COS433/Math 473: Cryptography

Mark Zhandry

Princeton University

Spring 2017

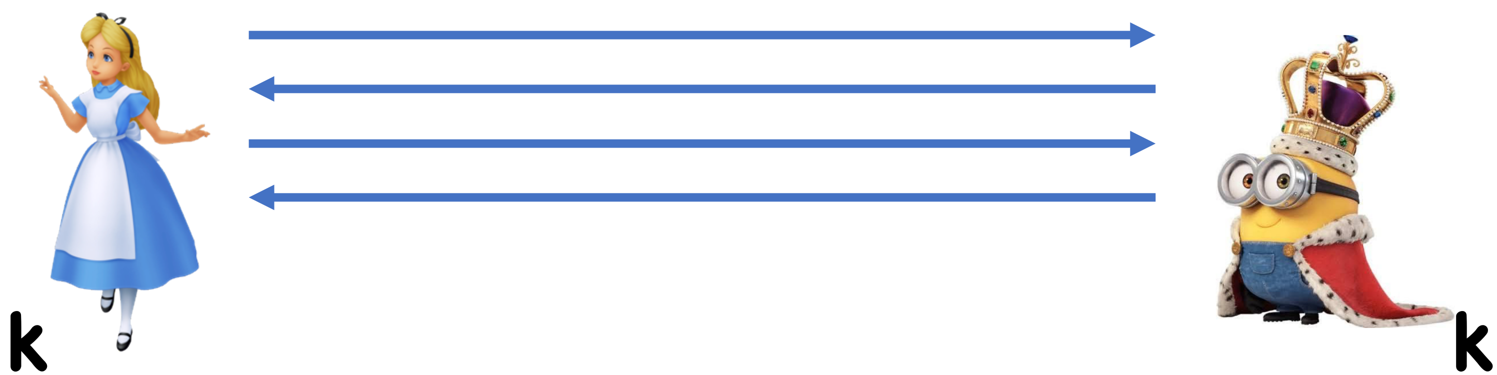
Previously

Exchanging keys and public key encryption

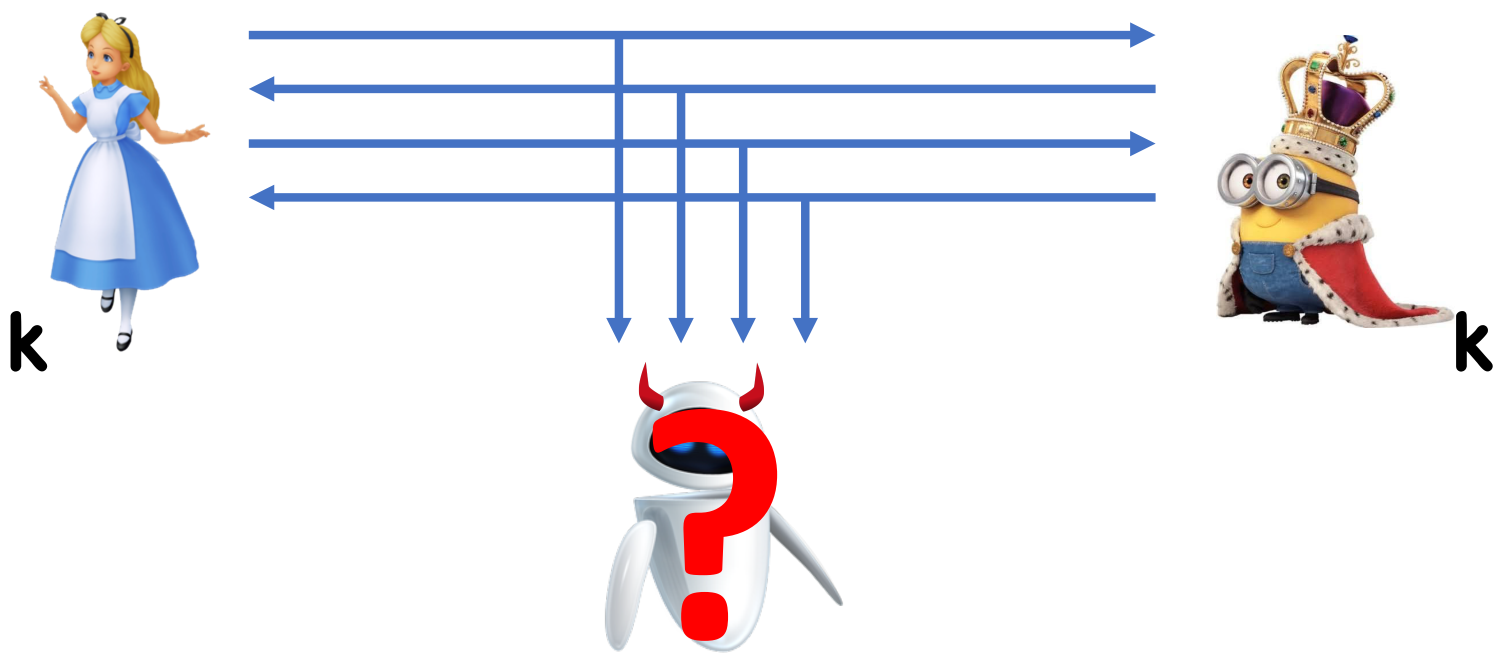
Public Key Distribution

Public Key Distribution

Public Key Distribution

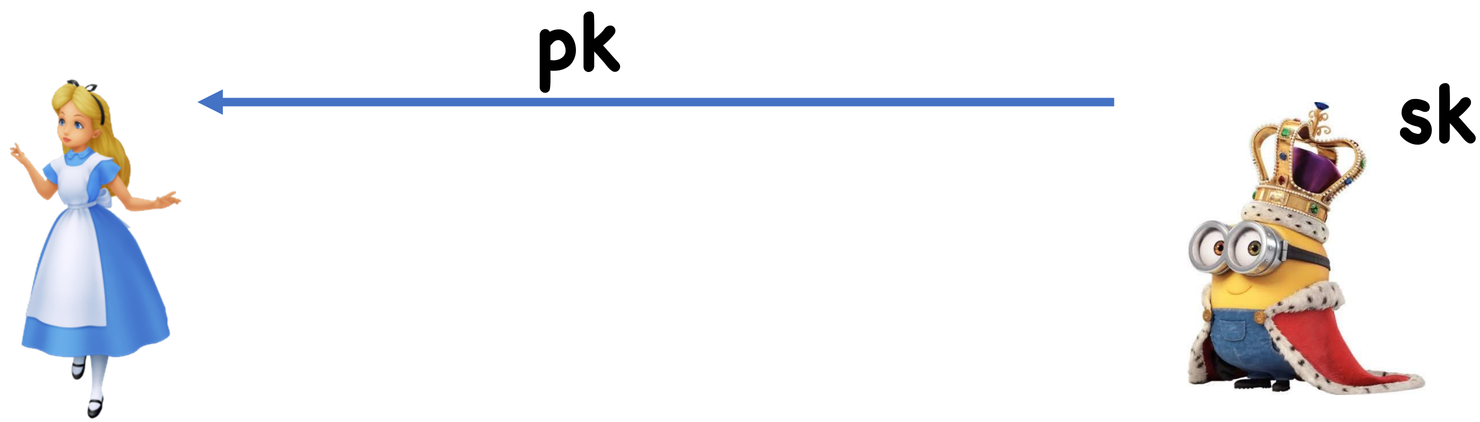


Public Key Distribution



Public Key Encryption

Public Key Encryption



Public Key Encryption

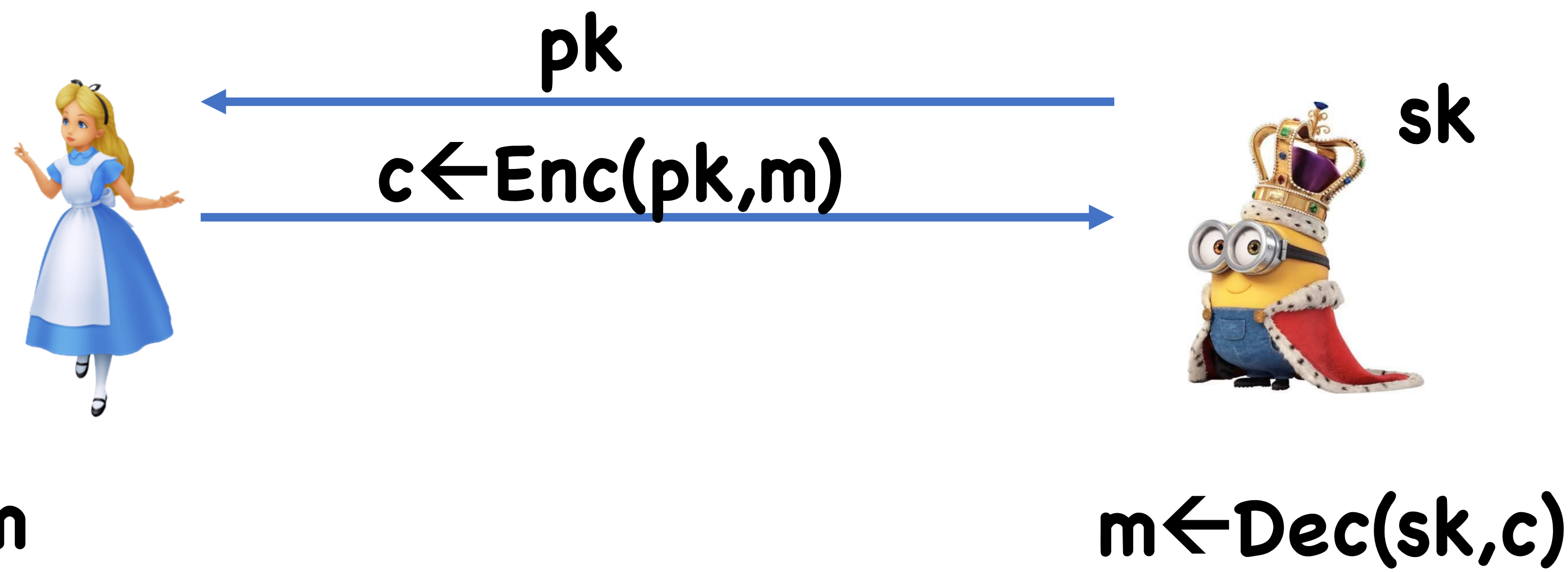
pk

$c \leftarrow Enc(pk, m)$

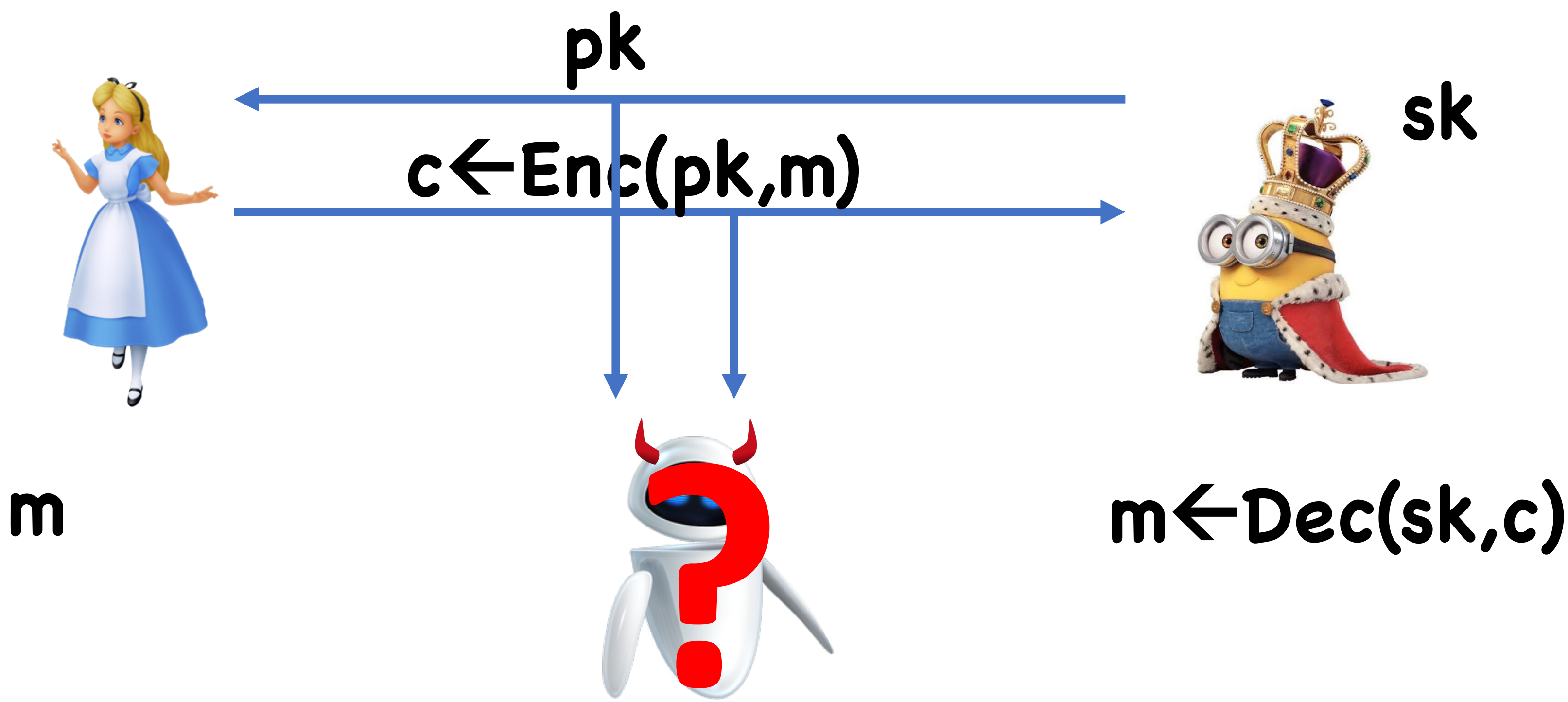
sk

m

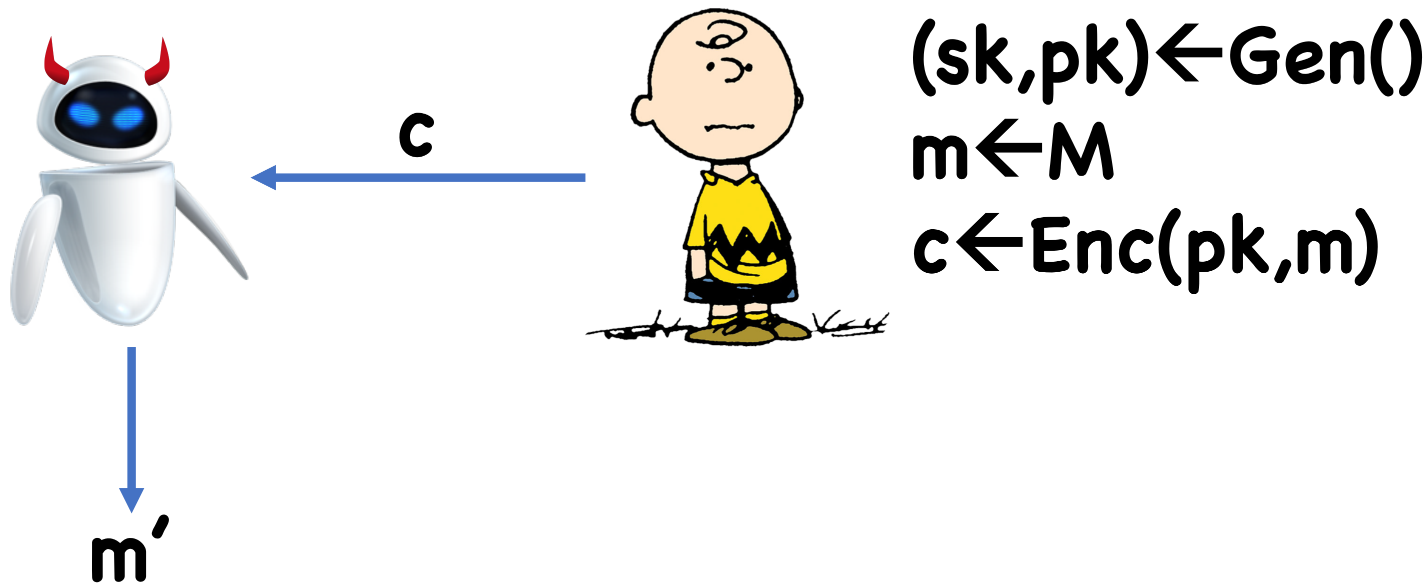
Public Key Encryption



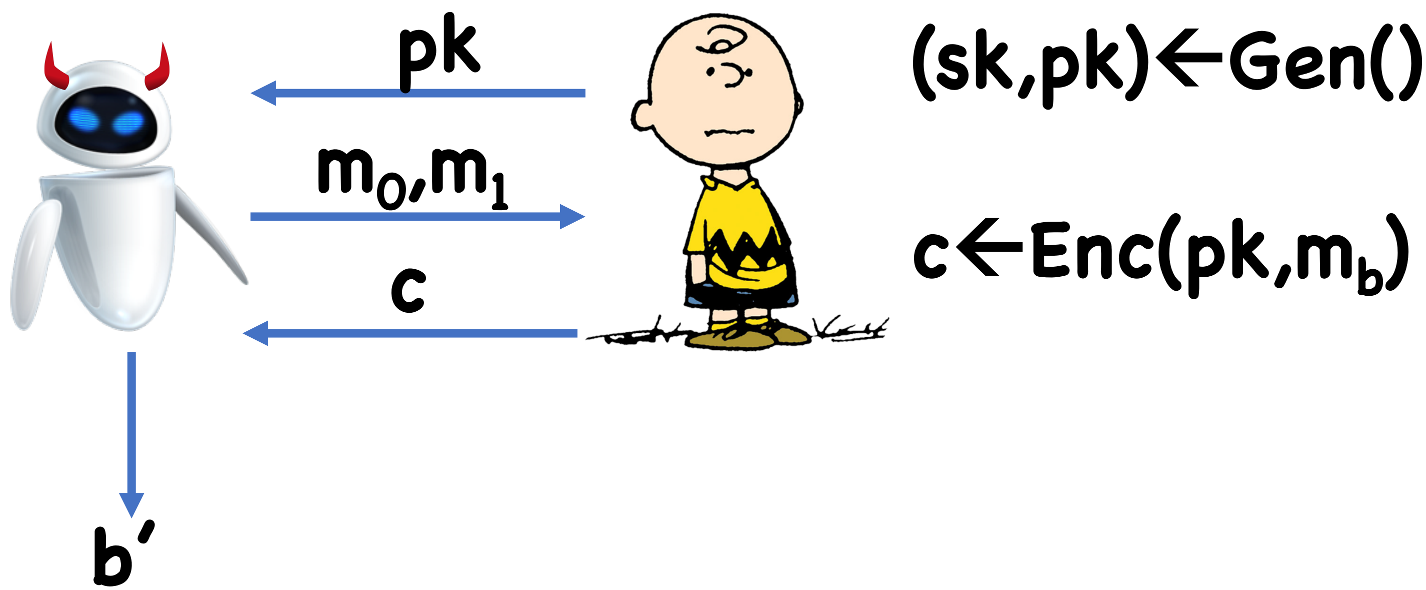
Public Key Encryption



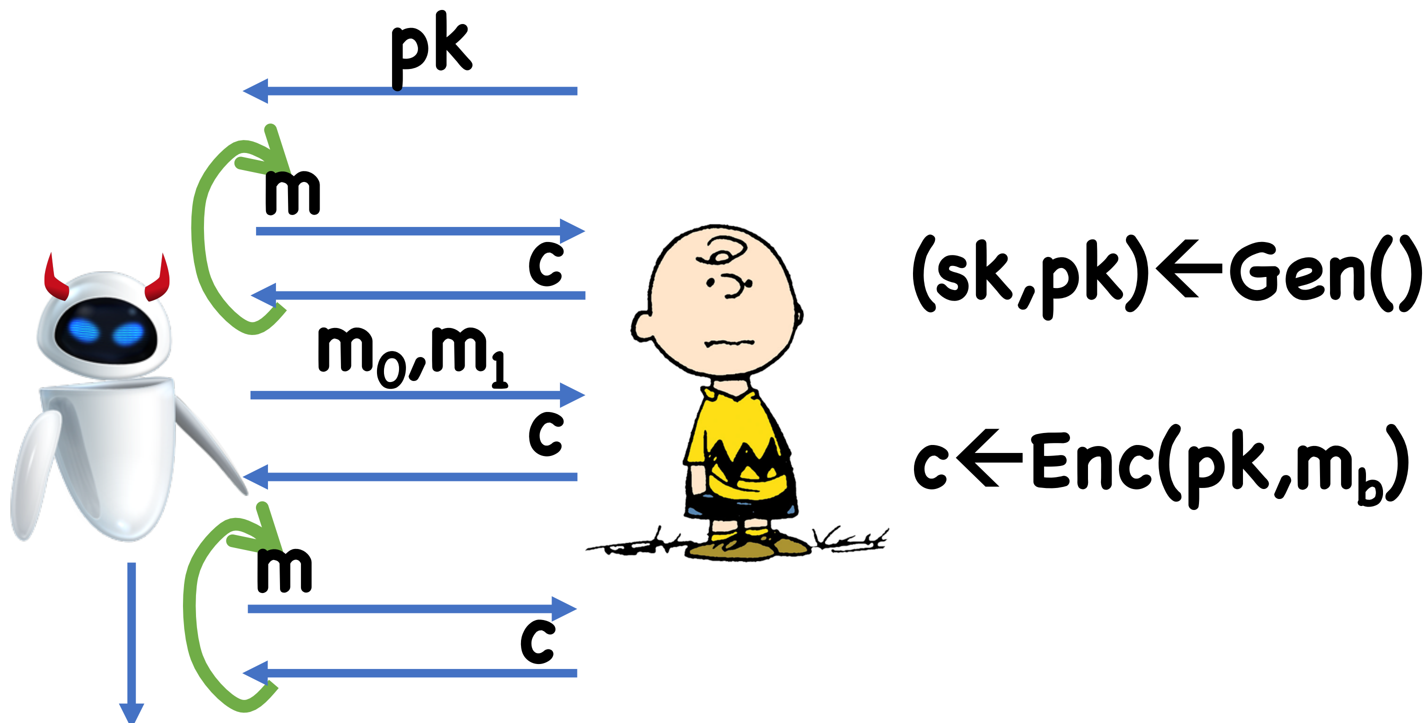
One-way Security



Semantic Security

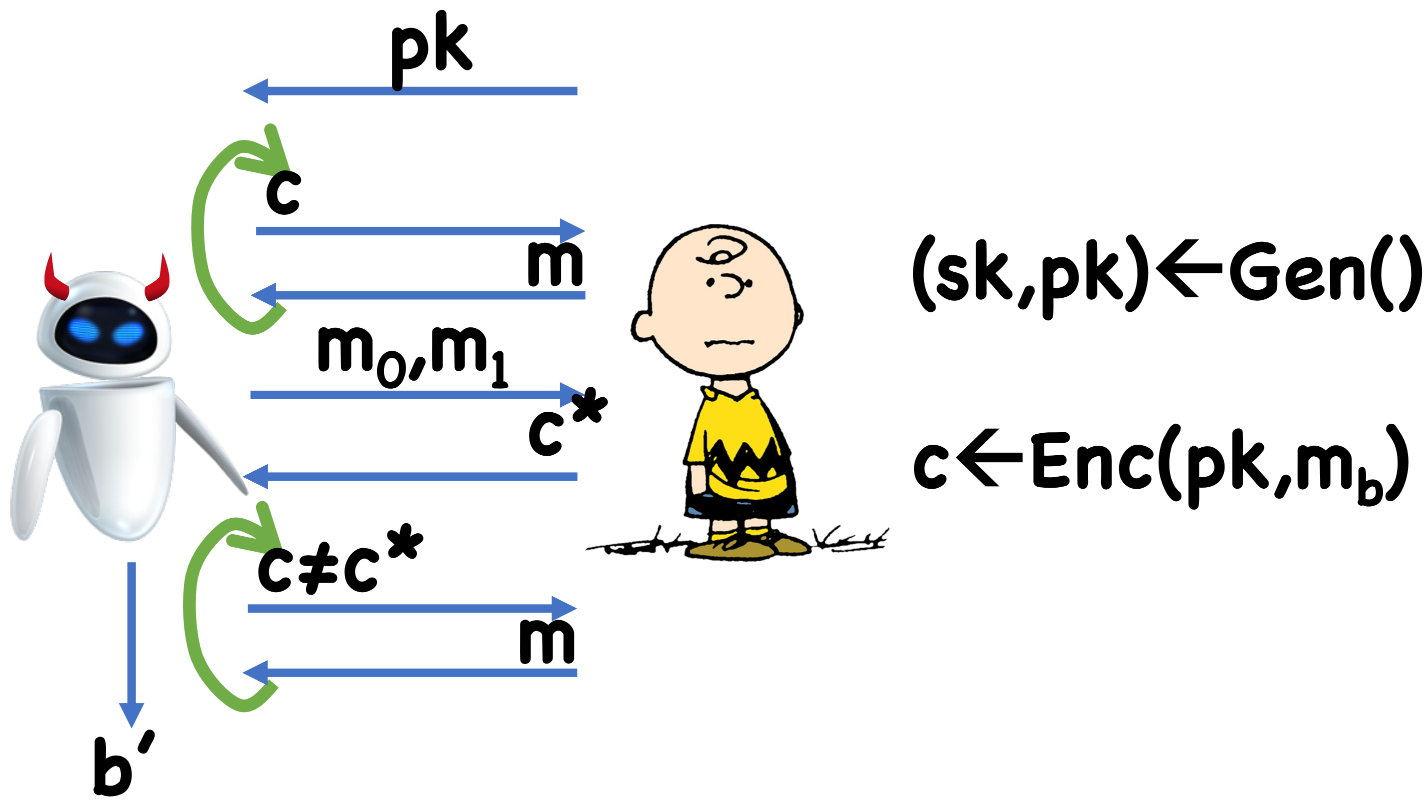


CPA Security



Theorem: An encryption scheme **(Gen,Enc,Dec)** is semantically secure if and only if it is CPA secure

CCA Security



One-way Encryption from RSA

Gen():

- Choose random primes **p,q**
- Let **N=pq**
- Choose **e,d** .s.t **ed=1 mod (p-1)(q-1)**
- Output **pk=(N,e), sk=(N,d)**

Enc(pk,m): Output $c = m^e \bmod N$

Dec(sk,c): Output $m' = c^d \bmod N$

Goldwasser-Micali

Gen():

- Choose random primes **p,q**
- Let **N=pq**
- Choose **x** a quadratic non-residue mod **p** and **q**
- Output **pk=(N,x), sk=(p,q)**

Enc(pk,m$\in${0,1}): $r \leftarrow \mathbb{Z}_N^*$, $c \leftarrow x^m r^2$ mod N

- If **m=0**, then c is a quadratic residue
- If **m=1**, then c is a non-residue

ElGamal

Group **G** of order **p**, generator **g**
Message space = **G**

Gen():
- Choose random $a \leftarrow \mathbb{Z}_p^*$, let $h \leftarrow g^a$
- $pk = h$, $sk = a$

Enc(pk,m∈{0,1}):
- $r \leftarrow \mathbb{Z}_p$
- $c = (g^r, h^r \times m)$

Dec?

Today

Trapdoor Permutations: abstracting RSA

CCA-secure Encryption in the ROM

Begin: digital signatures

Trapdoor Permutations

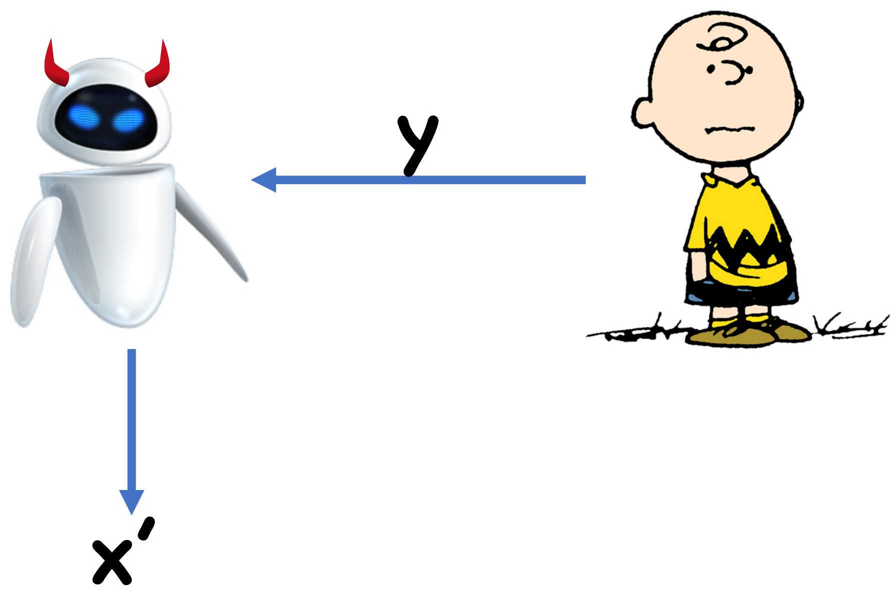
Domain $\mathbf{X}$

Gen(): outputs **(pk,sk)**
$\mathbf{F(pk, x \in X) = y \in X}$
$\mathbf{F^{-1}(sk, y) = x}$

Correctness:
$\mathbf{Pr[\ F^{-1}(sk,\ F(pk,\ x)) = x : (pk,sk) \leftarrow Gen()\ ] = 1}$

Correctness implies $\mathbf{F, F^{-1}}$ are deterministic, permutations

Trapdoor Permutation Security



(sk,pk)←Gen()
x←X
y←F(pk,x)

One-way Encryption from TDPs

$\mathbf{Gen_{Enc}() = Gen()}$

$\mathbf{Enc(pk,x) = F(pk, x)}$

$\mathbf{Dec(sk,c) = F^{-1}(sk, c)}$

Thus, TDPs are special case of one-way encryption where **Enc** is deterministic and $\mathbf{C = M}$

CPA-Secure Encryption from TDPs

Let **h** be a hardcore bit for the one-way function
$x \rightarrow F(pk,x)$

$\mathrm{Enc}(pk,b) = F(pk,r),\ h(r) \oplus b$

Constructing TDPs with hardcore bits?

- $F'(pk,\ (r,x)\ ) = (r\ ,\ F(pk,x))$
- $h(r,x) = r \oplus b$

Trapdoor Permutations from RSA

Gen():

- Choose random primes $\mathbf{p,q}$
- Let $\mathbf{N=pq}$
- Choose $\mathbf{e,d}$ .s.t $\mathbf{ed=1 \bmod (p-1)(q-1)}$
- Output $\mathbf{pk=(N,e),\ sk=(N,d)}$

F(pk,x): Output $\mathbf{y = x^e \bmod N}$

$\mathbf{F^{-1}}$(sk,c): Output $\mathbf{x = y^d \bmod N}$

Caveats

RSA is not a true TDP as defined

- Why???
- What’s the domain?

Nonetheless, distinction is not crucial to most applications

Other TDPs?

For long time, none known

- Still interesting object:
    - Useful abstraction in protocol design
    - Maybe more will be discovered…

Using obfuscation:

- Let P be a PRP
- $\mathbf{sk = k,\ pk = Obf(\ P(k, \cdot\ )\ )}$

Relaxation: Injective Trapdoor Functions

Domain $\mathbf{X}$, range $\mathbf{Y}$

$\mathbf{Gen()}$: outputs $\mathbf{(pk,sk)}$
$\mathbf{F(pk,x \in X) = y \in Y}$, deterministic
$\mathbf{F^{-1}(sk,y) = x}$

Correctness:
$\mathbf{Pr[\ F^{-1}(sk,\ F(pk,\ x)) = x : (pk,sk) \leftarrow Gen()\ ] = 1}$

Correctness implies $\mathbf{F}$ is injective

Injective TDFs from DH

Notation:

Let $\mathbf{A} \in \mathbb{Z}_p^{\mathbf{n\times n}}$

$\mathbf{g^A \in G^{n\times n}, \ (g^A)_{i,j} := g^{A_{i,j}}}$

Let $\mathbf{H} \in \mathbf{G^{n\times n}}, \ \mathbf{v} \in \mathbb{Z}_p^{\mathbf{n}}$

$\mathbf{H^v \in G^n, \ (H^v)_i := \Pi_j \ H_{i,j}{}^{v_j}}$

Note: $\mathbf{((g^A)^v)_i = \Pi_j \ g^{A_{i,j} v_j} = g^{(A\cdot v)_i}}$, so $\mathbf{(g^A)^v = g^{A\cdot v}}$

Injective TDFs from DH

Notation:

Let $\mathbf{h} \in \mathbf{G}^{\mathbf{n}}$, $\mathbf{A} \in \mathbb{Z}_{\mathbf{q}}^{\mathbf{n \times n}}$

$^{\mathbf{A}}\mathbf{h} \in \mathbf{G}^{\mathbf{n}}$, $(^{\mathbf{A}}\mathbf{h})_{\mathbf{i}} := \prod_{\mathbf{j}} \mathbf{h}_{\mathbf{j}}^{\mathbf{A}_{\mathbf{i,j}}}$

$^{\mathbf{A}}(\mathbf{g}^{\mathbf{v}})$?

$(^{\mathbf{A}}(\mathbf{g}^{\mathbf{v}}))_{\mathbf{i}} = \prod_{\mathbf{j}} \mathbf{g}^{\mathbf{A}_{\mathbf{i,j}}\mathbf{v}_{\mathbf{j}}} = \mathbf{g}^{(\mathbf{A \cdot v})_{\mathbf{i}}}$, so $^{\mathbf{A}}(\mathbf{g}^{\mathbf{v}}) = \mathbf{g}^{\mathbf{A \cdot v}}$

Injective TDFs from DH

First Attempt:

Gen(): choose random $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times n}$

$\mathbf{sk} = \mathbf{A},\ \mathbf{pk} = \mathbf{H} = g^{\mathbf{A}}$

$\mathbf{F(pk,\ x} \in \mathbb{Z}_q^{\mathbf{n}}\mathbf{)} = \mathbf{H}^{\mathbf{x}}\ (= g^{\mathbf{A} \cdot \mathbf{x}})$

$\mathbf{F^{-1}(sk,\ h)}$: $\mathbf{y} \leftarrow {}^{\mathbf{A}^{-1}}\mathbf{h}\ (= g^{\mathbf{A}^{-1} \cdot \mathbf{A} \cdot \mathbf{x}} = g^{\mathbf{x}})$

Then Dlog?????

Injective TDFs from DH

Theorem: If DDH holds, then (Gen,F,F^{-1}) is an injective TDF

Gen(): choose random $A \leftarrow \mathbb{Z}_q^{n\times n}$

$sk = A,\ pk = H = g^A$

$F(pk,\ x \in \{0,1\}^n) = H^x\ (= g^{A\cdot x})$

$F^{-1}(sk,\ h)$: $y \leftarrow {}^{A^{-1}}h\ (= g^{A^{-1}\cdot A\cdot x} = g^x\ )$

Then Dlog each component to recover x

Injective TDFs

Known constructions from most number theoretic problems

Useful in many cases where TDPs are used (but not all)

CCA Secure PKE from TDPs

Let $(\mathsf{Enc}_{\mathsf{SKE}},\mathsf{Dec}_{\mathsf{SKE}})$ be a CCA-secure secret key encryption scheme.

Let $(\mathsf{Gen},\mathsf{F},\mathsf{F}^{-1})$ be a TDP

Let H be a hash function (we'll pretend it's a random oracle)

CCA Secure PKE from TDPs

$\mathbf{Gen_{PKE}() = Gen()}$

$\mathbf{Enc_{PKE}(pk, m)}$:

- Choose random $\mathbf{r}$
- Let $\mathbf{c_0 \leftarrow F(pk,r)}$
- Let $\mathbf{c_1 \leftarrow Enc_{SKE}(H(r), m)}$
- Output $\mathbf{(c_0,c_1)}$

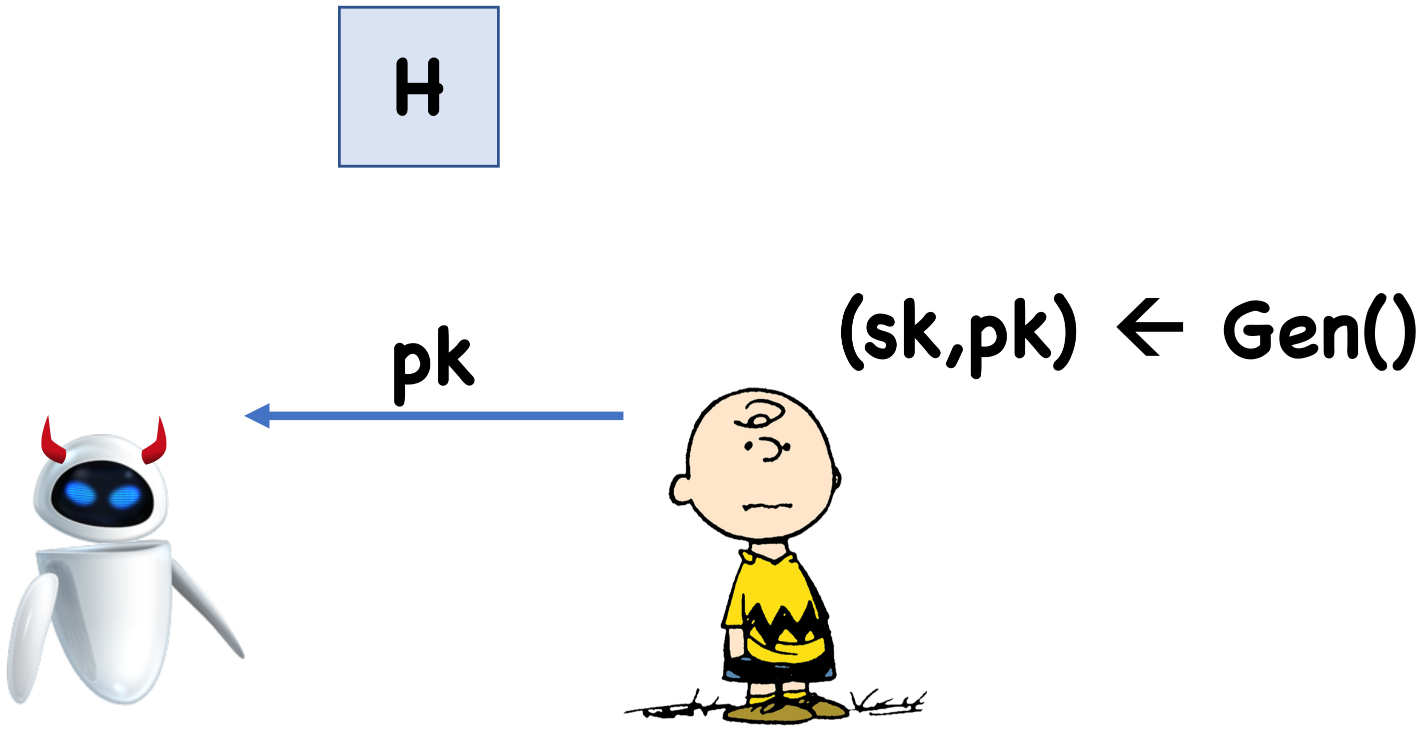
$\mathbf{Dec_{PKE}(sk, (c_0, c_1))}$:

- **Let** $\mathbf{r \leftarrow F^{-1}(sk, c_0)}$
- **Let** $\mathbf{m \leftarrow Dec_{SKE}(H(r), c_1)}$

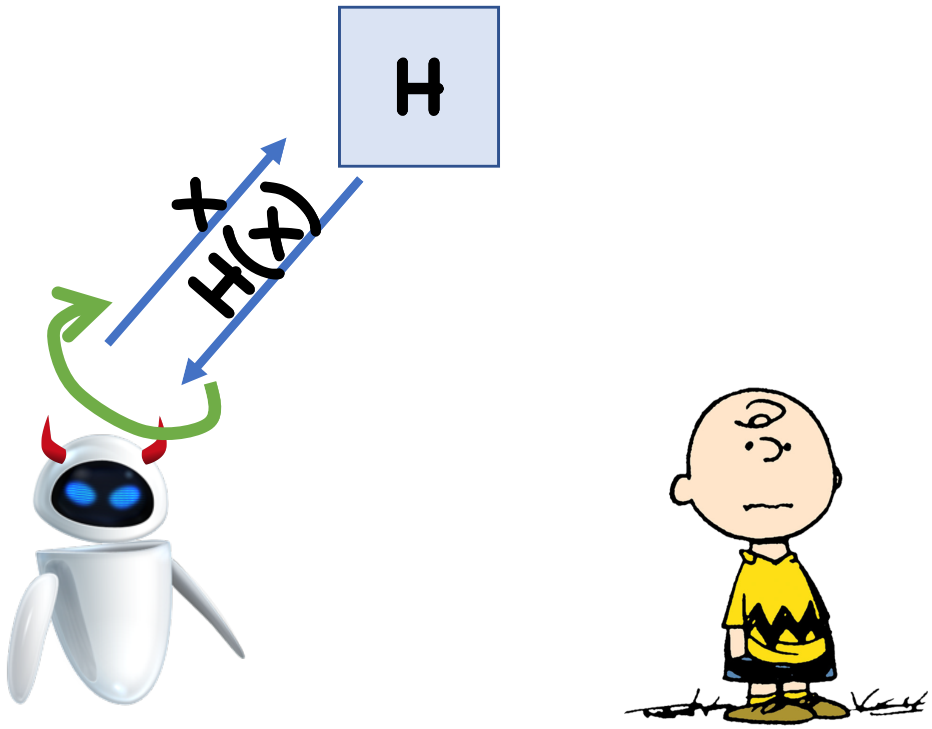
CCA Secure PKE from TDPs

Theorem: If $(\mathsf{Enc}_{SKE},\mathsf{Dec}_{SKE})$ is a CCA-secure secret key encryption scheme, $(\mathsf{Gen},\mathsf{F},\mathsf{F}^{-1})$ is a TDP, and H is modeled as a random oracle, then $(\mathsf{Gen}_{PKE},\mathsf{Enc}_{PKE},\mathsf{Dec}_{PKE})$ is a CCA secure public key encryption scheme

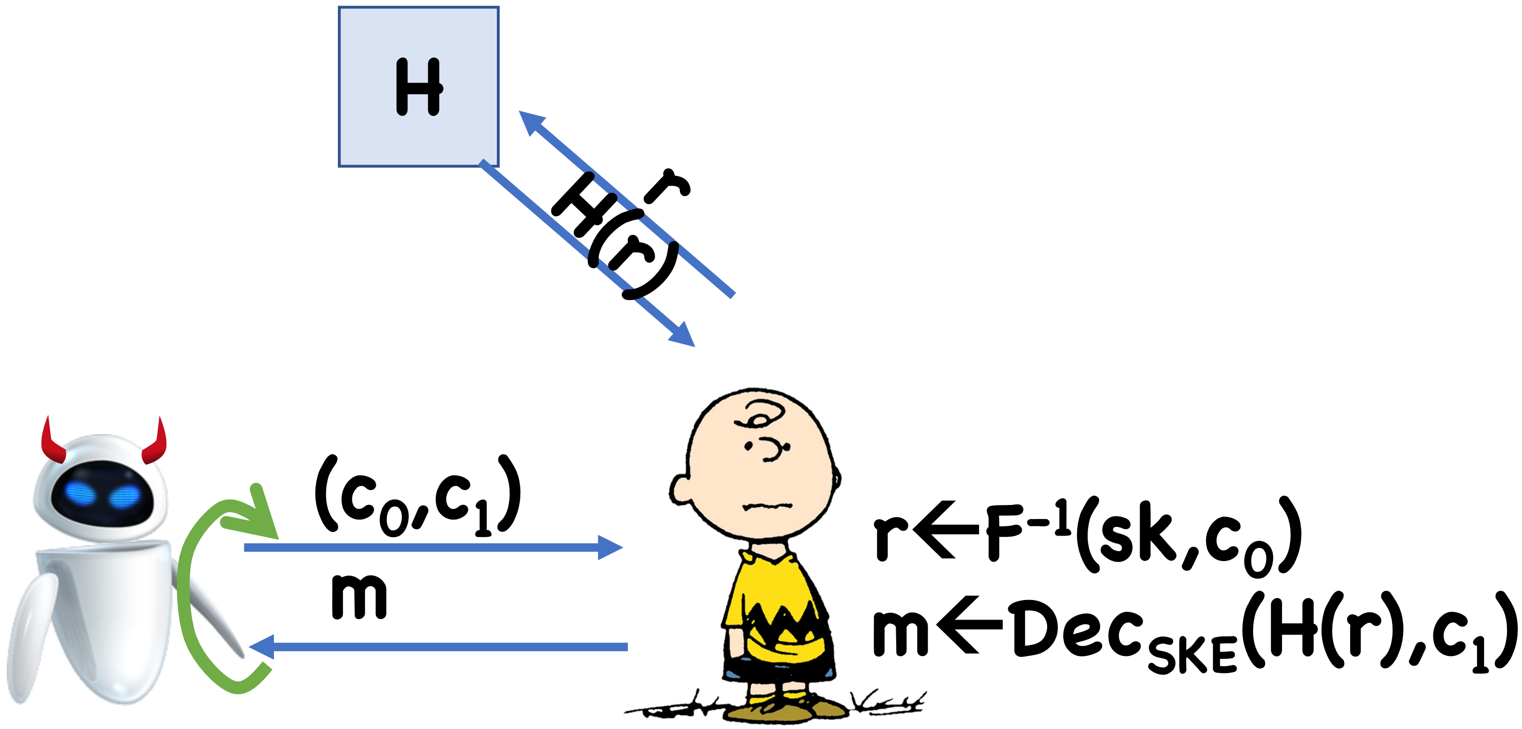
Proof



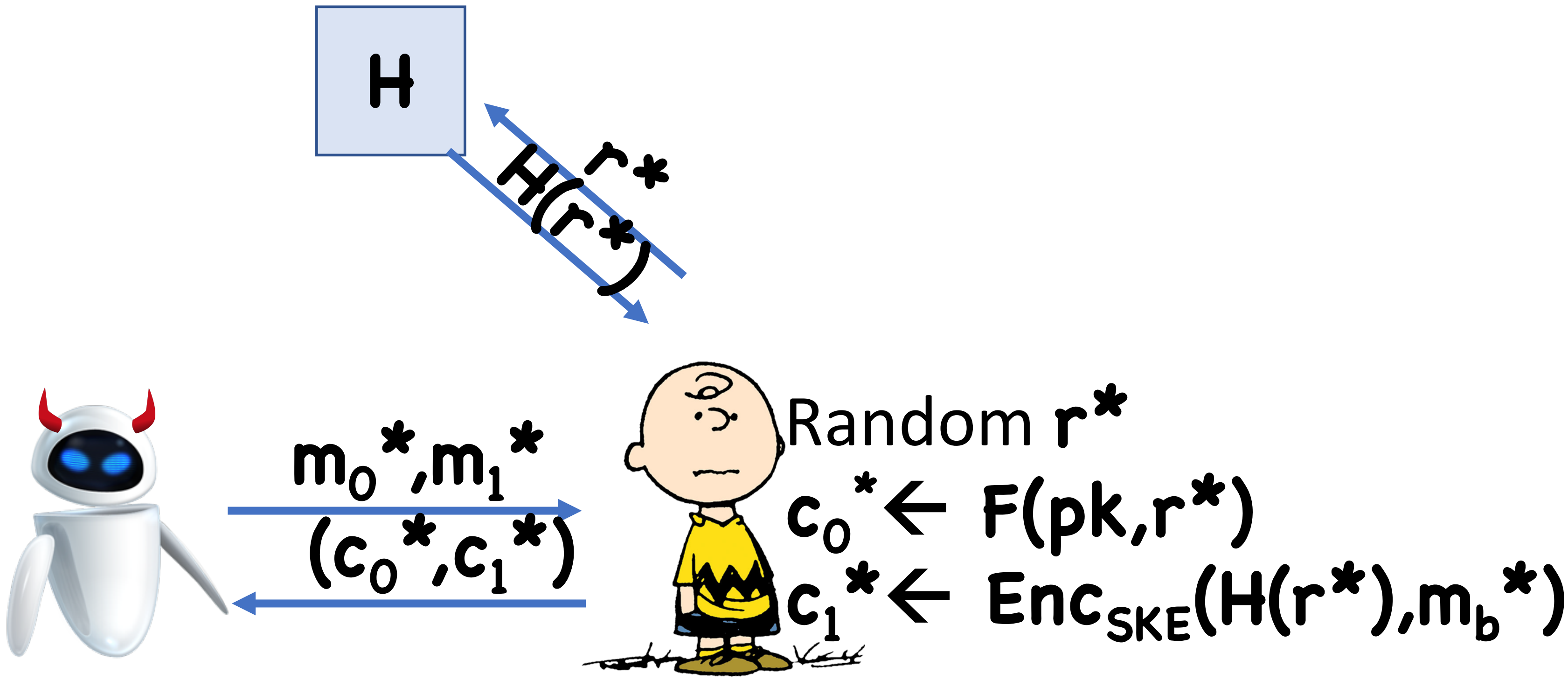
Proof



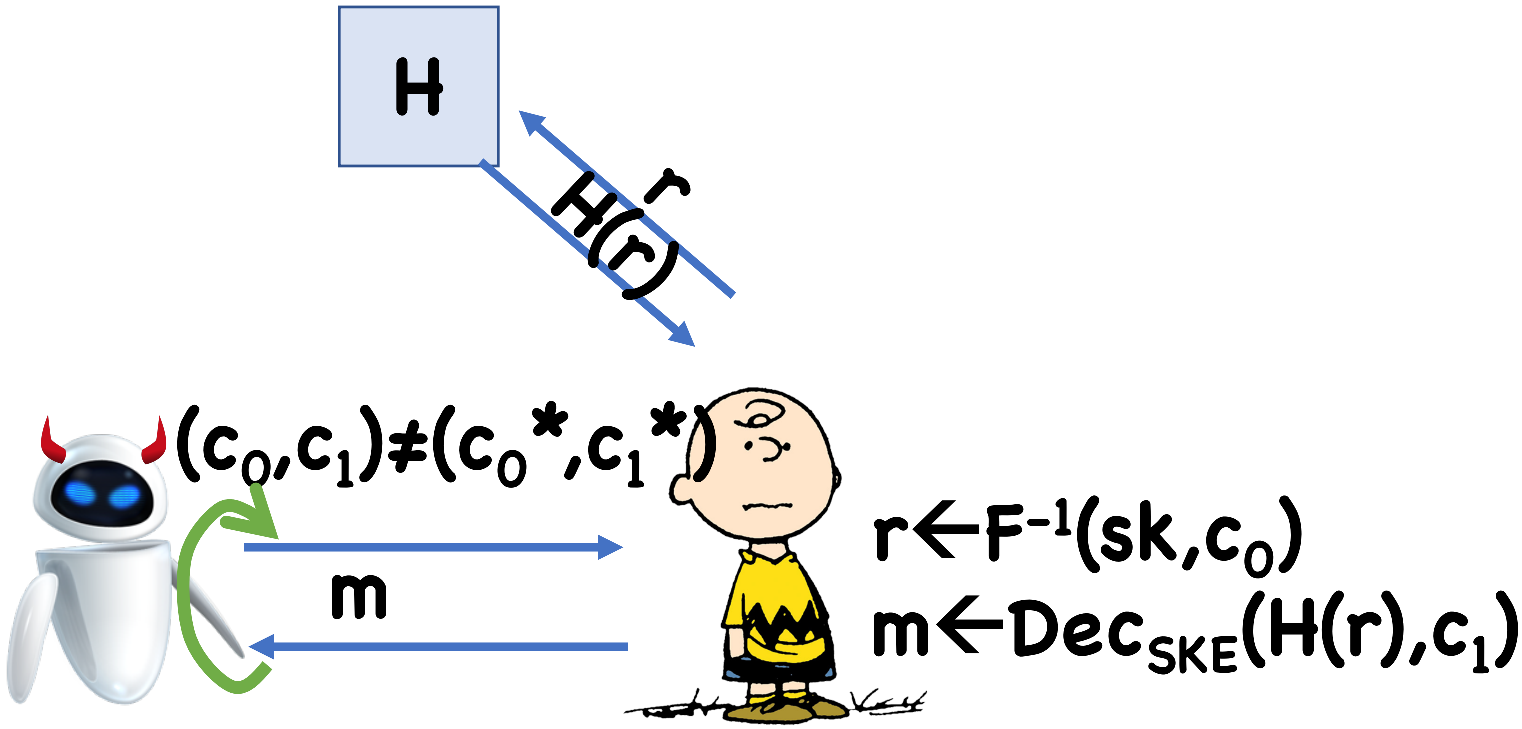
Proof



Proof



Proof



Proof

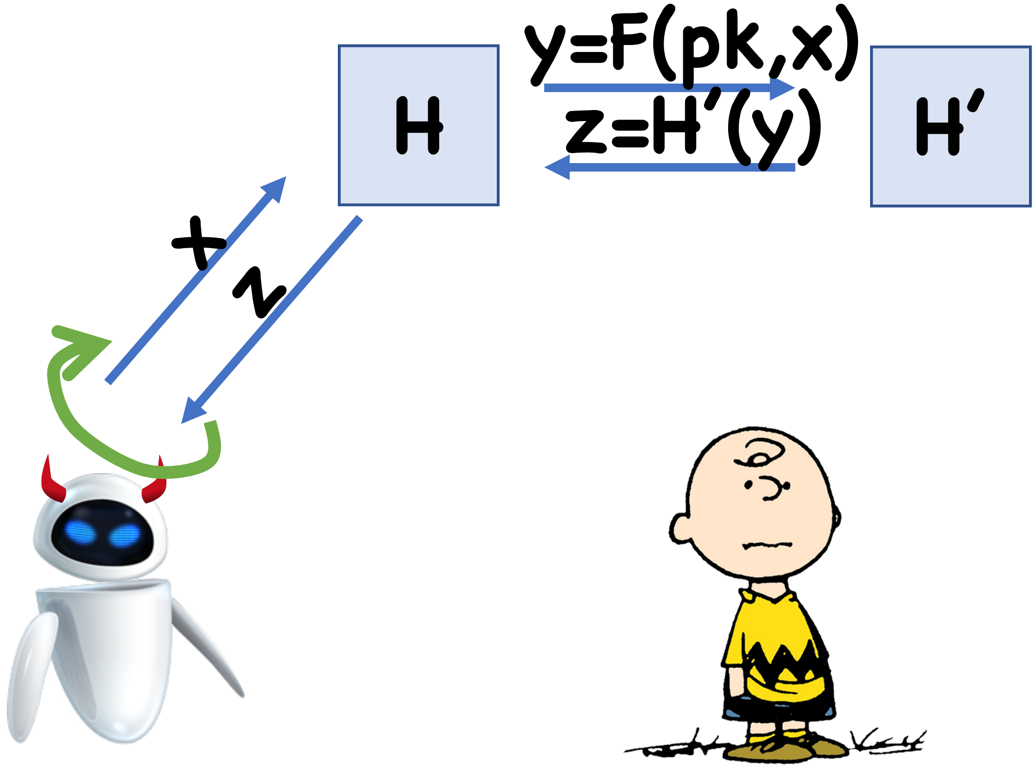
Step 1: sample H as follows:

- Choose a random function H'
- Let $H(x) = H'(\ F(pk,\ x)\ )$

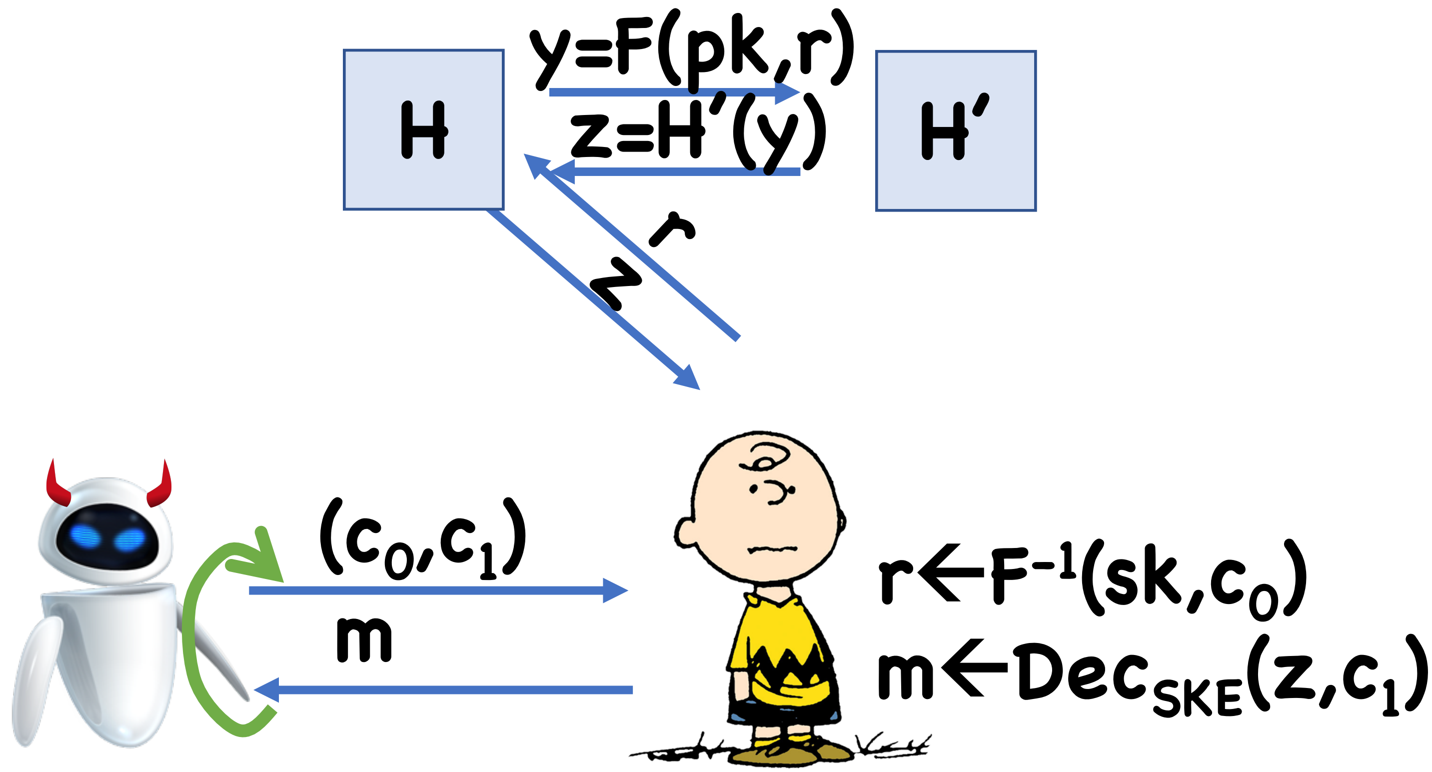
Since $F(pk, \cdot\ )$ is a permutation, all outputs of $H(x)$ are independent and uniform

Therefore, $H(x)$ is still a random oracle

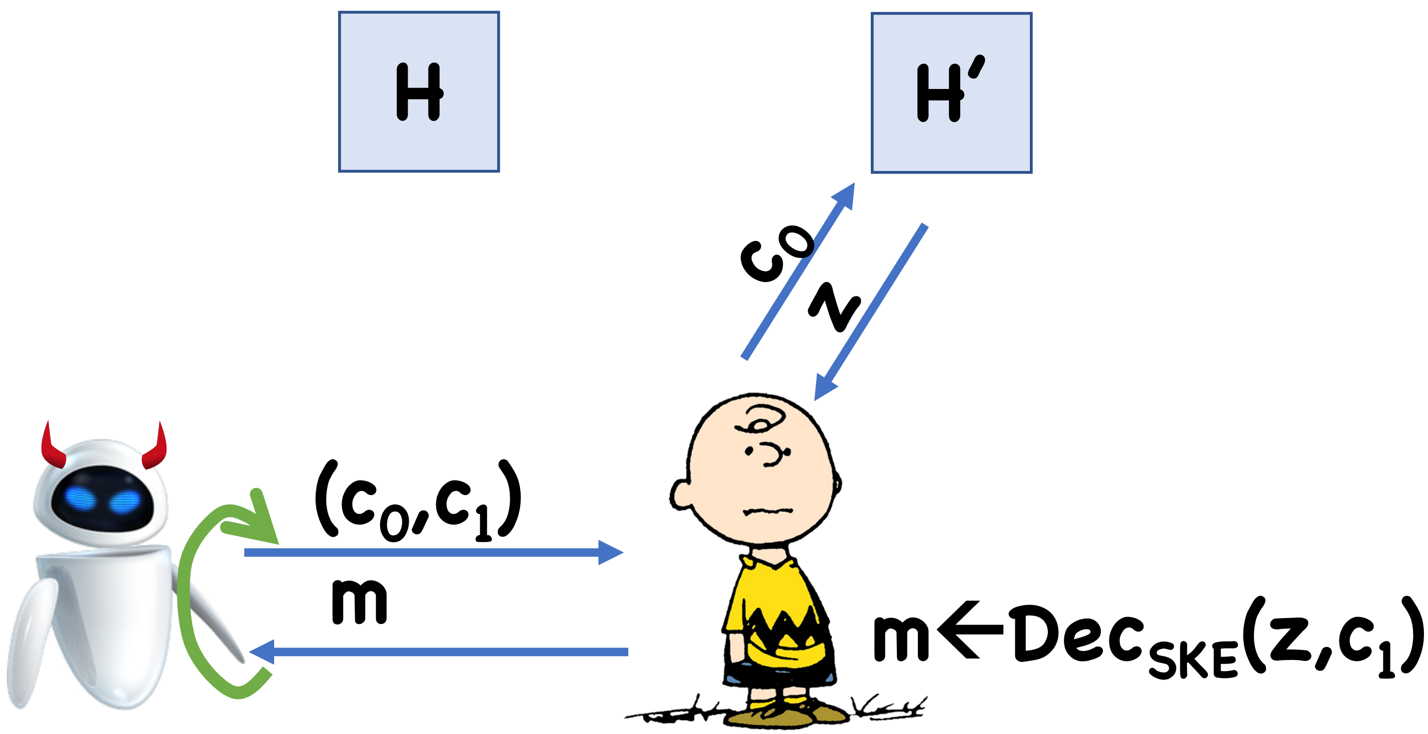
Proof



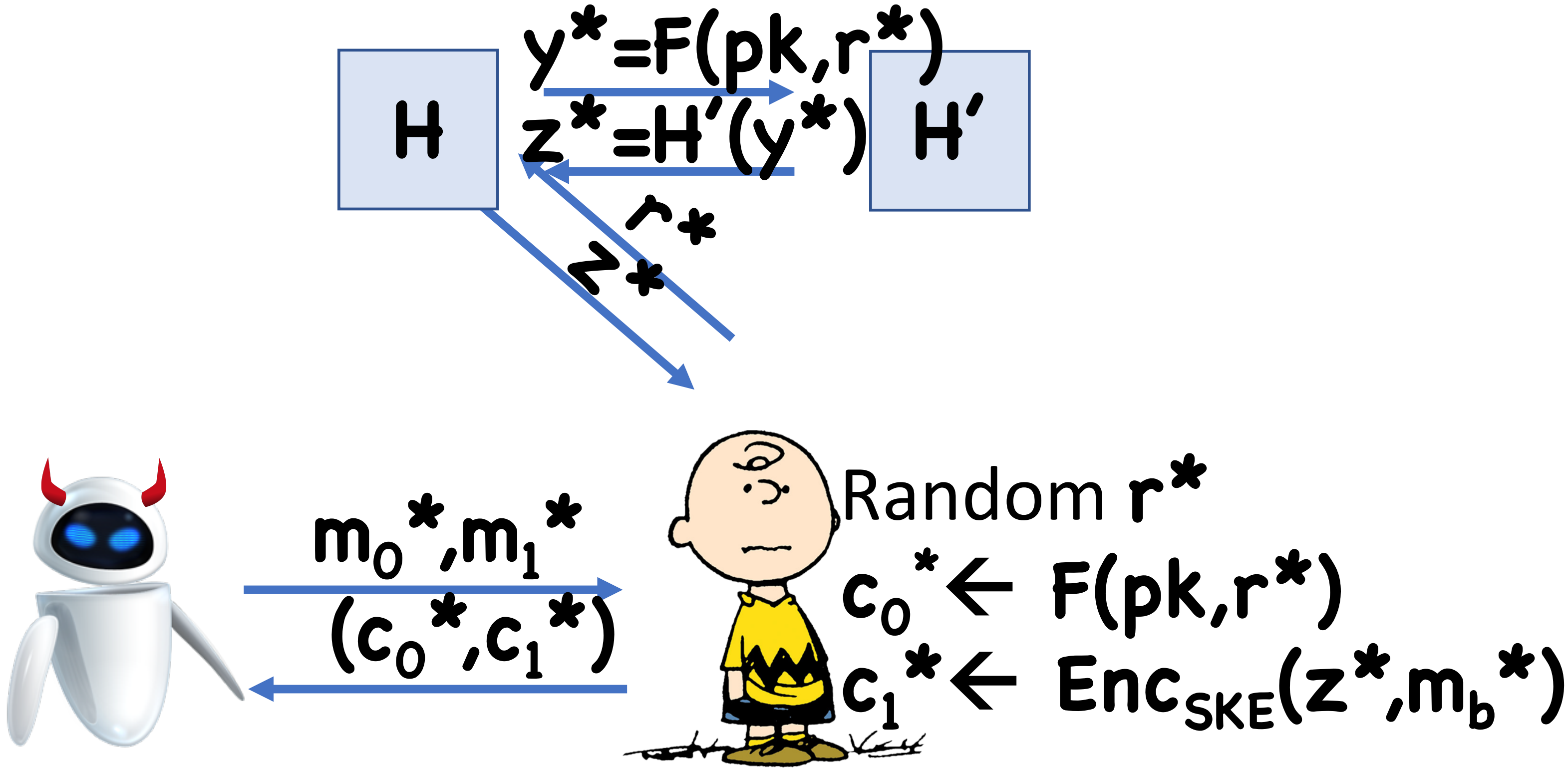
Proof



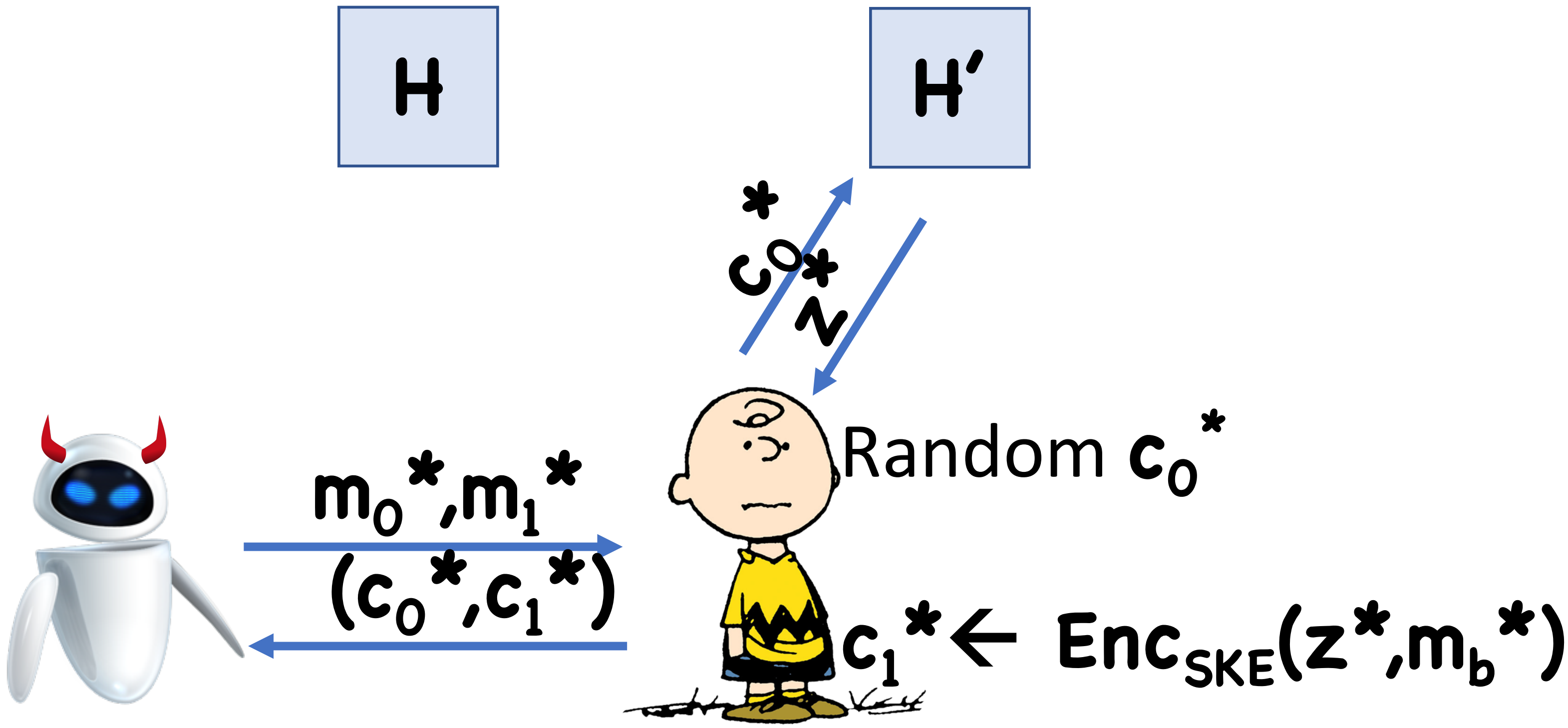
Proof



Proof



Proof



Observation: now Charlie doesn't need **sk** to run experiment

Proof

Consider two cases:

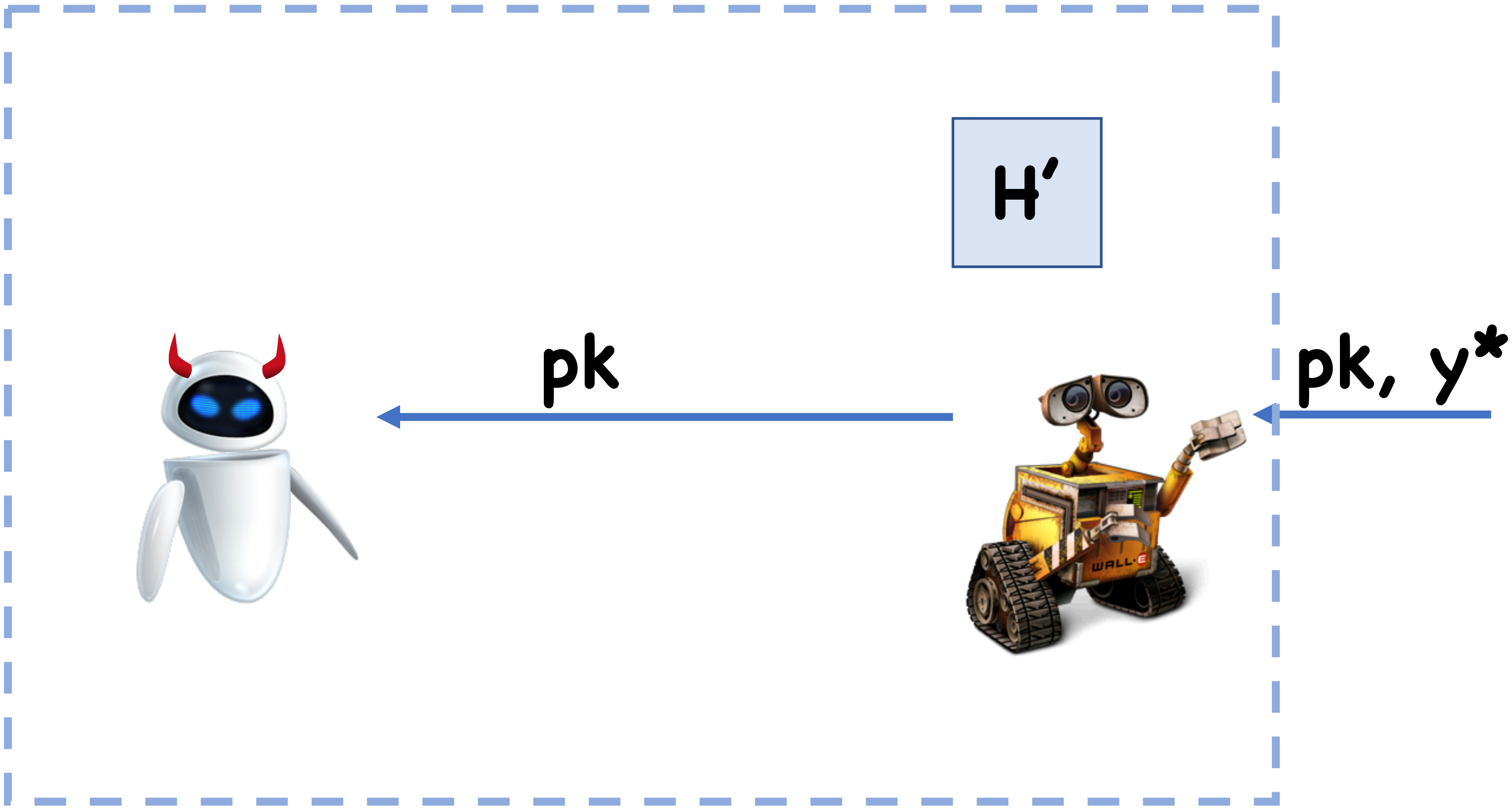
Case 1: adversary makes a RO query to $\mathbf{H}$ on

$$\mathbf{r^* = F^{-1}(sk,\ c_0^*)}$$

Case 2: adversary never makes a RO query on $\mathbf{r^*}$

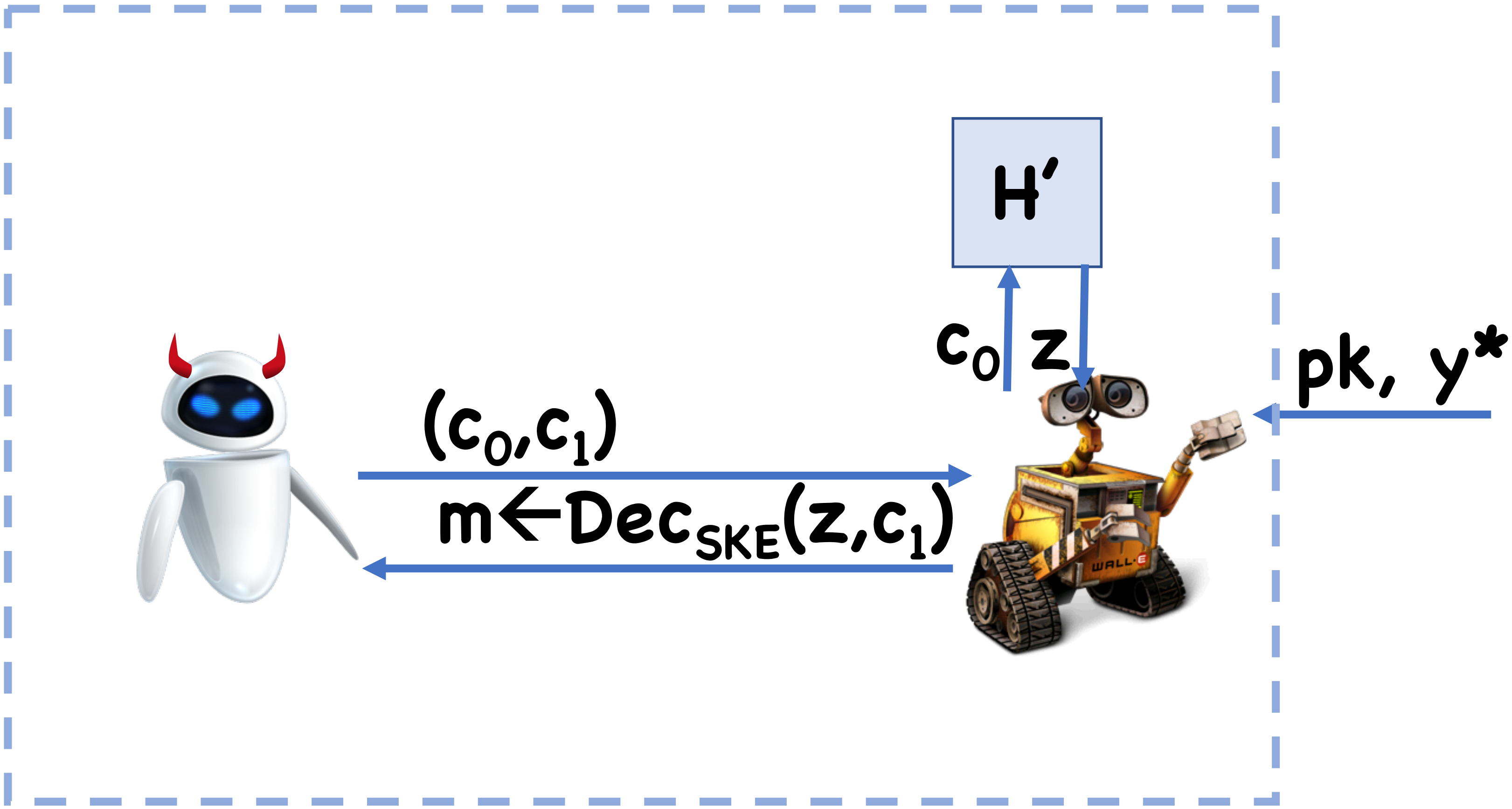
Proof

Case 1: construct TDP adversary 



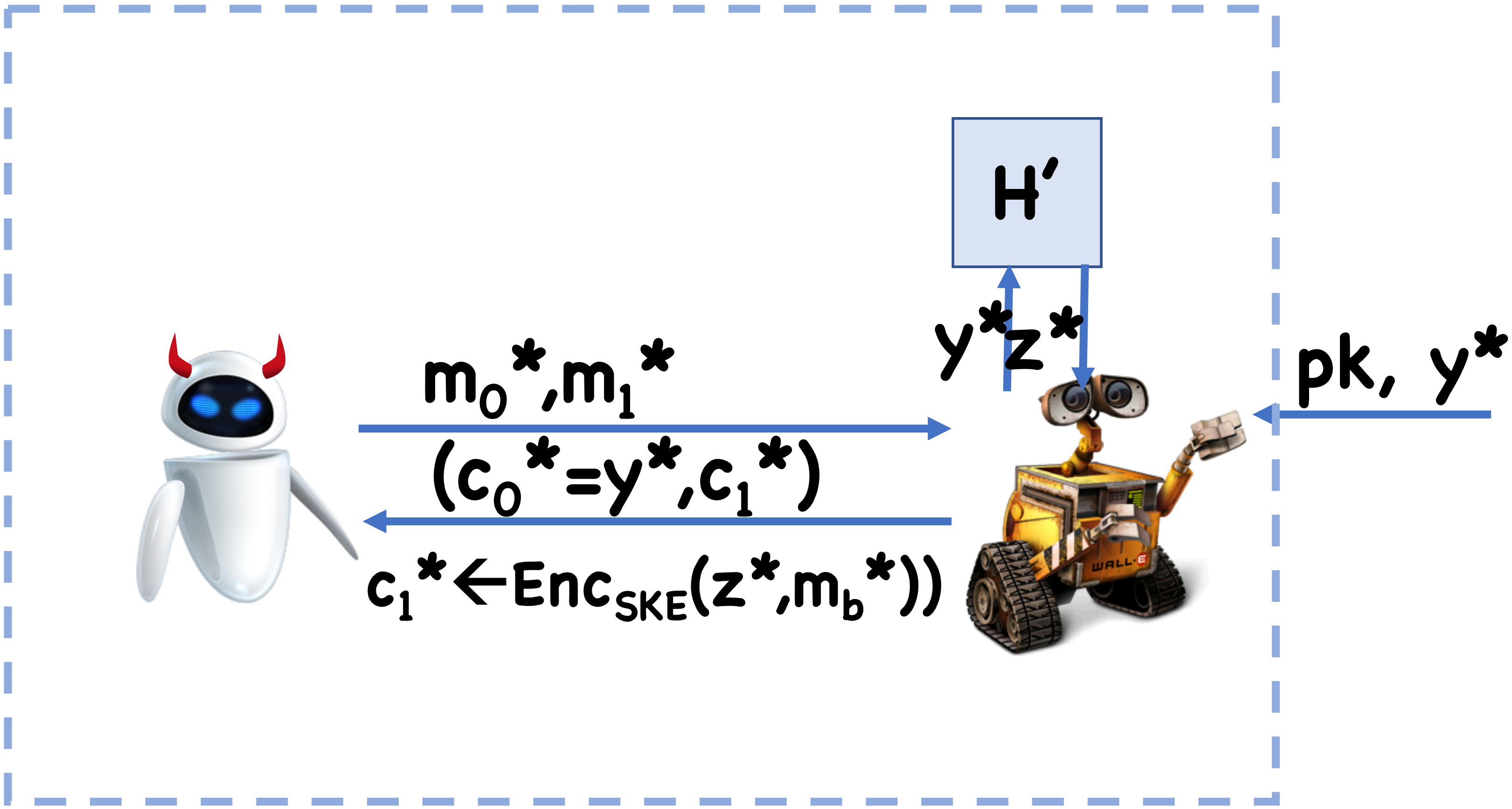
Proof

Case 1: construct TDP adversary 



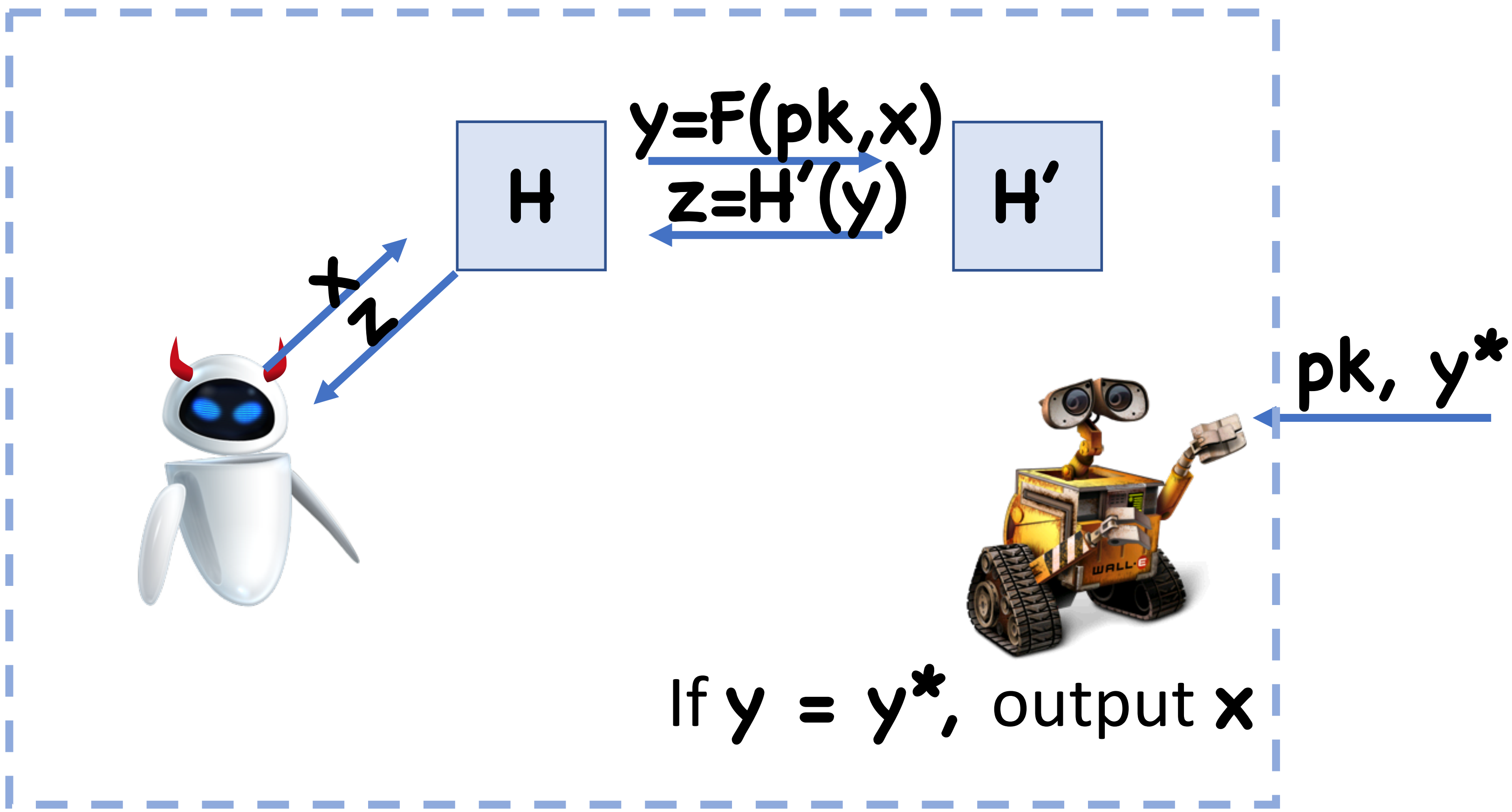
Proof

Case 1: construct TDP adversary 



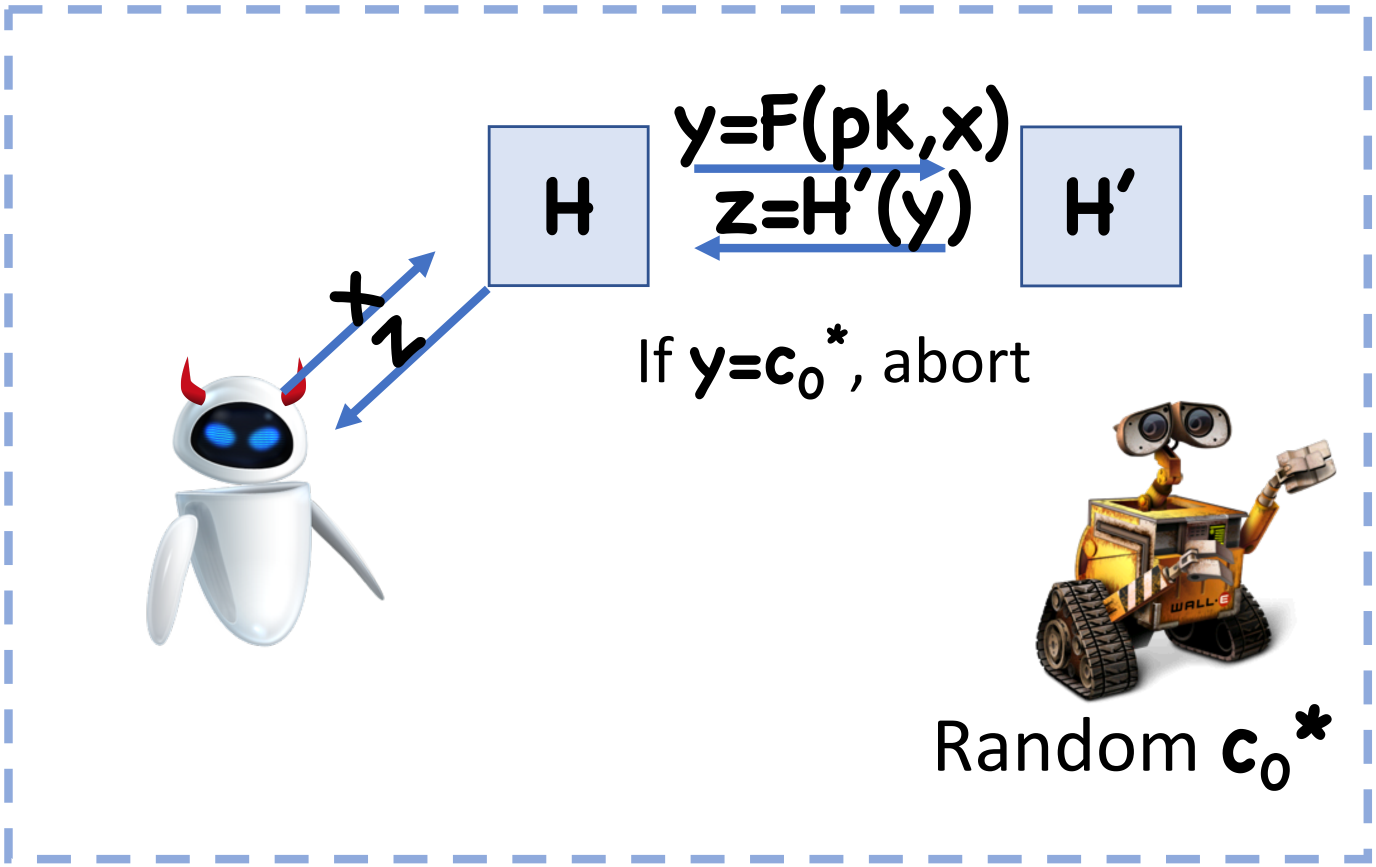
Proof

Case 1: construct TDP adversary 

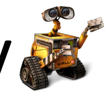


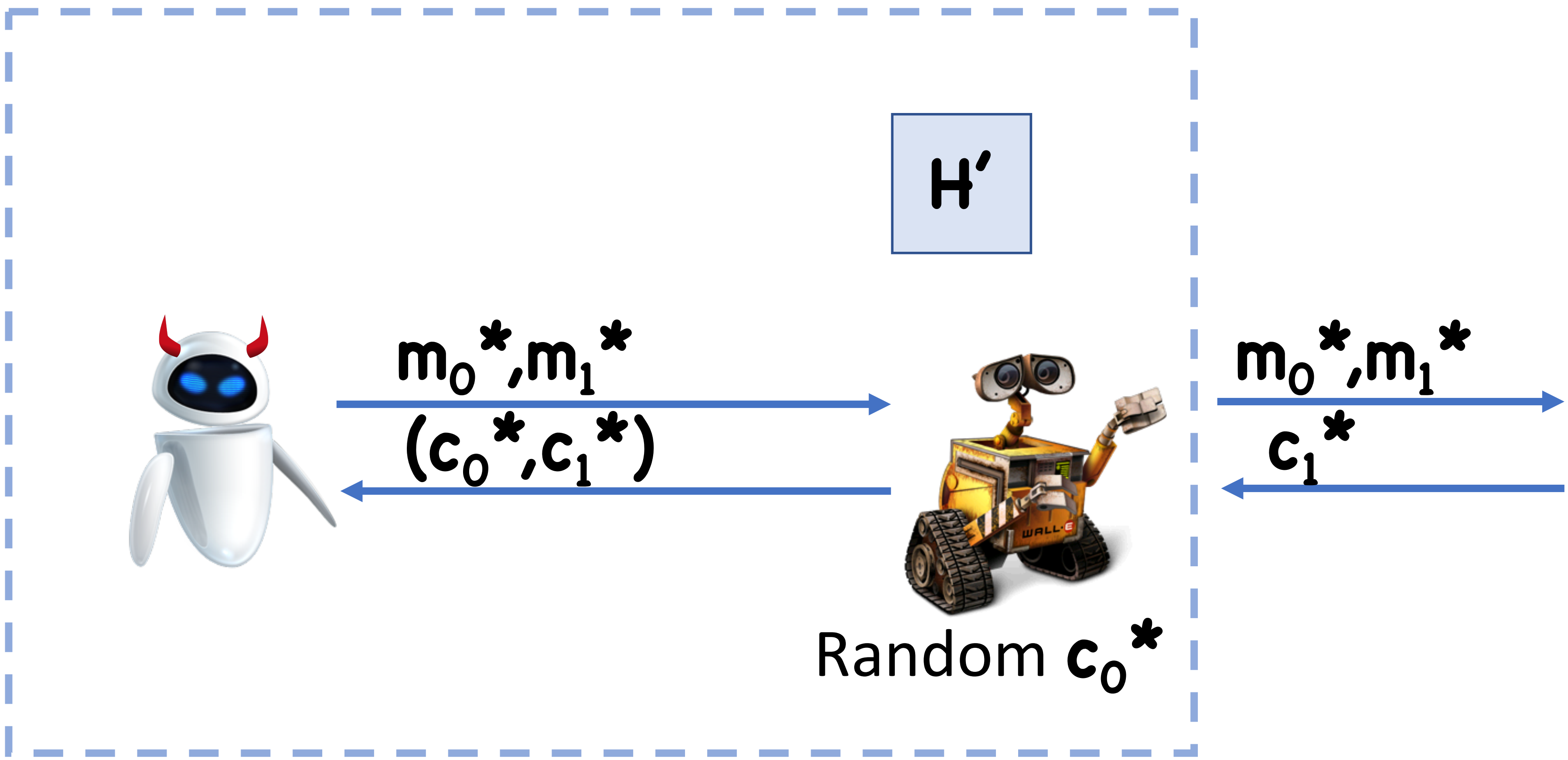
Proof

Case 2: construct **Enc**$_{\mathbf{SKE}}$ adversary

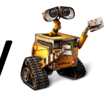


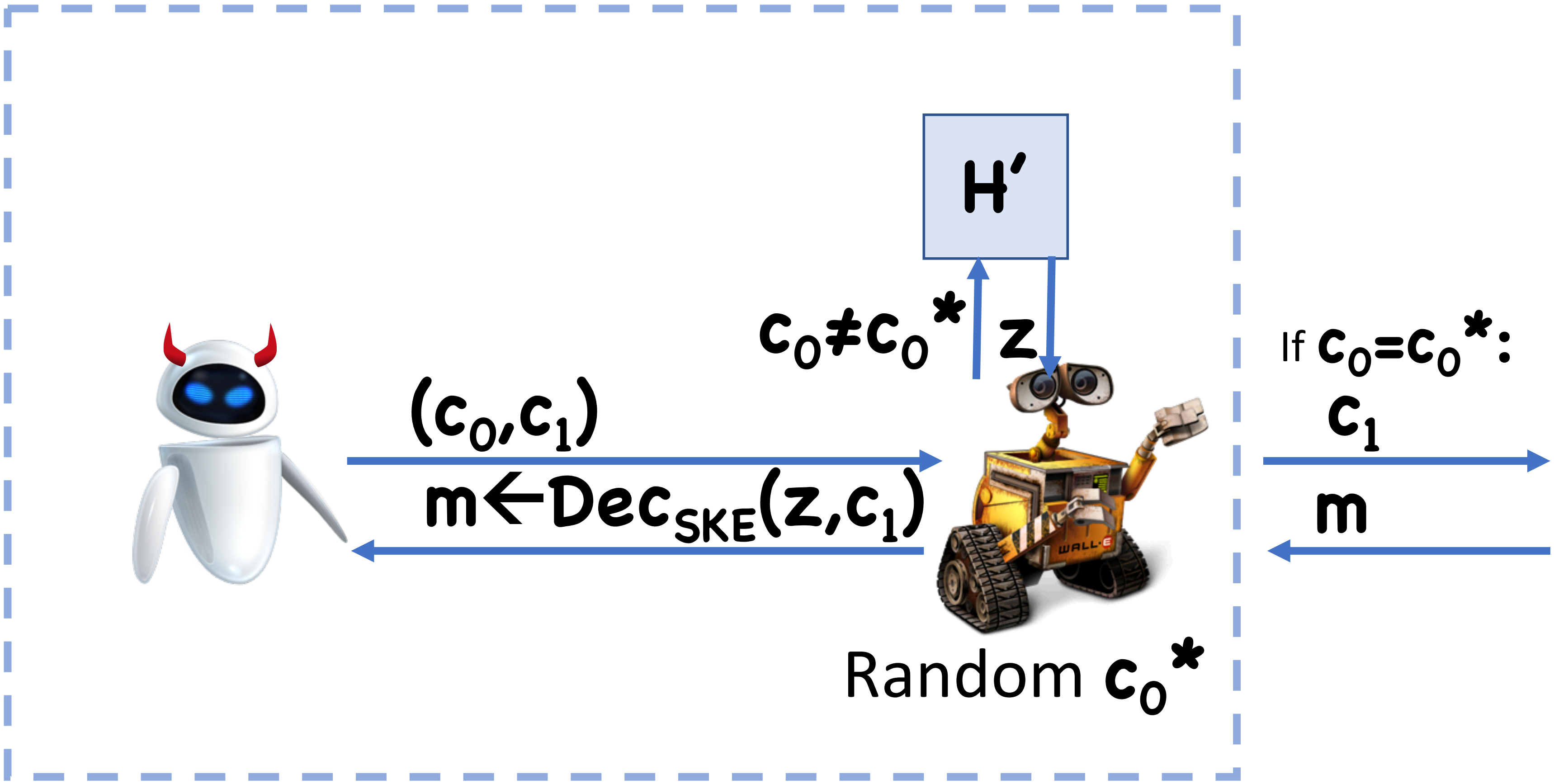
Proof

Case 2: construct $\mathbf{Enc}_{\mathbf{SKE}}$ adversary 

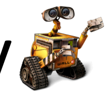


Proof

Case 2: construct $\mathbf{Enc_{SKE}}$ adversary 



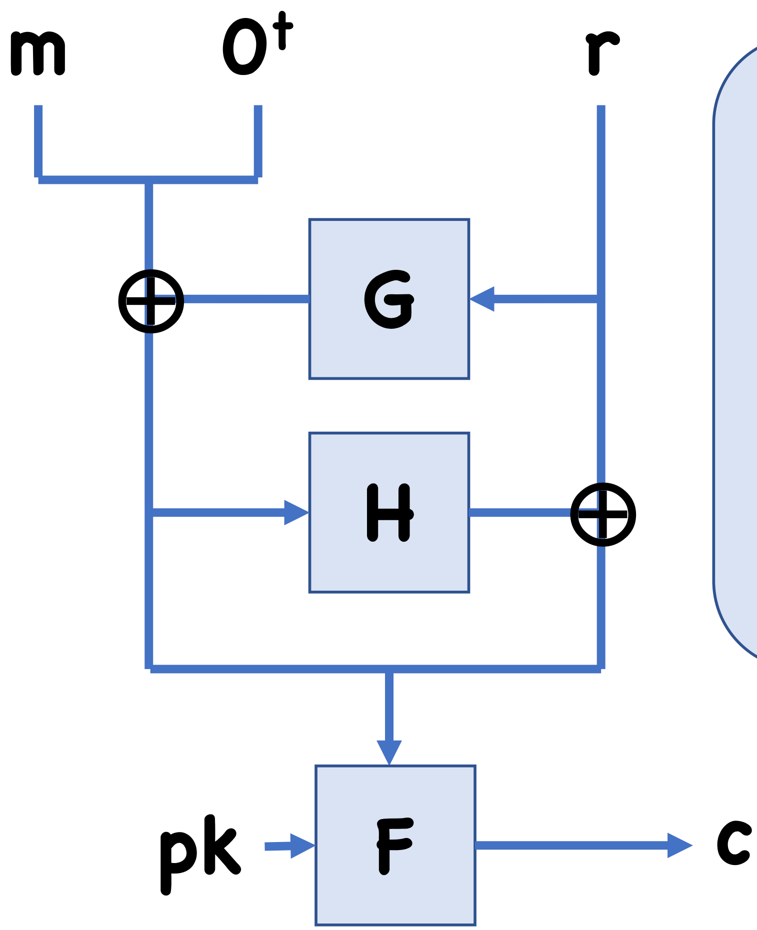
Proof

Case 2: construct $\mathsf{Enc}_{\mathsf{SKE}}$ adversary 

Analysis:

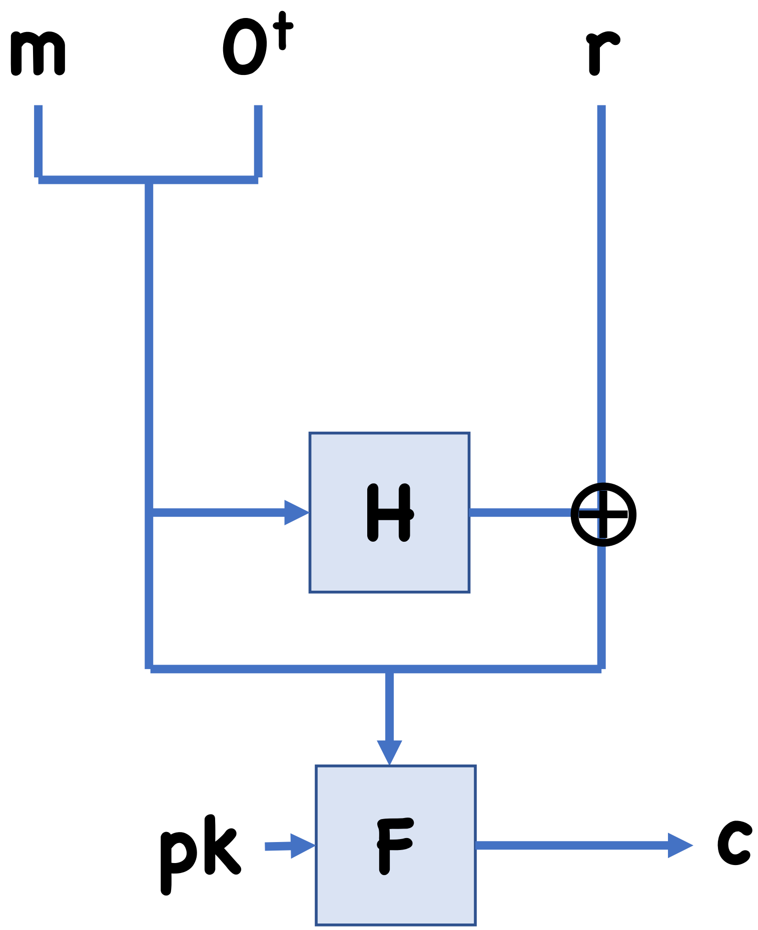
- Effectively set $\mathsf{H'}(\mathsf{c}_0^*) = \mathsf{k}$, where k is (unknown) challenger key
- Answers all queries correctly, provided adversary never queries RO on c_0^*
- Therefore, breaks security of $\mathsf{Enc}_{\mathsf{SKE}}$ in case 2

OAEP



Theorem: For RSA TDP, if **G,H** are modeled as a random oracles, then **(Gen$_{PKE}$,Enc$_{PKE}$,Dec$_{PKE}$)** is a CCA secure public key encryption scheme

Insecure OAEP Variants

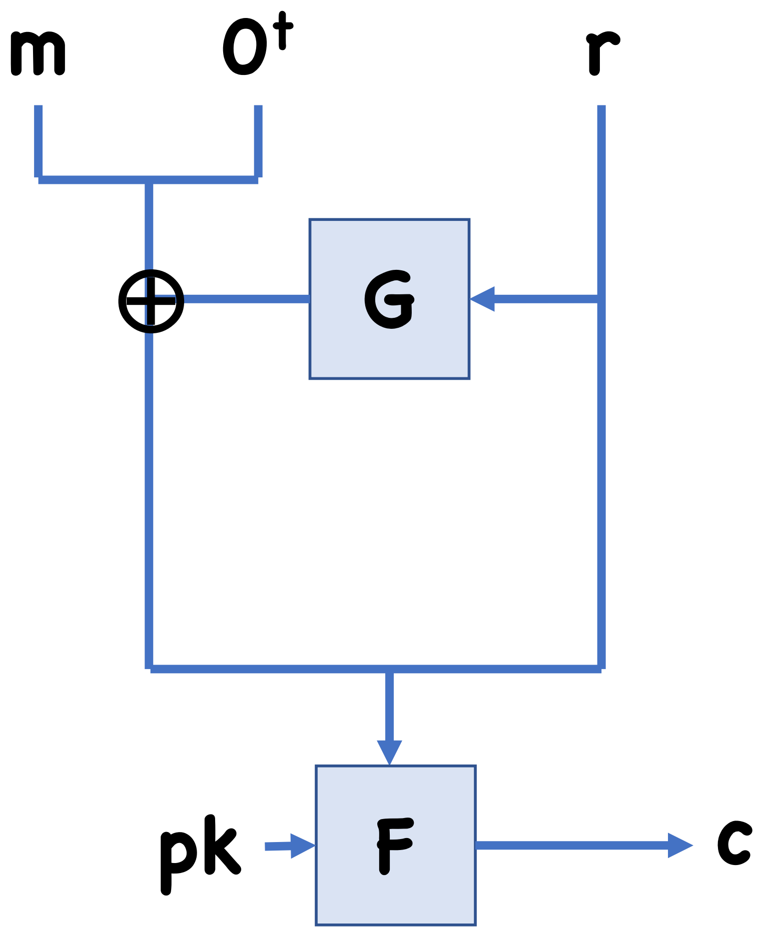


$c = F(pk, (m, 0^t, y))$

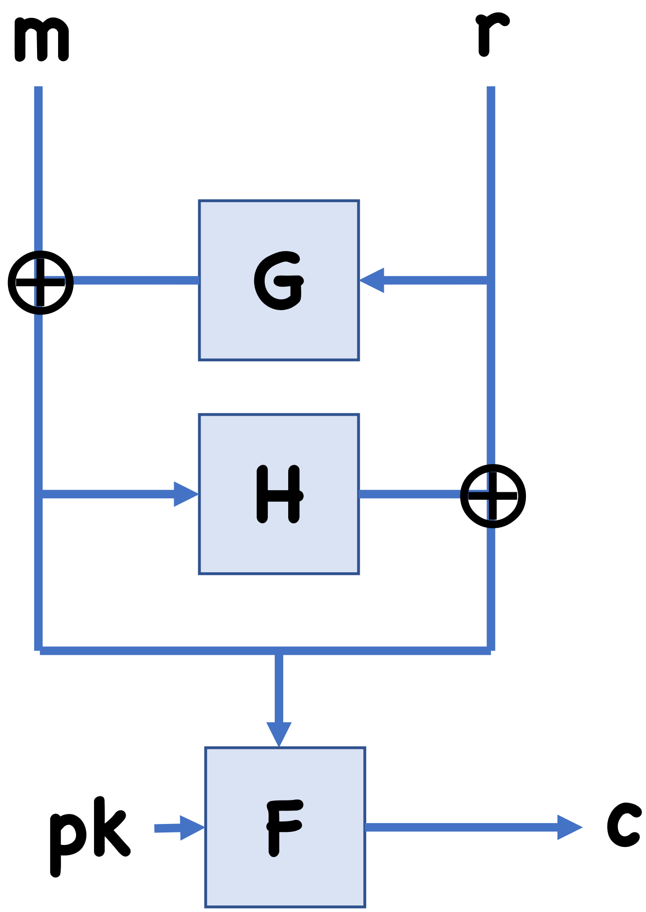
May contain **m** in the clear

- $F(pk, (m,x,y))$
  $= (m, F(pk, (x,y)))$

Insecure OAEP Variants



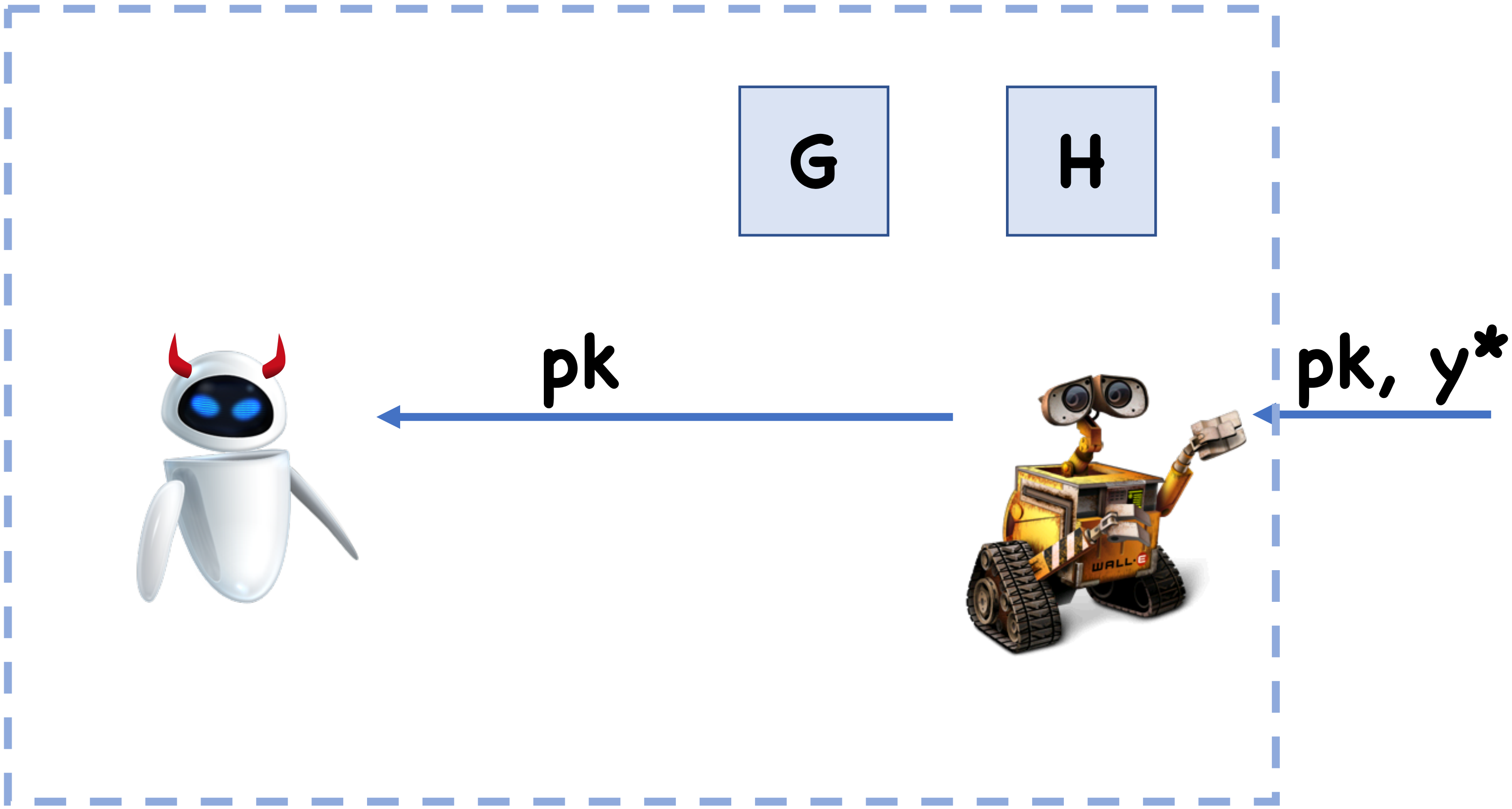
Why padding?



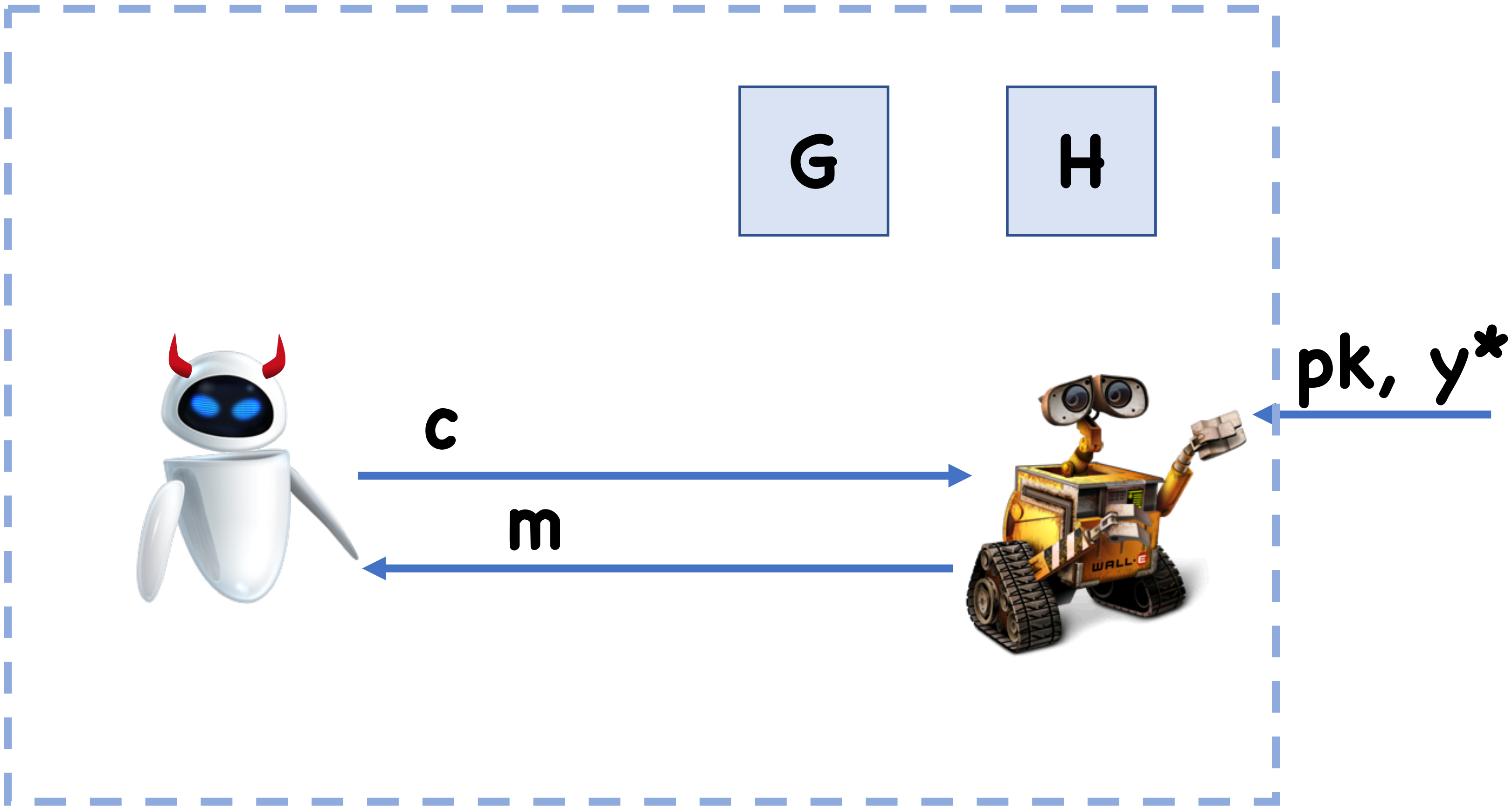
All ciphertexts decrypt to valid messages

- Makes it hard to argue security

High Level Proof Sketch



Claim: For any valid ctxt **c** queried by adv, adv must have created **c** by running **Enc(pk,m;r).** In this case, **m** can be decoded by looking at queries to **G,H**



Advantages of RSA-OAEP

RSA domain is at least 2048 bits

In hybrid encryption, ciphertext overhead =2048 bits

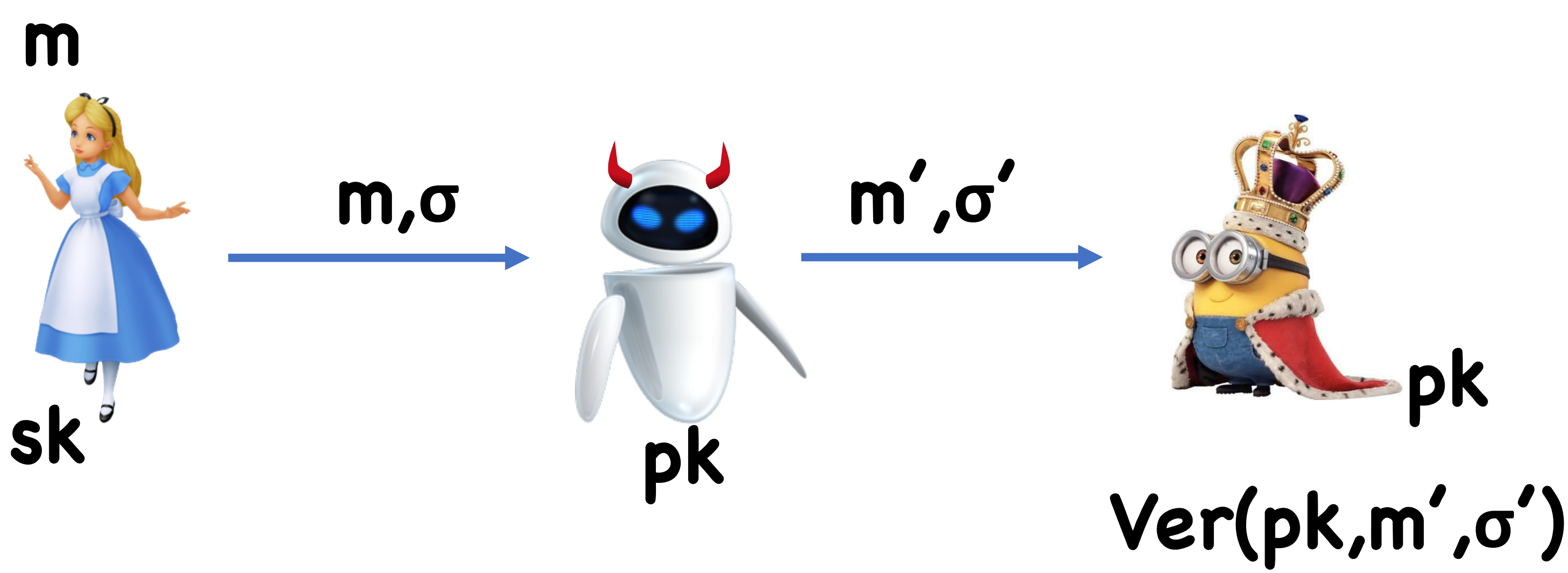
With OAEP (optimal asymmetric encryption padding), plaintext size can be, say 2048-256 bits with ciphertext size = 2048 bits

- Overhead = 256 bits

Digital Signatures

(aka public key MACs)

Message Integrity



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

Syntax and Correctness

Algorithms:

- **Gen() → (sk,pk)**
- **Sign(sk,m) → σ**
- **Ver(pk,m,σ) → 0/1**

Correctness:

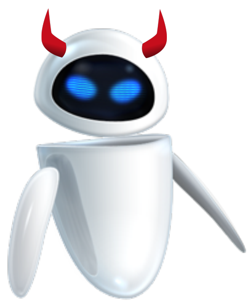
Pr[Ver(pk,m,Sign(sk,m))=1: (sk,pk)←Gen()] = 1

Security Notions?

Much the same as MACs, except adversary gets verification key

1-time Security For MACs

$(sk,pk) \leftarrow Gen()$

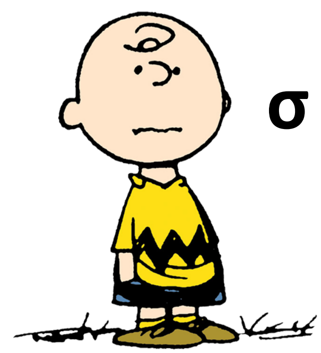


pk

m

σ

(m*,σ*)



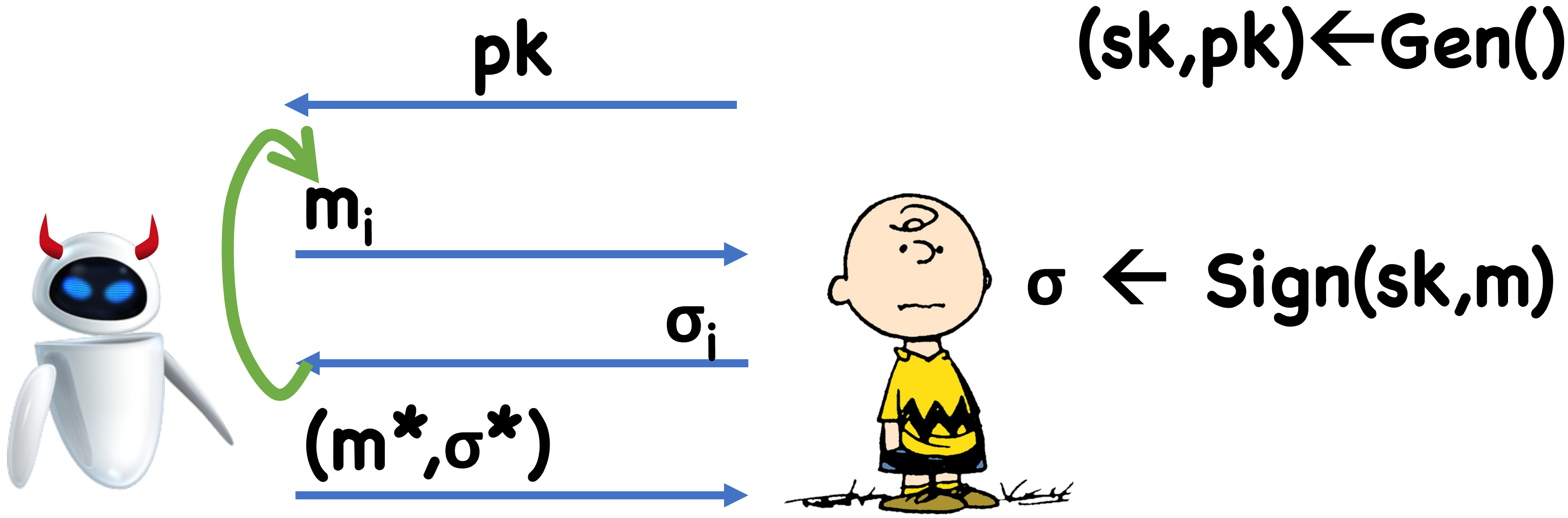
$\sigma \leftarrow Sign(sk,m)$

Output 1 iff:

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

1CMA-Adv() = Pr[ outputs 1]

Unbounded Use MACs



Output 1 iff:

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

CMA-Adv() = Pr[ outputs 1]

Signatures from TDPs?

$\mathrm{Gen}_{\mathrm{Sig}}() = \mathrm{Gen}()$

$\mathrm{Sign}(sk,m) = F^{-1}(sk,m)$

$\mathrm{Ver}(pk,m,\sigma)$: $F(pk, \sigma) == m$

Signatures from TDPs

$\mathbf{Gen_{Sig}() = Gen()}$

$\mathbf{Sign(sk,m) = F^{-1}(sk,\ H(m)\ )}$

$\mathbf{Ver(pk,m,\sigma):\ F(pk,\ \sigma) == H(m)}$

Theorem: If $\mathbf{(Gen,F,F^{-1})}$ is a secure TDP, and $\mathbf{H}$ is modeled as a random oracle, then $\mathbf{(Gen_{Sig},Sign,Ver)}$ is CMA-secure

Signatures from Injective TDFs?

What goes wrong?

Next Time

More digital signatures