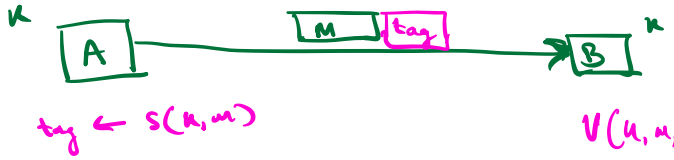
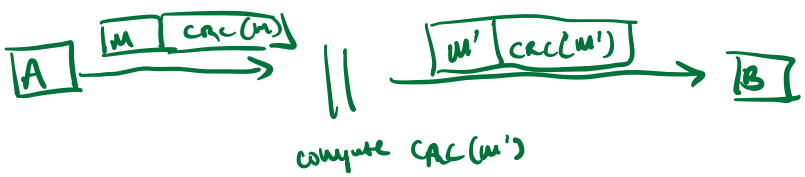
Message Integrity

→ Want to make sure messages over a network are not tampered with



→ used for integrity, not privacy
→ requires secret key k

checksum (CRC): protects against random errors, not malicious errors



MAC (Message auth code)

Pair of Eff algs (S,V) over (K, M, T) s.t.

$S(k, m) \rightarrow t \in T$

$V(k, m, t) \rightarrow$ yes/no

st $\forall k \in K, \forall m \in M$: $V(k, m, S(k,m)) \rightarrow$ yes

Secure MAC

Power: chosen msg attack

for $m_1, m_2 \ldots \in M$ attacker gets

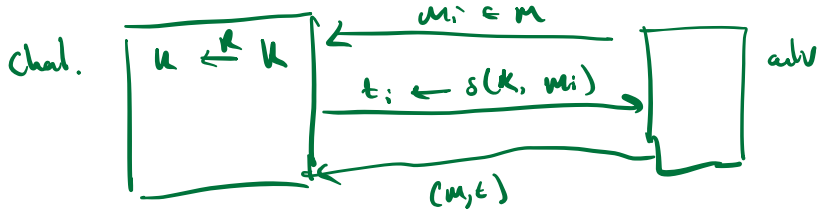
$t_i \leftarrow S(k, m_i)$ $i = 1, \ldots q$

Goal: existential forgery

(produce valid (m,t) pair $\notin \{(m_i, t_i)\}$)

attacker cannot produce a new valid (m,t)

For mac (S,V) and adv A:



A wins if $\begin{cases} V(k,m,t) = \text{yes}, & \text{and} \\ (m,t) \notin \{m_i, t_i\} \end{cases}$

Def. (S,V) is secure MAC if for all eff A:

$\text{Adv}_{MAC}[A, I] = \Pr[A \text{ wins}]$

is negligible

Constructions

Every PRF (where range is large) gives a secure MAC

Thm. If F is a secure PRF over (K, X, Y), where $1/|Y|$ is negligible, then I_F is a secure MAC.

$$\Pr[t = F(k,m)] = \frac{1}{|Y|}$$

so, ensure $|Y| > 2^{96}$

So, AES gives a secure MAC for 16-byte messages

Main question: small MAC $\Rightarrow$ Big MAC?

(actually: PRF w/ small domain $\Rightarrow$ PRF w/ large domain)

(AES)

Common constructions

1. CBC-MAC: used in banking (CMAC std.)
2. HMAC: used in Internet protocols (TLS)

} sequential

3. PMAC - parallel MAC - not widely used.

Remark: truncating MACs

Suppose MAC I_F built from PRF F, outputs w-bit tags ($Y = \{0,1\}^w$)

OK to truncate MAC to output MAC to w/n bits as long as 2^w is considered negligible

(truncating secure PRF is also secure PRF).

ex. Suppose I_F is a MAC that produces 2-bit tags

$\Rightarrow$ adv randomly guesses, prob 1/4 of breaking

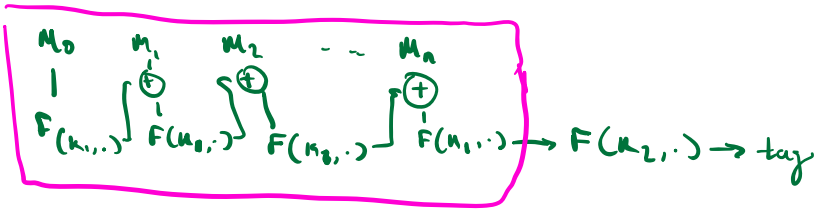
ex.2 Galileo (European GNSS)

uses 32-bit MAC, adv can forge tags w/prob $1/2^{32}$

Encrypted CBC-MAC

Let F be PRF over (K, X, X) where $X = \{0,1\}^n$

Define new PRF F_{CBC} (also MAC):



Raw CBC

F_{CBC} is a PRF over $(K^2, X^{\leq L}, X)$

$X^{\leq L}$: tag space

CBC-MAC thm

for every $L > 0$

F secure PRF $\Rightarrow$ F_{BC} secure PRF

In particular, for every q-query adv A attacking F_{CBC}, there is an adv B (same runtime) where

$$PRF_{adv}[A, F_{CBC}] \leq PRF_{adv}[B, F] + \frac{q^2 L^{O(1)}}{|X|}$$

$\Rightarrow$ CBC-MAC secure as long as

$$\frac{q^2}{|X|} \leq \text{negligible} \iff q \ll \sqrt{|X|} \quad (2^{64})$$

Why last step? Raw_{CBC} is insecure

Adv A:
1. choose $m \in X$
2. request tag for m

   Get $t \leftarrow raw_{CBC}(k, m)$

   $= F_k(k, m)$
3. output (msg, tag) forgery

   where $\begin{cases} msg = (m, t \oplus m) \in X^2 \\ tag = t \end{cases}$

Then:

$$Raw_{CBC}(k, (m, t \oplus m)) =$$

$$= F(k, F(k, m) \oplus (t \oplus m)) =$$

$$= F(k, m) = t$$

$\Rightarrow$ existential forger for Raw CBC

Note: Raw CBC is secure for fixed-size messages.

CBC-MAC padding

What if msg len is not multiple of block size?

Can't pad w/ 0's

attack: ask for tag for 14 char msg

⇒ get tag t

⇒ now, adv has tag for $\underbrace{m \| 0}_{\text{15 bytes}} = m'$

Instead: use one-to-one padding fns.

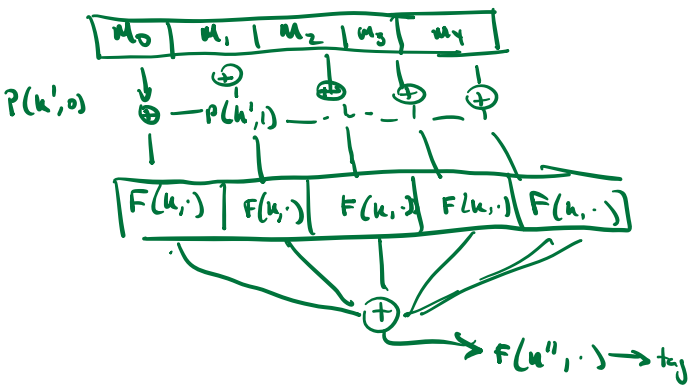
pad w/ "100...0" to mult of 16 bytes

add dummy block of $\underbrace{100\cdots0}_{\text{16 bytes}}$ to pad if needed

or- CMAC - clever padding (randomized) or never add dummy block.

Problem: CBC-MAC is sequential

Better: parallel-MAC (PMAC)



$P(k',\cdot)$: easy to compute