Cryptographic hashing

◆ Two families of hash functions:

1. Non-keyed hash functions:
   - H: $\{0,1\}^* \rightarrow \{0,1\}^n$ (e.g. n=160)
   - Used for password protection, digital signatures, …
2. Keyed hash functions:
   - H_{key}: $\{0,1\}^* \rightarrow \{0,1\}^n$ (e.g. n=96)
   - Used for message integrity (MAC) .

1

Non-keyed hash functions

H: $\{0,1\}^* \rightarrow \{0,1\}^n$

◆ The hash H(M) of a message M is called a **Message Digest**.

◆ Hash functions satisfy different properties depending on the application.

2

Preimage resistance

◆ H: $\{0,1\}^* \rightarrow \{0,1\}^n$ is preimage resistant if:

- Given random y
  it is hard to find M s.t. H(M) = y .

◆ Application: protecting the password file.

$Username_1$	$H(pwd_1, salt_1)$	$salt_1$
$Username_2$	$H(pwd_2, salt_2)$	$salt_2$

➢ Never store pwd in clear. Store hash of pwd.

3

2^{nd} preimage resistance

◆ H: $\{0,1\}^* \rightarrow \{0,1\}^n$ is 2^{nd} preimage resistant if:

- Given random M_1
  it is hard to find M_2 s.t. $H(M_1) = H(M_2)$.

◆ Application: virus protection (Tripwire)

File System	Floppy
File1	H(File1)
File2	H(File2)

➢ Defeat Tripwire: virus must find F s.t. H(F) = H(File1)

4

Collision resistance

◆ H: $\{0,1\}^* \rightarrow \{0,1\}^n$ is collision resistant if:

- It is hard to find M_1, M_2 s.t. $H(M_1) = H(M_2)$.

◆ Application: digital signatures.

- Signature = Sig_{alice} [H(M) , alice-priv-key]

◆ Suppose adversary has M_1, M_2 s.t. $H(M_1) = H(M_2)$

- Adversary asks Alice to sign M_1.
- Alice's sig is also a sig on M_2.

5

Relation between properties

◆ Roughly speaking:

Collision resistance ⇒
2^{nd} preimage resistance ⇒
preimage resistance.

◆ In other words:

- Hardest to construct collision resistant hashing.
- Much easier to construct 2^{nd} preimage resistance.

◆ From here on: focus on collision resistance.

6

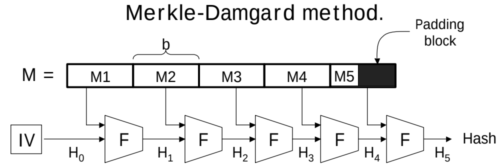
Birthday attack

- Birthday paradox:
  $r_1, \ldots, r_n \in [0,1,\ldots,B]$ indep. random integers.
  When $n = 1.2\sqrt{B}$ then
  $\Pr[\ \exists\, i \neq j:\ r_i = r_j\ ] > ½$
- msg-digest only 64 bits long $\Rightarrow$
  can find collision in 2^{32} tries.
- Typical digest size = 160 bits. (e.g. SHA-1)
  $\Rightarrow$ collision time is 2^{80} tries.

7

Constructions

- All constructions are iterated:
  Merkle-Damgard method.



- Terminology: $F(M_i, H_i)$ compression func.
  $|M_i|$ = block-size = 512 bits ; $|H_i|$ = chain-var = 160 bits

8

Motivation

- Why Merkle-Damgard iterated construction?
- Lemma: Suppose compression func $F(M_i, H_i)$ is collision resistant.
  $\Rightarrow$ resulting hash function is coll. resistant.
- Proof:
  Adversary finds M_1, M_2 s.t. $H(M^1) = H(M^2)$
  Then $\exists\, i$ s.t. $F(M^1_i, H^1_i) = F(M^2_i, H^2_i)$

9

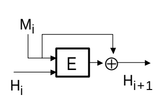
Constructions

- Main point:
  To construct CRHF suffices to construct collision resistant compression functions.
  $F:\ \{0,1\}^{512} \times \{0,1\}^{160} \rightarrow \{0,1\}^{160}$
- Compression functions:
  1. Based on block ciphers. Typically slow.
  2. Customized compression functions. Faster

10

Comp. func. from block ciphers

- Let $E(M,k)$ be a block cipher.
- Matyas-Meyer function:
  $F(M, H) = E(\ M,\ g(H) \oplus M\ )$



- Why is this collision resistant?

Thm: suppose $E_k(M) = E(M,k)$ is an ideal cipher.
$\Rightarrow$ finding collision takes $2^{n/2}$ evals of E .

- $2^{n/2}$: best possible! note: “black box security”

11

Customized compression func.

- Several special build compr. functions exist.

On 200MhZ Pentium:

Name	hash-len	speed	comment
MD4	128		Prop. Broken. Time: 2^{26}
MD5	128	28.5 MB/sec	collis. for compr. func.
SHA-1	160	15.25 MB/sec	NIST
RIPE-MD	160	12.6 MB/sec	RIPE

- Note: faster than 3DES, IDEA, etc.

12

Keyed hash functions

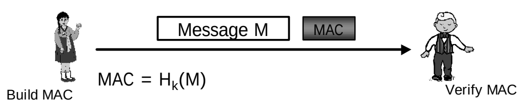
Keyed hash functions

H_k: $\{0,1\}^* \to \{0,1\}^n$

- Note: key k needed to evaluate function.
- Main application:
  Message Authentication Codes (MAC)
  Guarantee message integrity.
- $H_k(M)$ is a cryptographic "checksum".
  Ensures message has not been tampered.

Two scenarios

- Network scenario:
  Alice and Bob share a secret key k.



- Adversary can't build valid MAC for $M' \neq M$.
- Note: MAC used for integrity. Not privacy.
- Digital signatures work, but are too slow.

Second scenario

- File system:
  Bob protects a file on his file system.

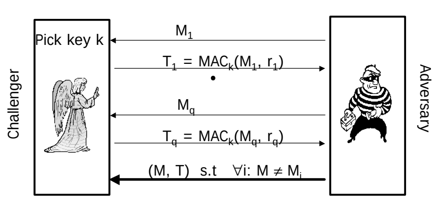
File	MAC

MAC = $H_{\text{bob-pwd}}$ (File)

- When accessing file, Bob verifies MAC.
- No one can modify file (without Bob's pwd).

What is a secure MAC?

- MAC $H_k(M,r)$ (r – random)
  is secure if not efficient adversary can win the following game with prob. > ε

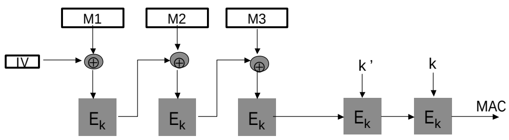


Constructing MACs

- Two types of constructions:
  - Cryptographic MAC:
    based on block ciphers (CBC-MAC)
    or non-keyed hash functions (HMAC).
  - Information theoretic MAC:
    Based on universal hashing.

CBC-MAC

- Most commonly used in banking industry.
  secret key = (k, k', IV)



- If E is a MAC then CBC-E is also a MAC.

MAC length

- Typical CBC-MAC length = 40 bits.

  ⇒ security of 2^{40} (guessing prob).

- Note: no birthday attack on MACs.
  ⇒ MACs are shorter than message-digest.

Hash based MAC

- MACs based on a non-keyed hash function h.
- Attempt 1: $MAC_k(M) = h(k \| M)$
  Insecure. Adv. can elongate M.
- Attempt 2: $MAC_k(M) = h(M \| k)$
  Insecure. Birthday attack.
- Envelope method:
  $MAC_{k,k'}(M) = h(k \| M \| k')$

Preferred method: HMAC

- HMAC used in IPsec, SSL, TLS.
- $HMAC_k(M) = h(k\|pad_1 \| h(k\|pad_2\|M))$
- "Thm":
  If compr. func. In h is a MAC and
  h is collision resistant
  then HMAC is a MAC.
- In IPsec, SSL use 96 bit HMAC.

Performance

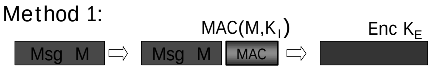
- HMAC is much faster than CBC-MAC.

On 200MhZ Pentium:

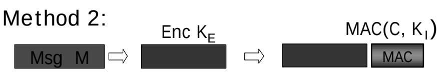
Name	hash-len	speed
MD5	128	28.5 MB/sec
SHA-1	160	15.25 MB/sec
3DES	64	1.6 MB/sec
IDEA	64	3 MB/sec

Both encryption and integrity

- Encryption key K_E. MAC key = K_I
- Method 1:



- Method 2:



- Wrong: