Lecture 6a

Message Authentication II

Stefan Dziembowski

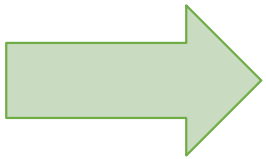
www.crypto.edu.pl/Dziembowski

University of Warsaw

7.11.18

version 1.0

Plan



1. Constructions of MACs from hash functions
2. Authenticated encryption
3. Outlook

Some practitioners don't like the CBC-MAC

They prefer to use the **hash functions** for authentication.

Why?

- hash functions tend to be a bit **more efficient**
- **no export regulations** (important in the past)

How to use hash functions for authentication?

A natural idea used by the practitioners:

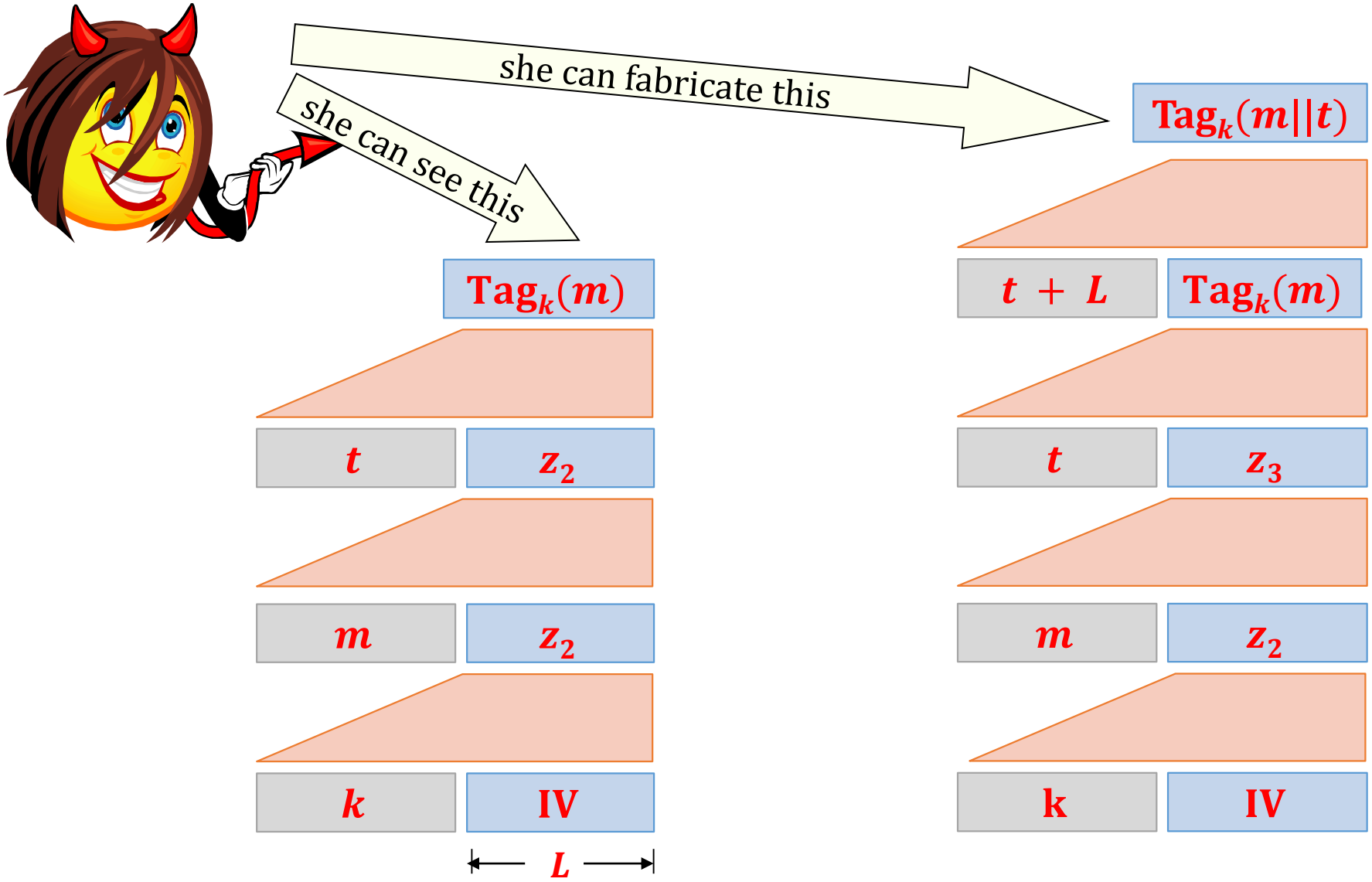
H – hash function

Hash a message together with the key:

$$\mathbf{Tag}_k(m) = H(k \,||\, m)$$

this is not secure!

Message extension attack: Suppose H was constructed using the **MD-transform**



Still, used in practice in the past

For example in **SSL v.2**:

The **MAC-DATA** is computed as follows:

MAC-DATA = HASH[SECRET, ACTUAL-DATA, PADDING-DATA, SEQUENCE-NUMBER]

A better idea

M. Bellare, R. Canetti, and H. Krawczyk (1996):

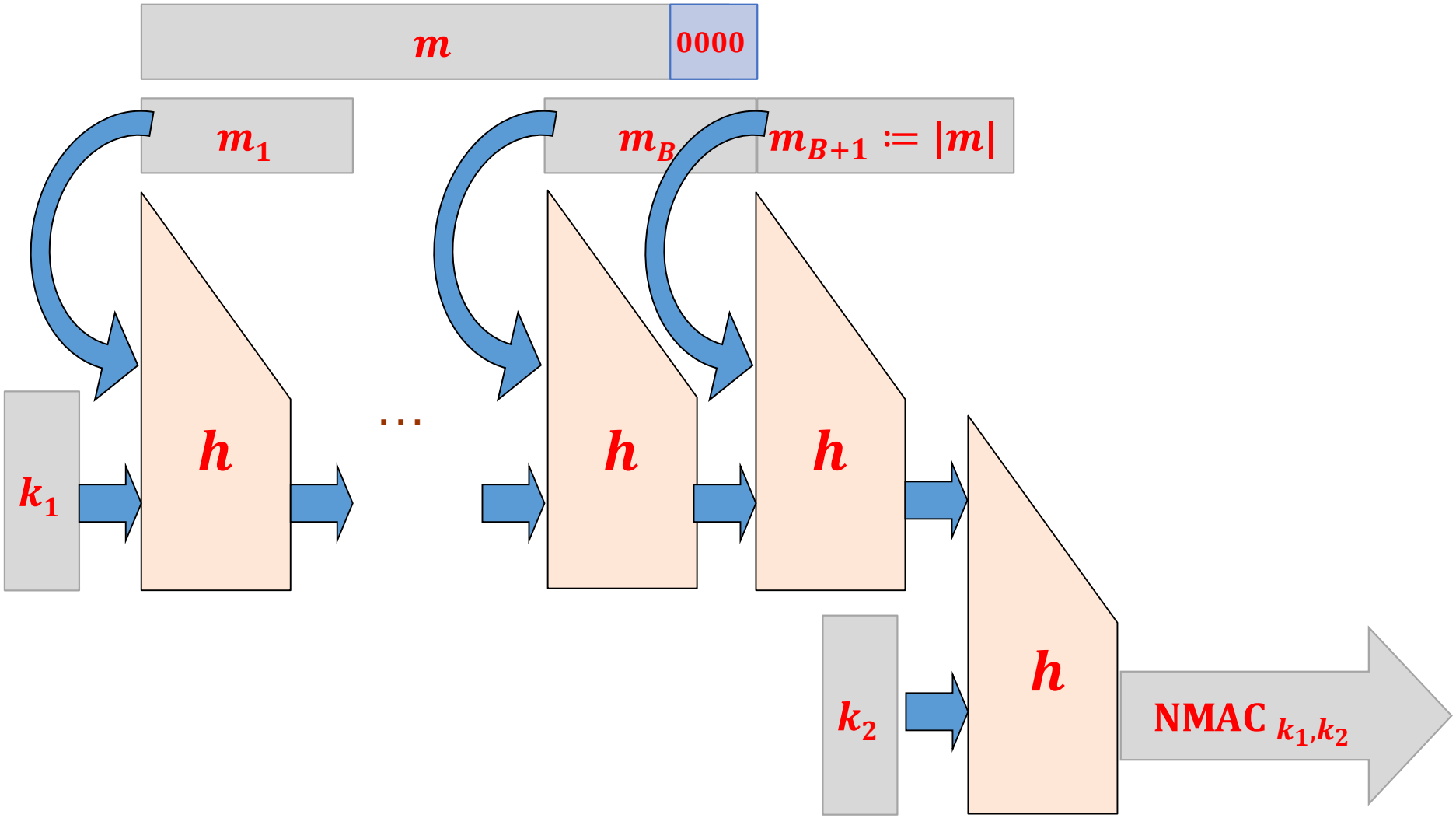
- **NMAC** (Nested MAC)
- **HMAC** (Hash based MAC)

have some "provable properties"

They both use the **Merkle-Damgård** transform.

Again, let $h: \{0,1\}^{2L} \rightarrow \{0,1\}^{L}$ be a compression function.

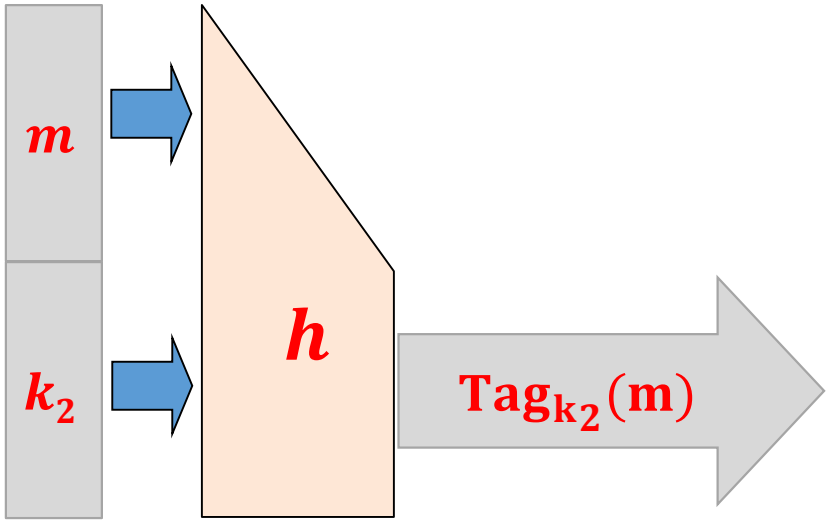
NMAC



What can be proven

Suppose that

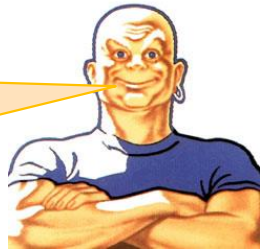
1. h is collision-resistant
2. the following function is a secure **MAC**:



Then NMAC is a secure **MAC**.

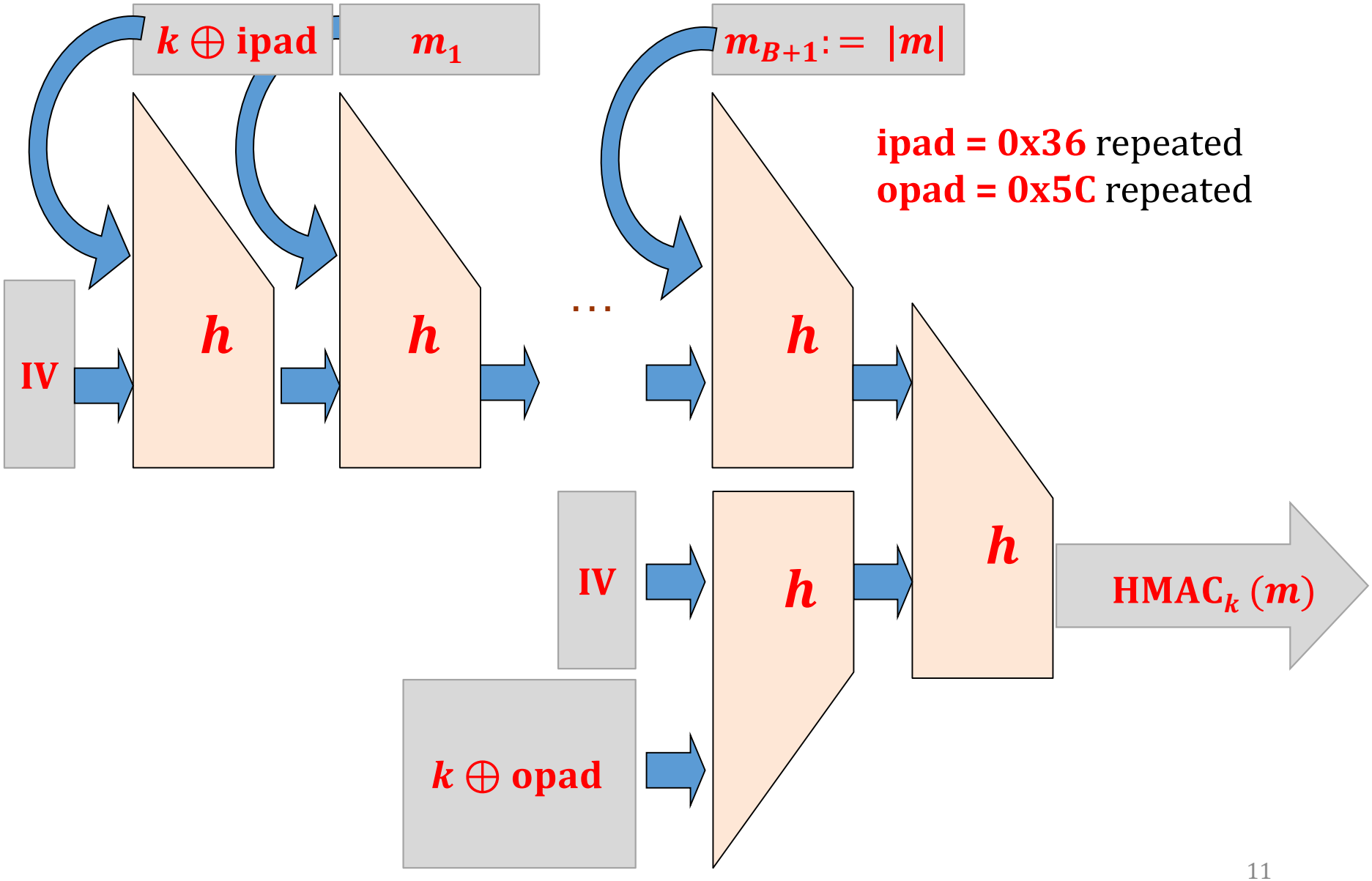
We don't like it:

1. our libraries do not permit to change the **IV**
2. the key is too long: (k_1, k_2)



HMAC is the solution!

HMAC


$k \oplus$ ipad
m_1
$m_{B+1} := |m|$
ipad = 0x36 repeated
opad = 0x5C repeated
IV
h
h
...
h
IV
h
h
$k \oplus$ opad
$\text{HMAC}_k(m)$

Why such a choice for **ipad** and **opad**?

in binary:

ipad = 0x36363636… ← | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |

opad = 0x5C5C5C5C… ← | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |

Properties:

- **simple representation** (easier to implement, less error-prone)
- **Hamming distance** between the pads around $\frac{n}{2}$ (where $n = |\mathbf{opad}| = |\mathbf{ipad}|$).

HMAC – the properties

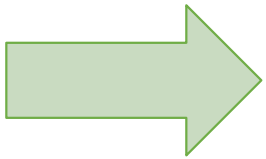
Looks **complicated**, but it is very easy to implement (given an implementation of H):

$$\mathbf{HMAC}_k(m) = H((k \oplus \mathbf{opad}) \,||\, H(k \oplus \mathbf{ipad} \,||\, m))$$

It has some "provable properties" (slightly weaker than **NMAC**).

Widely used in practice.

Plan



1. Constructions of MACs from hash functions
2. Authenticated encryption
3. Outlook

What is needed to establish secure channels?

In practice one needs both

encryption

and

authentication.

This can be achieved as follows:

- **combine encryption** with **authentication**

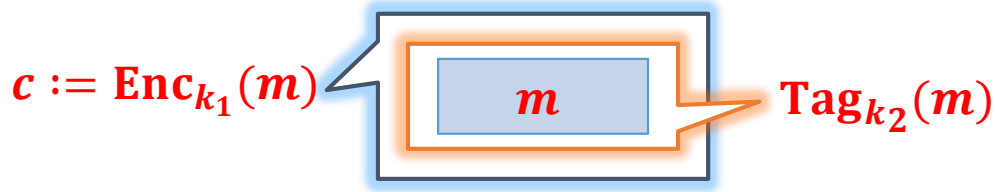
or

- design "**authenticated encryption**" from scratch.

Authentication + encryption, options:

- **Encrypt-and-authenticate:**

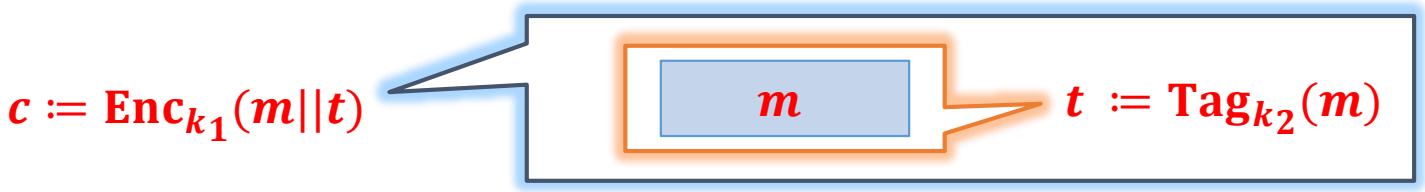
$c := \mathrm{Enc}_{k_1}(m)$ and $t := \mathrm{Tag}_{k_2}(m)$, send (c, t)



wrong

- **Authenticate-then-encrypt:**

$t := \mathrm{Tag}_{k_2}(m)$ and $c := \mathrm{Enc}_{k_1}(m||t)$, send (c, t)

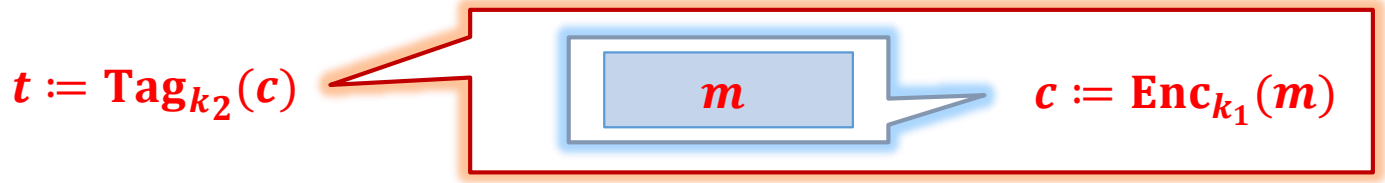


better

- **Encrypt-then-authenticate:**

$c := \mathrm{Enc}_{k_1}(m)$ and $t := \mathrm{Tag}_{k_2}(c)$, send (c, t)

the best



By the way…

Never use the same key for encryption and authentication.

Actually:

Never use the **same key in two different applications** (or two different instantiations of the same application).

Authenticated encryption

In principle: should be more efficient than the

A popular method: **Galois/Counter Mode**.

An ongoing competition for a new authenticated encryption **scheme**:

CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustness

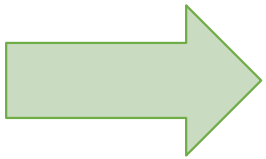
not formally organized by any institution, supported by a grant from NIST
webpage: competitions.cr.yp.to/caesar.html

Caesar competition **finalists** (March 2013)

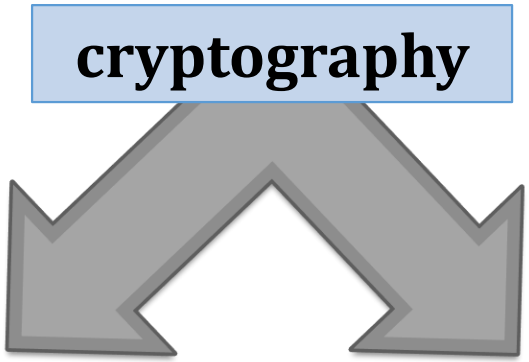
candidate	designers
ACORN	Hongjun Wu
AEGIS	Hongjun Wu, Bart Preneel
Ascon	Christoph Dobraunig, Maria Eichlseder, Florian Mendel, Martin Schläffer
COLM	Elena Andreeva, Andrey Bogdanov, Nilanjan Datta, Atul Luykx, Bart Mennink, Mridul Nandi, Elmar Tischhauser, Kan Yasuda
Deoxys-II	Jérémy Jean, Ivica Nikolić, Thomas Peyrin, Yannick Seurin
MORUS	Hongjun Wu, Tao Huang
OCB	Ted Krovetz, Phillip Rogaway

Plan

1. Constructions of MACs from hash functions
2. Authenticated encryption
3. Outlook



Outlook



"information-theoretic", "unconditional"

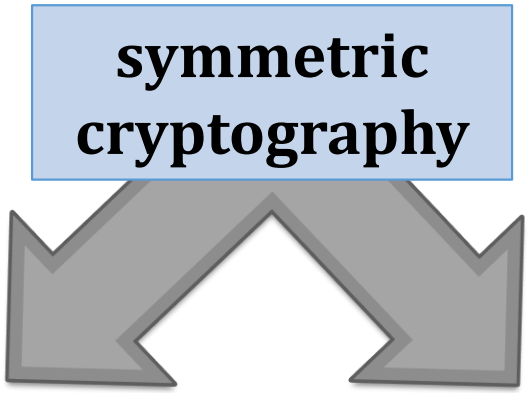
- one time pad,
- quantum cryptography,
- ...

"computational"

based on **2** simultanious assumptions:

1. some problems are computationally difficult
2. our understanding of what "computational difficulty" means is correct.

Symmetric cryptography



encryption **authentication**

(and **hash functions**)

The basic information-theoretic tool

xor (one-time pad)

Basic tools from the computational cryptography

- **one-way functions**
- **pseudorandom generators**
- **pseudorandom functions/permutations**
- **hash functions**

A method for proving security: **reductions**