Radboud University

Hashing

Cryptography, Spring 2020

L. Batina, J. Daemen

March 4, 2020

Institute for Computing and Information Sciences
Radboud University

Outline

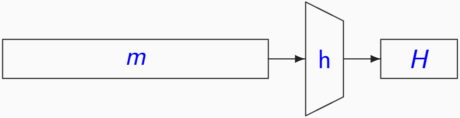
Hash function applications and requirements

Merkle-Damgård mode and provable security

MD5 and standards SHA-1 and SHA-2

Hash function applications and requirements

Hash function definition

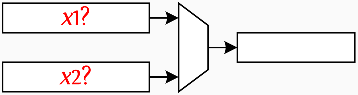


$$H \leftarrow \mathsf{h}(m)$$

- Function h from $\{0,1\}^*$ to $\{0,1\}^\ell$
  - no dedicated key input
  - input m has arbitrary length
  - output H, called *digest* or just *hash*, has fixed length ℓ
- *Secure if it behaves as a $\mathcal{RO}$, with output truncated to ℓ bits*
- So strength defined in terms of output length ℓ

Message compression and collision-resistance

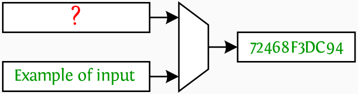
- Applications
  - signing m with private key PrK: sign $\mathsf{h}(m)$ instead
  - identification of a file m with its hash $\mathsf{h}(m)$ (e.g., in git, bittorrent)
- These rely on $\mathsf{h}(m)$ being *unique*
- Security notion: *collision-resistance*:
  - hard to find $x1 \neq x2$ such that $\mathsf{h}(x1) = \mathsf{h}(x2)$



- For $\mathcal{RO}$: $\Pr(\text{Success}) \approx N^2/2^{\ell+1}$ with N : # calls $\mathsf{h}(\cdot)$
  - expected cost of generating collision about $2^{\ell/2}$
  - collision resistance security strength $\leq \ell/2$
  - this is the birthday bound on the digest length ℓ

2nd pre-image resistance

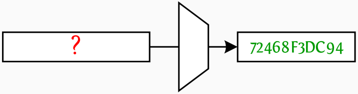
- Sometimes collision-resistance is not required
- Examples
  - using an existing signature on m to forge a signature on m'
  - forge a file m identified by $\mathsf{h}(m)$ to m'
- Security notion: *2nd preimage resistance*:
  - given m and $\mathsf{h}(m)$, find $m' \neq m$ such that $\mathsf{h}(m') = \mathsf{h}(m)$



- Generic attack (on $\mathcal{RO}$) has succes probability $N/2^{\ell}$
  - security strength limited to ℓ instead of $\ell/2$

Hashing passwords and pre-image resistance

- Application
  - storage of hashed passwords on servers: $\mathsf{h}(\mathsf{password} \| \mathsf{salt})$
- Security notion: *preimage resistance*:
  - given y, find any m such that $\mathsf{h}(m) = y$



- Security strength $\leq \ell$
- Sometimes it is not pure preimage resistance what we want
  - m may have to satisfy *certain criteria*, e.g., ASCII-coded
  - problem may be: given y obtained as $y \leftarrow \mathsf{h}(m)$, find *that* m

Keyed hashing

- MAC computation: $\mathrm{h}(K \| m) = T$
- Stream cipher: $\mathrm{h}(K \| D \| i) = z_i$ (keystream block)
- Key derivation $\mathrm{h}(\mathrm{Master}K \| \text{“Bob”}) = K_{\mathrm{Bob}}$
  - different diversifier values give independent subkeys
  - in payment systems: MK in bank, K_i in IC card
  - knowledge of K_i shall not reveal MK
  - also used in TLS for computing symmetric keys ...
- Hashed password storage: $\mathrm{h}(\mathrm{password} \| \mathrm{salt})$

Keyed hashing security notion: pseudorandom function (PRF)

- PRF security: difficulty of distinguishing $\mathsf{h}(K\|\cdot)$ from $\mathcal{RO}$
- Single-target security strength $\leq |K|$
- Caution: strength upper bound by entropy of K
- Same notion as for stream cipher and MAC functions
- MAC function: forgery success probability $\mathsf{h}(K\|\cdot)$ sum of:
  - probability of guessing a random ℓ-bit tag correctly: $2^{-\ell}$
  - advantage of distinguishing $\mathsf{h}(K\|\cdot)$ from $\mathcal{RO}$

Other applications and requirements

- There are many applications of hash functions
  - destroying algebraic structure, e.g.,
    - encryption with RSA: OAEP [PKCS #1]
    - signing with RSA: PSS [PKCS #1]
  - more than 800 uses of hash function MD5 in MS Windows
- Problems:
  - for designer: what to aim for?
  - for user: what are the (claimed) security properties?
  - **expressing security claim is non-trivial**
- Design approach: try to build hash function that *behaves like a* $\mathcal{RO}$
  - there exist counterexamples proving this is impossible
  - still the best we can come up with and *intuitively kind of clear*

Domain separation

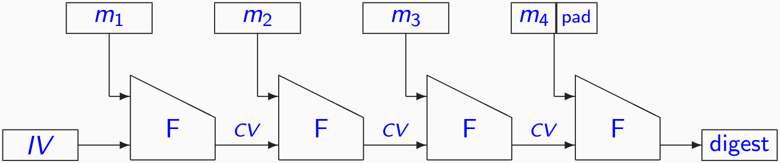
- Some applications need multiple *independent* hash functions
- This can be done with a single h using *domain separation*
  - output of $\mathsf{h}(m \| \mathbf{0})$ and $\mathsf{h}(m \| \mathbf{1})$ are independent
  - ... unless h has a cryptographic weakness
- Generalization to 2^w functions with D a w-bit *diversifier*

$$\mathsf{h}_D(m) = \mathsf{h}(m \| D)$$

- Variable-length diversifiers: suffix-free set of strings

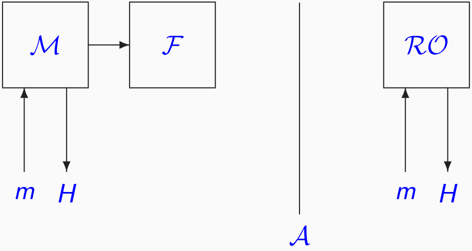
Merkle-Damgård mode and provable security

Classical iterative hashing: Merkle-Damgård



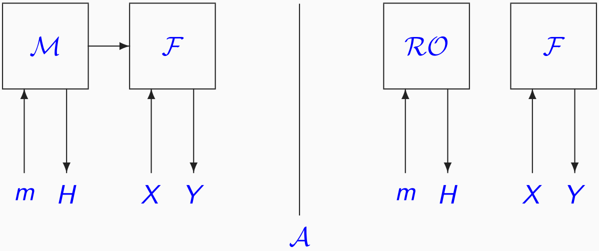
- Mode of use of a fixed-input-length compression function F
- *Collision-resistance preserving*
  - collision in hash function implies collision in F
  - reduces hash function design to fixed-input-length compression function design
  - implies fixing initial value (IV) of chaining value (CV) and conditions on the padding
- Important
  - used in MD5 and standards SHA-1, SHA-2
  - many experts (still) believe this is a good idea

Security of the hashing mode: (black-box) distinguishing setup



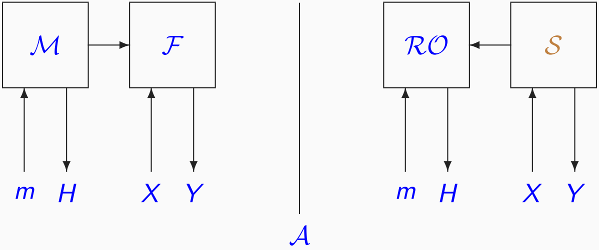
- Advantage of distinguishing between:
  - real world: mode $\mathcal{M}$ calling the ideal $\mathcal{F}$: $\mathcal{M}(\mathcal{F})$
  - ideal world: $\mathcal{RO}$
- Can be used to analyze concrete modes like Merkle-Damgård
- Problem: this adversary model is too weak
  - in real world adversary should be able to query $\mathcal{F}$
  - we don't want to base hash function security on secrecy of $\mathcal{F}$

Hashing mode security: attempt to fix distinguishing setup



- We give adversary access to $\mathcal{F}$ in real and ideal world
- Unfortunately, now any $\mathcal{M}$ can be distinguished in a few queries:
  - Adversary queries h ($\mathcal{M}(\mathcal{F})$ or $\mathcal{RO}$) with m
  - Adversary simulates mode $\mathcal{M}(\mathcal{F})$ by making calls to $\mathcal{F}$ herself
- $(\mathcal{M}(\mathcal{F}), \mathcal{F})$ will behave $\mathcal{M}$-consistently
- $(\mathcal{RO}, \mathcal{F})$ both return random responses so not likely $\mathcal{M}$-consistent
- Note: keyed modes do not have this problem:
  - unknown key K prevents simple $\mathcal{M}$-inconsistency check

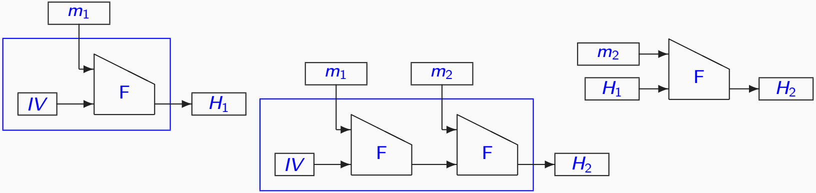
Modeling public compression function: indifferentiability



- Concept by [Maurer et al. (2004)], for hashing [Coron et al. (2005)]
  - adversary gets access to $\mathcal{F}$ in real world
  - introduces counterpart in ideal world: *simulator* $\mathcal{S}$
- Methodology for proving bounds on the advantage:
  - build $\mathcal{S}$ that makes left/right distinguishing difficult
  - prove bound for advantage given this simulator $\mathcal{S}$
  - $\mathcal{S}$ may query $\mathcal{RO}$ for acting $\mathcal{M}$-consistently: $\mathcal{S}(\mathcal{RO})$
- Advantage in this setting is the benchmark for hash mode security

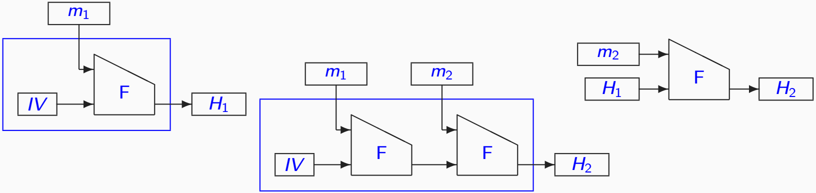
Merkle-Damgård weakness: length extension

- Take indifferentiability setup with $\mathcal{M}$ = Merkle-Damgård
- Distinguish $(\mathcal{M}(\mathcal{F}), \mathcal{F})$ from $(\mathcal{RO}, \mathcal{S}(\mathcal{RO}))$ in 3 queries:



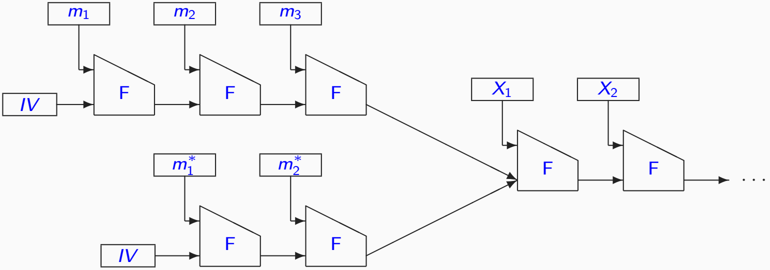
- Query h with m_1 resulting in H_1
- Query h with $m_1 \| m_2$ resulting in H_2
- Query $\mathcal{F}$ with $H_1 \| m_2$ resulting in H'
- For $(\mathcal{M}(\mathcal{F}), \mathcal{F})$ we have $H' = H_2$.
- Simulator cannot enforce this because it doesn't know m_1 to ask $\mathcal{RO}$
- This is called the *length extension weakness*:
  - one can compute $\mathsf{h}(m_1 \| m_2)$ from $H_1 = \mathsf{h}(m_1)$ and m_2 only
  - generalizes to multi-block strings m_1 and m_2
  - major problem for MAC function $\mathsf{h}(K \| \cdot)$

Merkle-Damgård weakness: length extension (cont'd)



- Why does Merkle-Damgård have the length extension weakness?
  - adversary gets CVs (here H_1) by queries to $\mathcal{M}(\mathcal{F})$
  - if $\mathcal{M}(\mathcal{F})$ cannot return CVs, $\mathcal{S}$ can be made $\mathcal{M}$-consistent
- Easy fix by dedicating bit in F input to indicate final/non-final
  - $CV \leftarrow \mathsf{F}(m_1 \| IV \| 0)$ for first block
  - $CV \leftarrow \mathsf{F}(m_i \| CV \| 0)$ for intermediate block
  - $H \leftarrow \mathsf{F}(m_n \| \mathsf{pad} \| CV \| 1)$ for last block
  - $H \leftarrow \mathsf{F}(m \| \mathsf{pad} \| IV \| 1)$ for short message
- This was never applied for standard Merkle-Damgård hash functions

The limit of iterative hashing: internal collisions



- There exist inputs $m \neq m^*$ leading to same CV
- Messages $m \| X$ and $m^* \| X$ always collide for any string X
- This effect does not occur in $\mathcal{RO}$
- Security strength upper bound by birthday bound in CV length

Distinguishing iterative hashing modes from $\mathcal{RO}$

- Send N queries to $\mathcal{RO}/\mathcal{M}(\mathcal{F})$ of form $m^{(i)}\|X$ with X always same
  - if there is no collision, say $\mathcal{RO}$
  - otherwise, we have one or more collisions for some $i \neq j$
  - for each, query $m^{(i)}\|X'$ and $m^{(j)}\|X'$ for some $X' \neq X$
  - if equal: say $\mathcal{M}(\mathcal{F})$, otherwise: say $\mathcal{RO}$
- $\mathrm{Adv} \approx N^2 2^{-(|CV|+1)}$
  - security strength of iterative hashing $\leq |CV|/2$
  - truncating output to $\ell < |CV|$ does not affect advantage
- Attack success probability on hashing mode with ideal $\mathcal{F}$ at most:
  1. success probability of that attack on $\mathcal{RO}$ plus
  2. distinguishing advantage $N^2 2^{-(|CV|+1)}$

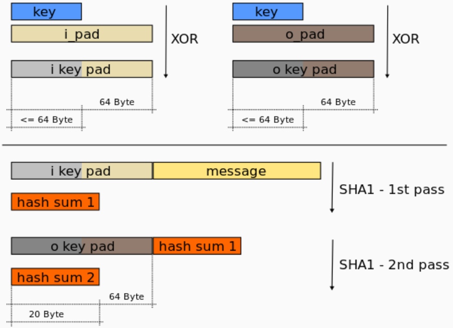
2nd preimage resistance of Merkle-Damgård

- In Merkle-Damgård: $|CV| = \ell$ (digest length)
- Success probability of 2nd preimage attack is upper bound by:
  (1) 2nd preimage attack on $\mathcal{RO}$ truncated to ℓ bits: $N2^{-\ell}$
  (2) distinguishing advantage: $N^2 2^{-(|CV|+1)} = N^2 2^{-(\ell+1)}$
- This leaves room for 2nd preimage attacks with $\Pr(succ.) \ggg N2^{-\ell}$
- Such attacks surfaced in 2004-2006 much to the surprise of the establishment
- E.g., 2nd pre-image of 2^d-block message in about $2^{|CV|-d}$ $\mathcal{F}$ calls
- Remedy: take $|CV| = 2\ell$
  - called *wide-pipe* hashing
  - Merkle-Damgård loses its collision-resistance preservation

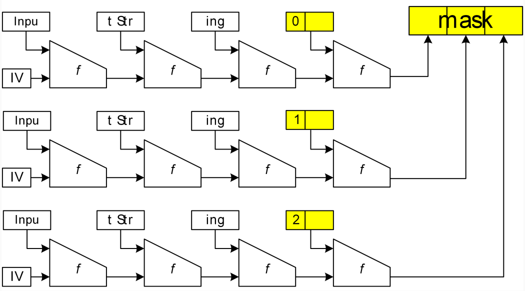
Patching length extension: HMAC mode [FIPS 197]

MAC mode with length extension patch for Merkle-Damgård

- Two calls to the hash function, like $T \leftarrow \mathsf{h}(K \| \mathsf{h}(K \| m))$
- Remember: $\mathsf{h}(K \| m)$ allows tag forgery by using length-extension
- Wikipedia figure:



Extending the output length: The mode MGF1 [PKCS #1]



- Repeating hash computation multiple times
- On message followed by counter
- Only last block must be processed multiple times

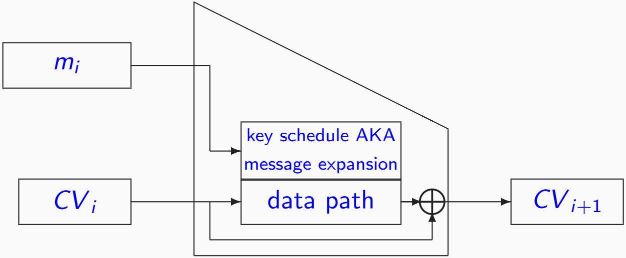
MD5 and standards SHA-1 and SHA-2

MD5 and standards SHA-1 and SHA-2

- MD5 [Ron Rivest, 1991]
  - based on MD4 that was an original design
  - 128-bit digest
- SHA-1 [NIST, 1995] (after SHA-0 [NIST, 1993])
  - inspired by MD5, designed at NSA
  - 160-bit digest
- SHA-2 series [NIST, 2001 and 2008]
  - *reinforced versions of SHA-1*, designed at NSA
  - 6 functions with 224-, 256-, 384- and 512-bit digest
- Internally:
  - Merkle-Damgård iteration mode
  - F based on a block cipher in Davies-Meyer mode

The Davies-Meyer mode for building a compression function

MD5, SHA-1 and SHA-2 all use a block cipher internally:

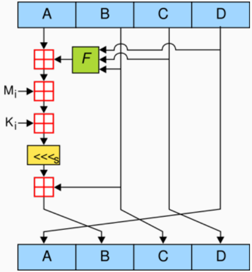


- This is called the Davies-Meyer mode
- Separation data path and message expansion (*key schedule*)
- Feedforward
  - due to Merkle-Damgård proof: collision-resistance preservation
  - otherwise it is trivial to generate collisions for F
- Why a block cipher: we don't know how to design a decent compression function from scratch

MD5 internals

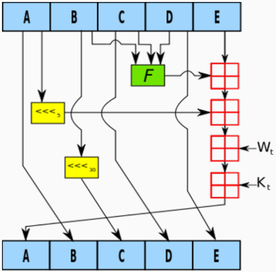
- Software oriented with 32-bit words
- 4-word *CV* and datapath
- 16-word message block
- 64 rounds, each taking one message word
- Hoped strength by combining **arithmetic, rotation and XOR (ARX)**

Round function:



SHA-1 internals

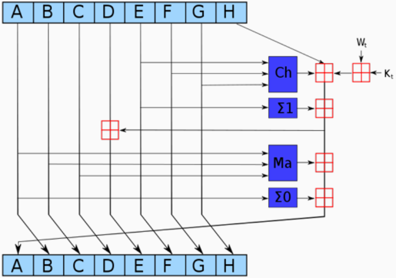
- Similar to MD5 but
  - 5-word state and 80 rounds
  - round i takes a word $w[i]$ of the *expanded message*
- Message expansion:
  - $i < 16 : w[i] = m[i]$
  - $i \geq 16 : w[i] = (w[i-3] \oplus w[i-8] \oplus w[i-14] \oplus w[i-16]) \lll 1$
  - similar to AES key schedule (this is where we got it)

Round function: 

SHA-2 internals

- 8-word state and nonlinear message expansion
- 6 versions:
  - SHA-256 and SHA-224: 32-bit words and 64 rounds
  - SHA-512, SHA-384, SHA-512/256 and SHA-512/224: 64-bit words and 80 rounds

Round function:



Security status of MD5, SHA-1 and SHA-2

- Problems of Merkle-Damgård:
  - perceived: strength against a.o. 2nd preimages below ℓ
  - real: length-extension weakness
- MD5
  - 1993: F shown weak (**before widespread adoption**)
  - 2003-2004: great advances in breaking MD5
  - despite weaknesses, corporate IT co. unwilling to abandon MD5
  - 2005: Lenstra, Wang, and De Weger use MD5 collisions to generate fake TLS certificates
  - 2016: MD5 largely replaced by SHA-256
- SHA-1
  - 2004-2007: theoretical collision attacks in effort $\approx 2^{61}$
  - 2017: collisions by Marc Stevens et al. published at `shattered.io`
- SHA-2 series: no specific problems outside of length extension

Conclusions

- Hash functions are modes built on underlying primitives
- Classical hash standards based on block ciphers
  - industry standard MD5 very badly broken
  - SHA-1 practically broken
  - SHA-2 has Merkle-Damgård length-extension weakness
  - dedicated modes are required: HMAC and MGF1

All in all a messy situation