Crypto Engineering '23

✧

Hash functions

Pierre Karpman
pierre.karpman@univ-grenoble-alpes.fr
https://membres-ljk.imag.fr/Pierre.Karpman/tea.html

2023–09–29

Hash functions: why

Hash functions are versatile primitive, good complement of block ciphers. Some possible applications:

- Hash-and-sign (RSA signatures, (EC)DSA, ...)
- Padding schemes (e.g. for RSA) (OAEP, FDH, PSS...)
- Message-authentication codes (NMAC, HMAC, SandwichMAC...)
- Password hashing (with a grain of salt)
- Hash-based signatures (not very efficient but PQ)
- Symmetric encryption
- As generic one-way functions (OWF)

First definition

Hash function

A hash function is a mapping $\mathcal{H} : \mathcal{M} \to \mathcal{D}$

So it really is just a function...

Usually:

- $\mathcal{M} = \bigcup_{\ell < N} \{0,1\}^{\ell}$, $\mathcal{D} = \{0,1\}^{n}$, $N \gg n$
- N is typically $\geq 2^{64}$, $n \in \{$~~128~~, ~~160~~, ~~224~~, 256, 384, 512$\}$

Also popular now: extendable-output functions (XOFs): $\mathcal{D} = \bigcup_{\ell < N'} \{0,1\}^{\ell}$

N.B.: Hash functions are *keyless*

Hash function security

Proceed as for block ciphers

- Define ideal hash functions (not standard model security)
- Derive search-based definitions
- (((Derive decision-based definitions)))

Idealized hash functions: Random oracles

Random oracle

A function $\mathcal{H} : \mathcal{M} \to \mathcal{D}$ s.t. $\forall x \in \mathcal{M}$, $\mathcal{H}(x) \twoheadleftarrow \mathcal{D}$

- Equivalent to an ideal BC (Coron et al., 2008; + later patches)

Search-based security definitions

What is hard for a RO should be hard for a "good" HF

$\rightsquigarrow$

1. **First preimage**: given t, find m s.t. $\mathcal{H}(m) = t$
2. **Second preimage**: given m, find $m' \neq m$ s.t. $\mathcal{H}(m) = \mathcal{H}(m')$
3. **Collision**: find $(m, m' \neq m)$ s.t. $\mathcal{H}(m) = \mathcal{H}(m')$

Generic complexity (for a constant success probability):
1), 2): $\Theta(N)$;
3): $\Theta(\sqrt{N})$

Decision-based security definitions

- Hard to do for single hash functions: no non-trivial distribution over it $\rightsquigarrow$ pseudorandomness defs. not implementable
- Can somewhat be done for families of functions, but doesn't match typical usage

Hash function design

Bottom-up approach similar to encryption scheme design from BC

- Define a subprimitive that you know how to build (e.g. *compression functions*; permutations)
- Find ways to build hash functions from (**any black-box instance** of) this primitive
- Prove appropriate security reductions ("collision/preimage-resistance of the compression function ⇒ — of the hash function")

Compression functions

Compression function

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

- A family of functions from n to n bits
- Not unlike a block cipher, only not (necessarily) invertible

Security defs. for compression functions:

- The same as for "full" hash functions
- But with some additional freedom from the "index" parameter

Compression function design

Can do it from scratch, or as a "mode" for block ciphers:

1. Take a block cipher, decide what goes where
2. Optionally add feedforward to prevent invertibility

Examples:
"Davies-Meyer": $f(h,m) = \mathcal{E}_m(h) + h$
"BRSS/PGV–13": $f(h,m) = \mathcal{E}_m(h)$
"Matyas-Meyer-Oseas": $f(h,m) = \mathcal{E}_h(m) + m$

- Systematic analysis of simple BC-based constructions by Preneel, Govaerts and Vandewalle (1993). "PGV" constructions
- Then rigorous proofs **in the ideal cipher model** (Black et al., 2002), (Black et al., 2010)

ICM PROOFS ARE NOT STANDARD

- Proofs in the ICM are **NOT REDUCTION PROOFS**
- Only rule-out “generic” attacks that don’t exploit structural properties of the BC
- $\rightsquigarrow$ Don’t give much guarantee about black-box instantiation

What does a security proof in the ICM say?

- Possibly a good basis for a construction
- But any instantiation needs a dedicated security analysis (e.g. through cryptanalysis)

Idealised models $\neq$ standard model

Microsoft needed a hash function for ROM integrity check of the XBOX

- Used TEA (Wheeler and Needham, 1994) in DM mode (Steil, 2005)
- Because of an earlier break of their RC4-CBC-MAC scheme (ibid.)
- Terrible idea, because of existence of equivalent keys for TEA (Kelsey et al., 1996)!
  - Keyspace is partitioned into (easy-to-define) classes of size 4
- For every k, it is easy to compute $\hat{k}$ s.t. $\mathsf{TEA}(k, m) = \mathsf{TEA}(\hat{k}, m) \Rightarrow \mathsf{DM\text{-}TEA}(h, k) = \mathsf{DM\text{-}TEA}(h, \hat{k}) \Rightarrow$ trivial collisions!

The XBOX got hacked...

And yet, TEA is a "good" PRP (as far as we know)!

It doesn't *have* to be bad, tho

- AES in a PGV construction so far unbroken (see *e.g.* Sasaki (2011))
  - But small parameters‽
- Ditto, SHA-256's compression function as a block cipher: "SHACAL-2" (Handschuh and Naccache, 2001)
  - Enormous keys, 512 bit!

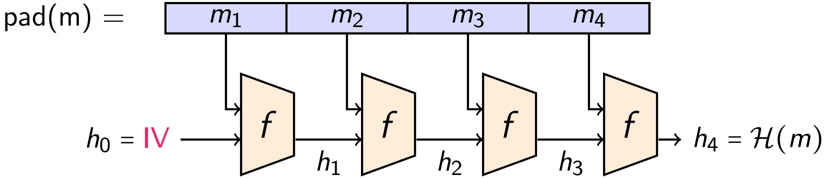
Domain extension of compression functions

Assume a good f

- Main problem: fixed-size domain $\{0,1\}^n \times \{0,1\}^b$
- Objective: *domain extension* to $\bigcup_{\ell<N}\{0,1\}^\ell$

The classical answer: the Merkle-Damgård domain extender (1989)

MD: with a picture



That is: $\mathcal{H}(m_1 \| m_2 \| m_3 \| \ldots) = f(\ldots f(f(f(\mathrm{IV}, m_1), m_2), m_3), \ldots)$

$\mathrm{pad}(m) \approx m \| 1000\ldots00\langle\text{length of } m\rangle \leftarrow$ *strengthening*

MD security proof (sktech)

Method: simple contrapositive arguments

- Attack $\{1^{\text{st}}\text{preim., coll.}\}$ on $\mathcal{H} \Rightarrow$ attack $\{1^{\text{st}}\text{preim., coll.}\}$ on f

First preimage case

If $\mathcal{H}(m_1\|m_2\|\ldots\|m_\ell) = t$, then $f(\mathcal{H}(m_1\|m_2\|\ldots\|m_{\ell-1}), m_\ell) = t$

Collision case (sketch)

If $\mathcal{H}(m_1\|m_2\|\ldots\|m_\ell) = \mathcal{H}(m'_1\|m'_2\|\ldots\|m'_\ell)$, show that $\exists i$ s.t. $(h_i := \mathcal{H}(m_1\|m_2\|\ldots\|m_{i-1}), m_i) \neq (h'_i := \mathcal{H}(m'_1\|m'_2\|\ldots\|m'_{i-1}), m'_i)$ and $f(h_i, m_i) = f(h'_i, m'_i)$

- Proper message padding (such as stenghtening) necessary to make it work!

What about 2nd preimages??

No proof (with optimal resistance), can't have one:

- Generic attack on messages of 2^k blocks for a cost $\approx k2^{n/2+1} + 2^{n-k+1}$ (Kelsey and Schneier, 2005)
- Idea: exploit internal collisions in the h_is

This is not nice, but:

- Requires (very) long messages to gain something
- At least as expensive as collision search
  - Always going to be the case, as preimage $\Rightarrow$ collision
- If n is chosen s.t. generic collisions are out of reach, we're somewhat fine

$\Rightarrow$ Didn't make people give up MD hash functions (MD5, SHA-1, SHA-2 family)

MD variants w/ optimal 2nd preimage resistance

Simple MD variants: Chop-MD/Wide-pipe MD (Coron et al., 2005) and (Lucks, 2005)

- Build $\mathcal{H}$ from $f : \{0,1\}^{2n} \times \{0,1\}^{b} \to \{0,1\}^{2n}$, truncate output to n bits (say)
- Collision in the output $\not\Rightarrow$ collision in the internal state
- Very strong provable guarantees (in an ideal model) (Coron et al.)
    - Secure domain extender for fixed-size RO (*ideal* compression function)
- Concrete instantiations: SHA-512/224, SHA-512/256 (2015)

Careful with models (again)!

- Coron et al. prove very strong *indifferentiability* properties for Chop-MD w/ an ideal CF
- But this in fact doesn't guarantee things such as preservation of collision-resistance (Bellare & Ristenpart, 2006)!
  - One can do "stupid things" with a non-ideal compression function
  - $\rightsquigarrow$ Chop-MD with a (real) CR c.f. is not (necessarily) CR!
  - (In essence, one needs strengthening in the padding)

Proofs: what do they tell us? (An MD exegesis)

- If one can't attack collision/preimage security of f underlying $\mathcal{H}$, all is well
- Else, ...???
- $\rightsquigarrow$ Attacking f is a meaningful goal for cryptographers ($\approx$ *(semi-)freestart attacks*)
- Don't use a $\mathcal{H}$ with broken f
  - Same as not using CTR[$\mathcal{E}$] w/ a broken $\mathcal{E}$ (w.r.t. PRP security)

The MD5 failure

- MD5: designed by Rivest (1992)
- 1993: very efficient collision attack on the compression function (den Boer and Bosselaers); mean time of 4 minutes on a 33 MHz 80386
- MD5 still massively used...
- 2005: very efficient collision attack on the *hash function* (Wang and Yu)
- Still used...
- 2007: practically threatening collisions (Stevens et al.)
- Still used...
- 2009: even worse practical collision attacks (Stevens et al.)
- Hmm, maybe we should move on?

Was this avoidable?

Yes!

- Early signs of weaknesses $\rightsquigarrow$ move to alernatives ASAP!
- What were they (among others)?
  - 1992: RIPEMD (RIPE); practically broken (collisions) 2005 (Wang et al.)
  - 1993: SHA-0 (NSA); broken (collisions) 1998 (Chabaud and Joux); practically broken 2005 (Biham et al.)
  - 1995: SHA-1 (NSA); broken (collisions) 2005 (Wang et al.); practically broken 2017 (Stevens et al. (and me!))
  - 1996: RIPEMD-128 (Dobbertin et al.); broken (collisions) 2013 (Landelle and Peyrin)
  - 1996: RIPEMD-160 (Dobbertin et al.); unbroken so far
  - 2001: SHA-2 (NSA); unbroken so far

Lesson to learn?

- Don't use broken algorithms
- Care about "theoretical" attacks
- An attack that's "too expensive" may become practical in the future

Perfect bad example: Git

- Don't use SHA-1 in 2005!
- Don't hide/misunderstand needed security properties!

Also:

- Don't use MD5, SHA-1..., even if you just care about preimage attacks