Collision-resistant hash functions in practice
(no keys, fixed-length functions)

\[ MD5, SHA-1, SHA-2 (SHA-256, SHA-512), SHA-3 \]

*Keeke* in draft NIST standard
- Selected in 2012 after competition
- NIST proposes changing security parameters 2013
- Broken down after uproar in crypto community

"MD5 Considered Harmful Today"
- 1991: Rivest publishes MD5
- 1993: "some weaknesses"
- 1995: SHA-1 proposed by NIST
- 2004: Vangel et al. collision for MD5 in Crypto rump session
  "MD5 to be considered harmful someday" - Kamskey
- 2009: Rogue CA certificate
  "MD5 to be considered harmful today"
- 2010: NIST recommends replacing SHA-1
- 2012: Flaw malware discovered
- MD5 collision in the wild against Microsoft certificate
- 2013: 32,000 trusted SSL certificates w/MD5 (0.54%)
  last signed April 2013 (Dormanc 2013)

Merkel-Damgard in MD5

\[
\begin{array}{c}
\text{msg} \rightarrow (m_0, \text{seq} \bmod 256) \\
L \rightarrow (L_0, L_1, 0) \\
| HV_0 \rightarrow | HV_1 \rightarrow | HV_2 \rightarrow | \ldots
\end{array}
\]

Intermediate hash value

Wang & Yu 2004: Fixed-prefix collision

\[
\begin{array}{c}
M_i \rightarrow | HV_i \\
M_{i+1} \rightarrow | HV_{i+1} \\
M_{i}' \rightarrow | HV_{i}' \\
M_{i+1}' \rightarrow | HV_{i+1}'
\end{array}
\]

Stevens 2007: Chosen-plaintext collision

\[
\begin{array}{c}
\text{plaintext} \rightarrow | A \rightarrow (K_1, K_2) \rightarrow \text{ciphertext}
\end{array}
\]

\[
\begin{array}{c}
\text{plaintext} \rightarrow | A \rightarrow (K_1, K_2) \rightarrow \text{ciphertext}
\end{array}
\]

\[ \text{pad to 512-bit} \]

\[ \text{birthday attack} \]

\[ \text{block boundary} \]

\[ \text{with} \]

\[ \text{differential} \]

\[ \text{hash specific form} \]
How to make and use a MAC from hashing

Non-working idea:

\[ \text{Mac}(m) = H(m) \]

"encryption with redundancy"

\[ \text{Enc}_k(m) = \text{Enc}(m || H(m)) \]

\[ \text{Dec}_k(m) = \text{Decrypt then verify } H(m) \]

Motivation: even if \( m \) is masked hard to produce \( \text{Enc}_k(H(m)) \)

Problem:

Consider CBC-mode \( \text{Enc} \):

\[ \begin{align*}
    A & \xrightarrow{m, H(m)} V \\
    C & \leftarrow \text{Enc}(m, h(C), h(m))
\end{align*} \]

\[ \text{Dec}(\text{IV}, C_0, C) \]

Succeeds for any non-secret-key redundancy function.

---

How not to construct a MAC from a Merkle-Damgård hash function

\[ \text{MAC}(m) = H(k || m) \]

Length extension attack:

Given \( \text{MAC}(m) = H(k || m) = H(k || m || \text{padding}) \) for \( k, m \) unknown

forge

\[ \text{Mac}(m') = H(k || m' || \text{padding} || r) \]

\[ H(k || m || \text{padding}) \xrightarrow{r} H \xrightarrow{r} H \xrightarrow{r} H(m || \text{padding} || r) \]
Secure Connection: HMAC

Gen: random key k ∈ {0,1}^n
Mac: H(k ⊕ opad || H(k ⊕ ipad || m))

| opad = 0x36 |
| ipad = 0x5c |

How does use MAC?

Encrypt-and-MAC?

\( c = \text{Enc}_{k}(m) \quad t = \text{Mac}_{k}(m) \quad \text{send } (c, t) \)

- Not CPA-secure:
  - Let \( \text{Mac}_{k}(m) = F_{k}(m) \)
  - MAC deterministic \( \Rightarrow \) can detect repeat
  - Security def. of MAC doesn't promise secrecy: \( (m, F_{k}(m)) \) is also unforgeable

- Used in SSH, secure for SSH

Mac-then-Encrypt?

\( t = \text{Mac}_{k}(m) \quad c = \text{Enc}_{k}(m || t) \quad \text{send } c \)

- CPA-secure

- Not CCA-secure:
  - Let \( \text{Enc}_{k}(m) = \text{CTR-AES}(\text{Enc}_{k}(m)) \quad \text{Encode}(m) = \begin{cases} 00 & \text{if } m = 0 \\ 01 & \text{if } m = 1 \end{cases} \)
  - Indistinguishability game:
    \[ A \begin{array}{c} m = 0 \quad m = 1 \\ \text{c} \end{array} \]
    \[ \begin{array}{c} \text{Enc} \quad \text{Dec} \quad \text{c} \end{array} \]
    - if 00 -? 11, mac valid \( \Rightarrow m_{b} = 0 \)
    - if 01 -? 00, mac valid still \( \Rightarrow m_{b} = 1 \)
    - Used in SSL/TLS, is secure
Encrypt-Then-MAC
CCA-secure encryption

Gen: $k_e, k_m$

Enc: input $(k_e, k_m)$, message $m$
    $c = Enc_{k_e}(m)$, $t = Mac_{k_m}(c)$
    output $(c, t)$

Dec: input $(k_e, k_m)$, ciphertext $(c, t)$
    check if $Verify_{k_m}(c, t) = 1$
    if so, output $Dec_{k_e}(c)$

Thus Above scheme is CCA-secure

pf (Inbook.)

Def: Ciphertext Integrity Game

\[\begin{align*}
A & \quad V \\
\text{oracle} & \quad k = (k_e, k_m) = \text{Gen}(1^n) \\
\text{Enc}_{k_e}(m) & \quad \text{in } Enc_{k_m}(m) \\
\rightarrow & \quad c \\
\rightarrow & \quad m = Dec_k(c) \\
A \text{ succeeds, } t & \in \{0, 1 \ldots, 2^n\} \\
Dec_k(c) & \text{ reject}
\end{align*}\]

Def: "authenticated communication"
\[\Pr(A \text{ succeeds}) \leq 2/n \text{ negl.} \]

Def: "secure" = CCA-secure + "authenticated communication"

"Authenticated Encryption"

Authenticated Encryption Standards

GCM: CTR mode + CTR-MAC
CCM: CBC-MAC + CTR mode
EAX: CTR mode + MAC