MACs

HW4 released

(Msg Authentication Codes)

Goal: CCA secure scheme

MAC syntax:

- KeyGen : outputs key k
- $MAC(k, m)$: outputs a "tag" / MAC

Idea: Someone who doesn't know k can't produce the MAC of a new msg "forgery" even after seeing many MACs of chosen msgs.

Note: Authenticity vs Privacy (hiding info)

Security Def:

"real MAC"

```
k ← KeyGen

GETMAC(m):
    $t = MAC(k, m)$
    return t

VER(m, t):
    ret $t \overset{?}{=} MAC(k, m)$
```

$\approx$

"fake MAC"

```
k ← KeyGen
SEEN = ∅

GETMAC(m):
    $t = MAC(k, m)$
    add (m, t) to SEEN
    return t

VER(m, t):
    ret $(m, t) \overset{?}{\in} SEEN$
```

Discuss:

libraries hide internal differences

libraries hide whether MAC is actually checked for Adv-generated (m, t)

⇒ only way to get diff behavior from libs is to find (m,t) not generated by lib and yet $t = MAC(k,m)$] forgery

"real" lib says $VER(m,t) = 1$

"fake" lib says $VER(m,t) = 0$

MAC constructions

▶ $MAC(k,m) = PRF(k,m)$

good MAC scheme ✓
but for short m

idea: If you never query $F(k,m^*)$, then $F(k,m^*)$ "looks random", hence hard to guess

▶ MAC for longer msgs?

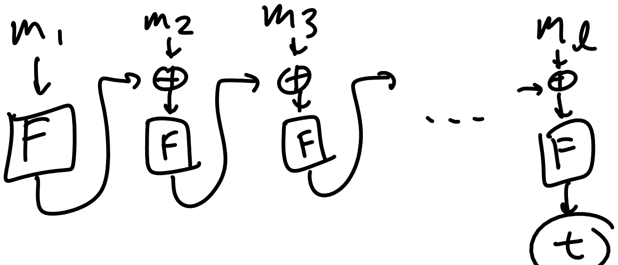
idea: $MAC(k, m_1 m_2 \cdots m_\ell) = F(k, \underline{m_1 \oplus \cdots \oplus m_\ell})$

Obs: If $m_1 \oplus m_2 \oplus \cdots \oplus m_\ell = m_1' \oplus m_2' \oplus \cdots$ then these 2 msgs have same MAC

Attack:	"real"	"fake"
$t = GETMAC(0^\lambda 1^\lambda)$ return $VER(1^\lambda 0^\lambda, t)$	$t = MAC(0^\lambda 1^\lambda)$ $= MAC(1^\lambda 0^\lambda)$ returns true	$(1^\lambda 0^\lambda, t)$ $\notin$ SEEN ⇒ returns false

CBC-MAC:

$MAC(k, m_1 \cdots m_\ell) =$



CBC mode, but no IV,
▶ only last block

Disclaimer: CBC-MAC is secure on msgs of a single length

CCA security

Encrypt - then - MAC