COS433/Math 473: Cryptography

Mark Zhandry

Princeton University

Spring 2017

Midterm Details

Available Monday 3pm

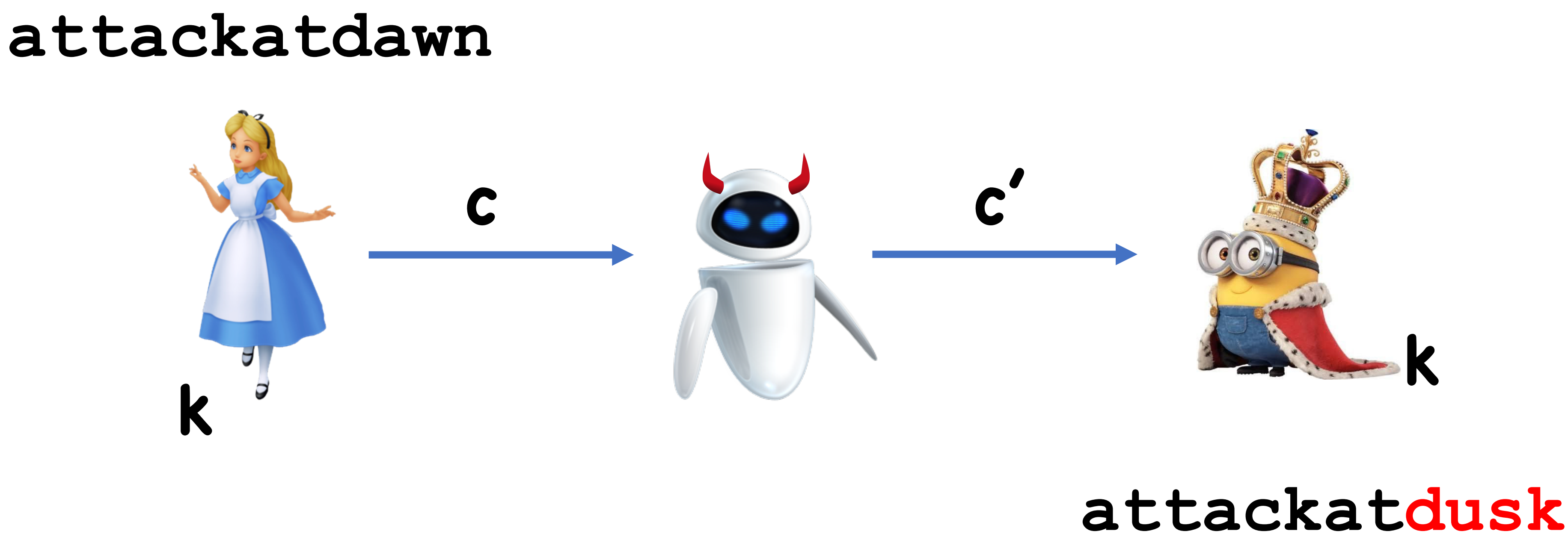
Due Wednesday 1pm
- Submitted via blackboard like the homeworks

Midterms are to be completely individually

Topics: through today's lecture

Please don't discuss midterms until 1pm Friday March 17

Malleability



Malleability

Some encryption schemes of operation are *malleable*

- Can modify ciphertext to cause predictable changes to plaintext

Examples: basically everything we’ve seen so far

- Stream ciphers
- CTR
- CBC
- ECB
- …

Message Integrity

We cannot stop adversary from changing the message in route to Bob

However, we can hope to have Bob perform some check on the message he receives to ensure it was sent by Alice

- If check fails, Bob rejects the message

For now, we won't care about message secrecy

- We will add it back in later

Message Authentication Codes

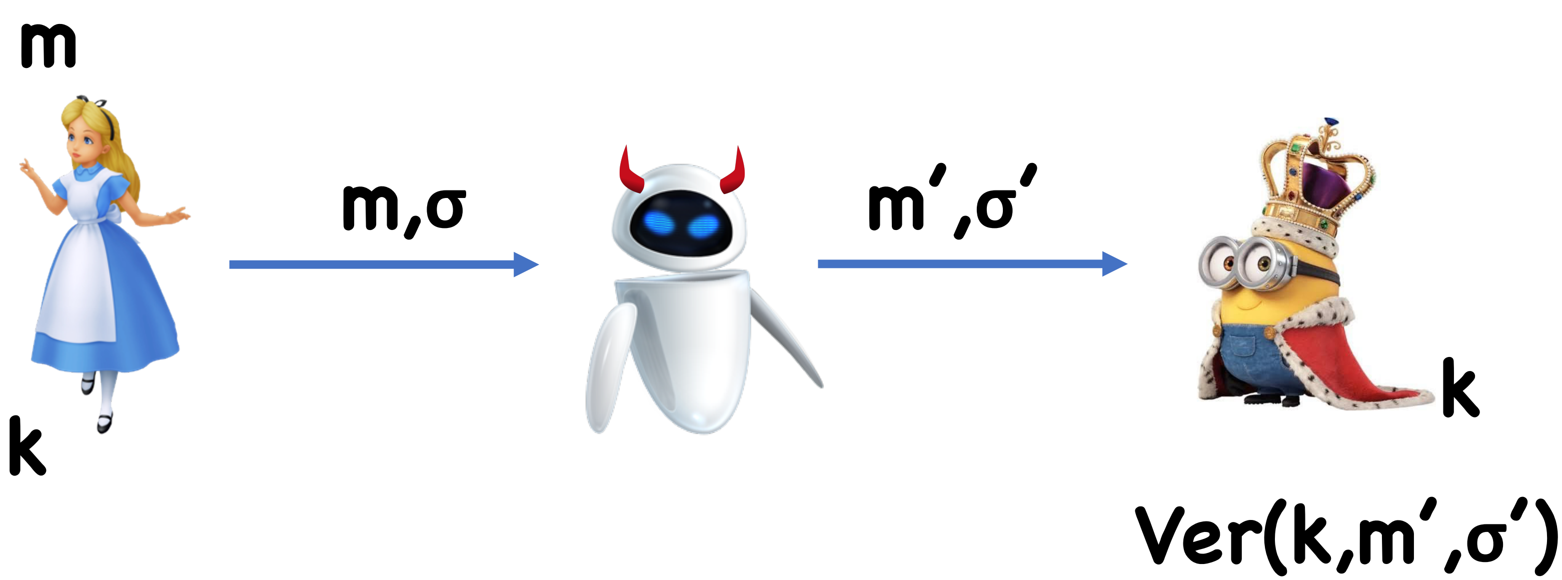
Syntax:

- Key space $\mathbf{K}_\lambda$
- Message space **M**
- Tag space $\mathbf{T}_\lambda$
- **MAC(k,m) → σ**
- **Ver(k,m,σ) → 0/1**

Correctness:

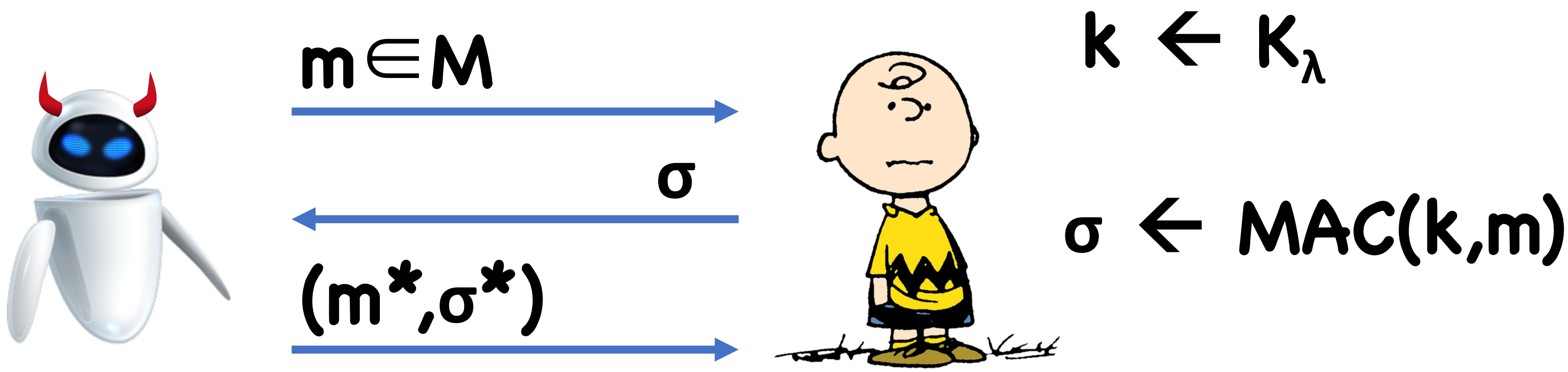
- $\forall$**m,k, Ver(k,m, MAC(k,m)) = 1**

Message Authentication Codes



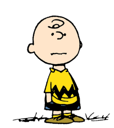
Goal: If Eve changed **m**, Bob should reject

1-time Security For MACs



Output 1 iff:

- $m^* \neq m$
- $Ver(k,m^*,\sigma^*) = 1$

1CMA-Adv( , λ) = Pr[ outputs 1]

Definition: (MAC,Ver) is 1-time statistically secure under a chosen message attack (**1CMA-secure)** if, for all , there exists a negligible ε such that

$$\textbf{1CMA-Adv}(\text{ adversary }, \lambda) \leq \varepsilon(\lambda)$$

Impossibility of Perfect Security?

A Simple 1-time MAC

Suppose $\mathbf{H}_\lambda$ is a family of *pairwise independent* functions from $\mathbf{M}$ to $\mathbf{T}_\lambda$

For any $m_0 \neq m_1 \in M,\ \sigma_0, \sigma_1 \in T_\lambda$

$$\Pr_{h \leftarrow H_\lambda}[\ h(m_0)=\sigma_0 \wedge h(m_1)=\sigma_1] = 1/|T_\lambda|^2$$

$K = H_\lambda$

$MAC(h, m) = h(m)$

$Ver(h,m,\sigma) = (h(m) == \sigma)$

Theorem: (MAC,Ver) is 1-time secure, provided $\mathbf{T_\lambda}$ is large enough. In particular, for any ,

1CMA-Adv(**, λ) = 1/|$\mathbf{T_\lambda}$|**

So to have security, just need $|\mathbf{T_\lambda}|$ to be superpolynomial

- Ex: $\mathbf{T_\lambda = \{0,1\}^\lambda}$

Proof

Idea:

- For every two inputs, outputs are independent
- Therefore, knowing one input/output pair does not tell you anything about the output at any other input

Constructing Pairwise Independent Functions

$\mathbf{T} = \mathbb{F}$ (finite field of size $\approx 2^{\lambda}$)

- Example: $\mathbb{Z}_p$ **for some prime p**

Easy case: let $\mathbf{M}=\mathbb{F}$

- $\mathbf{H} = \{h(x) = a\,x + b:\ a,b \in \mathbb{F}\}$

Slightly harder case: Embed $\mathbf{M} \subseteq \mathbb{F}^n$

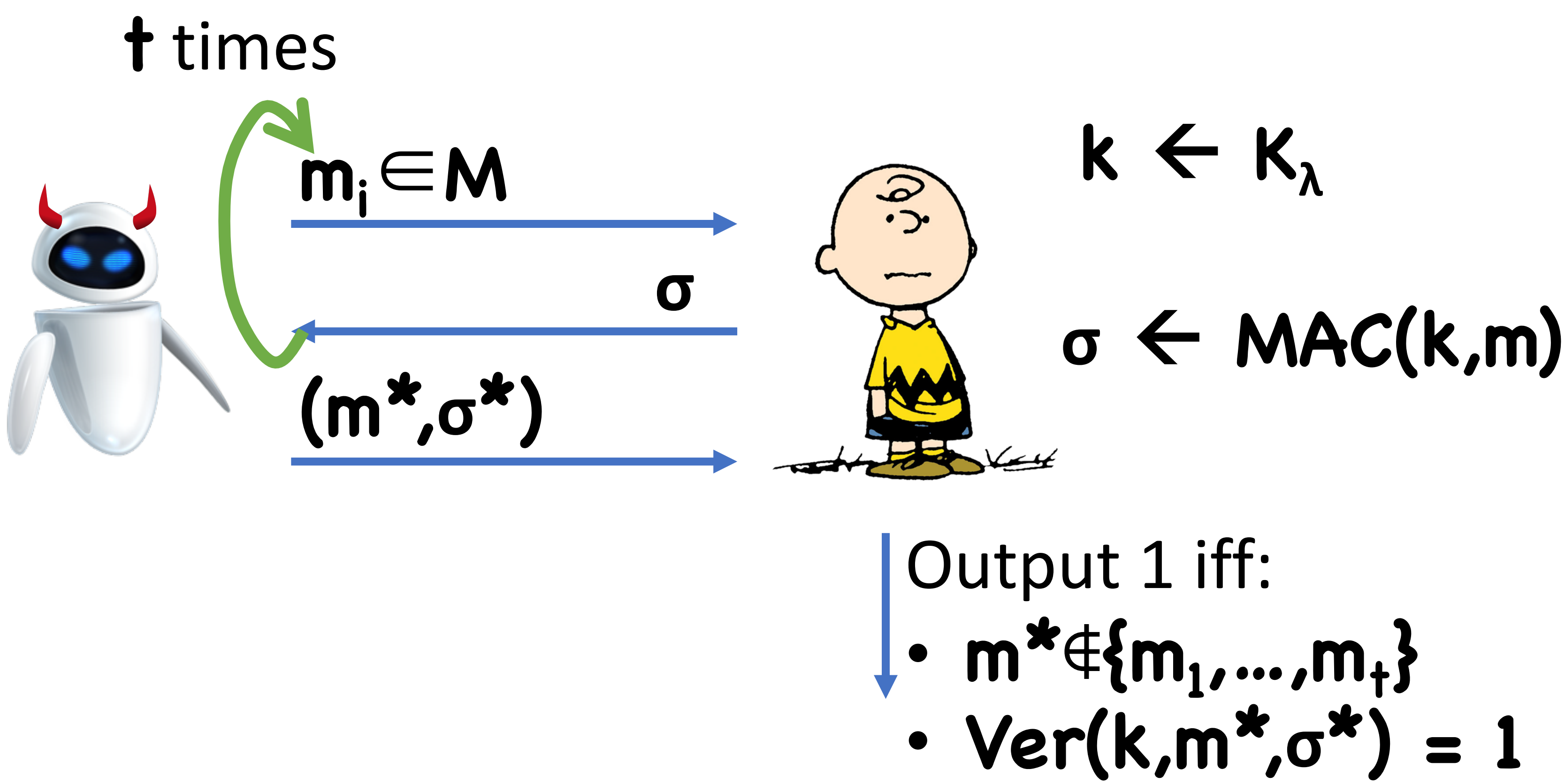
- $\mathbf{H} = \{h(x) = \langle a,x \rangle + b:\ a \in \mathbb{F}^n,\ b \in \mathbb{F}\}$

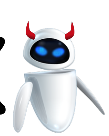
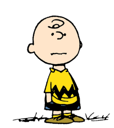
Multiple Use MACs?

Just like with OTP, if use 1-time twice, no security

Why?

t-Time MACs



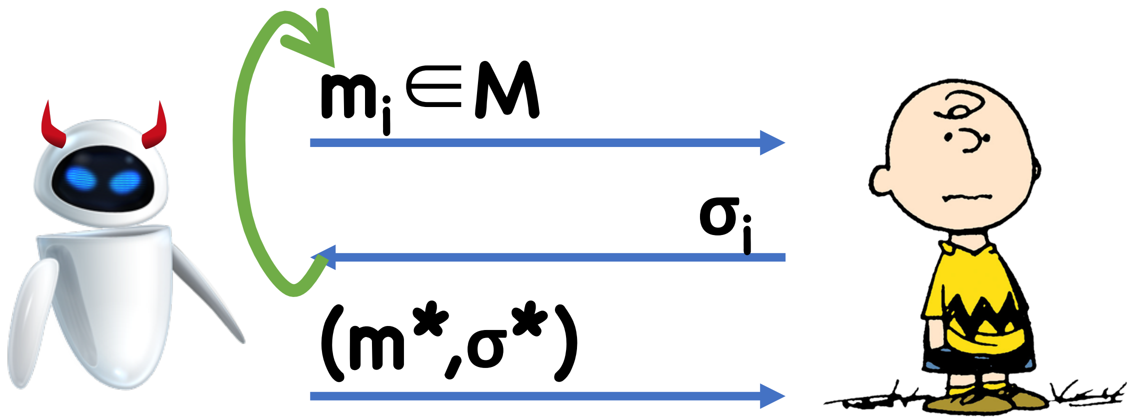
tCMA-Adv(, λ) = Pr[ outputs 1]

Constructing t-time MACs

Ideas?

Unbounded Use MACs

No restriction

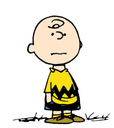


$k \leftarrow K_\lambda$

$\sigma \leftarrow MAC(k, m_i)$

Output 1 iff:

- $m^* \notin \{m_1, \ldots\}$
- $Ver(k, m^*, \sigma^*) = 1$

CMA-Adv( , λ) = Pr[ outputs 1]

Definition: (MAC,Ver) is statistically secure under a chosen message attack (**CMA-secure**) if, for all , there exists a negligible ε such that

CMA-Adv( **,** $\lambda) \leq \varepsilon(\lambda)$

Impossibility

Theorem: There are no MACs that are statistically CMA secure

Proof

Idea:

- By making $q \gg \log |K|$ queries, you *should* be able to uniquely determine key
- One key is determined, can forge any message

Problem:

- What if certain bits of the key are ignored
- Intuition: ignoring bits of key shouldn't help

Proof

Define $\mathbf{r_q}$ as follows:

- Challenger chooses random key $\mathbf{k}$
- Adversary repeatedly choose random (distinct) messages $\mathbf{m_i}$ in $\mathbf{M}$
- Query the CMA challenger on each $\mathbf{m_i}$, obtaining $\boldsymbol{\sigma_i}$
- Let $\mathbf{K'_q}$ be set of keys $\mathbf{k'}$ such that $\mathbf{MAC(k',m_i)=\sigma_i}$ for $\mathbf{i=1,\dots,q}$
- Let $\mathbf{r_q}$ be the expected size of $\mathbf{K'_q}$

Claim: If **(MAC,Ver)** is statistically CMA-secure, then $r_q \leq r_{q-1}/2$

If not, then with probability at least ¼,

$$|K'_q| > |K'_{q-1}|/4$$

Attack:

- Make $q-1$ queries on random messages m_i
- Choose key k from K'_{q-1}
- Choose random m_q, compute $\sigma_q = MAC(k, m_q)$
- Output (m_q, σ_q)

Probability of forgery?

Claim: If **(MAC,Ver)** is statistically CMA-secure, then $r_q \leq r_{q-1}/2$

Finishing the impossibility proof:

- r_q is always at least 1 (since there is a consistent key)
- $r_0 = |K|$
- $1 \leq r_q \leq r_0/2^q \leq |K|/2^q$
- Setting $q > \log |K|$ gives a contradiction

Definition: (MAC,Ver) is (computationally) secure under a chosen message attack (**CMA-secure**) if, for all PPT , there exists a negligible ε such that

CMA-Adv(, λ) ≤ ε(λ)

Constructing MACs

Use a PRF

$F: K \times M \rightarrow T$

$\mathrm{MAC}(k,m) = F(k,m)$
$\mathrm{Ver}(k,m,\sigma) = (F(k,m) == \sigma)$

Theorem: (MAC,Ver) is CMA secure assuming **1/|T|** is negligible

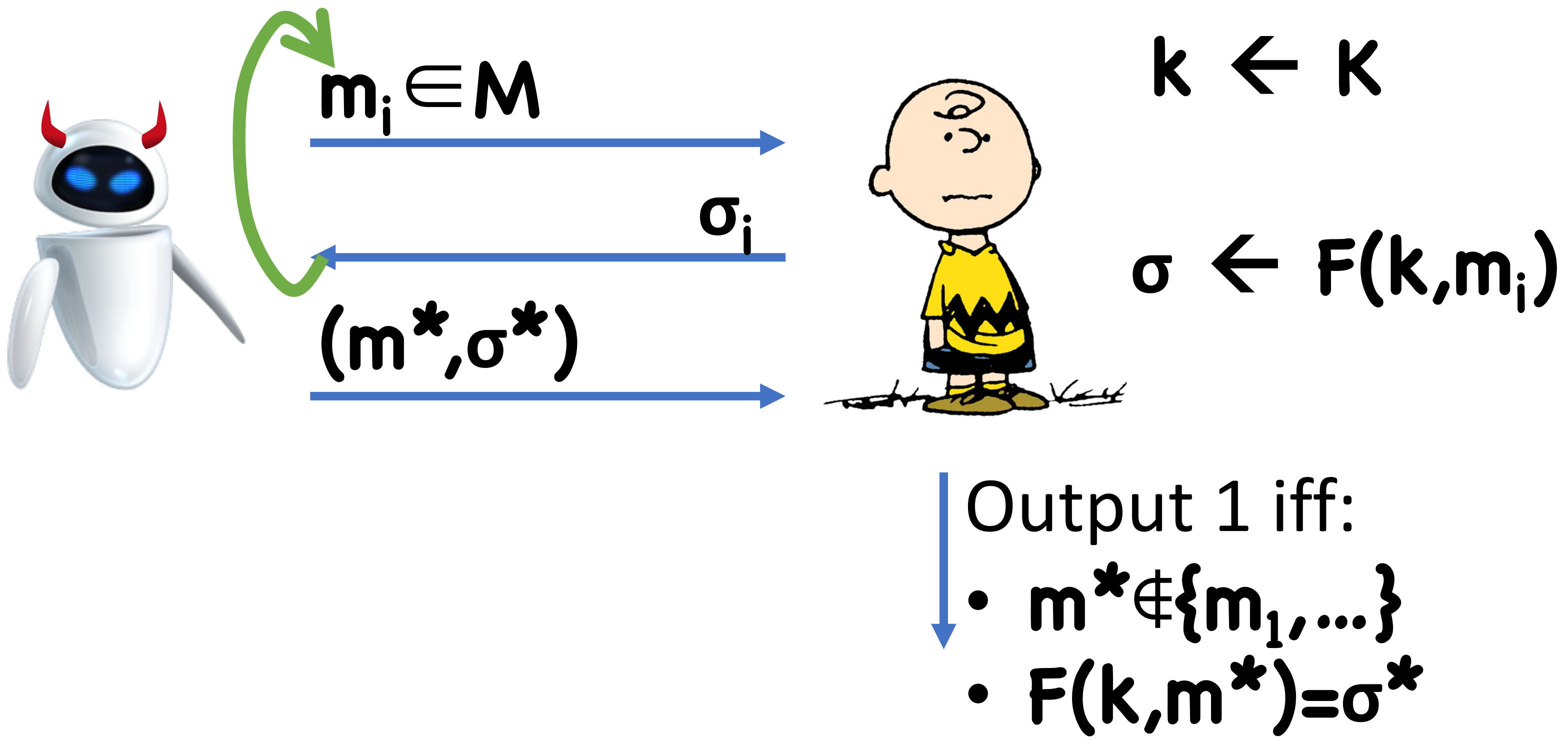
Security Proof

Assume toward contradiction PPT 

Hybrids!

Security Proof

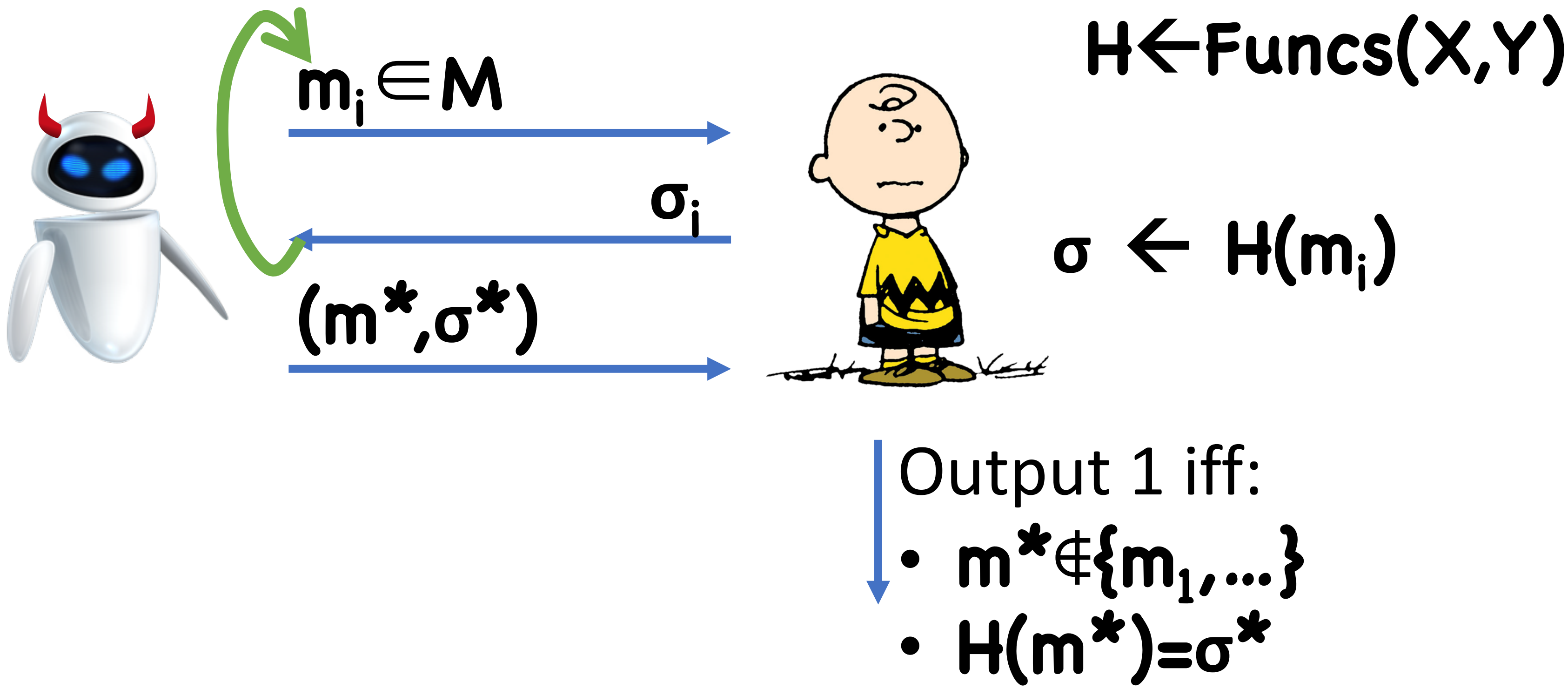
Hybrid 0



CMA Experiment

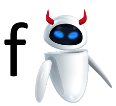
Security Proof

<u>Hybrid 1</u>



Security Proof

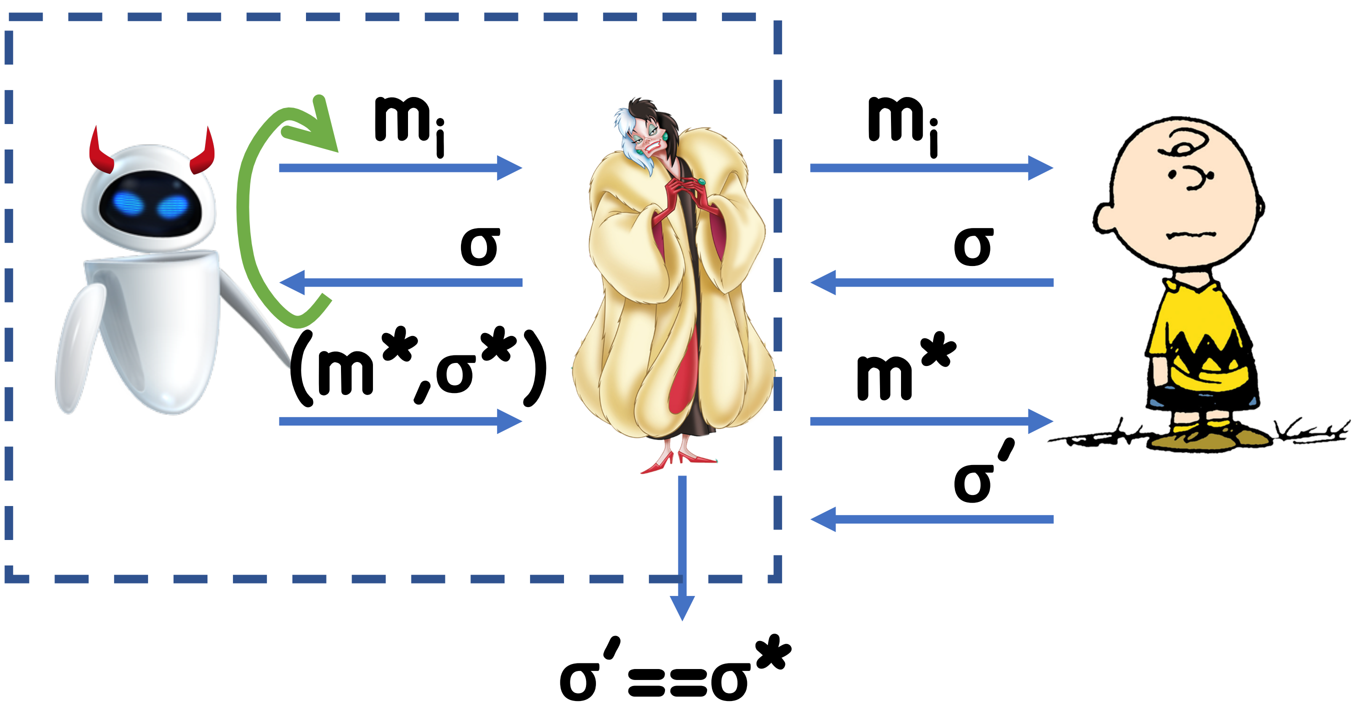
Claim: in Hybrid 1, output 1 with probability **1/|T|**

-  sees values of **H** on points $\mathbf{m_i}$
- Value on $\mathbf{m^*}$ independent of  ‘s view
- Therefore, probability $\boldsymbol{\sigma^*=H(m^*) = 1/|T|}$

Security Proof

Claim: $|\Pr[1 \leftarrow \text{Hyb1}] - \Pr[1 \leftarrow \text{Hyb2}]| < \text{negl}$

- Suppose not, construct PRF adversary 



Constructing MACs/PRFs

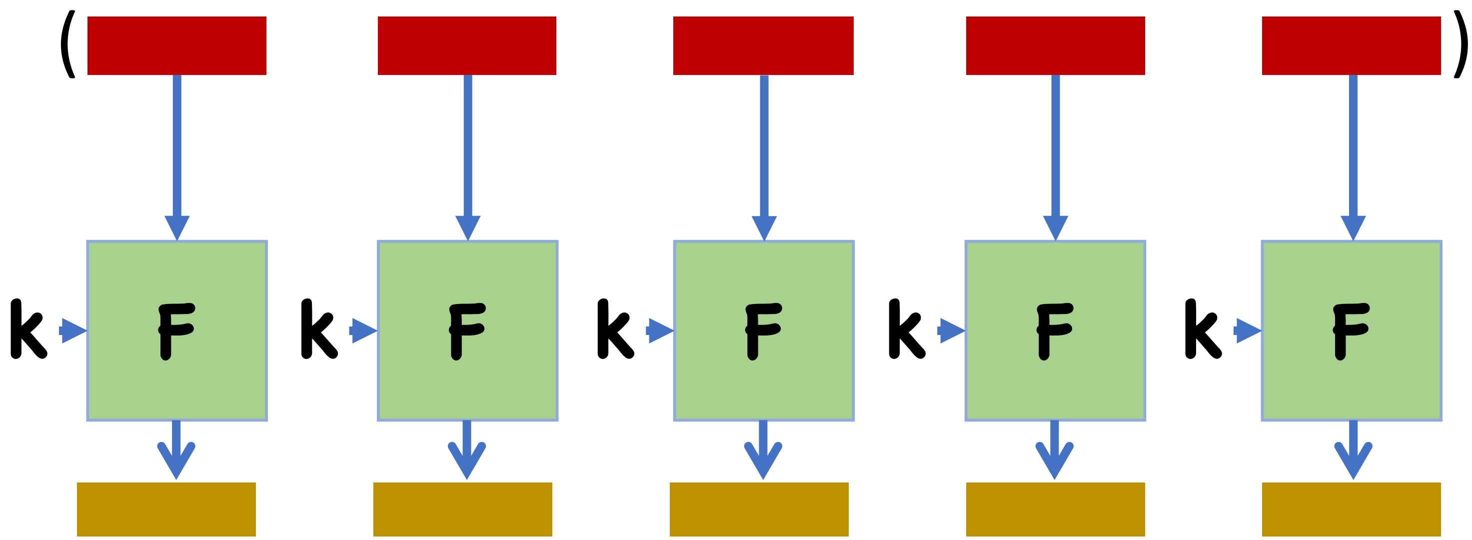
We saw that block ciphers are good PRFs

However, the input length is generally fixed

- For example, AES maximum block length is 128 bits

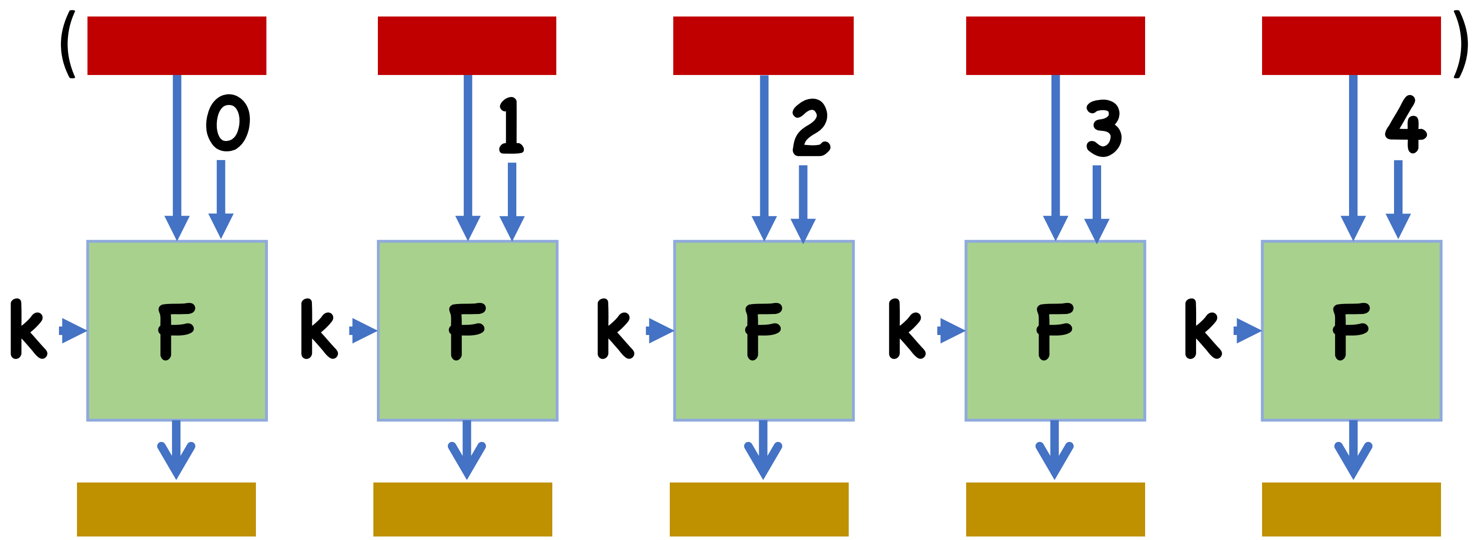
How do we handle larger messages?

Block-wise Authentication?



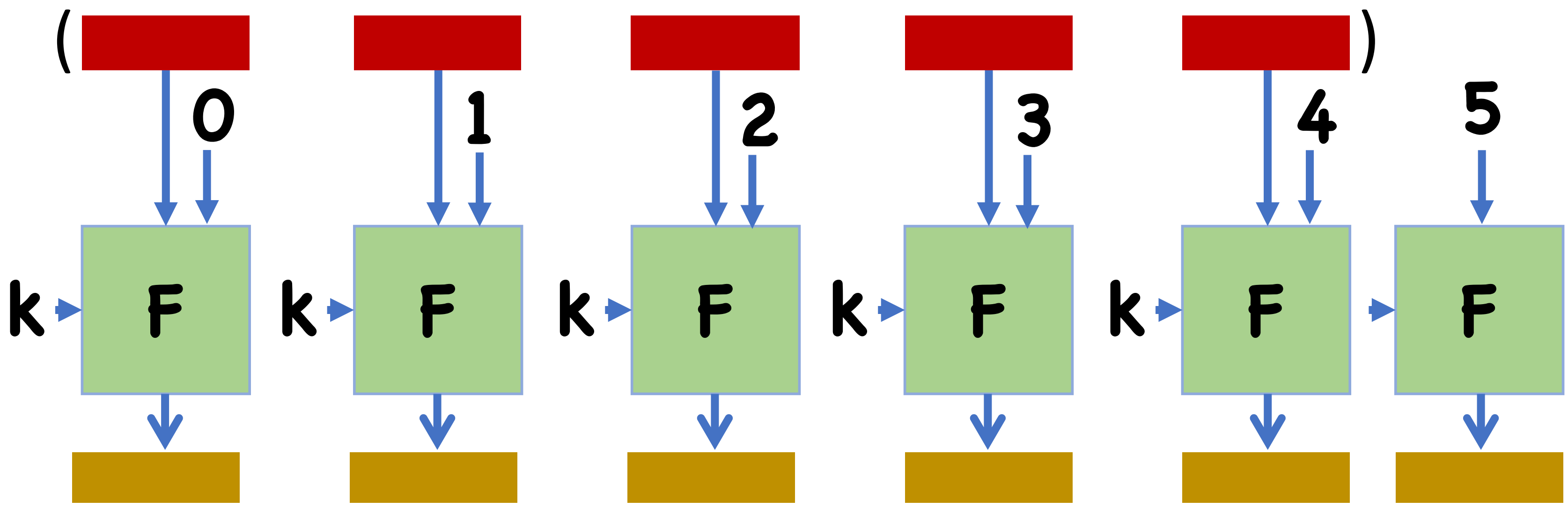
Why is this insecure?

Block-wise Authentication?



Why is this insecure?

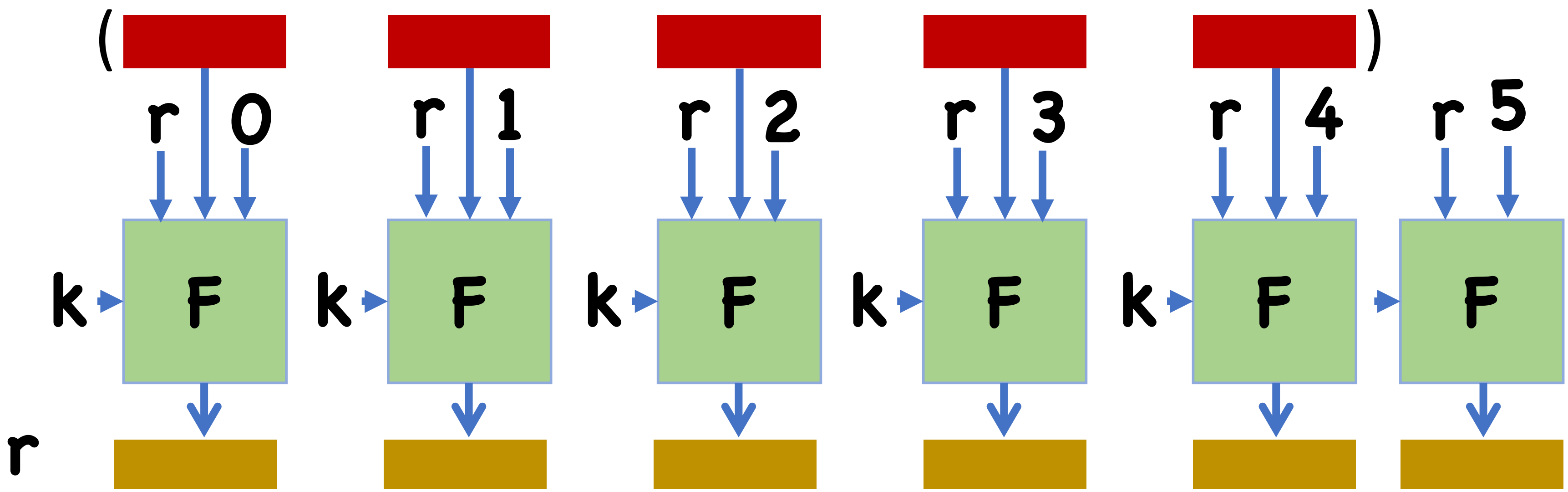
Block-wise Authentication?



Why is this insecure?

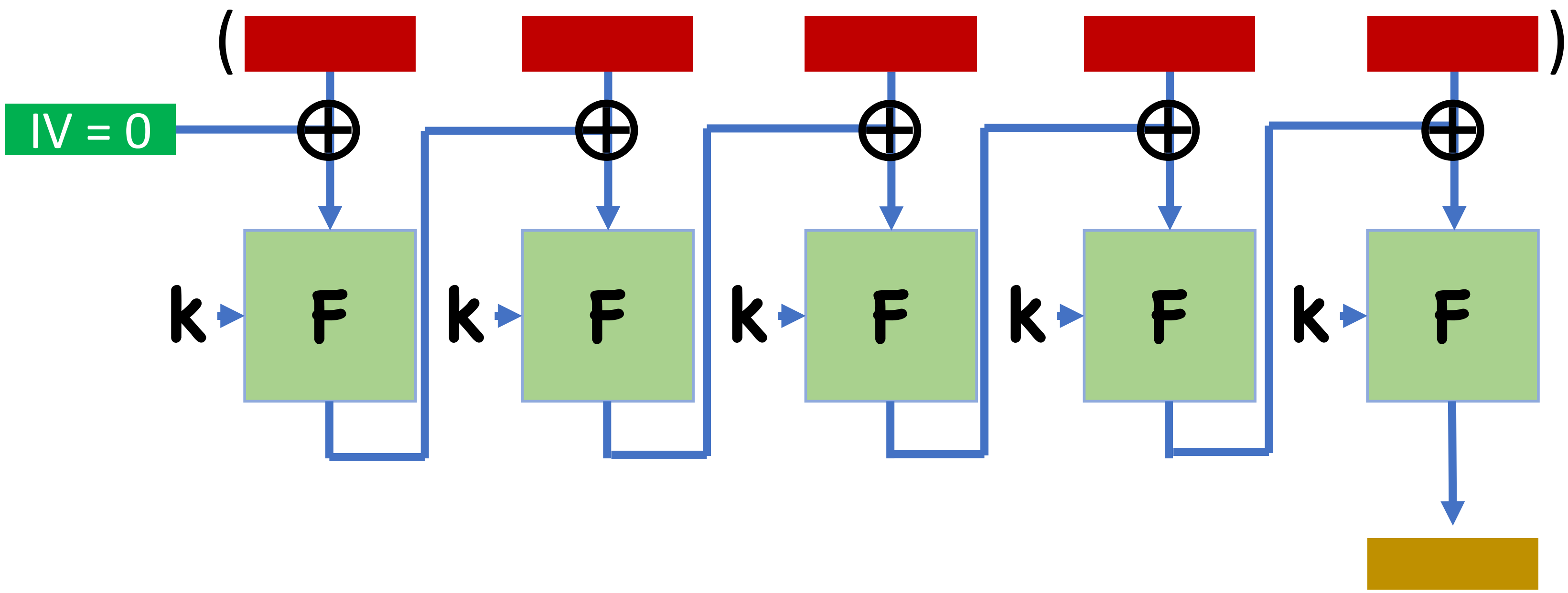
Block-wise Authentication?

r a random nonce



Secure, but not very useful in practice

CBC-MAC

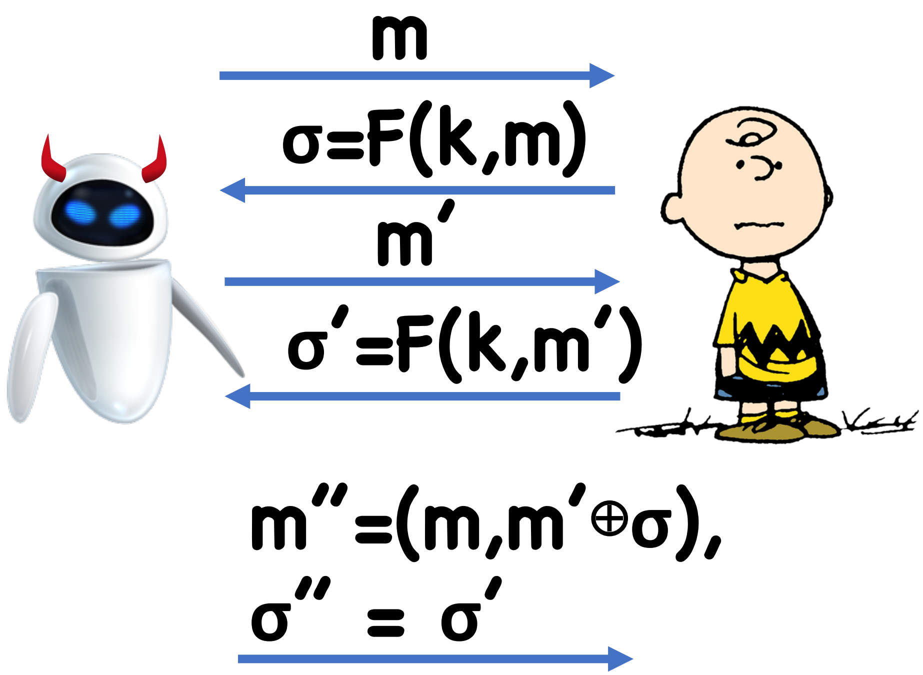


Theorem: CBC-MAC is a secure PRF for fixed-length messages

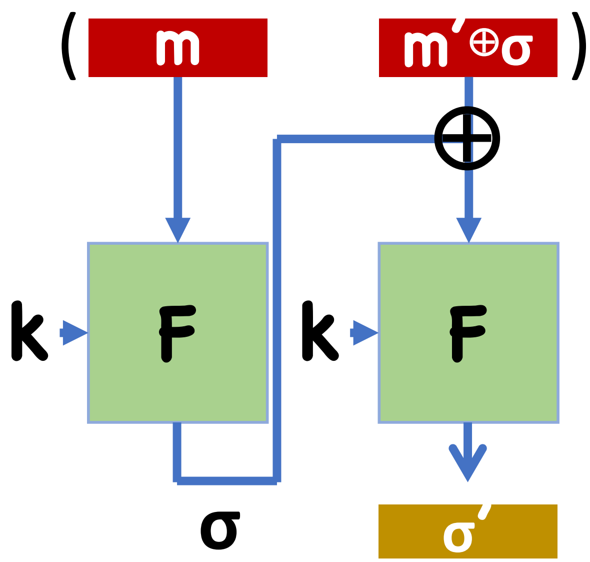
Variable Length Messages?

Basic CBC-MAC is insecure for variable length messages

Attack:



CBC-MAC



Handling Variable-Length Messages

Option 1:

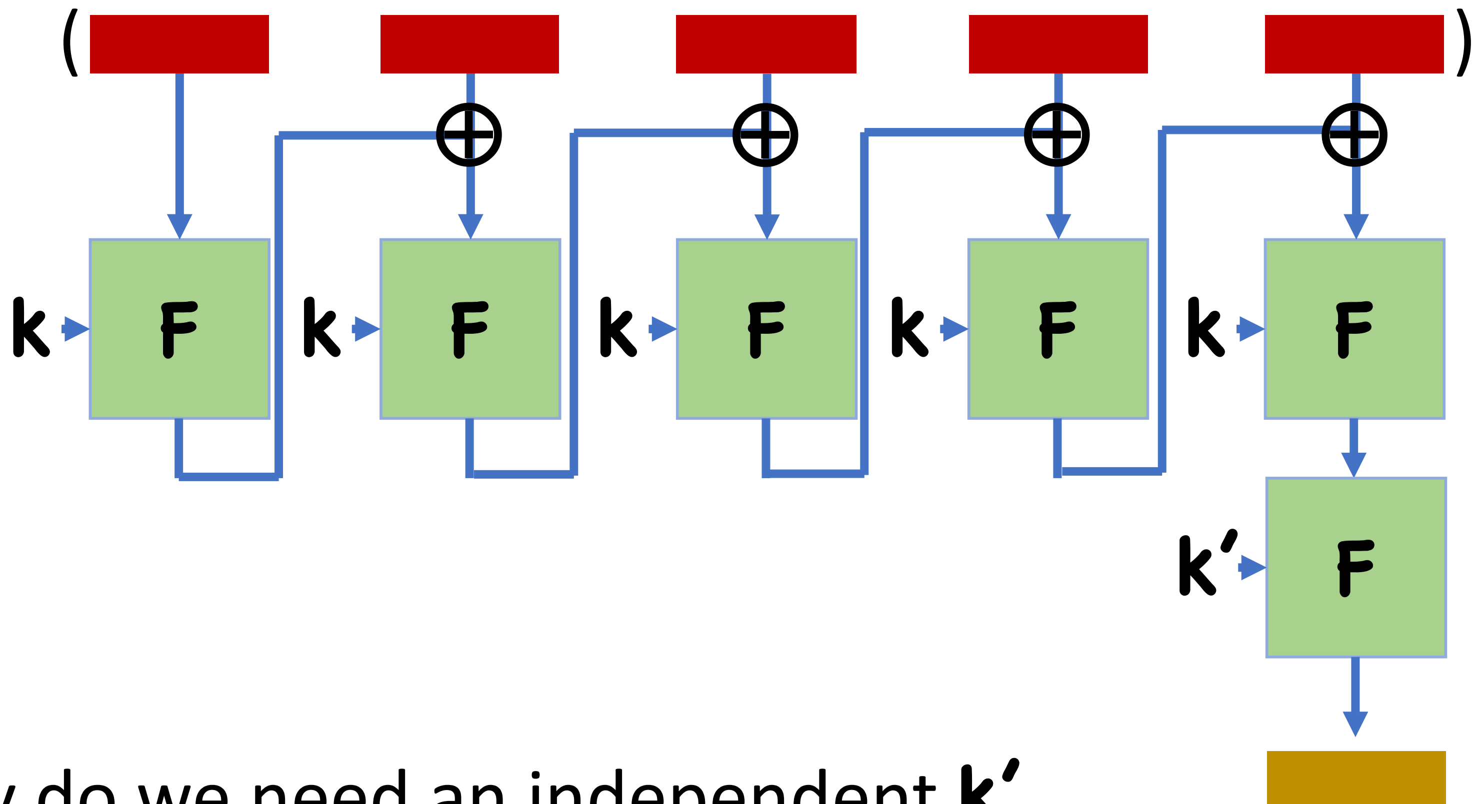
- Prepend with msg length before applying CBC-MAC
  - $\Rightarrow$ No two messages will have the same prefix

- Limitation: must know message length when you start computing MAC
  - Not always reasonable if you are authenticating a stream of data

- Why is appending msg length to end not good?

Handling Variable-Length Messages

Option 2: Encrypt-Last-Block

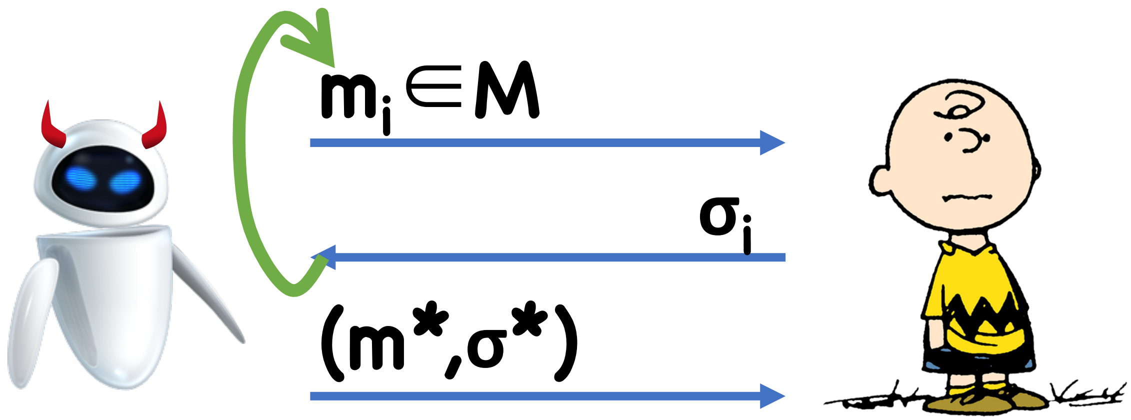


Q: Why do we need an independent **k'**

Alternate security notions

Strongly Secure MACs

No restriction

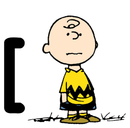


$k \leftarrow K_\lambda$

$\sigma \leftarrow MAC(k,m_i)$

Output 1 iff:

- $(m^*,\sigma^*) \notin \{(m_1,\sigma_1),\ldots\}$
- $Ver(k,m^*,\sigma^*) = 1$

SCMA-Adv( , λ) = Pr[ outputs 1]

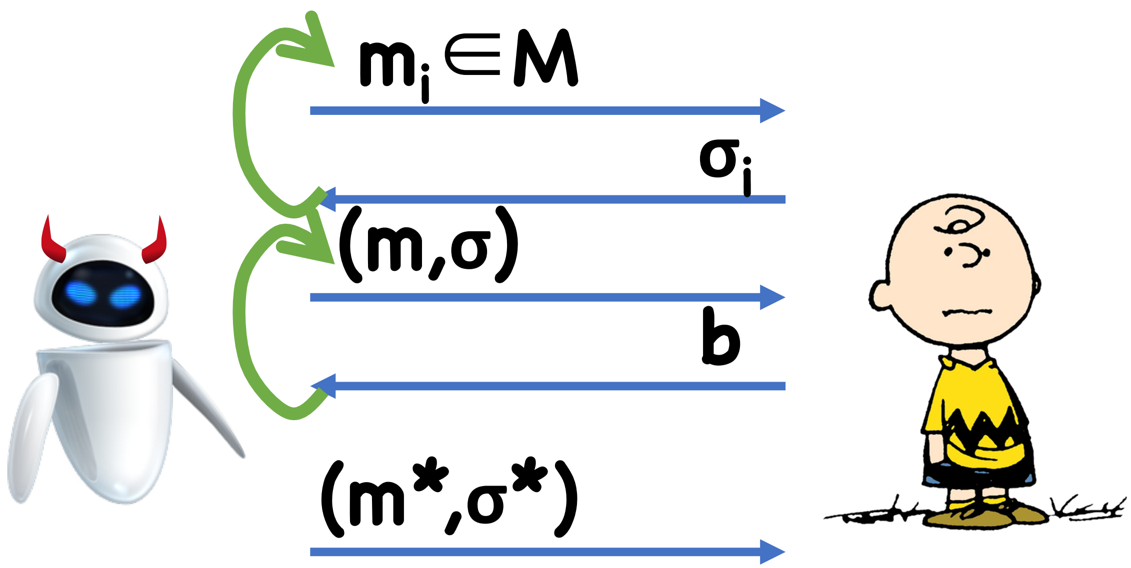
Strongly Secure MACs

Useful when you don't want to allow the adversary to change *any* part of the message

If there is only a single valid tag for each message (such as in the PRF-based MAC), then (weak) security also implies strong security

In general, though, strong security is stronger than weak security

Adding Verification Queries



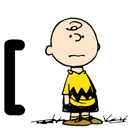
$k \leftarrow K_\lambda$

$\sigma_i \leftarrow MAC(k,m_i)$

$b \leftarrow Ver(k,m,\sigma)$

Output 1 iff:

- $m^* \notin \{m_1,\ldots\}$
- $Ver(k,m^*,\sigma^*) = 1$

CMA'-Adv( , λ) = Pr[ outputs 1]

Theorem: (MAC,Ver) is strongly CMA secure if and only if it is strongly CMA' secure

Proof

Strong CMA' → strong CMA: trivial

Strong CMA → strong CMA'

Idea: adversary could have always answered verification queries for himself

- If adv previously received the message/signature pair from challenger, then it must be valid
- If adv did not previously receive pair, most likely invalid

    (if not, then we have a strong forgery)

Timing Attacks on MACs

How do you implement check $F(k,m) == \sigma$?

String comparison often optimized for performance

Compare(A,B):

- **For i = 1,…,A.length**
  - **If A[i] != B[i], abort and return False;**
- **Return True;**

Time depends on number of initial bytes that match

Timing Attacks on MACs

To forge a message $\mathbf{m}$:

For each candidate first byte $\boldsymbol{\sigma_0}$:

- Query server on $(\mathbf{m}, \boldsymbol{\sigma})$ where first byte of $\boldsymbol{\sigma}$ is $\boldsymbol{\sigma_0}$
- See how long it takes to reject

First byte is $\boldsymbol{\sigma_0}$ that causes the longest response

- If wrong, server rejects when comparing first byte
- If right, server rejects when comparing second

Timing Attacks on MACs

To forge a message **m**:

Now we have first byte $\boldsymbol{\sigma_0}$

For each candidate second byte $\boldsymbol{\sigma_1}$:

- Query server on **(m, σ)** where first two bytes of **σ** are $\boldsymbol{\sigma_0, \sigma_0}$
- See how long it takes to reject

Second byte is $\boldsymbol{\sigma_1}$ that causes the longest response

•••

Thwarting Timing Attacks

Possibility:

- Use a string comparison that is guaranteed to take constant time
- Unfortunately, this is hard in practice, as optimized compilers could still try to shortcut the comparison

Possibility:

- Choose random block cipher key **k'**
- Compare by testing **F(k',A) == F(k', B)**
- Timing of "**==**" independent of how many bytes **A** and **B** share

Improving efficiency

Limitations of CBC-MAC

Many block cipher evaluations

Sequential

Carter Wegman MAC

k' = (k,h) where:

- **k** is a PRF key for $F:K\times R\rightarrow Y$
- **h** is sampled from a pairwise independent function family

MAC(k',m):

- Choose a random $r\leftarrow R$
- Set $\sigma = (r, F(k,r)\oplus h(m))$

Theorem: The Carter Wegman MAC is strongly CMA secure

Proof

Assume toward contradiction a PPT 

Hybrids…

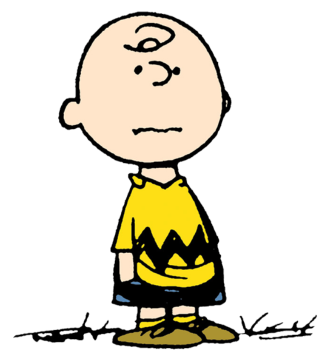
Proof

Hybrid 0

$m_i \in M$

$\sigma_i=(r_i,t_i)$

(m^*,r^*,t^*)



$k \leftarrow K$

h

$r_i \leftarrow R$

$t_i \leftarrow F(k,r) \oplus h(m)$

Output 1 iff:

- $(m^*,r^*,t^*) \notin \{(m_i,r_i,t_i)\}$
- $F(k,r^*) \oplus h(m^*) = t^*$

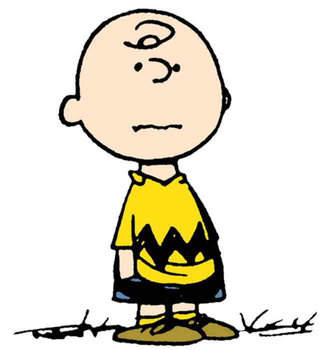
Proof

<u>Hybrid 1</u>

$m_i \in M$ →

← $\sigma_i = (r_i, t_i)$

(m^*, r^*, t^*) →



$k \leftarrow K$

h

(Distinct $\mathbf{r_i}$)

$r_i \leftarrow R$

$t_i \leftarrow F(k,r) \oplus h(m)$

Output 1 iff:

- $(m^*, r^*, t^*) \notin \{(m_i, r_i, t_i)\}$
- $F(k, r^*) \oplus h(m^*) = t^*$

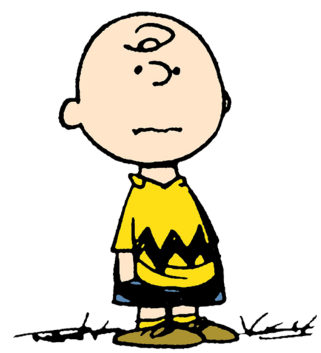
Proof

Hybrid 2

$m_i \in M$ →

← $\sigma_i=(r_i,t_i)$

(m^*,r^*,t^*) →



$H \leftarrow Funcs$

h

(Distinct r_i)

$r_i \leftarrow R$

$t_i \leftarrow H(r) \oplus h(m)$

Output 1 iff:

- $(m^*,r^*,t^*) \notin \{(m_i,r_i,t_i)\}$
- $H(r^*) \oplus h(m^*) = t^*$

Proof

Claim: In Hybrid 2, negligible success probability

Possibilities:

- $r^* \notin \{r_i\}$: then value of $H(r^*)$ hidden from adversary, so $\Pr[H(r^*) \oplus h(m^*) = t^*]$ is $1/|Y|$

- $r^* = r_i$ for some i: then $m^* \neq m_i$ (why?)

  h completely hidden from adversary

  $\Pr[H(r^*) \oplus h(m^*) = t^*]$

  $= \Pr[h(m^*) = t^* \oplus t_i \oplus h(m_i)] = 1/|Y|$

Proof

Hybrid 1 and 2 are indistinguishable

- PRF security

Hybrid 0 and 1 are indistinguishable

- W.h.p. random $\mathbf{r_i}$ will be distinct

Therefore, negligible success probability in Hybrid 0

Efficiency of CW MAC

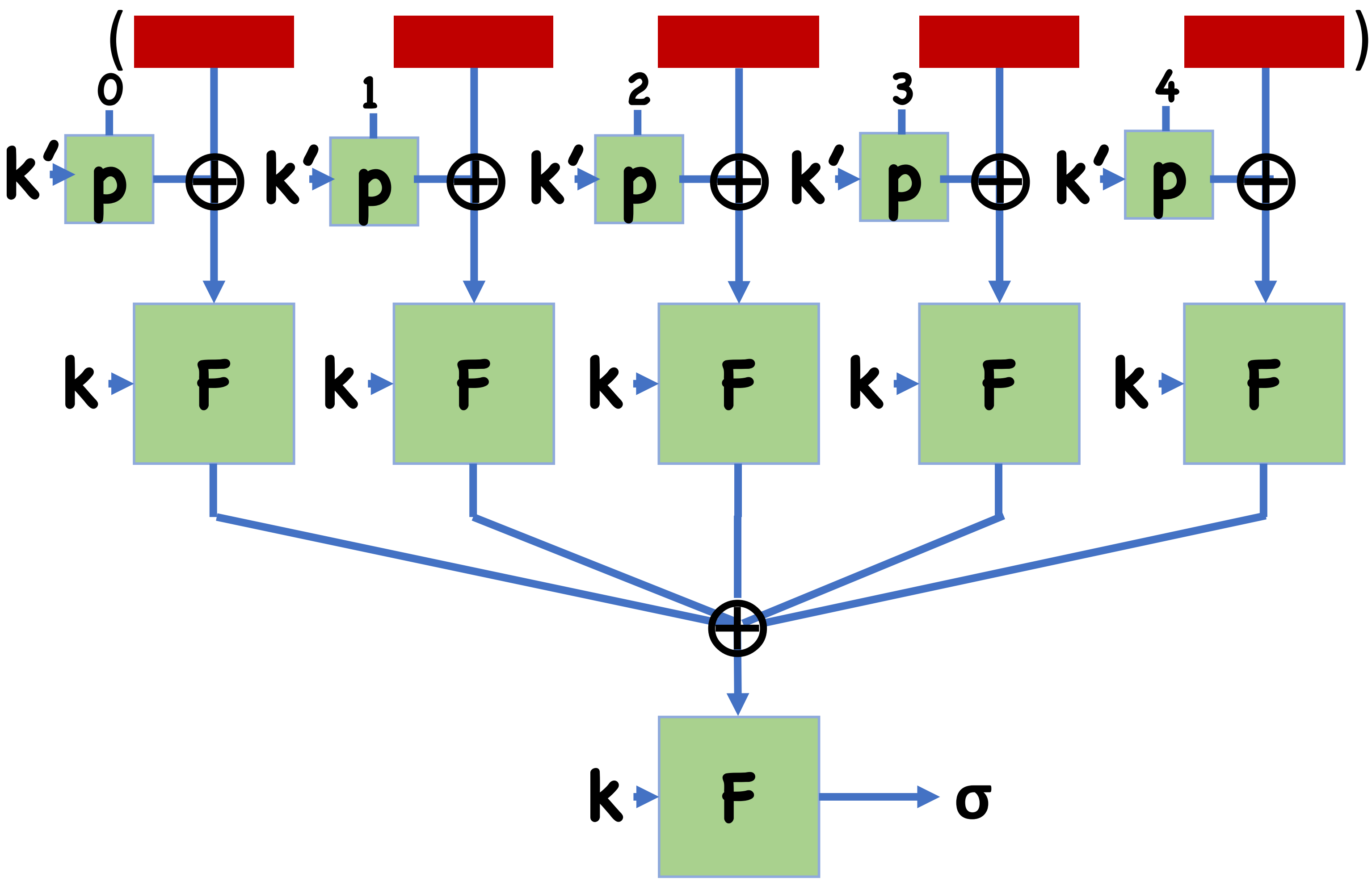
MAC(k',m):

- Choose a random **r←R**
- Set **σ = (r, F(k,r)⊕h(m))**

h much more efficient that PRFs

PRF applied only to small nonce **r**
h applied to large message **m**

PMAC: A Parallel MAC



Next Time

Authenticated Encryption: combining secrecy and integrity