COS 433/Math 473: Cryptography

Mark Zhandry

Princeton University

Fall 2020

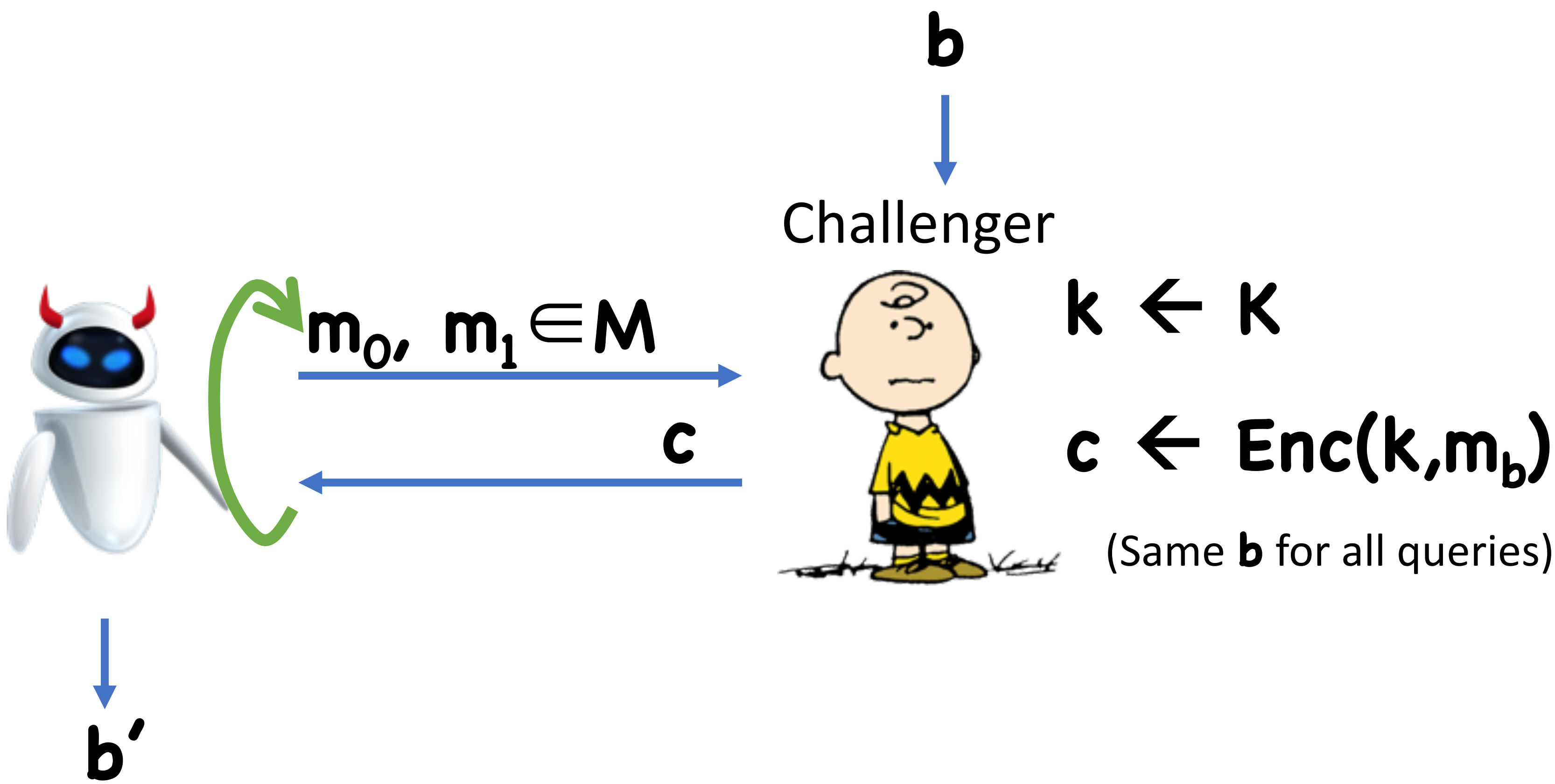
Announcements/Reminders

HW2 due September 29

PR1 Due October 6

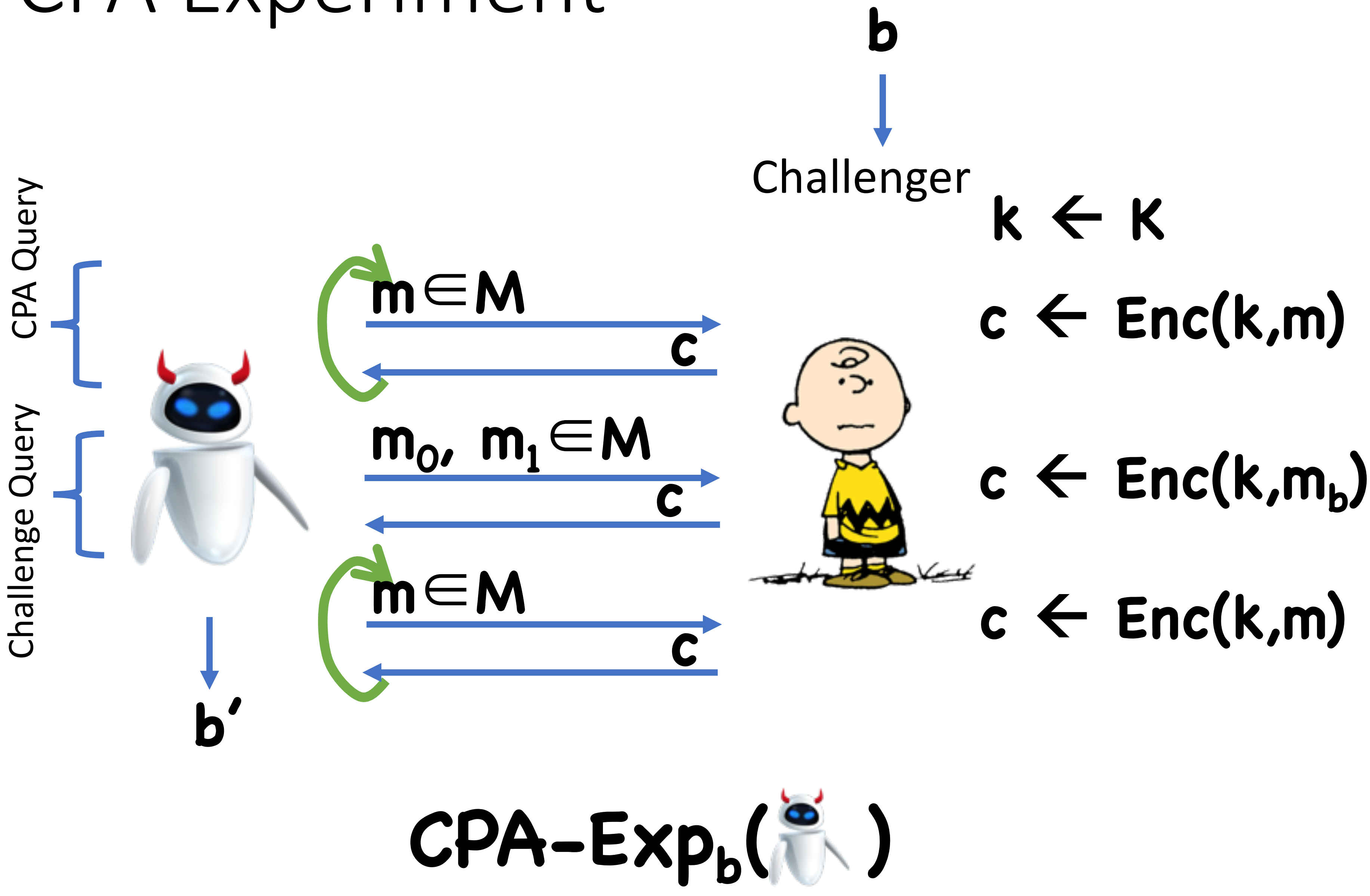
Previously on COS 433…

Left-or-Right Experiment

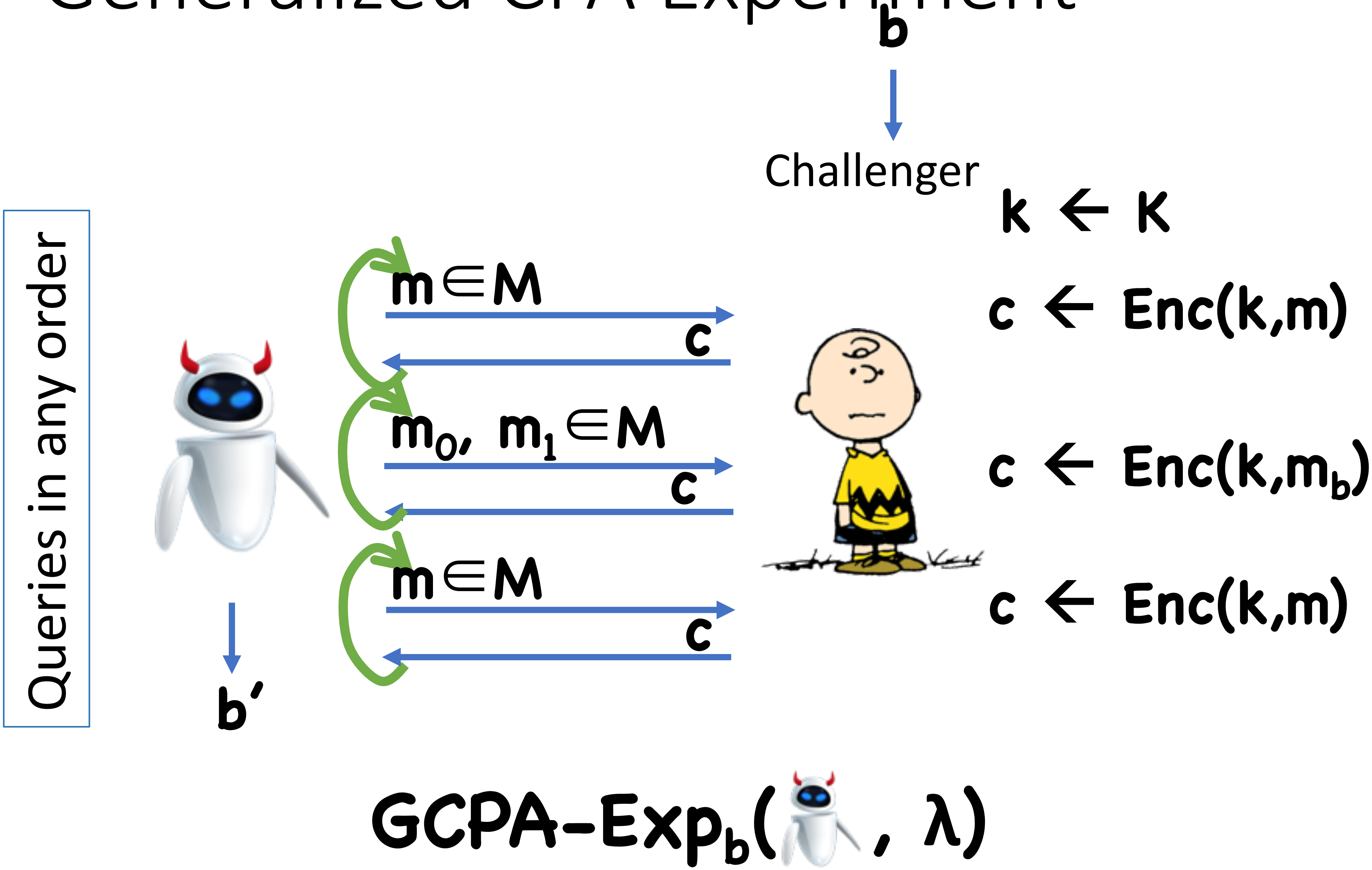


LoR-Exp$_b$( , λ)

CPA Experiment



Generalized CPA Experiment



Equivalences

Theorem:

Left-or-Right indistinguishability

$\Updownarrow$

CPA-security

$\Updownarrow$

Generalized CPA-security

Today

Finish proof

Constructing many-time schemes

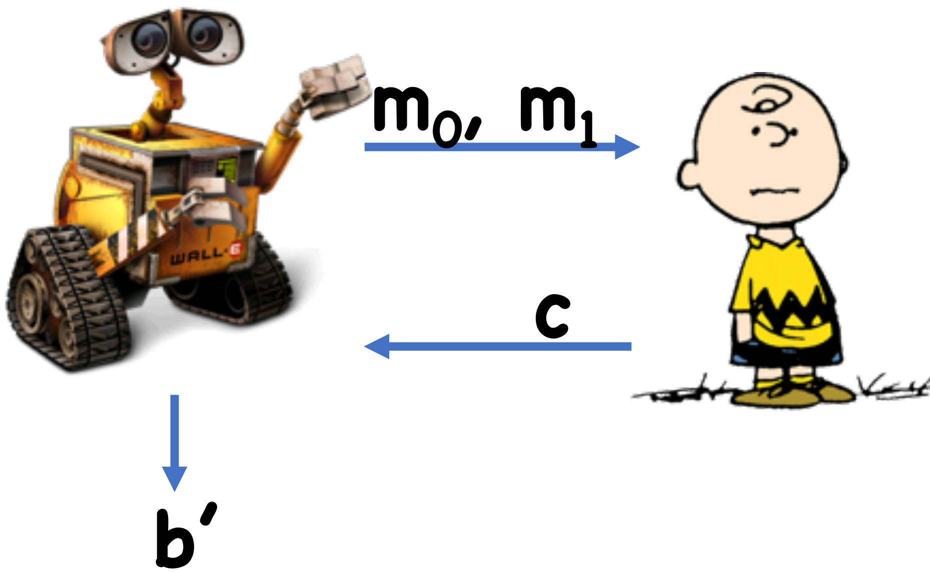
Proof

(regular) CPA → Left-or-Right

- Assume towards contradiction that we have an adversary  for the **LoR Indistinguishability**

- Hybrids!

Hybrid **i**:



$k \leftarrow K$

If at most **i** queries so far,

$c \leftarrow Enc(k,m_0)$

If more than **i** queries so far,

$c \leftarrow Enc(k,m_1)$

Proof

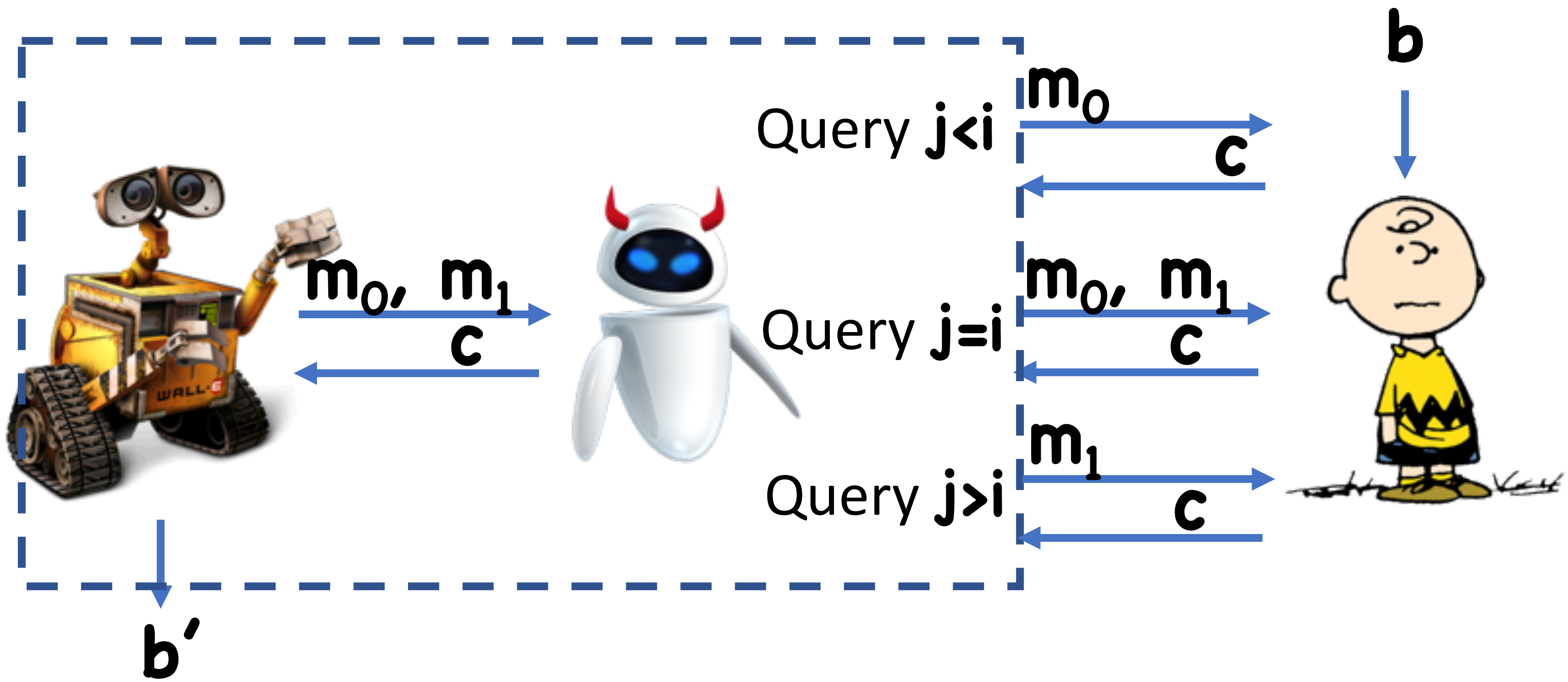
(regular) CPA → Left-or-Right

- Hybrid **0** is identical to **LoR–Exp$_1$(**  **, λ)**

- Hybrid **q** is identical to **LoR–Exp$_0$(**  **, λ)**

- We know that  distinguishes Hybrid **q** and Hybrid **0** with advantage **ε**

  $\Rightarrow$ $\exists$ **i** s.t.  distinguishes Hybrid **i** and Hybrid **i–1** with advantage **ε/q**



Pr[1←CPA-Exp$_b$(, λ)] = Pr[1← in Hybrid **i-b**]

Proof

(regular) CPA → Left-or-Right

| Pr[1←CPA-Exp$_0$( , λ)]

- Pr[1←CPA-Exp$_1$( , λ)] |

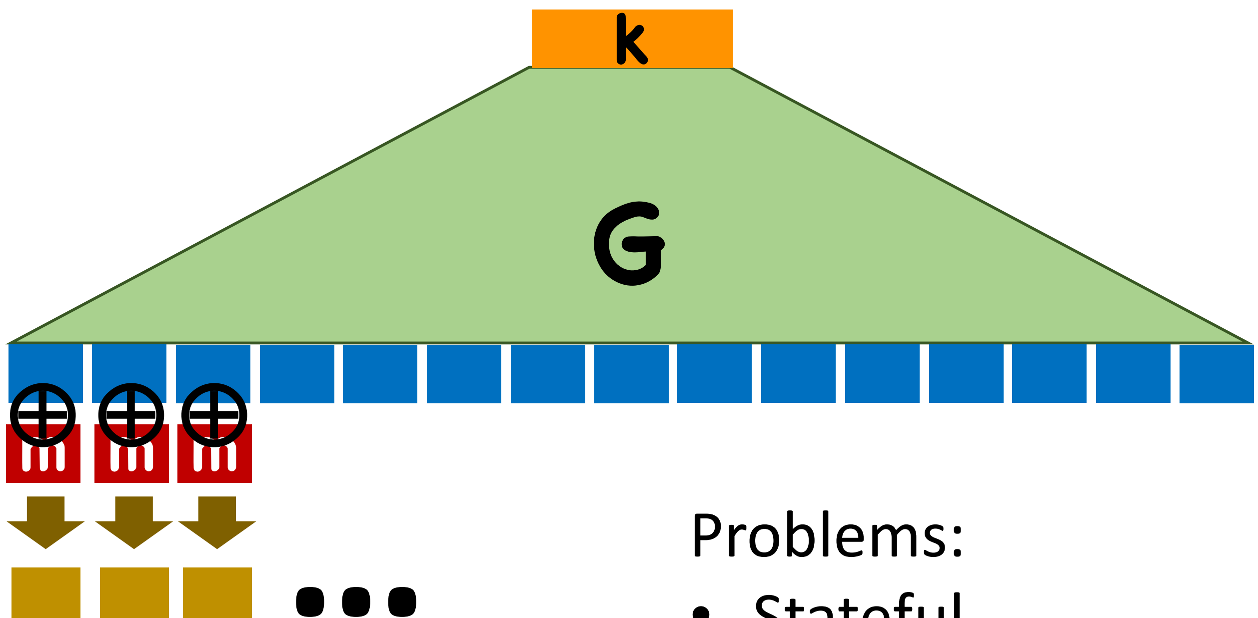
= | Pr[1←  in Hybrid **i**]

- Pr[1←  in Hybrid **i-1**] | ≥ ε/q

Constructing CPA-secure Encryption

Constructing CPA-secure Encryption

Starting point: stream ciphers = PRG + OTP for multiple messages

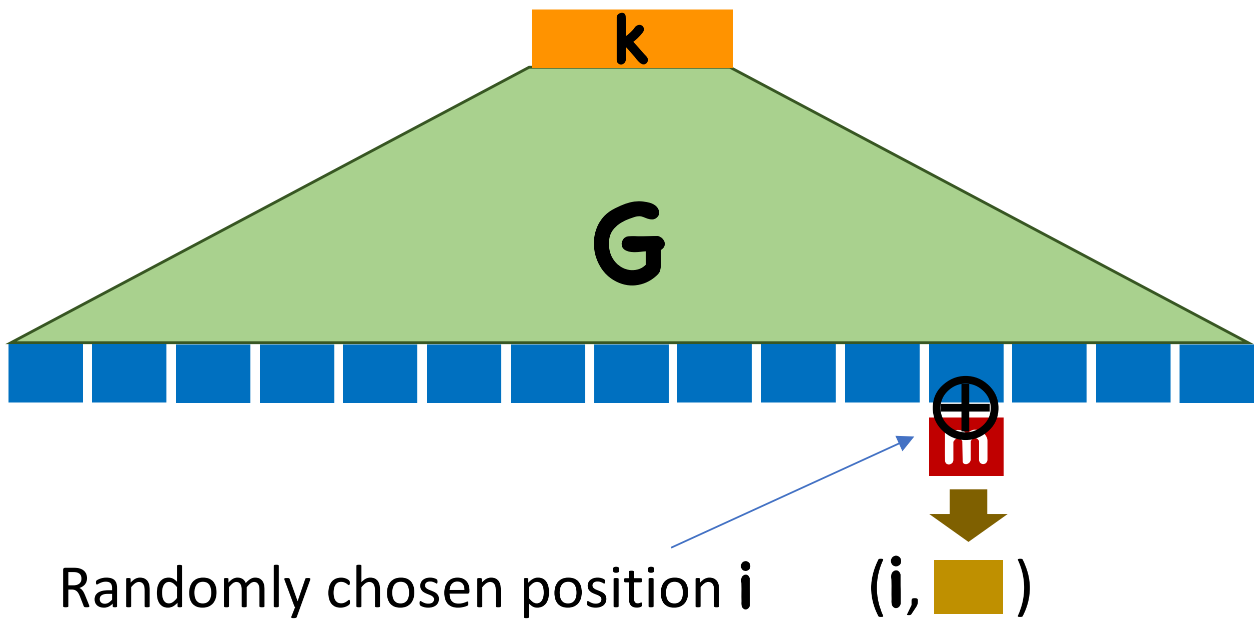


Problems:

- Stateful
- Need to synchronize with Bob

Constructing CPA-secure Encryption

Idea 1: Use random position to encrypt



Analysis

As long as the two encryptions never pick the same location, we will have security

Pr[Collision] = ?

Pr[Collision]

Consider event $E_{j,k} = (i_j = i_k)$

$$\Rightarrow \Pr[E_{j,k}] = 1/n$$

$$\Pr[\text{Collision}] = \Pr[E_{1,2} \text{ or } E_{1,3} \text{ or } \ldots \text{ or } E_{j,k} \text{ or } \ldots]$$

Union bound:

$$\Pr[\text{Collision}] \leq \Sigma_{j,k}\Pr[E_{j,k}] = \Sigma_{j,k}(1/n) = q(q-1)/2n$$

Analysis

As long as the two encryptions never pick the same location, we will have security

$Pr[Collision] < q^2/2n$, where

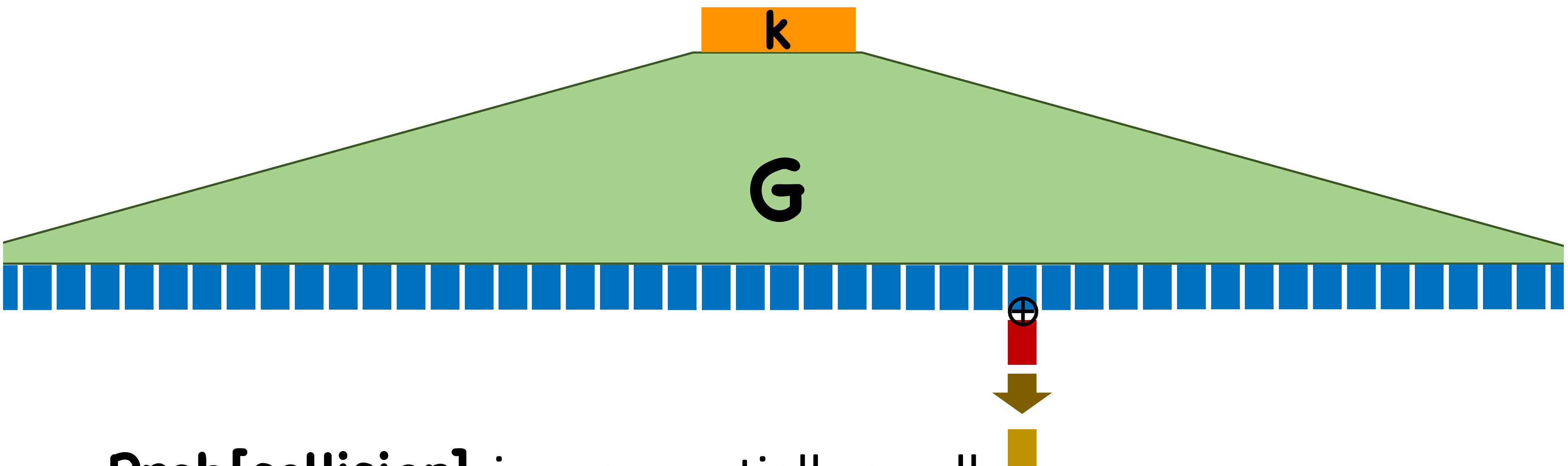
- **q =** number of messages encrypted
- **n =** number of blocks

If collision, then no security ("two-time pad")

If no collision, then security maintained

What if…

The PRG has **exponential** stretch



Prob[collision] is exponentially small
However, computing PRG takes exponential time

What if…

The PRG has **exponential** stretch

AND, it was possible to compute any 1 block of output of the PRG

- In polynomial time
- Without computing the entire output

In other words, given a key, can efficiently compute the function $\mathbf{F(k, x) = G(k)_x}$

Pseudorandom Functions

Functions that “look like” random functions

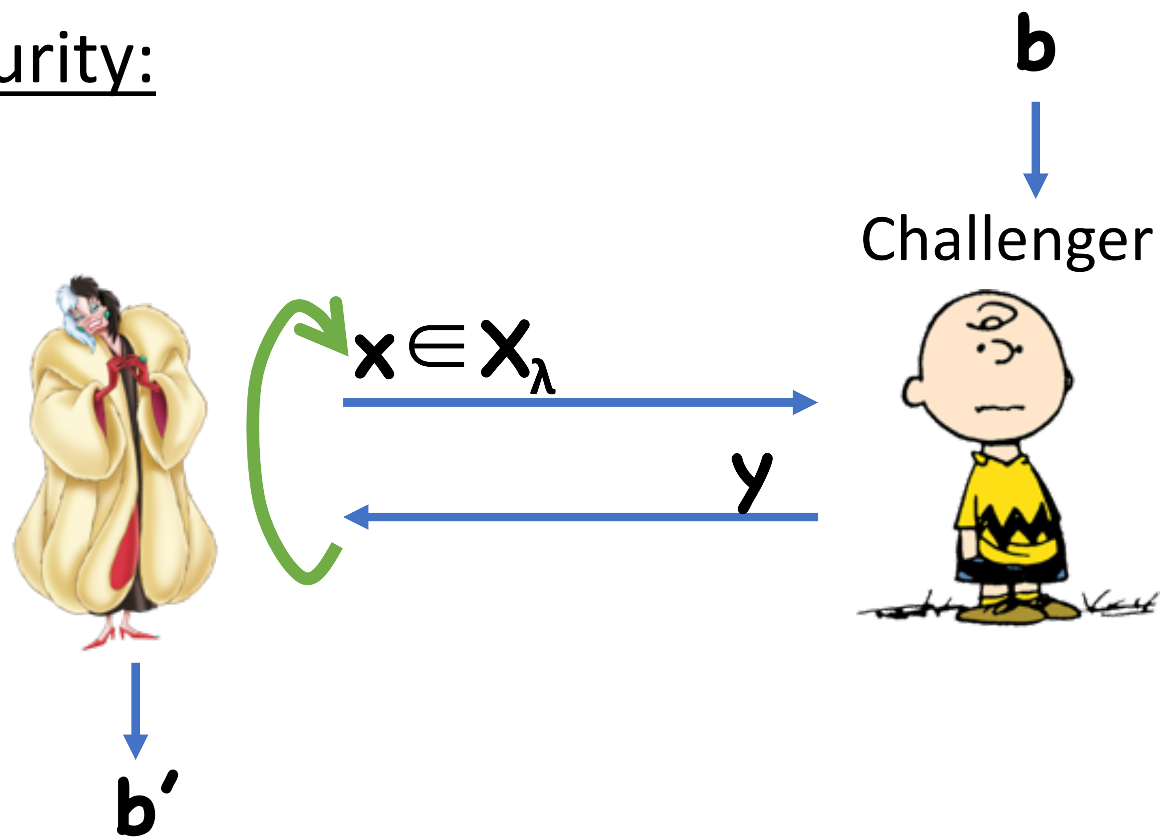
Syntax:

- Key space $\mathbf{K}_\lambda$
- Domain $\mathbf{X}_\lambda$
- Co-domain/range $\mathbf{Y}_\lambda$
- Function $\mathbf{F}:\mathbf{K}_\lambda \times \mathbf{X}_\lambda \rightarrow \mathbf{Y}_\lambda$

Correctness: $\mathbf{F}$ is a function (deterministic)

Pseudorandom Functions

<u>Security:</u>



Pseudorandom Functions

<u>Security:</u>

b=0

Challenger

$k \leftarrow K_\lambda$

$x \in X_\lambda$

y

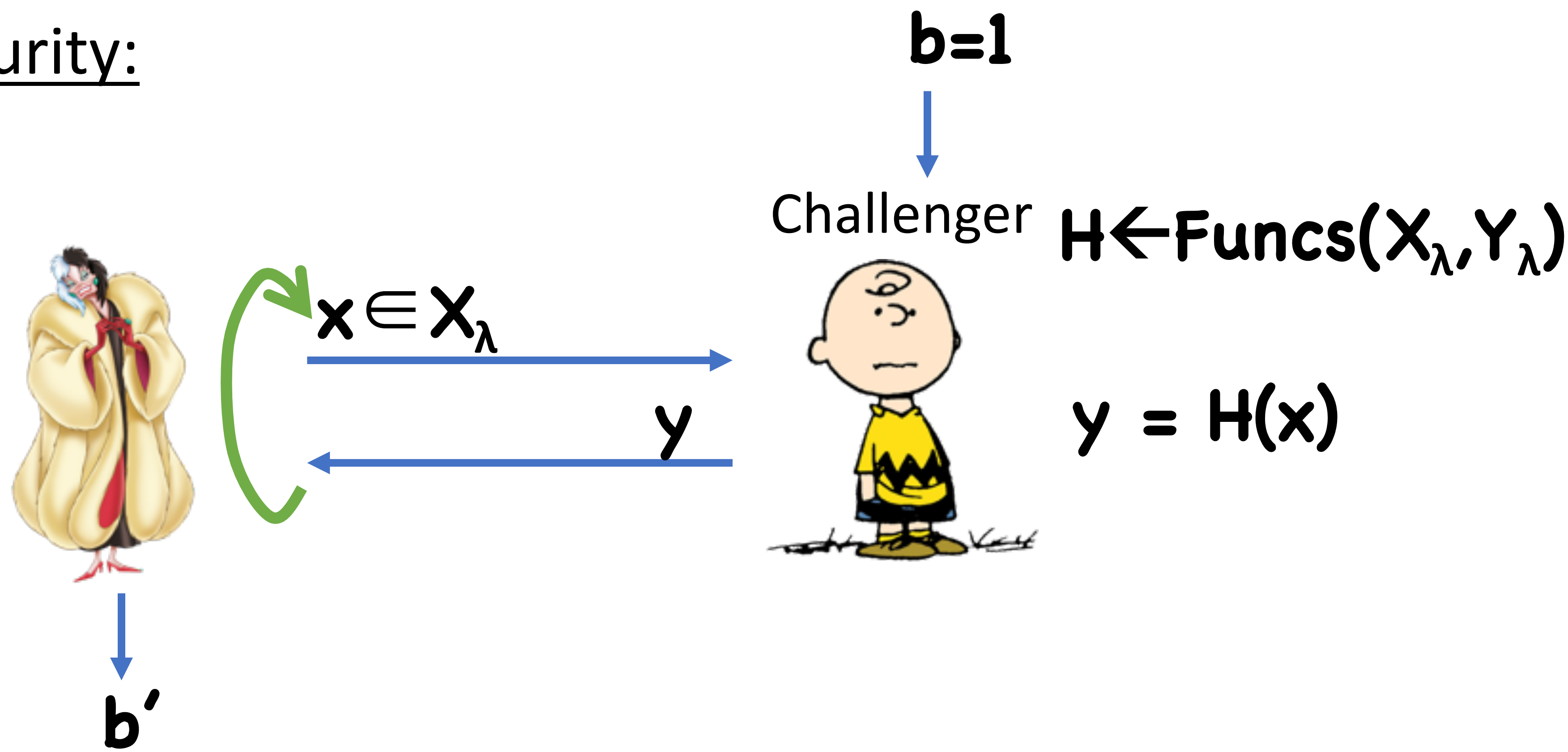
$y \leftarrow F(k,x)$

b′

PRF-Exp$_0$( **, λ)**

Pseudorandom Functions

Security:



PRF-Exp$_1$( **, λ)**

PRF Security Definition

Definition: F is a secure PRF if, for all  running in polynomial time, $\exists$ negligible **ε** such that:

$\Big|$ **Pr[1←PRF-Exp$_0$(**  **, λ)]**

− Pr[1←PRF-Exp$_1$( **, λ)]** $\Big|$ **≤ ε(λ)**

Using PRFs to Build Encryption

Enc(k, m):

- Choose random $\mathbf{r} \leftarrow \mathbf{X}_\lambda$
- Compute $\mathbf{y} \leftarrow \mathbf{F}(\mathbf{k},\mathbf{r})$
- Compute $\mathbf{c} \leftarrow \mathbf{y} \oplus \mathbf{m}$
- Output **(r,c)**

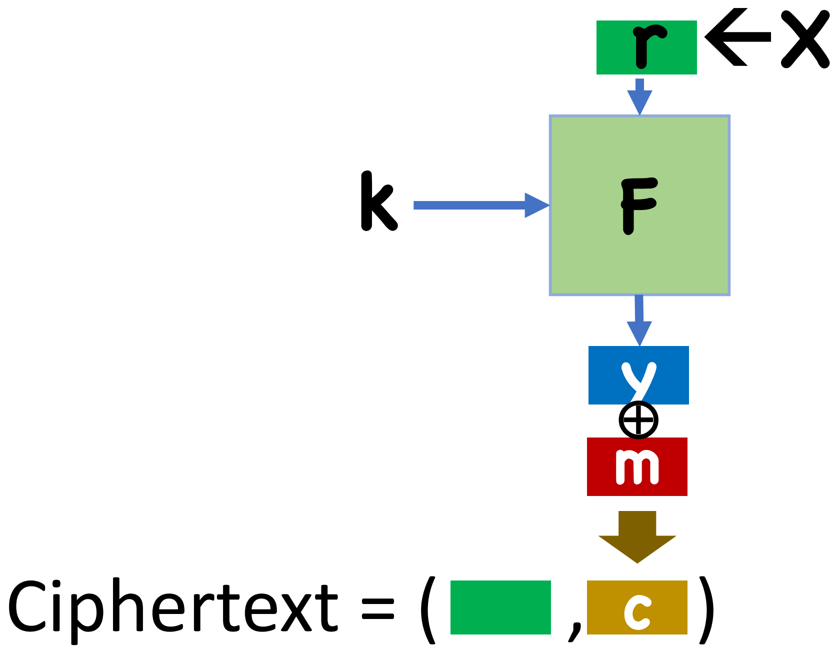
Dec(k, (r,c)):

- Compute $\mathbf{y}' \leftarrow \mathbf{F}(\mathbf{k},\mathbf{r})$
- Compute and output $\mathbf{m}' \leftarrow \mathbf{c} \oplus \mathbf{y}'$

Correctness:

- $\mathbf{y}' = \mathbf{y}$ since **F** is deterministic
- $\mathbf{m}' = \mathbf{c} \oplus \mathbf{y} = \mathbf{y} \oplus \mathbf{m} \oplus \mathbf{y} = \mathbf{m}$

Using PRFs to Build Encryption



Security

Theorem: If $\mathbf{F}$ is a secure PRF with domain $\mathbf{X_\lambda}$ and $|\mathbf{X_\lambda}|$ is superpoly, then **(Enc,Dec)** is **LoR** secure.

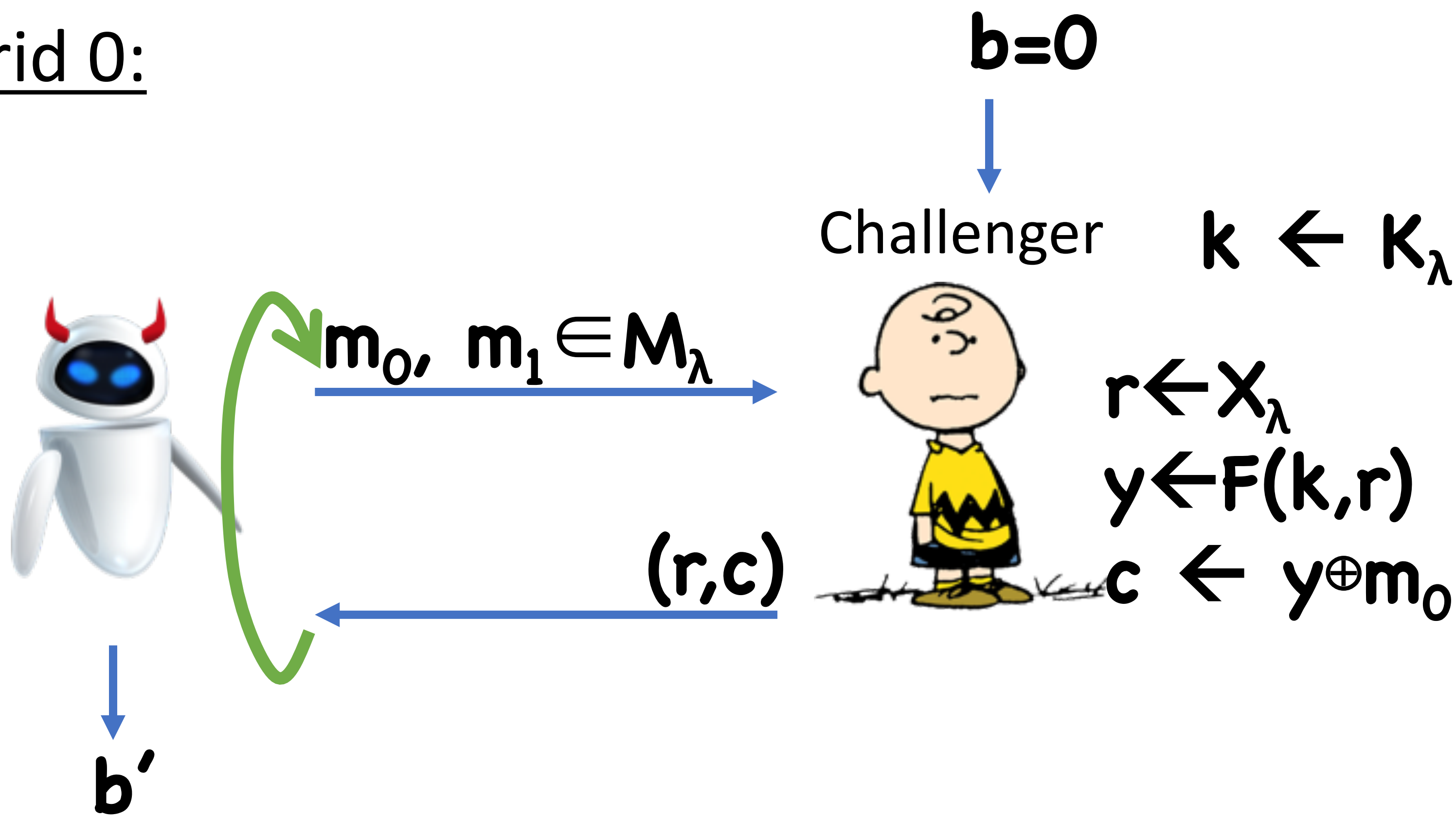
Proof

Assume toward contradiction that there exists a  breaking **(Enc,Dec)**

Hybrids…

Proof

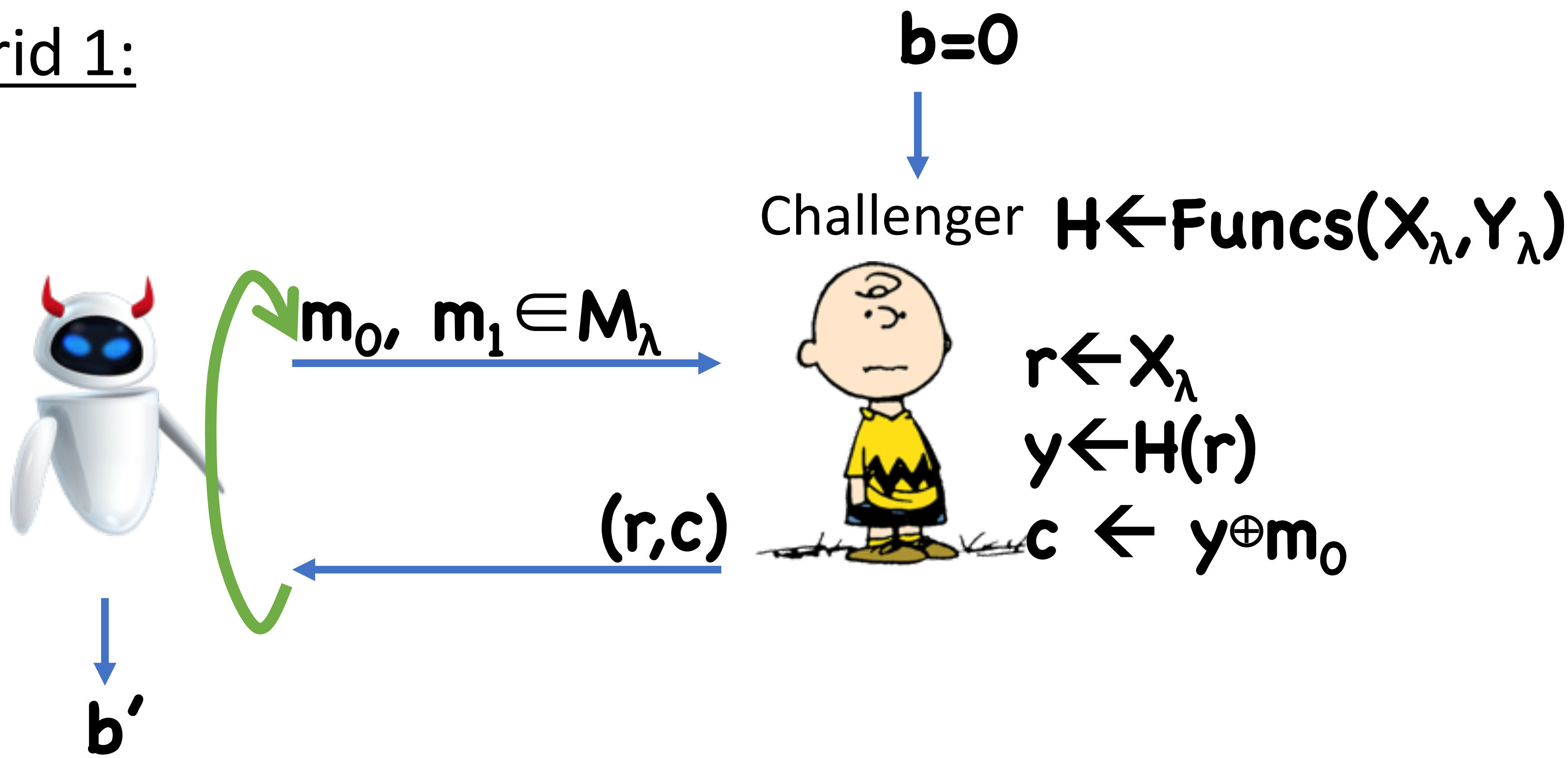
<u>Hybrid 0:</u>



LoR-Exp$_0$( **, λ)**

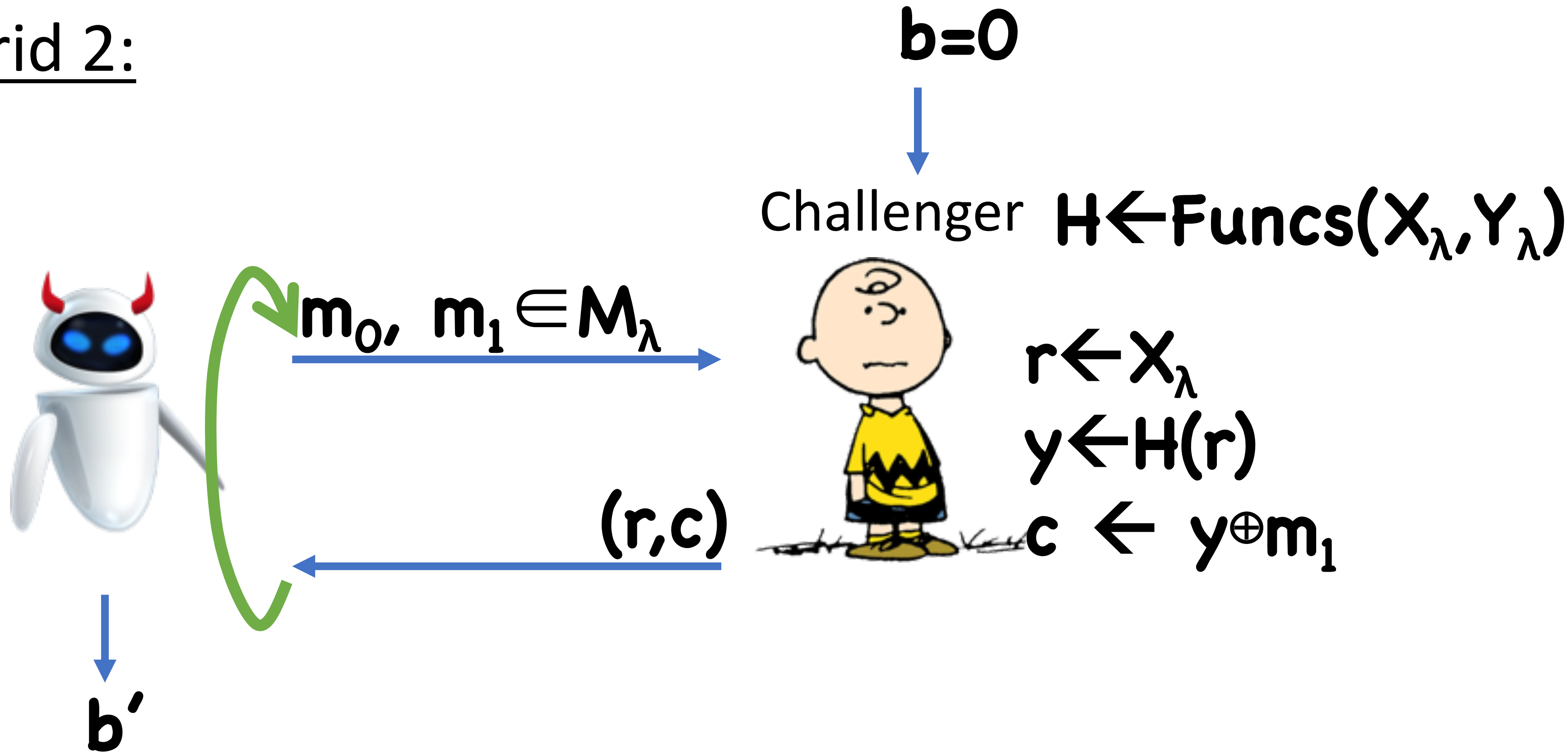
Proof

<u>Hybrid 1:</u>



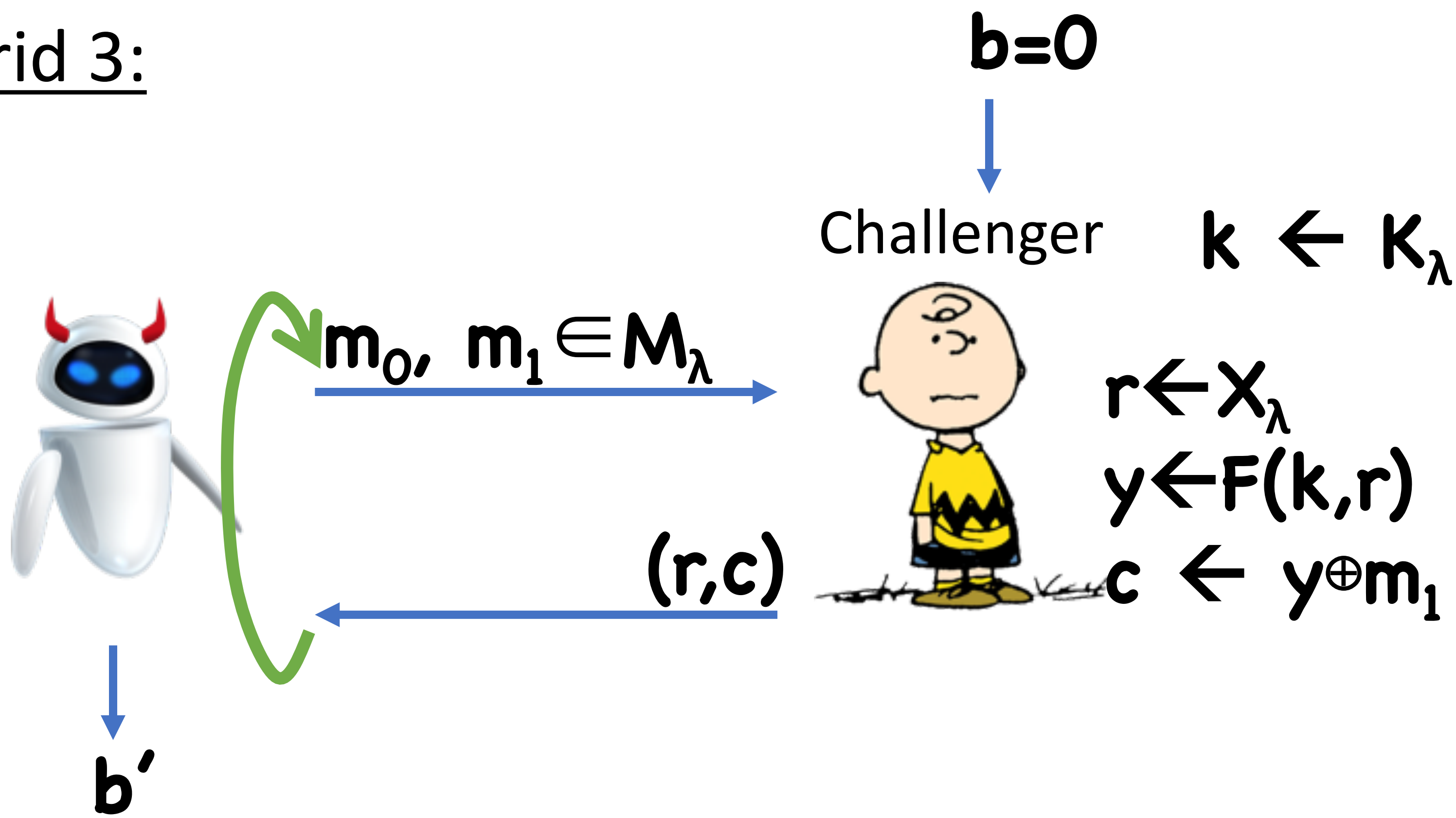
Proof

Hybrid 2:



Proof

Hybrid 3:



$LoR\text{-}Exp_1($  **$, \lambda)$**

Proof

Assume toward contradiction that there exists a  with advantage **ε** in breaking **(Enc,Dec)**

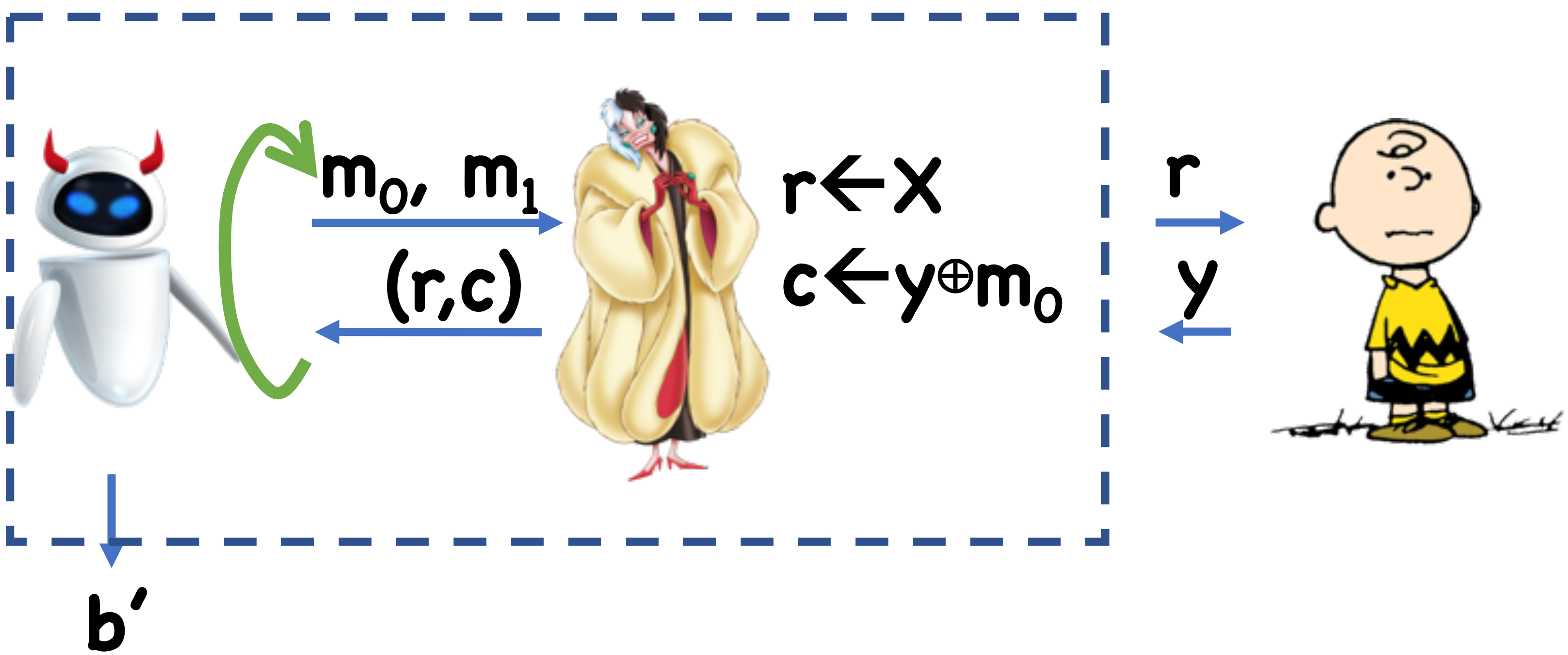
 distinguishes Hybrid 0 from Hybrid 3 with advantage **ε**, so either 

- Dist. Hybrid 0 from Hybrid 1 with adv. $(\varepsilon/2)-q^2/4|X|$
- Dist. Hybrid 1 from Hybrid 2 with adv. $q^2/2|X|$
- Dist. Hybrid 2 from Hybrid 3 with adv. $(\varepsilon/2)-q^2/4|X|$

Proof

Suppose  distinguishes Hybrid 0 from Hybrid 1

Construct 



Proof

Suppose  distinguishes Hybrid 0 from Hybrid 1

Construct 

- **PRF-Exp$_0$(****, λ)** corresponds to Hybrid 0
- **PRF-Exp$_1$(****, λ)** corresponds to Hybrid 1

Therefore,  has advantage **$(\varepsilon/2)-q^2/4|X|$**

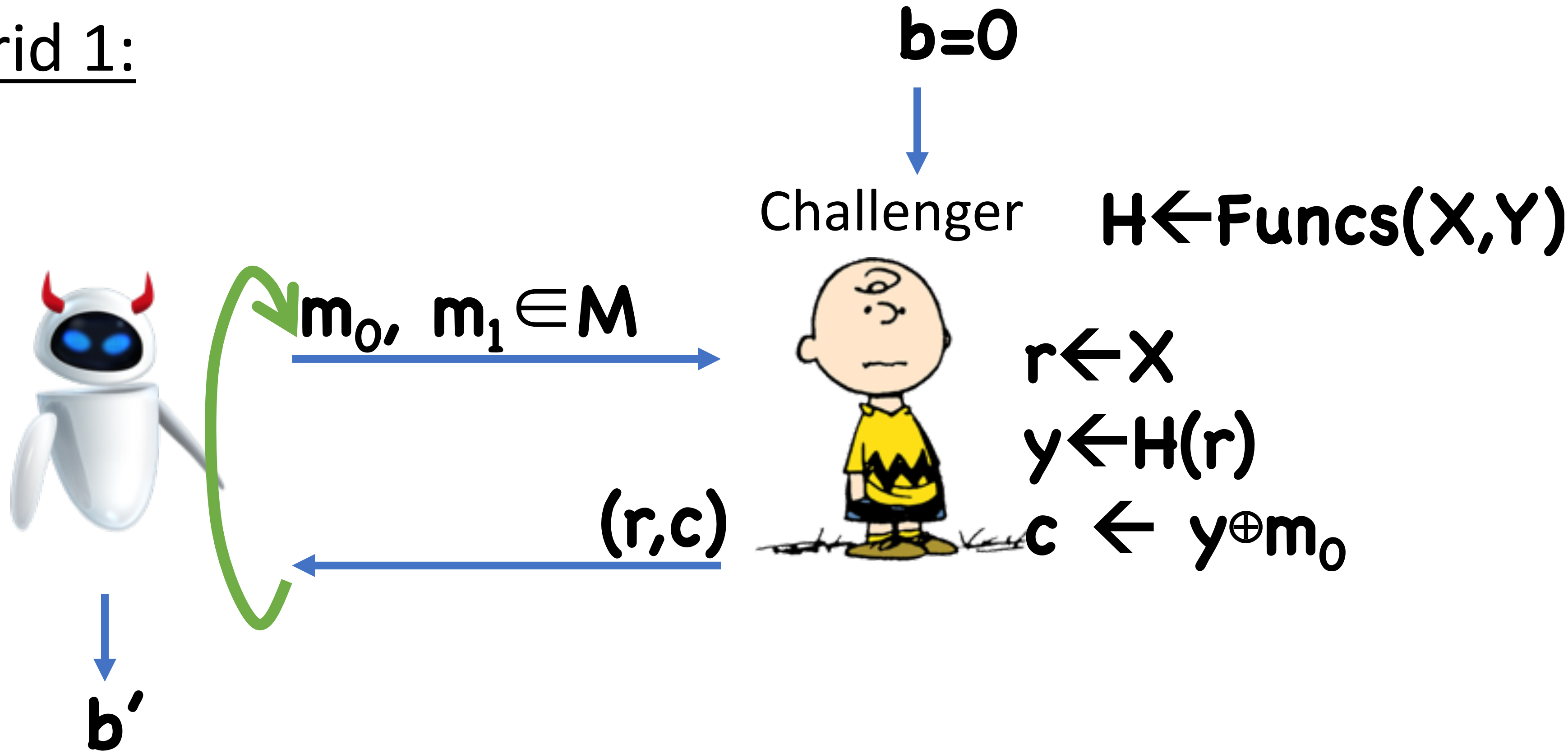
$\Rightarrow$ contradiction

Proof

Suppose  distinguishes Hybrid 1 from Hybrid 2

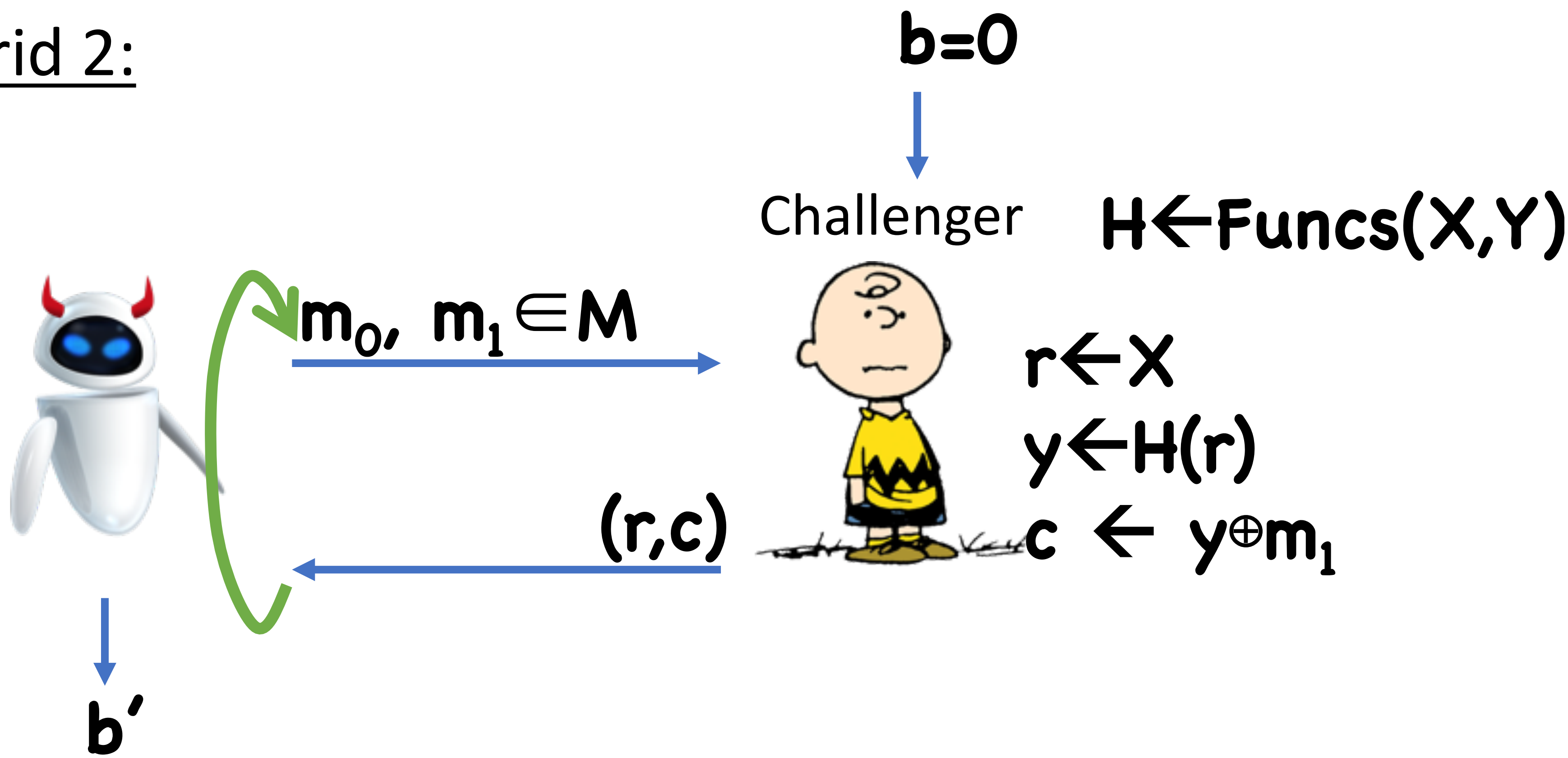
Proof

Hybrid 1:



Proof

Hybrid 2:



Proof

Suppose  distinguishes Hybrid 1 from Hybrid 2

As long as the **r**'s for every query are distinct, the **y**'s for each query will look like truly random strings

In this case, encrypting $\mathbf{m_0}$ vs $\mathbf{m_1}$ will be perfectly indistinguishable

- By OTP security

Proof

Suppose  distinguishes Hybrid 1 from Hybrid 2

Therefore, advantage is **≤Pr[**collision in the **r**'s**]**

$< q^2/2|X|$

Proof

Suppose  distinguishes Hybrid 2 from Hybrid 3

Almost identical to the 0/1 case…

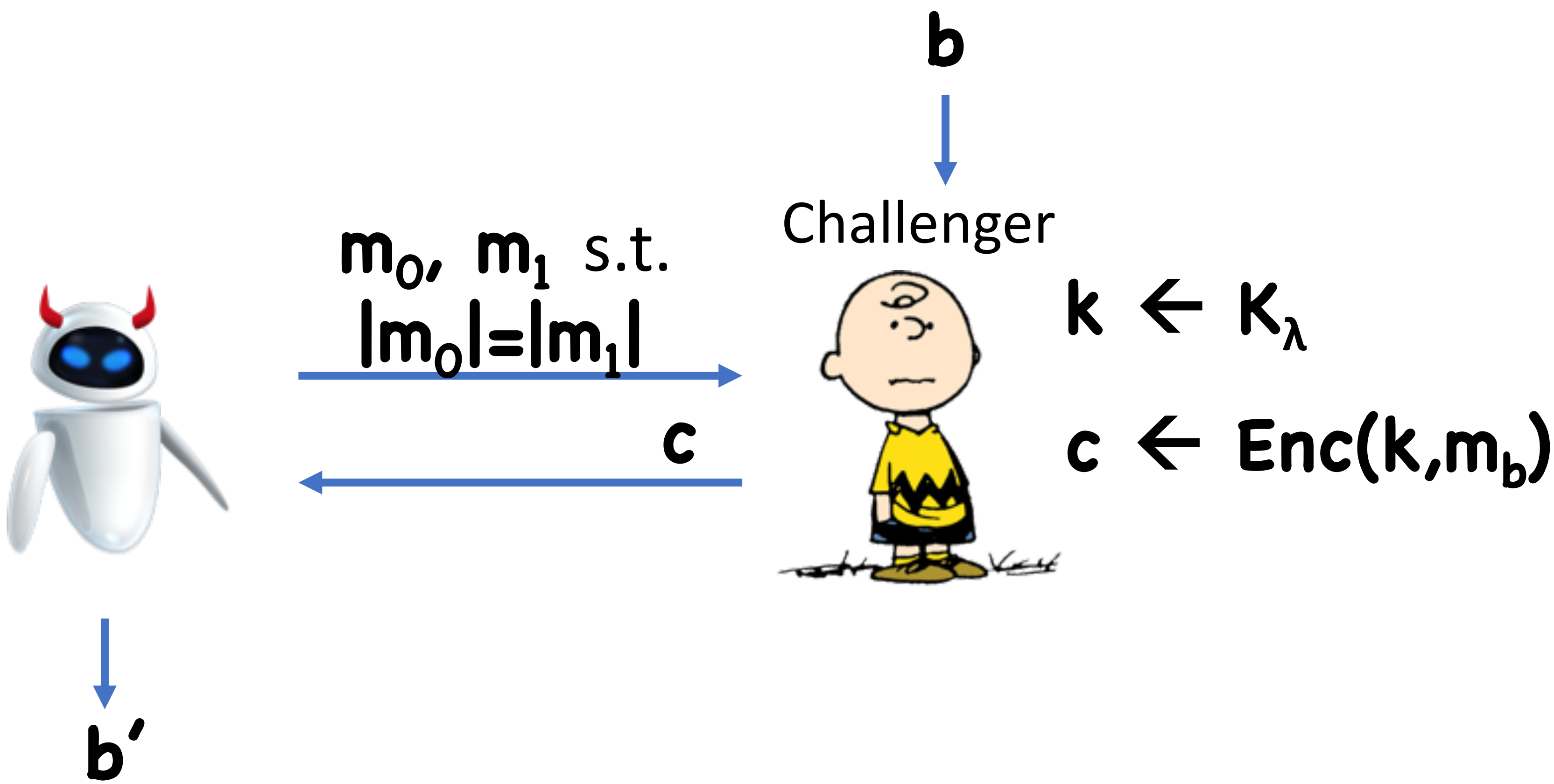
Using PRFs to Build Encryption

So far, scheme had fixed-length messages

- Namely, $\mathbf{M}_\lambda = \mathbf{Y}_\lambda$

Now suppose we want to handle arbitrary-length messages

Security for Arbitrary-Length Messages



IND-Exp$_b$(**, λ)**

Theorem: Given any CPA-secure **(Enc,Dec)** for fixed-length messages (even single bit), it is possible to construct a CPA-secure **(Enc,Dec)** for arbitrary-length messages

Construction

Let **(Enc,Dec)** be CPA-secure for single-bit messages

Enc'(k,m):

For $i=1,\dots,|m|$, run $c_i \leftarrow \mathrm{Enc}(k, m_i)$

Output $(c_1, \dots, c_{|m|})$

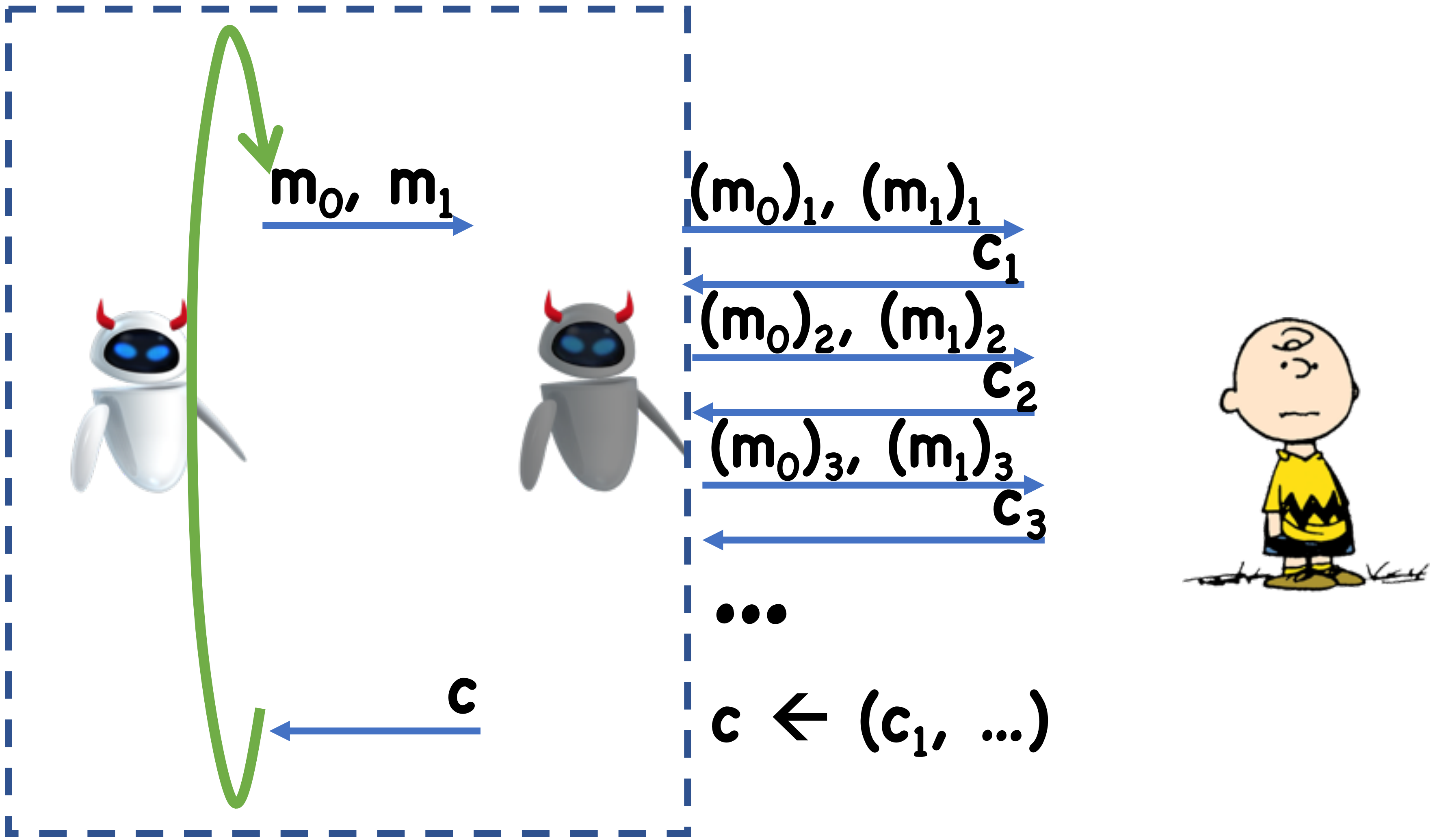
Dec'(k, $(c_1, \dots, c_l)$):

For $i=1,\dots, l$, run $m_i \leftarrow \mathrm{Dec}(k, c_i)$

Output $m = m_1 m_2 \dots, m_l$

Theorem: If **(Enc,Dec)** is LoR secure, then **(Enc',Dec')** is LoR secure

Proof (sketch)



Better Constructions Using PRFs

In PRF-based construction, encrypting single bit requires $\boldsymbol{\lambda+1}$ bits

$\Rightarrow$ encrypting $\boldsymbol{l}$-bit message requires $\boldsymbol{\approx\lambda l}$ bits

Ideally, ciphertexts would have size $\boldsymbol{\approx\lambda+l}$

Solution 1: Add PRG/Stream Cipher

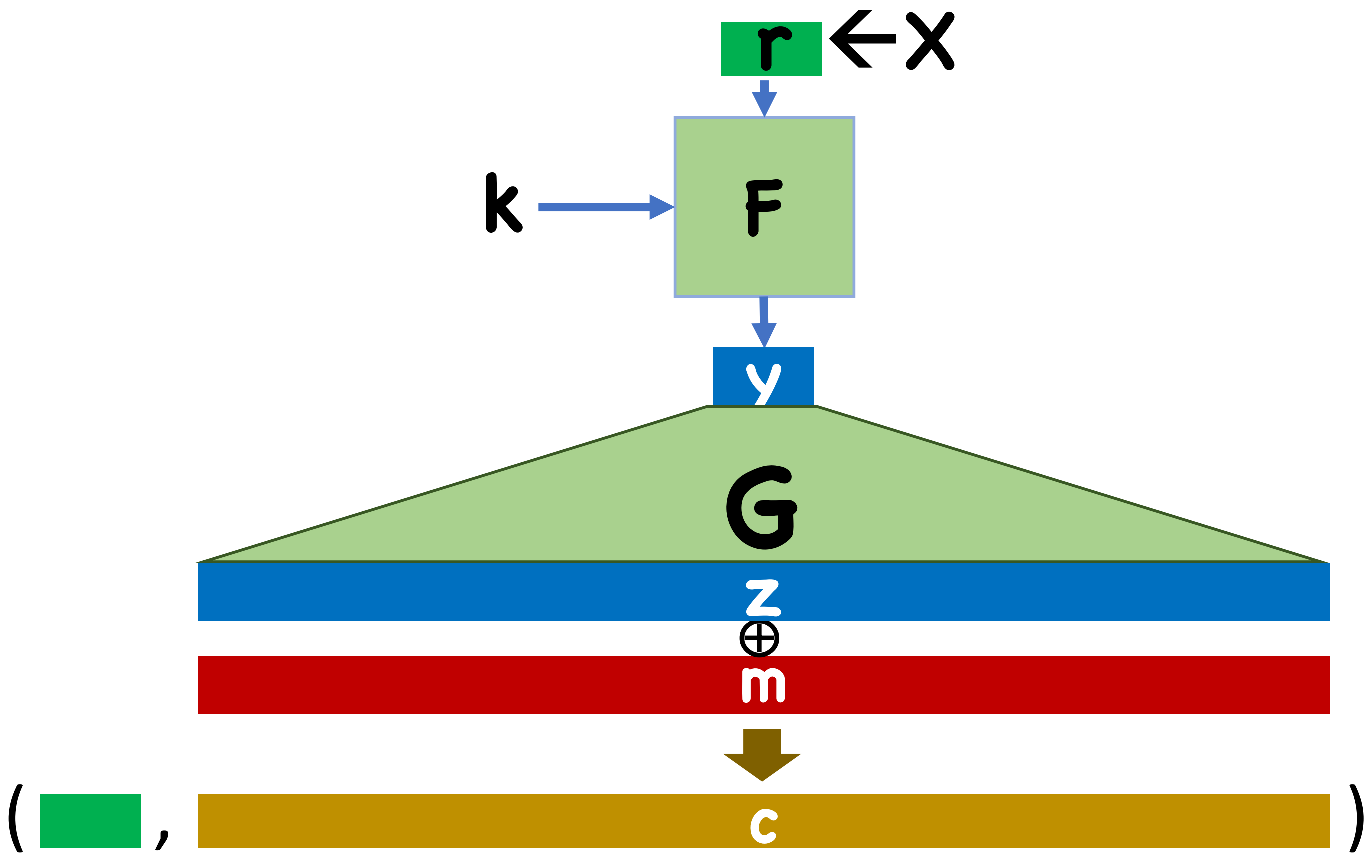
Enc(k, m):

- Choose random $r \leftarrow X$
- Compute $y \leftarrow F(k,r)$
- Get $|m|$ pseudorandom bits $z \leftarrow G(y)$
- Compute $c \leftarrow z \oplus m$
- Output (r,c)

Dec(k, (r,c)):

- Compute $y' \leftarrow F(k,r)$
- Compute $z' \leftarrow G(y')$
- Compute and output $m' \leftarrow c \oplus z'$

Solution 1: Add PRG/Stream Cipher



Solution 2: Counter Mode

Enc(k, m):

- Choose random $r \leftarrow \{0,1\}^{\lambda/2}$
- For $i=1,\ldots,|m|$,
  - Compute $y_i \leftarrow F(k, r||i)$
  - Compute $c_i \leftarrow y_i \oplus m_i$
- Output (r,c) where $c=(c_1,\ldots,c_{|m|})$

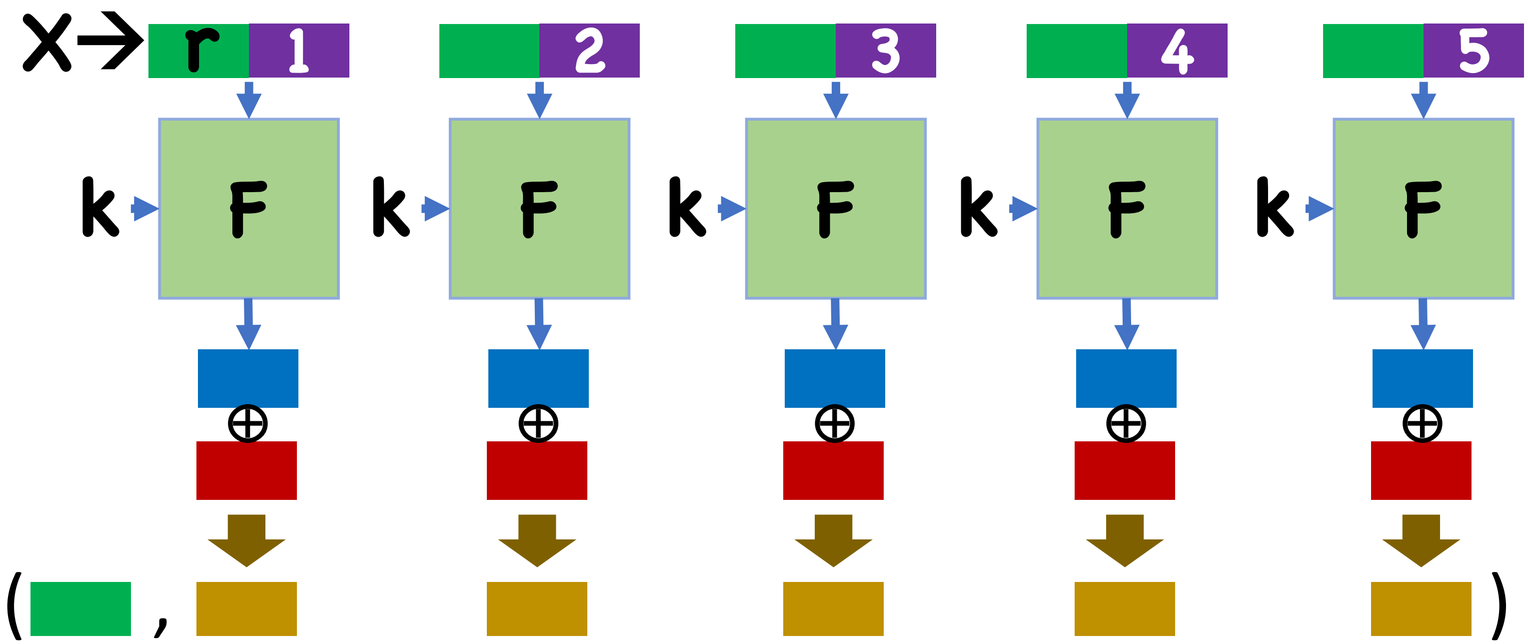
Write **i** as **λ/2**-bit string

Dec(k, (r,c)):

- For $i=1,\ldots,l$,
  - Compute $y_i \leftarrow F(k, r||i)$
  - Compute $m_i \leftarrow y_i \oplus c_i$
- Output $m=m_1,\ldots,m_l$

Handles any message of length at most $2^{\lambda/2}$

Solution 2: Counter Mode



Summary

PRFs = “random looking” functions

Can be used to build security for arbitrary length/number of messages with stateless scheme

Next time: block ciphers and other “modes” of operation