1 Message Auth. Code

Goal: Integrity, no confidentiality.
Examples

- Protecting public binaries on web
- Banner ads on webpages.

Generate tag: $S(k, m) \to \text{tag}$
Verify tag: $V(k, m, \text{tag}) =$ 'yes'
Def: MAC $I = (S, V)$ defined over (K, M, T) is a pair of algorithms:

- $S(k, m)$ outputs t in T
- $V(k, m, t)$ outputs 'yes' or 'no'.

Generate tag:
$CRC(m) \to \text{tag}$

- Attacker can easily modify message m and recompute CRC.
- CRC designed to detect random, not malicious changes.

Attackers power: Chosen message attack
for $m_1, m_2, \ldots, m_q$ attacker is given $t_i \leftarrow S(k_i, m_i)$
secure MAC size is 2^{80}

2 Protecting System Files

At install time the system computes: (File is: $F_1, t_1 = S(k, F_1)), F_2, t_2, S(k, F_2)$ where k is derived from system password.

3 Secure PRF $\implies$ Secure MAC

For a PRF $F : K \times X \to Y$ define a MAC $I_F = (S, V)$ as:

- $S(k, m) := F(k, m)$
- $V(k, m, t)$: output 'yes' if $t = F(k, m)$ and 'no' otherwise.

Suppose $F : K \times X \to Y$ is a secure PRF with $Y = \{0, 1\}^{10}$
Is the derived MAC $\mathrm{I}_F asecureMACsystem?$
$Notagsaretooshort : anyonecanguessthetagforanymessage.$$\mathrm{Adv}[A, I_F] = \frac{1}{1024}$

4 CBC-MAC and NMAC

Recall that a secure PRF implies a secure MAC, as long as Y is large $S(k, m) = F(k, m)$
Given a PRF for short messages (AES), construct a PRF for long messages.

From here on $X = \{0,1\}^n$ where $n \approx 128$.
The last encryption step in ECBC-MAC is because the MAC can be forged with one chosen message query:
Suppose we define a MAC $I_{RAW} = (S, V)$ where $S(k, m) = \text{rawCBC}(k, m)$. Then I_{RAW} is easily broken using a 1-chosen message attack. Adversary chooses and abritrary one block message $m \in X$. Requests tag for m. Get $t = F(k, m)$. Output t as a MAC forgery for the 2-block message $(m, t \oplus m)$.

$$\text{rawCBC}(k, (m, t \oplus m)) = F(k, F(k, m) \oplus (t \oplus m)) = F(k, t \oplus (t \oplus m)) = t$$

Theorem: For any L > 0, for every efficient q-query PRF advantage A attacking F_{ECBC} or F_{NMAC} there exists an efficient adversary B s.t.:

$$Adv_{PRF}[A, F_{ECBC}] \leq Adv_{PRP}[B, F] + \frac{2q^2}{|X|}$$

$$\text{Adv}_{PRF}[A, F_{NMAC}] \leq q \cdot L \cdot Adv_{PRF}[B, F] + \tfrac{q^2}{2|K|}$$

CBC-MAC is secure as long as $q \ll |X|^{\frac{1}{2}}$
NMAC is secure as long as $q \ll |K|^{\frac{1}{2}}$ $\quad \left(2^{64}\text{for AES-128}\right)$.

4.1 Example

$$Adv_{PRF}[A, F_{ECBC}] \leq Adv_{PRP}[B, F] + 2 \cdot \frac{q^2}{|X|}$$

$$\text{q} = \#\text{ messages MAC-ed with k}$$

Suppose we want $Adv_{PRF}[A, F_{ECBC}] \leq \frac{1}{2^{32}} \Leftarrow \frac{q^2}{|X|} < \frac{1}{2^{32}}$

- AES: $|X| = 2^{128} \implies q < 2^{48}$ So, after 2^{48} messages change key.
- 3DES: $|X| = 2^{64} \implies q < 2^{16}$

4.2 The Security Bounds are tight: an attack

After signing $|X|^{\frac{1}{2}}$ messages with ECBC-MAC or $|K|^{\frac{1}{2}}$ messages with NMAC the MACs become insecure.
Suppose the underlying PRF F is a PRP (e.g. AES) Then both PRFs (ECBC and NMAC) have the following extension property

$$\forall x, y, w : F_{BIG}(k, x) = F_{BIG}(k, y) \implies F_{BIG}(k, x \| w) = F_{BIG}(k, y \| w)$$

Let $F_{BIG} : K \times X \to Y$ be a PRF that has the extension property

$$F_{BIG}(k,x) = F_{BIG}(k,y) \implies F_{BIG}(k,x\|w) = F_{BIG}(k,y\|w)$$

Generic attack on the derived MAC:

1. Issue $|Y|^{\frac{1}{2}}$ message queries for random messages in X. Obtain (m_i, t_i) for $i = 1, \dots, |Y|^{\frac{1}{2}}$
2. Find a collision $t_u = t_v$ for $u \neq v$ (one exists w.h.p by birthday paradox)
3. Choose some w and query for $t := F_{BIG}(k, m_u\|w)$
4. Output forgery $(m_v\|w, t)$. Indeed $t := F_{BIG}(k, m_v\|w)$

5 PMAC - parallel MAC

Suppose $P(k,i)$ is an easy to compute function, using keys (k, k_1) and padding similar to CMAC. Let $F : K \times X \to X$ be a PRF. Define new PRF $F_{PMAC} : K^2 \times X^{\leq L} \to X$. Then the tag is calculated by xoring the message block and $P(k,i)$ and the output of that is put into the PRF $F(k_1, \cdot)$ except for the last block. Every output of the PRF's is xored together and put into another PRF $F(k_1, \cdot)$ to get the tag.

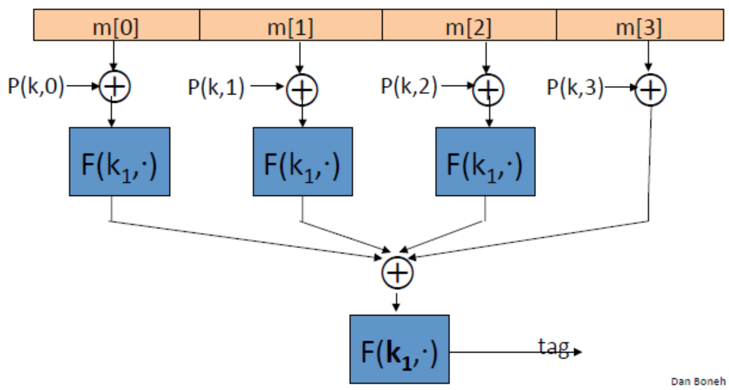

Figure 1: Parallel MAC taken from slides.

5.1 PMAC is incremental

When $m[1] \to m'[1]$ can we quickly update the tag?

$$\text{do } F^{-1}(k_1, tag) \oplus F(k_1, m[1] \oplus P(k,1)) \oplus F(k_1, m'[1] \oplus P(k,1)).$$

Then apply $F(k_1, \cdot)$ to receive the new tag.

6 One-time MAC

One-time MAC can be secure against **all** adversaries and faster than PRF-based MACs. Let q be a large prime (e.g. $q = 2^{128} + 51$). Key $= (a, b) \in \{1, \ldots, q\}^2$ (two random integers in $[1, q]$). Message $= (m[1], \ldots, m[L])$ where each block is 128 bit int.

$$S(\text{key, msg}) = P_{\text{msg}}(a) + b \,(\text{mod } q)$$

where $P_{\text{msg}}(x) = x^{L+1} + m[L] \cdot x^L + \ldots + m[1] \cdot x$ is a polynomial of degree $L + 1$.
we show: given $S(\text{key}, \text{msg}_1)$ adv. has no info about $S(\text{key}, \text{msg}_2)$.

6.1 One-time Security

Theorem: The one-time MAC satisfies (L=msg-len)

$$\forall m_1 \neq m_2, t_1, t_2 : \quad Pr_{a,b}[S((a, b), m_1) = t_1 \mid S((a, b), m_2) = t_2] \leq \frac{L}{q}$$

Proof: $\forall m_1 \neq m_2, t_1, t_2$:

1. $Pr_{a,b}[S((a, b), m_2) = t_2] = Pr_{a,b}[P_{m_2}(a) + b = t_2] = \frac{1}{q}$
2. $Pr_{a,b}[S((a, b), m_1) = t_1 \text{ and } S((a, b), m_2) = t_2] =$
   $Pr_{a,b}[P_{m_1}(a) - P_{m_2}(a) = t_1 - t_2 \text{ and } P_{m_2}(a) + b = t_2] \leq \frac{L}{q^2}$

$\implies$ given valid (m_2, t_2), adv. outputs (m_1, t_1) and is right with prob. $\leq \frac{L}{q}$.

7 One-time MAC $\implies$ Many-time MAC

Let (S, V) be a secure one-time MAC over $(K_l, M, \{0,1\}^n)$.
Let $F : K_F \times \{0,1\}^n \to \{0,1\}^n$ be a secure PRF.

Carter-Wegman MAC: $CW((k_1, k_2), m) = (r, F(k_1, r) \oplus S(k_2, m))$ for random $r \leftarrow \{0,1\}^n$.
Theorem: if (S, V) is a secure **one-time** MAC and F a secure PRF then CW is a secure MAC outputting tags in $\{0,1\}^{2n}$.
How would you verify a CW tag (r, t) on message m?
Recall that $V(k_2, m, \cdot)$ is the verification algorithm for the one time MAC.