Special Topics in Cryptography

Mohammad Mahmoody

Last time

- Authentication (MAC) using shared keys
- Getting MACs from PRFs

Today

- How to combine CPA security + MACS:
- Security against active attacks (CCA security)

Authentication:

How would Bob know Alice sent this message?

Strong MAC = for all (m, k). there is a unique acceptable t

Last time

$t = F_k(m)$

Formal definition of security

game

The message authentication experiment $\mathsf{Mac\text{-}forge}_{\mathcal{A},\Pi}(n)$**:**

1. *A key* k *is generated by running* $\mathsf{Gen}(1^n)$.
2. *The adversary* $\mathcal{A}$ *is given input* 1^n *and oracle access to* $\mathsf{Mac}_k(\cdot)$. *The adversary eventually outputs* (m, t). *Let* $\mathcal{Q}$ *denote the set of all queries that* $\mathcal{A}$ *asked to its oracle.*
3. $\mathcal{A}$ succeeds *if and only if (1)* $\mathsf{Vrfy}_k(m, t) = 1$ *and (2)* $m \notin \mathcal{Q}$. *In that case the output of the experiment is defined to be 1.*

m → (m, t)

ADV can NOT ~~the~~ find (m, t') $t' \neq t$ that Verify(m, t').

def

DEFINITION 4.2 *A message authentication code* $\Pi = (\mathsf{Gen}, \mathsf{Mac}, \mathsf{Vrfy})$ *is* existentially unforgeable under an adaptive chosen-message attack, *or just* secure, *if for all probabilistic polynomial-time adversaries* $\mathcal{A}$, *there is a negligible function* negl *such that:*

$$\Pr[\mathsf{Mac\text{-}forge}_{\mathcal{A},\Pi}(n) = 1] \leq \mathsf{negl}(n).$$

Constructing MACs using PRFs

- Suppose $F_k(\cdot)$ is a PRF with key, input, output lengths: n, $*$, ℓ
- How do we generate MAC tags for messages?

- To tag message m, attach $t = F_k(m)$ to it.

$\ell = \ell g(n) \rightarrow 2^{\ell} = 2^{\ell g(n)} = n^{\ell g(n)}$

- The proof needs $\frac{1}{2^{\ell}}$ to be smaller than any $poly(n)$. We can let $\ell = n$

Block-cipher / AES, DES

What we have achieved.

- CPA-secure encryption based on PRFs.
  - Randomized.
  - Needs input of PRF to be large enough.

$Enc_k(m) \longrightarrow$ Pick r $[r, F_k(r) \oplus m]$

- MACs for authentication based on PRFs.
  - Deterministic
  - Needs output of PRF to be large enough.

$MAC_k(m) = t = F_k(m)$.

- PRFs could be obtained from:
  - Theory: PRGs and even one-way functions
  - Practice: Any "good" cryptographic hash function.

PRF

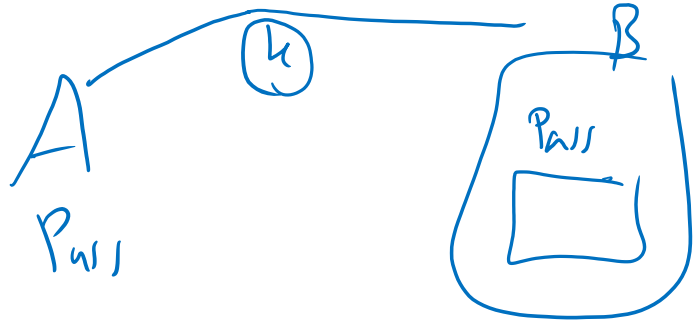
String Concaten

$H(k, x) = F_k(x)$

Chosen cipher-text security:

- Combining CPA security with MACs to handle active attacks.

Password verification example



- Alice (client) wants to login on Bob's computer (server)

Pass = $b_1 \dots b_{100}$

- Alice's browser has a shared key k with Bob
- Alice encrypts the password $pass$ using k and sends $c = \mathrm{Enc}_k(pass)$
- Bob decrypts c and if the password is correct it allows Alice to login.

we just need 100 tries to recover it.

- Issue: the fact that there is a "feedback" to modified messages given to "Alice" (or an adversary) might lead to recovering the full $pass$

$m = m_1 \dots m_k$

CPA secure Enc could be "resettable": there might be $\mathrm{Res}(c, i, b) \to c'$

$\to \mathrm{Dec}(c') \to m_1, m_2, \dots, m_i' = \dots m_k$ ($m_i' = \bar{b}$)

Idea: stronger security

how we encrypted $n+\ell$ bit → $2^{n+\ell}$

if we somehow encrypt by $2n+\ell$ → $2^{\ell+2n}$

- Hope: if c is an encryption, make any modification of c "useless". → not decryptable!

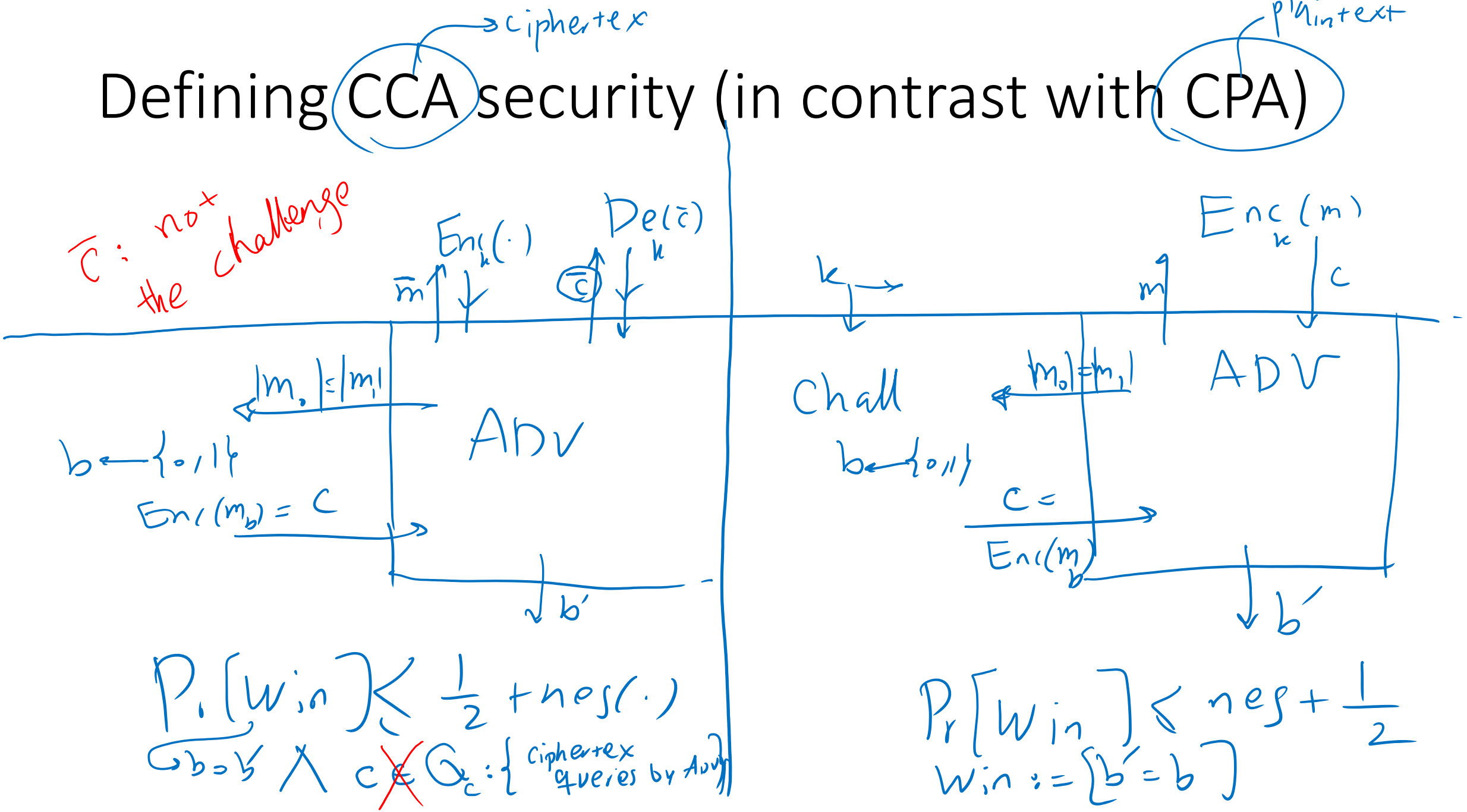
$Enc(k, m, r) \to c$

$Dec(k, c) \to m'$

Correctness: $\forall k, \forall m, \forall r$: $Enc(k,m,r) \to c$, $Dec(k,c) \to m'$, $m = m'$

Specific symbol : (1)

Defining CCA security (in contrast with CPA)



Exercise

- If S=(Enc,Dec) is CCA secure, then using scheme S for the application of encrypting passwords will be “safe” (will need to formalize it).

- Note: the specific attack we discussed does not work anymore if we use CCA secure: if one can “fix” the i’th bit to zero, it is NOT CCA secure

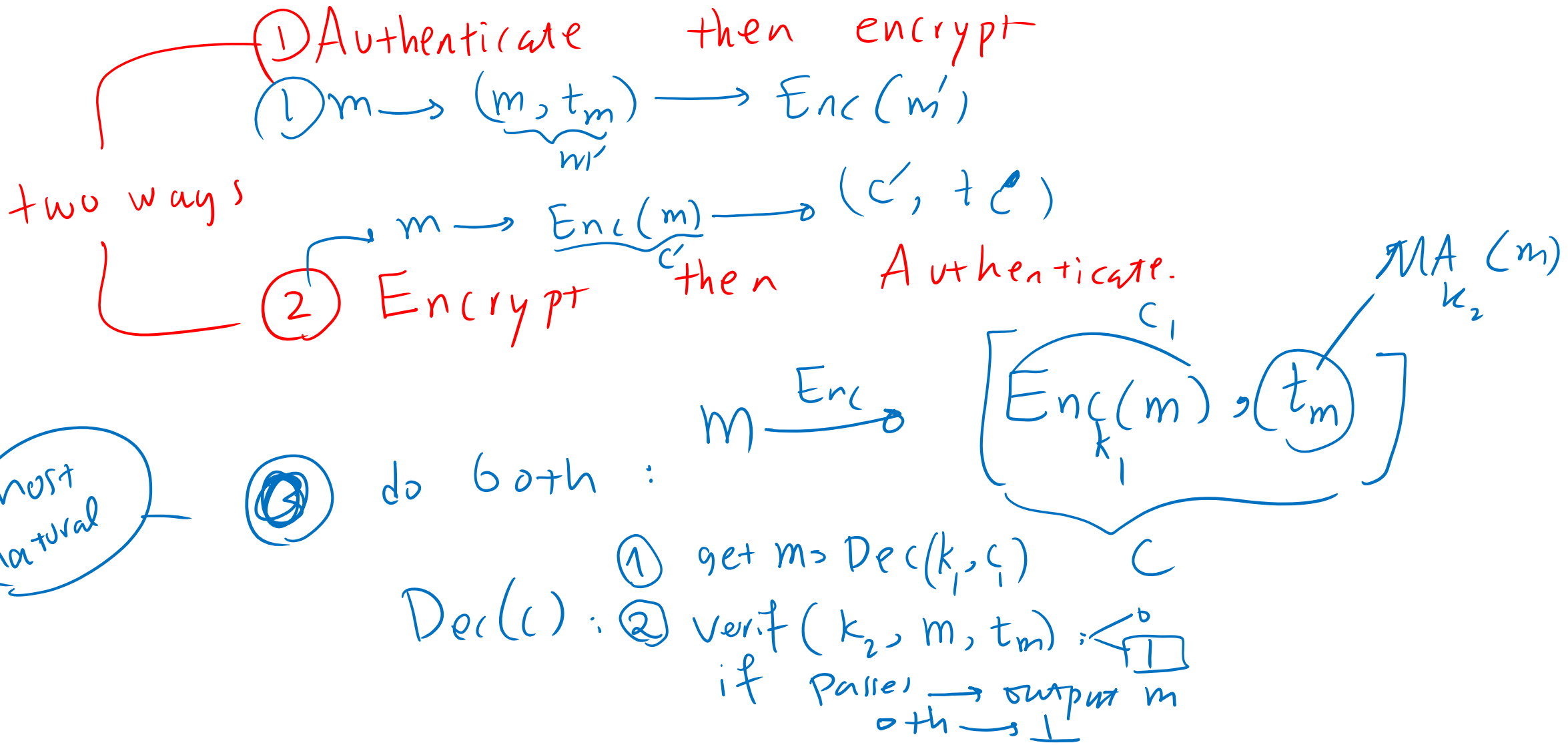
- More generally: if adversary modifies the ciphertext in any way, the decryptor will reject and output “error”.

CPA security + MAC to → CCA security

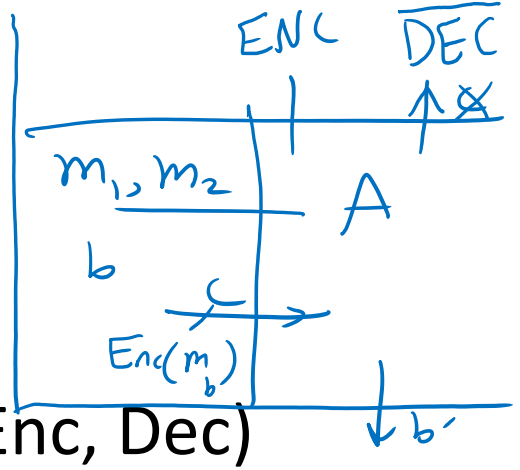
- We have:
- CPA-secure encryption: (Enc, Dec) based on key k_1
- Strongly-secure MAC: (Mac,Vrf) – based on key k_2

- We want:
- CCA secure encryption : (ENC, DEC)

What are most "natural" was to do it?



CPA security + MAC to → CCA security
1st (wrong) try



- Suppose k_1 is key for MAC and k_2 is key for CPA scheme (Enc, Dec)
- We want to encrypt message m

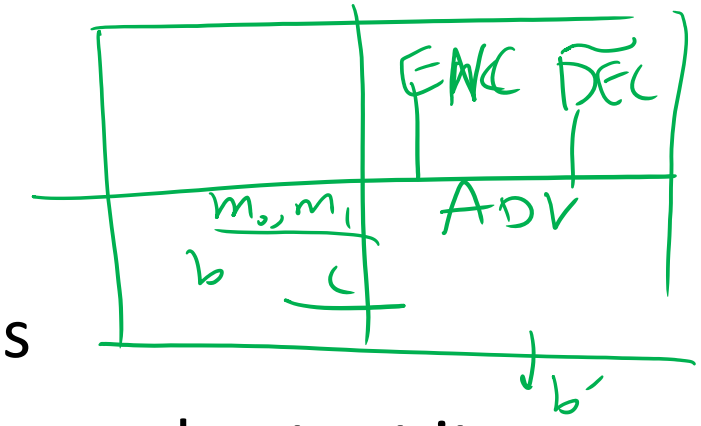
- First generate a tag using MAC: $t = \mathrm{MAC}_{k_1}(m)$
- Encrypt both of $[m, t]$ and get $C = \mathrm{Enc}_{k_2}([m, t])$
- To decrypt: First get back $[m, t]$ using $\mathrm{Dec}_{k_2}(C)$
- Then run $\mathrm{Verify}_{k_1}(m, t)$ and output $\perp$ if it rejects… otherwise output m

CPA security + MAC to → CCA security
2st (correct) way:

ENC

- First encrypt ad then authenticate
- $c = \mathrm{Enc}_{k_2}(m)$ and $tag = \mathrm{MAC}_{k_1}(c)$ and send $C = [c, tag]$

c



- Decryption:
- First run $\mathrm{Verify}_{k_1}(c, t)$ and output $\perp$ if it rejects
- If verify passes: then decrypt c using k_2 to get m and output it

Intuition: proving DEC is useless . . .
we will be back to CPA Security!

Proof of security (bit picture)

Construction $S = (ENC, DEC)$. Sec-game CCA-sec

2- Assumptions: $(Enc, Dec)_{S_1}$ —— CPA-sec

$(MAC, Ver)_{S_2}$ —— Strongly sec.

---

if A is a poly time attacker against S

winning by non-neg ε

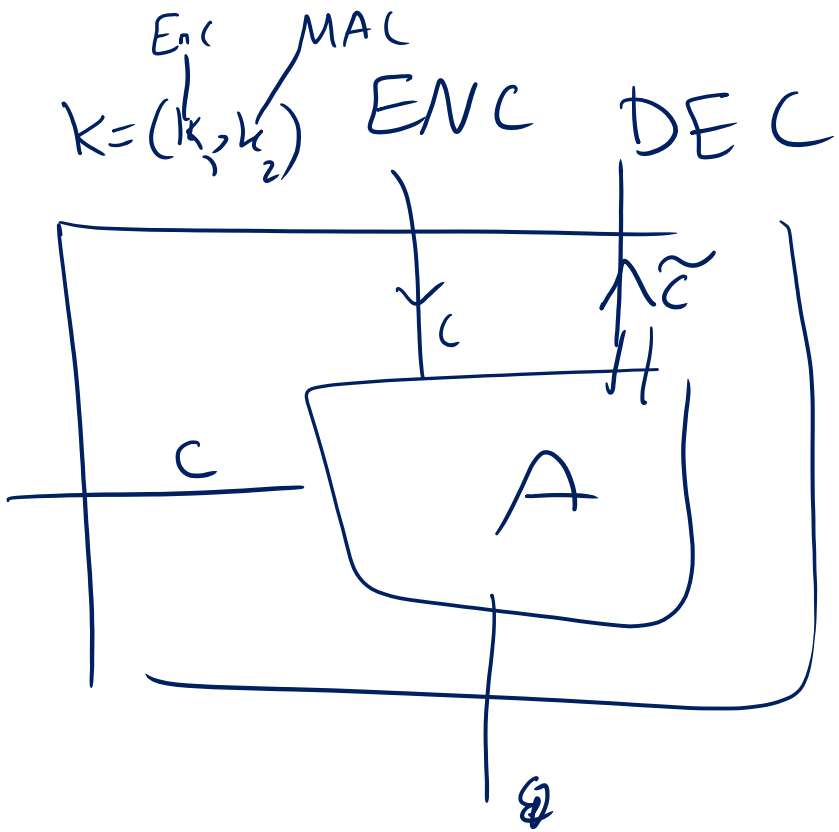
↳ Either of these:

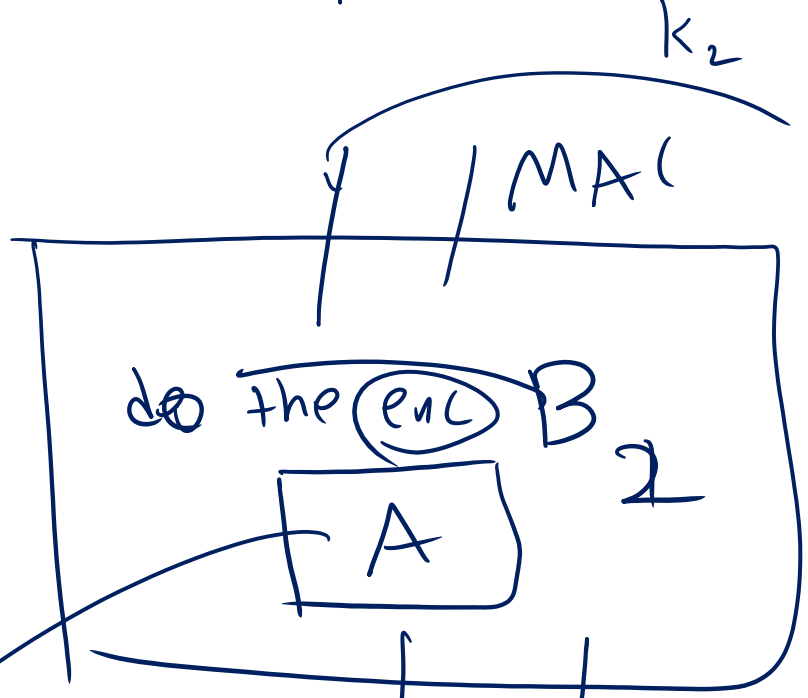
- $\exists$ B_1 poly time against S_1 win $\geq \varepsilon_1$
- $\exists$ B_2 poly-time against S_2 win $\geq \varepsilon_2$

α ↳ Condition: ADV asks some $\tilde{c}$ to $Dec(\tau)$ that was not given to him.

Proof of security (breaking MAC)

if α happens.





$(\tilde{m}, \tilde{t})$ pass the verif.

Simulate environment for (A)

Proof of security (breaking CPA security)