Message integrity

Message Auth. Codes

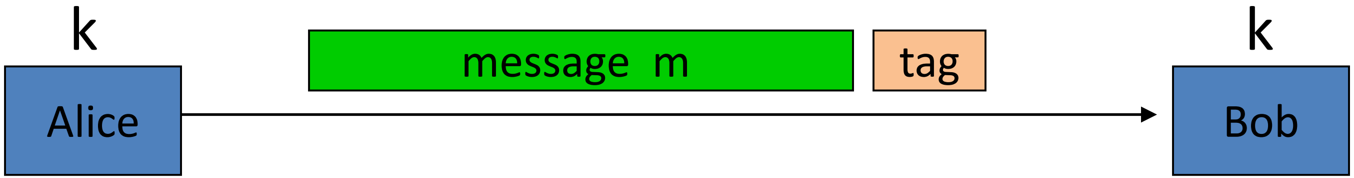
Message Integrity

Goal: **integrity**, no confidentiality.

Examples:

- Protecting public binaries on disk.
- Protecting banner ads on web pages.

Message integrity: MACs



Generate tag:

tag ← S(k, m)

Verify tag:

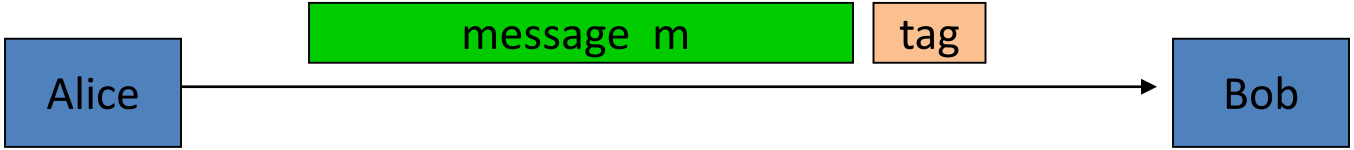
V(k, m, tag) $\stackrel{?}{=}$ `yes'

Def: **MAC** I = (S,V) defined over (K,M,T) is a pair of algs:

– S(k,m) outputs t in T

– V(k,m,t) outputs `yes' or `no'

Integrity requires a secret key



Generate tag:

tag ← CRC(m)

Verify tag:

V(m, tag) $\stackrel{?}{=}$ `yes'

- Attacker can easily modify message m and re-compute CRC.
- CRC designed to detect **random**, not malicious errors.

Secure MACs

Attacker's power: **chosen message attack**

- for $m_1, m_2, \ldots, m_q$ attacker is given $t_i \leftarrow S(k, m_i)$

Attacker's goal: **existential forgery**

- produce some <u>**new**</u> valid message/tag pair (m,t).

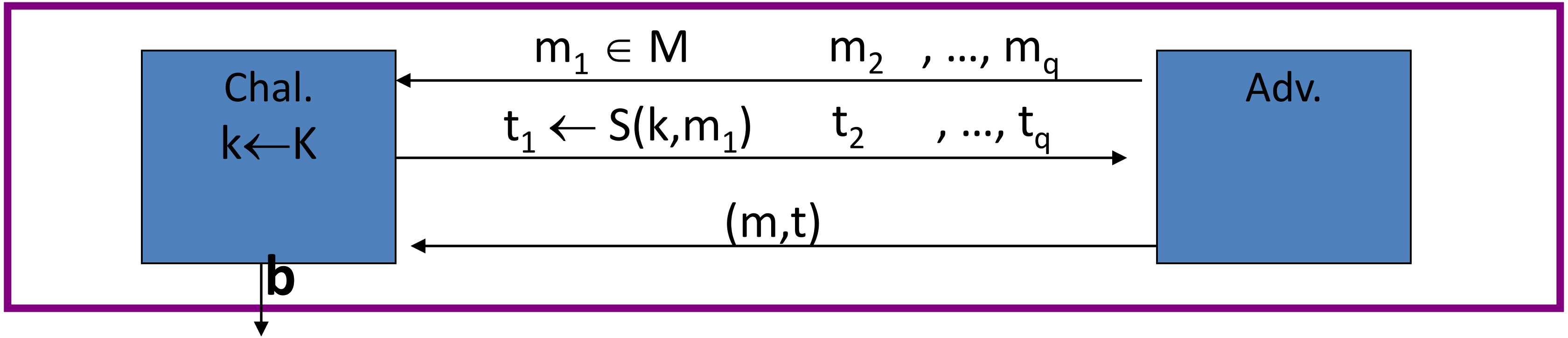
$$(m,t) \notin \{ (m_1,t_1), \ldots, (m_q,t_q) \}$$

⇒ attacker cannot produce a valid tag for a new message

⇒ given (m,t) attacker cannot even produce (m,t') for $t' \neq t$

Secure MACs

- For a MAC I=(S,V) and adv. A define a MAC game as:



$$\begin{cases} \mathbf{b}=1 & \text{if } V(k,m,t) = \text{`yes'} \text{ and } (m,t) \notin \{(m_1,t_1), \ldots, (m_q,t_q)\} \\ \mathbf{b}=0 & \text{otherwise} \end{cases}$$

Def: I=(S,V) is a **secure MAC** if for all "efficient" A:

$Adv_{MAC}[A,I]$ = Pr[Chal. outputs 1] is "negligible."

Let I = (S,V) be a MAC.

Suppose an attacker is able to find $m_0 \neq m_1$ such that

$$S(k, m_0) = S(k, m_1) \quad \text{for } ½ \text{ of the keys } k \text{ in } K$$

Can this MAC be secure?

- ○ Yes, the attacker cannot generate a valid tag for m_0 or m_1
- ⇒ ○ No, this MAC can be broken using a chosen msg attack
- ○ It depends on the details of the MAC
- ○

$Adv[A, I] = 1/2$

Let I = (S,V) be a MAC.

Suppose S(k,m) is always 5 bits long

Can this MAC be secure?

- ⇒ ○ No, an attacker can simply guess the tag for messages
- ○ It depends on the details of the MAC
- ○ Yes, the attacker cannot generate a valid tag for any message
- ○

$Adv[A,I] = 1/32$

Example: protecting system files

Suppose at install time the system computes:

filename F_1	filename F_2	…	filename F_n
$t_1 = S(k,F_1)$	$t_2 = S(k,F_2)$		$t_n = S(k,F_n)$

k derived from user's password

Later a virus infects system and modifies system files

User reboots into clean OS and supplies his password

– Then: secure MAC $\Rightarrow$ all modified files will be detected

End of Segment

Message Integrity

MACs based on PRFs

Review: Secure MACs

MAC: signing alg. **$S(k,m) \longrightarrow t$** and verification alg. **$V(k,m,t) \longrightarrow 0,1$**

Attacker's power: **chosen message attack**

- for $m_1,m_2,\ldots,m_q$ attacker is given $t_i \leftarrow S(k,m_i)$

Attacker's goal: **existential forgery**

- produce some **new** valid message/tag pair (m,t).

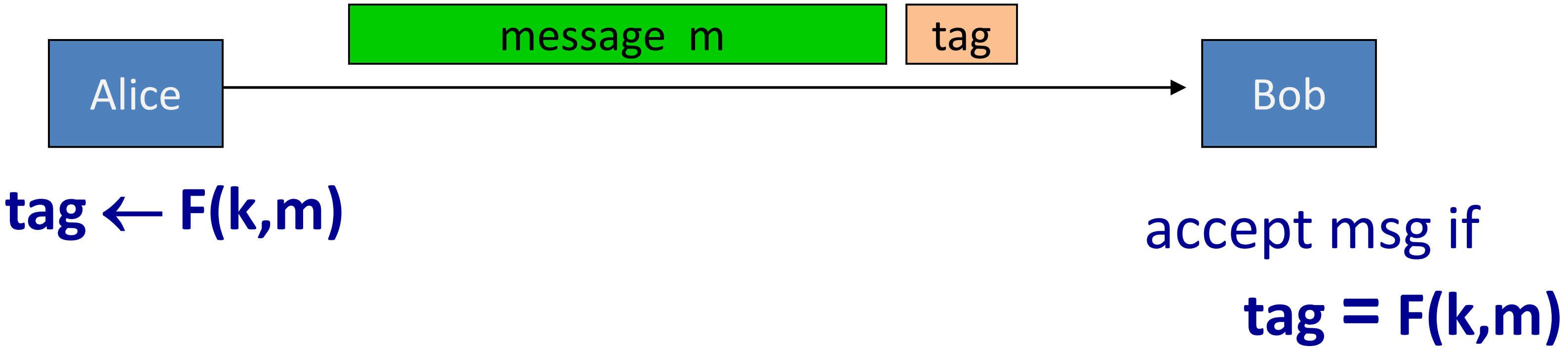
$$(m,t) \notin \{ (m_1,t_1), \ldots, (m_q,t_q) \}$$

$\Rightarrow$ attacker cannot produce a valid tag for a new message

Secure PRF $\Rightarrow$ Secure MAC

For a PRF $F: K \times X \longrightarrow Y$ 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.



A bad example

Suppose **F: K × X ⟶ Y** is a secure PRF with $Y = \{0,1\}^{10}$

Is the derived MAC I_F a secure MAC system?

- ○ Yes, the MAC is secure because the PRF is secure
- ○ No tags are too short: anyone can guess the tag for any msg
- ○ It depends on the function F
- ○

$\text{Adv}[A, I_F] = 1/1024$

Security

Thm: If $\mathbf{F: K \times X \longrightarrow Y}$ is a secure PRF and $1/|Y|$ is negligible (i.e. $|Y|$ is large) then I_F is a secure MAC.

In particular, for every eff. MAC adversary A attacking I_F there exists an eff. PRF adversary B attacking F s.t.:

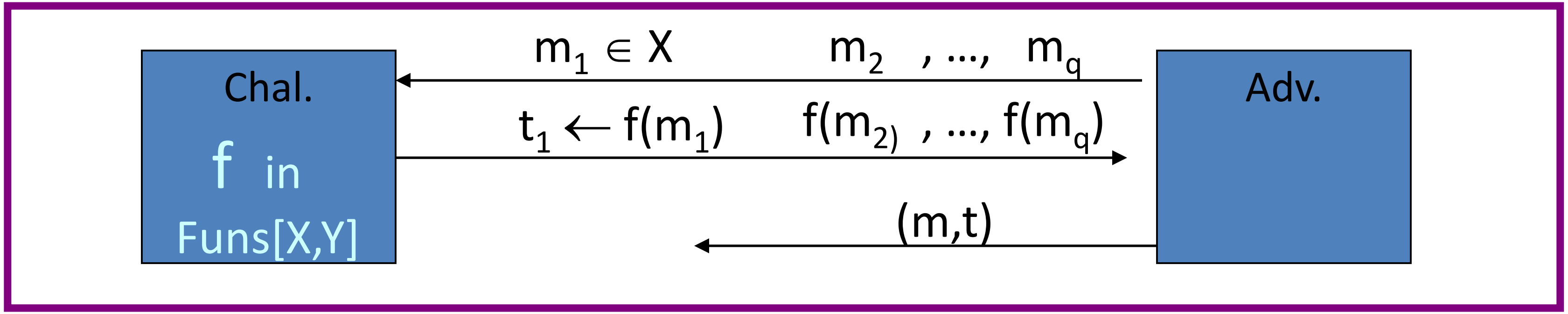
$$\text{Adv}_{\text{MAC}}[A, I_F] \leq \text{Adv}_{\text{PRF}}[B, F] + 1/|Y|$$

$\Rightarrow$ I_F is secure as long as $|Y|$ is large, say $|Y| = 2^{80}$.

Proof Sketch

Suppose $f: X \longrightarrow Y$ is a truly random function

Then MAC adversary A must win the following game:



A wins if $t = f(m)$ and $m \notin \{ m_1 , \ldots , m_q \}$

$\Rightarrow$ $\Pr[\text{A wins}] = 1/|Y|$ same must hold for $F(k,x)$

Examples

- AES: a MAC for 16-byte messages.

- Main question: how to convert Small-MAC into a Big-MAC ?

- Two main constructions used in practice:
  - **CBC-MAC** (banking – ANSI X9.9, X9.19, FIPS 186-3)
  - **HMAC** (Internet protocols: SSL, IPsec, SSH, …)

- Both convert a small-PRF into a big-PRF.

Truncating MACs based on PRFs

Easy lemma: suppose $F: K \times X \longrightarrow \{0,1\}^n$ is a secure PRF.

Then so is $F_t(k,m) = F(k,m)[1...t]$ for all $1 \leq t \leq n$

first t-bit of output

$\Rightarrow$ if (S,V) is a MAC is based on a secure PRF outputting n-bit tags

the truncated MAC outputting w bits is secure

... as long as $1/2^w$ is still negligible (say $w \geq 64$)

End of Segment

Message Integrity

CBC-MAC and NMAC

MACs and PRFs

Recall: secure PRF **F** ⇒ secure MAC, as long as |Y| is large

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

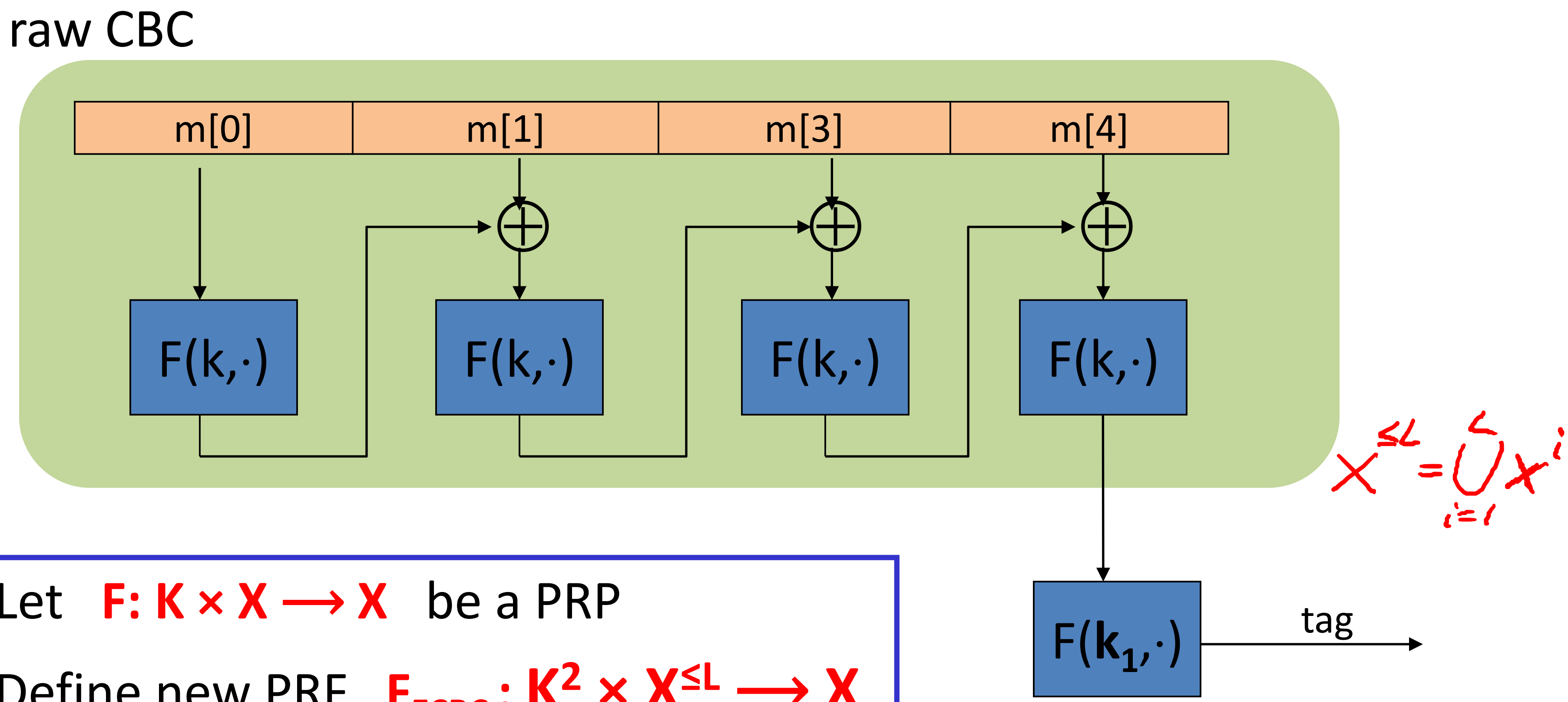
Our goal:

given a PRF for short messages (AES)

construct a PRF for long messages

From here on let $X = \{0,1\}^n$ (e.g. $n=128$)

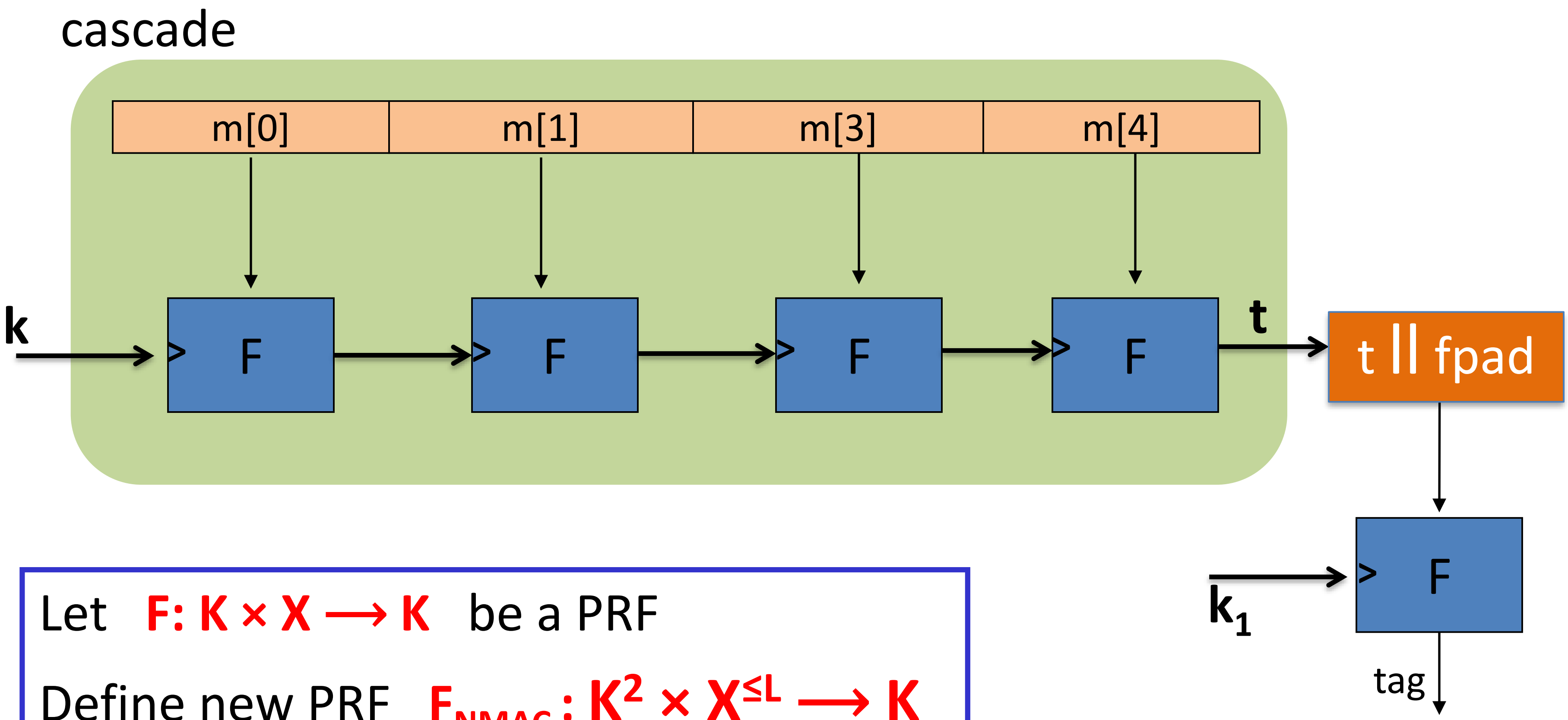
Construction 1: encrypted CBC-MAC



Let $\mathbf{F: K \times X \longrightarrow X}$ be a PRP

Define new PRF $\mathbf{F_{ECBC} : K^2 \times X^{\leq L} \longrightarrow X}$

Construction 2: NMAC (nested MAC)



Let $F: K \times X \longrightarrow K$ be a PRF

Define new PRF $F_{NMAC} : K^2 \times X^{\leq L} \longrightarrow K$

Why the last encryption step in ECBC-MAC?

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 msg attack.

Adversary works as follows:

- Choose an arbitrary one-block message $m \in X$
- Request tag for m. Get $t = F(k,m)$
- Output t as MAC forgery for the 2-block message $(m, t \oplus m)$

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

Why the last encryption step in ECBC-MAC and NMAC?

NMAC: suppose we define a MAC $I = (S,V)$ where

$$S(k,m) = \text{cascade}(k, m)$$

- This MAC is secure
- This MAC can be forged without any chosen msg queries
- This MAC can be forged with one chosen msg query
- This MAC can be forged, but only with two msg queries

$\text{Cascade}(k,m) \Rightarrow \text{cascade}(k, m\|w)$ for any w

Prefix-free secure PRF

rawCBC and cascade are prefix-free secure PRFs

i.e., secure PRFS if no message is a prefix of another

ECBC-MAC and NMAC Security

rawCBC/cascade are <u>prefix-free secure</u> and <u>extendable</u> PRF + their output is encrypted by a secure PRF

Extendable PRF:

$$\forall x,y,w: \quad F(k, x) = F(k, y) \quad \Rightarrow$$

$$F(k, \mathbf{x \| w}) = F(k, \mathbf{y \| w})$$

ECBC-MAC and NMAC analysis

<u>Theorem</u>: For any L>0,

For every eff. q-query PRF adv. A attacking F_{ECBC} or F_{NMAC}

there exists an eff. adversary B s.t.:

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

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

CBC-MAC is secure as long as $q \ll |X|^{1/2}$

NMAC is secure as long as $q \ll |K|^{1/2}$ (2^{64} for AES-128)

An example

$$\mathrm{Adv}_{PRF}[A, F_{ECBC}] \leq \mathrm{Adv}_{PRP}[B, F] + \mathbf{2\,q^2 / |X|}$$

q = # messages MAC-ed with k

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

- AES: $|X| = 2^{128}$ $\Rightarrow$ $q < 2^{48}$

  So, after 2^{48} messages must, must change key

- 3DES: $|X| = 2^{64}$ $\Rightarrow$ $q < 2^{16}$

The security bounds are tight: an attack

After signing $|X|^{1/2}$ messages with ECBC-MAC or
$|K|^{1/2}$ 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:

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

The security bounds are tight: an attack

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

$$F_{BIG}(k, x) = F_{BIG}(k, y) \quad \Rightarrow \quad F_{BIG}(k, \mathbf{x \| w}) = F_{BIG}(k, \mathbf{y \| w})$$

Generic attack on the derived MAC:

step 1: issue $|Y|^{1/2}$ message queries for rand. messages in X.

obtain (m_i, t_i) for $i = 1 ,\ldots, |Y|^{1/2}$

step 2: find a collision $t_u = t_v$ for $u \neq v$ (one exists w.h.p by b-day paradox)

step 3: choose some w and query for $t := F_{BIG}(k, \mathbf{m_u \| w})$

step 4: output forgery $(\mathbf{m_v \| w,\ t})$. Indeed $t := F_{BIG}(k, \mathbf{m_v \| w})$

Comparison

ECBC-MAC is commonly used as an AES-based MAC

- CCM encryption mode (used in 802.11i)
- NIST standard called CMAC

NMAC not usually used with AES or 3DES

- Main reason: need to change AES key on every block
  requires re-computing AES key expansion
- But NMAC is the basis for a popular MAC called HMAC (next)

End of Segment