Problem:

"previously solve eavesdropping, how about message integrity, i.e. messages not changed en-route?"

↳ active adversay

↳ only focus on msg integrity, not secrecy.

threat model

Motivation:

e.g 1 : financial news / stock quotes over internet

e.g.2 : application program integrity check against virus malicious modification.

High-level claim:

"message integrity between communication parties requires that the sending party has a secret key unknown to the adversary"

keyless integrity mechanism → detect random (transmission) errors, NOT malicious errors.

Message Authentication Code (MAC)

- **Def:** a MAC system $I = (S, V)$ over (K, M, T)

  S: signing algorithm (probabilistic)

  V: verification algorithm (deterministic)

  $t \xleftarrow{R} S(K, m)$

  $r \leftarrow V(K, m, t)$: reject or accept

NOTE: deterministic MAC: given $K, m \longrightarrow$ unique tag t.

randomized MAC: $\longrightarrow$ many possible tags.

↳ !! NOTE: randomized MAC is not necessary for security, but yields better efficiency / security trade-offs.

- **Secure MAC (attack game)** —— "chosen message attack"

chal. $K \xleftarrow{R} K$ | adv (Q-query adv.)

m_i →

← $t_i \leftarrow S(K, m_i)$ (Q times)

(m, t) ⇐ existential forgery

(i.e. $(m, t) \notin \{(m_1, t_1), (m_2, t_2), \ldots\}$)

NOTE:

- Secure deter. MAC ⇔ unpredictable $S(K, \cdot)$

  randomized ⇔ ..... & difficult to produce another valid t' of an old signed msg m.

- Security for randomized MAC is a stronger assumption.

- **MAC Security (with verification queries)**

↳ either signing query $t_i \xleftarrow{R} S(K, m_i)$

or verification query $V(\hat{m}_j, \hat{t}_j)$

NOTE:

$$MAC^{vq}adv[A, I] \leq MACadv[B, I] \cdot Q_v$$

both definitions are equivalent (only a factor on error term)

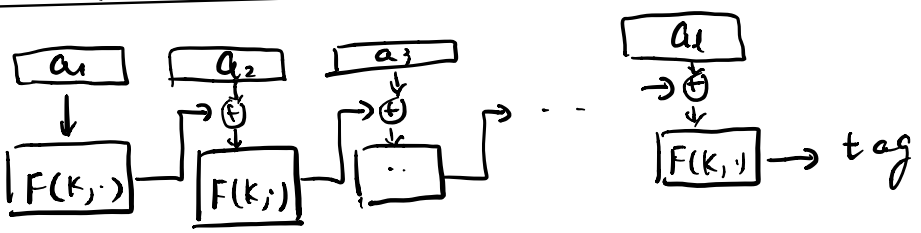
• Constructing MAC from PRF.

↳ any secure PRF is also a secure MAC.

$$\text{MACadv}[A, I] \le \text{PRFadv}[B, F] + \frac{1}{|y|}$$

• Prefix-free PRF for long msgs.

CBC prefix-free secure PRF:



NOTE: diff between F_{cbc} (above) and CBC mode encryption

1. F_{cbc} doesn't output intermediate value along the CBC chain
2. F_{cbc} uses fixed IV, CBC encryption uses a random IV per msg.

$$\text{PRF}^{\text{Pf}}\text{adv}[A, F_{cbc}] \le \text{PRFadv}[B, F] + \frac{(ql)^2}{2|x|}$$

visualize "prefix-free":

$$\text{root} \xrightarrow{a_1} p_1 \xrightarrow{a_2} p_2 \xrightarrow{a_3} p_3 \cdots \xrightarrow{a_v} p_v$$

m and m' both start at the root, not prefix of the other, but could share common initial subpath.

cascade prefix-free secure PRF:

$$\text{PRF}^{\text{Pf}}\text{adv}[A, F^*] \le ql \cdot \text{PRFadv}[B, F]$$

(an implicit factor of q hiding)

Extension Attack: CBC & cascade are insecure MACs:

cascade: $t' := F^*(k, m \| m')$ w/o knowledge of k.

CBC: let $a_2 = a_1 \oplus F(k, a_1)$

• fully secure PRF —— encrypted PRF

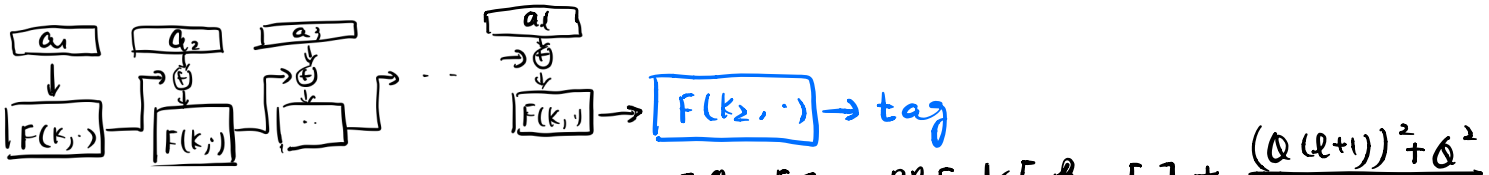
$$EF((k_1, k_2), m) := F(k_2, PF(k_1, m))$$

$$\text{PRFadv}[A, EF] \le \text{PRFadv}[B_1, F] + \text{PRF}^{\text{Pf}}\text{adv}[B_2, PF] + \frac{q^2}{2|y|}$$

↳ EF is secure PRFs if PF is prefix-free PRF & extendable PRF

if $PF(k, x) = PF(k, y)$, then $PF(k, x\|a) = PF(k, y\|a)$

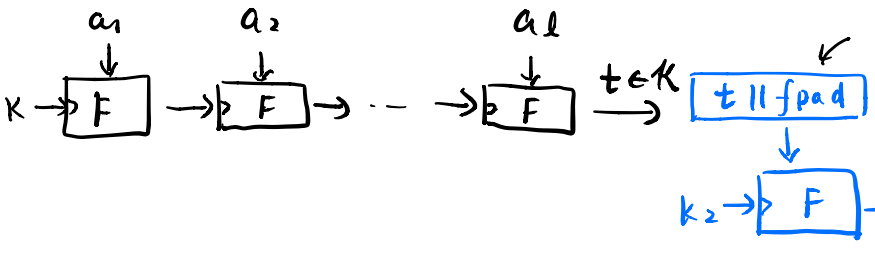
↳ Encrypted CBC (ECBC)



$$PRFadv[A, ECBC] \le PRFadv[B_1, F] + PRFadv[B_2, F] + \frac{(Q(l+1))^2 + Q^2}{2|X|}$$

NOTE: when $Q \approx 2^{n/2}$, attackers break this PRF w/ constant adv.

↳ Nested MAC (NMAC)



$|fpad| = n - K$ bits

$$PRFadv[A, NMAC] \le Q(l+1) \cdot PRFadv[B_1, F] + PRFadv[B_2, F] + \frac{Q^2}{2|K|}$$

NOTE: ECBC & NMAC are streaming MAC since the length of msg doesn't have to be known ahead of time.

- Fully Secure PRF — prefix free encoding

↳ prepend length: $pf(m) := (\langle v \rangle, a_1, \cdots a_v) \in X^{\le l}_{>0}$

↳ not streaming MAC

↳ stop bit: $pf(m) := ((a_1 \| 0), (a_2 \| 0), \cdots, (a_v \| 1))$

↳ increase the length of msg to MAC by v bits.

- Fully Secure PRF — CMAC (randomize prefix free encoding)

randomized ϵ-prefix-free: $\Pr[rpf(K, m_0) \sim rpf(K, m_1)] \le \epsilon$
prob. over choice of K in $\mathcal{K}$.

↳ Simple rpf:

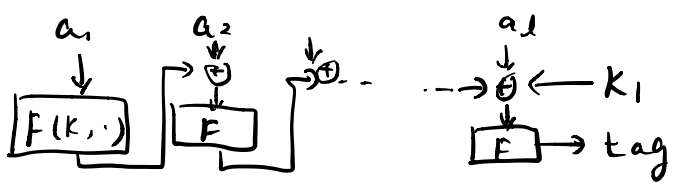
$rpf(k, (a_1, \cdots, a_v)) := (a_1, \cdots, a_{v-1}, (a_v \oplus k))$

$\Rightarrow F((k, k_1), m) := PF(k, rpf(k_1, m))$

$$PRFadv[A, F] \le PRF^{pf}adv[B_1, PF] + PRF^{pf}adv[B_2, PF] + Q^2\epsilon/2$$

because security of ϵ-rpf depends on $PRF^{pf}adv[B, PF]$

↳ rpf applied to CBC:



Personal NOTE : error term heuristic (mental shortcut)

$\frac{Q^2}{2|y|}$: birthday paradox

$\frac{(Ql)^2}{2|y|}$: chained / cascading w/ Difference Lemma

($|Pr[z]| \leq \frac{B^2}{2|X|}$ B : number of distinguished pair)

↳ also note that, this can be an implicit term sometimes.

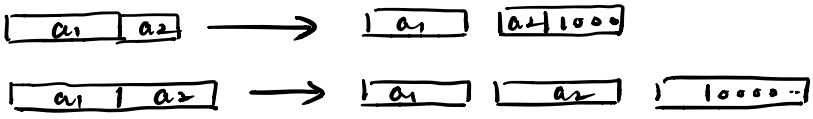
- Converting block-wise PRF ⟶ bit-wise PRF.

$F_{bit}(k, x) := F(k, inj(x))$

$inj: \{0,1\}^{\leq nl} \longrightarrow X^{\leq l+1}$ is an injective function

NOTE: padding ⊆ injective function

E.g.:



NOTE: ① padding can't be all 0, otherwise vulnerable to existential forgery

② injective function MUST expand.

NOTE: "truncating a secure PRF has NO effect on security of PRF, however affects the derived MAC"

↳ affects "existential forgery", NOT "semantic security"

- CMAC — cipher-based MAC
  - variant of EMAC, adopted by NIST
  - uses randomized prefix-free encoding
    - ↳ 2 keys to avoid dummy block
  - best approach to build bit-wise secure PRF from CBC^{Pf} PRF.

2 steps ⟶ subkey generation $K_0, K_1, K_2 \xleftarrow{R} KeyGen(K)$

⟶ compute MAC

rpf encoding:

input : $m \in \mathcal{M}$, $(k_1, k_2) \in X^2$

if $|m|$ is not a positive multiple of n :

$u \leftarrow |m| \bmod n$

partition m into a sequence of bit strings

$m = a_1 \| \cdots \| a_v$

($a_1, a_2, \cdots a_{v-1}$ are all n-bit strings)

if $|m|$ is positive multiple of n

  output : $(a_1, \cdots, a_{v-1}, \boxed{a_v \oplus K_1})$ → use 2 keys to avoid dummy block.

else

  output : $(a_1, \cdots, a_{v-1}, \boxed{a_v \| 1 \| 0^{n-u-1} \oplus K_2})$

subkey generation:

input : $K \in \mathcal{K}$

output: $K_0, K_1, K_2 \in \mathcal{X}$

$K_0 \leftarrow K$

$L \leftarrow F(K, 0^n)$

if $msb(L) = 0$:

  $K_1 \leftarrow (L \ll 1)$

else

  $K_1 \leftarrow (L \ll 1) \oplus R_n$

if $msb(K_1) = 0$:

  $K_2 \leftarrow (K_1 \ll 1)$

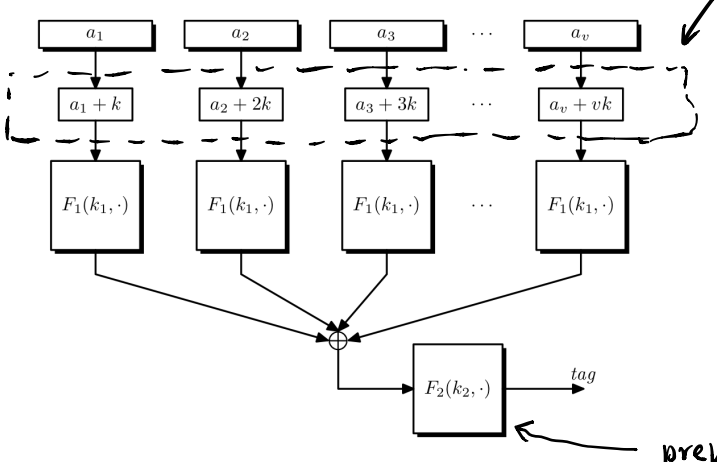
else

  $K_2 \leftarrow (K_1 \ll 1) \oplus R_n$

R_n is a fixed parameter, for $R_{128} := 0^{120}10000111$

Q: why related keys still result in secure MAC?

- PMAC



prefix-free encoding (masking)

$\mathrm{PRFadv}[A, \mathrm{PMAC}_0] \le$

$\mathrm{PRFadv}[B_1, F_1] +$

$\mathrm{PRFadv}[B_2, F_2] +$

$\frac{Q^2}{2|y|} + \frac{(Ql)^2}{2p}$

prevent existential forgery

Figure 6.9: PMAC_0 construction

↑ in $\mathbb{Z}_p$.

PMAC better than PMAC$_0$:

↳ use $GF(2^n)$ instead of $\mathbb{Z}_p$
- elements represented as n-bit string
- addition is bit-wise XOR
- $F_1 = F_2 = F$

↳ $\gamma_i \cdot K$: all γ_i are fixed constant and specially chosen, s.t. computing $\gamma_{i+1} \cdot K$ from $\gamma_i \cdot K$ is cheap.

↳ $K \leftarrow F(k_1, 0^n)$ $\Rightarrow$ PMAC only uses 2 keys.
$k_2 \leftarrow k_1$

↳ PMAC uses trick to save 1 application of F

↳ PMAC uses a variant of the CMAC prf to provide bit-wise PRF.

NOTE : ① on sequential machine. PMAC is as efficient as ECBC & NMAC
② PMAC$_0$ is incremental