Hash functions

Crypto Engineering

Bruno Grenet

Université Grenoble-Alpes

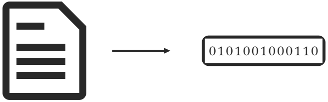
https://membres-ljk.imag.fr/Bruno.Grenet/CryptoEng.html

What are hash functions?

Definition

A(n unkeyed) hash function is a mapping $H : \mathcal{M} \to \mathcal{H}$, with

- $\mathcal{M} = \bigcup_{\ell < N} \{0,1\}^{\ell}$: the *message space* typically $N \geq 2^{64}$
- $\mathcal{H} = \{0,1\}^{n}$, with $N \gg n$: the *digests* $n \in \{\cancel{128}, \cancel{160}, \cancel{224}, 256, 384, 512\}$



x $\qquad H(x)$

Variants

- *extendable-output function* (XOF) $\to \mathcal{H} = \bigcup_{\ell < n} \{0,1\}^{\ell}$
- keyed hash function $H : \mathcal{K} \times \mathcal{M} \to \mathcal{H}$ family of hash functions

A hash function is simply a function: when is it *good*?

Usefulness of hash functions

Hash functions are an essential tool underlying most of (modern) cryptography!

- *Hash-and-sign* — RSA signatures, (EC)DSA, …
- Message authentication codes — HMAC, … $\rightarrow$ tomorrow!
- Password hashing (with a grain of salt)
- Hash-based signatures
- Commitment
- Key derivation
- As one-way functions or *random oracle*
- …

What are good hash functions?

Efficiency

- A few dozen cycles per byte
- Small memory
- ...

Security

- First preimage resistance: given t, *hard* to find m such that $H(m) = t$
- Second preimage resistance: given m, *hard* to find m' such that $H(m') = H(m)$
- Collision resistance: *hard* to find $m \neq m'$ such that $H(m) = H(m')$

Remarks

- No definition of *hard*
- In some sense, be careful!
  - collision resistance is stronger than 2^{nd} preimage resistance
  - 2^{nd} preimage is stronger than 1^{st} preimage resistance

The ideal world: random oracles

Definition

A random oracle is a function $H : \mathcal{M} \to \mathcal{H}$ such that $\forall x \in \mathcal{M}, H(x) \twoheadleftarrow \mathcal{H}$

- As random as possible
- Used in proof as the *random oracle model* — eq. to ideal cipher model
- Irrealistic but good hash functions are *approximations* — whatever this means

Generic attacks

- 1st preimage: $O(2^n)$ — exhaustive search
- 2nd preimage: $O(2^n)$ — *idem*
- Collision: $O(2^{n/2})$ — *"birthday attack"*

$\to$ A hash function is *good* if the generic attack is (almost) the best one

1. Hash functions from compression functions

2. Hash functions from permutations

Compression functions

Definition

A compression function is a mapping $f : \{0,1\}^n \times \{0,1\}^w \to \{0,1\}^n$

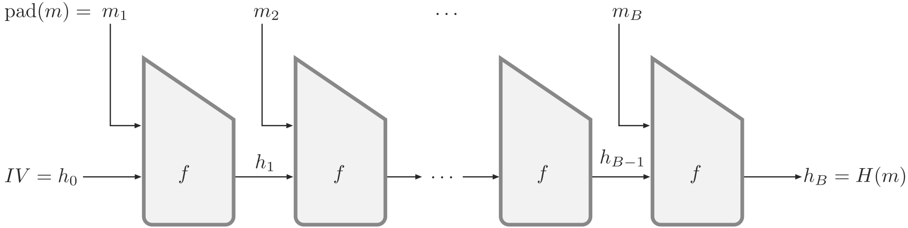
- Family of functions from $\{0,1\}^n$ to itself
- Compare to hash functions: fixed-length input
- Compare to block ciphers: not invertible

Goal

Assuming a *good* f is given, how to construct a *good* hash function?

- Fixed-size $\to$ Variable-size *domain extension*
- Compare to bock cipher modes of operation

The Merkle-Damgård construction (1989)



- $f : \{0,1\}^n \times \{0,1\}^w \to \{0,1\}^n$
- $\mathrm{pad}(m) = m \| 10 \cdots 0 \| \langle \text{length of } m \rangle \rightsquigarrow |\mathrm{pad}(m)| = B \times w$
- $H(m) = f(\cdots f(f(IV, m_1), m_2) \ldots, m_B)$

Efficiency

- B sequential calls to $f \to$ OK

Merkle-Damgård construction: security

Warm-up: first preimage resistance

- If f is 1st preimage resistant, then H is 1st preimage resistant too
- Contrapositive:
  - Assume that given t, an attacker can compute m s.t. $H(m) = t$
  - Then writing $\mathrm{pad}(m) = m_1 \| \cdots \| m_B$, $f(H(m_1 \| \ldots \| m_{B-1}), m_B) = t$

Collision resistance

- Attacker produces $m \neq m'$ s.t. $H(m) = H(m')$
  - let $\mathrm{pad}(m) = m_1 \| \cdots \| m_B$ and $\mathrm{pad}(m') = m'_1 \| \cdots \| m'_{B'}$
  - attacker also computes each h_i and each h'_i
- If $|m| \neq |m'|$, $m_B \neq m'_{B'}$ and $f(h_{B-1}, m_B) = f(h'_{B'-1}, m'_{B'})$ is a collision
- Otherwise, let b maximal s.t. $(h_{b-1}, m_b) \neq (h'_{b-1}, m'_{b-1})$, then
  - $h_b = h'_b$ since b is maximal
  - $f(h_{b-1}, m_b) = f(h'_{b-1}, m'_b)$ is a collision

Merkle-Damgård construction: 2^{nd} preimage vulnerability

Given m, find $m' \neq m$ s.t. $H(m') = H(m)$

Attack: very rough sketch

- Write $\mathrm{pad}(m) = m_1 \| \cdots \| m_B$, and $h_i = f(h_{i-1}, m_i)$ for all i
- Find a preimage of **any** h_i, of the form (h_0, m_0) $\simeq 2^n/B$
  - $m_0 \| m_{i+1} \| \cdots \| m_B$ almost works
  - But m_B contains the wrong length $\rightsquigarrow$ this is **not** $\mathrm{pad}(m')$ for any m'
- *If* we could find a family of m_0 of variable lengths $\rightsquigarrow$ OK
  - from *fixed points* $h_f = f(h_f, m_f)$ $\simeq 2^{n/2}$ (in some cases)
  - from *multicollisions* $m^1, \ldots, m^K$ with same hash $\simeq B \cdot 2^{n/2}$

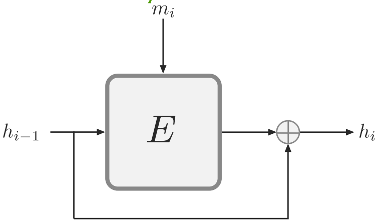
$\Rightarrow$ 2^{nd} preimage in $\simeq 2^n/B + B \cdot 2^{n/2}$ instead of $O(2^n)$ large B!

Patch: Chod-MD / Wide-pipe MD (2005)

- Use $f : \{0,1\}^{n+k} \times \{0,1\}^w \to \{0,1\}^{n+k}$
- Only keep the first n bits of $f(h_{i-1}, m_i)$ as input to next f
- Very strong provable guarantees

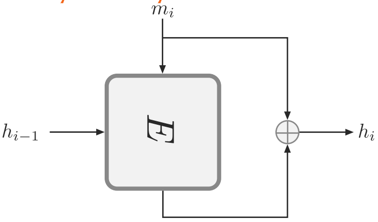
How to design compression functions?

Davies-Meyer construction



$$f(h_{i-1}, m_i) = E(m_i, h_{i-1}) \oplus h_{i-1}$$

Matyas-Meyer-Oseas construction



$$f(h_{i-1}, m_i) = E(h_{i-1}, m_i) \oplus m_i$$

Security

- Systematic analysis of possible constructions (“PGV constructions”)
- Rigorous proofs in the **ideal cipher model**
  - Not sufficient since actual block ciphers are not ideal!
  - Example: XBOX used a Davies-Meyer based construction with non-ideal cipher

Final words on Merkle-Damgård construction

- Many examples: MD4, MD5, SHA-0, SHA-1, SHA-2, ...
- MD5 failure:
  - 1992: Designed by Rivest
  - 1993: Collision attack on the compression function
  - 2005: Collision attack on the hash function
  - 2007-9: Practical useful collisions

  Used up to 2008 (at least), while alternatives were available since (at least) 1996!
- Another bad example: Git chose SHA-1 in 2005 while weaknesses were known

Lessons

- Care about attacks! Even *theoretical*!
- Most (every?) weaknesses can evolve to damaging attacks

Don't design your own crypto!

1. Hash functions from compression functions

2. Hash functions from permutations

Hash function from a permutation

Definition

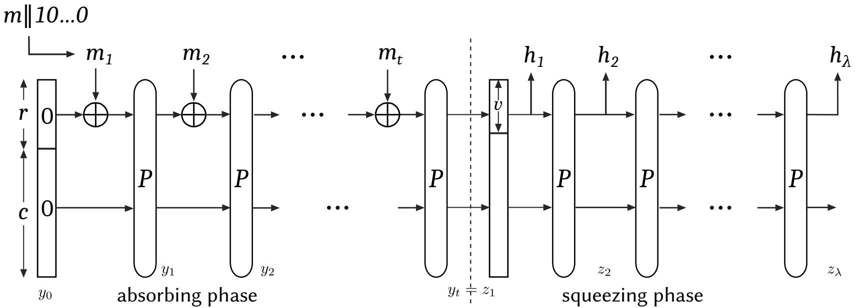
A permutation of $\{0,1\}^n$ is an invertible mapping $p : \{0,1\}^n \to \{0,1\}^n$.

- No key – no security notion such as PRP
- Ex.: for any block cipher, $E(0,\cdot)$ is a permutation
- Possible view: block cipher where key and plaintext are given together
- A permutation is invertible, but its inverse is often non necessary

Construction of a hash function

- *Sponge* construction : permutation $\to$ hash function
- Same general idea (but completely different construction) than Merkle-Damgård

The sponge construction



- $\mathrm{pad}(m) = m \| 10 \cdots 0 \rightsquigarrow$ length multiple of r
- Absorbing phase: compute $y_i = P(y_{i-1} \oplus (m_i \| 0^c))$ for $i = 1$ to t, with $y_0 = 0^r$
- Squeezing phase:
  - compute $z_i = P(z_{i-1})$ for $i = 2$ to λ, with $z_1 = y_t$
  - output $h_i =$ first v bits of z_i
- Finally: $H(m) = h_1 \| h_2 \| \cdots \| h_\lambda$

Source : J. Katz, Y. Lindell. Introduction to modern cryptography. 3rd ed, CRC Press, 2021. (modif.)

Sponge features

Sponge are convenient!

- If f is a random permutation, H is indifferentiable from a RO
- Flexible:
  - For a fixed permutation size, values of c, r, t, v and $\lambda \to$ speed/security trade-off
  - Natively a XOF (choose λ)
- Simplicity: easier to design a (good) permutation

SHA-3 – Keccak

- Hash function using the sponge construction, from a permutation of $\{0,1\}^{1600}$
- Standardized by NIST, after an academic competition (2008-2012)
- Best current choice for a hash function
- Four main variants: SHA3-224, SHA3-256, SHA3-384 and SHA3-512

If you need a hash function, use SHA-3!

Conclusion

Two main families

- Merkle-Damgård construction from a compression function
- Sponge construction from a random permutation
- Many broken constructions, few good ones…

… therefore:

Don't design crypto yourself!

- No generic way to build a hash function
- Every small detail counts!

Use SHA-3 (or maybe SHA-2)

- Don't use MD5!
- Don't use SHA-1!