# MIT 6.875

# Foundations of Cryptography
# Lecture 5

# TODAY

**More Applications of PRFs**:

a. Identification Protocols

b. Applications to Learning Theory

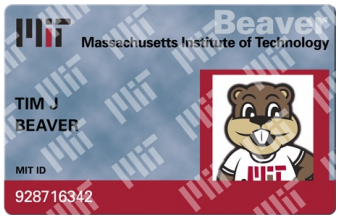
c. Authentication (EUF-CMA Security)

d. IND-CCA Security

**Logistics:**

- Problem Set 1 is due today at 11:59:59pm.
- Remember that you have 10 late days for this class, and you may use up to 5 for any one problem set.

# Friend-or-Foe Identification





- **Adversary**: person-in-the-middle.
- Can listen to / modify the communications. Wants to impersonate Tim.

# A Simple Lemma about Unpredictability

Let $f_s: \{0,1\}^\ell \to \{0,1\}^m$ be a pseudorandom function.

- Consider an adversary who requests and obtains $f_s(x_1), \ldots, f_s(x_q)$ for a polynomial $q = q(n)$.
- Can she predict $f_s(x^*)$ for some $x^*$ of her choosing where $x^* \notin \{x_1, \ldots, x_q\}$? How well can she do it?

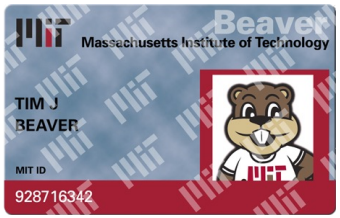
**Lemma**: If she succeeds with probability $\frac{1}{2^m} + 1/\mathrm{poly}(n)$, then she breaks PRF security. This is negligible in $n$ if $m$ is large enough, i.e. $\omega(\log n)$.

# A Simple Lemma about Unpredictability

Let $f_s: \{0,1\}^\ell \to \{0,1\}^m$ be a pseudorandom function.

- Consider an adversary who requests and obtains $f_s(x_1), \ldots, f_s(x_q)$ for a polynomial $q = q(n)$.
- Can she predict $f_s(x^*)$ for some $x^*$ of her choosing where $x^* \notin \{x_1, \ldots, x_q\}$? How well can she do it?
- Unpredictability ≡ Indistinguishability *for bits* (lecture 3)
- Indistinguishability ⟹ Unpredictability (*but not vice versa*).

# Challenge-Response Protocol




PRF Key $s$

Random $r$

$(ID, f_s(r))$


(ID number $ID$, PRF Key $s$)

"Proof": Adversary collects $(r_i, f_s(r_i))$ for poly many $r_i$ (potentially of her choosing). She eventually has to produce $f_s(r^*)$ for a fresh random $r^*$ when she is trying to impersonate.

This is hard as long as the input and output lengths of the PRF are long enough, i.e. $\omega(\log n)$.

# TODAY

**More Applications of PRFs**:

a. Identification Protocols

b. Applications to Learning Theory

c. Authentication (EUF-CMA Security)

d. IND-CCA Security

# Negative Results in Learning Theory

**Theorem [Kearns and Valiant 1994]**:
Assuming PRFs exist, there are hypothesis classes that cannot be learned by polynomial-time algorithms.

# Machine Learning and Cryptography (A quick aside)

Planting Undetectable Backdoors in Machine Learning Models


Shafi Goldwasser, UC Berkeley
Michael P. Kim, UC Berkeley
Vinod Vaikuntanathan, MIT
Or Zamir, IAS


**On the Cryptographic Hardness of Learning Single Periodic Neurons**


**Min Jae Song***
Courant Institute
New York University
minjae.song@nyu.edu

**Ilias Zadik***
Department of Mathematics
Massachusetts Institute of Technology
izadik@mit.edu

**Joan Bruna**
Courant Institute
Center for Data Science
New York University
bruna@cims.nyu.edu


Continuous LWE is as Hard as LWE
& Applications to Learning Gaussian Mixtures


Aparna Gupte*
MIT
agupte@mit.edu

Neekon Vafa†
MIT
nvafa@mit.edu

Vinod Vaikuntanathan‡
MIT
vinodv@mit.edu

# TODAY

**More Applications of PRFs**:

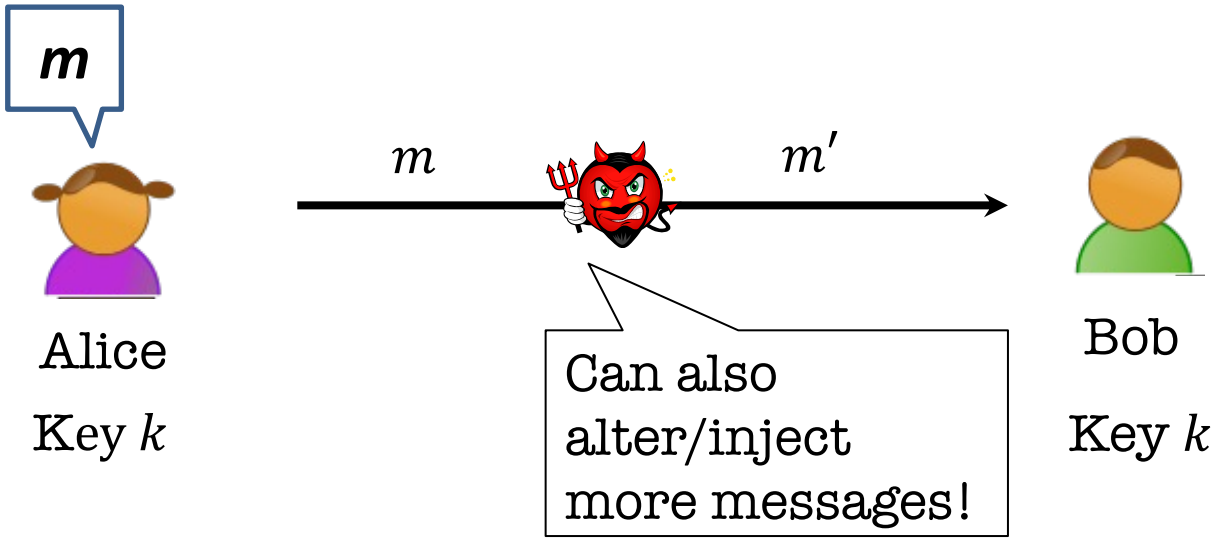
a. Identification Protocols

b. Applications to Learning Theory

c. Authentication (EUF-CMA Security)

d. IND-CCA Security

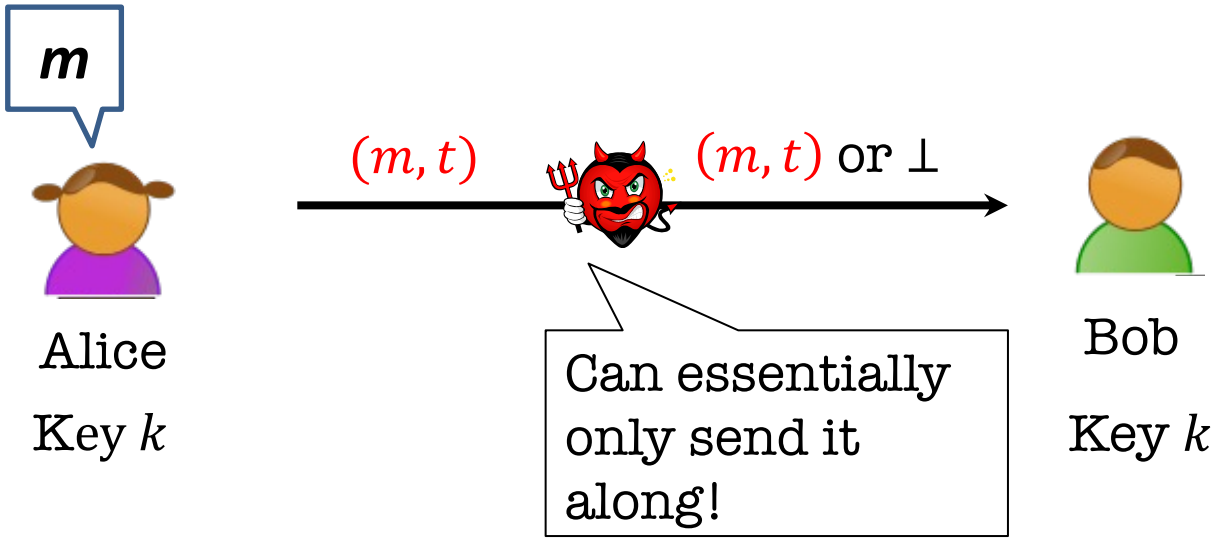
# The authentication problem




This is known as a **man-in-the-middle attack.**

How can Bob check if the message is indeed from Alice?

# The authentication problem




We want Alice to generate a tag for the message *m* which is **hard to generate** without the secret key *k*.

# Message Authentication Codes (MACs)

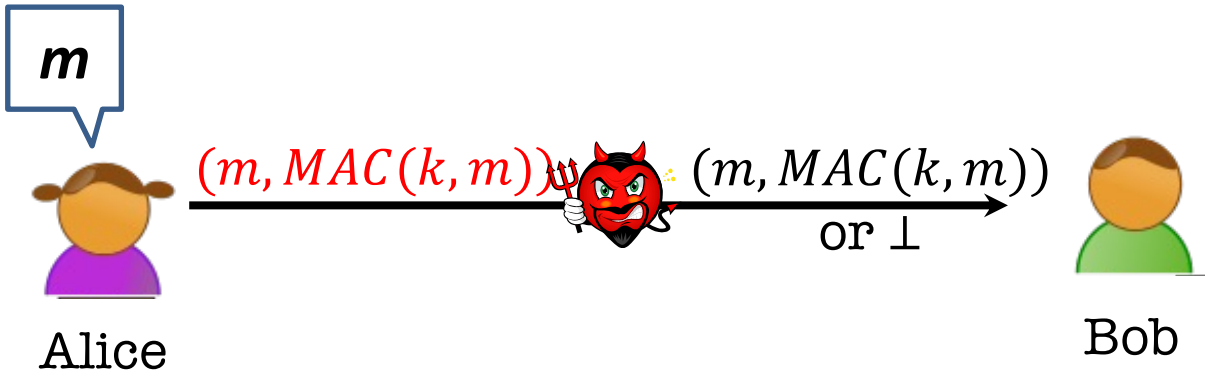
A triple of algorithms (Gen, MAC, Ver):

- Gen$(1^n)$: Produces a key $k \leftarrow K$.
- MAC$(k, m)$: Outputs a tag $t$ (may be deterministic).
- Ver$(k, m, t)$: Outputs Accept or Reject.

**Correctness**: Pr[Ver$\big(k, m, MAC(k, m)\big) =$ Accept] = 1

**Security:** *Hard to forge.* Intuitively, it should be hard to come up with a new pair *(m', t')* such that Ver accepts.

# What is the power of the adversary?




- Can see many pairs $\big(m, MAC(k,m)\big)$.
- Can access a MAC oracle $MAC(k, \cdot)$
  - Obtain tags for message of choice.

This is called a *chosen message attack (CMA).*

# Defining MAC Security

- **Total break:** The adversary should not be able to recover the key *k*.
- **Universal break:** The adversary can generate a valid tag for every message.
- **Existential break:** The adversary can generate a new valid tag *t* for some message *m*.

We will require MACs to be secure against the existential break!!

# EUF-CMA Security

**E**xistentially **U**n**f**orgeable against **C**hosen **M**essage **A**ttacks

$k \leftarrow K$

$m_1$

$t_1 = MAC(k, m_1)$

$m_2$

$t_2 = MAC(k, m_2)$

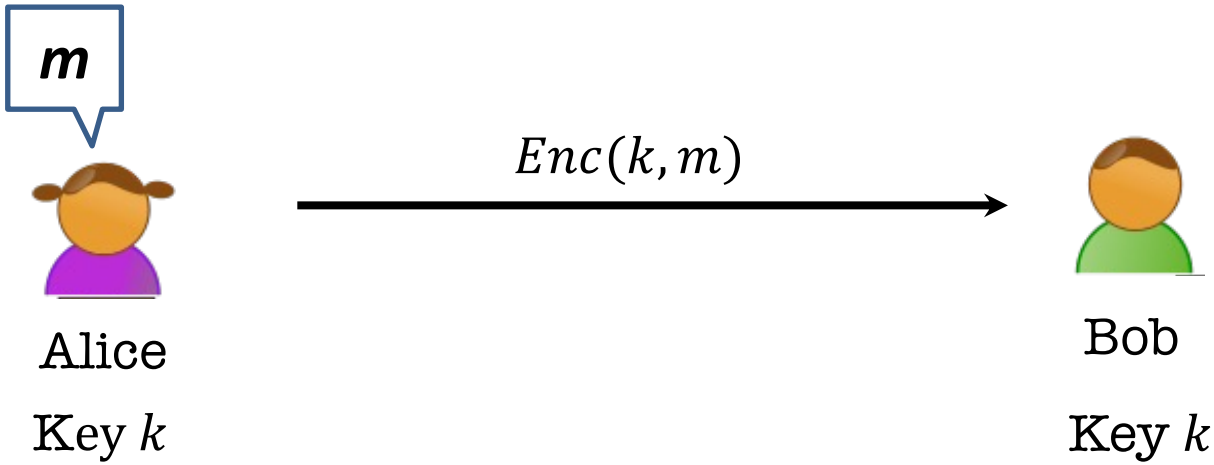
$\vdots$

$(m, t)$

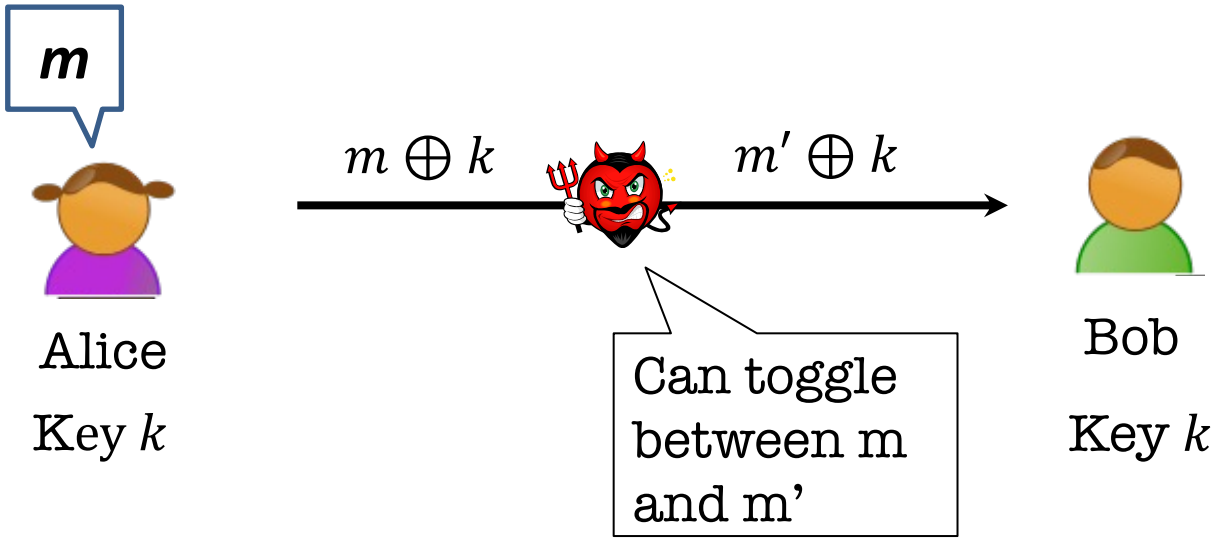
Accept if $(m, t) \neq (m_i, t_i)$ for all $i$, and $Ver(k, m, t) = 1$.

**Want:** $\Pr((m,t) \leftarrow A^{MAC(k,\cdot)}(1^n), Ver(k,m,t) = 1, (m,t) \notin Q)) = negl(n)$.
where $Q$ is the set of queries $\{(m_i, t_i)\}_i$ that $A$ makes.

# Wait… Does encryption not solve this?

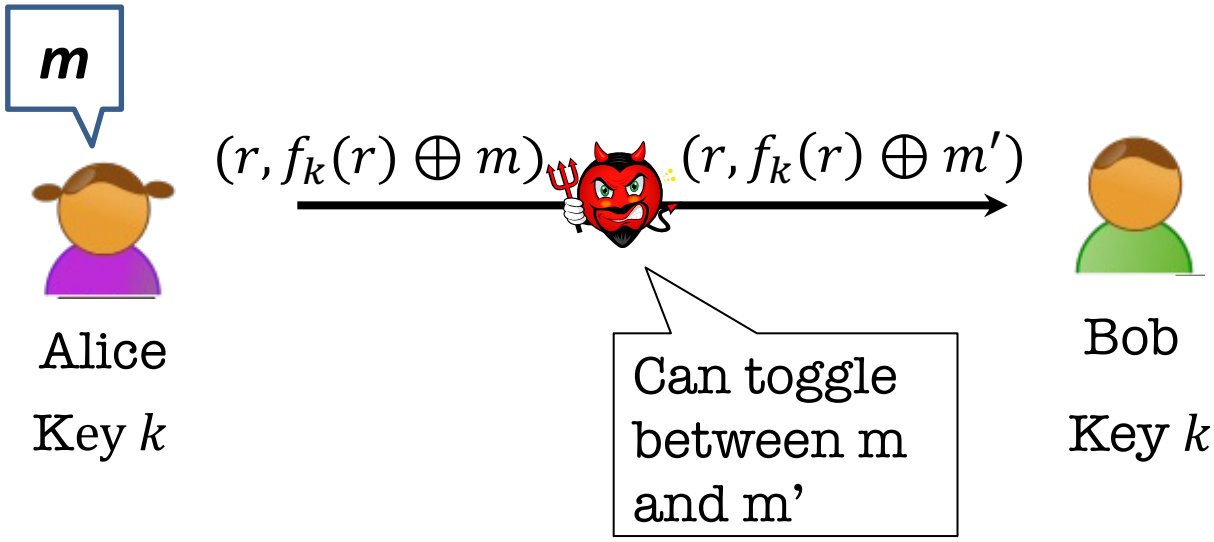


# Wait… Does encryption not solve this?




One-time pad (and encryption schemes in general) are ***malleable***.

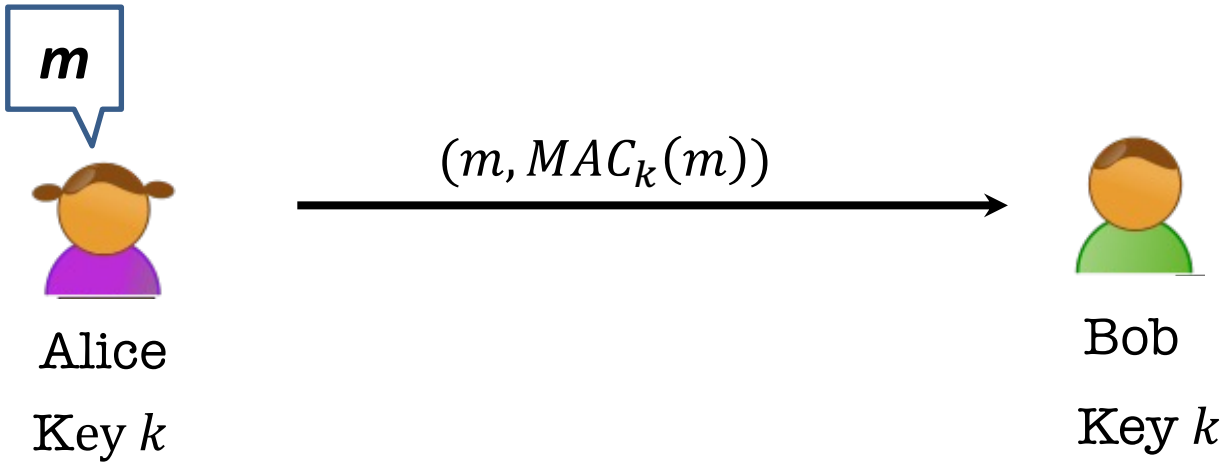
# Wait… Does encryption not solve this?




One-time pad (and encryption schemes in general) are ***malleable***.

Privacy and Integrity are very **different goals!**

# Constructing a MAC




Gen$(1^n)$: Produces a PRF key $k \leftarrow K$.
MAC$(k, m)$: Output $f_k(m)$.
Ver$(k, m, t)$: Accept if $f_k(m) = t$, reject otherwise.

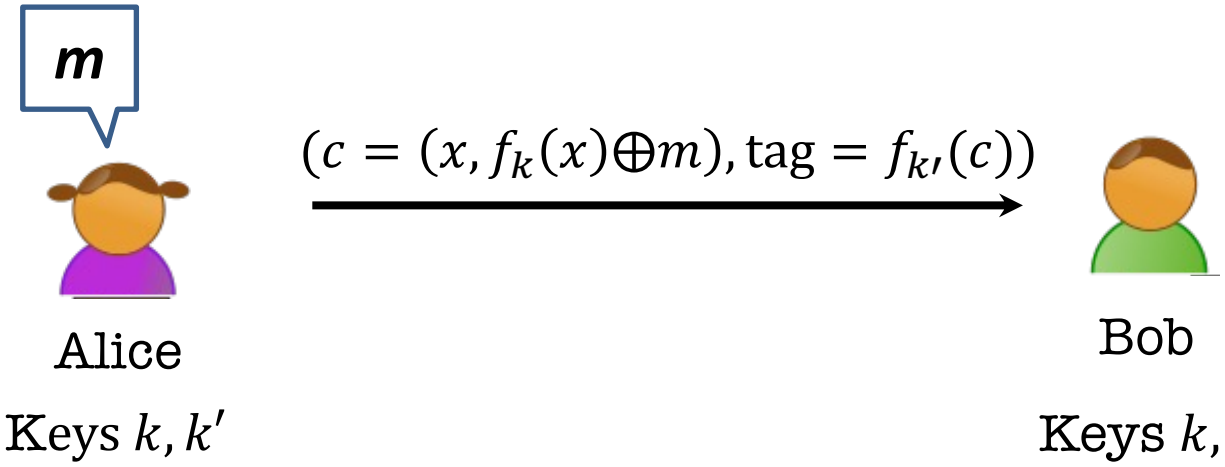
**Security:** Our earlier unpredictability lemma about PRFs essentially proves that this is secure!

# Dealing with Replay Attacks

- The adversary could send an old valid *(m, tag)* at a later time.
  - In fact, our definition of security does not rule this out.

- **In practice:**
  - Append a time-stamp to the message. Eg. (m, T, MAC(m, T)) where T = 21 Sep 2022, 1:47pm.
  - Sequence numbers appended to the message (this requires the MAC algorithm to be *stateful*).

# Privacy and Integrity!



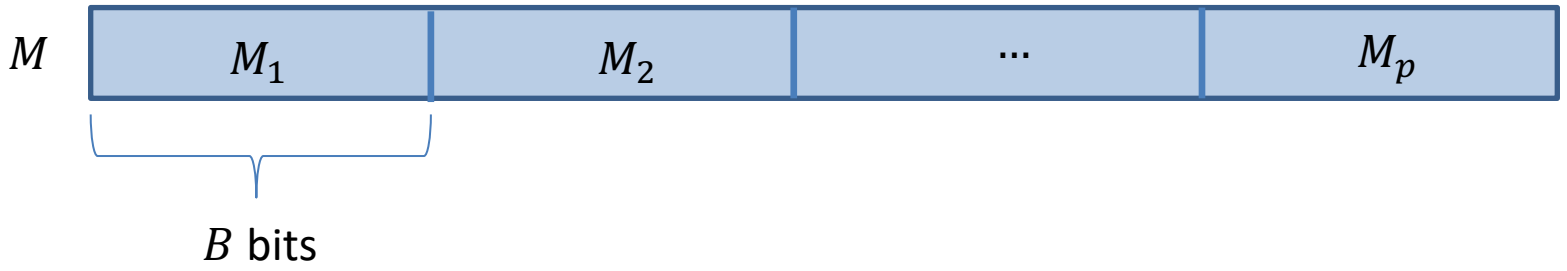

MACs give us integrity, but not necessarily privacy (why?)

**Solution**: Encrypt, then MAC (More in Problem Set 2)

# MACs for Long Messages

Suppose we have PRF Family $f_k$: $\{0,1\}^B \rightarrow \{0,1\}^m$.

How do we MAC long messages? (Eg. A document)



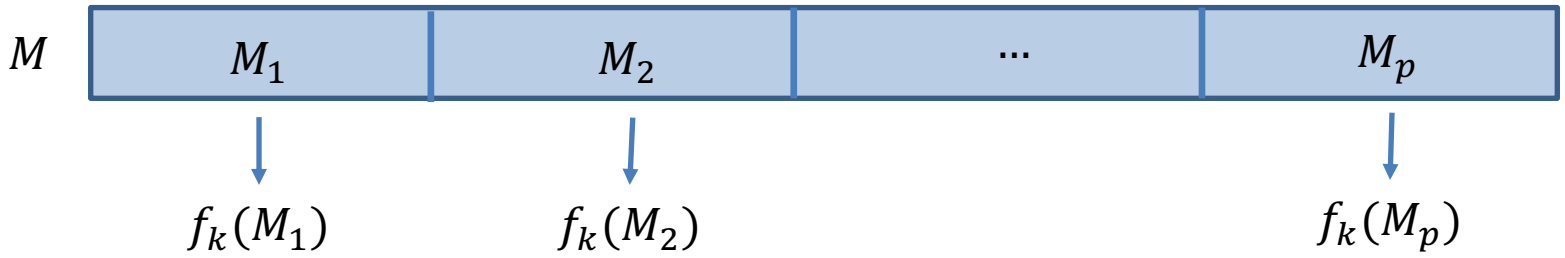

**Take 1:** $MAC\left(k, M_1, M_2, \ldots, M_p\right) = f_k(\bigoplus_i M_i)$.

**Issue:** Can come up with MAC for anything with the same XOR.

# MACs for Long Messages

Suppose we have PRF Family $f_k$: $\{0,1\}^B \rightarrow \{0,1\}^m$.

How do we MAC long messages?



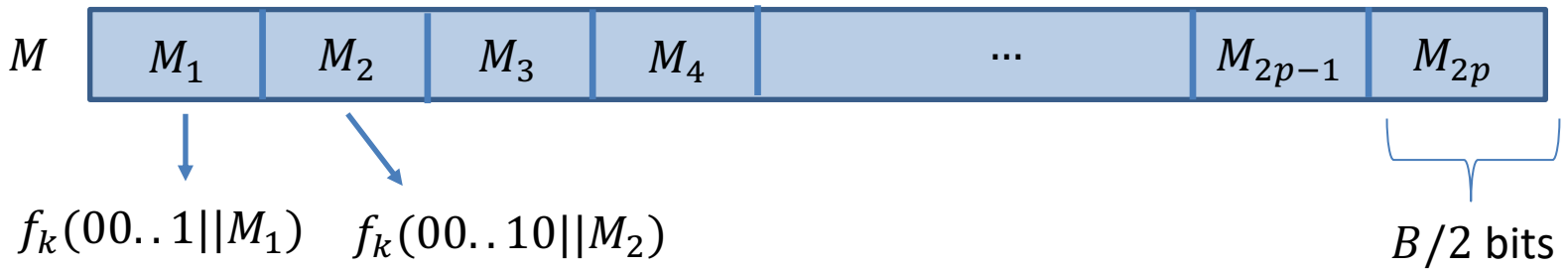

**Take 2:** $MAC\left(k, M_1, M_2, \ldots, M_p\right) = \bigoplus_i f_k(M_i)$.

**Issue:** Can permute the messages and MAC it!

# MACs for Long Messages

Suppose we have PRF Family $f_k$: $\{0,1\}^B \rightarrow \{0,1\}^m$.

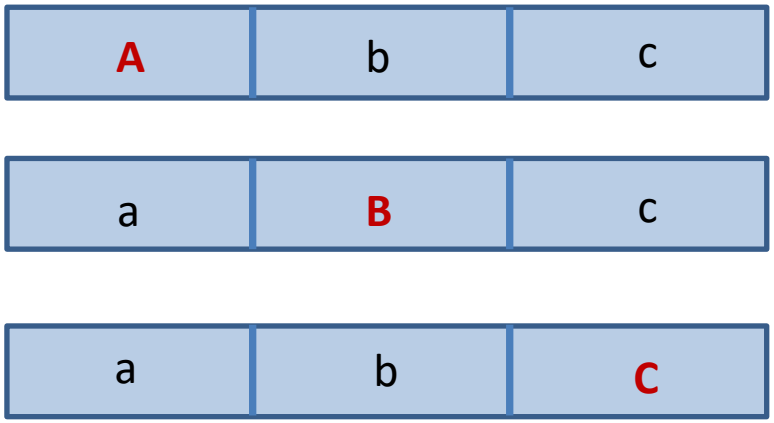
How do we MAC long messages?




**Take 3:** $MAC\left(k, M_1, M_2, \ldots, M_{2p}\right) = \bigoplus_i f_k(\langle i \rangle || M_i)$, where $\langle i \rangle$ is the $B/2$-bit representation of $i$.

**Issue:** Cut-and-paste attack.

# MACs for Long Messages




$f_k(1||A) \oplus f(2||b) \oplus f(3||c)$

$f_k(1||a) \oplus f(2||B) \oplus f(3||c)$

$f_k(1||a) \oplus f(2||b) \oplus f(3||C)$

$$f_k(1||A) \oplus f_k(2||B) \oplus f_k(3||C) = MAC_k(A||B||C)$$

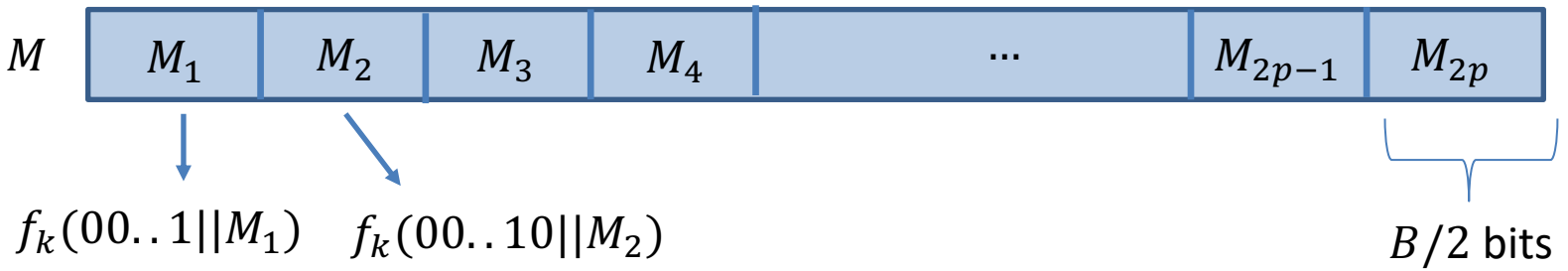
**Take 3:** $MAC(k, M_1, M_2, \ldots, M_{2p}) = \bigoplus_i f_k(\langle i \rangle || M_i)$, where $\langle i \rangle$ is the $B/2$-bit representation of $i$.

**Issue:** Cut-and-paste attack.

# MACs for Long Messages

Suppose we have PRF Family $f_k$: $\{0,1\}^B \rightarrow \{0,1\}^m$.

How do we MAC long messages?




**Randomised construction** by Bellare, Guerin, Rogaway:

$$MAC_k(M_1, M_2, \ldots, M_{2p}; r) = (r, f_k(r) \oplus (\oplus_i f_k(\langle i \rangle || M_i)))$$

**Proof:** Exercise ☺ (Similar to secret-key proof)

# Hash-then-Sign

- Let $H: \{0,1\}^* \rightarrow \{0,1\}^B$ be a *collision resistant hash function (CRHF).*
  - Public function which compresses long messages to B bits.
  - Hard to find $x, x'$ such that $H(x) = H(x')$.
- $MAC_k(m) = f_k\big(H(m)\big)$.
- **Exercise:** Show that this is a EUF-CMA secure MAC!

# TODAY

**More Applications of PRFs**:

a. Identification Protocols

b. Applications to Learning Theory

c. Authentication (EUF-CMA Security)

d. IND-CCA Security

# Recall IND-CPA Security

- Indistinguishable against chosen-plaintext attack.
  - i.e. Adversary has access to Enc oracle.
  - Exercise: This is equivalent to definition from Lec 4.

$poly(n)$ times

$m$

$Enc(k, m)$

$k \leftarrow K$

$(m_0^*, m_1^*)$

$b \leftarrow \{0,1\}$

$Enc(k, m_b)$

$poly(n)$ times

$m$

$Enc(k, m)$

$b'$

Accept if $b' = b$

# IND-CCA2 Security

- Indistinguishable against chosen-ciphertext attack.
  - i.e. Adversary has access to Enc and Dec oracle.

$poly(n)$ times

$m, c$

$Enc(k, m), Dec(k, c)$

$k \leftarrow K$

$(m_0^*, m_1^*)$

$b \leftarrow \{0,1\}$

$c^* = Enc(k, m_b)$

$poly(n)$ times

$m, c$

$Enc(k, m), Dec(k, c)$

Check $c \neq c^*$

$b'$

Accept if $b' = b$

# Our SKE is not IND-CCA2 Secure

- Given a decryption oracle, $Enc(k, m; r) = (r, f_k(r) \oplus m)$ is not secure!

$k \leftarrow K$

$b \leftarrow \{0,1\}$

$(m_0^*, m_1^*)$ →

← $c^* = (r, f(r) \oplus m_b^*)$

Choose some $s \neq 0$ $\quad c = (r, f(r) \oplus m_b^* \oplus s)$ → $\quad$ Check $c \neq c^*$

Recover $m_b^*$ $\quad m_b^* \oplus s$ ←

$b$ →

Accept if $b' = b$

**If only it were hard to create a valid ciphertext to decrypt….**

# IND-CCA2 Secure SKE

Let (Gen, Enc, Dec) be IND-CPA Secure.

A triple of algorithms (Gen, Enc', Dec'):

- $\text{Gen}(1^n)$: Produces a secret key $k \leftarrow K_{\text{sk}}$ and MAC key $k' \leftarrow K_{mac}$ .
- $Enc'(k, k', m)$: Outputs $c = Enc(k, m)$ along with tag $t = MAC(k', c)$.
- $Dec'(k, k', (c, t))$: If $Ver(k', c, t) =$ Reject, then output $\perp$. Otherwise, $Dec(k, c)$.

**Intuition:** The decryption oracle is useless because it is difficult for valid tags $t$.