Cryptographic Hash Functions

BİL 448/548
Internet Security Protocols

Ali Aydın Selçuk

Cryptographic Hash Functions

- Maps an arbitrary length input to a fixed-size output.
- Was originally proposed to generate input to digital signatures.
- Desirable features:
  - one-way (preimage and second preimage resistant)
  - pseudorandom
  - collision resistant

Pre-image & Collision Resistance

- Pre-image resistance:

  Given y, it should be hard to find M s.t.

  $$H(M) = y.$$
- Second pre-image (weak collision) resistance:

  Given M_1, it should be hard to find $M_2 \neq M_1$,

  $$H(M_2) = H(M_1).$$

  (Why necessary?)
- (Strong) Collision resistance:

  It should be hard to find *any* $M_1 \neq M_2$,

  $$H(M_1) = H(M_2).$$

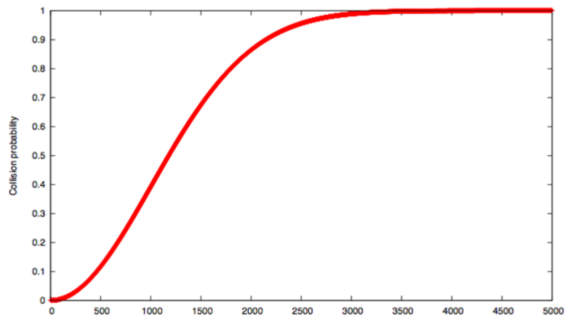
  (Why necessary?)

Collision Resistance

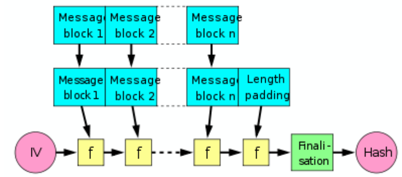
- But why “collision resistance”? (i.e., not just one-wayness?)
  - Assume a collision can be found (i.e., two messages with the same hash)
  - Alice generates two such messages and signs one of them. Later, she denies her signature and claims she in fact signed the other one.
- Birthday Problem (“paradox”): When $\sqrt{N}$ or more are chosen randomly from a domain of N, there is a significant chance of collision.
- Hence, output size ≥ 256 bits is desirable.

"Birthday Paradox"

E.g. $N = 10^6$:



Merkle-Damgard Construction



- Input is broken into equal-sized blocks and fed into the compression function.
- Length padding: 100…0 || (length of message). (Why?)
- Finalization: Optional
- Provable security: If f is collusion resistant, the hash function is collusion resistant.

Hash Fnc. from a Block Cipher

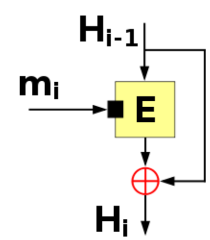
Compression fnc. from block cipher (Rabin):

- Split the message into *key blocks.* (why not pt.?)
- Encrypt a constant (e.g. 0) with this seq. of keys.
- Ciphertext is the hash output.

Hash Fnc. from a Block Cipher (cont.)

Davies-Meyer Construction:

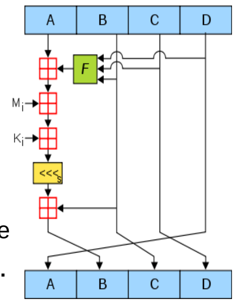
- $H_i = H_{i-1} \oplus E_{m_i}(H_{i-1})$
- Compression function is provably secure (collision resistant) if E is a secure block cipher.



MD5

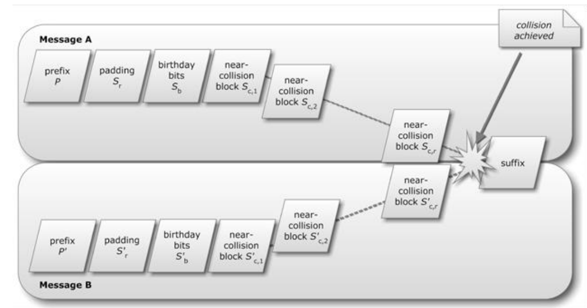
- Rivest, 1991
- Based on Davies-Meyer const.
- Very popular until recently.
  - 2004: First collision attacks
  - 2008: Practical collision attack; SSL cert. with same MD5 hash.
  - ~2010: Forged Microsoft MD5 certificates used in Flame malware
- Preimage resistance: Mostly ok.

64 rounds of:



Flame's MS Windows MD5 Attack

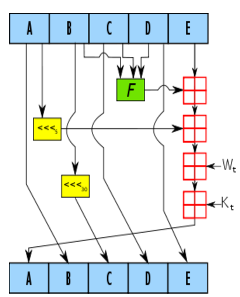
- Chosen-prefix coll. attack: Meaningful initial blocks, followed by random blocks to obtain collision.



SHA-1

- Designed by NSA; based on Rivest's MD4 & MD5 designs
- SHA 1993; SHA-1 1995
- 160-bit output size
- 2005: Some flaws discovered.
- SHA-2: 256- and 512-bit extension; secure
- SHA-3: By public competition

80 rounds of:



SHA-3

Public competition by NIST, similar to AES:

- NIST's request for proposals (2007)
- 51 submissions (2008)
- 14 semi-finalists (2009)
- 5 finalists (2010)
- Winner: Keccak (2012)
  - Designed by Bertoni, Daemen, Peeters, Van Assche.
  - Based on "sponge construction", a completely different structure.

Speed Comparisons

Algorithm	Speed (MiByte/s.)
AES-128 / CTR	198
MD5	335
SHA-1	192
SHA-256	139
SHA-3	~ SHA-256

Crypto++ 5.6 benchmarks, 2.2 GHz AMD Opteron 8354

- NIST expects SHA-2 to be used for the foreseeable future.
- SHA-3: A companion algorithm with a different structure and properties.

Things to Do with a Hash Function

- Hash long messages for signing
- Stream ciphers
- Block ciphers
- MACs
- Authentication protocols
- . . .

Stream Cipher

- CFB:
  $O_i = H(K \,||\, C_{i-1})$
  $C_i = P_i \oplus O_i$
  $P_i = C_i \oplus O_i$
- OFB:
  $O_i = H(K \,||\, O_{i-1})$
  $C_i = P_i \oplus O_i$
  $P_i = C_i \oplus O_i$
- CTR:
  $C_i = P_i \oplus H(K \,||\, IV + i)$
  $P_i = C_i \oplus H(K \,||\, IV + i)$

MACs from Hash Functions

A natural relative; but how to do it best?

- prefix: $MAC_K(x) = H(K \,||\, x)$
  - not secure; extension attack.
- suffix: $MAC_K(x) = H(x \,||\, K)$
  - mostly ok; problematic if H is not collision resistant.
- envelope: $MAC_K(x) = H(K_1 \,||\, x \,||\, K_2)$
- HMAC: $MAC_K(x) = H(K_2 \,||\, H(K_1 \,||\, x))$
  - provably secure; popular in Internet standards.

VMAC

- Proposed by Ted Krovetz in 2006.
- Based on a universal hash rather than collision resistant hash. (which is fine for MAC)
- Extremely fast (3 GB/sec); adjustable security-speed tradeoff.
- VMAC-64 is about 10x faster than HMAC-MD5; has a security proof that Pr(forgery) < 2^{-60}.
- Very suitable for infrastructure (routers) or low-end (RFID, WSN) authentication.

Authentication Protocol

- Challenge-response authentication instead of a password protocol, with a shared secret K.
- Typically implemented with a block cipher.

- Possible with a hash function instead of block cipher: