Message Authentication Code

Elena Kirshanova

Course “Information and Network Security”
Lecture 5
4 апреля 2020 г.

Confidentiality & Integrity

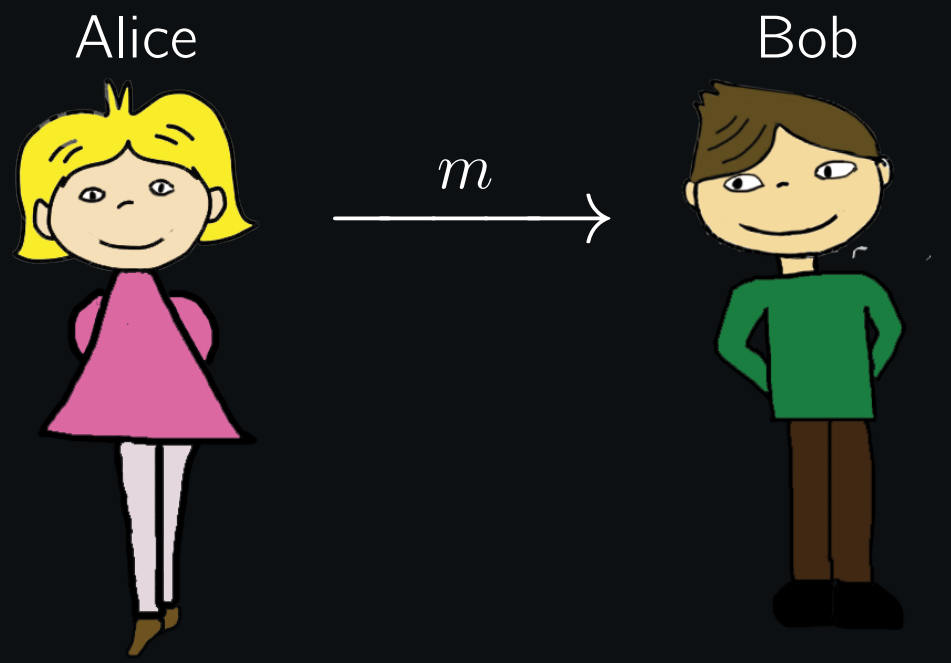
- Previous lectures dealt with confidentiality
- This lecture's goal: integrity (целостность)

Primitive: Message Authentication Code (MAC)

Имитовставка или Код Аутентификации Сообщения

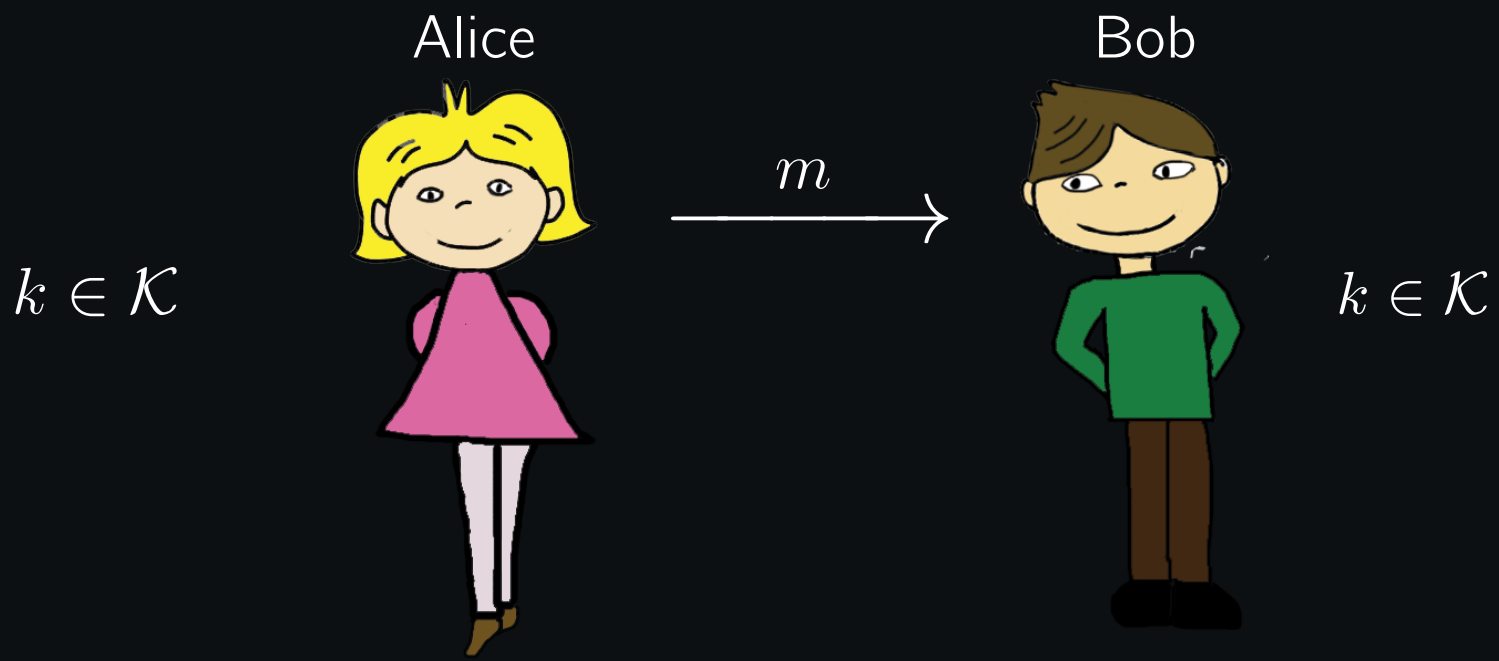
MAC: definition

Goal: send a message m from Alice to Bob s.t. an adversary cannot modify m without Bob noticing



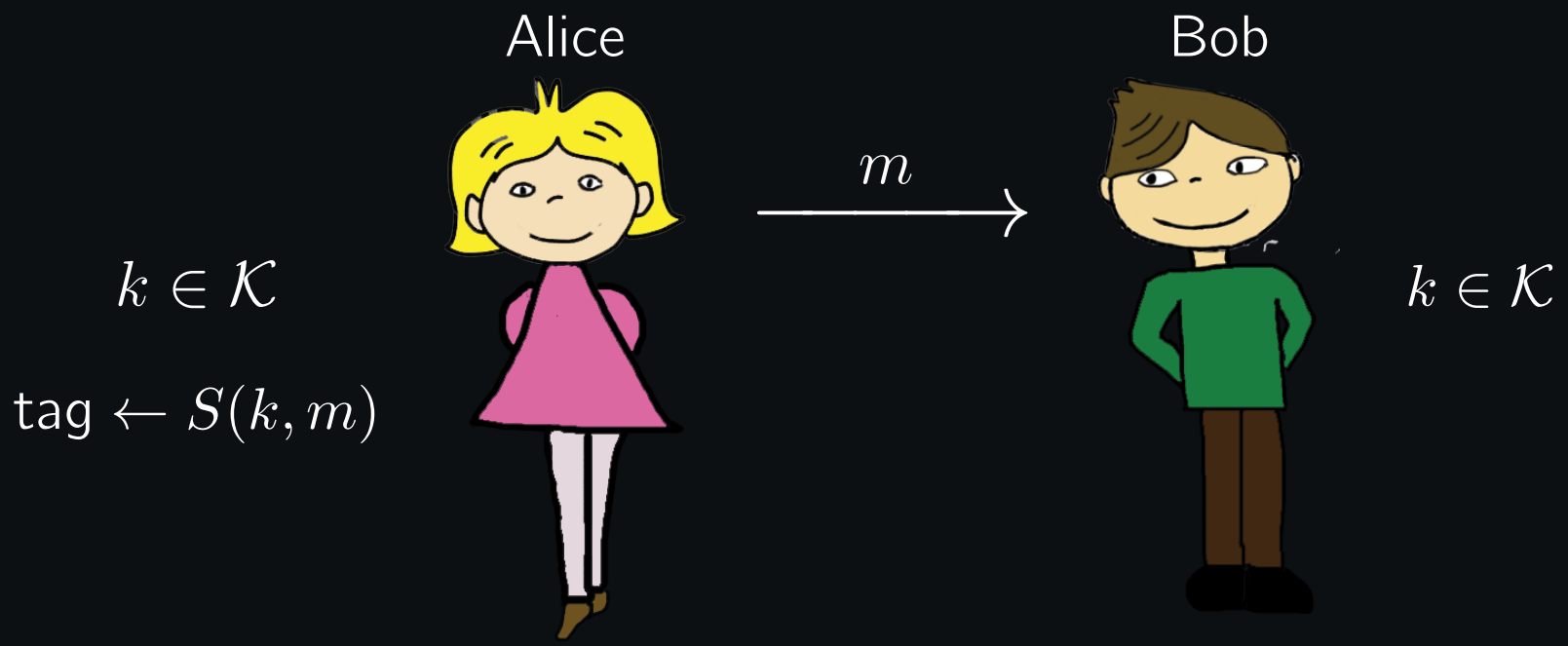
MAC: definition

Goal: send a message m from Alice to Bob s.t. an adversary cannot modify m without Bob noticing



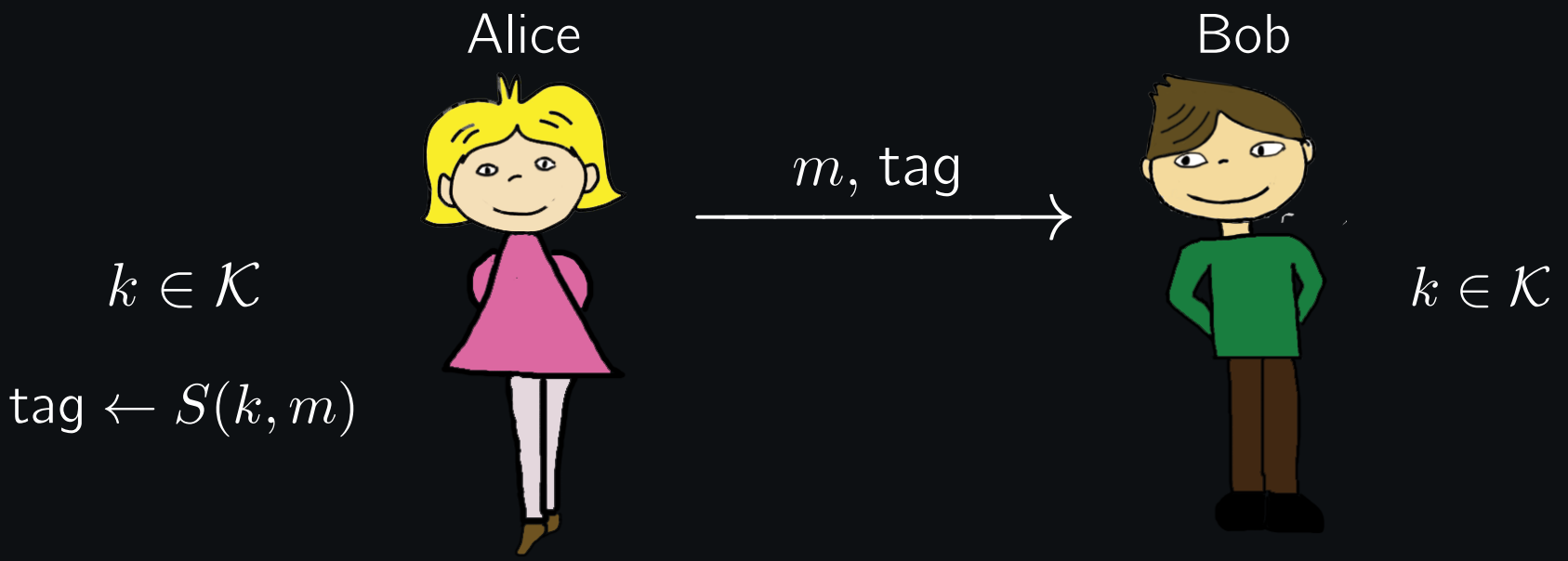
MAC: definition

Goal: send a message m from Alice to Bob s.t. an adversary cannot modify m without Bob noticing



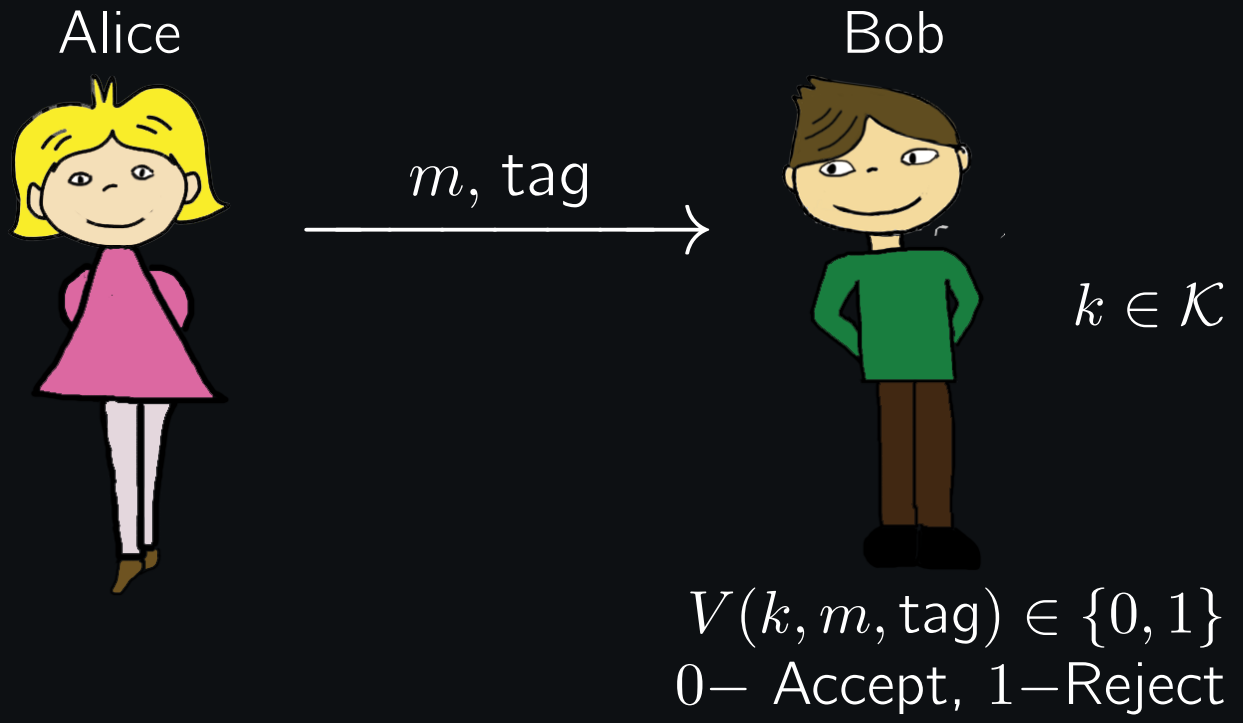
MAC: definition

Goal: send a message m from Alice to Bob s.t. an adversary cannot modify m without Bob noticing



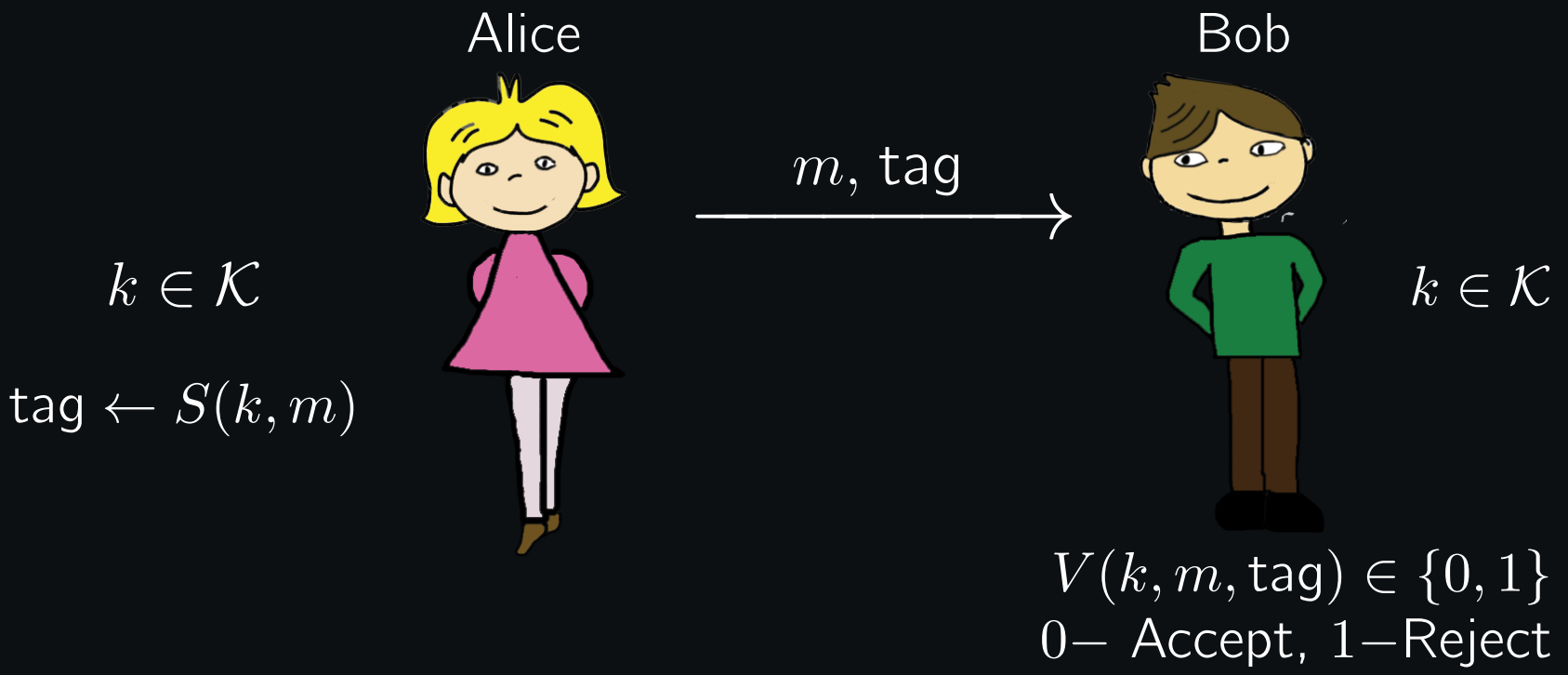
MAC: definition

Goal: send a message m from Alice to Bob s.t. an adversary cannot modify m without Bob noticing



MAC: definition

Goal: send a message m from Alice to Bob s.t. an adversary cannot modify m without Bob noticing



A Message Authentication Code consists of three ppt algorithms:

- Key generation: $\text{KeyGen}(1^\lambda) : k \leftarrow \mathcal{K}$
- Tag generation: $S(k, m) : \text{tag} \leftarrow \mathcal{T}$
- Tag verification: $V(k, m, \text{tag}) : \{0, 1\}$

MAC: correctness and security

A Message Authentication Code consists of three ppt algorithms:

- Key generation: $\mathsf{KeyGen}(1^\lambda) : k \leftarrow \mathcal{K}$
- Tag generation: $S(k, m) : \mathsf{tag} \leftarrow \mathcal{T}$
- Tag verification: $V(k, m, \mathsf{tag}) : \{0, 1\}$

MAC: correctness and security

A Message Authentication Code consists of three ppt algorithms:

- Key generation: $\mathsf{KeyGen}(1^\lambda) : k \leftarrow \mathcal{K}$
- Tag generation: $S(k, m) : \mathsf{tag} \leftarrow \mathcal{T}$
- Tag verification: $V(k, m, \mathsf{tag}) : \{0, 1\}$

Correcntess: $V(k, m, S(k, m)) = \mathsf{Accept}\ \forall\ m \in \mathcal{M}, k \in \mathcal{K}$

Security: For a ppt adversary $\mathcal{A}$ who sees many (message, tag) pairs $\{(m_1, t_1), \ldots, (m_N, t_N)\}$ of its choice, $\mathcal{A}$ cannot produce a new valid (message, tag) pair (m, t)

$$(m, t) \notin \{(m_1, t_1), \ldots, (m_N, t_N)\}$$

Therefore, secure tag length: $96, 128, 256$ bits.

Constructions of MACs

1. A block-cipher (AES, GOST) can be used to instantiate MAC for 16-bytes messages

2. For longer messages:
   – CBC-MAC (integrity of bank transactions)
   Standards: ANSI, FIPS 186-3, GOST
   – HMAC (SSL, IPSec)

Constructions of MACs

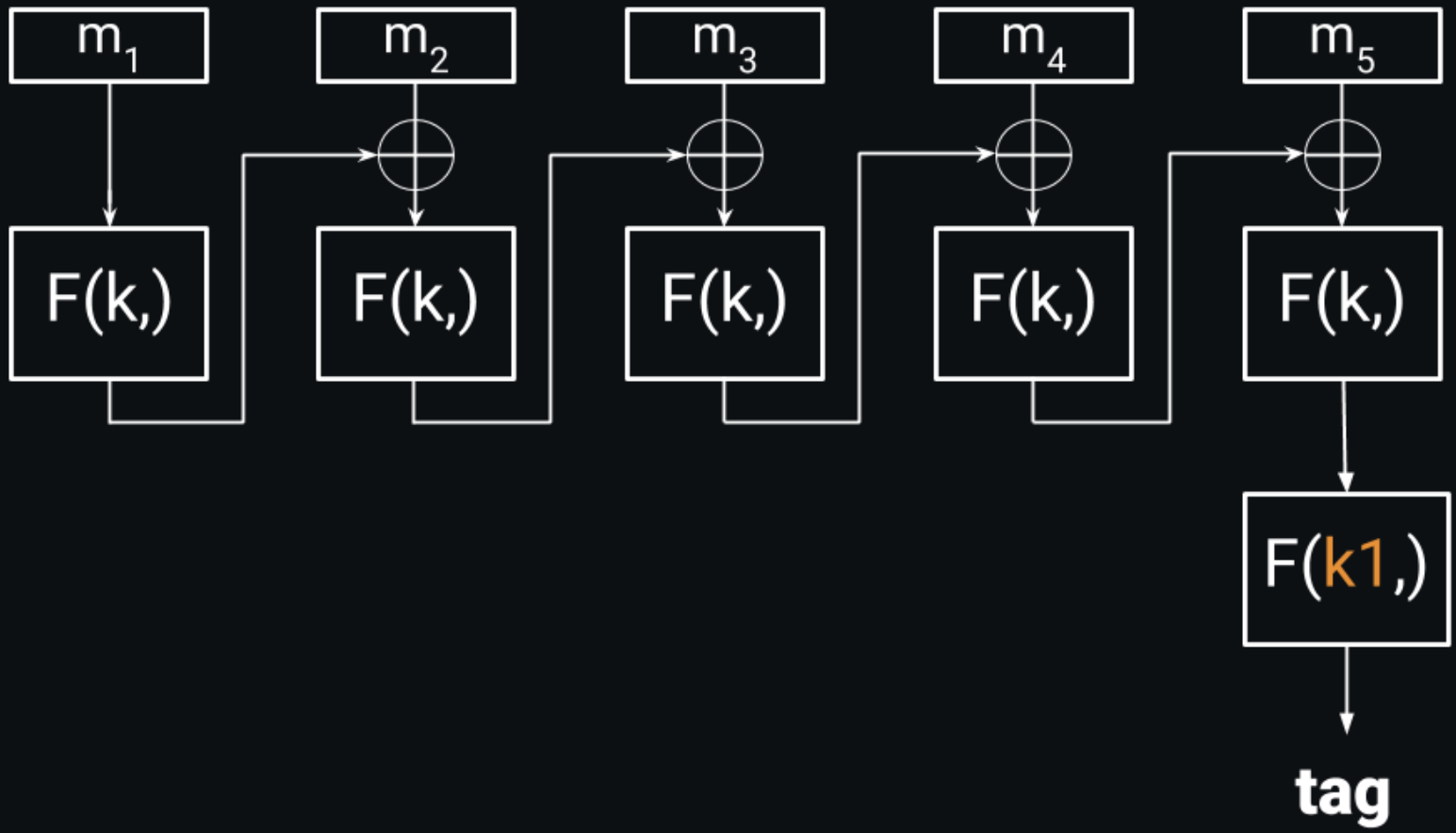
1. A block-cipher (AES, GOST) can be used to instantiate MAC for 16-bytes messages

2. For longer messages:
   – CBC-MAC (integrity of bank transactions)
   Standards: ANSI, FIPS 186-3, GOST
   – HMAC (SSL, IPSec)

This time: CBC-MAC

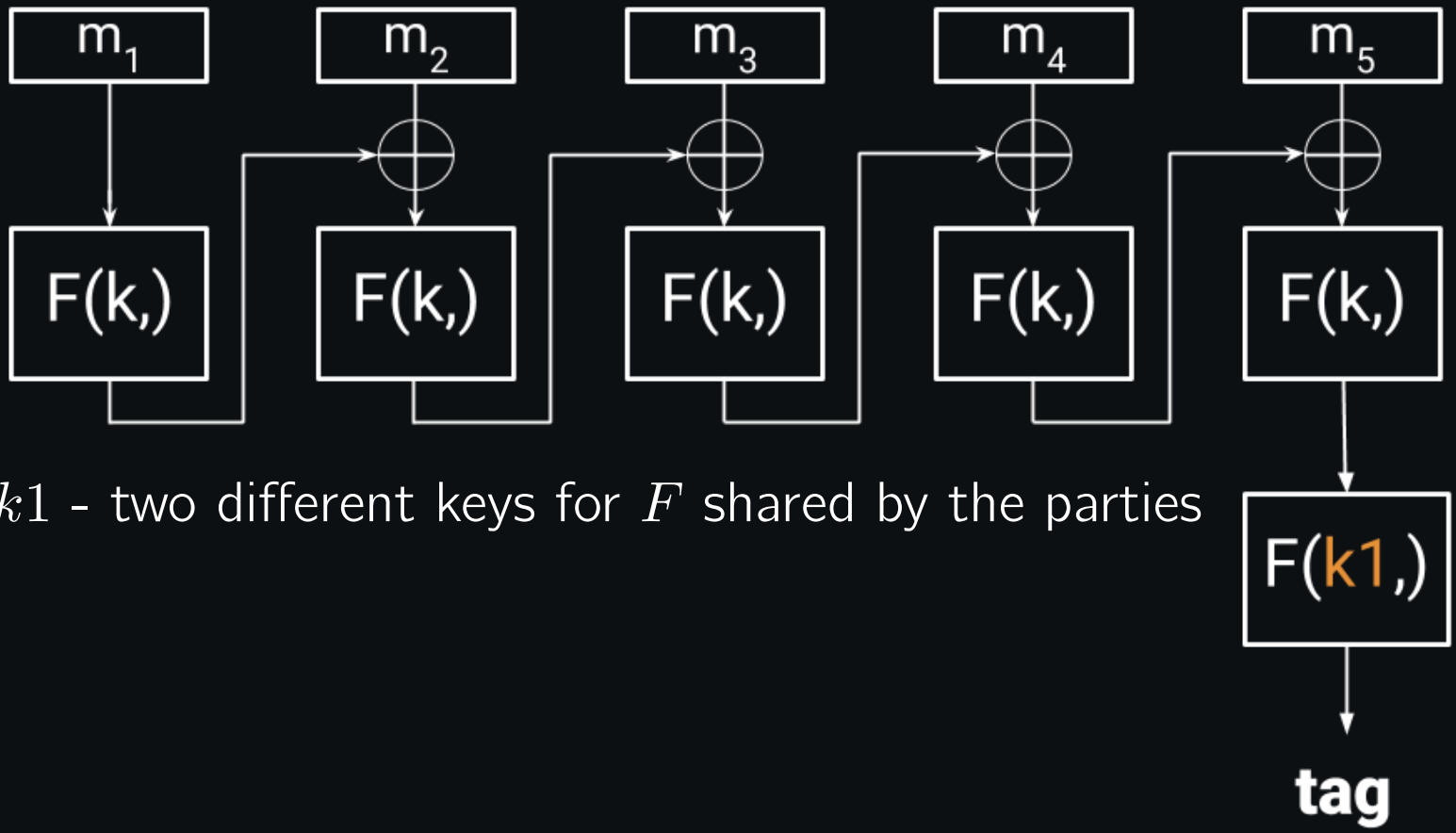
CBC-MAC

$m = (m_1, m_2, m_3, ...)$, $F-$ a block cipher of block length $|m_i|$



CBC-MAC

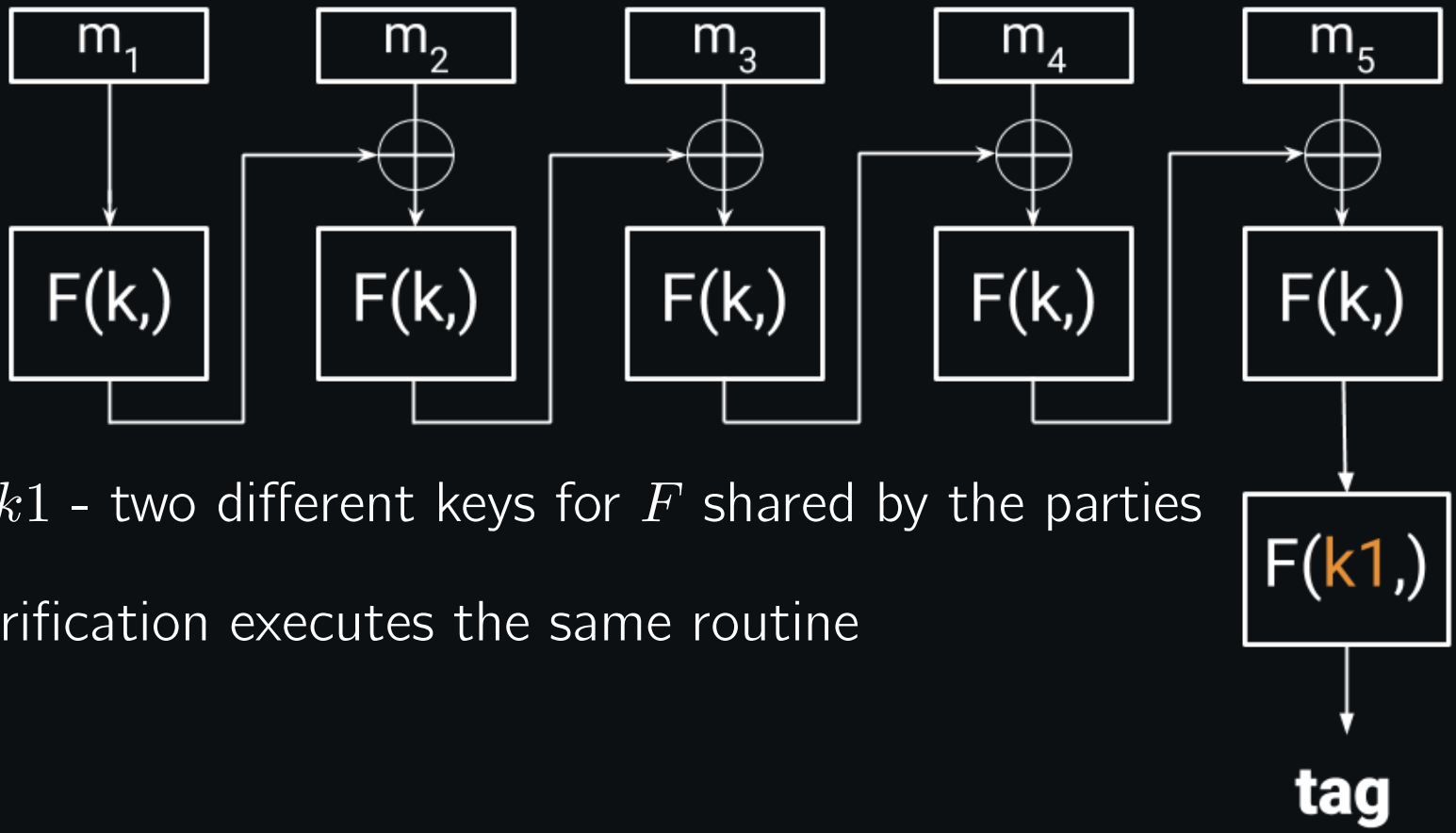
$m = (m_1, m_2, m_3, ...)$, $F-$ a block cipher of block length $|m_i|$



$k, k1$ - two different keys for F shared by the parties

CBC-MAC

$m = (m_1, m_2, m_3, ...)$, $F-$ a block cipher of block length $|m_i|$

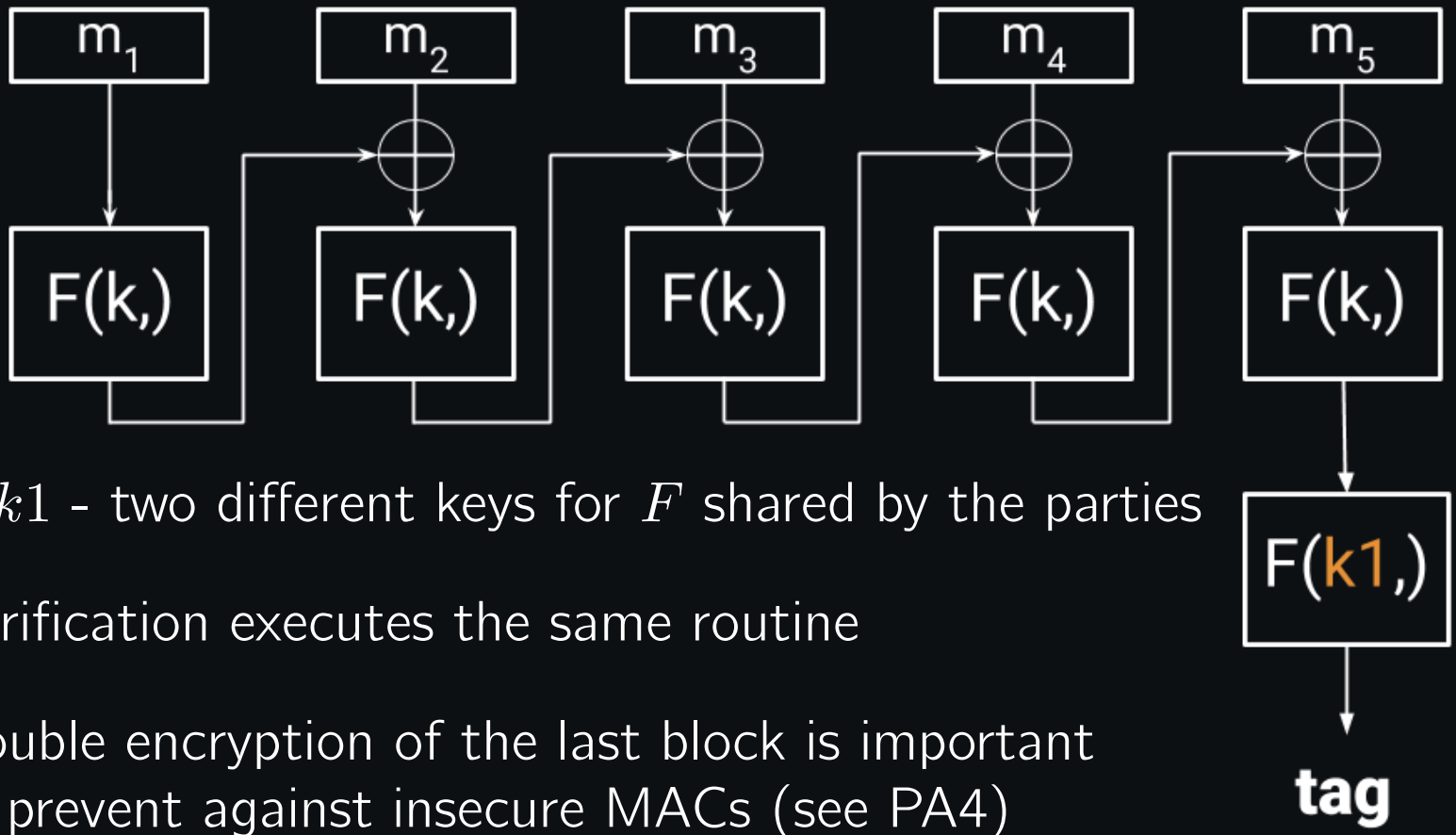


$k, k1$ - two different keys for F shared by the parties

Verification executes the same routine

CBC-MAC

$m = (m_1, m_2, m_3, ...)$, $F-$ a block cipher of block length $|m_i|$



$k, k1$ - two different keys for F shared by the parties

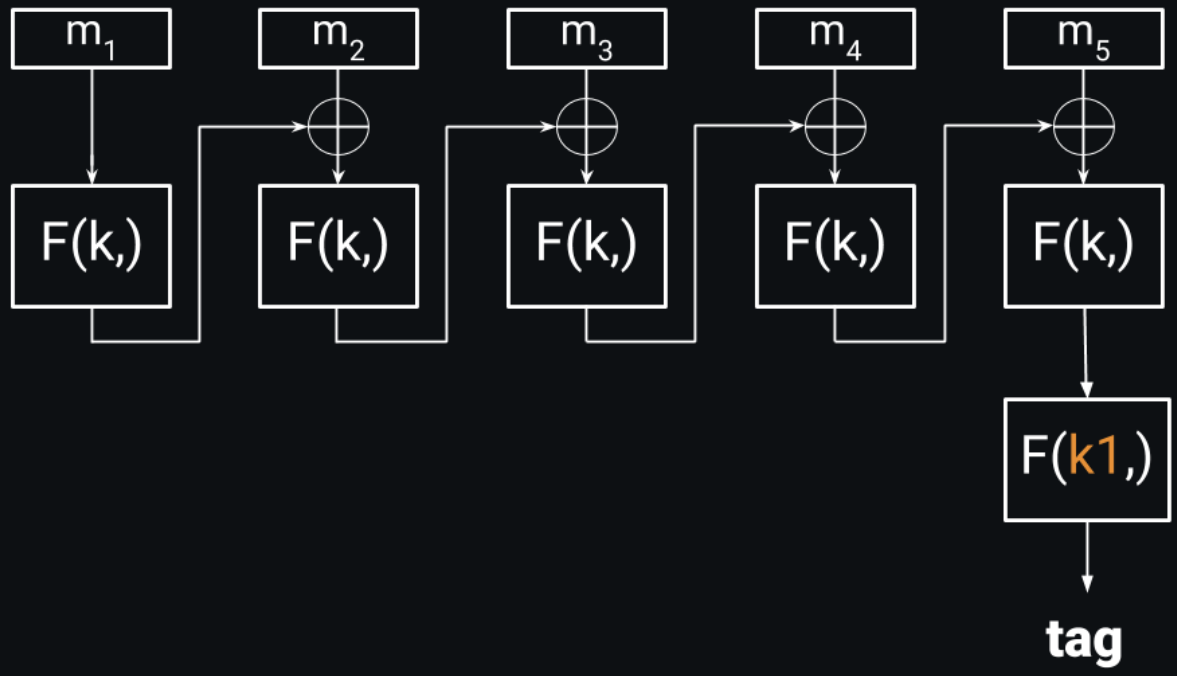
Verification executes the same routine

Double encryption of the last block is important
to prevent against insecure MACs (see PA4)

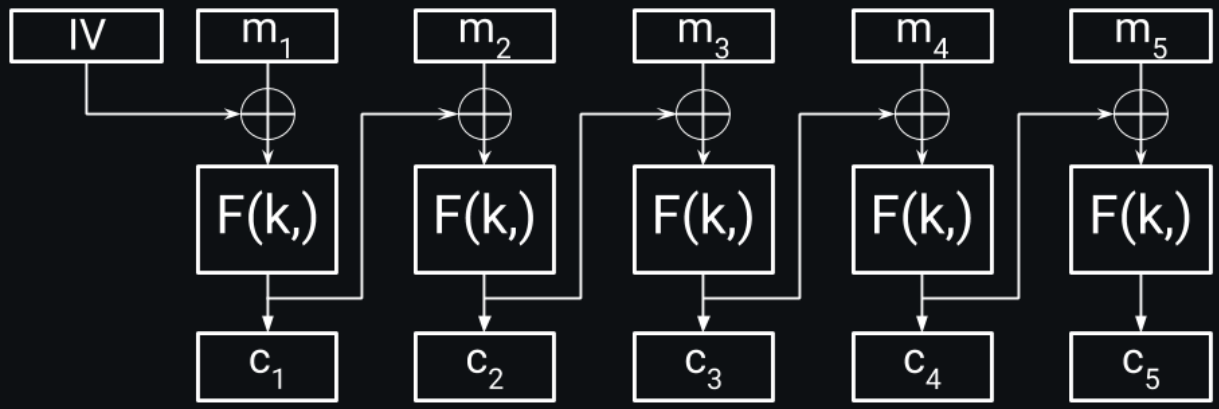
The output tag can be truncated to the desired bit length, but not less than 64 bits.

Compare CBC-MAC with CBC-mode for block-ciphers

CBC-MAC:



CBC-mode:



Padding

What if $\mathbf{m} = (m_1, m_2, ...)$ is not a multiple of the block length?

Example of bad padding: filling up with $0'$s.

$$m = (\star\star\star)$$

that has length $<$ block length. Padding of such message:

$$m_{\text{padded}} = (\star\star\star\, 0 \ldots 0)$$

. Then for another message m':

$$S(k, m_{\text{padded}}) = S(k, m' = (\star\star\star\, 0 \ldots 0))$$

For any m, $S(m) = S(m||0)$.

Padding

What if $\mathbf{m} = (m_1, m_2, ...)$ is not a multiple of the block length?

Example of bad padding: filling up with $0'$s.

$$m = (\star\star\star)$$

that has length $<$ block length. Padding of such message:

$$m_{\text{padded}} = (\star\star\star\, 0 \ldots 0)$$

. Then for another message m':

$$S(k, m_{\text{padded}}) = S(k, m' = (\star\star\star\, 0 \ldots 0))$$

For any m, $S(m) = S(m||0)$.

For $\mathbf{m} \neq \mathbf{m}'$ we want $\mathbf{m}||\text{pad} \neq \mathbf{m}'||\text{pad}$. In particular, padding must be $1 \leftrightarrow 1$.

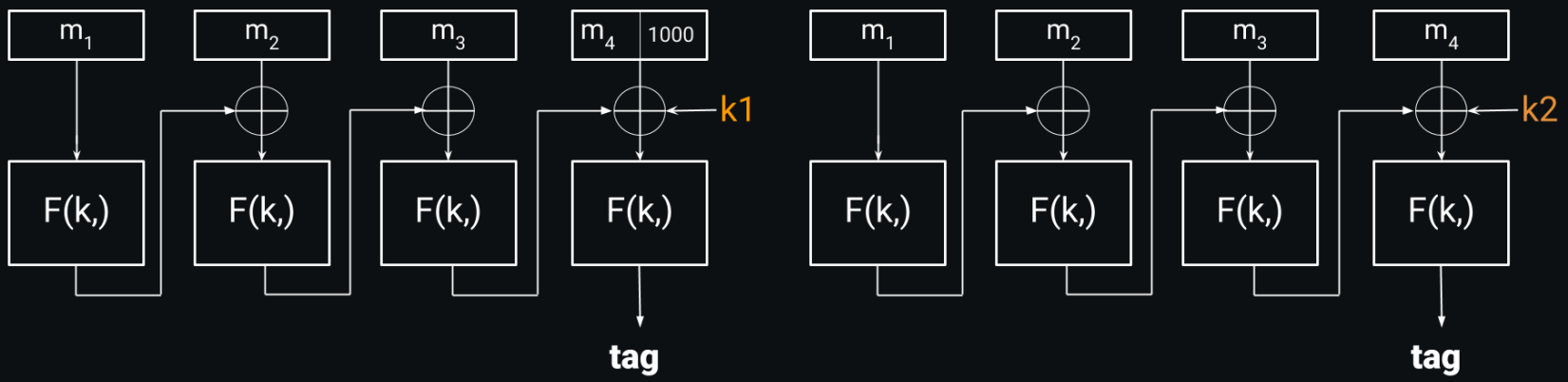
Padding

1. ISO. Pad with $10\ldots0$. '1' indicates beginning of the padding. Add a dummy block if $|m| <$ block length

Padding

1. ISO. Pad with $10 \ldots 0$. '1' indicates beginning of the padding. Add a dummy block if $|m| <$ block length

2. NIST, GOST



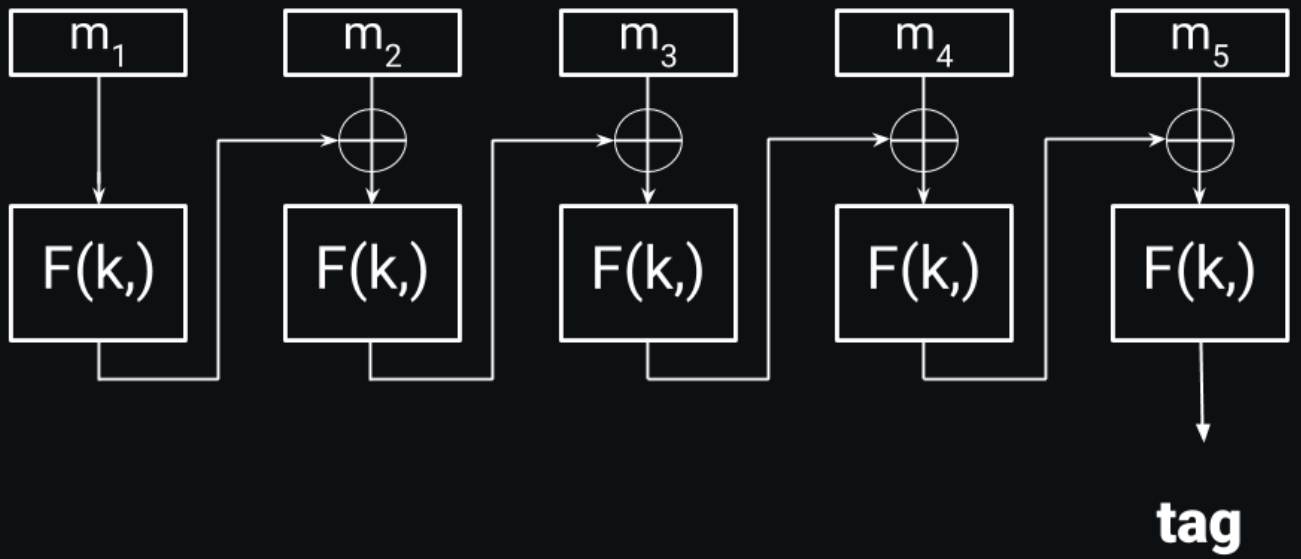
$m_4 <$ block-length

$m_4 =$ block-length

Advantages: No dummy block, no additional application of F.

Some details on Programming Assignment 4

Assume our tag is generated without the last encryption step for one-block message m ($|m| = 128$ bits).



Your task is to demonstrate an efficient tag forgery for such construction: by having (m, t) - a valid (message, tag) pair, provide another valid pair (m', t).
Hint: m' might consist of two blocks.

Concluding slide

1. Other types of MACs exist: Parallel Mac (PMAC), One-time MAC
2. MACs are not Checksums! MACs require secret keys
3. DO NOT follow the Russian Wikipedia article on MACs
4. For Russian project follow GOST'15 guidelines (link is on the course's webpage)

Next time: Message integrity with cryptographic hash functions