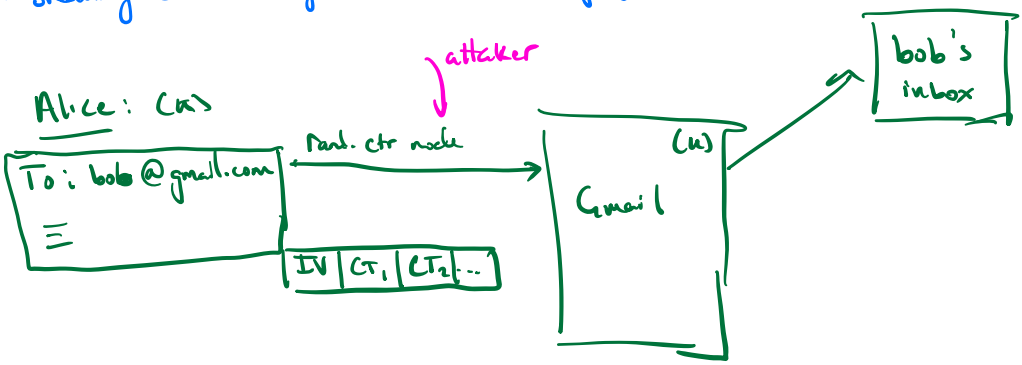
Authenticated Encryption

Combine confidentiality and data integrity

ex. Breaking confidentiality without data integrity.



Attacker Mal:

$\hat{ct}_1 \leftarrow ct_1 \oplus$ [bitmask bob→mal]

send [IV, $\hat{ct}_1$, ct2, ct3, ...] to gmail

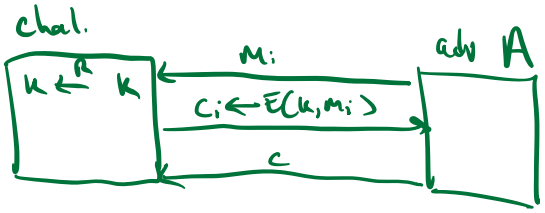
⇒ server routes message for Bob to Mal

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 $D(k, c) \neq$ "reject"

and $c \notin \{c_1, c_2, \ldots\}$

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 } Busted! (POODLE attack)

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

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

enc: $c \leftarrow E(k_e, m)$ $t \leftarrow S(k_m, c)$ send (c, t) } only use this

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

3. enc-and-MAC (SSH)

$t \leftarrow S(k_m, m)$ $c \leftarrow E(k_e, m)$ send (c, t)

Busted! (if MAC leaks info 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.

$k_e \leftarrow PRF(k, \text{"enc"})$

$k_m \leftarrow PRF(k, \text{"MAC"})$

Galois Counter Mode

(Nonce-based CTR mode enc)-then-MAC

Required for TLS 1.3

Auth Enc w/ Associated Data

BoringSSL GCM API: enc-then-mac

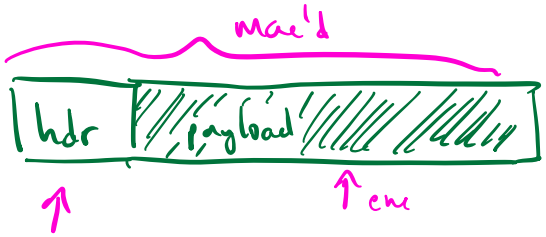
```
int encrypt (char *pt, int pt_len          // PT
             char *add, int add_length     // assoc data
             char * key,
             char * iv,
             ---------------------------
             char *ct,                     // output
             char *tag
            );
```

add': add assoc data

$C \leftarrow E(K_e, m)$, $t \leftarrow S(K_m, aad \| c)$

send (c, t)

aad': not part of ct, supplied to decrypt

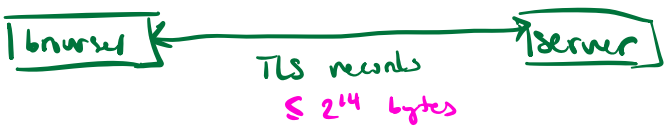


clear, for routing (aad)

decrypt dec': supply key, ct, assoc. data

(if tampered: dec fails)

Case study: GCM in TLS 1.3



Unidirectional keys: $k_{B \to S}$, $k_{S \to B}$

generated from a master secret (HKDF)

Stateful encryption: each side maintains 64-bit ctr

WSC_B, WSC_S : write seq counters

init to 0 at session setup

WSC_B ++ when browser sends record to server

server does same when record received

When browser sends record:

key = $k_{B \to S}$ = (enc key, mac key) (GCM key)

assoc data = (WSC_B, record type, "prot ver. 3.1")

nonce = $WSC_B \oplus$ (client_write_iv)

random, gen. at session setup

Transmitted record:

type	version (3.1)	length	GCM ct

Note: WSC_B not sent

WSC_B prevents replay (retransmit of record)