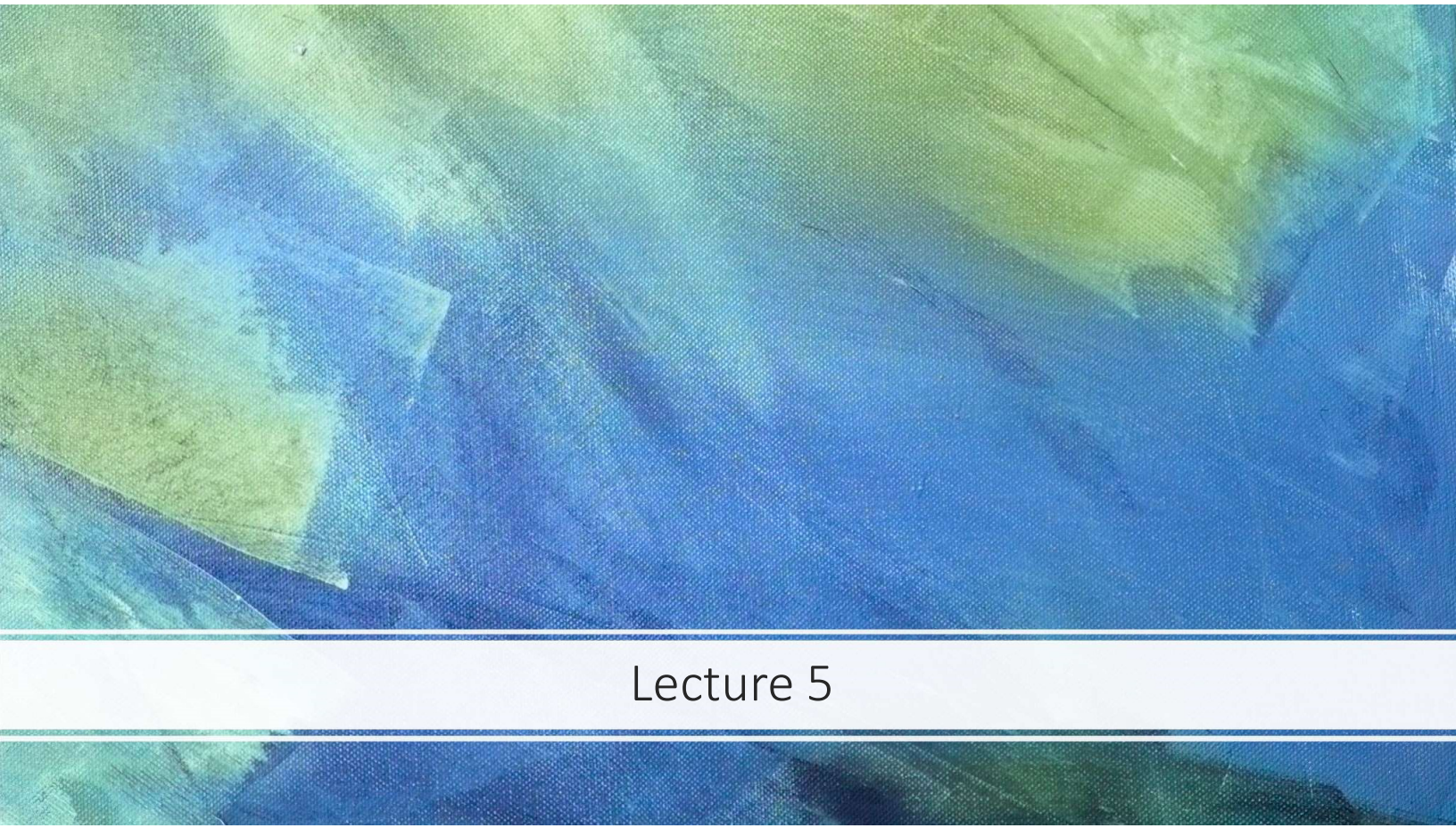
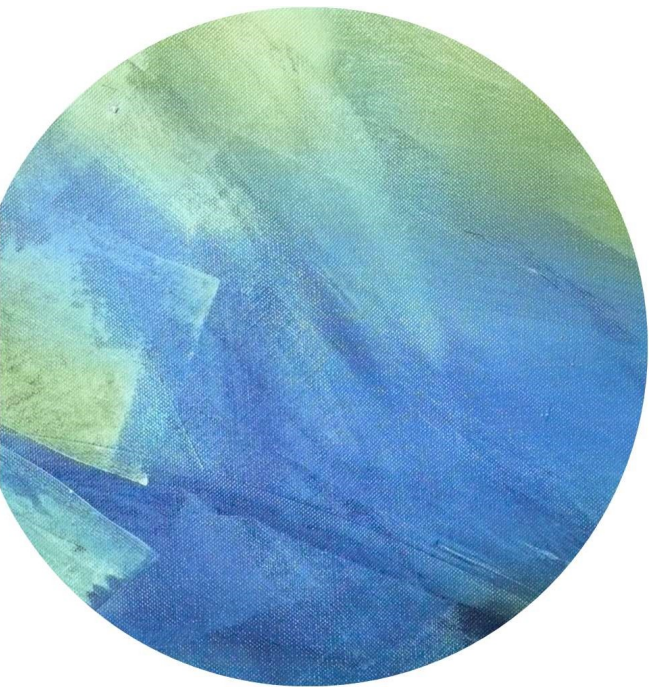

Lecture 5



Outline

Modes of Operation for Block Ciphers

Data Integrity: MACS

Administrative Details

- Course website: https://courses.grainger.illinois.edu/cs498ac3/fa2020/
- Has syllabus, instructor and TA info, office hours
- IMPORTANT: **Join Piazza!** piazza.com/illinois/fall2020/ececs498ac/home

I strongly encourage class participation.
If you don't understand something in class, please interrupt me and ask questions.
Please make abundant use of office hours.

Modes of Operation for Block Ciphers

Many-time Keys

Example applications:

- 1. File systems: Same AES key used to encrypt many files.
- 2. IPSEC (used in VPN): Same AES key used to encrypt many packets.

Defining Security:

Recall: One-time security: $\forall m_0, m_1 \in \{0,1\}^{128}$,

$$E(k, m_0) \underset{k \leftarrow K}{\approx} E(k, m_1) \quad k \leftarrow K$$

Many-time security: $\forall \vec{m_0} = (m_0^1, m_0^2 \ldots\ldots m_0^n)$

$\vec{m_1} = (m_1^1, m_1^2 \ldots\ldots m_1^n)$

$$\left(E(k, m_0^1), E(k, m_0^2) \ldots E(k, m_0^n)\right) \underset{k \leftarrow K}{\approx} \left(E(k, m_1^1), E(k, m_1^2) \ldots E(k, m_1^n)\right) \quad k \leftarrow K$$

Many-time Keys

If secret key is to be used multiple times ⇒
given the same plaintext message twice, encryption must produce different outputs.

Why?

Solutions?

E(k, m) will not work!

* randomize, or
* nonce, or
* counter

Many-time Keys : SOLUTION 1 : PRF

Let F: K × R ⟶ M be a secure PRF.

For m∈M define E(k,m) = [r⟵R, output **(r, F(k,r)⊕m)**]

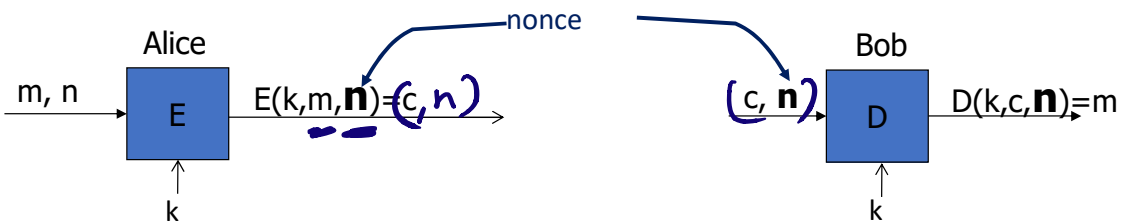
Is E semantically secure under CPA?

$$Dec(r, c, k) = c \oplus F(k, r)$$

$$\left[\oplus F(k, r_0^1) \quad \oplus F(k, r_0^2) \quad \cdots \quad \oplus F(k, r_0^n) \right] \approx \left[\oplus \text{random}^{m_0^1} \quad \oplus \text{random}^{m_0^2} \quad \cdots \quad \text{random}^{m_0^n} \right]$$

$$= [\text{random} \quad \text{random} \quad \cdots \quad \text{random}]$$

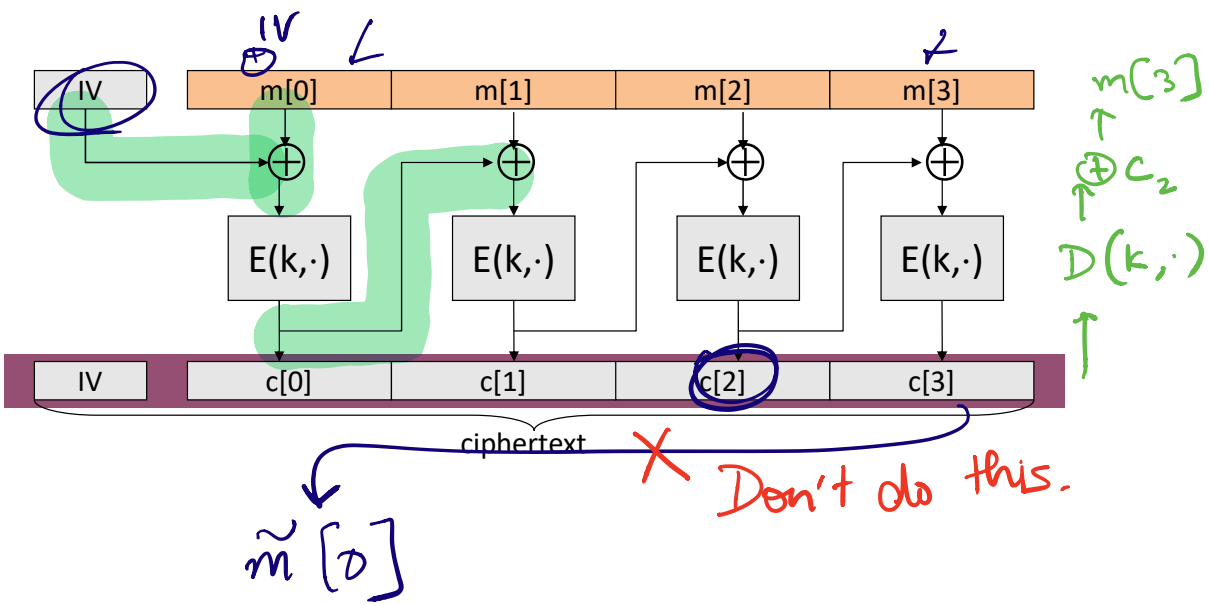
Solution 2: nonce-based Encryption



- nonce n: a value that changes from msg to msg.
  (k,n) pair never used more than once
- method 1: nonce is a **counter** (e.g. packet counter) [SSL, IPSec]
  - used when encryptor keeps state from msg to msg
  - if decryptor has same state, need not send nonce with CT
- method 2: nonce is **random** [File Encryption]

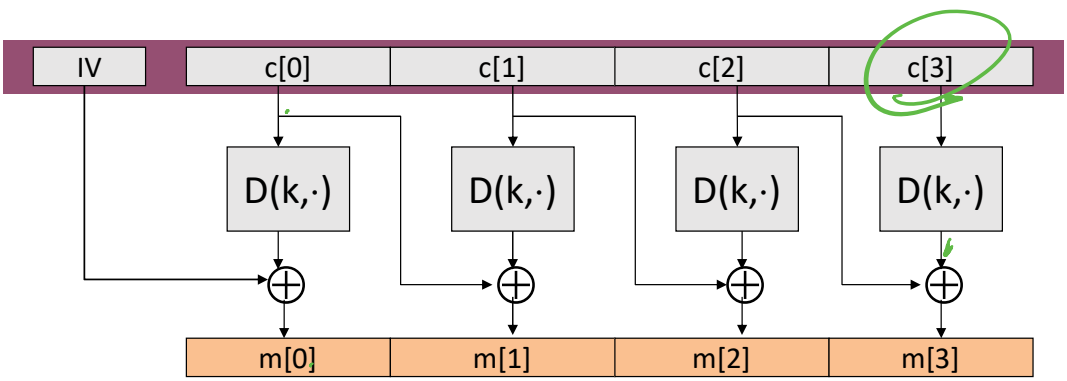
CBC (Cipher Block Chaining) mode

Let (E,D) be a PRP. $E_{CBC}(k,m)$: choose **random** Initialization Vector and do:



Decryption Circuit

In symbols: $c[0] = E(k, IV \oplus m[0])$ $\Rightarrow$ $m[0] = D(k, c[0]) \oplus IV$



Note: choose fresh IV for each message.

CPA Security of CBC

- CBC Theorem: For small enough L>0,
  If E is a secure PRP over (K,X) then
  E_{CBC} is CPA-secure over (K, X^L, X^{L+1}).

- In particular, security error in CBC = (2 x sec. error in PRP) + $(q^2L^2/|X|)$
- What if IV was predictable? Is it still CPA-secure?

messages
blocks
block length
2^{-k}
No.

Bug in SSL/TLS 1.0: IV for record #i is last CT block of record #(i-1)

$\vec{m}_0 = (0, IV_1 \oplus IV_2)$ $\quad \vec{m}_1 = (0, 1)$

$\vec{c}_0 = (E(k, IV_1), E(k, IV_1))$ $\quad \vec{c}_1 = (E(k, IV_1),$ something different$)$

What happens if adversary can predict IV

Insecure!

Challenger

$k \leftarrow K$

IV_1

IV_2

Adv

0

$c_1 \leftarrow (IV_1, E(k, 0 \oplus IV_1))$

$m_0 = IV_1 \oplus IV_2$ $\quad m_1 \neq m_0$

Suppose Adv predicts IV_2

If m_0

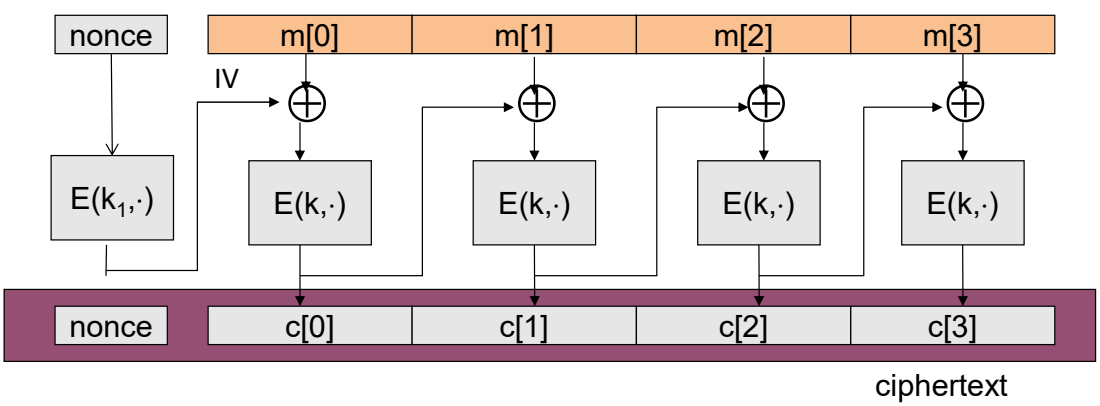
$c_2 \leftarrow (IV_2, ?)$ $\quad E(k, m_0 \oplus IV_2) = E(k, IV_1)$

Else $c_2 \leftarrow ?$

CBC (Cipher Block Chaining) mode: Version 2

- Cipher block chaining with <u>unique</u> nonce: key = (k,k_1)

  unique nonce means: (key, n) pair is used for only one message

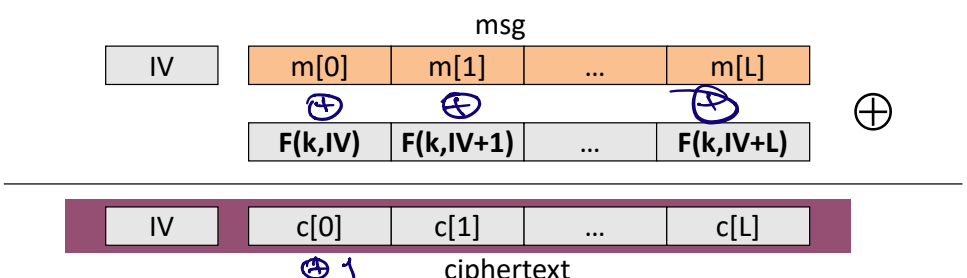


RANDOM - COUNTER

Rand-ctr mode

Let F: $K \times \{0,1\}^n \longrightarrow \{0,1\}^n$ be a secure *PRF*.

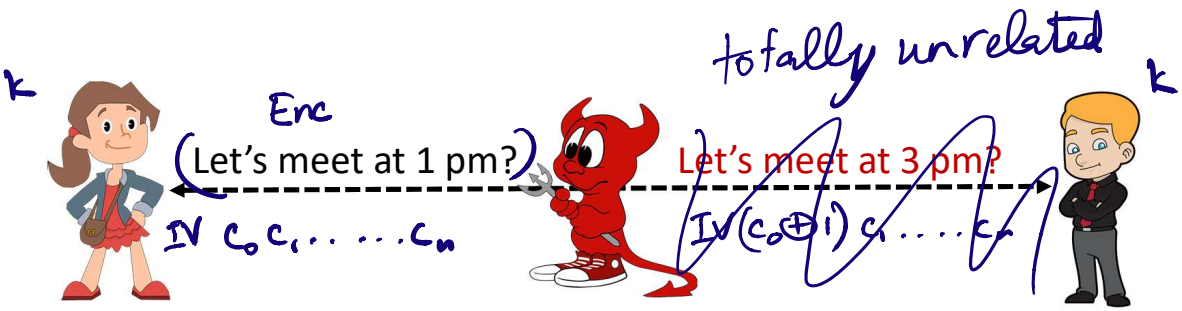
E(k,m): choose a random IV $\in \{0,1\}^n$ and do:



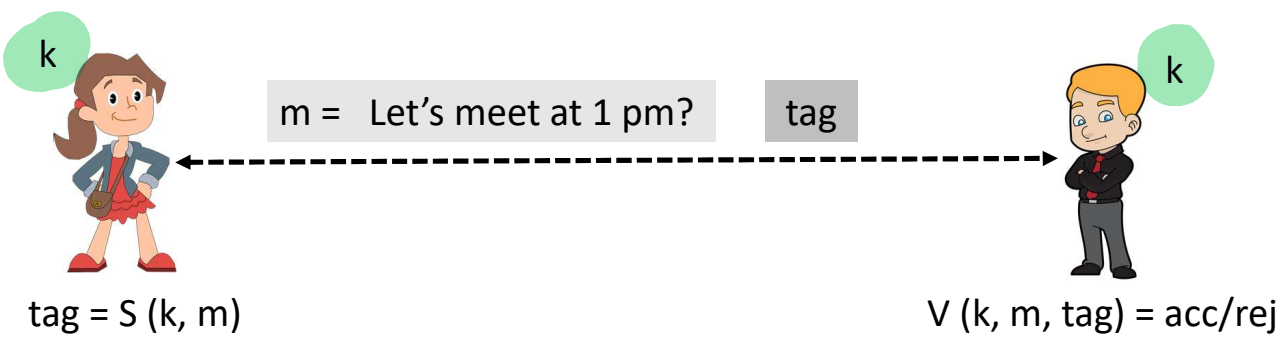
(1) Can use PRF instead of PRP and (2) is parallelizable.

Message Integrity: MACs

Goal: Integrity (not secrecy)



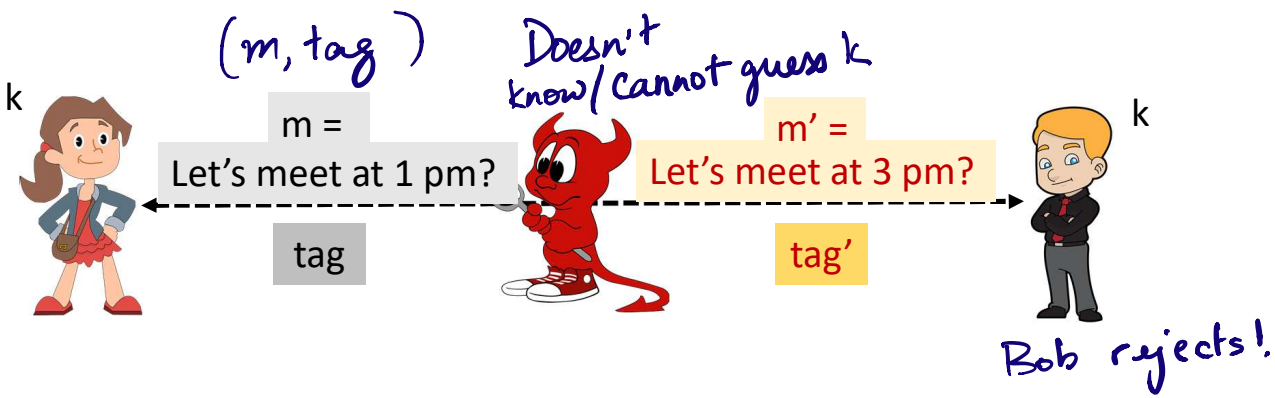
Shared key setting



MAC = (S,V) is a pair of algorithms that satisfy:

1. CORRECTNESS. V(k, m, tag) = acc when tag = S (k, m)

Shared key setting: security of MACs



MAC = (S,V) is a pair of algorithms that satisfy:

2. SECURITY. Attacker unable to compute (m', tag') different from (m, tag) such that V (k, m', tag') = 1 ≈ accepting

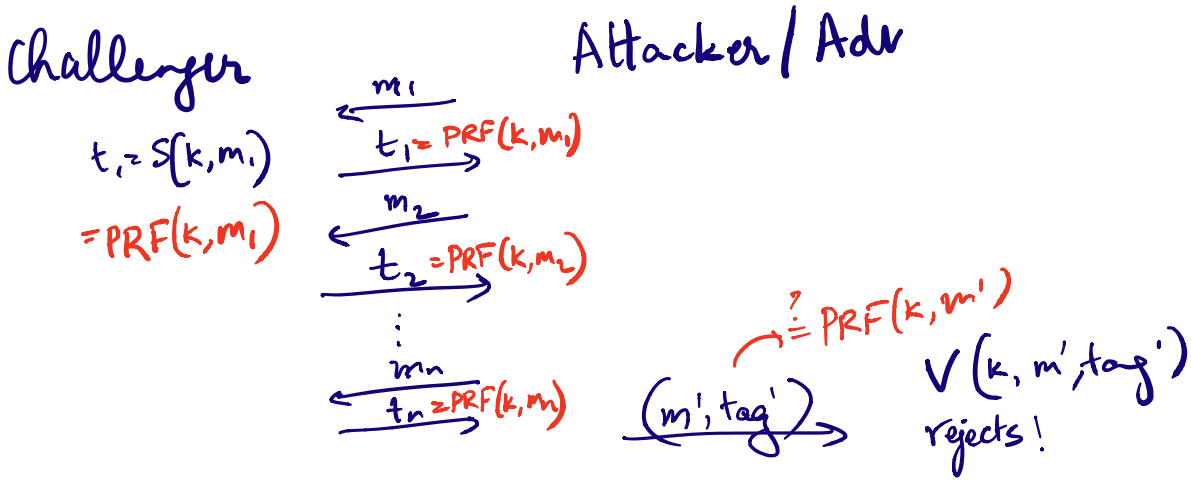
Shared key setting: security of MACs

Security, more generally:

- Attacker can demand tags (t_1, t_2, …. t_n) for messages (m_1, m_2, … m_n)
- Attacker wins if it outputs (m', tag') not in {(m_1, t_1), (m_2, t_2), …. (m_n, t_n)} such that V (k, m', tag') = 1

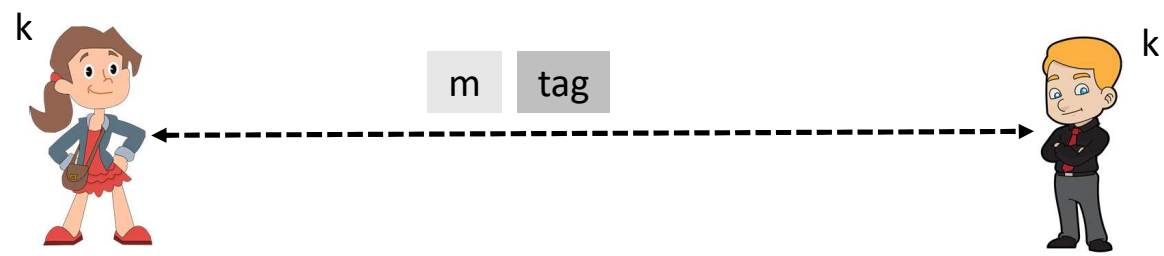
Use PRF with large enough output size.

$k \leftarrow K$



How do you build a secure MAC?

- Say *k* is a PRF key



= PRF(k, m)

V(k, m, tag)
= acc if PRF(k, m) = tag
= rej otherwise.

How do you build a secure MAC?

- Say *k* is a PRF key
- S (k, m) = PRF (k, m)
- V (k, m, tag) = 1 *if and only if* tag = PRF (k, m)

- Correctness?

How do you build a secure MAC?

- S (k, m) = PRF (k, m)
- V (k, m, tag) = 1 *if and only if* tag = PRF (k, m)

- Security?

$PRF : K \times M \rightarrow \{0,1\}^{\alpha}$

By PRF security,

$\forall m' \notin \{m_1, \ldots, m_n\}$,

$PRF(k, m') \approx$ uniform random string r of size α bits.

$\varepsilon_{MAC} \geq Pr[\text{Adv outputs } (m', tag') \text{ s.t. } tag' = PRF(k, m')]$ (P_1)

$= (P_1 - P_2) + P_2$

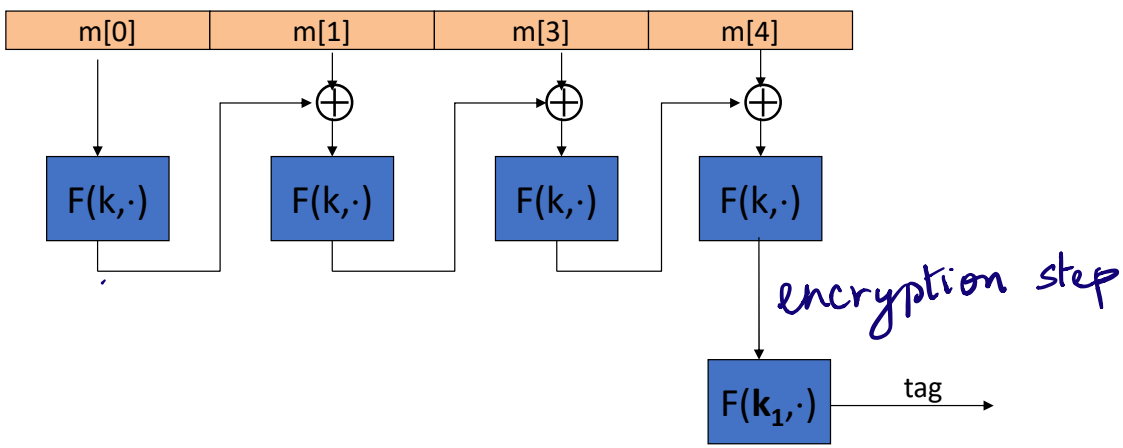
$= \varepsilon_{PRF} + Pr[\text{Adv outputs } (m', tag') \text{ s.t. } tag' = r]$ (P_2, $\hookrightarrow \frac{1}{2^{\alpha}}$)

$= \varepsilon_{PRF} + 2^{-\alpha}$.

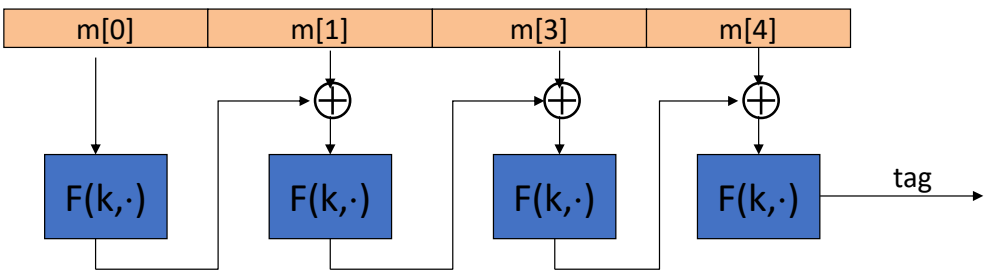
Examples

- AES: MAC for ~~64~~ 128-bit messages
- What about larger messages?
- For larger messages, we use:
  - CBC-MAC
  - HMAC

Encrypted CBC-MAC

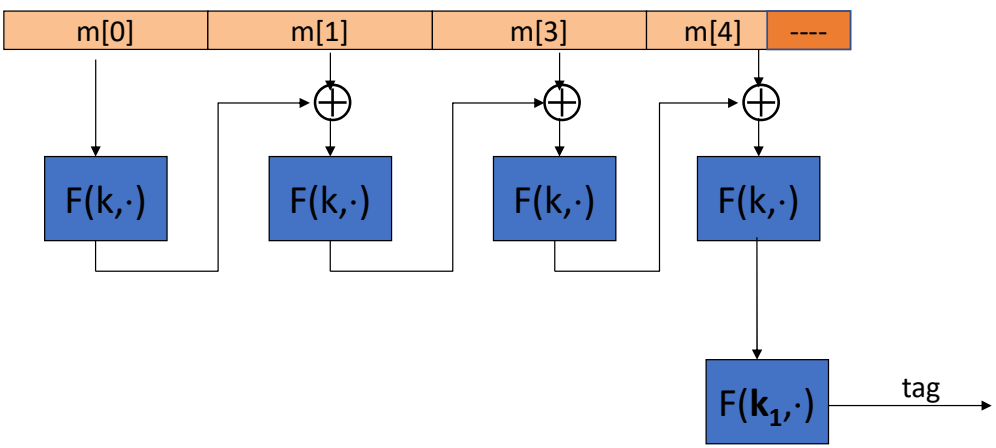


What happens if we remove encryption step?

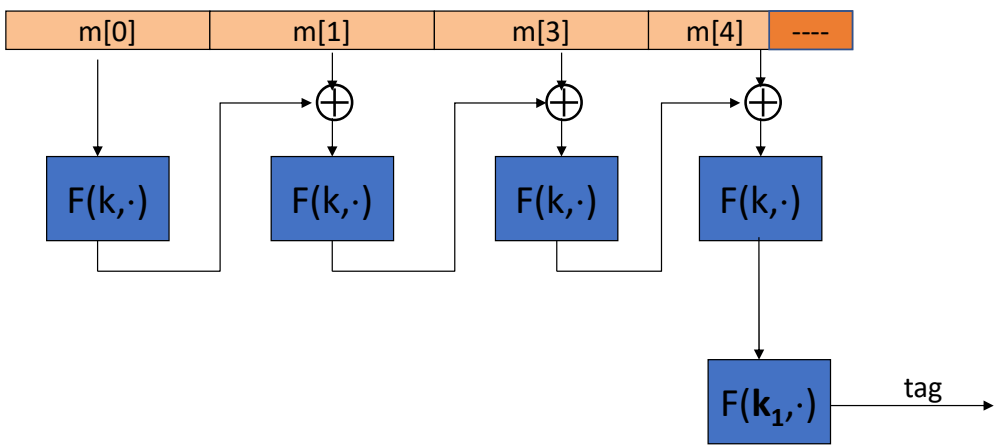


Obtain tag for m, then get t = F (k, m)
Output forgery (m', t') where m' = (m||(t \xor m)) and t' = t.

What if message length is not a multiple of block size?



First idea?



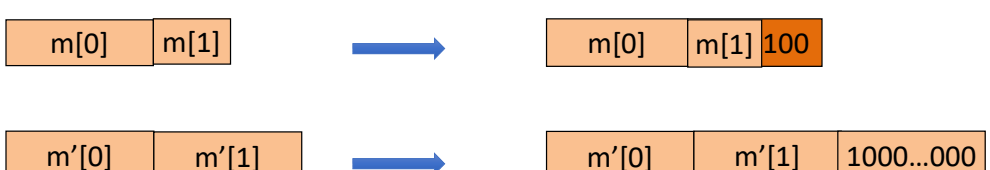
Need invertible padding!

For security, padding must be invertible !

$$m_0 \neq m_1 \quad \Rightarrow \quad \text{pad}(m_0) \neq \text{pad}(m_1)$$

Pad with "1000…00". Add new dummy block if needed.

- The "1" indicates beginning of pad.



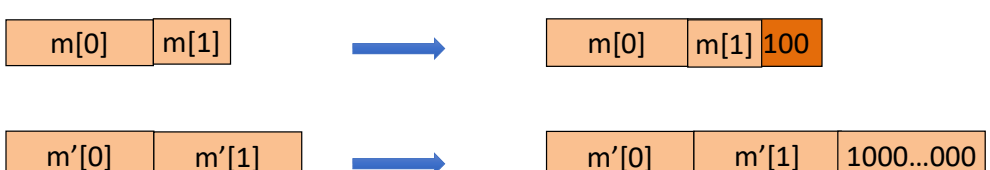
Need invertible padding!

For security, padding must be invertible !

$$m_0 \neq m_1 \quad \Rightarrow \quad \text{pad}(m_0) \neq \text{pad}(m_1)$$

Pad with "1000…00". Add new dummy block if needed.

- The "1" indicates beginning of pad.



Summary

- Modes of operation of block-ciphers (contd.), and
- Message Authentication Codes (MACs)