Introduction to Modern Cryptography

Michele Ciampi

(Slides courtesy of Prof. Jonathan Katz)

Lecture 10, Part 2

Message Authentication Code (MAC)

So Far

Last lecture

- Introduced **message integrity**
- Introduced **message authentication codes (MAC)**

This lecture

MAC algorithms and proof of security

A Fixed-length MAC: Intuition

We need a keyed function Mac such that:

- Given $\mathsf{Mac}_k(m_1), \mathsf{Mac}_k(m_2), \ldots$
- ...it is infeasible to predict the value $\mathsf{Mac}_k(m)$ for any $m \notin \{m_1, \ldots\}$

PRF

Let f be PRF. Knowledge of $f(x_1), f(x_2), \ldots$ does not reveal any information on $f(x): \ x \notin \{x_1, x_2, \ldots\}$.

Idea

Let Mac be a PRF i.e. set $\mathsf{Mac}_k \equiv F_k$

A Fixed-length MAC Construction

Fixed-length MAC

Let F be a length-preserving PRF (i.e. block cipher). Construct the following MAC $\mathbf{\Pi}$:

- Gen: choose a uniform key k for F
- $\mathsf{Mac}_k(m)$: output $F_k(m)$
- $\mathsf{Vrfy}_k(m, t)$: output $\mathbf{1}$ iff $F_k(m) = t$

A Fixed-length MAC Construction

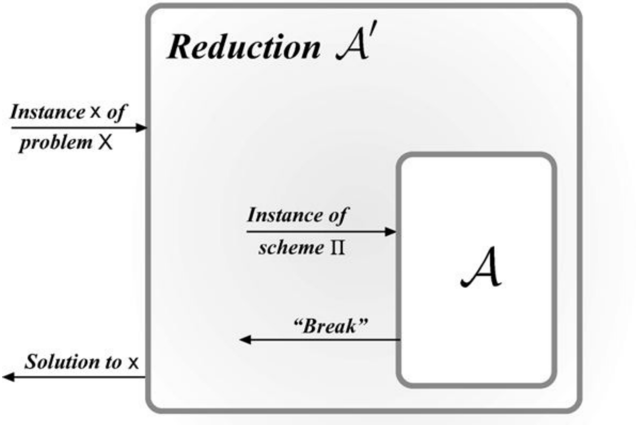
Theorem

$\boldsymbol{F}$ *is a PRF* $\implies$ $\mathbf{\Pi}$ *is a secure MAC*

Proof

By reduction

Proof by Reduction



IMC Textbook 2nd ed. CRC Press 2015

Proof by Reduction (see CPA-security)

High level

- Attacker $\boldsymbol{A}$ attacks MAC $\boldsymbol{\Pi}$ (as was defined)
- Design distinguisher $\boldsymbol{D}$ that uses $\boldsymbol{A}$ as a subroutine to attack the PRF $\boldsymbol{F}$
  - i.e. $\boldsymbol{D}$ tries to distinguish $\boldsymbol{F}$ from a random function (RF)
- $\boldsymbol{D}$ simulates to $\boldsymbol{A}$ the steps in the $\mathsf{Forge}_{\boldsymbol{A},\boldsymbol{\Pi}}(\boldsymbol{n})$ experiment for $\boldsymbol{F}$ and for a RF
- Relate the success $\mathbf{Pr}$ of $\boldsymbol{A}$ to the success $\mathbf{Pr}$ of $\boldsymbol{D}$
- If $\boldsymbol{A}$ succeeds $\implies$ $\boldsymbol{D}$ succeeds $\implies$ $\boldsymbol{F} \neq$ PRF
- contradicts $\boldsymbol{F}$ PRF $\implies$ $\boldsymbol{A}$ can not succeed $\implies$ $\boldsymbol{\Pi}$ is a secure MAC

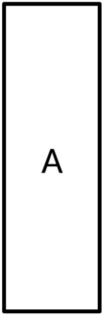
The $\mathsf{Forge}_{A,\Pi}(n)$ Experiment (Recall)

Fix A, Π. Define randomized experiment $\mathsf{Forge}_{A,\Pi}(n)$:

- $k \leftarrow \mathsf{Gen}(1^n)$
- A interacts with an oracle $\mathsf{Mac}_k(\cdot)$:
  - A submits $m_1, \ldots, m_i$ to $\mathsf{Mac}_k(\cdot)$
  - A collects back $t_1, \ldots, t_i$ from $\mathsf{Mac}_k(\cdot)$
  - Let $M = \{m_1, \ldots, m_i\}$ be the set of messages submitted to the oracle
- A outputs (m, t)
- A succeeds, and the experiment evaluates to 1, if $\mathsf{Vrfy}_k(m, t) = 1$ and $m \notin M$

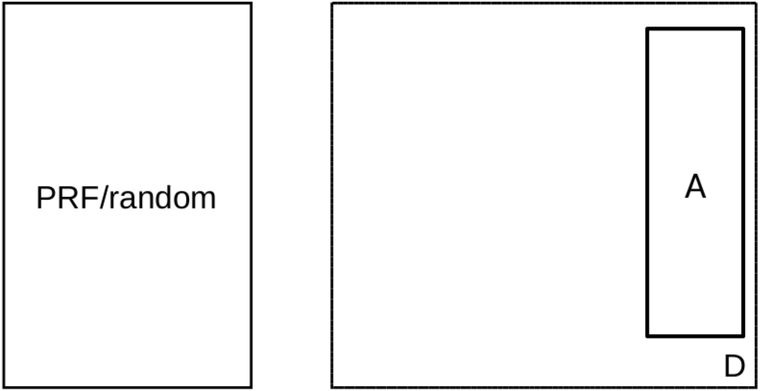
Π is secure if $\forall$ PPT A, $\exists \epsilon$ negl. such that $\mathbf{Pr}[\mathsf{Forge}_{A,\Pi}(n) = 1] \leq \epsilon(n)$

Proof by Reduction (in Picture)



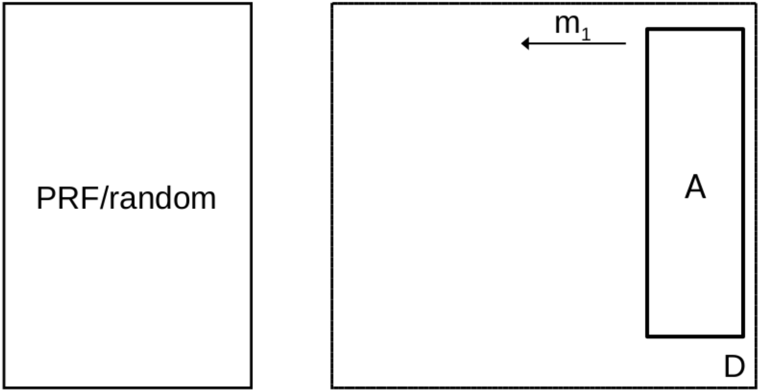
$\boldsymbol{A}$ attacks the MAC $\boldsymbol{\Pi}$

Proof by Reduction (in Picture)



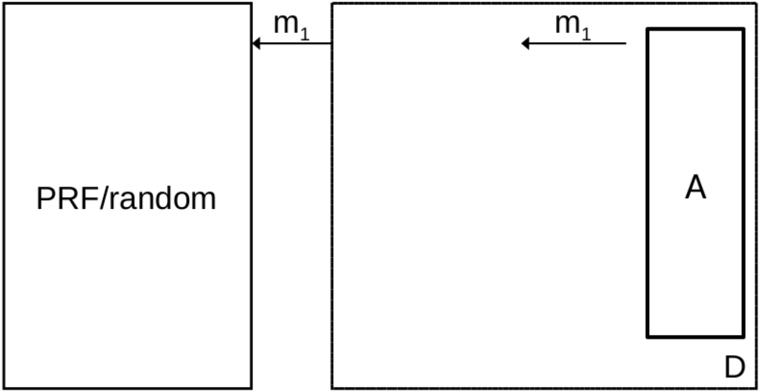
D uses A as a subroutine in distingishing between RF f and PRF F_k for uniform k

Proof by Reduction (in Picture)



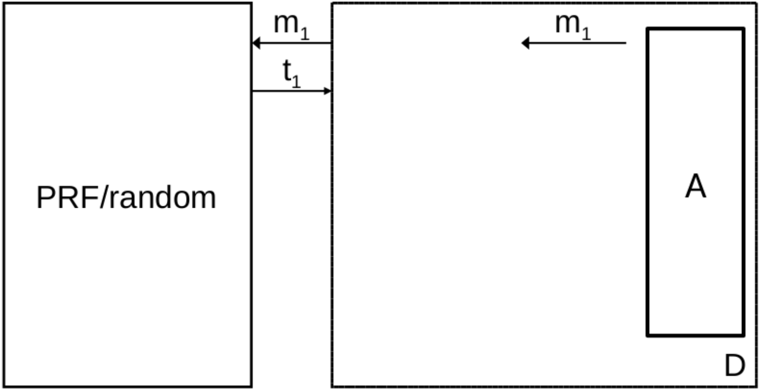
A requests the tag on message $\boldsymbol{m_1}$

Proof by Reduction (in Picture)



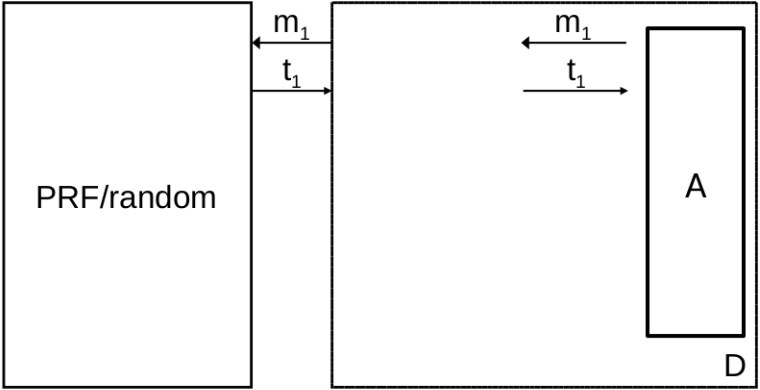
D forwards m_1 to the oracle $\mathcal{O} \in \{f, F_k\}$

Proof by Reduction (in Picture)



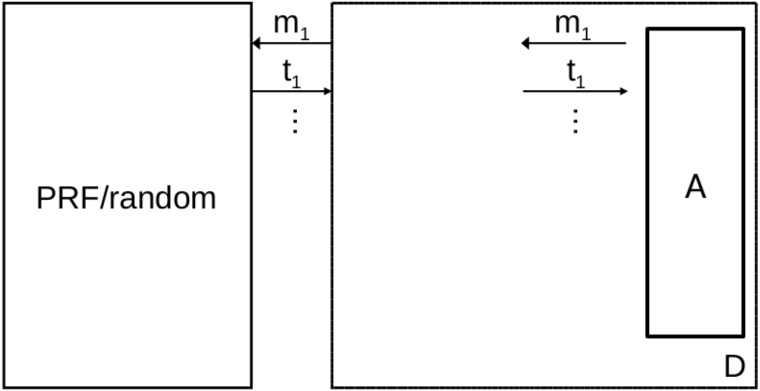
$\boldsymbol{D}$ gets back $\boldsymbol{t_1 = \mathcal{O}(m_1)}$

Proof by Reduction (in Picture)

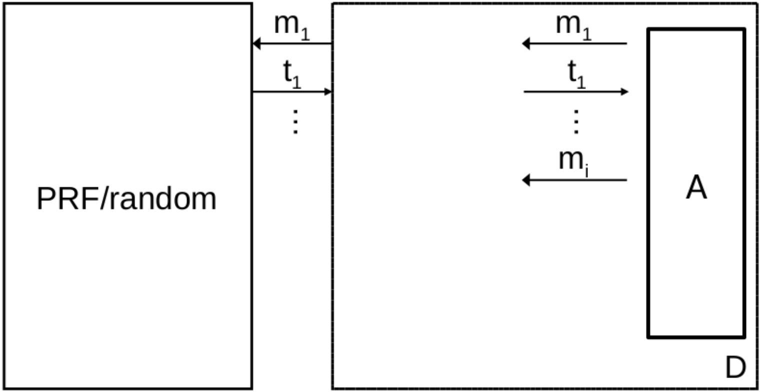


$\boldsymbol{D}$ forwards $\boldsymbol{t_1 = \mathcal{O}(m_1)}$ to $\boldsymbol{A}$. From the perspective of $\boldsymbol{A}$, $\boldsymbol{t_1}$ is the tag of $\boldsymbol{m_1}$

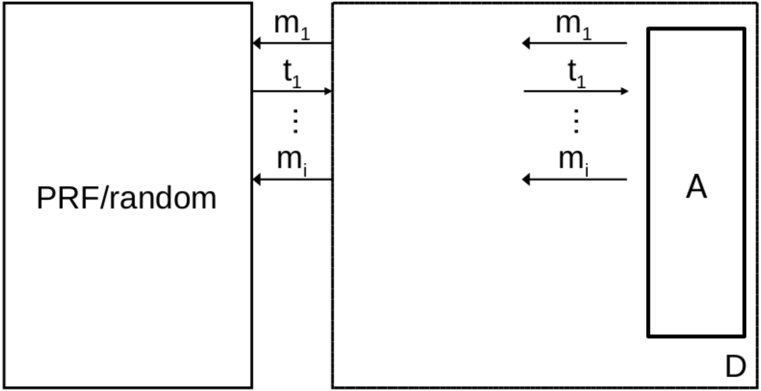
Proof by Reduction (in Picture)



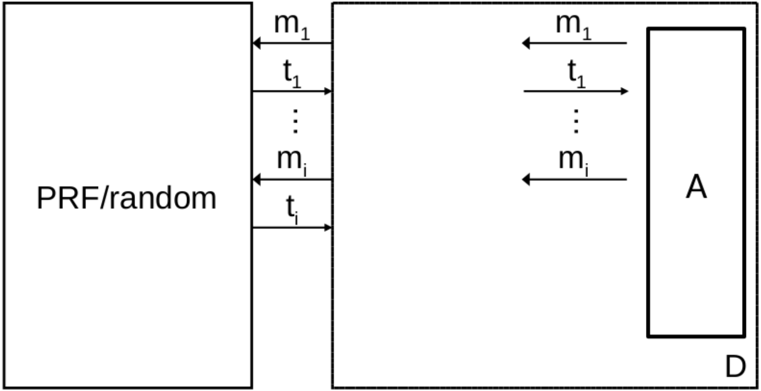
Proof by Reduction (in Picture)



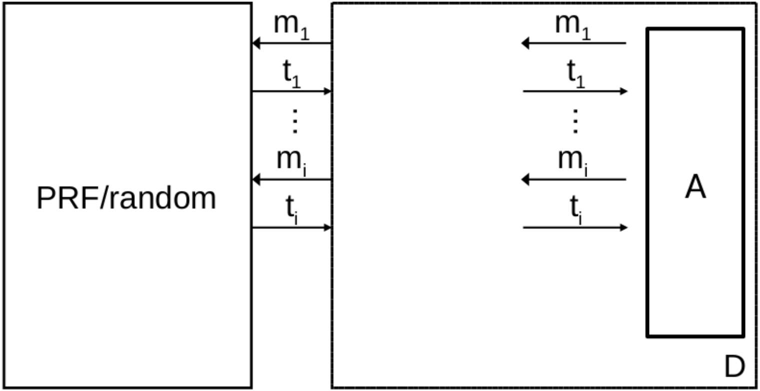
Proof by Reduction (in Picture)



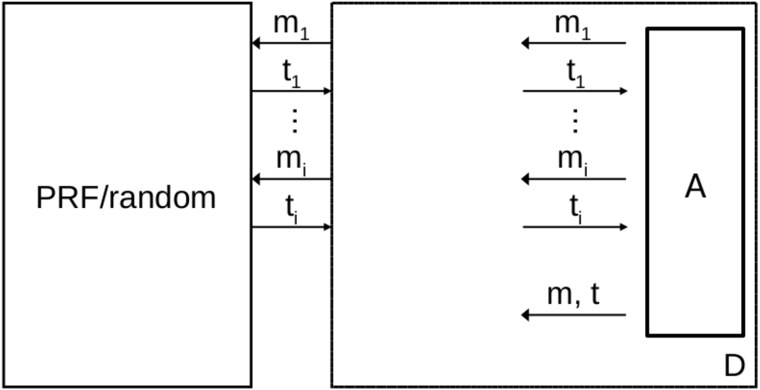
Proof by Reduction (in Picture)



Proof by Reduction (in Picture)

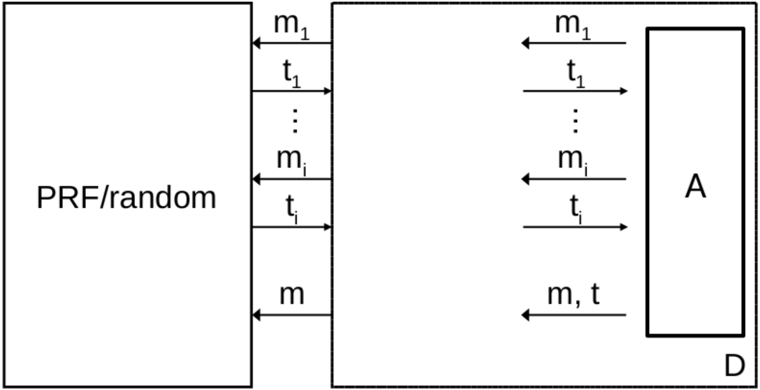


Proof by Reduction (in Picture)



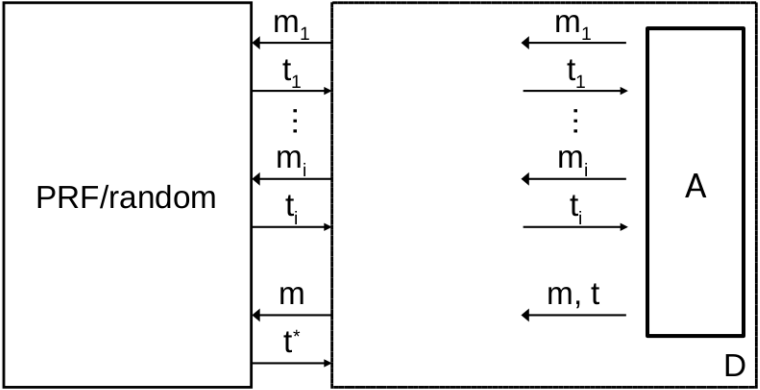
$\boldsymbol{A}$ outputs its forgery $\boldsymbol{(m, t)}$: $\boldsymbol{m \notin \{m_1, m_2 \ldots\}}$, $\boldsymbol{t}$ – tag for $\boldsymbol{m}$

Proof by Reduction (in Picture)



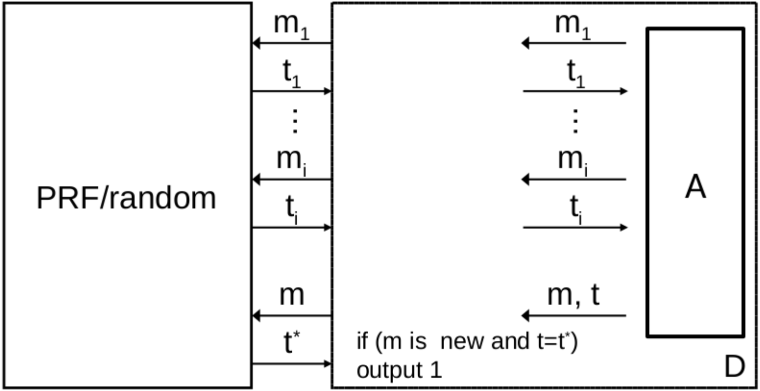
$\boldsymbol{D}$ forwards $\boldsymbol{m}$ to the oracle $\boldsymbol{\mathcal{O}} \in \{\boldsymbol{f}, \boldsymbol{F_k}\}$

Proof by Reduction (in Picture)



D gets back $t^* = \mathcal{O}(m)$

Proof by Reduction (in Picture)



If $t^* = t \implies D$ outputs **1**; otherwise **0**;

Proof by Reduction

The Simulation

$\boldsymbol{D}$ simulates $\mathsf{Forge}_{\boldsymbol{A},\boldsymbol{\Pi}}(\boldsymbol{n})$ for $\boldsymbol{A}$ with $\boldsymbol{f}$–RF or $\boldsymbol{f}$–PRF:

1. $\boldsymbol{A}$ submits $\boldsymbol{m_i} : \ \boldsymbol{i = 1, 2 \ldots}$ to the MAC $\boldsymbol{\mathcal{O}}$
2. $\boldsymbol{D}$ simulates the interaction with the MAC $\boldsymbol{\mathcal{O}}$ for $\boldsymbol{A}$:
   - $\boldsymbol{D}$ forwards $\boldsymbol{m_i}$ to $\boldsymbol{f}$; receives $\boldsymbol{t_i = f(m_i)}$
   - $\boldsymbol{D}$ returns $\boldsymbol{t_i}$ to $\boldsymbol{A}$
3. $\boldsymbol{A}$ outputs $\boldsymbol{(m, t)}$; $\boldsymbol{m \notin \{m_1, m_2, \ldots\}}$
4. $\boldsymbol{D}$ forwards $\boldsymbol{m}$ to $\boldsymbol{f}$; receives $\boldsymbol{t^* = f(m)}$
5. If $\boldsymbol{t^* = t \implies D}$ outputs $\mathbf{1}$ (success); otherwise $\mathbf{0}$ (fail)

World $\mathbf{0}$: $\boldsymbol{D}$ with a Truly Random Function $\boldsymbol{f}$

$\boldsymbol{D}^{\boldsymbol{f}}$ simulates $\mathsf{Forge}_{\boldsymbol{A},\boldsymbol{\Pi}}(\boldsymbol{n})$ for $\boldsymbol{A}$ with truly random $\boldsymbol{f}$

- By definition of RF observing $\boldsymbol{f(m_1), f(m_2), \ldots}$ does not reveal information on $\boldsymbol{f(m) : m \notin \{m_1, m_2, \ldots\}}$
- Therefore

$$\mathbf{Pr}[D^{f(\cdot)} = 1] = \mathbf{Pr}[f(m) = t] = \mathbf{Pr}[t^* = t] = 2^{-n}$$

where $\boldsymbol{n = |m|}$

World $\mathbf{1}$: $\boldsymbol{D}$ with a Pseudoandom Function $\boldsymbol{f = F_k}$

$\boldsymbol{D^{F_k}}$ simulates $\mathsf{Forge}_{\boldsymbol{A,\Pi}}(\boldsymbol{n})$ for $\boldsymbol{A}$ with truly random $\boldsymbol{F_k}$

- The view of $\boldsymbol{A}$ in this case is **exactly** as in the $\mathsf{Forge}_{\boldsymbol{A,\Pi}}(\boldsymbol{n})$ experiment
- Therefore

$$\mathbf{Pr}[\boldsymbol{D}^{\boldsymbol{F_k}(\cdot)} = \mathbf{1}] = \mathbf{Pr}[\mathsf{Forge}_{\boldsymbol{A,\Pi}}(\boldsymbol{n}) = \mathbf{1}]$$

The Reduction

Proof.

By the assumption that F is a PRF $\exists \epsilon(n) = \mathbf{negl}$:

$$|\Pr_{k \leftarrow \{0,1\}^n}[D^{F_k(\cdot)} = 1] - \Pr_{f \leftarrow \mathcal{F}_n}[D^{f(\cdot)} = 1]| \leq \epsilon(n)$$

By the simulation of $\mathsf{Forge}_{A,\Pi}(n)$ by D^f with RF:

$$\Pr_{f \leftarrow \mathcal{F}_n}[D^{f(\cdot)} = 1] = \Pr[f(m) = t] = 2^{-n}$$

By the simulation of $\mathsf{Forge}_{A,\Pi}(n)$ by D^{F_k} with PRF:

$$\Pr_{k \leftarrow \{0,1\}^n}[D^{F_k(\cdot)} = 1] = \Pr[\mathsf{Forge}_{A,\Pi}(n) = 1]$$

Therefore

$$\Pr[\mathsf{Forge}_{A,\Pi}(n) = 1] \leq \epsilon(n) + 2^{-n} = \mathbf{negl}(n)$$

$\implies$ Π is a secure MAC □

Limitations of the MAC $\mathbf{\Pi}$

- Block ciphers (i.e. PRFs) have short, fixed-length block size
- e.g. AES has a **128**-bit block size (shorter than a tweet!)
- Therefore $\mathbf{\Pi}$ is **limited to authenticating only short, fixed-length messages**
- In practise we want to be able to send messages much longer than **128** bits
- We also want to be able to send messages of different (i.e. not fixed) length
- **A solution:** CBC-MAC (next)

Variable-length MAC

Suggestion

Can you construct a secure MAC for variable-length messages from a MAC for fixed-length messages?

Idea

$$\mathsf{Mac}'_k(m_1 \dots m_l) = \mathsf{Mac}_k(m_1) \dots \mathsf{Mac}_k(m_l)$$
$$\mathsf{Vrfy}'_k(m_1 \dots m_l, t_1 \dots t_l) = 1 \iff \forall i : \mathsf{Vrfy}_k(m_i, t_i) = 1$$

Is this secure?

A Construction

Probem

Need to prevent (at least):

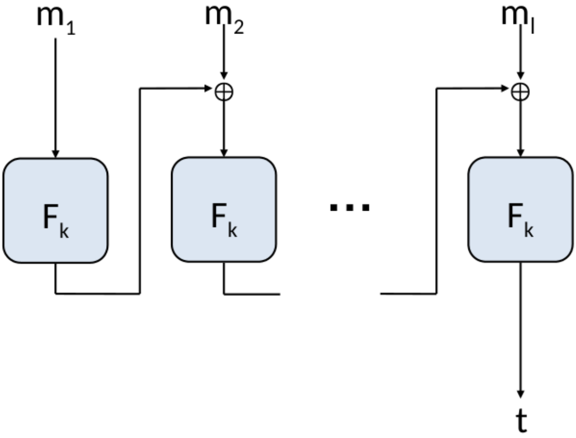
- Block reordering
- Truncation
- Mixing-and-matching blocks from multiple messages

One solution

$$\mathsf{Mac}'_k(m_1 \ldots m_l) = r, \mathsf{Mac}_k(r|l|1|m_1), \mathsf{Mac}_k(r|l|2|m_2), \ldots$$

Not very efficient – can we do better? Yes: CBC-MAC.

Basic CBC-MAC



CBC-MAC vs. CBC-mode

- CBC-MAC is deterministic (no IV)
  - MACs do not need to be randomized to be secure
  - Verification is done by re-computing the result
- In CBC-MAC, only the final value is output
- Both are essential for security

Security of Basic CBC-MAC

Theorem

If F is a length-preserving PRF with input length n, then for any **fixed** l *basic CBC-MAC is a secure MAC for messages of length* ln

Proof

By reduction (omitted)

Note

- The sender and receiver must agree on the length parameter l in advance
- Basic CBC-MAC is not secure if this is not done!

CBC-MAC for Variable Length Messages

Method 1

Prepend the message with its block length $\boldsymbol{l}$

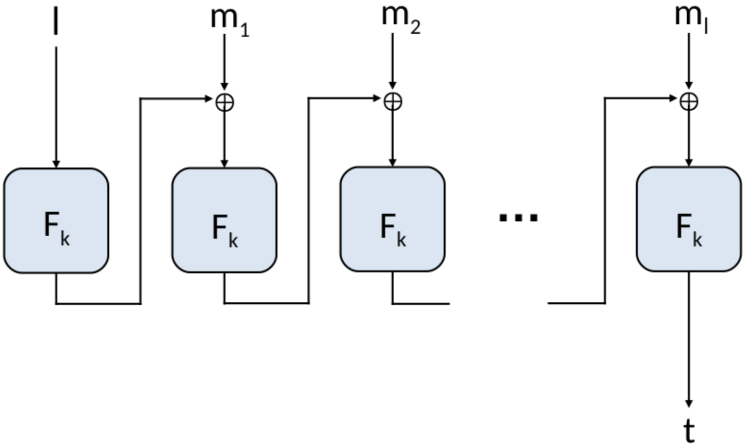
Method 2

- Apply $\boldsymbol{F_k}$ to the block length $\boldsymbol{l}$ to obtain key $\boldsymbol{k_l}$
- Compute the tag with Basic CBC-MAC and key $\boldsymbol{k_l}$
- Send $\boldsymbol{(t, l)}$

Method 3

- Choose two keys $\boldsymbol{k_1 \leftarrow \{0,1\}^n}$, $\boldsymbol{k_2 \leftarrow \{0,1\}^n}$
- Compute $\boldsymbol{t_1}$ with Basic CBC-MAC using key $\boldsymbol{k_1}$
- Compute final tag using $\boldsymbol{k_2}$ as $\boldsymbol{t = F_{k_2}(t_1)}$

CBC-MAC for Variable Length Messages: Method **1**



Prepend the message with its block length $\boldsymbol{l}$

Hash Functions

Hash functions

Another way for constructing MACs for variable length messages

$\Longrightarrow$ next lecture

End

References: Sec. 4.3 (not Theorem 4.8) and Sec 4.4.1