COS433/Math 473: Cryptography

Mark Zhandry

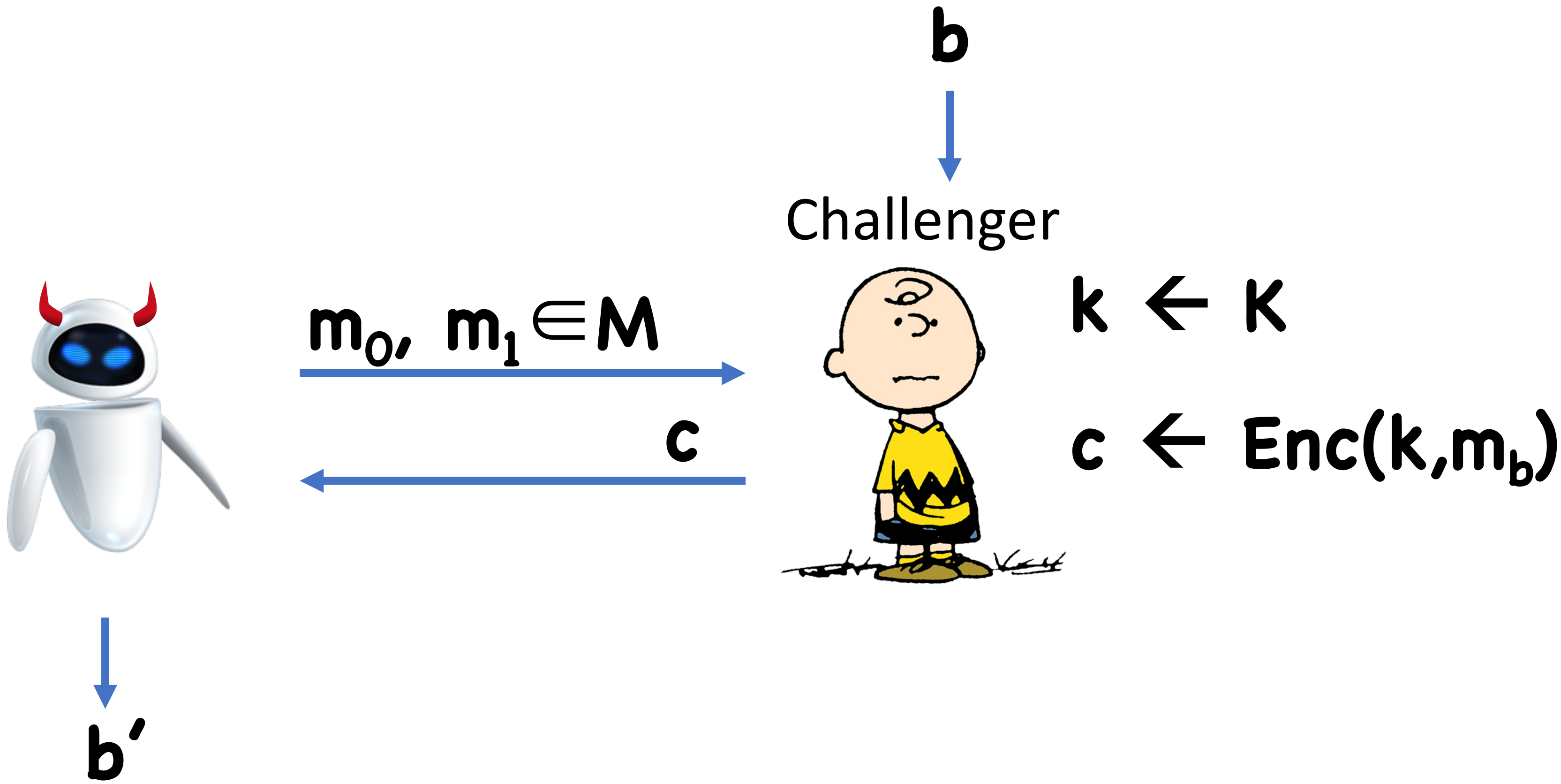
Princeton University

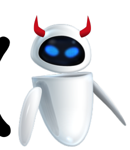
Spring 2017

Previously on COS 433…

Security Experiment/Game

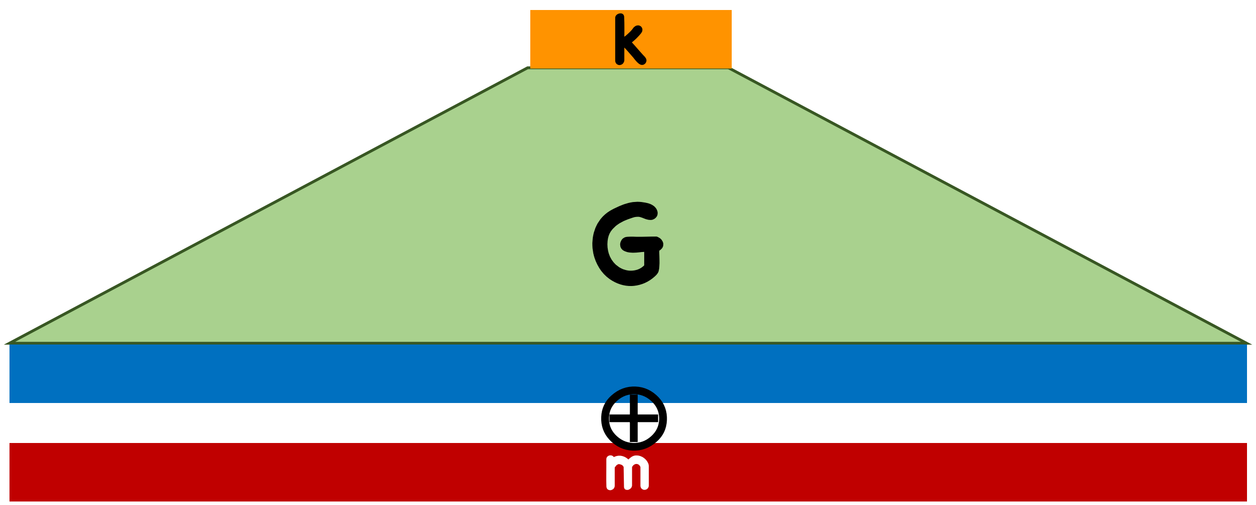
(One-time setting)



IND-Exp$_b$(**)**

Construction with |k| << |m|

Idea: use OTP, but have key generated by some expanding function **G**



What Do We Want Out of **G**?

Definition: $G:\{0,1\}^\lambda \rightarrow \{0,1\}^n$ is a **(t,ε)-secure pseudorandom generator** (PRG) if:

- $n > \lambda$
- G is deterministic
- For all  running in time at most t,

$\Big|\ \Pr[$  $(G(s))=1 : s \leftarrow \{0,1\}^\lambda]$

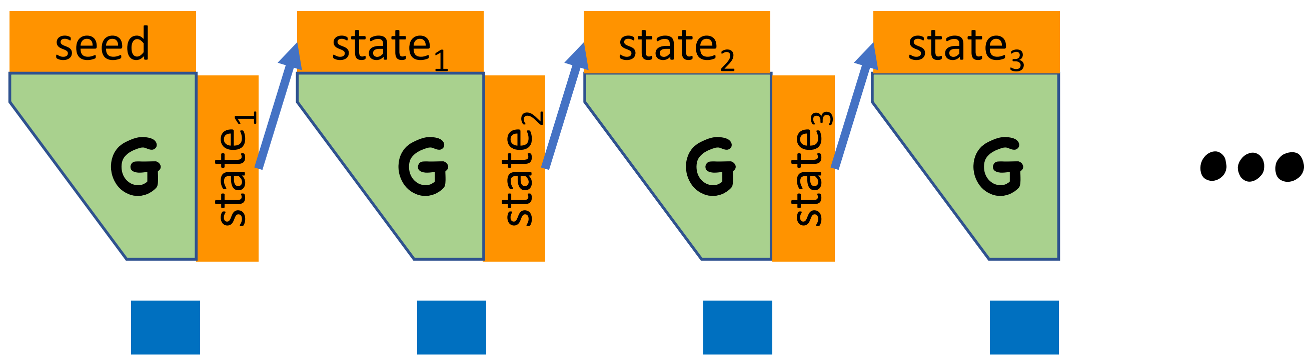
$- \Pr[$  $(x)=1 : x \leftarrow \{0,1\}^n]\ \Big| \leq \varepsilon$

Today

Length Extension for PRGs

Moving to many-time security

Extending the Stretch of a PRG

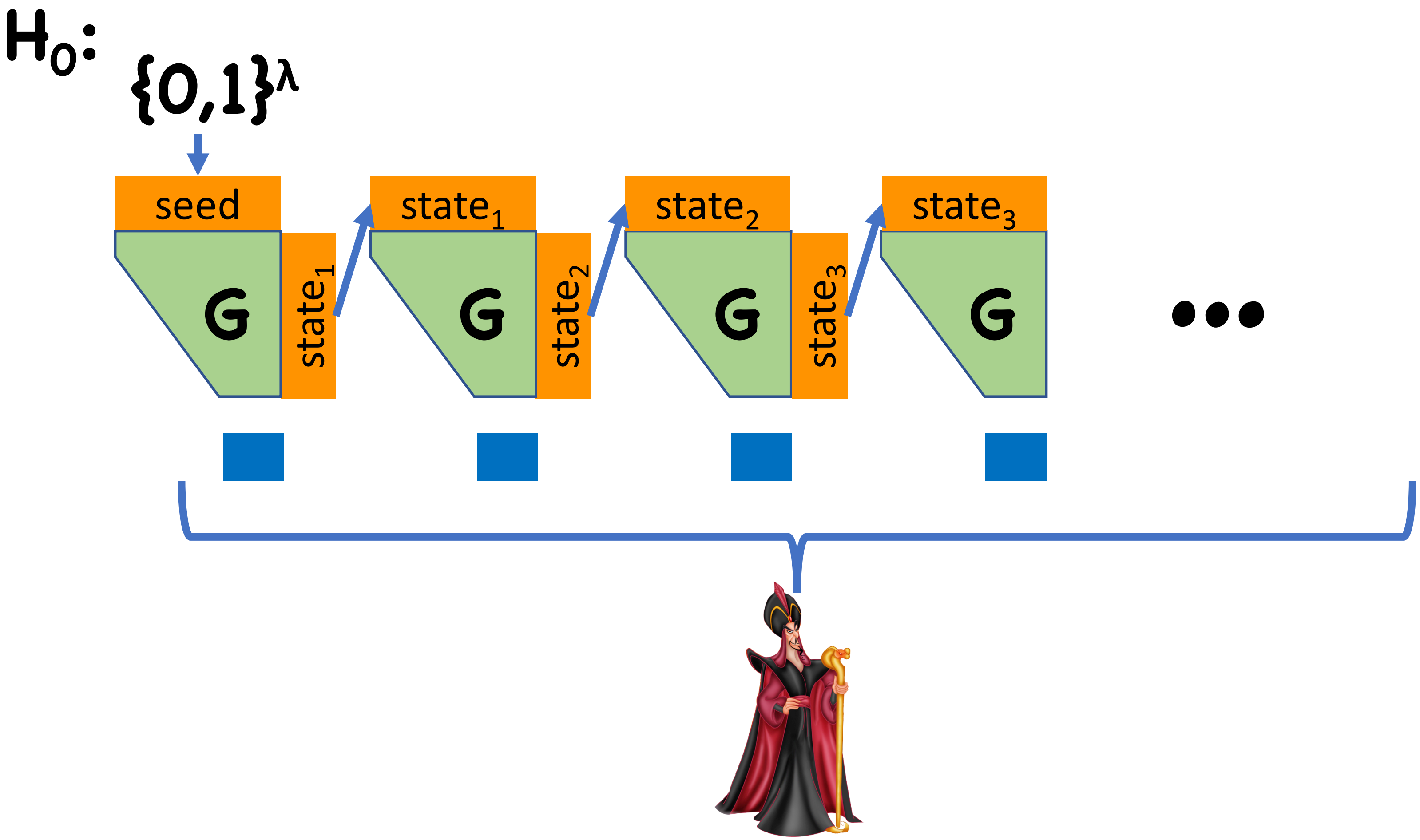


Security Proof

Assume towards contradiction  …

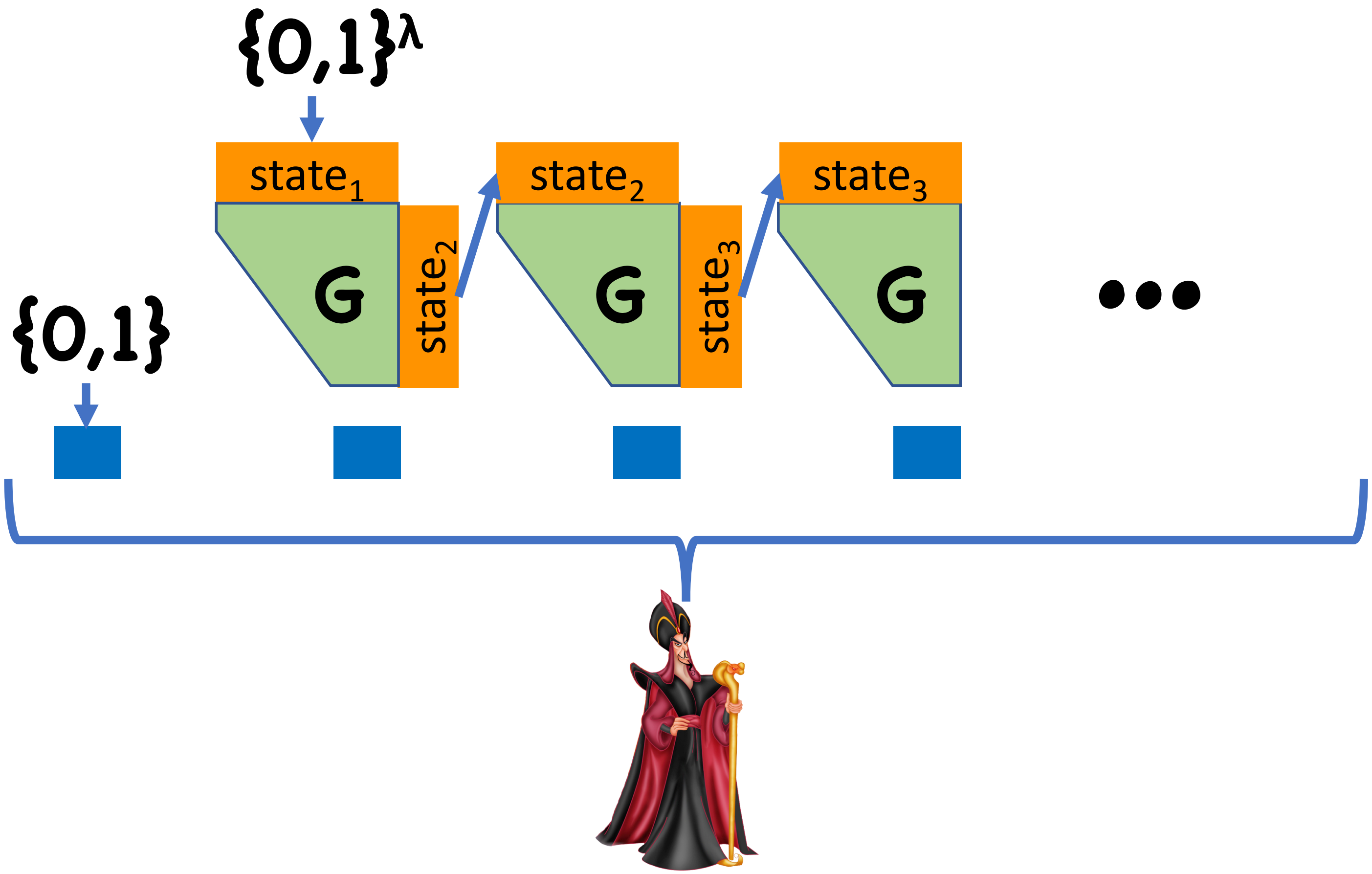
Define hybrids…

Security Proof



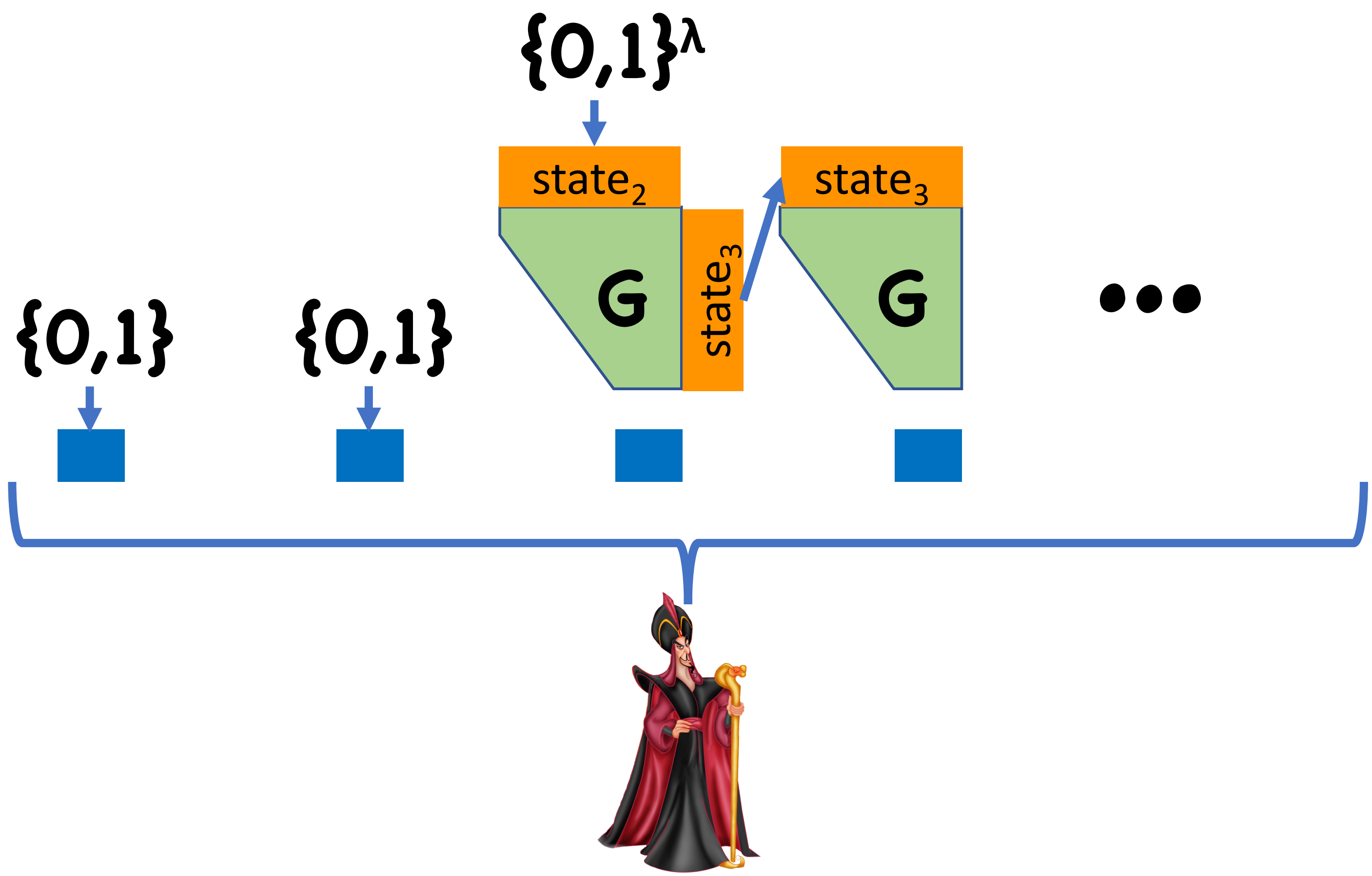
Security Proof

H_1:



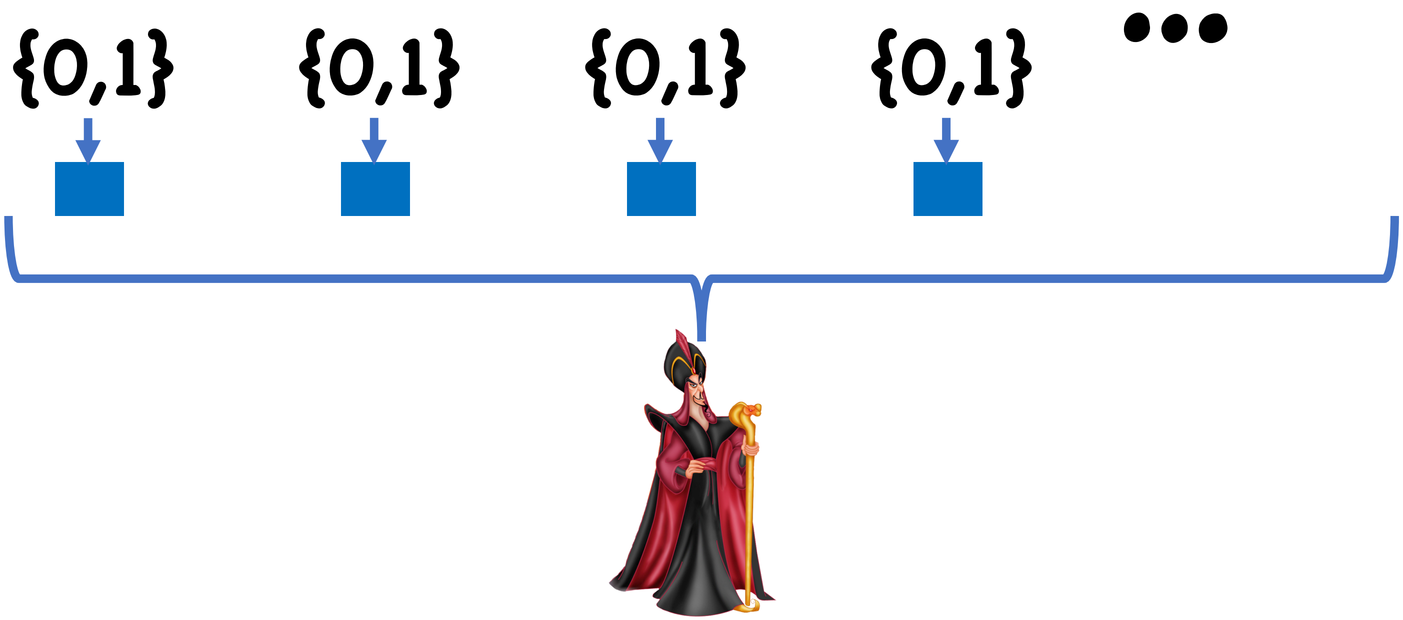
Security Proof

H_2:



Security Proof

H_t:



Security Proof

H_0 corresponds to pseudorandom x

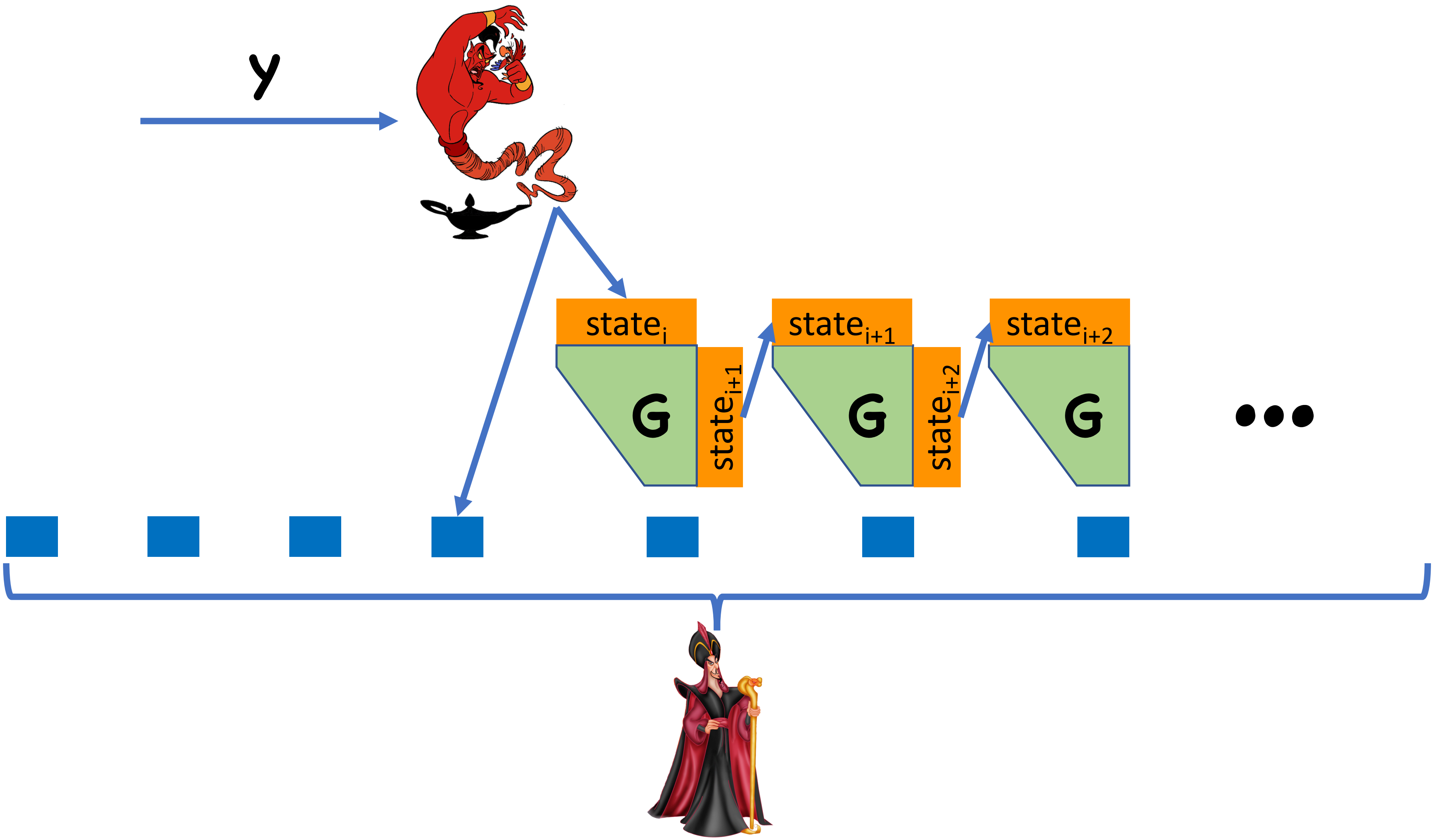
H_t corresponds to truly random x

Let $q_i = \Pr[$  $(x)=1 : x \leftarrow H_i\ ]$

By assumption, $|q_t - q_0| > \varepsilon$

$\Rightarrow \exists i$ s.t. $|q_i - q_{i-1}| > \varepsilon/t$

Security Proof



Security Proof

Analysis

- If $y = G(s)$, then  sees H_{i-1}

  $\Rightarrow$ Pr[ outputs 1] $= q_{i-1}$

  $\Rightarrow$ Pr[ outputs 1] $= q_{i-1}$

- If y is random, then  sees H_i

  $\Rightarrow$ Pr[ outputs 1] $= q_i$

  $\Rightarrow$ Pr[ outputs 1] $= q_i$

Summary So Far

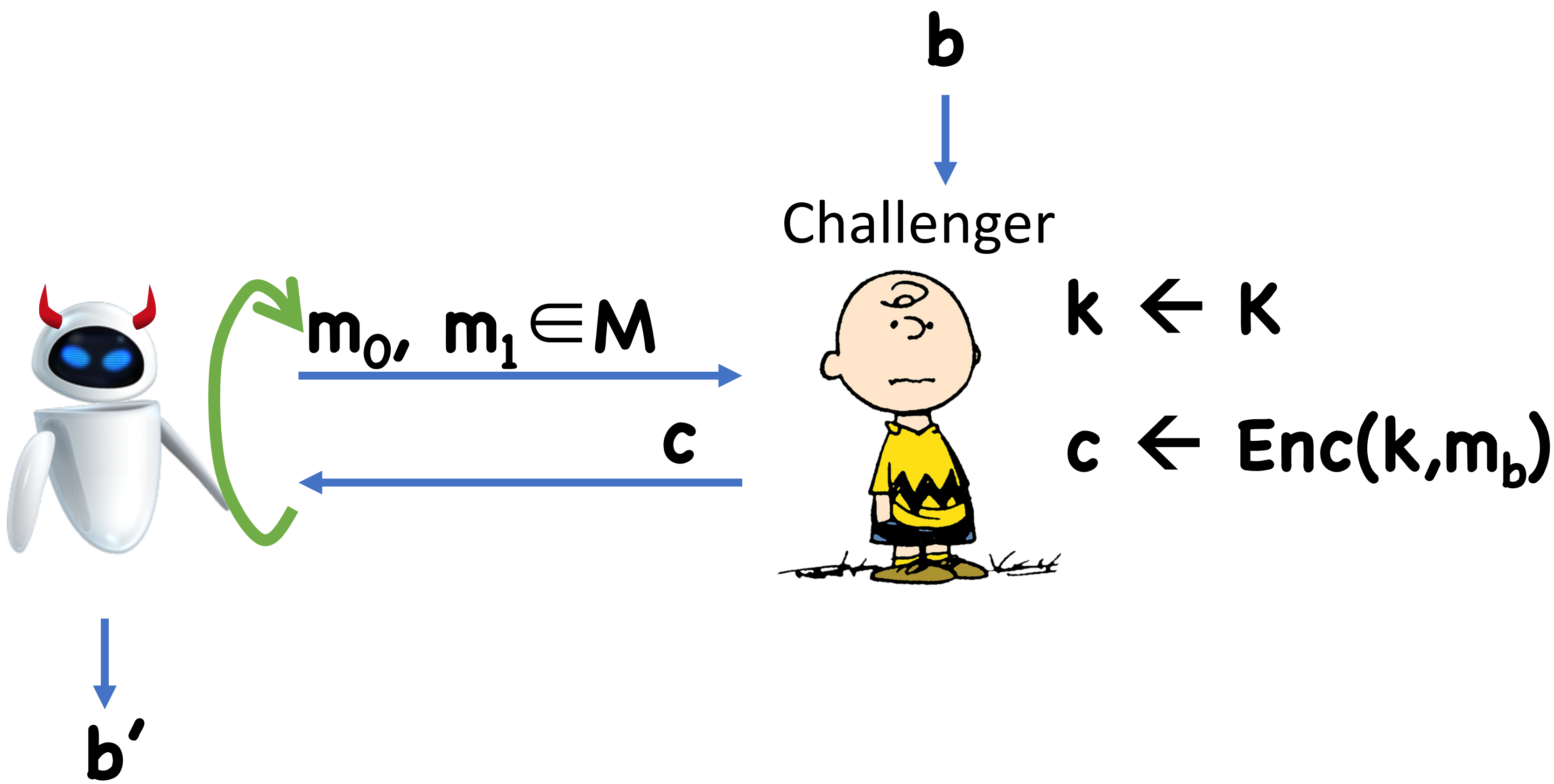
Stream ciphers = Encrytpion with PRG

- Secure encryption for arbitrary length, number of messages (though we did not completely prove it)

However, implementation difficulties due to having to maintaining state

Multiple Message Security

Left-or-Right Experiment

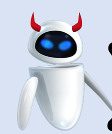


$\text{LoR-Exp}_b($  $)$

LoR Security Definition

Definition: (Enc, Dec) has **(t,q,ε)–Left-or-Right indistinguishability** if, for all  running in time at most **t** and making at most **q** queries,

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

$- \Pr[1 \leftarrow \text{LoR-Exp}_1($  $)\,]$ $\big| \;\leq\; \varepsilon$

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”

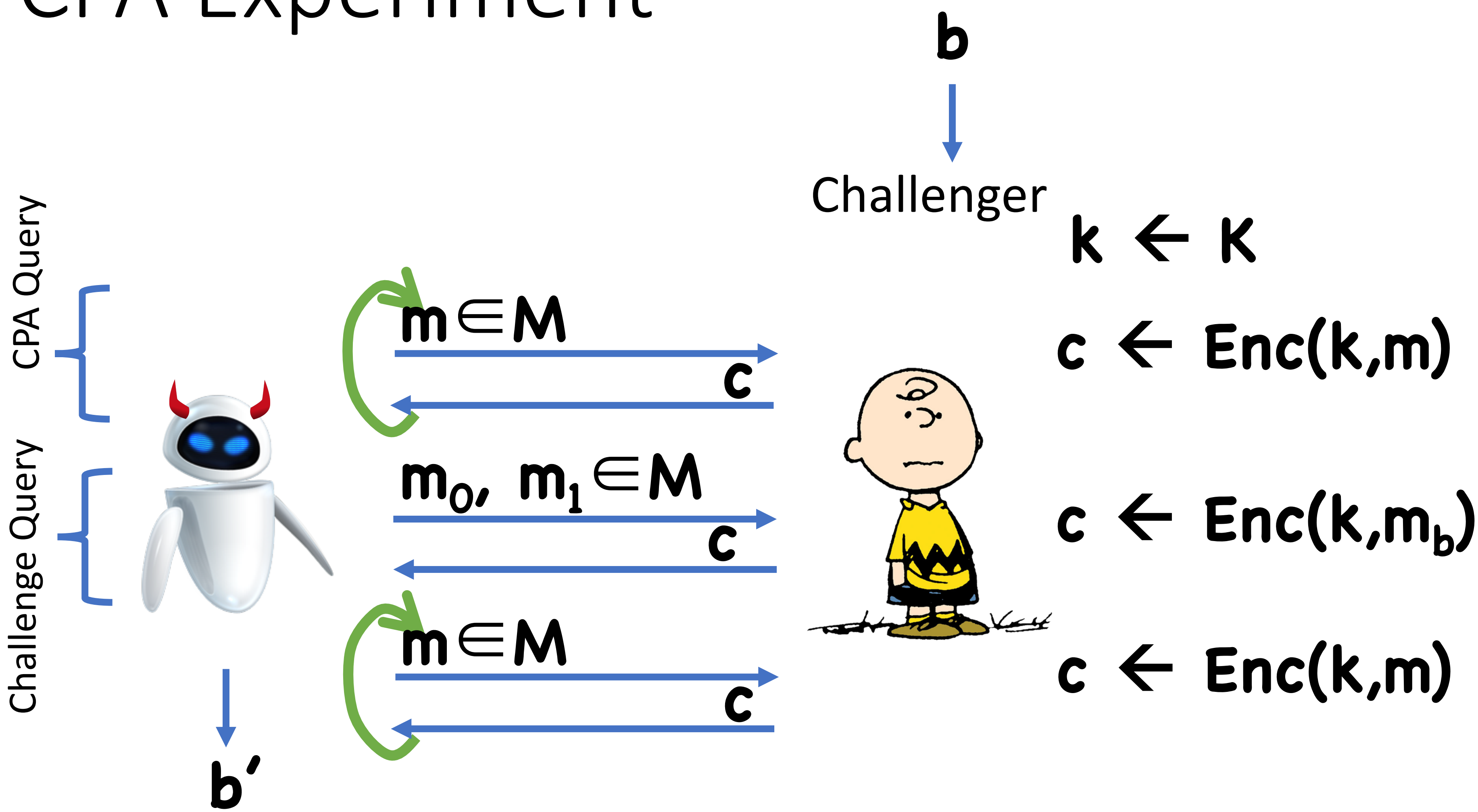
Alternate Notion: CPA Security

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

Examples:

- 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$_b$()

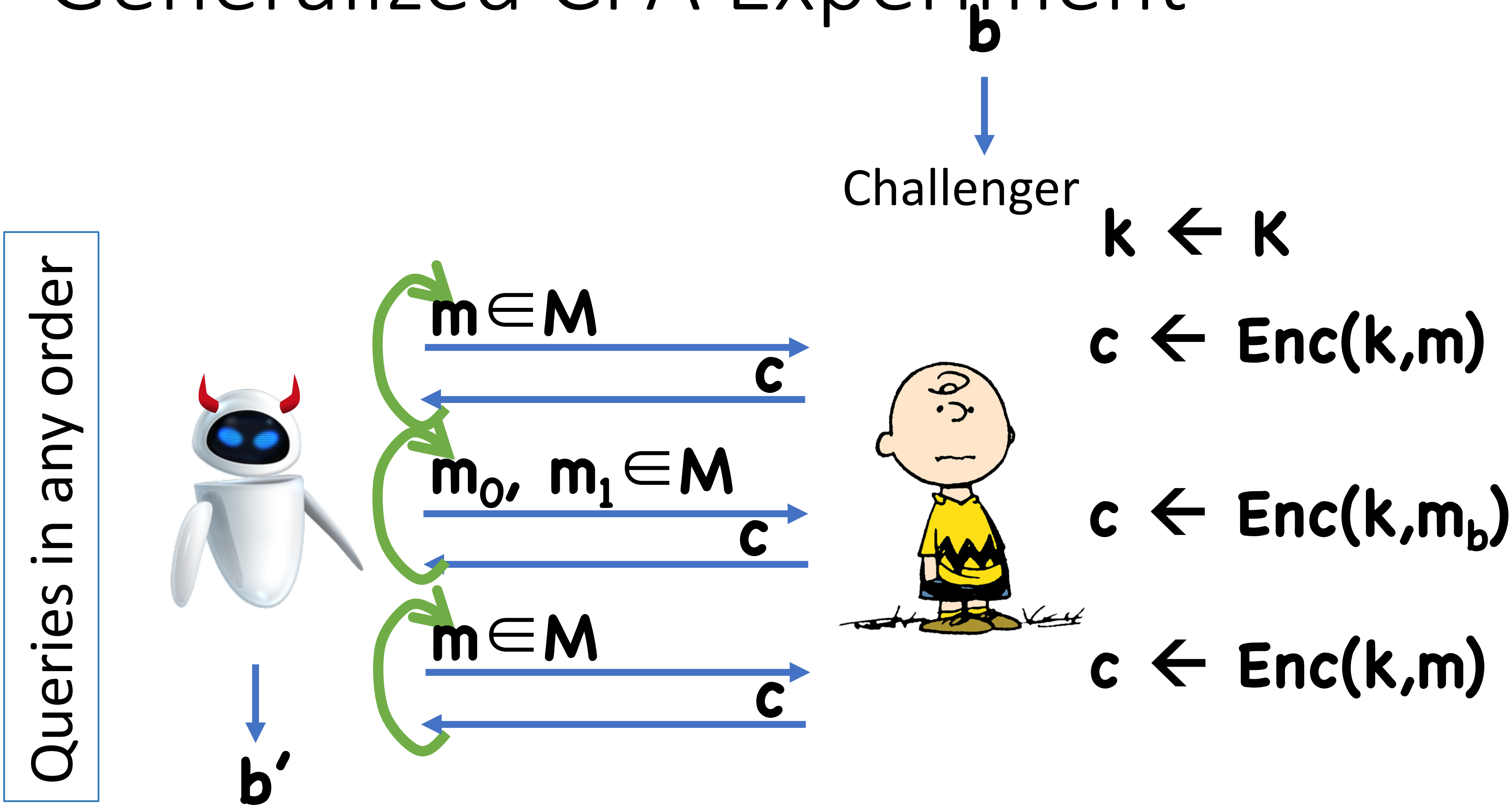
CPA Security Definition

Definition: (Enc, Dec) is **(t,q,ε)-CPA Secure** if, for all  running in time at most **t** and making at most **q** CPA queries,

| Pr[1←CPA-Exp$_0$(**)]**

− Pr[1←CPA-Exp$_1$(**)] | ≤ ε**

Generalized CPA Experiment



GCPA-Exp$_b$( **)**

GCPA Security Definition

Definition: (Enc, Dec) is **(t,c,q,ε)-Generalized CPA secure** if, for all  running in time at most **t** and making at most **c** challenge and **q** CPA queries,

$\big|$ $\Pr[1 \leftarrow \text{GCPA-Exp}_0($  $)]$

$-\ \Pr[1 \leftarrow \text{GCPA-Exp}_1($  $)]$ $\big| \le \varepsilon$

Equivalences

Theorem:

Left-or-Right indistinguishability

$\Updownarrow$

CPA-security

$\Updownarrow$

Generalized CPA-security

Equivalences

Theorem:

- (t,q,ε)-LoR $\Longrightarrow$ $(t-t',c,q-c,\varepsilon)$-GCPA
- $(t,1,q,\varepsilon)$-GCPA $\Longrightarrow$ $(t-t',q,\varepsilon)$-CPA
- (t,q,ε)-CPA $\Longrightarrow$ $(t-t',q+1,\ \varepsilon(q+1))$-LoR

Proof

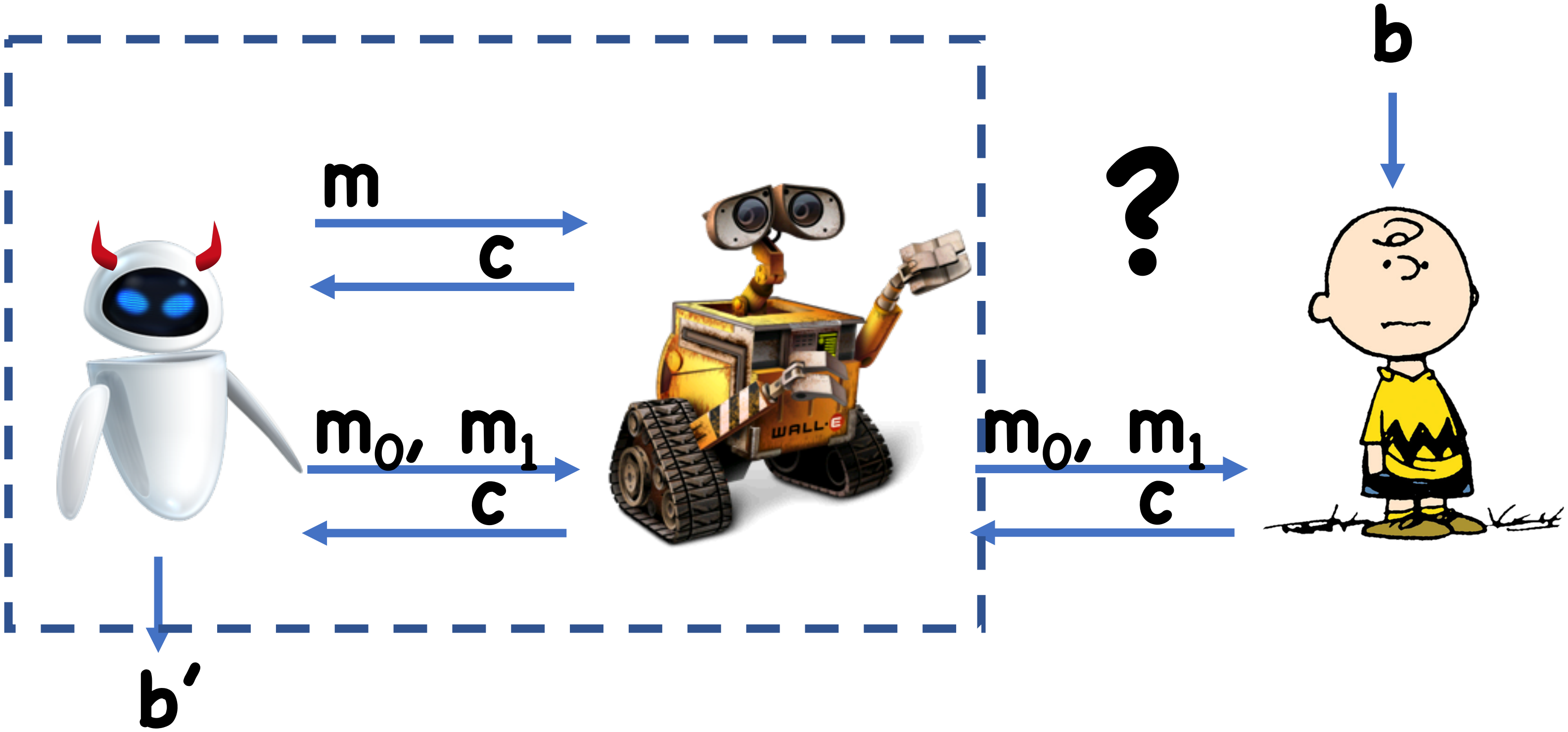
Generalized CPA-security → CPA-security

- 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 challenge/LoR queries

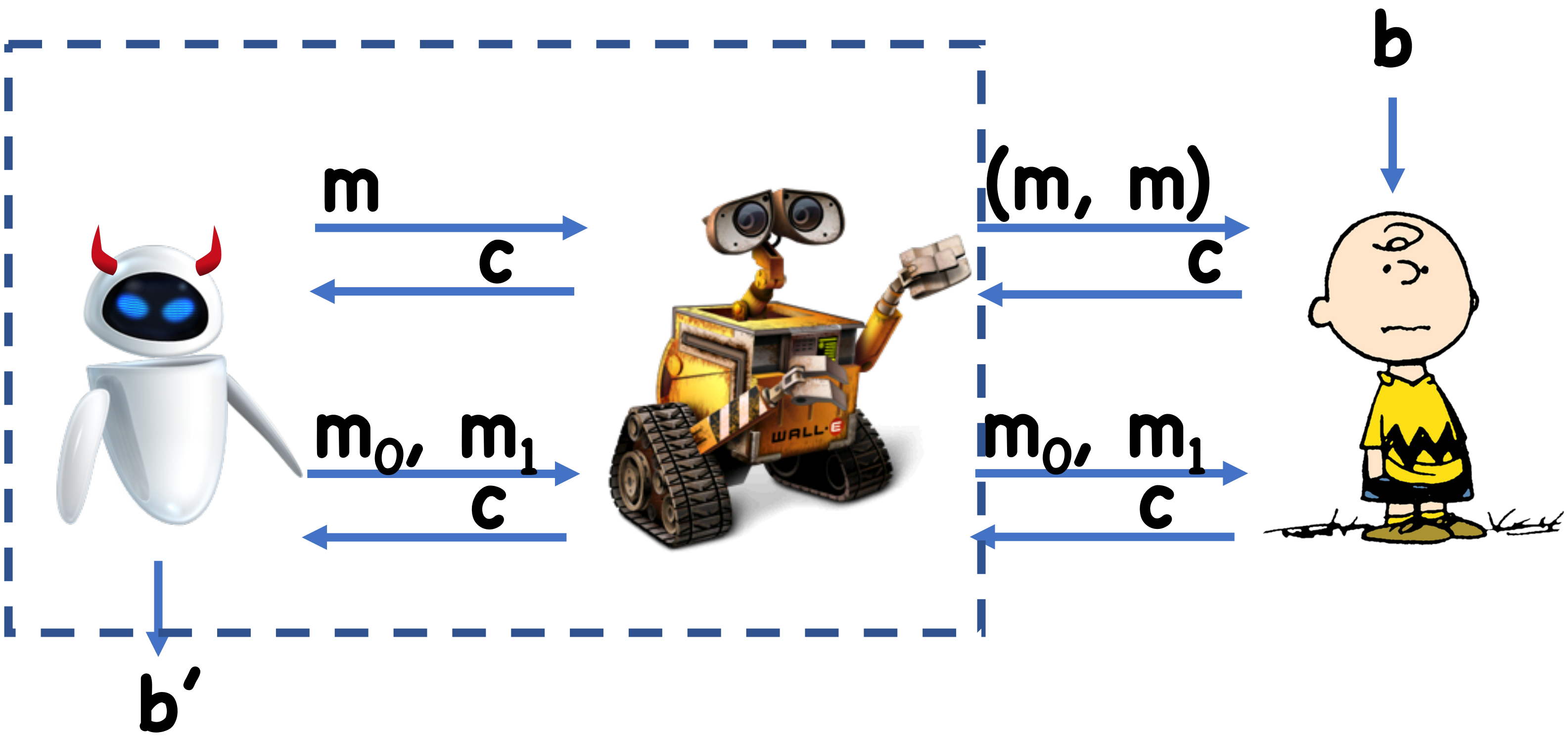
Proof

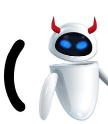
Left-or-Right → Generalized CPA

- 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



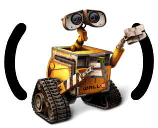
Pr[1←LoR-Exp$_b$()] = Pr[1←GCPA-Exp$_b$()]

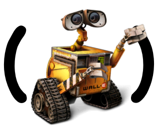


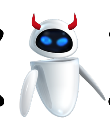
Pr[1←LoR-Exp$_b$()] = Pr[1←GCPA-Exp$_b$()]

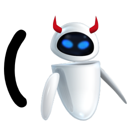
Proof

Left-or-Right → Generalized CPA

| Pr[1←LoR-Exp$_0$()]

 − Pr[1←LoR-Exp$_1$()] |

= | Pr[1←GCPA-Exp$_0$()]

 − Pr[1←GCPA-Exp$_1$()] | = ε

Proof

(regular) CPA → Left-or-Right

- Assume towards contradiction that we have an adversary  for the **$(t, q+1,\ \varepsilon(q+1))$–LoR**

- Hybrids!

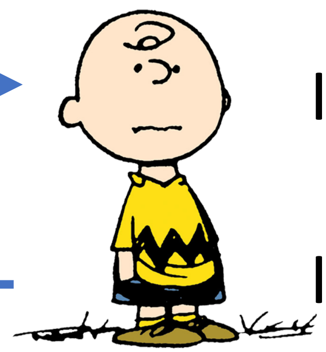
Hybrid **i**:

$m_0, m_1 \rightarrow$

$\leftarrow c$

$\downarrow$

b'



k ← K

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

c ← Enc(k,m_0)

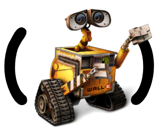
If more than **i** queries so far,

c ← Enc(k,m_1)

Proof

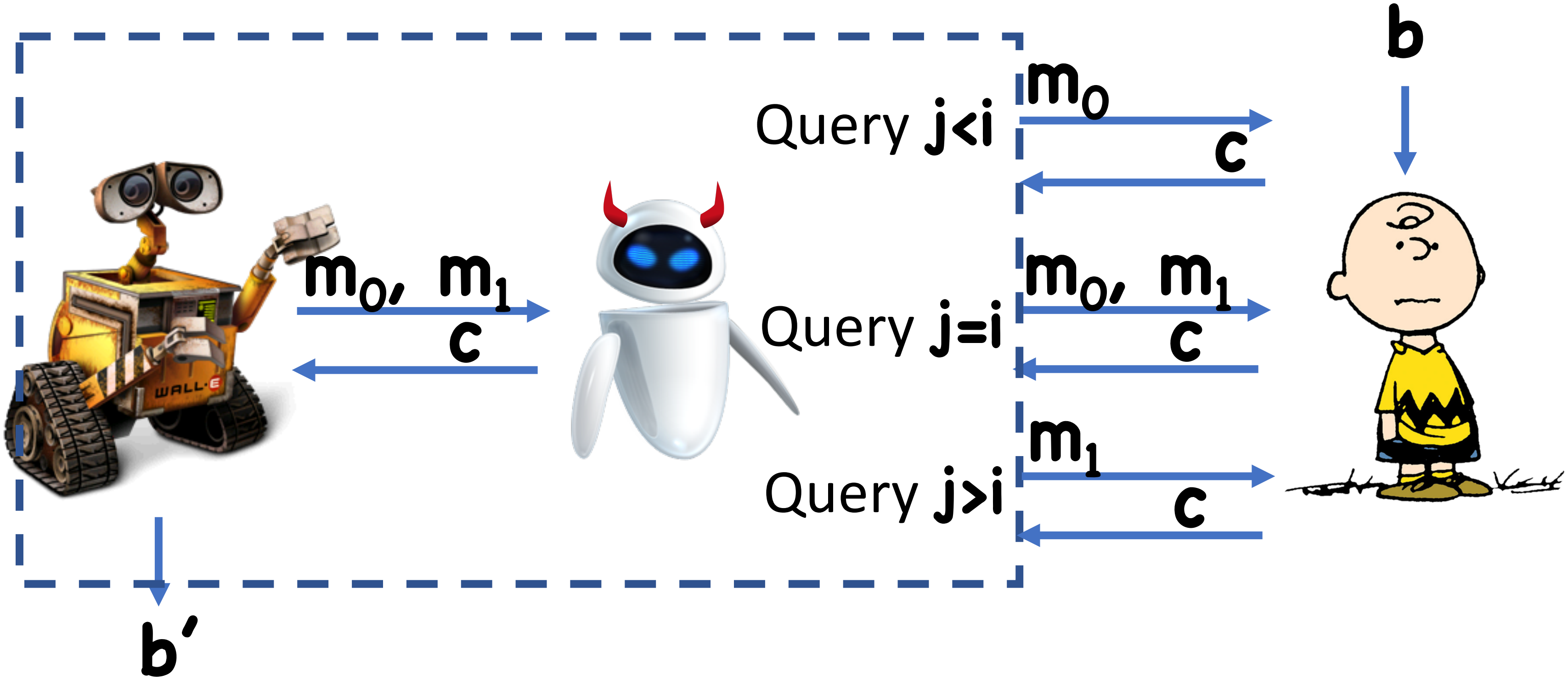
(regular) CPA → Left-or-Right

- Hybrid **0** is identical to $\mathbf{LoR\text{-}Exp}_1($$)$

- Hybrid **q+1** is identical to $\mathbf{LoR\text{-}Exp}_0($$)$

- We know that  distinguishes Hybrid **q+1** and Hybrid **0** with advantage $\boldsymbol{\varepsilon(q+1)}$

  $\Rightarrow$ $\exists$ **i** s.t.  distinguishes Hybrid **i** and Hybrid **i–1** with advantage $\boldsymbol{\varepsilon}$



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

Proof

(regular) CPA → Left-or-Right

| Pr[1←CPA-Exp$_0$()]

− Pr[1←CPA-Exp$_1$()] |

= | Pr[1←  in Hybrid **i**]

− Pr[1←  in Hybrid **i−1**] | = ε

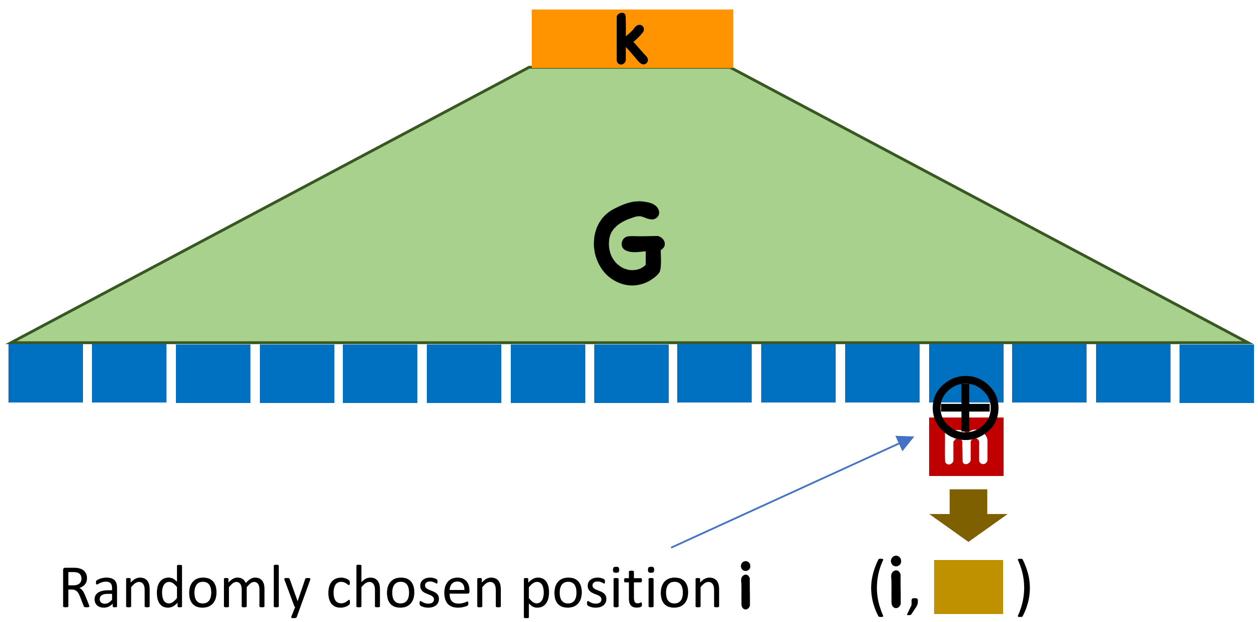
Equivalences

Theorem:
- (t,q,ε)-LoR $\Longrightarrow$ $(t-t',c,q-c,\varepsilon)$-GCPA
- $(t,1,q,\varepsilon)$-GCPA $\Longrightarrow$ $(t-t',q,\varepsilon)$-CPA
- (t,q,ε)-CPA $\Longrightarrow$ $(t-t',q+1,\ \varepsilon(q+1))$-LoR

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

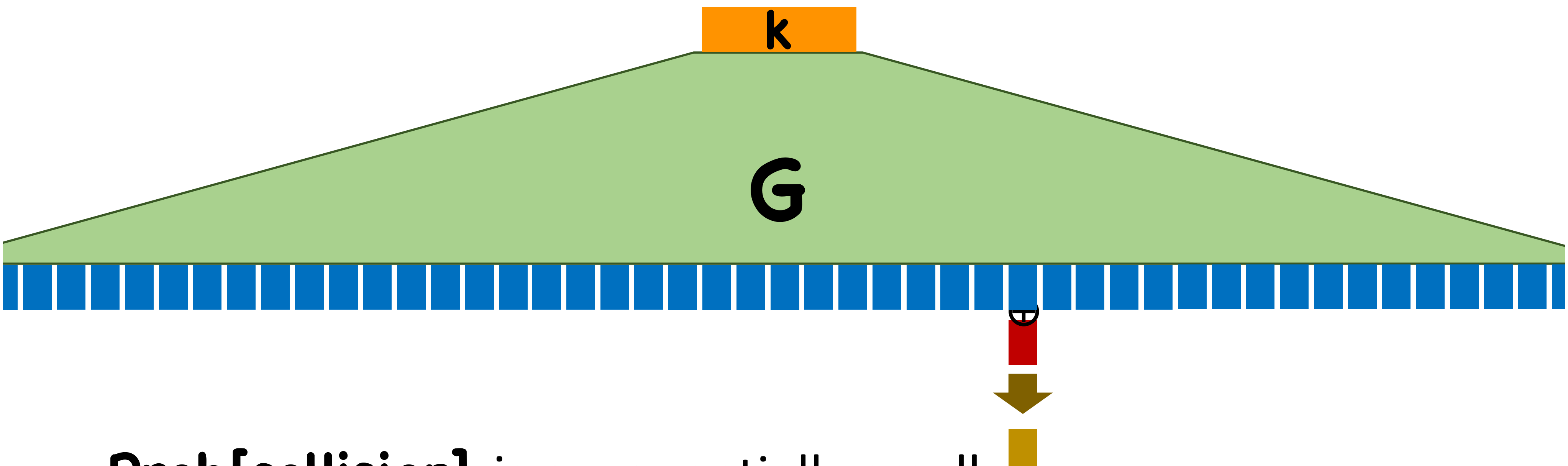
$Pr[Collision] \leq q^2/2n$, where
- **q =** number of messages encrypted
- **n =** number of blocks

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

So we get **$(t,q,2\varepsilon+q^2/2n)$-LoR** security for small **n**

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

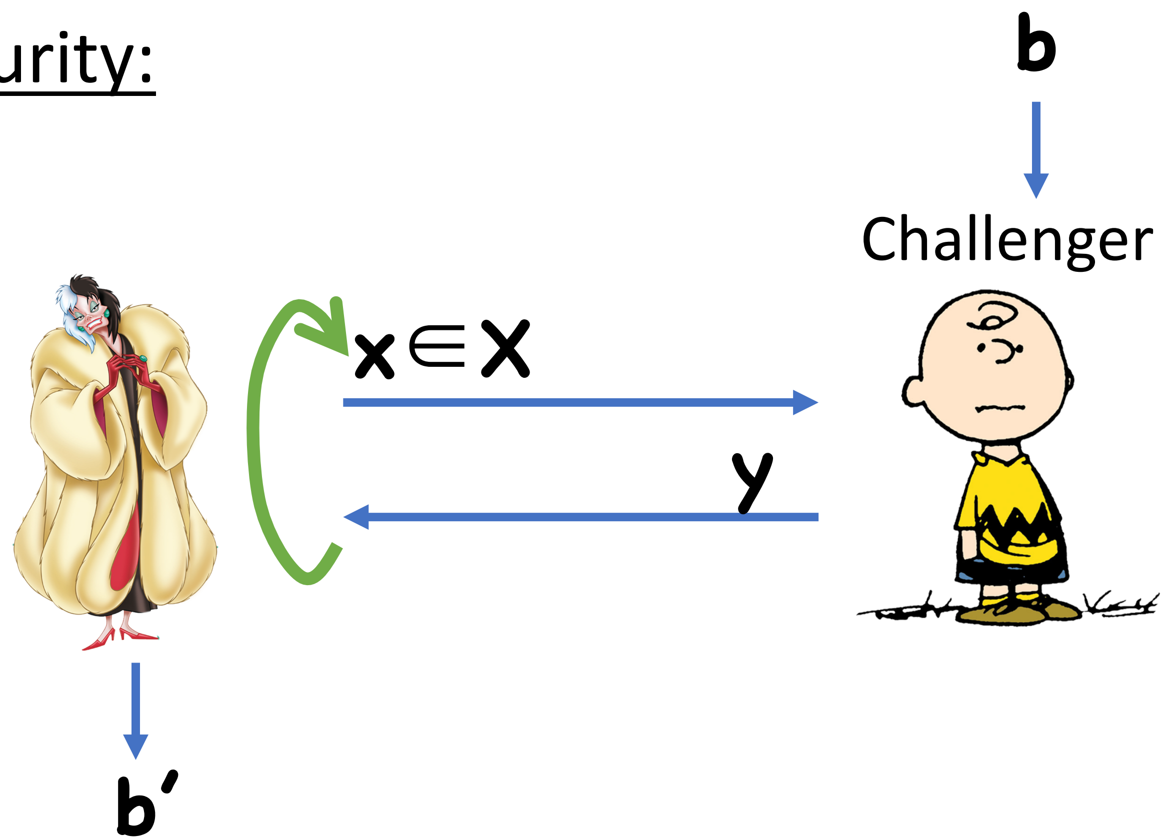
Functions that “look like” random functions

Syntax:

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

Pseudorandom Functions

Security:



Pseudorandom Functions

<u>Security:</u>

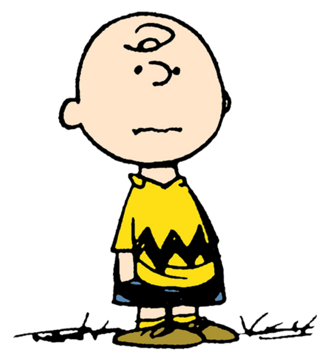
b=0

Challenger

$k \leftarrow K$

$x \in X$

y



$y \leftarrow F(k,x)$

b'

PRF-Exp$_0$( **)**

Pseudorandom Functions

<u>Security:</u>

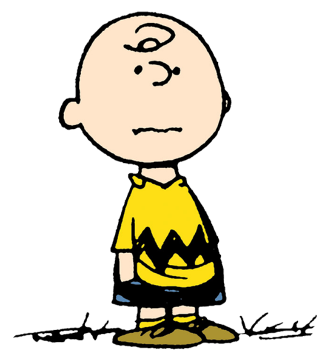
b=1

Challenger

$H \leftarrow Funcs(X,Y)$

$x \in X$

y



$y = H(x)$

b'

PRF-Exp$_1$()

PRF Security Definition

Definition: F is a **(t,q,ε)-**secure PRF if, for all  running in time at most **t** and making at most **q** queries,

| Pr[1←PRF-Exp$_0$( **)]**

− Pr[1←PRF-Exp$_1$( **)] | ≤ ε**

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)

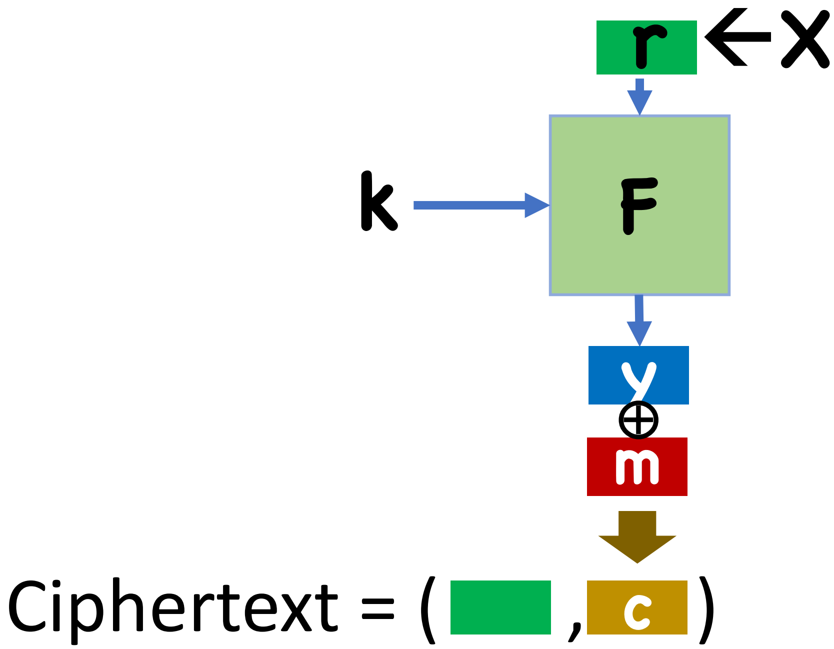
Dec(k, (r,c)):

- 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

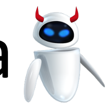


Security

Theorem: If $\mathbf{F}$ is a $(\mathbf{t},\mathbf{q},\boldsymbol{\varepsilon})$-secure PRF with domain $\mathbf{X}$, then **(Enc,Dec)** is

$$(t-t',\ q,\ 2\varepsilon + q^2/2|X|)\text{-}\mathbf{LoR}\text{ secure.}$$

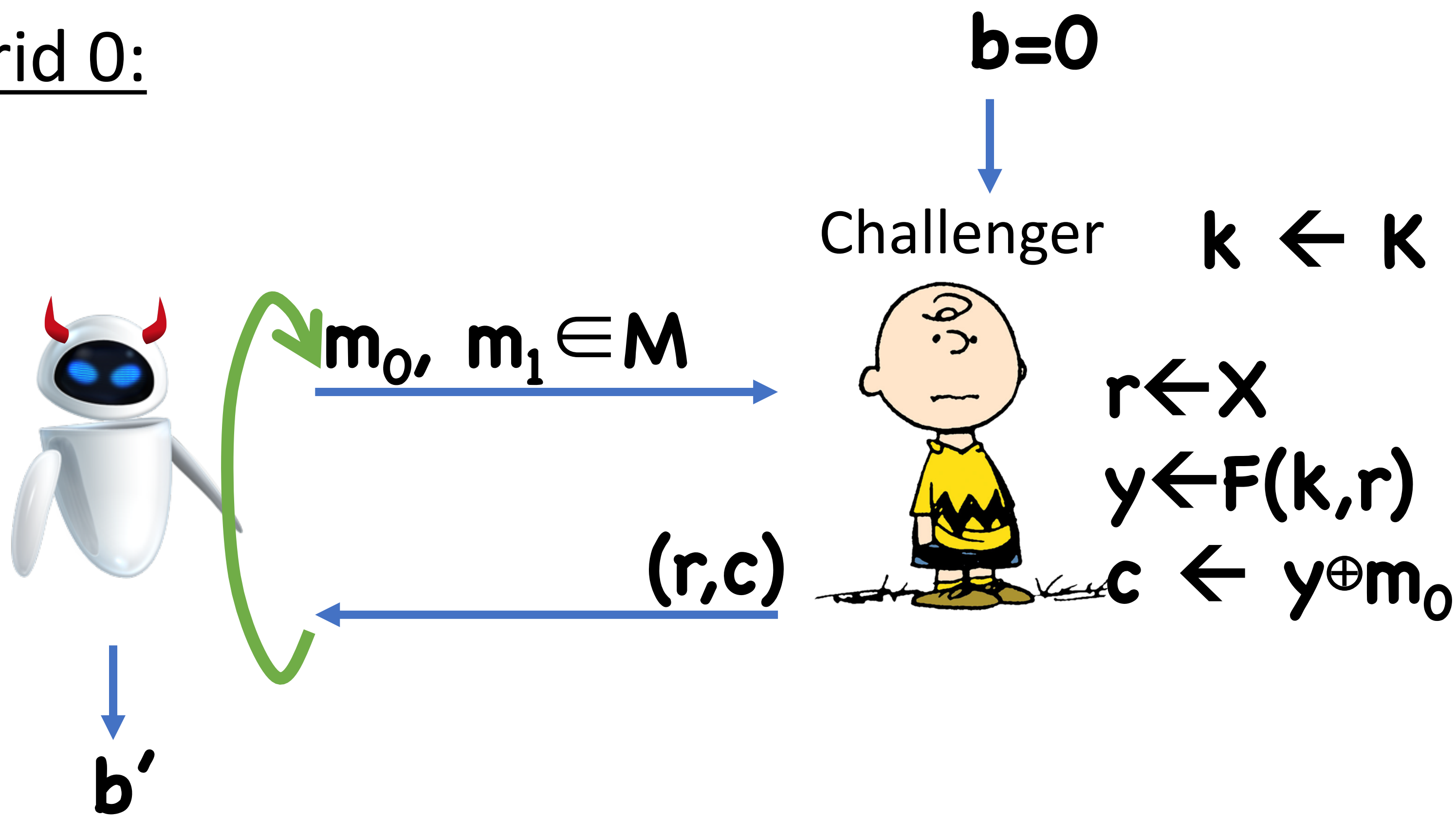
Proof

Assume toward contradiction that there exists a  running in time at most **t** that has advantage **$2\varepsilon + q^2/2|X|$** in breaking **(Enc,Dec)**

Hybrids…

Proof

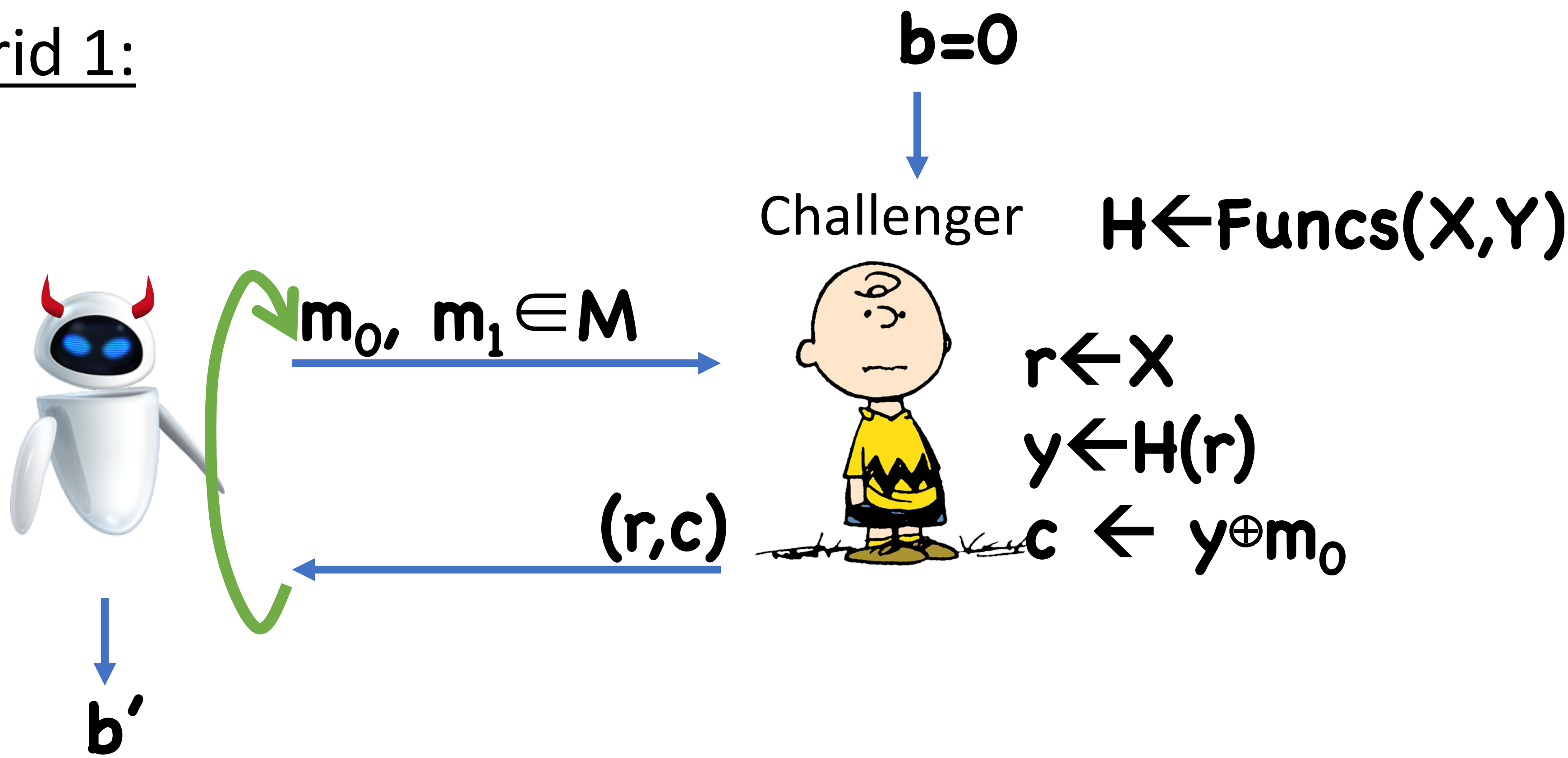
Hybrid 0:



$\text{LoR-Exp}_0($$)$

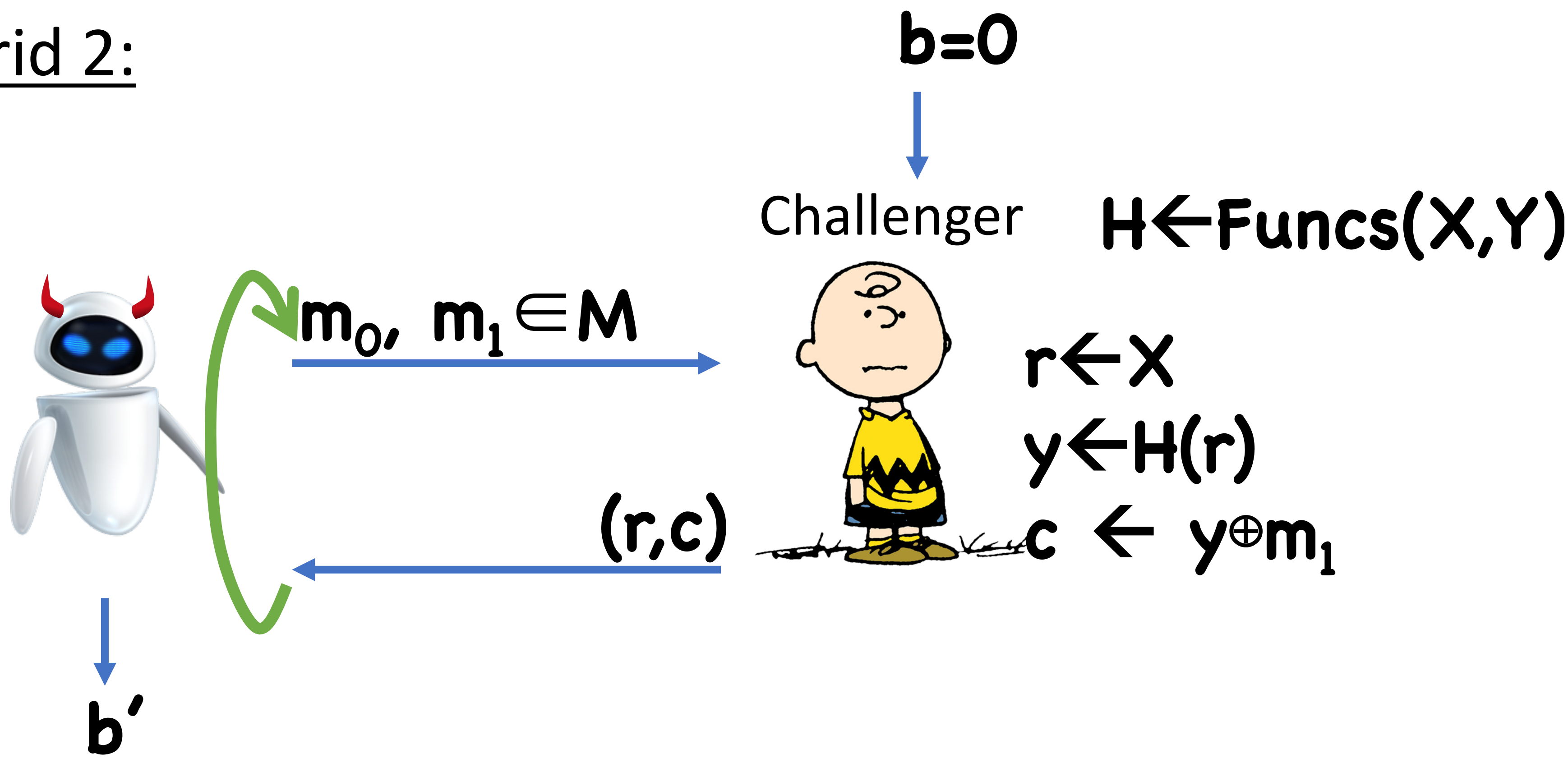
Proof

<u>Hybrid 1:</u>



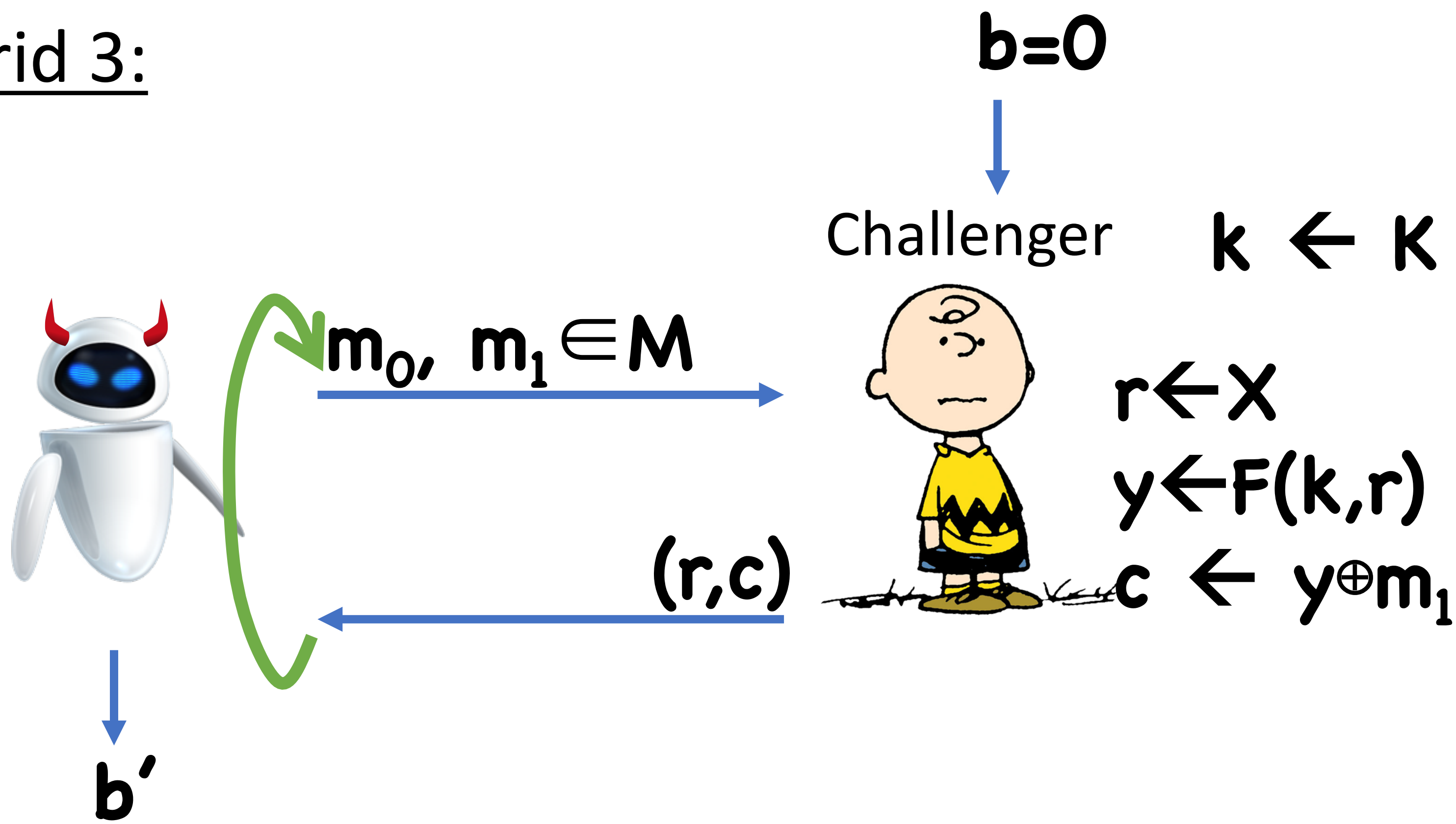
Proof

<u>Hybrid 2:</u>



Proof

<u>Hybrid 3:</u>



LoR-Exp$_1$( **)**

Proof

Assume toward contradiction that there exists a  running in time at most t that has advantage $2\varepsilon + q^2/2|X|$ in breaking **(Enc,Dec)**

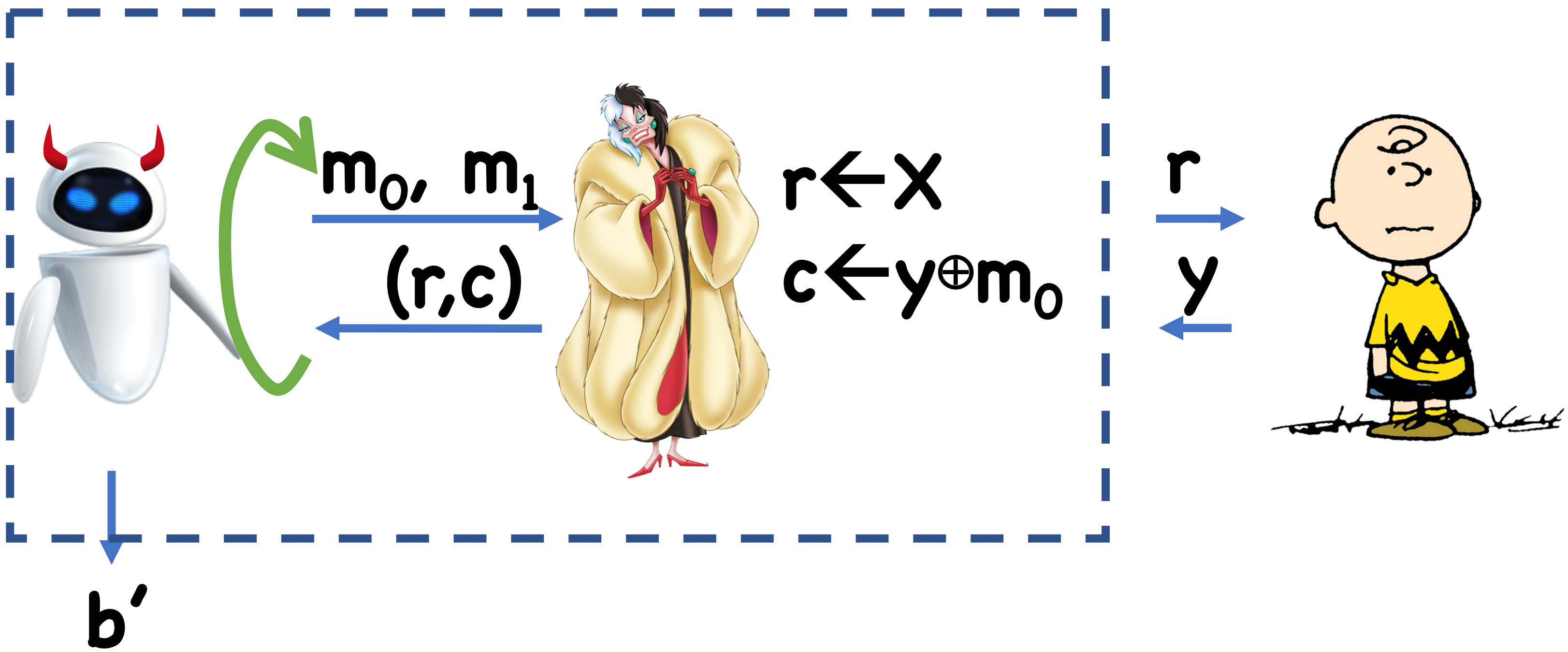
 distinguishes Hybrid 0 from Hybrid 3 with advantage ε, so either 

- Dist. Hybrid 0 from Hybrid 1 with adv. ε
- Dist. Hybrid 1 from Hybrid 2 with adv. $q^2/2|X|$
- Dist. Hybrid 2 from Hybrid 3 with adv. ε

Proof

Suppose  distinguishes Hybrid 0 from Hybrid 1

Construct 



Proof

Suppose  distinguishes Hybrid 0 from Hybrid 1

Construct 

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

Therefore,  has advantage $\boldsymbol{\varepsilon}$

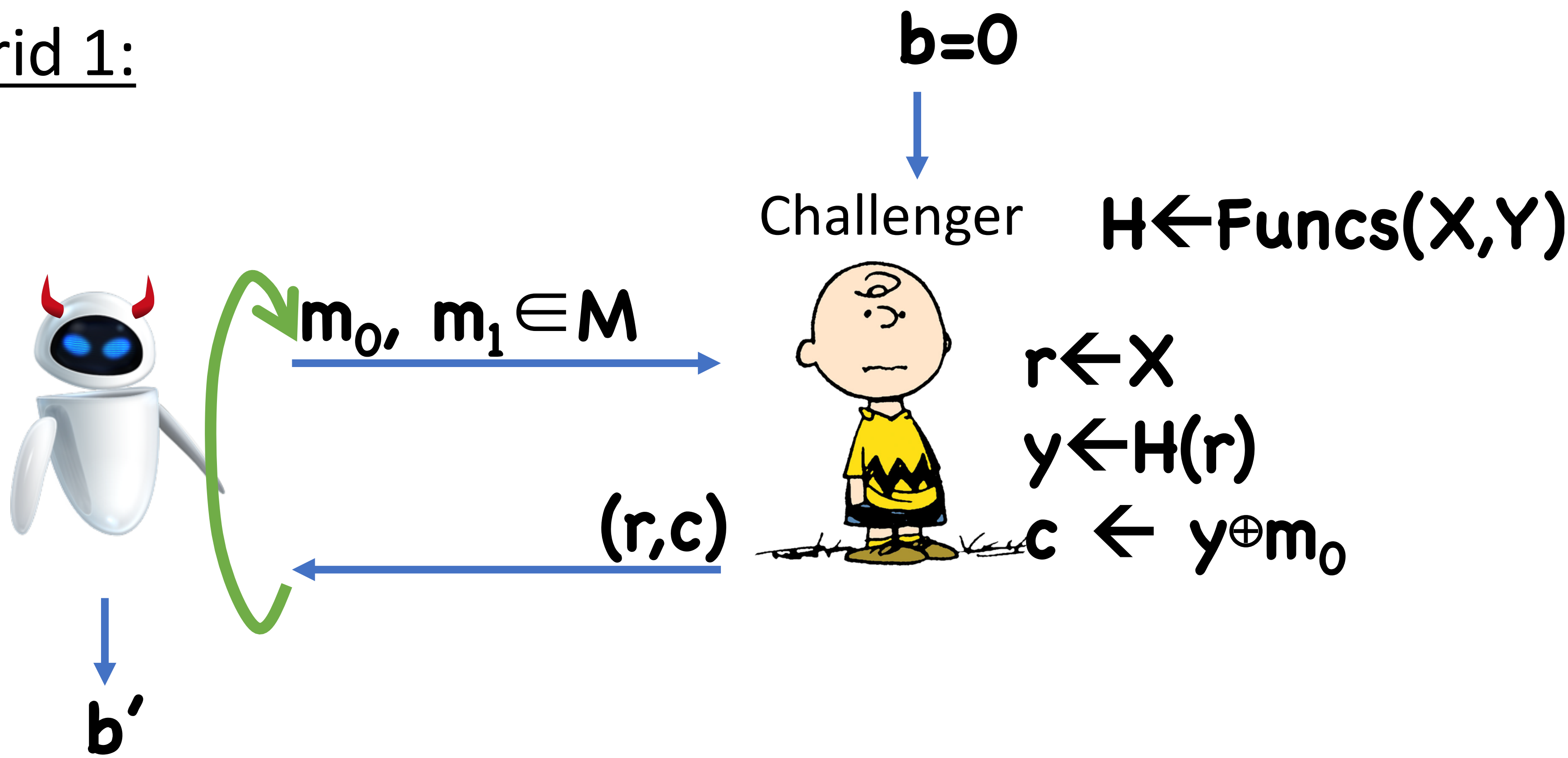
$\Rightarrow$ contradiction

Proof

Suppose  distinguishes Hybrid 1 from Hybrid 2

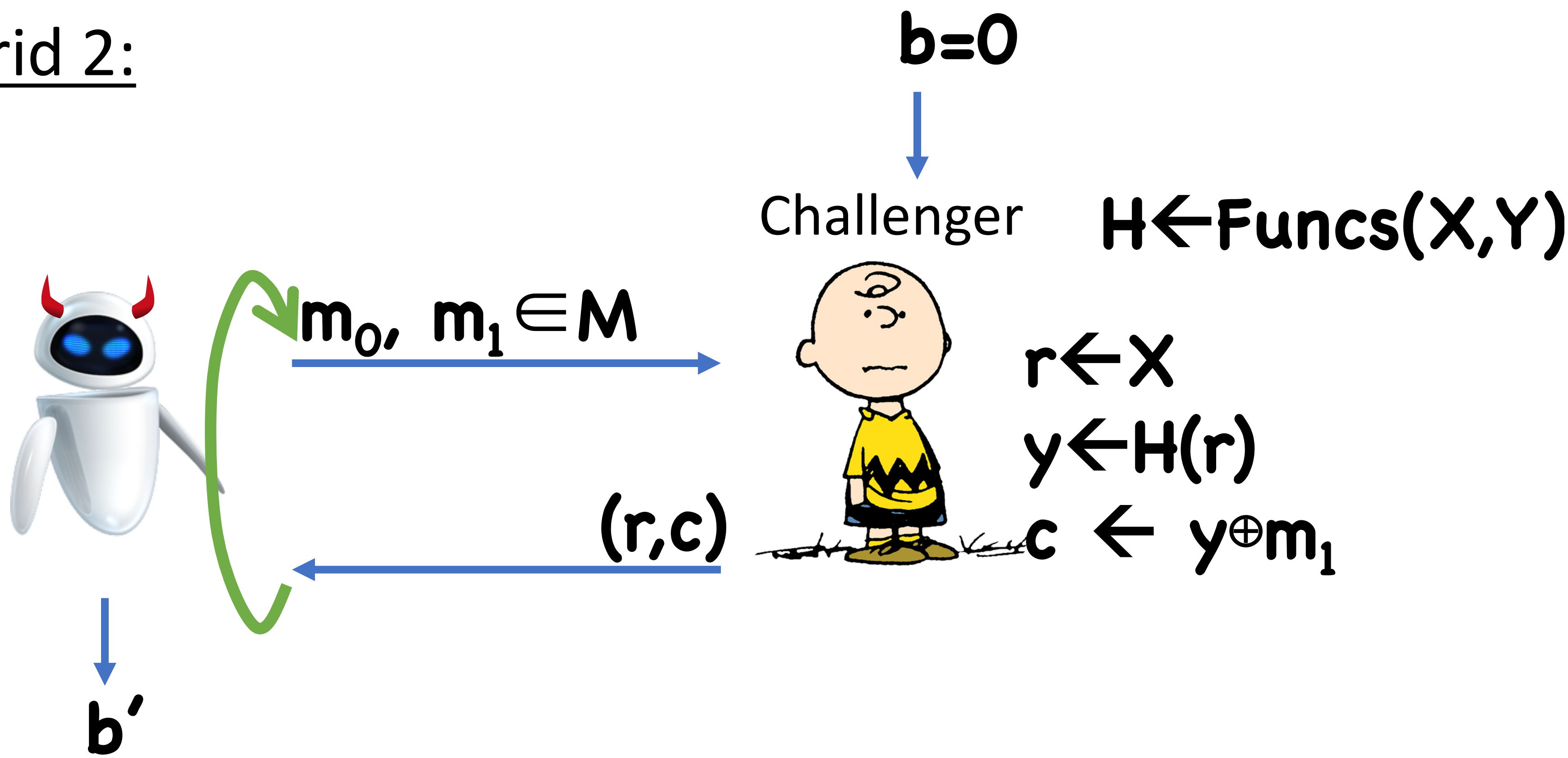
Proof

<u>Hybrid 1:</u>



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 $\mathbf{r}$'s$]$

$$= \Pr[r^{(1)}=r^{(2)} \text{ or } r^{(1)}=r^{(3)} \text{ or } \ldots \text{ or } r^{(1)}=r^{(q)}$$

$$\text{or } r^{(2)}=r^{(3)} \text{ or } \ldots\ ]$$

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

$$+ \Pr[r^{(2)}=r^{(3)}] + \ldots$$

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

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

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

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

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

How big to choose **X**?

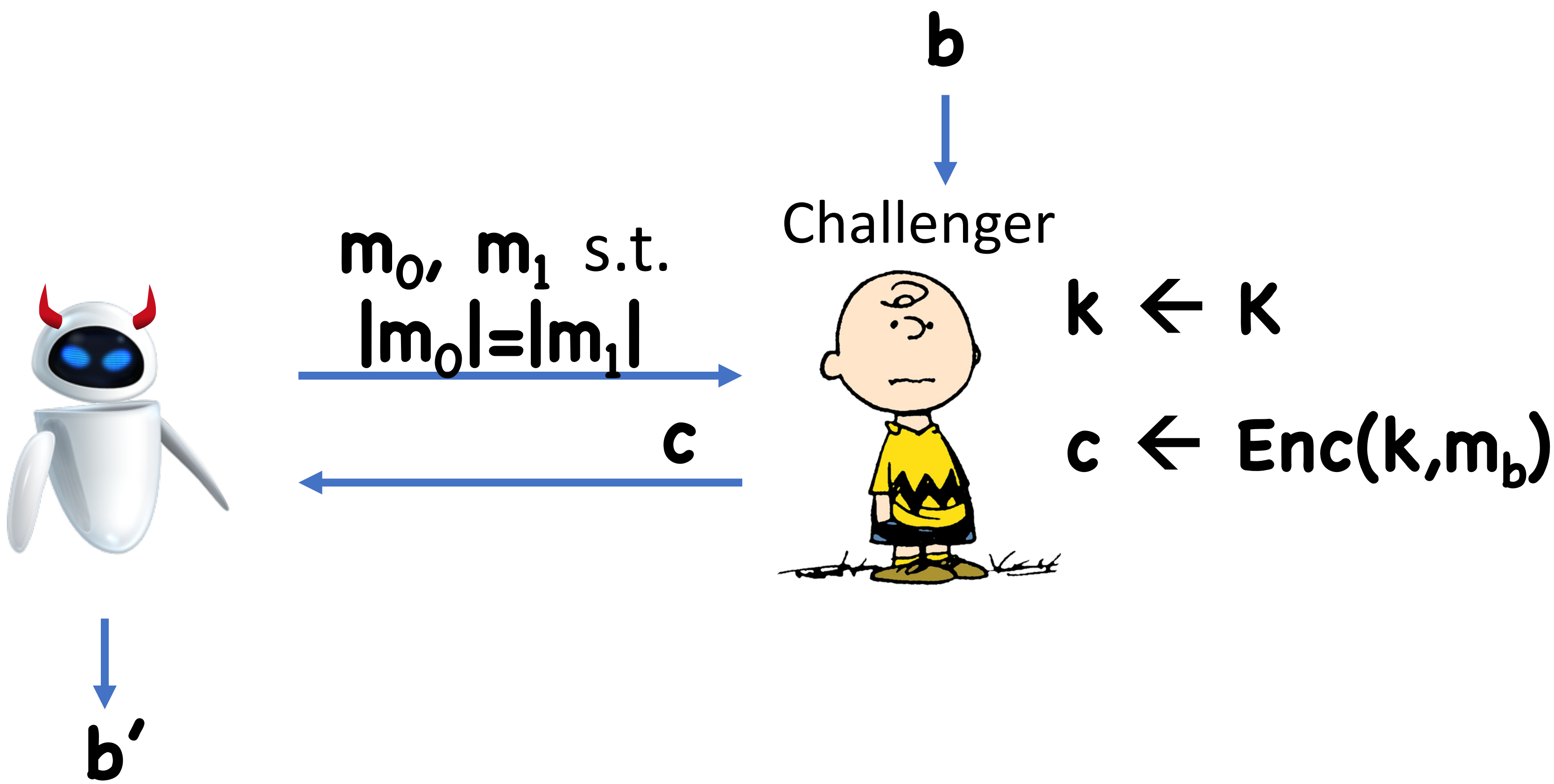
Using PRFs to Build Encryption

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$_b$()

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

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

Dec'(k, (c_1, …, c_l)):

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

Output $m = m_1 m_2 \ldots, m_l$

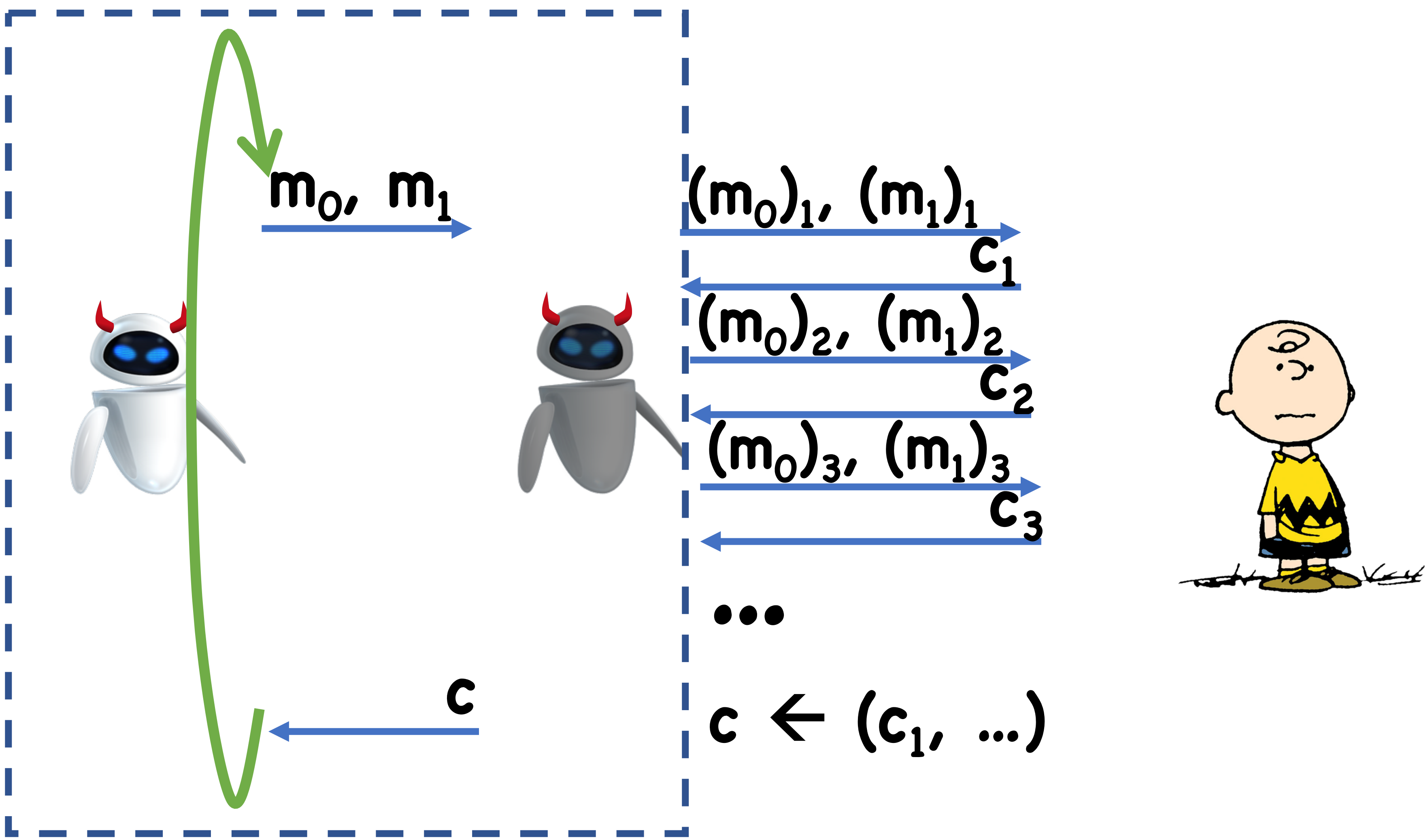
Theorem: If **(Enc,Dec)** is **(t,q,ε)**-LoR secure, then **(Enc',Dec')** is **(t–t',q/n,ε)**-LoR secure for messages of length up to **n**

Proof

Assume toward contradiction that there exists a  running in time at most **t–t'**, making **q/n** LoR queries on messages of length up to **n**, which 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

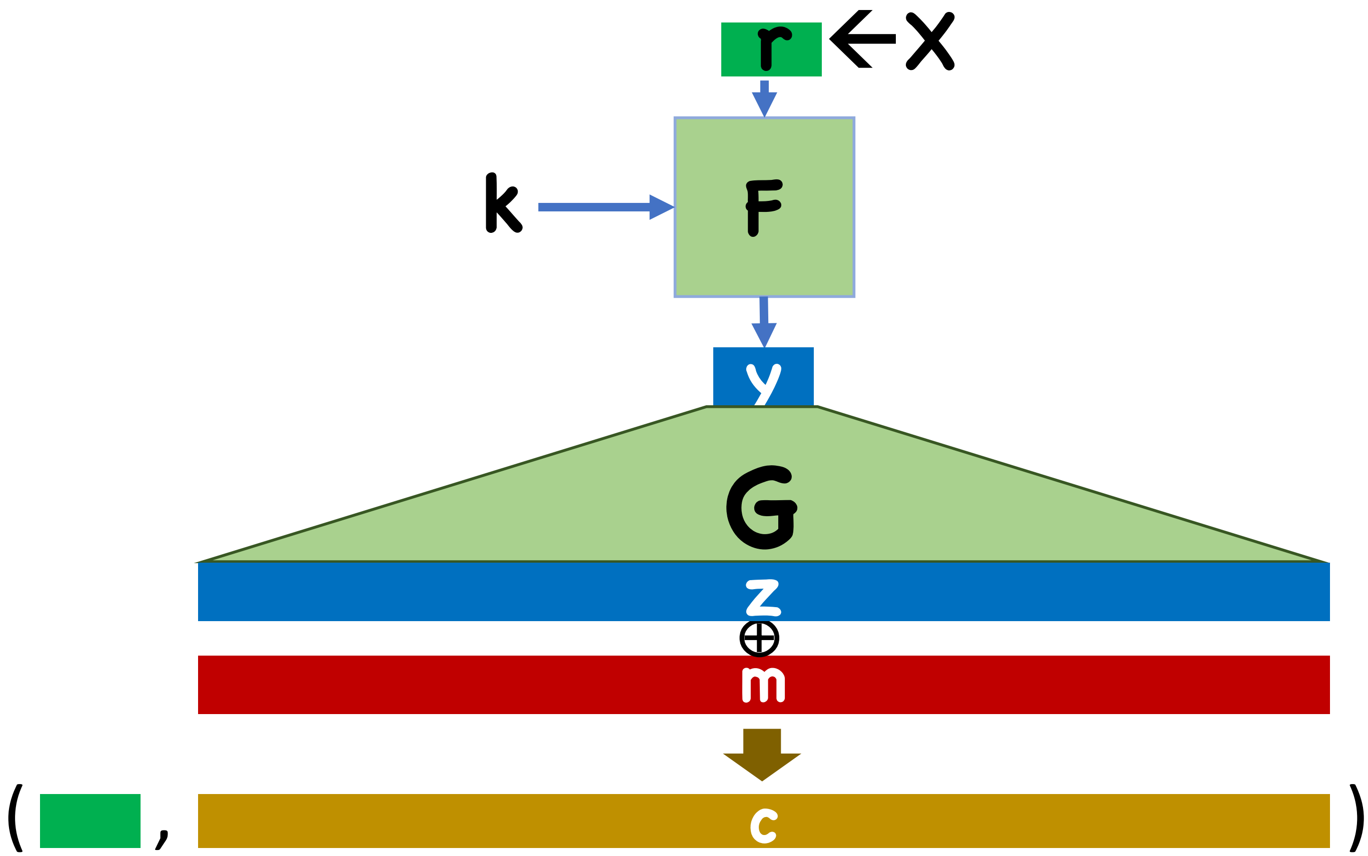
Enc(k, m):

- 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 $r \leftarrow \{0,1\}^{\lambda/2}$
- 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 $c=(c_1,\ldots,c_{|m|})$

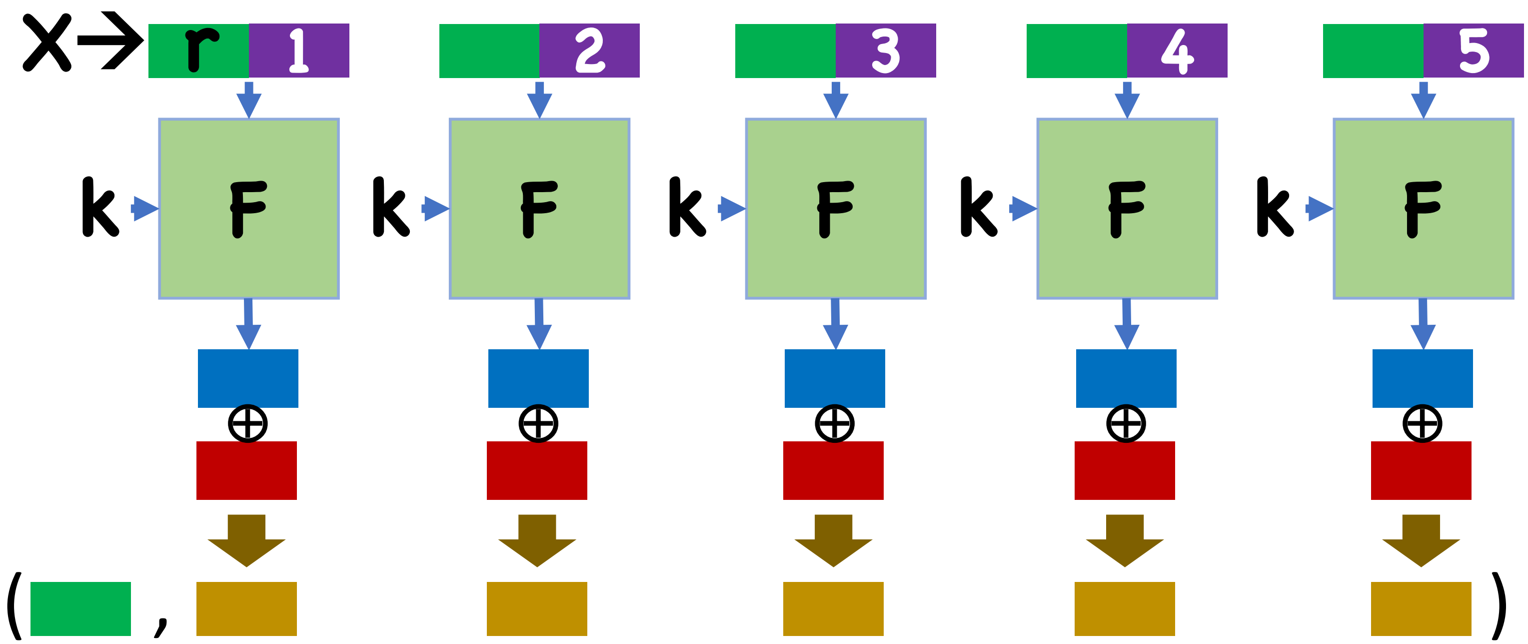
Write i as $\lambda/2$-bit string

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 Modes of Operation