Message Authentication Code

Gaspare FERRARO
CyberSecNatLab
Matteo ROSSI
Politecnico di Torino

CYBER CHALLENGE.IT

CYBERSECURITY NATIONAL LABORATORY

cini

https://cybersecnatlab.it

License & Disclaimer

License Information

This presentation is licensed under the
Creative Commons BY-NC License

BY NC

To view a copy of the license, visit:

http://creativecommons.org/licenses/by-nc/3.0/legalcode

Disclaimer

- We disclaim any warranties or representations as to the accuracy or completeness of this material.
- Materials are provided “as is” without warranty of any kind, either express or implied, including without limitation, warranties of merchantability, fitness for a particular purpose, and non-infringement.
- Under no circumstances shall we be liable for any loss, damage, liability or expense incurred or suffered which is claimed to have resulted from use of this material.

Goal

- Provide the definition and usage of MAC
- Show differences between hash and MAC
- Show different implementations of MACs:
  - Using symmetric-key ciphers
  - Using hash functions

Prerequisites

- Lectures:
  - *CR_1.3 - Block Ciphers*
  - *CR_1.4 - Stream Ciphers*
  - *CR_2 - Public-key cryptography*
  - *CR_3.1 - Hash Functions*

Outline

- Introduction
- MAC from symmetric ciphers: CBC-MAC and NMAC
- MAC from Hash functions: HMAC

Outline

- Introduction
- MAC from symmetric ciphers: CBC-MAC and NMAC
- MAC from Hash functions: HMAC

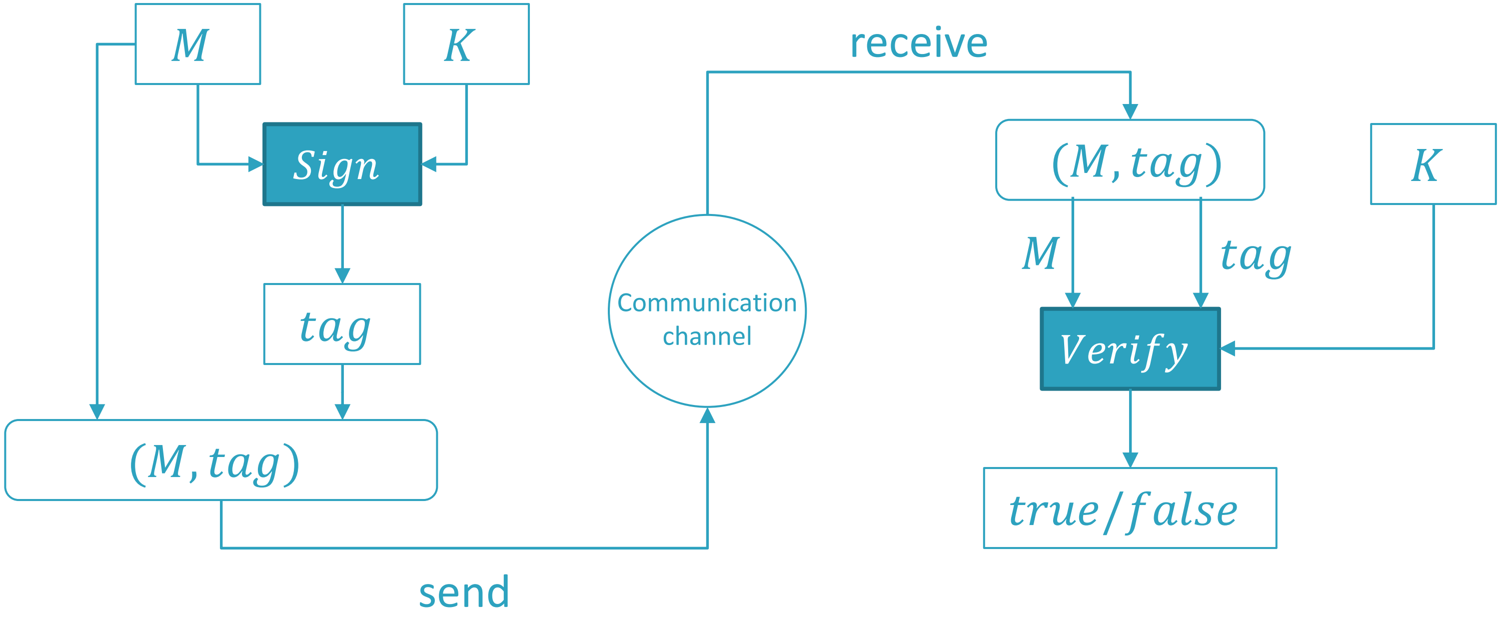
Encrypted message authentication

- In order to guarantee both integrity and authentication:
  - A shared secret key can be generated using for example the Diffie-Hellman algorithm
  - Cryptographic hash functions ensure integrity
  - If the digest is encrypted, authentication is reached
  - If also the message is encrypted, confidentiality is reached

Message Authentication Code (MAC)

- A Message Authentication Code (MAC) is a pair of functions, Sign and Verify, such that:
  - Sign takes a message M of arbitrary length and a key k and produces a fixed-length string, called tag
  - Verify takes the message M, the key k and the tag, and outputs *true* if the tag is valid and *false* otherwise

Message Authentication Code (MAC)



Hash vs MAC

- With hash functions we can reach integrity but not authentication
- With MACs we can reach both integrity and authentication

Hash vs MAC

Primitive	Integrity	Authentication
Hash	Yes	No
MAC	Yes	Yes

Attacks on MAC

- A MAC can be subject to several types of attacks by external attackers who do not know the key:
  - Existential Forgery Attack: The attacker can create a valid message M and a tag for M without knowing the key. The attacker defines both M and the corresponding tag.
  - Selective Forgery Attack: Given a message M, the attacker is able to produce a valid tag for M.
  - Universal Forgery Attack: The attacker can create a valid tag for any possible message M. This attack is the most powerful and implies the *total break* of the MAC scheme.

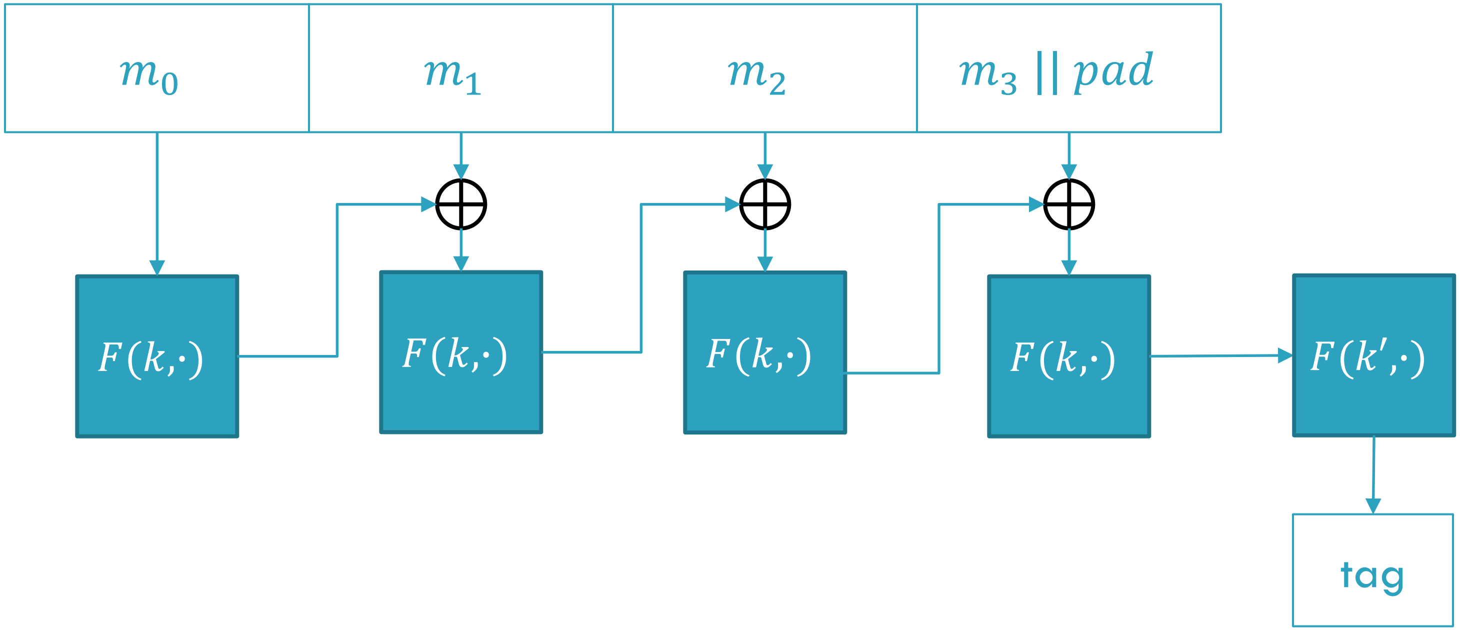
Outline

- Introduction
- MAC from symmetric ciphers: CBC-MAC and NMAC
- MAC from Hash functions: HMAC

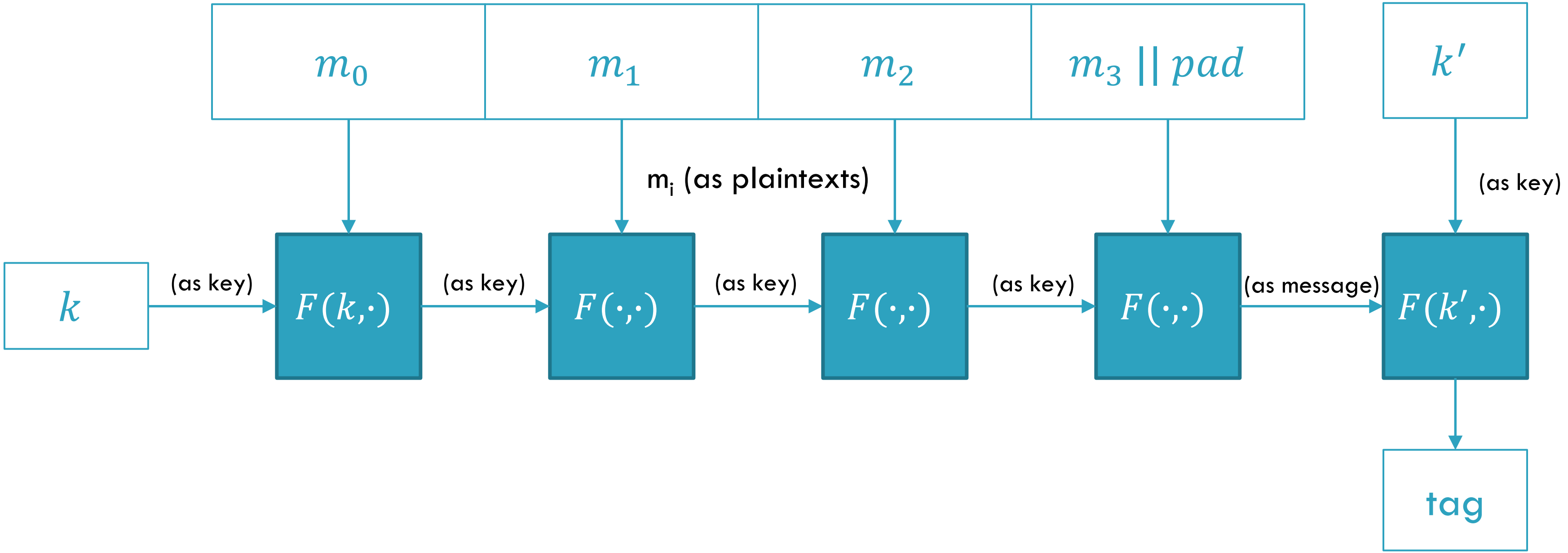
MAC from symmetric ciphers

- In this section we introduce two implementations of a secure MAC:
  - The CBC-MAC, built from the CBC mode of operation
  - The Nested-MAC (NMAC), a more natural construction
- Both constructions need:
  - A secure block cipher $F(key, message)$ (defined in the lecture CR_1.3 – Block Ciphers)
  - A pair of keys (k, k') for the block cipher

CBC-MAC scheme



NMAC scheme



m_0
m_1
m_2
$m_3 \,||\, pad$
k'
m_i (as plaintexts)
(as key)
k
(as key)
$F(k,\cdot)$
(as key)
$F(\cdot,\cdot)$
(as key)
$F(\cdot,\cdot)$
(as key)
$F(\cdot,\cdot)$
(as message)
$F(k',\cdot)$
tag

Why the last encryption step?

- Without the last encryption block we can perform a so called 1-chosen message attack
- For example, in CBC-MAC:
  - Choose an arbitrary one-block message m
  - Get the associated tag t
  - Now t is also the tag for the 2-block message $(m, t \oplus m)$

CBC-MAC / NMAC Comparison

- CBC-MAC is usually used with AES:
  - CCM encryption mode used in 802.11i
  - NIST standard called CMAC
- NMAC is not used with AES:
  - Changing the key each time is too slow (because of the key schedule of AES)
  - It is instead used with hash functions (HMAC)

Outline

- Introduction
- MAC from symmetric ciphers: CBC-MAC and NMAC
- MAC from Hash functions: HMAC

Naïve approach

- We want to build a MAC using Hash functions
- First idea:
  - Take a key k, a message m and a collision resistant hash function H
  - Build the MAC as $MAC(k, m) = H(k||m)$
- Issues?

Length extension attack

- This is vulnerable to a *length extension attack*
- We can forge MACs for new messages:
  - Take a message m' and $S(k, m)$
  - Then $\mathrm{S}(\mathrm{k}, m||m') = \mathrm{H}(\mathrm{k}||m||pad||m')$
  - Since $\mathrm{H}(\mathrm{k}||m||pad) = \mathrm{H}(\mathrm{k}||m)$ we simply need to compute the compression for the new blocks, not caring about the key!

A better idea

- We now explain a popular strategy, called HMAC (hash message authentication code)
- Ingredients:
  - A collision resistant* hash function H
  - Two padding strings $ipad, opad$ (fixed and known)
  - A secret key k and a message m

*: collision resistance is not really required here, but the real requirements for H are out of the scope of this lecture

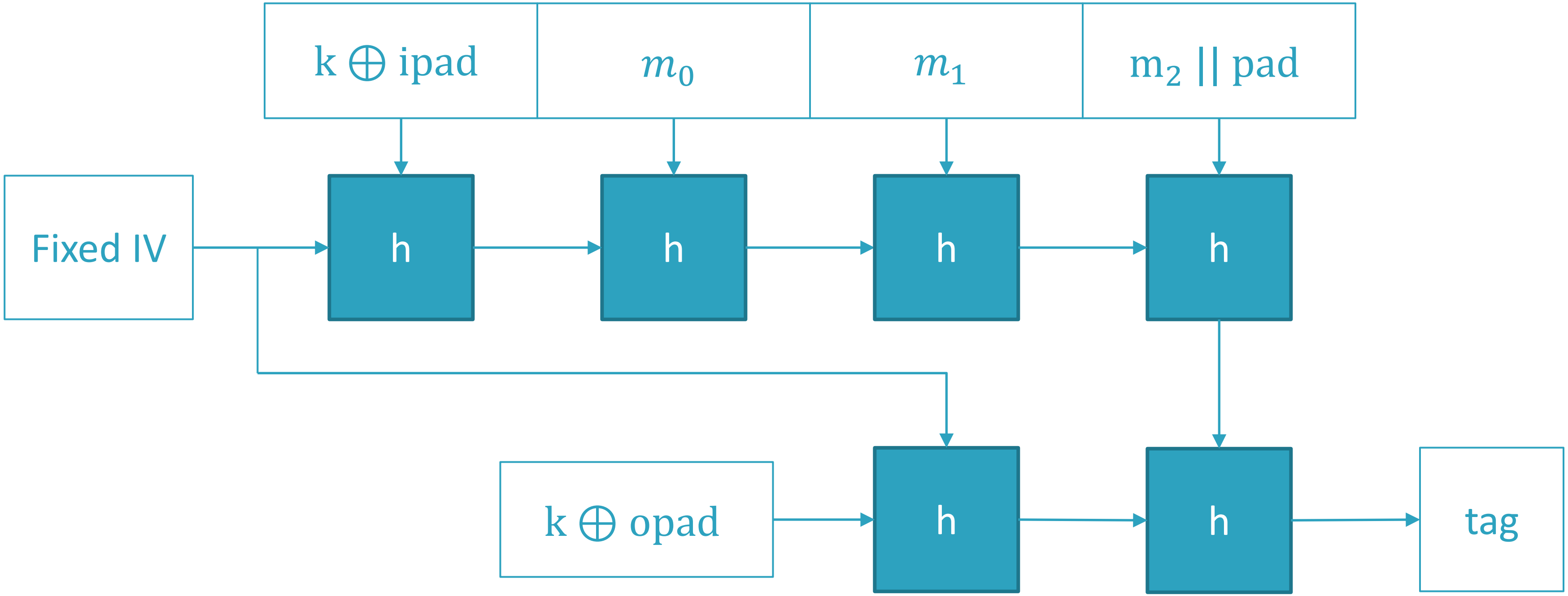
HMAC

- HMAC can be used to verify both integrity and authentication of a message, at the same time
- Any available hash function can be used, like SHA-1 or SHA-2, without having to modify the scheme
- The resulting version of the MAC is called HMAC-X, where X is the used hash function

HMAC

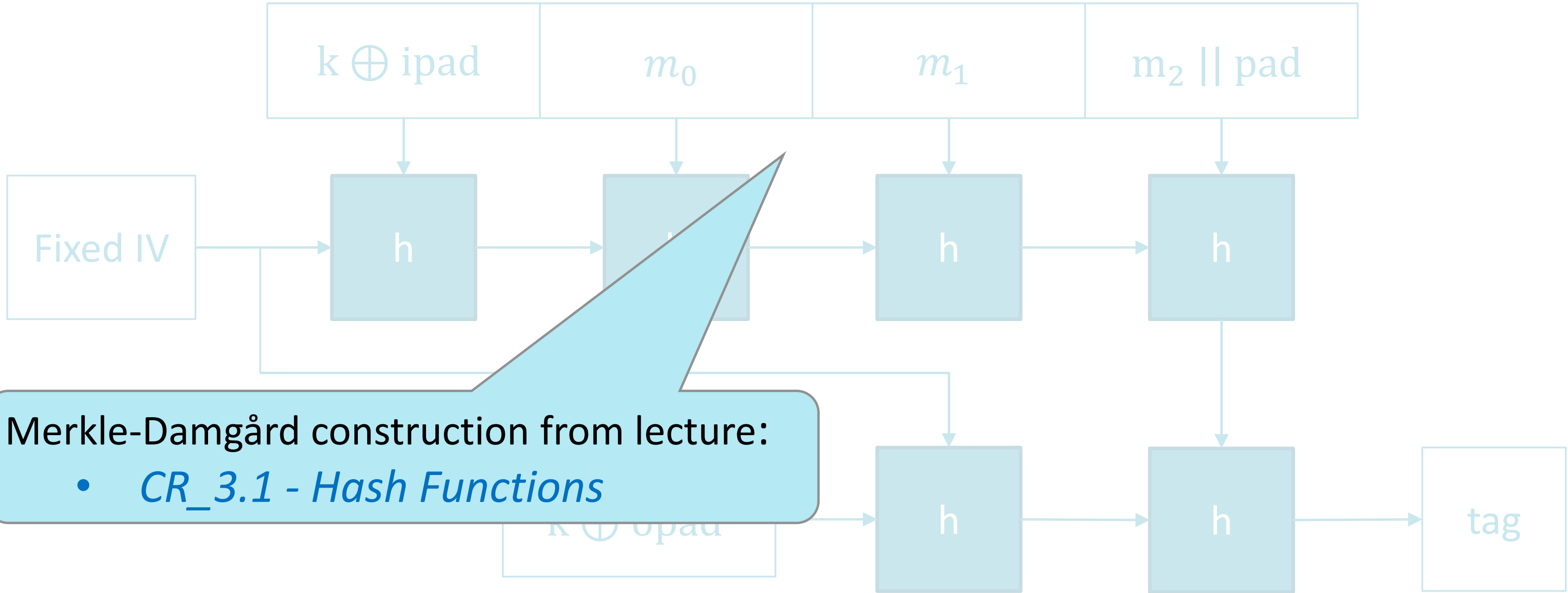
- We define HMAC as:
  - $HMAC_k(m) = H((k \oplus opad) || H((k \oplus ipad) || m))$
- Where:
  - $(k \oplus opad)$ and $(k \oplus ipad)$ are the two "secret keys"
  - The construction is very similar to NMAC, except that the two keys are related
  - Length extension attack cannot be performed, since it needs the knowledge of the result of internal call of the hash function H

HMAC scheme



$k \oplus \text{ipad}$
m_0
m_1
$m_2 \,||\, \text{pad}$
Fixed IV
h
h
h
h
$k \oplus \text{opad}$
h
h
tag

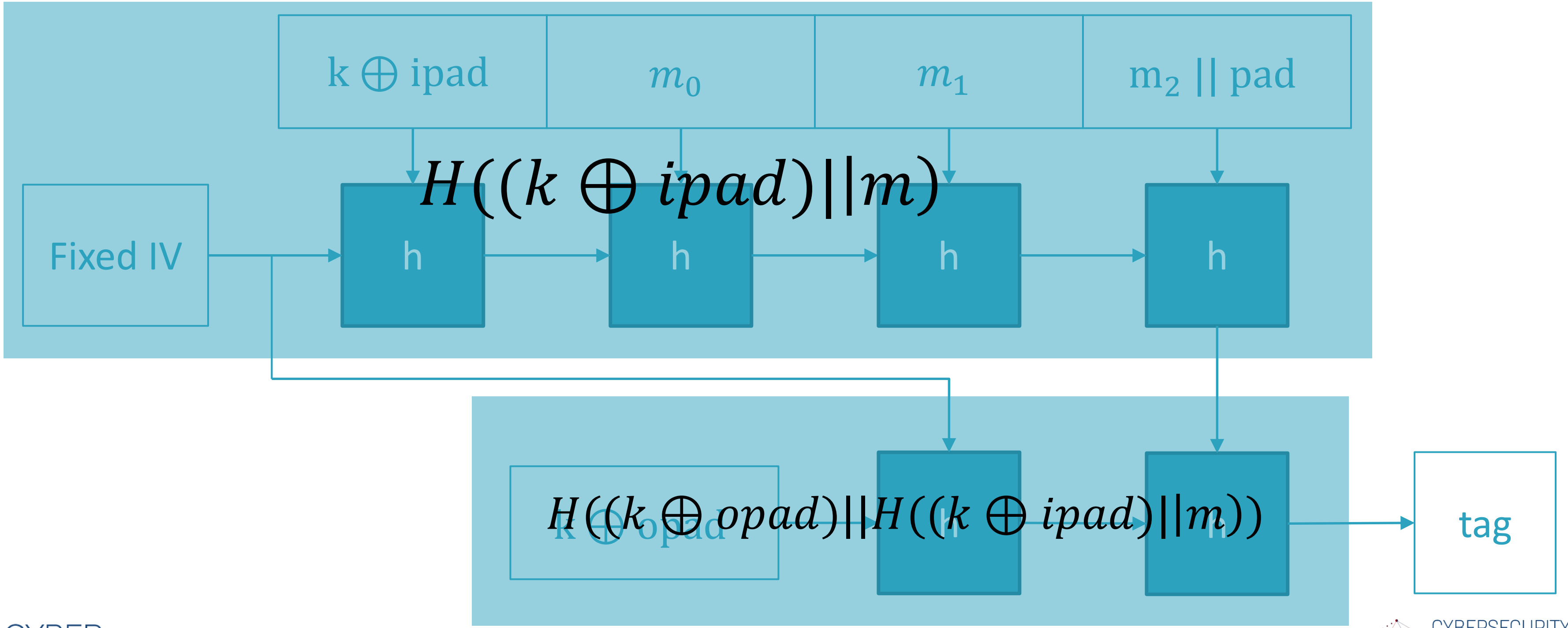
HMAC scheme



Merkle-Damgård construction from lecture:

- *CR_3.1 - Hash Functions*

HMAC scheme



$k \oplus ipad$
m_0
m_1
$m_2 \,||\, pad$
$H((k \oplus ipad)||m)$
Fixed IV
h
h
h
h
$k \oplus opad$
$H((k \oplus opad)||H((k \oplus ipad)||m))$
h
h
tag

Gaspare FERRARO

CyberSecNatLab

Matteo ROSSI

Politecnico di Torino

CYBER CHALLENGE.IT

CYBERSECURITY NATIONAL LABORATORY

Message Authentication Code

https://cybersecnatlab.it