CS171: Cryptography

Lecture 9

Sanjam Garg

Integrity vs Secrecy

Could be modified
before it is received
by Bob!

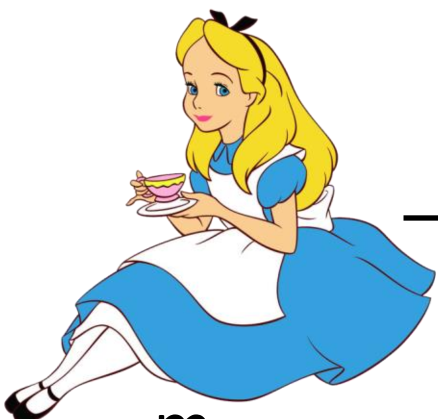
ciphertext

key

c

key

k



k

m

$c \leftarrow \mathrm{Enc}_k(m)$

message/plaintext

$m := \mathrm{Dec}_k(c)$

encryption

decryption

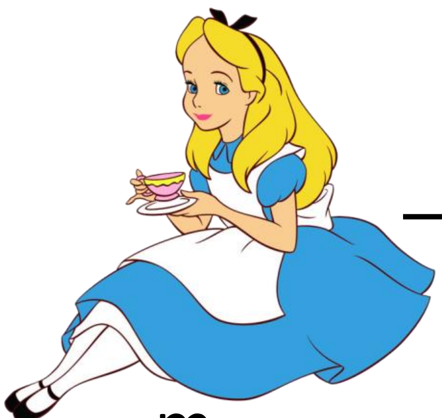
Message Authentication Code (MAC)

message/tag pair

key

key

k



m,t

k

m

$t \leftarrow \mathrm{Mac}_k(m)$

message/plaintext

$0/1 := \mathrm{Vrfy}_k(m,t)$

verify

tag

MACs - Formally

- $(Gen, Mac, Vrfy)$
- $Gen(1^n)$: Outputs a key k.
- $Mac_k(m)$: Outputs a tag t.
- $Vrfy_k(m, t)$: Outputs 0/1.
- Correctness: $\forall\ n, k \leftarrow\ Gen(1^n), \forall m \in \{0,1\}^*$, we have that $Vrfy_k(m, Mac_k(m)) = 1$.
- Default Construction of $Vrfy$ (for deterministic Mac): $Vrfy_k(m, t)$ outputs 1 if and only $Mac_k(m) = t$.

Unforgeability/Security of MAC

$\mathrm{MacForge}_{\mathrm{A},\Pi}(1^n)$

1. Sample $k \leftarrow$ Gen(1^n).
2. Let (m^*, t^*) be the output of $A^{Mac_k(\cdot)}$. Let M be the list of queries A makes.
3. Output 1 if $Vrfy_k(m^*, t^*) = 1 \wedge m^* \notin M$ and 0 otherwise.

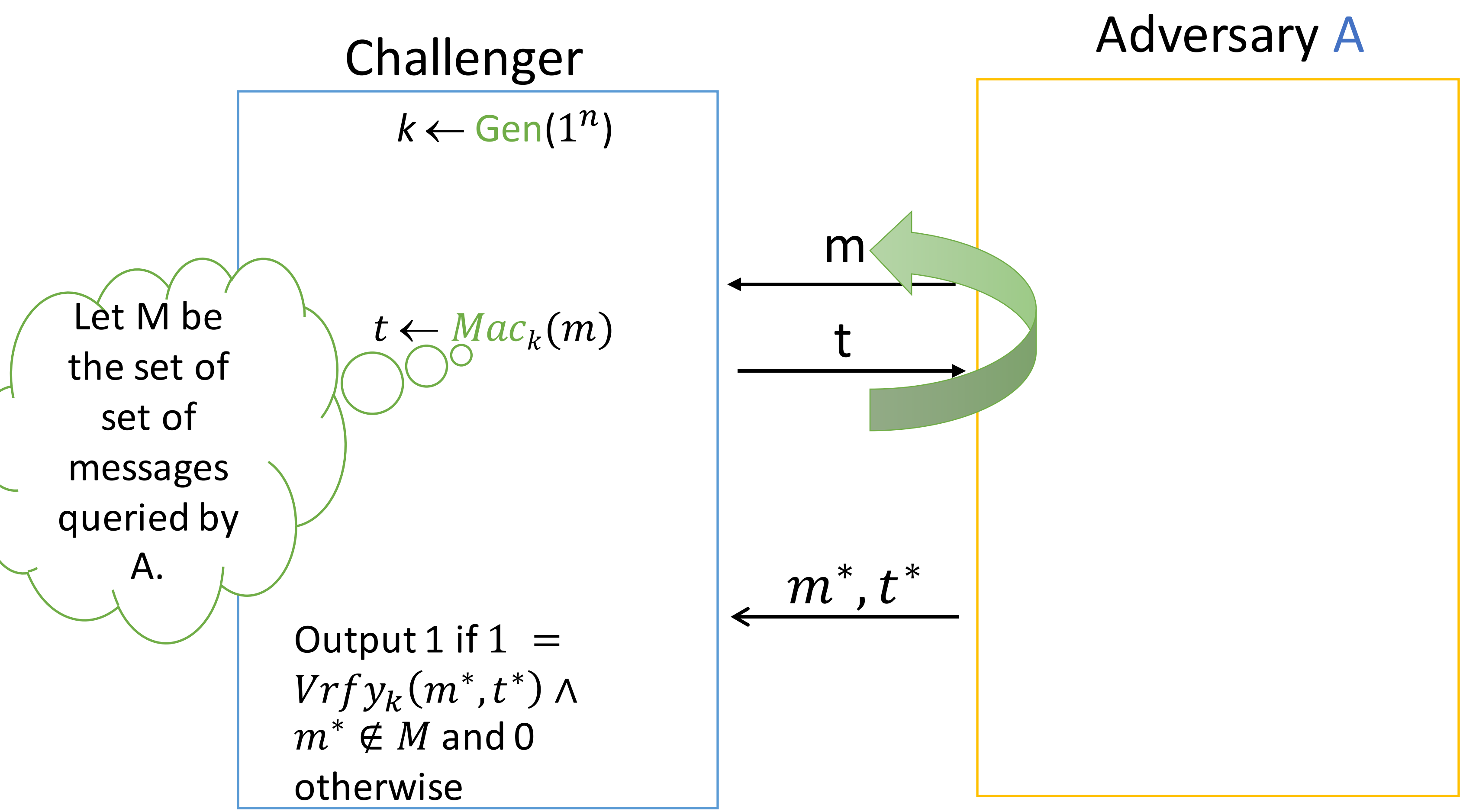
$\Pi = (Gen, Mac, Vrfy)$ is existentially unforgeable under adaptive chosen attack, or is *eu-cma-secure* if

$\forall$ PPT A it holds that:

$\Pr[\mathrm{MacForge}_{\mathrm{A},\Pi} = 1] \leq$ negl(n)

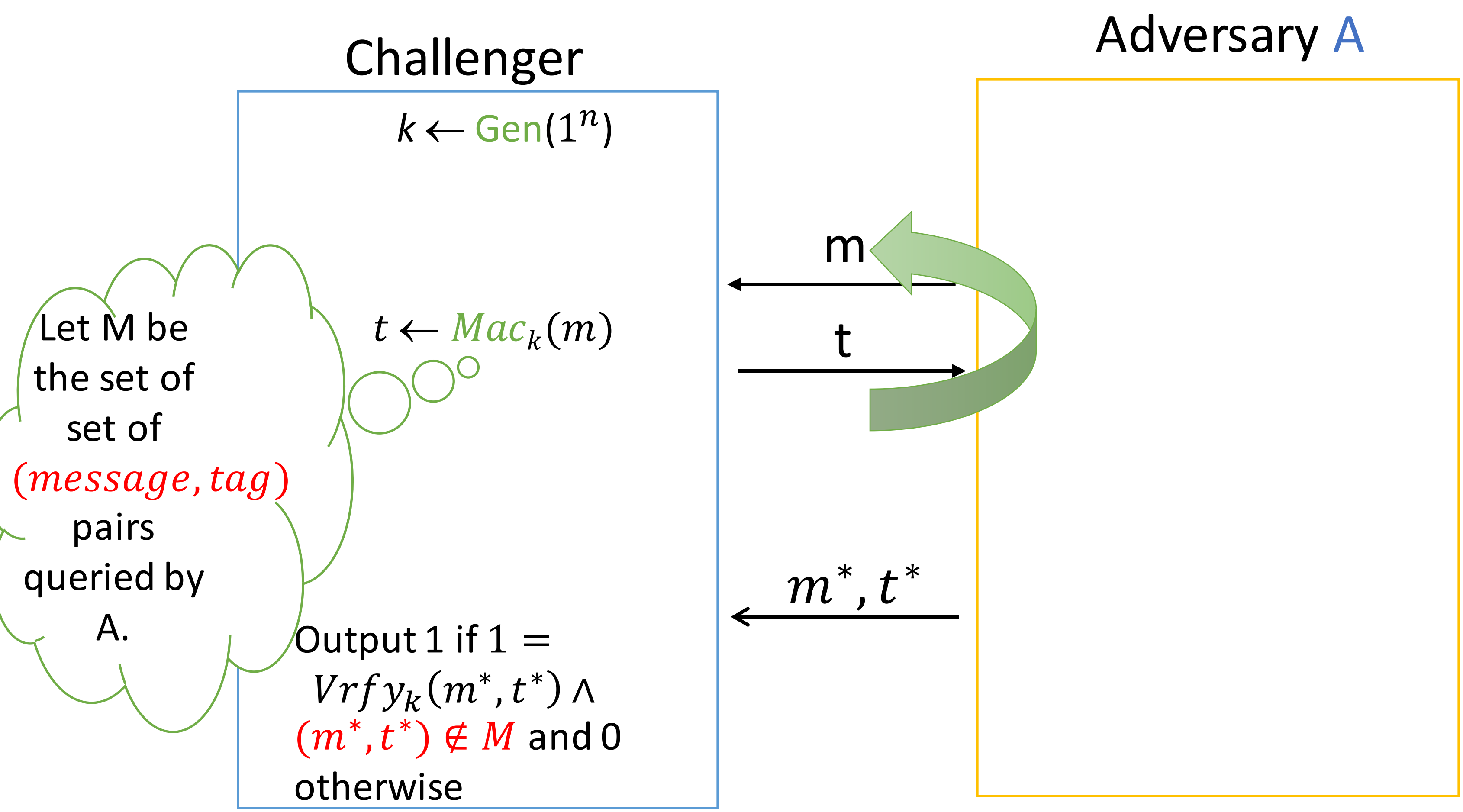
Unforgeability (Pictorially)

$\mathrm{MacForge}_{A,\Pi}(1^n)$



Strong Unforgeability

$\mathrm{MacForge}^{Str}_{A,\Pi}(1^n)$



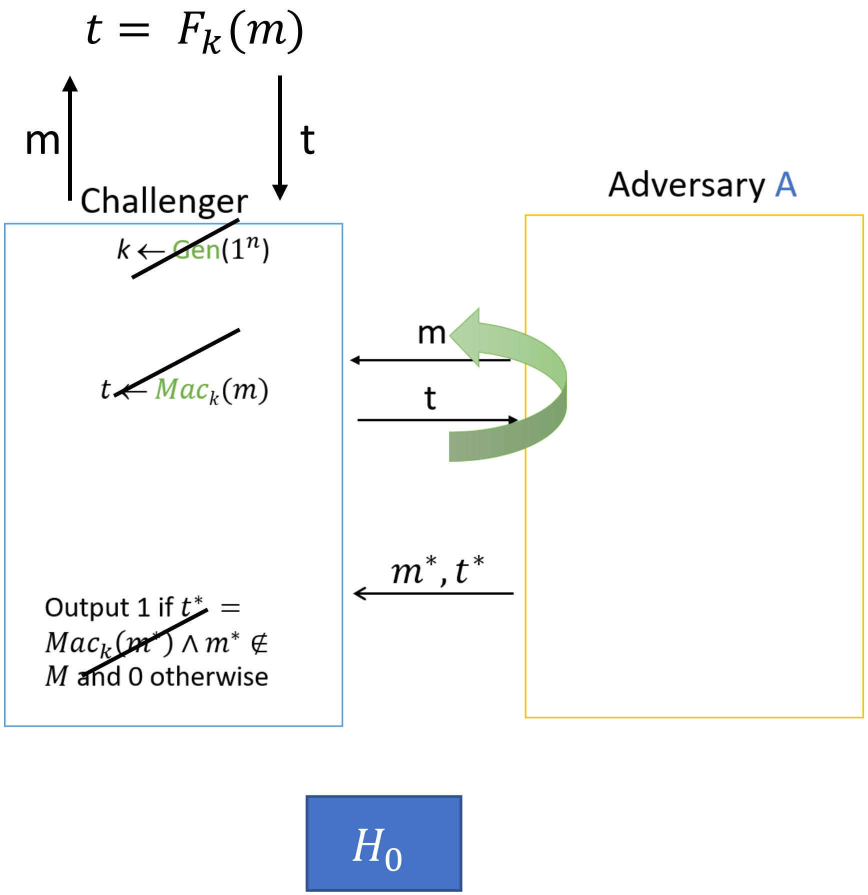
MAC Construction (for fixed-length message)

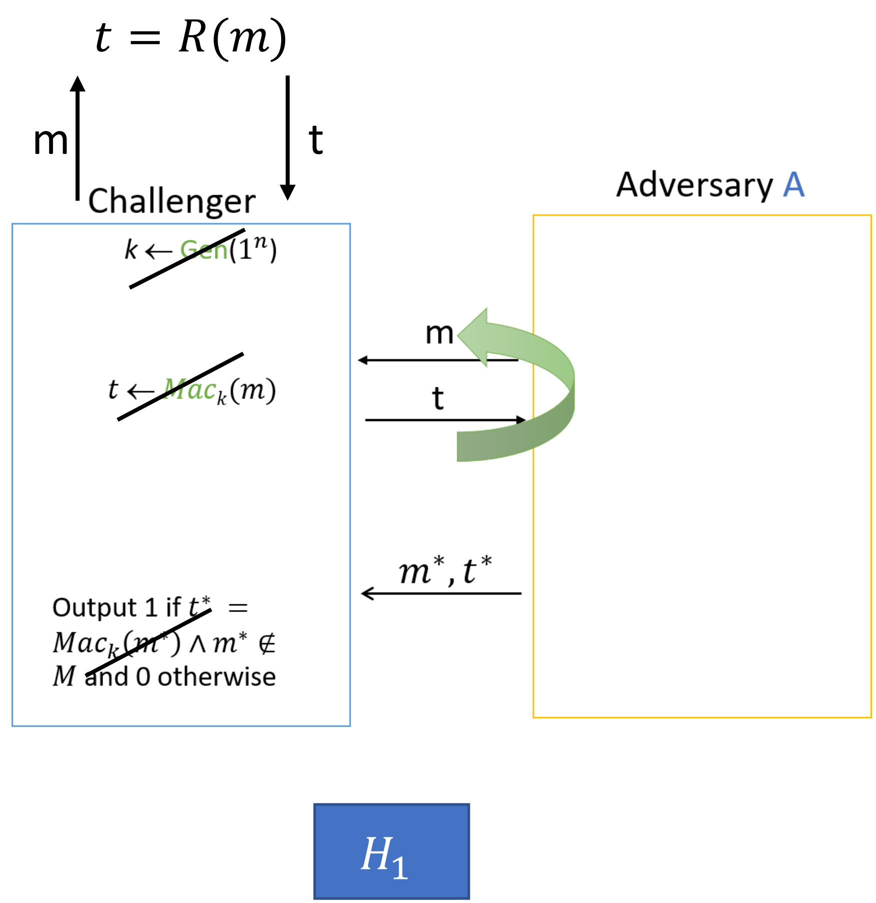
- Let F: $\{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$ be a PRF

- $Gen(1^n)$: Sample $k \leftarrow \{0,1\}^n$.
- $Mac_k(m)$: Output tag $t = F_k(m)$.
- $Vrfy_k(m, t)$: Use default construction.

Proof

All A can do is guess t^*. $\Pr[t^* = Mac_k(m)] = 1/2^n$





MAC (from fixed-length to arbitrary-length messages)

Attempt 1

Construct Mac' (arbitrary-length) from Mac (fixed-length)

- $Mac'_k(m \in \{0,1\}^*)$:
  1. Parse m as $m_1 \cdots m_d$ where each m_i is of length n
  2. Output $t_1 \dots t_d$, where for each i we have
     - $t_i = Mac_k(m_i)$

Change the order of blocks to get a forgery!

MAC (from fixed-length to arbitrary-length messages)

Attempt 2

Construct Mac' (arbitrary-length) from Mac (fixed-length)

- $Mac'_k(m \in \{0,1\}^*)$:
  1. Parse m as $m_1 \cdots m_d$ where each m_i is of length $n/2$
  2. Output $t_1 \ldots t_d$, where for each i we have
     - $t_i = Mac_k(i||m_i)$

Mix and match blocks from different MACs to get a forgery!

MAC (from fixed-length to arbitrary-length messages)

Attempt 3

Construct Mac' (arbitrary-length) from Mac (fixed-length)

- $Mac'_k(m \in \{0,1\}^*)$:
  1. Parse m as $m_1 \cdots m_d$ where each m_i is of length $n/3$
  2. Sample $r \leftarrow \{0,1\}^{n/3}$
  3. Output $t_1 \ldots t_d$, where for each i we have
     - $t_i = Mac_k(r||i||m_i)$

Drop a few blocks to get a forgery!

MAC (from fixed-length to arbitrary-length messages)

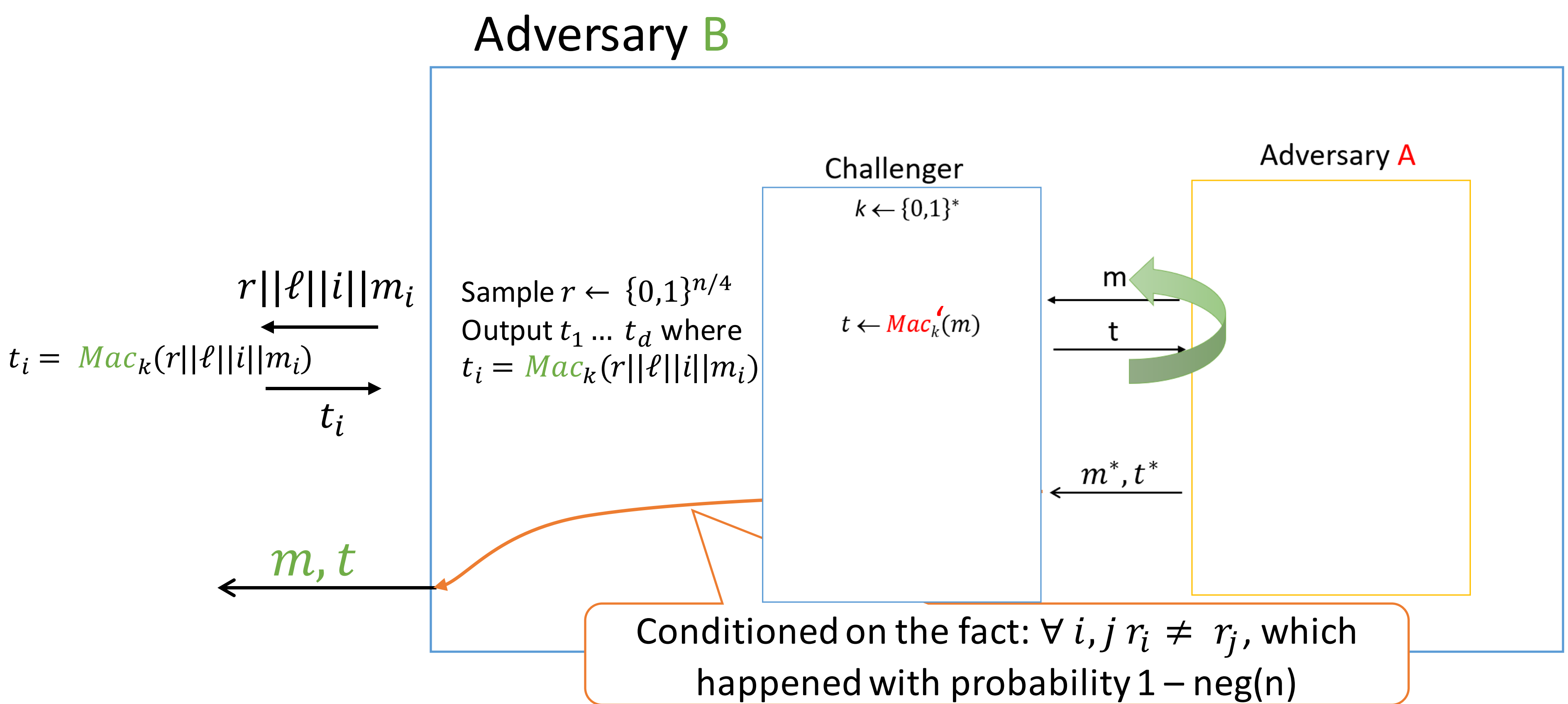
Construct Mac' (arbitrary-length) from Mac (fixed-length)

- $Mac'_k(m \in \{0,1\}^*)$:
  - Parse m as $m_1 \cdots m_d$ where each m_i is of length $n/4$
  - $r \leftarrow \{0,1\}^{n/4}$
  - Output r, $t_1 \ldots t_d$, where for each i we have
    - $t_i = Mac_k(r||\ell||i||m_i)$, where ℓ is the number of blocks

Mac is not deterministic! How do we define $Vrfy$?

Proof of Security

- Consider an adversary A that breaks Mac' then we construct an adversary B that breaks Mac



Proof of Security: Case Analysis

- A outputs $t^* = (r^*, t_1^*, \dots t_{\ell^*}^*)$:
  - Case I: $\forall\ i,\ r^* \neq r_i$ then we have a forgery
  - Case II: $\exists\ i,\ r^* = r_i$ but $\ell^* \neq \ell_i$, again a forgery as ℓ^* appears in each block.
  - Case III: $\exists\ i,\ r^* = r_i$ but $\ell^* = \ell_i, m^* \neq m_i$, a forgery on at least one block.
- Thus, B can use the forgery above as its output.

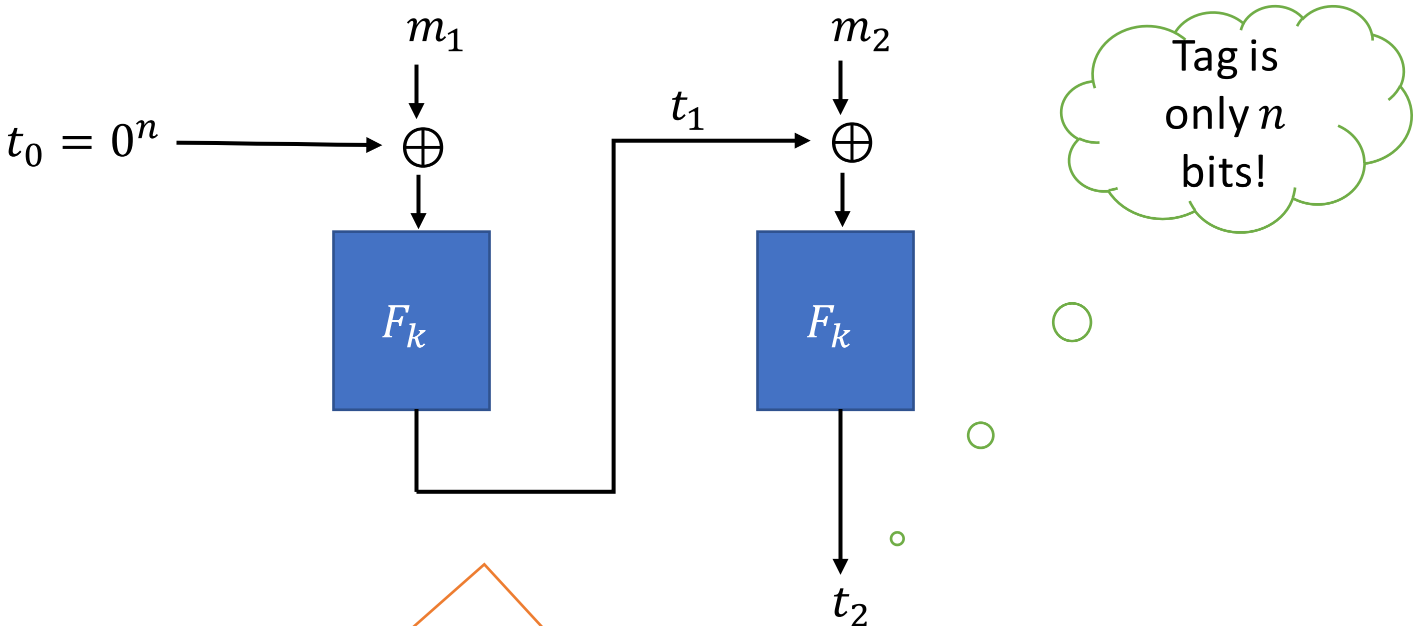
For a message of length $\ell \cdot n$ bits, what is the length of the Mac?

Proof of Security: Case Analysis

- A outputs $t^* = (r^*, t_1^*, \ldots t_{\ell^*}^*)$:
  - Case I: $\forall\, i,\ r^* \neq r_i$ then we have a forgery
  - Case II: $\exists\, i,\ r^* = r_i$ but $\ell^* \neq \ell_i$, again a forgery as ℓ^* appears in each block.
  - Case III: $\exists\, i,\ r^* = r_i$ but $\ell^* = \ell_i, m^* \neq m_i$, a forgery on at least one block.
- Thus, B can use the forgery above as its output.

For a message of length $\ell \cdot n$ bits we get a Mac of length $4 \cdot \ell \cdot n$! Very inefficient!

CBC-MAC (Using Block Cipher)



Insecure: Adversary can forge tags (of larger lengths)!

Attack on CBC-MAC

- Adversary obtains tag t_1 on a random message m_1
- Next, adversary obtains tag t_2 on message $t_1 \oplus m_2$.
- Note that t_2 serves as a tag on message $m_1 || m_2$

Thm: Let $\ell(\cdot)$ be a polynomial. If F is a PRF, then the CBC-MAC is ef-cma for messages of length $\ell(n) \cdot n$.

Proof of Security for fixed length

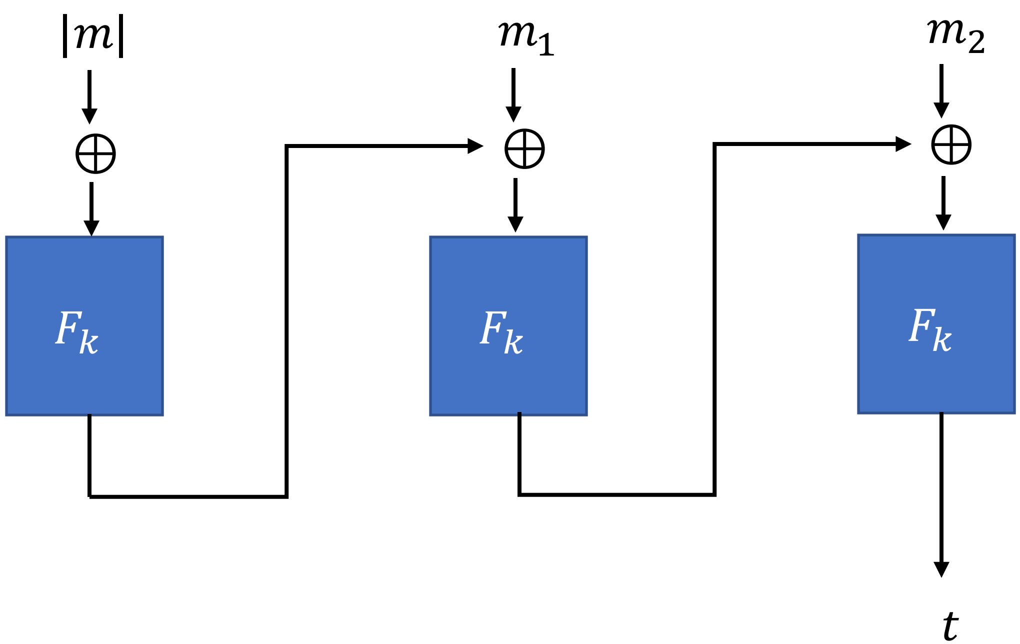
Suffices to prove that CBC is a PRF!

$CBC_k(x_1, \dots x_\ell) = F_k(F_k(\dots \mathrm{F_k}(\mathrm{F_k}(\mathrm{x_1}) \oplus x_2) \oplus \cdots) \oplus x_\ell)$, where $|x_1| = |x_2| \dots = |x_\ell|$.

- In fact more: $CBC_k(\cdot)$ is a PRF as long as inputs are from the set $P \subset (\{0,1\}^n)^*$ that is *prefix-free*
  - P doesn't contain the empty string
  - There doesn't exist $x, x' \in P$ such that x is prefix of x'
- Intuitive, we will not prove it!

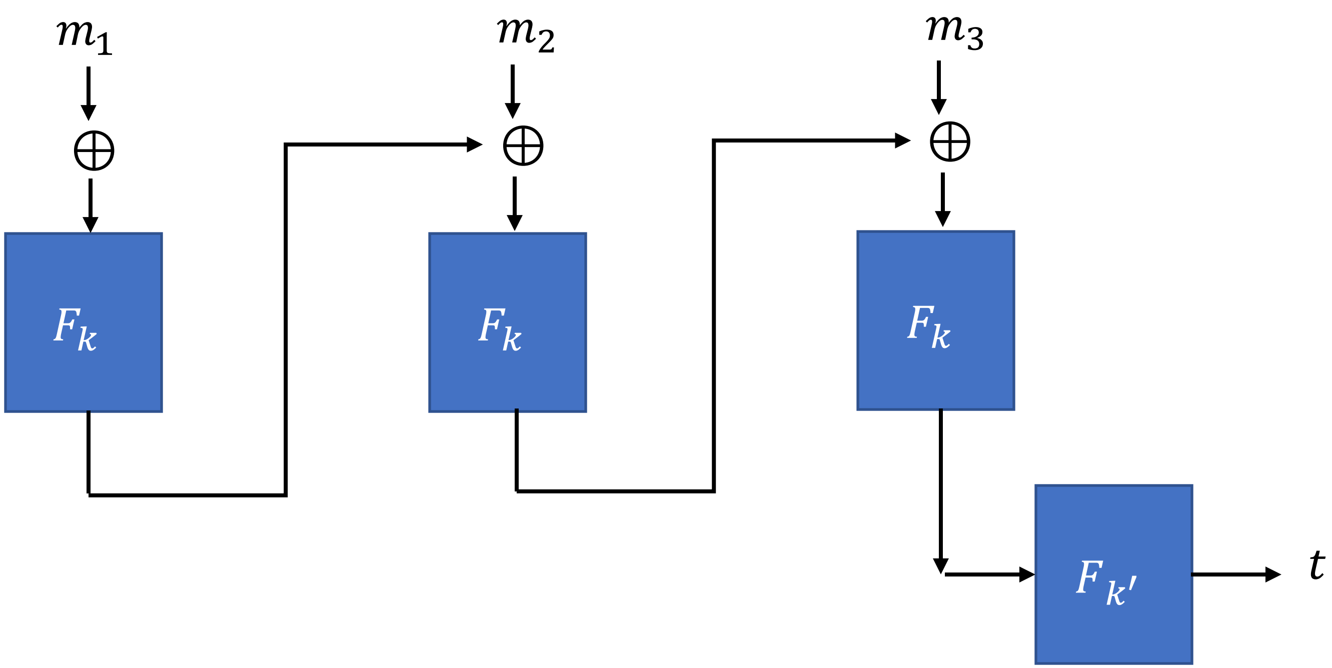
Use this to Mac messages of arbitrary length (multiples of n)

- Method 1: Mac on message m is the CBC-Mac on message $|m|\ ||\ m$



Use this to sign messages of arbitrary length (multiples of n)

- Method 2: Mac of the CBC-Mac



Thank You!