empty
  - **Enter** input X
  - **If** no pair $(X, *)$ in the List **then**

    Choose $H(X) \in \{0,1\}^n$ at random
    Insert $(X, H(X))$ into the list
  - ($*$ Given X, there is exactly one $H(X)$ with $(X, H(X))$ on the list $*$)
  - **Return** $H(X)$

Example: Digital Signatures ("Full Domain Hash")

"Using RSA to sign the hash of a message"
Given: message M, RSA-key (N, e, d) and $H : \{0,1\}^* \to \mathbb{Z}_N$

$$M \longrightarrow \boxed{X \leftarrow H(M)} \longrightarrow \boxed{Y \leftarrow X^d \bmod N} \longrightarrow Y$$

Sign(M)	**Verify**(M, Y)
$X \leftarrow H(M)$ return $Y \leftarrow X^d \bmod N$	$X \leftarrow Y^e \bmod N$ $X' \leftarrow H(M)$ if $X = X'$ then return *true* else return *false*

Formal Background

RSA Problem:

Given N, e and a random $Y \in \mathbb{Z}_N^*$, find X with $X^e \bmod N = Y$.

Chosen Message Attack (against a signature scheme):

For $i \in \{1, \ldots, q\}$ **loop**

1. Choose M_i
2. Receive the signature Y_i (here: $Y_i = H(M_i)^d \bmod N$).

The adversary knows the public key and the query bound q is polynomial in the security parameter.

Existential Forgery:

A pair $(M, Y) \notin \{(M_i, Y_i) \mid 1 \leq i \leq q\}$ with Verify(M, Y)=true.

Main Result

Theorem 1 (Bellare-Rogaway 1993)

In the random oracle model, Full Domain Hash is secure.
Namely, any efficient adversary able to provide an existential forgery in a chosen message attack can efficiently solve the RSA problem.

Un-Instantiability

- What happens, when we replace the random oracle by a “real” hash function (hopefully a “secure” one)?
- Hope: The complex cryptosystem remains secure.
- Problem:

Theorem 2 (Canetti-Goldreich-Halevi, 1998)

Assume a secure signature scheme, either in the random oracle model or even in the standard model. Then one can define another signature scheme, which is

- *Provably secure in the random oracle model, but*
- *Completely insecure in the standard model.*

1.4: Passwords and Password-Hashing

Search for “passwords compromised”

An Update on LinkedIn Member Passwords Compromised
blog.linkedin.com/2012/.../linkedin-member-**passwords-compromise**...
Jun 6, 2012 – We can confirm that some of the **passwords** that were **compromised** correspond to LinkedIn accounts. We are continuing to investigate this ...

eHarmony Hacked - 1.5 Million Passwords Compromised - Techn...
technorati.com › Technology › Articles
Jun 7, 2012 – Dating website eHarmony joins LinkedIn in suffering from a hack attack, with 1.5 million **passwords** cracked.

Yahoo hacked, 450,000 passwords posted online - CNN.com
www.cnn.com/2012/07/12/tech/web/...**hacked**/index.html

by Doug Gross - in 720 Google+ circles - More by Doug Gross
Jul 13, 2012 – Hackers posted online what they say is login information for more than 450000 Yahoo users.

⋮

How to Check if Your Yahoo, Gmail or AOL Passwords Were Leaked
mashable.com/2012/07/12/yahoo-voices-**hacked**/

by Samantha Murphy - in 266 Google+ circles - More by Samantha Murphy
Jul 12, 2012 – Yahoo has been the subject of a massive data breach, and your email address may be among the thousands **compromised**.

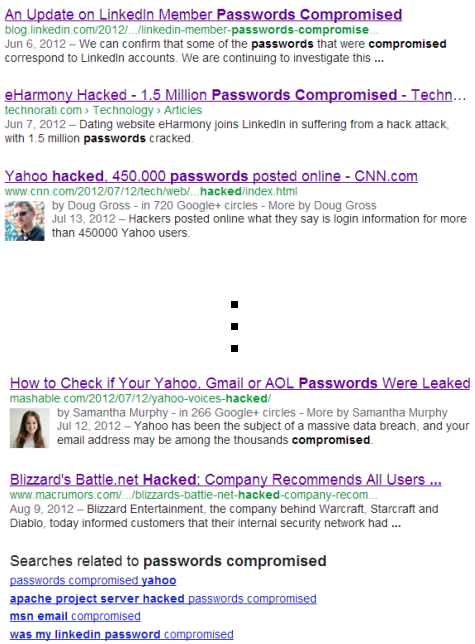
Blizzard's Battle.net Hacked; Company Recommends All Users ...
www.macrumors.com/.../blizzards-battle-net-**hacked**-company-recom...
Aug 9, 2012 – Blizzard Entertainment, the company behind Warcraft, Starcraft and Diablo, today informed customers that their internal security network had ...

Searches related to **passwords compromised**

passwords compromised **yahoo**
apache project server hacked passwords compromised
msn email compromised
was my linkedin password compromised

The Pain with Passwords

- Do not store **PW** in the clear
- But store “password-hash” H(**PW**)
- *H* should be
  - As slow as possible (so slow down trying out probable **PW**s)
  - . . . But not too slow for the user



Password Scrambling

A.k.a. “Password Hashing” a.k.a. “Password-based Key Derivation”

- Stone age: (username, **PW**, . . .) in the clear
- Unix crypt (username, S, H(S, **PW**), . . .)
  - Repeat DES-like operation 25 times (“key stretching”)
  - 12 bit salt S (hinders “dictionary attacks”)
- 1988: “Shadow passwords”
  (username, S, dummy, . . .), separate file with H(**PW**, S)
- 1995: Abadi, Lomas, Needham: “Pepper”
  (some bits of salt S remain secret)
- 1997: Kelsey, Schneier, Hall, Wagner: i iterations (“key stretching”)
- 2007: Boyen: unknown number of iterations (“halting”)
- 2013: Call for algorithms for new standard Password-Hashes
- March 2014: Deadline for submissions, Catena

Password Rules and Myths

Not from the dictionary: Good!

Mixture of UPPER and lower, digits and symbols: Does not harm, if you can still memorize the password.
Enforced on lots of systems. Unnecessary if password is sufficiently long.

L33t Sp3ak W0rd5: Doesn't harm! But password crackers have known the trick for a long time.

Don't write down passwords: Good rule, if THEY (NSA, Mafia, Interpol, . . .) are up to search you. But usually, a good password written down is better than a mediocre password that you are confident to remember.