COS433/Math 473: Cryptography

Mark Zhandry

Princeton University

Spring 2017

Announcements

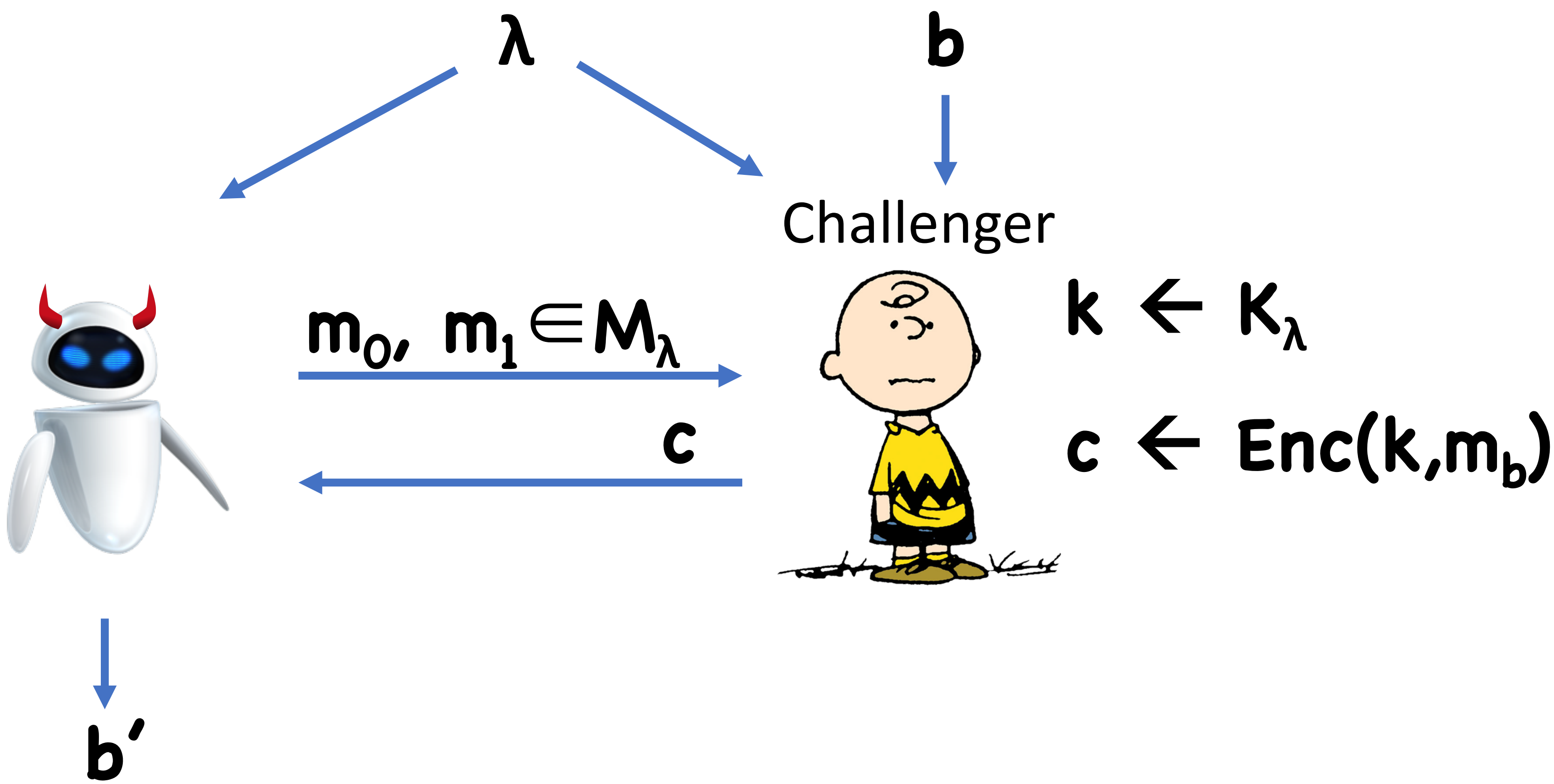
Homework 3 up

Last Time

Stream Ciphers

Design of PRGs

Encryption Security Experiment



$IND\text{-}Exp_b($  $, \lambda)$

Encryption Security Definition

Definition: (Enc, Dec) has **ciphertext indistinguishability** if, for all probabilistic polynomial time (PPT)  , there exists a negligible function **ε** such that

$\Big|\ \Pr[1 \leftarrow \text{IND-Exp}_0($  $, \lambda)\ ]$

$\quad - \Pr[1 \leftarrow \text{IND-Exp}_1($  $, \lambda)\ ]\ \Big| \leq \varepsilon(\lambda)$

This Time

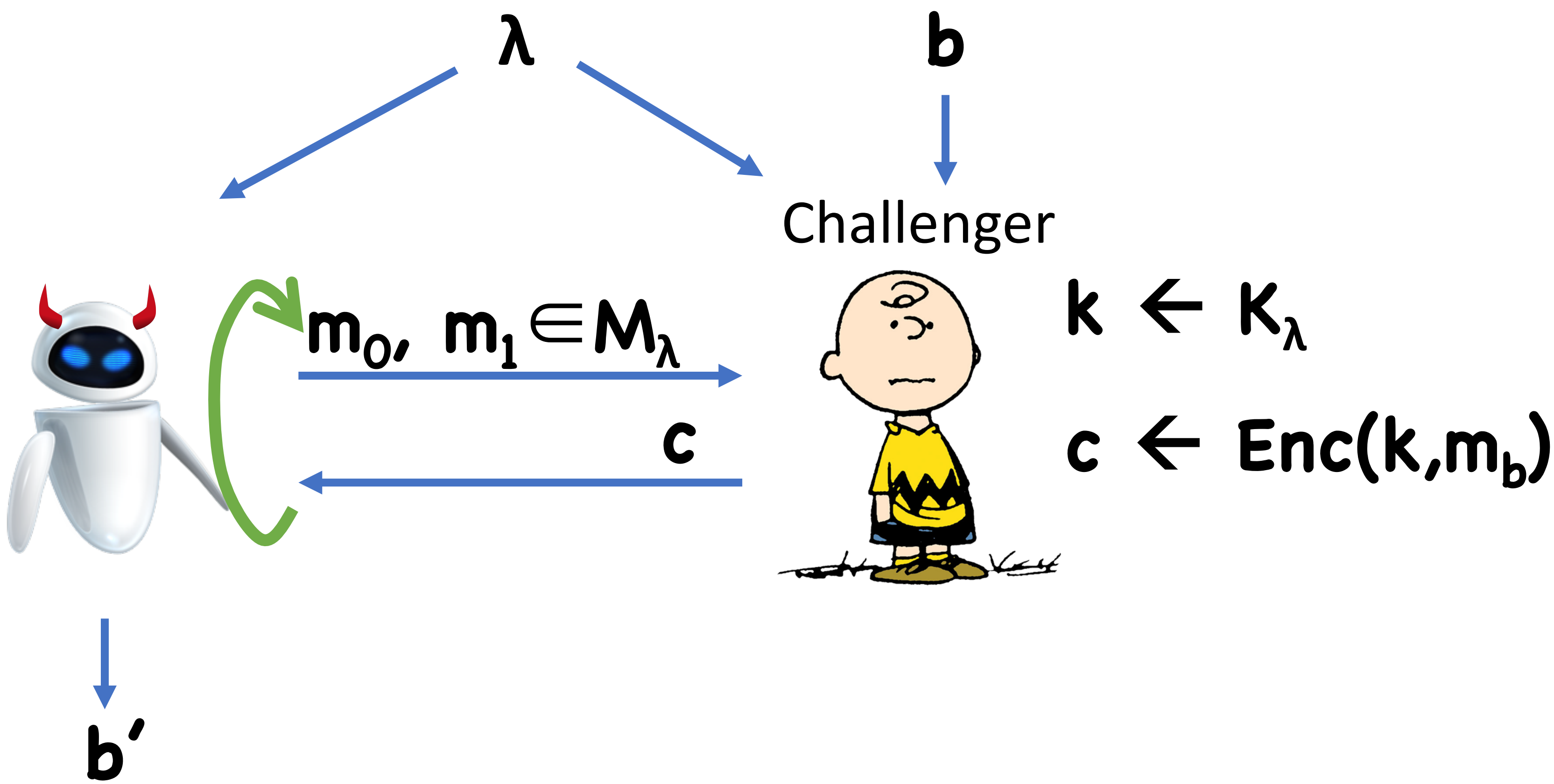
Multiple message security

Stateless encryption

Pseudorandom Functions

Multiple Message Security

Left-or-Right Experiment



LoR-Exp$_b$( **, λ)**

LoR Security Definition

Definition: (Enc, Dec) has **Left-or-Right indistinguishability** if, for all probabilistic polynomial time (PPT)  , there exists a negligible function **ε** such that

| Pr[1 $\leftarrow$ LoR-Exp$_0$(, λ)]

− Pr[1 $\leftarrow$ LoR-Exp$_1$(, λ)] | $\leq$ ε(λ)

Alternate Notion: CPA Security

What if adversary can additionally learn encryptions of messages of her choice?

Examples:

- Midway Island, WWII:
  - US cryptographers discover Japan is planning attack on a location referred to as “AF”
  - Guess that “AF” meant Midway Island
  - To confirm suspicion, sent message in clear that Midway Island was low on supplies
  - Japan intercepted, and sent message referencing “AF”

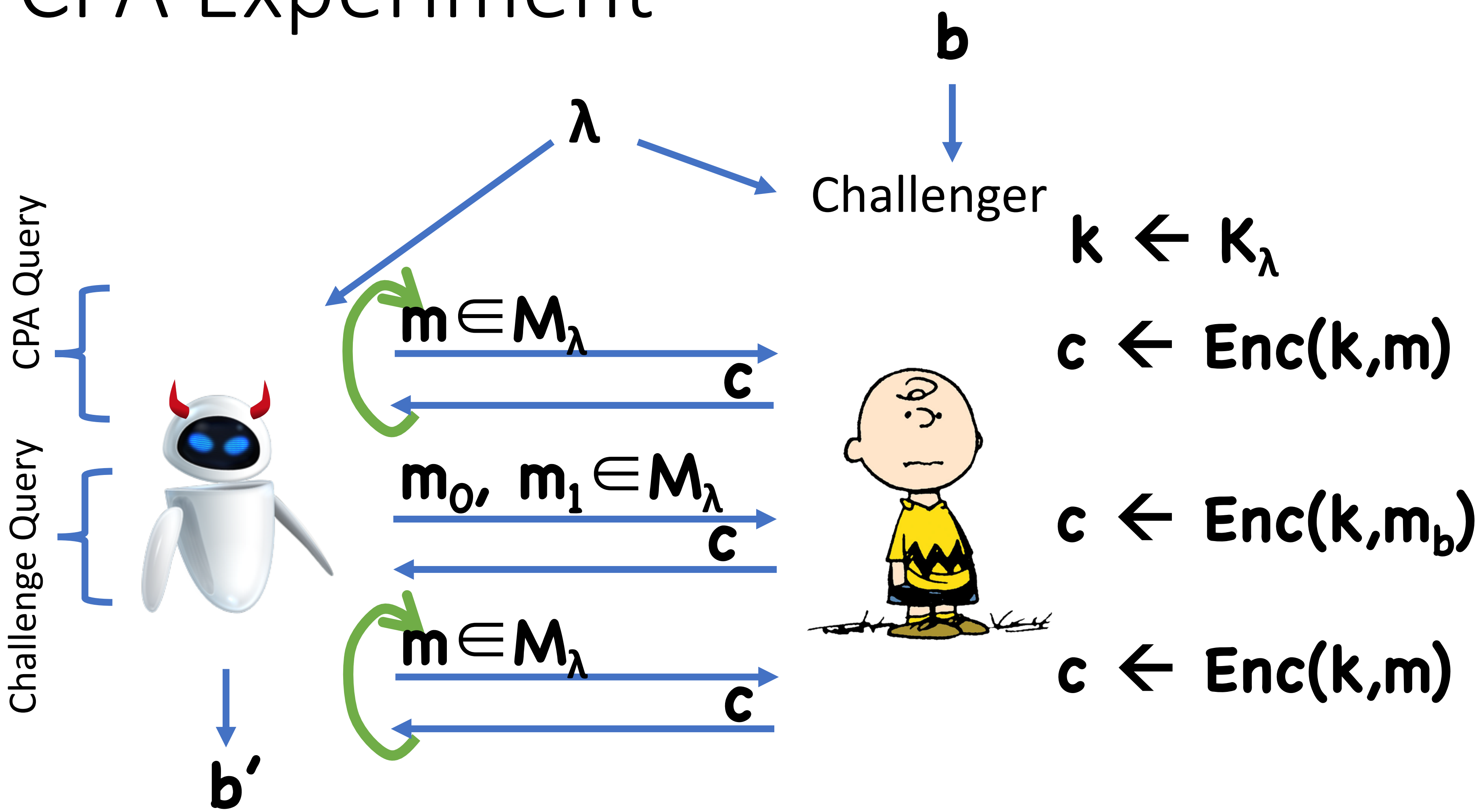
Alternate Notion: CPA Security

What if adversary can additionally learn encryptions of messages of her choice?

Examples:

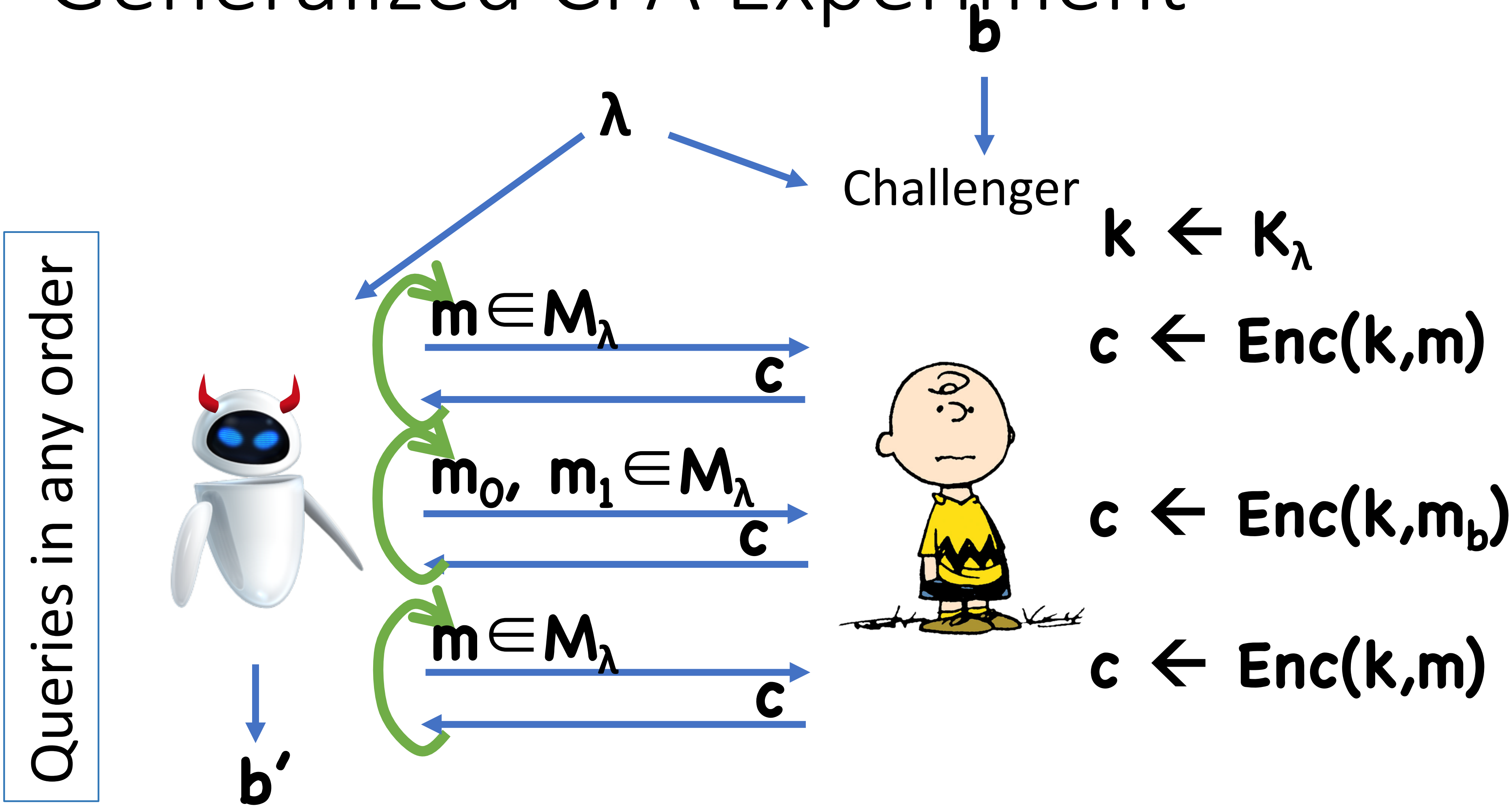
- Land mines, WWII:
    - Allies would lay mines at specific locations
    - Wait for Germans to discover mine
    - Germans would broadcast warning message about the mines, encrypted with Enigma
    - Would also send an “all clear” message once cleared

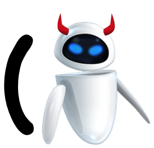
CPA Experiment



CPA-Exp$_b$( , λ)

Generalized CPA Experiment



GCPA-Exp$_b$(, λ)

Equivalences

Theorem:

Left-or-Right indistinguishability

⇕

CPA-security

⇕

Generalized CPA-security

Proof

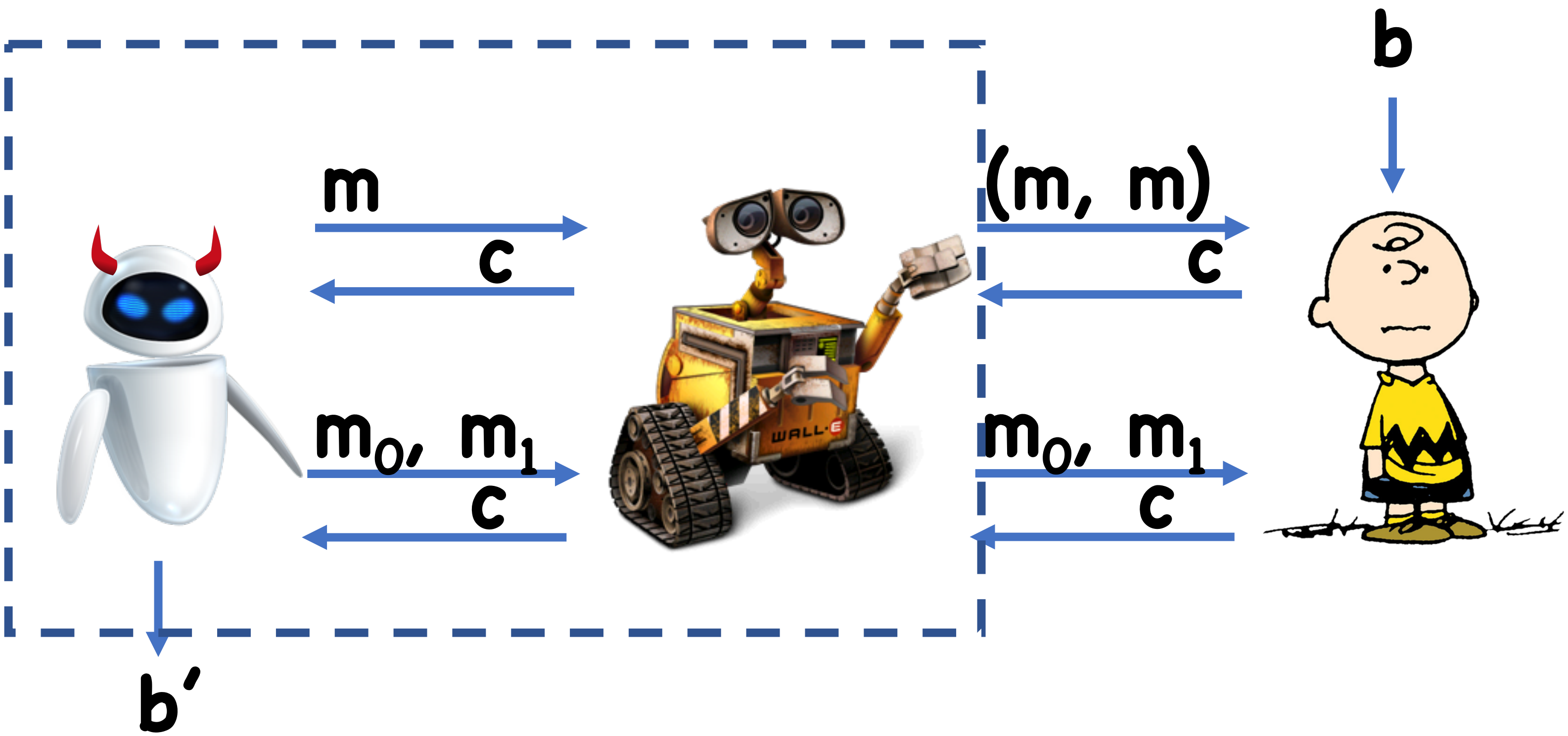
Generalized CPA-security → CPA-security

- Trivial: any adversary in the CPA experiment is also an adversary for the generalized CPA experiment that just doesn't take advantage of the ability to make multiple Left-or-Right queries

Proof

Left-or-Right → Generalized CPA

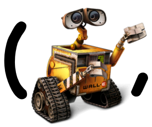
- Assume towards contradiction that we have an adversary  for the generalized CPA experiment
- Construct an adversary  that runs  as a subroutine, and breaks the Left-or-Right indistinguishability

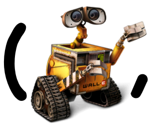


Pr[1←LoR-Exp$_b$(, λ)] = Pr[1←GCPA-Exp$_b$(, λ)]

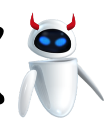
Proof

Left-or-Right → Generalized CPA

$\Big|$ $\Pr[1 \leftarrow \text{LoR-Exp}_0($  $, \lambda)]$

$- \Pr[1 \leftarrow \text{LoR-Exp}_1($  $, \lambda)]$ $\Big|$

$= \Big|$ $\Pr[1 \leftarrow \text{GCPA-Exp}_0($  $, \lambda)]$

$- \Pr[1 \leftarrow \text{GCPA-Exp}_1($  $, \lambda)]$ $\Big| = \varepsilon(\lambda)$

Proof

(regular) CPA → Left-or-Right

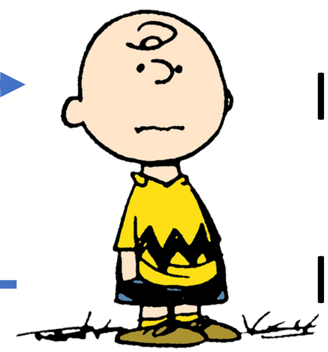
- Assume towards contradiction that we have an adversary  for the LoR experiment
- Hybrids!

Hybrid **i**:

$\mathbf{m_0, \ m_1}$ →

← $\mathbf{c}$

↓ $\mathbf{b'}$



$\mathbf{k \leftarrow K_\lambda}$

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

$\mathbf{c \leftarrow Enc(k,m_0)}$

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

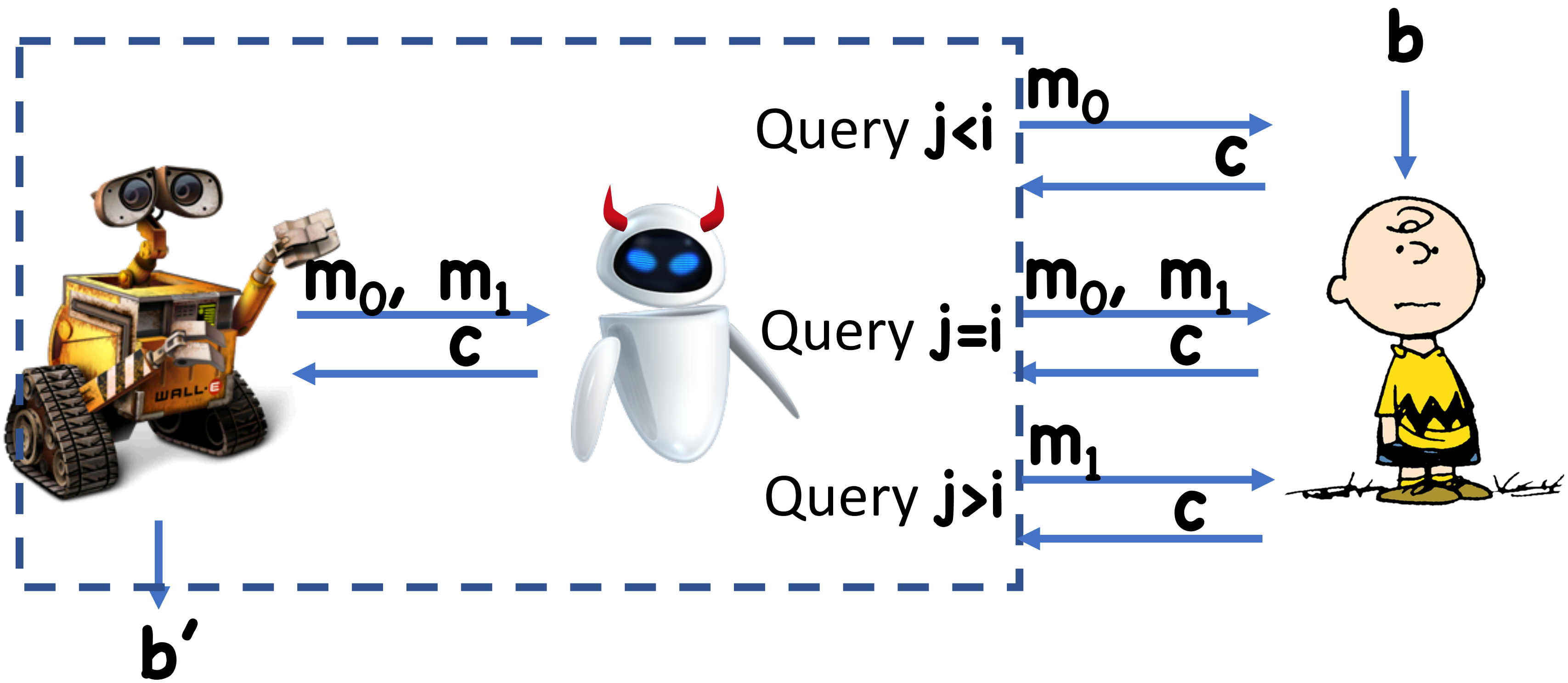
$\mathbf{c \leftarrow Enc(k,m_1)}$

Proof

(regular) CPA → Left-or-Right

- Hybrid **0** is identical to **LoR–Exp**$_1$(, λ)
- Let **t** be maximum number of queries by 
  (**t** $\leq$ running time of  $\leq$ polynomial)
- Hybrid **t** is identical to **LoR–Exp**$_0$(, λ)

- We know that  distinguishes Hybrid **t** and Hybrid **0** with advantage ε
  $\Rightarrow$ $\exists$ **i** s.t.  distinguishes Hybrid **i** and Hybrid **i–1** with advantage ε/t



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/t

Equivalences

Theorem:

Left-or-Right indistinguishability

$\Uparrow$

$\Downarrow$ **CPA-security**

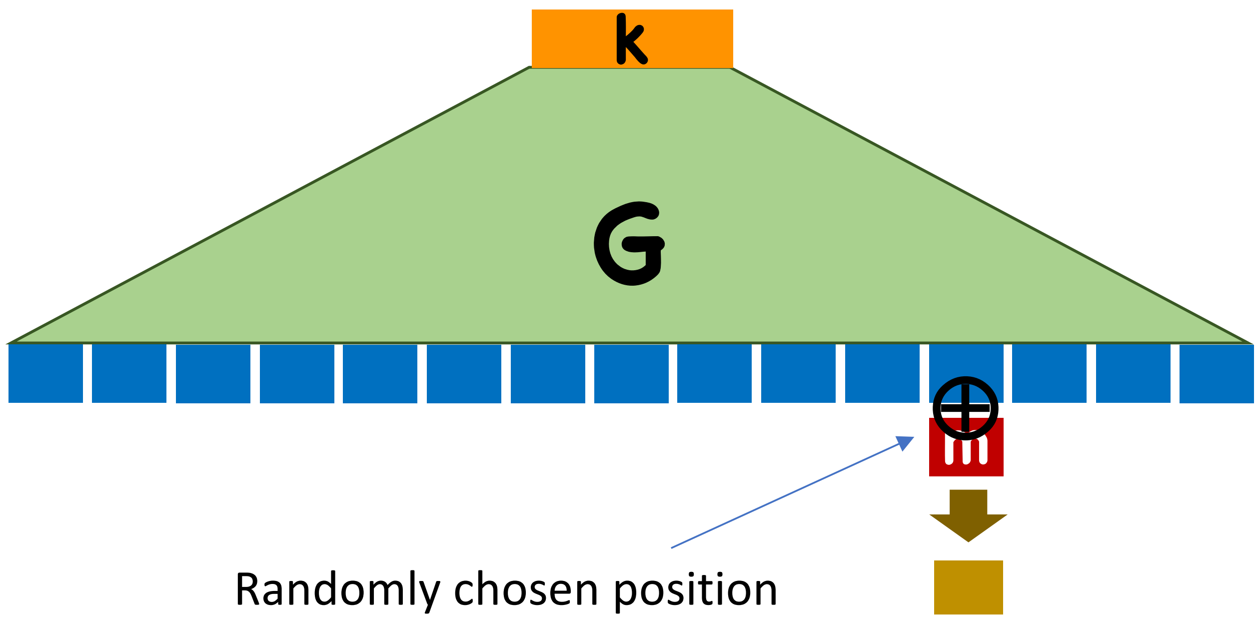
$\Uparrow$

Generalized CPA-security

Therefore, you can use whichever notion you like best

Constructing CPA-secure Encryption

Starting point: A simple randomized encryption scheme from PRGs:



Analysis

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

Pr[Collision] ≤ $q^2/2n$, where

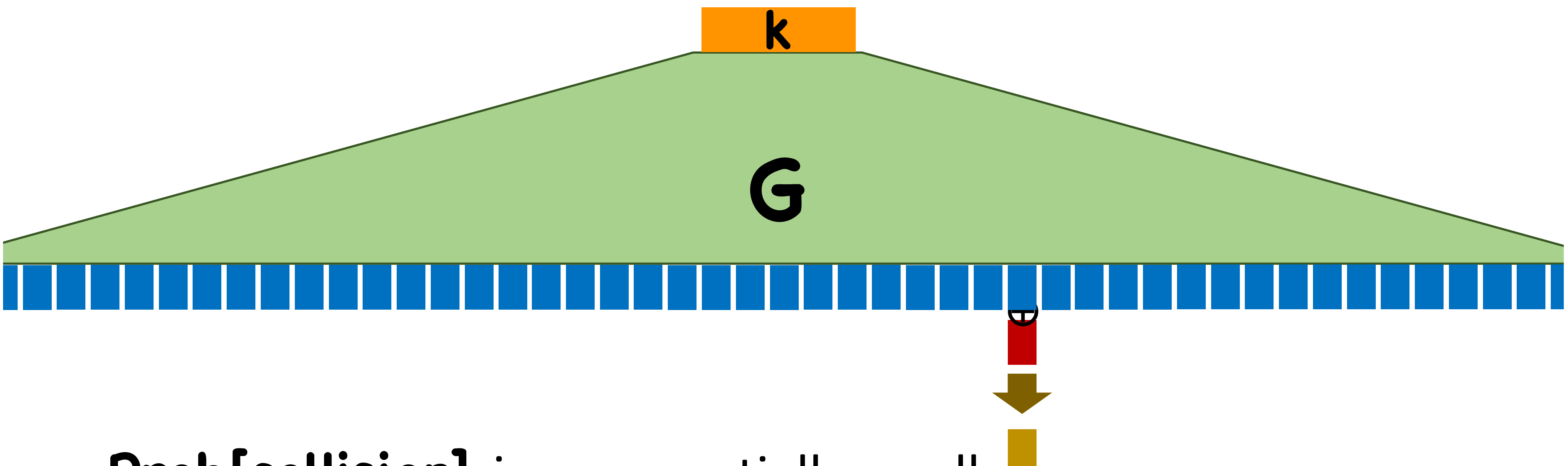
- **q =** number of messages encrypted
- **n =** number of blocks

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

For small **q**, we get small, but non-negligible security

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 $F(k, x) = G(k)_x$

Pseudorandom Functions

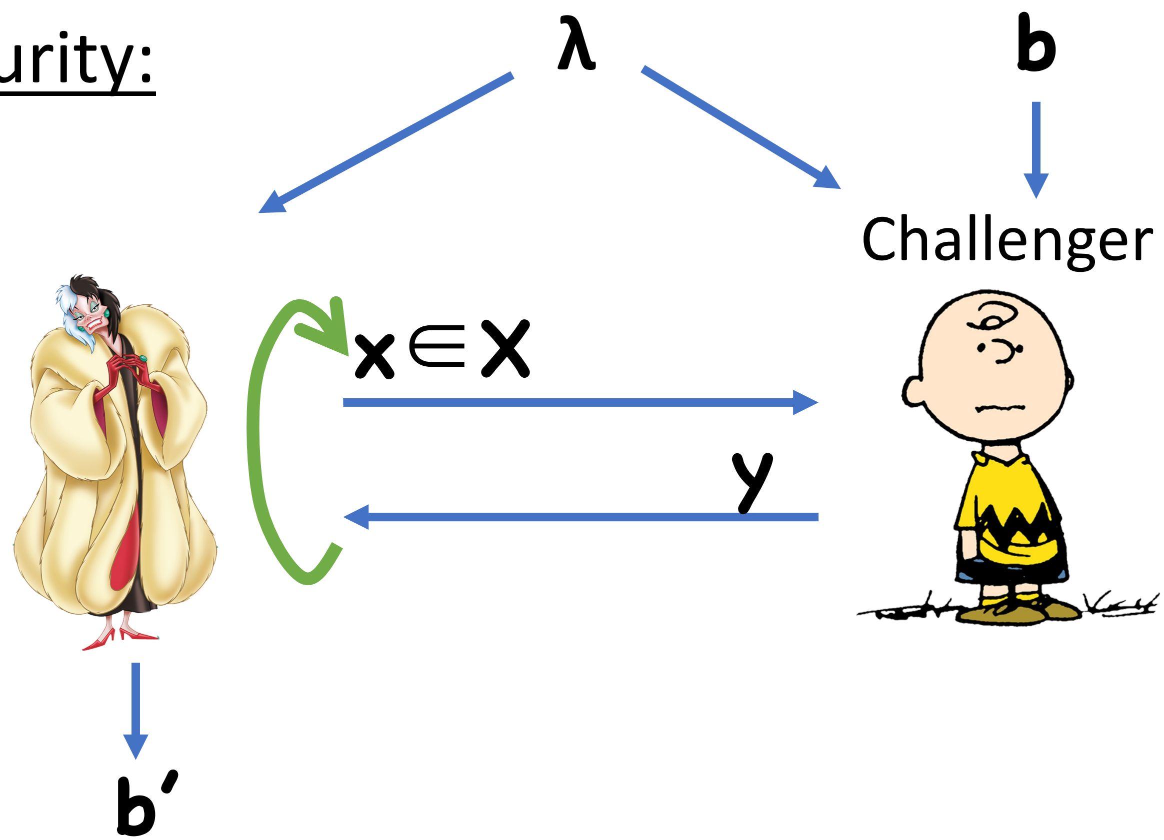
Functions that “look like” random functions

Syntax:

- Key space $\{0,1\}^\lambda$
- Domain X (usually $\{0,1\}^m$, m may depend on λ)
- Co-domain/range Y (usually $\{0,1\}^n$, may depend on λ)
- Function $F:\{0,1\}^\lambda \times X \rightarrow Y$

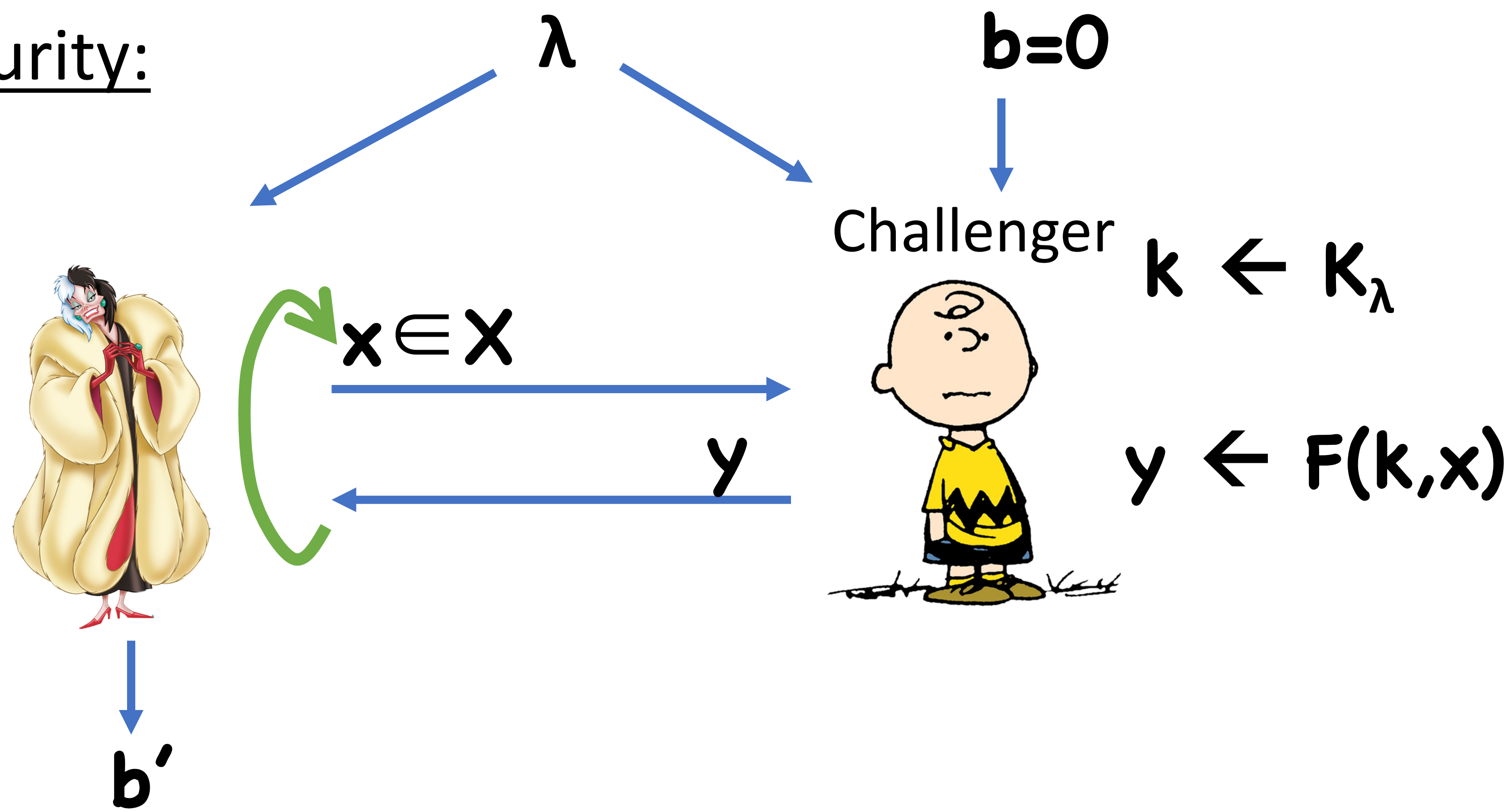
Pseudorandom Functions

<u>Security:</u>



Pseudorandom Functions

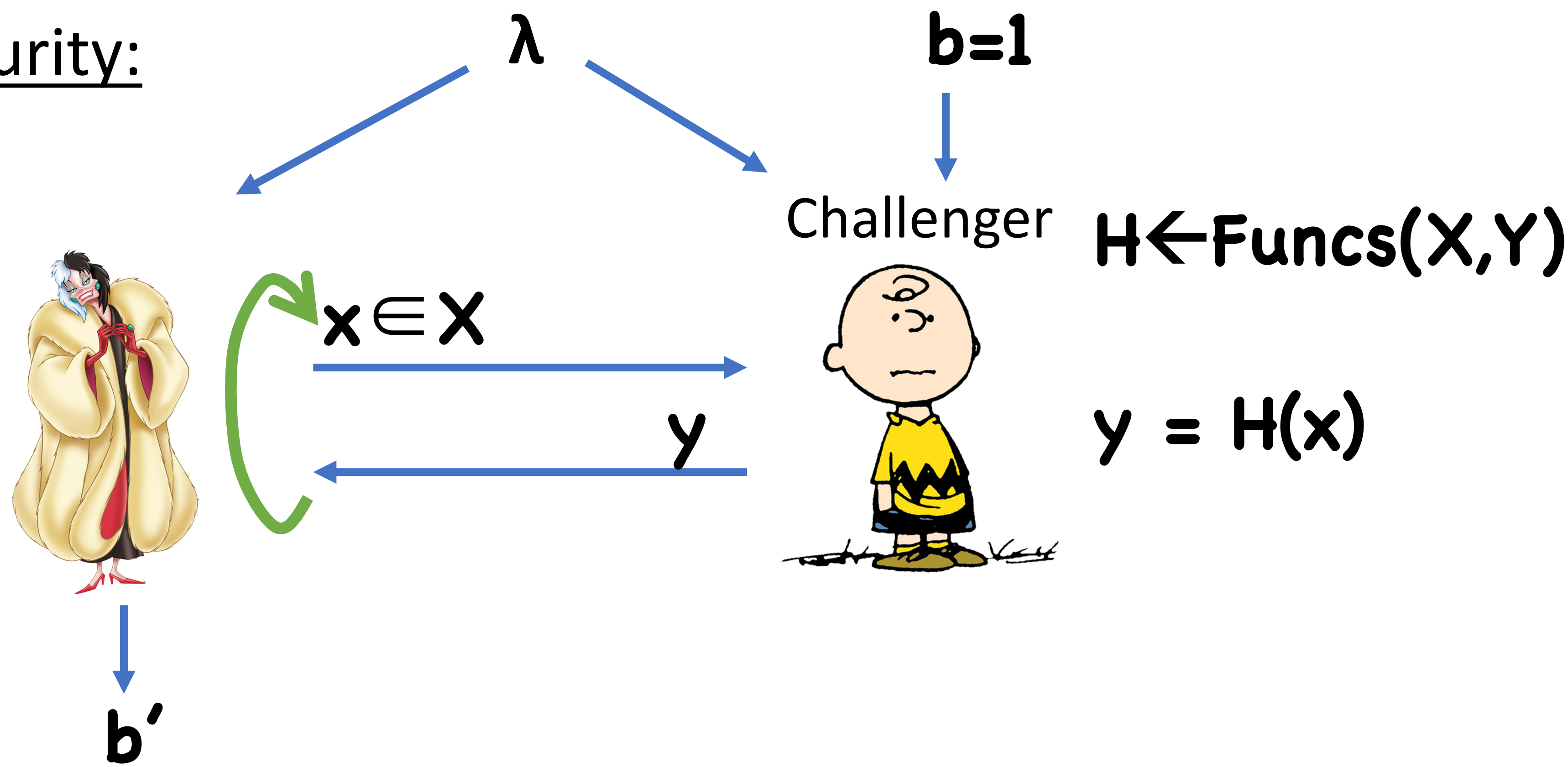
Security:



$\mathbf{PRF\text{-}Exp_0(}$  $\mathbf{,\ \lambda)}$

Pseudorandom Functions

<u>Security:</u>



PRF-Exp$_1$( **, λ)**

PRF Security Definition

Definition: F is a secure PRF if, for all probabilistic polynomial time (PPT)  , there exists a negligible function **ε** such that

| Pr[1←PRF-Exp$_0$( **, λ)]**

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

Using PRFs to Build Encryption

Enc(k, m):

- Choose random $r \leftarrow X$
- Compute $y \leftarrow F(k,r)$
- Compute $c \leftarrow y \oplus m$
- Output (r,c)

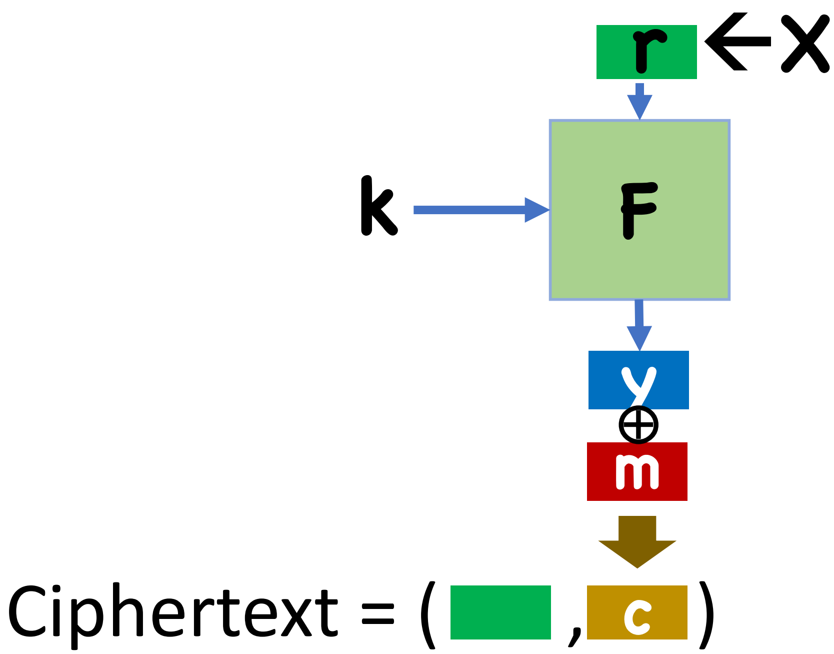
Dec(k, (r,c)):

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

Correctness:

- $y'=y$ since F is deterministic
- $m' = c \oplus y = y \oplus m \oplus y = m$

Using PRFs to Build Encryption



Security

Theorem: If **F** is a secure PRF and **X** is exponentially large in **λ** (e.g. $X=\{0,1\}^\lambda$) , then **(Enc,Dec)** is CPA-secure

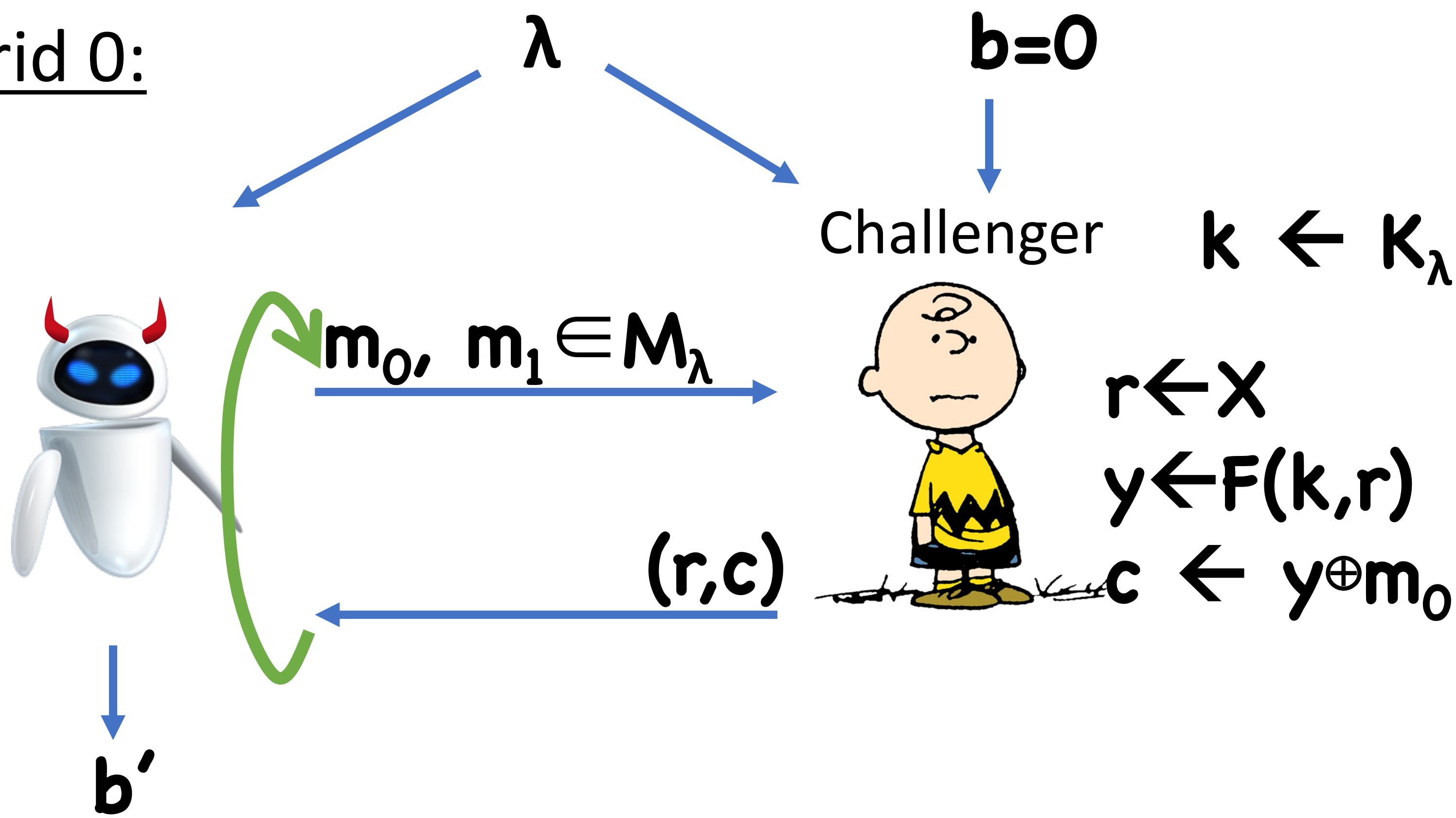
Proof

Assume toward contradiction that there exists a PPT  and non-negligible **ε** such that  has advantage **ε** in breaking **(Enc,Dec)**

Hybrids…

Proof

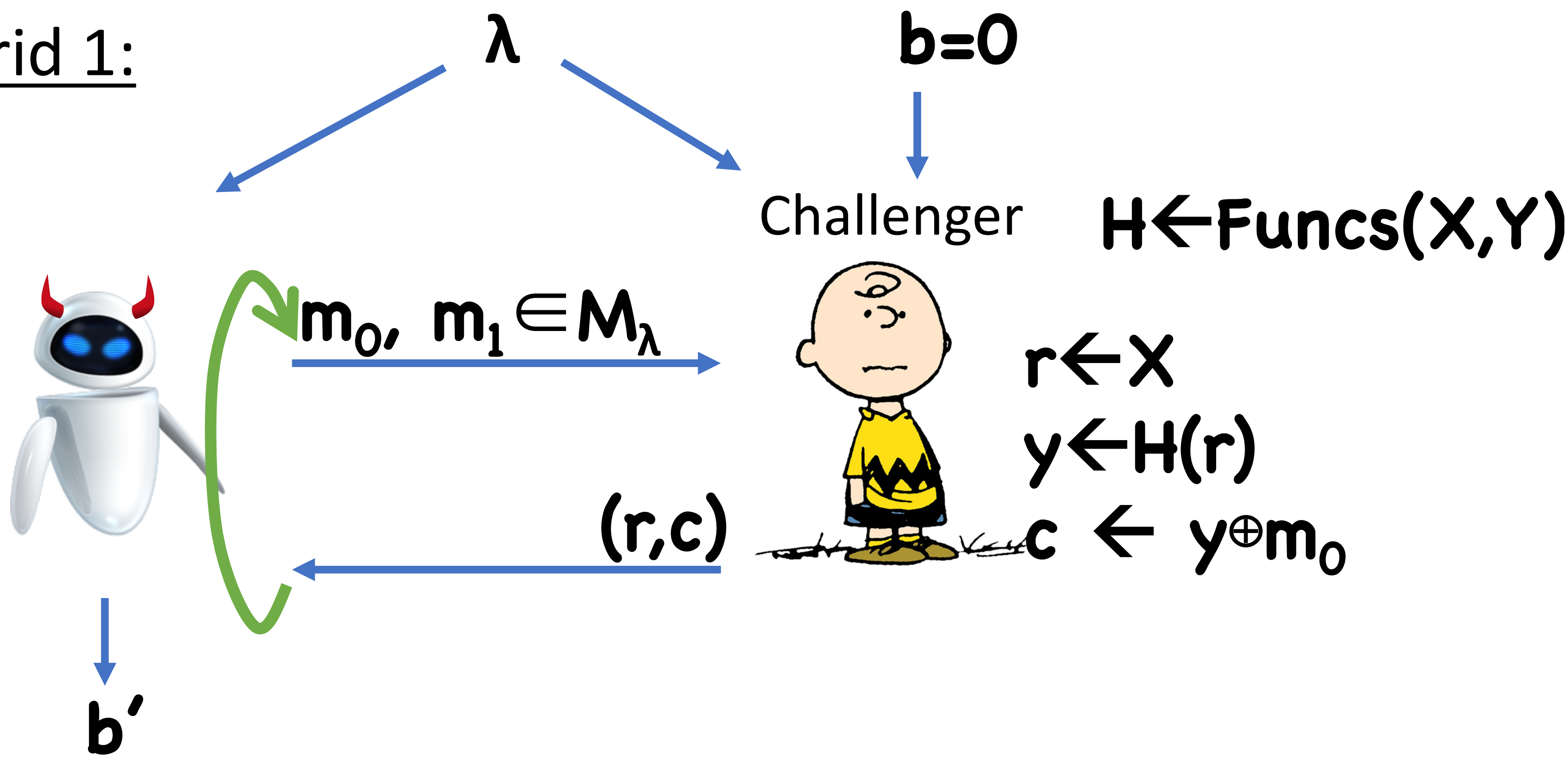
Hybrid 0:



LoR-Exp$_0$( **, λ)**

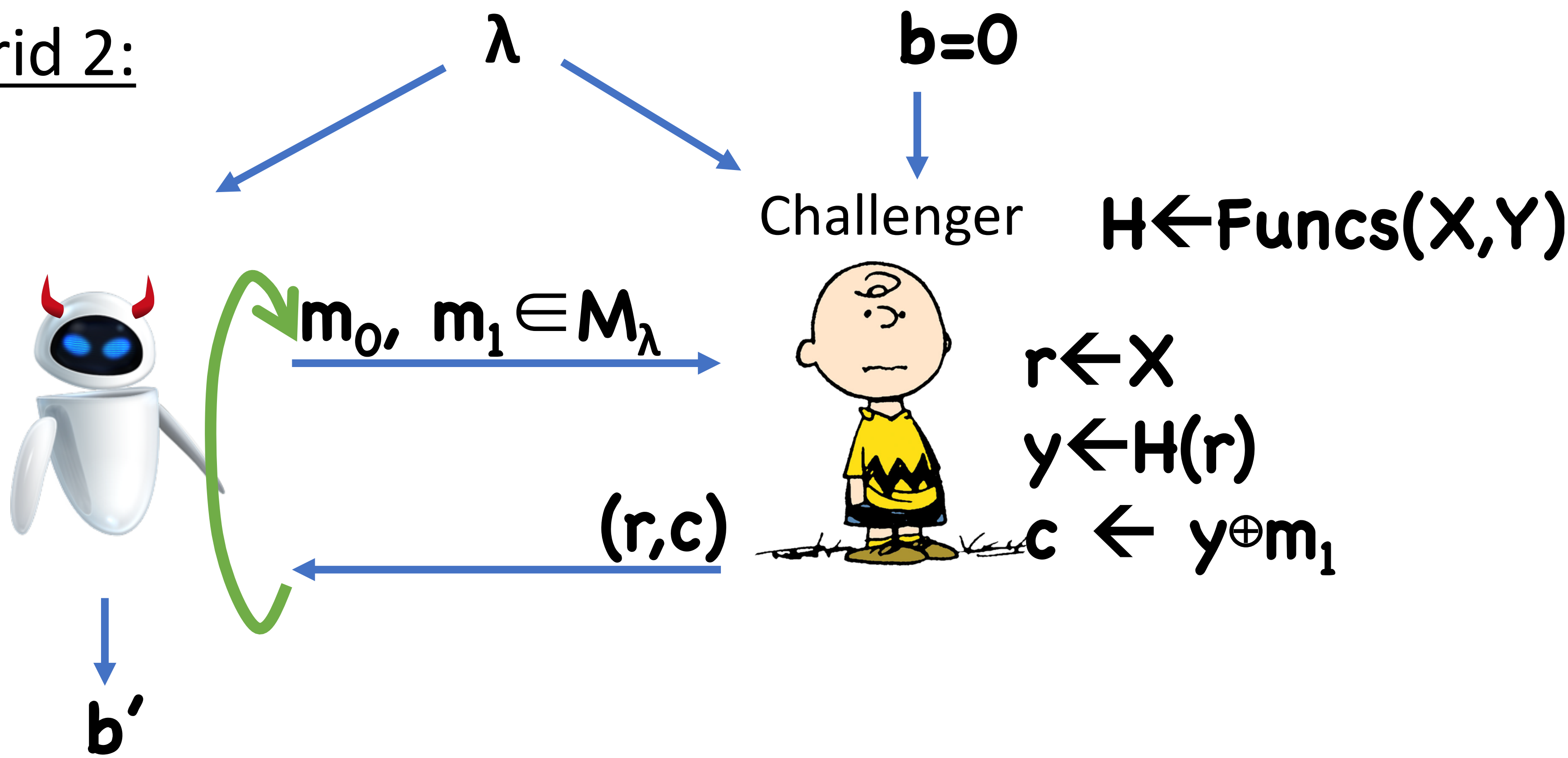
Proof

Hybrid 1:



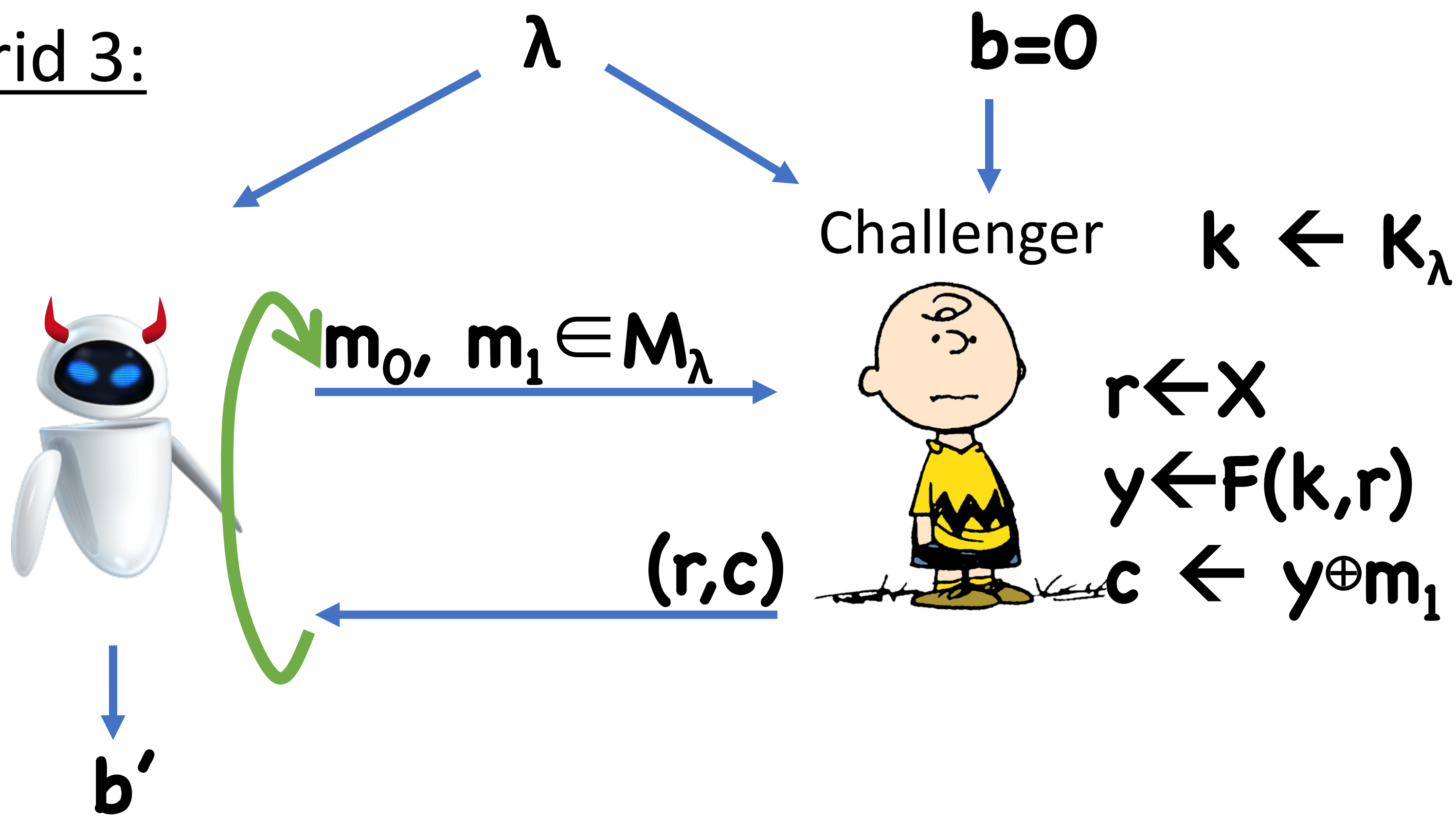
Proof

Hybrid 2:



Proof

Hybrid 3:



LoR-Exp$_1$( **, λ)**

Proof

Assume toward contradiction that there exists a PPT  and non-negligible **ε** such that  has advantage **ε** in breaking **(Enc,Dec)**

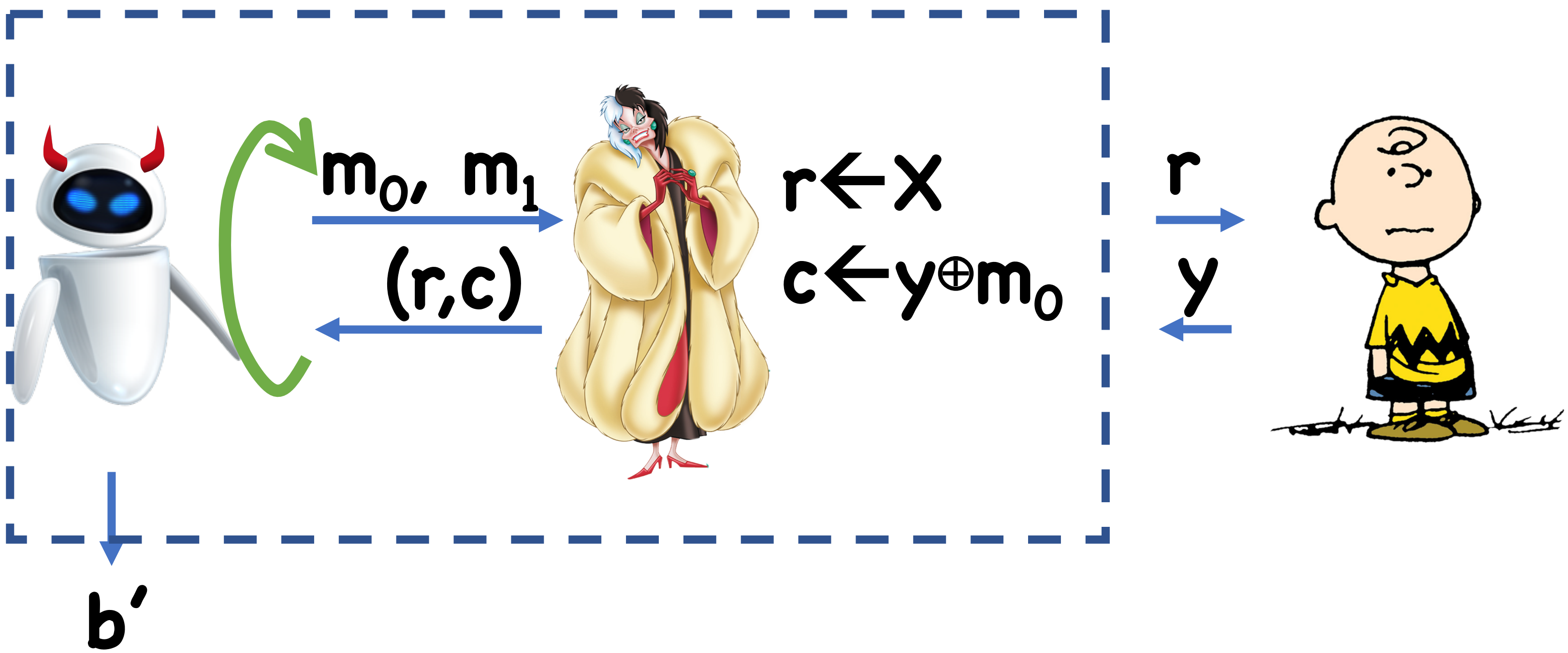
 distinguishes Hybrid 0 from Hybrid 3 with advantage **ε**

$\Rightarrow$ $\exists$**i** such that  distinguishes Hybrid **i-1** from Hybird **i** with advantage **ε/3**

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 **ε/3**

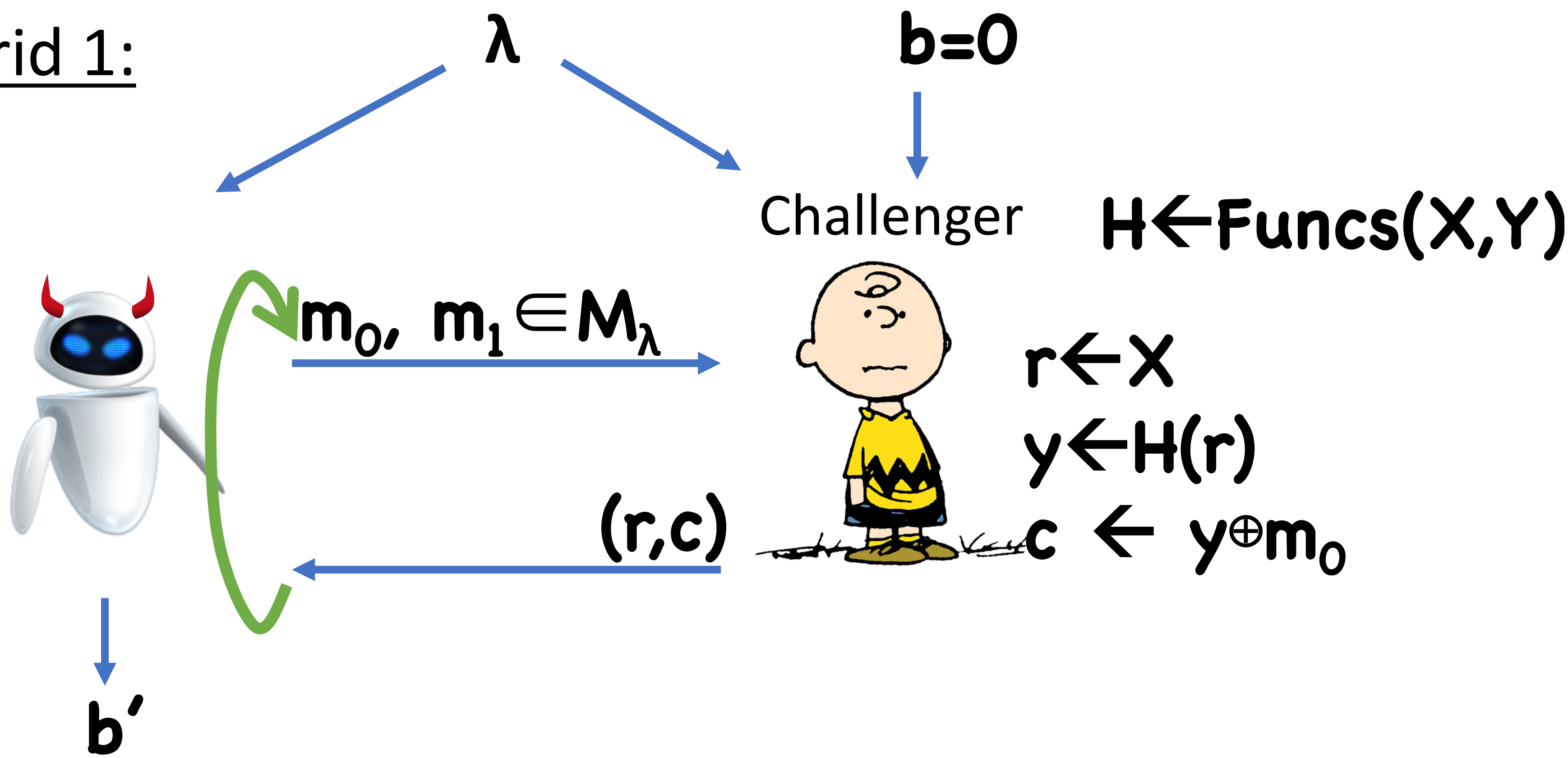
$\Rightarrow$ contradiction

Proof

Suppose  distinguishes Hybrid 1 from Hybrid 2

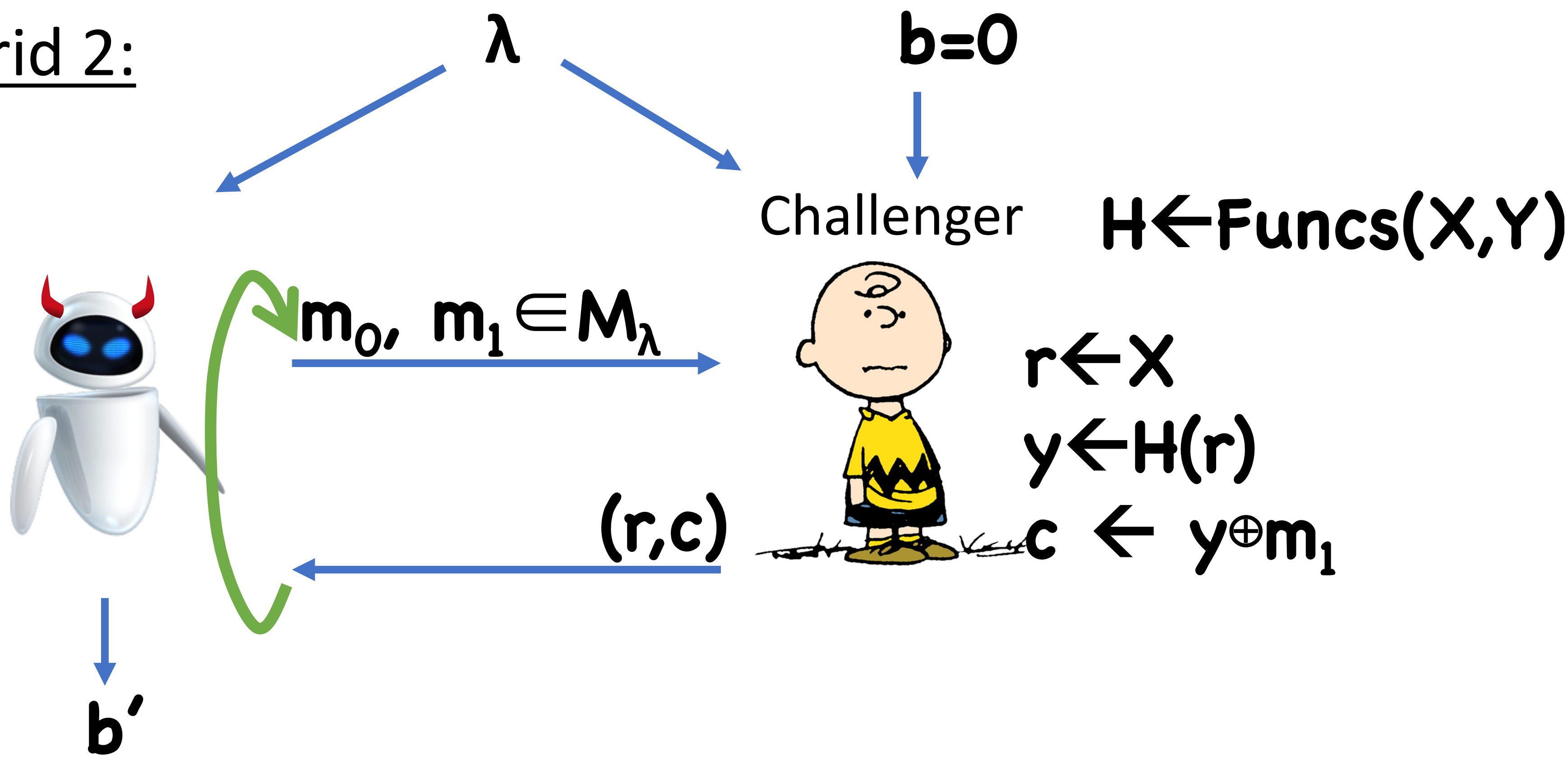
Proof

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 $\leq \Pr[$collision in the r's$]$

$$= \Pr[r^{(1)}=r^{(2)} \text{ or } r^{(1)}=r^{(3)} \text{ or } \dots \text{ or } r^{(1)}=r^{(d+1)} \text{ or } r^{(2)}=r^{(3)} \text{ or } \dots]$$

$$\leq \Pr[r^{(1)}=r^{(2)}] + \Pr[r^{(1)}=r^{(3)}] + \dots + \Pr[r^{(1)}=r^{(t)}] + \Pr[r^{(2)}=r^{(3)}] + \dots$$

$$= (1/|X|) \binom{t}{2}$$

$$\leq t^2/2|X|$$

Exponentially small $\Rightarrow$ contradiction

Proof

Suppose  distinguishes Hybrid 2 from Hybrid 3

Almost identical to the 0/1 case…

Using PRFs to Build Encryption

Enc(k, m):

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

Correctness:

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

Dec(k, (r,c)):

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

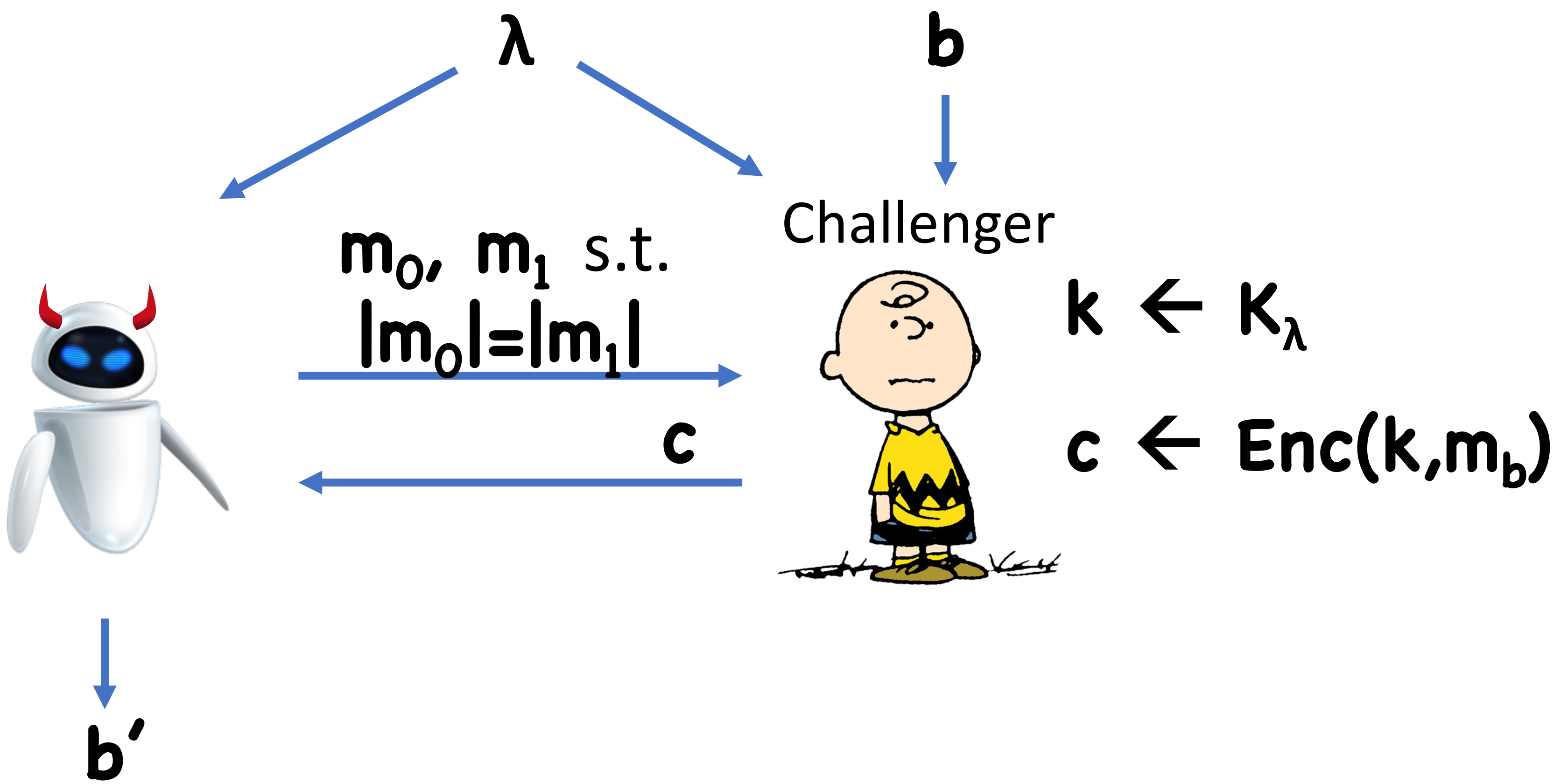
Using PRFs to Build Encryption

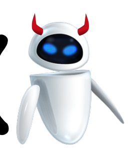
So far, scheme had fixed-length messages

- Namely, $\mathcal{M} = \mathcal{Y}$

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

- If messages are more than single bit, can always pad to message length

Enc'(k,m):

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

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

Dec'(k, ($\mathbf{c_1,\ \ldots,\ c_l}$**)):**

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

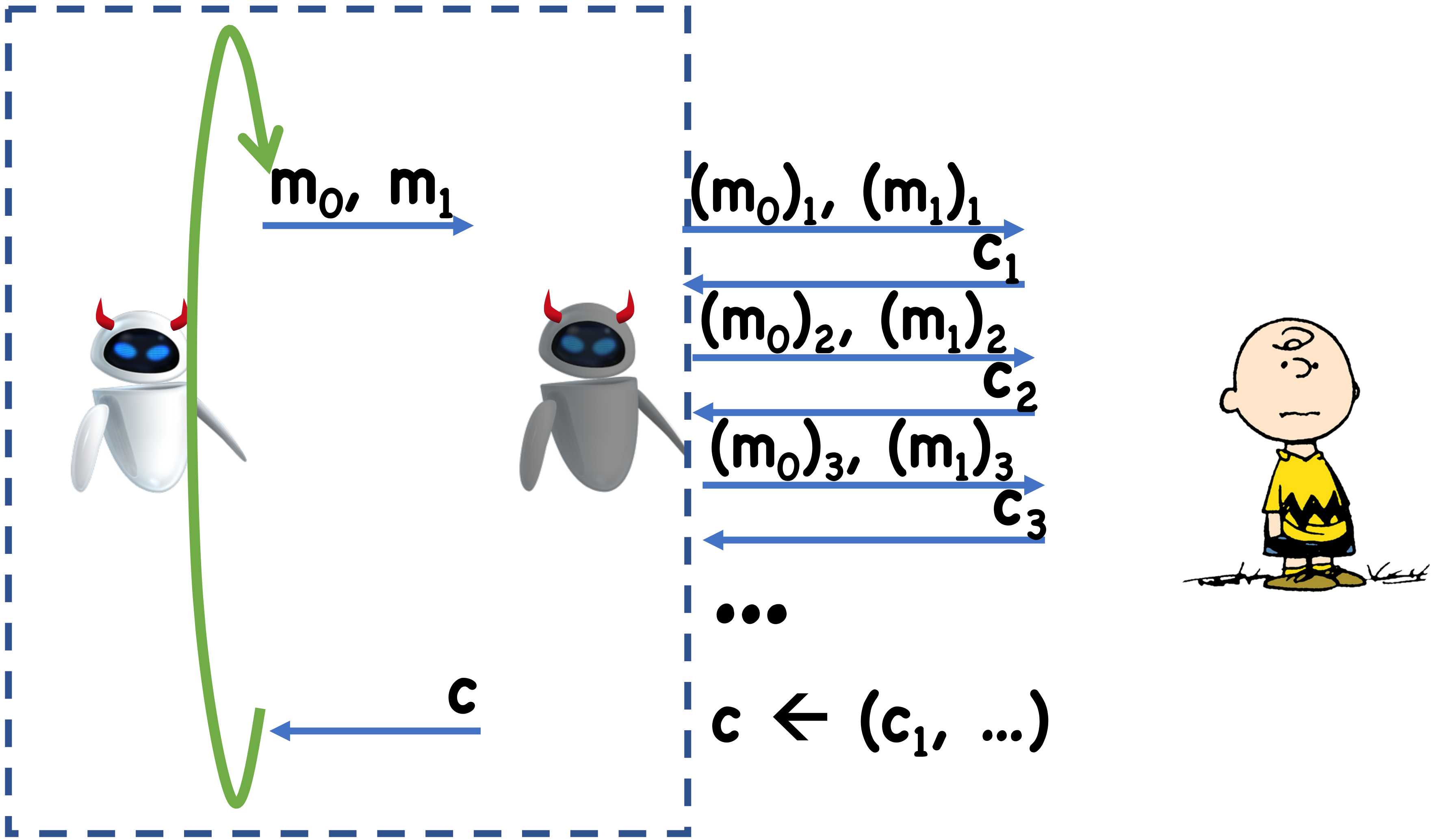
Output $\mathbf{m\ =\ m_1 m_2 \ldots, m_l}$

Proof

Assume toward contradiction that there exists a PPT  and non-negligible **ε** such that  has advantage **ε** in breaking **(Enc',Dec')**

Construct  that has advantage **ε** in breaking **(Enc,Dec)**

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

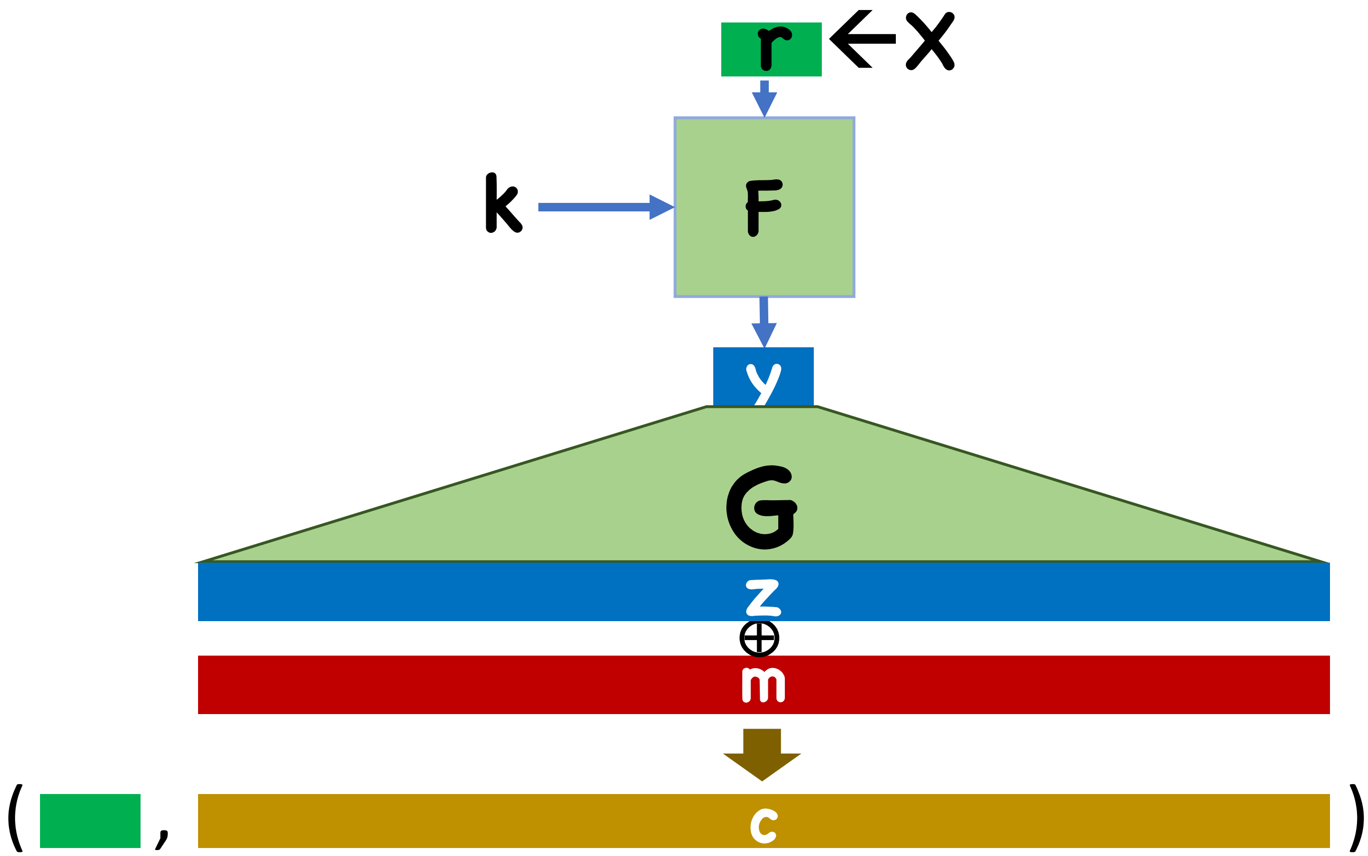
Ideally, ciphertexts would have size $\boldsymbol{\approx\lambda+l}$

Solution 1: Add PRG/Stream Cipher

Enc(k, m):
- Choose random **r←X**
- Compute **y←F(k,r)**
- Get **|m|** pseudorandom bits **z←G(y)**
- Compute **c←z⊕m**
- Output **(r,c)**

Dec(k, (r,c)):
- Compute **y'←F(k,r)**
- Compute **z'←G(y')**
- Compute and output **m'←c⊕z'**

Solution 1: Add PRG/Stream Cipher



Solution 2: Counter Mode

Enc(k, m):

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

Write **i** as **λ/2**-bit string

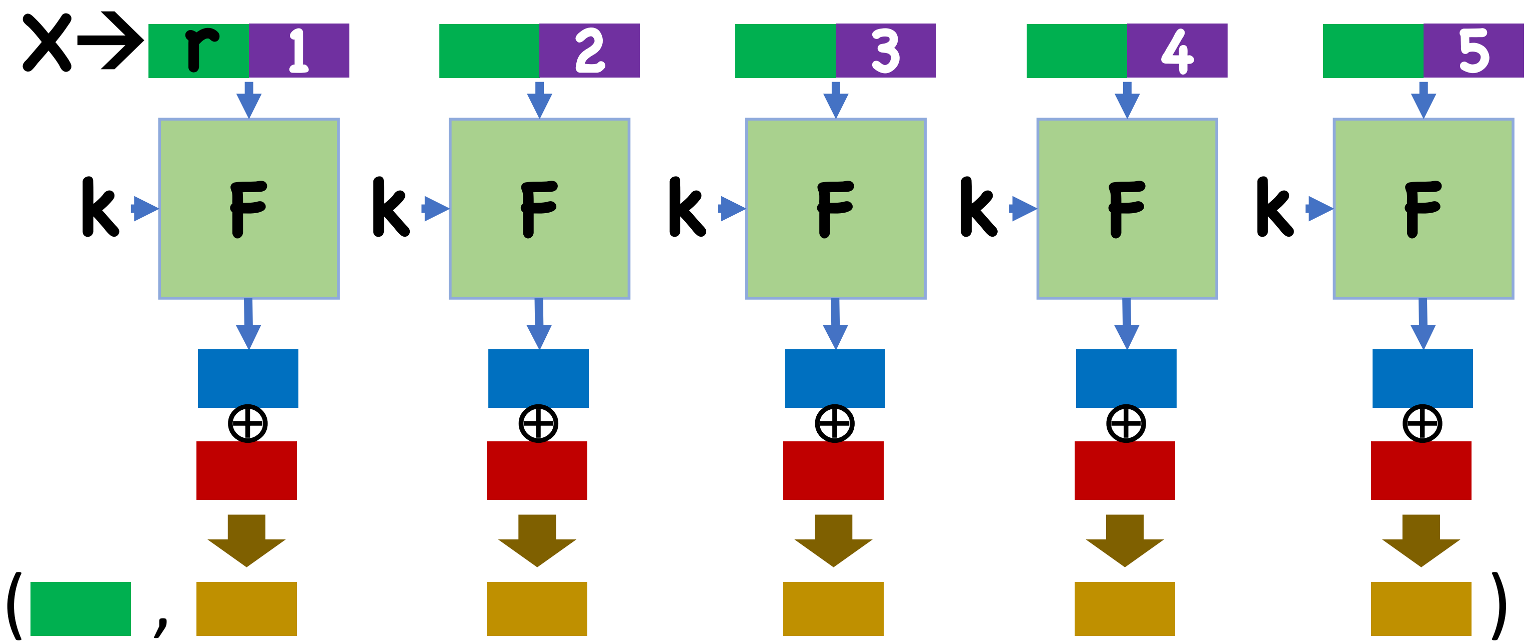
Dec(k, (r,c)):

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

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

- Includes all polynomial-length messages

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

Pseudorandom Permutations/Block Ciphers

- PRFs that are permutations