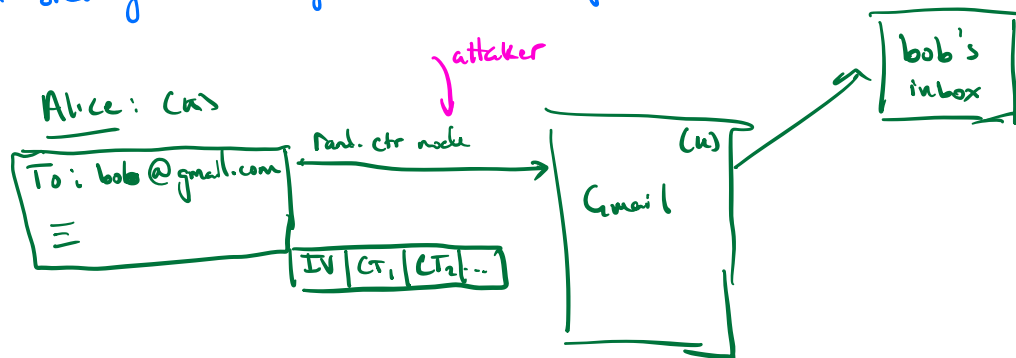


## Authenticated Encryption

Combine confidentiality and data integrity

ex. Breaking confidentiality without data integrity.



Attacker Mel:

$$\hat{ct}_1 \leftarrow ct_1 \oplus [\text{bitmask bob} \leftrightarrow \text{mel}]$$

send  $[IV, \hat{ct}_1, ct_2, ct_3, \dots]$  to gmail

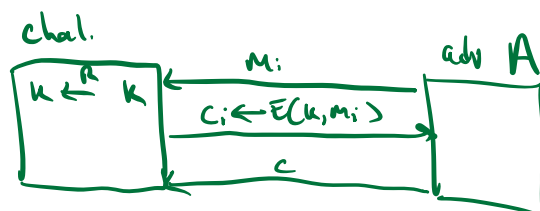
$\Rightarrow$  server routes message for Bob to mel

Lesson: never use a cipher that doesn't provide data integrity.

## Defining security

Goal: CPA security + ciphertext integrity.

C.I. game:



A wins game if  $D(k, c) \neq \text{"reject"}$   
and  $c \notin \{c_1, c_2, \dots\}$

Def.  $(E, D)$  provides auth enc.  
 if  $(E, D)$  is both CPA-secure  
 and has ciphertext integrity.

### Constructions using MAC and cipher

$I = (S, V)$  secure MAC

$(E, D)$  CPA secure cipher

$K = (K_e, K_m)$

Options:

1. MAC-then-enc (TLS 1.0)

enc:  $t \leftarrow S(K_m, m)$ ,  $c \leftarrow E(K_e, m \| t)$  send  $c$

dec: dec, check tag, output  $m$  if valid then reject.

Busted!  
(FLOOD) attack

2. enc-then-MAC (GCM, IPsec)

enc:  $c \leftarrow E(K_e, m)$   $t \leftarrow S(K_m, c)$  send  $(c, t)$

dec: check tag, reject if invalid  
 otherwise dec, output  $m$

only use  
this

3. enc-and-MAC (SSH)

$t \leftarrow S(K_m, m)$   $c \leftarrow E(K_e, m)$  send  $(c, t)$

Busted! (if MAC leaks into about msg)

Thm. enc-then-MAC provides A.E.

if  $(E, D)$  is CPA secure

and  $(S, V)$  is secure MAC.

Note:  $K_e, K_m$  must be independent keys.

$K_e \leftarrow \text{PRF}(K, \text{"enc"})$

$K_m \leftarrow \text{PRF}(K, \text{"mac"})$