COS433/Math 473: Cryptography

Mark Zhandry

Princeton University

Fall 2020

Announcements/Reminders

HW5 due Nov 10

PR2 due Dec 5

Previously on COS 433…

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

pk,y

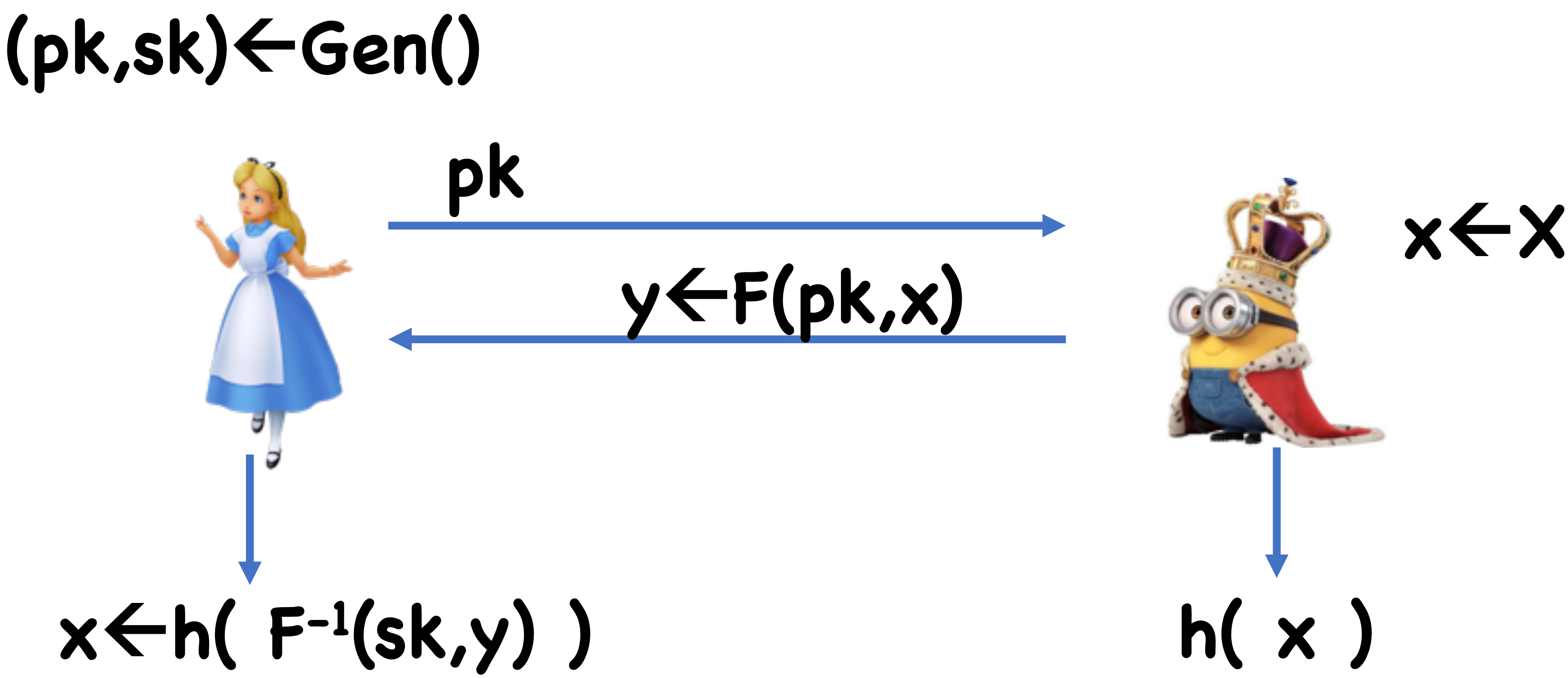
(sk,pk)←Gen()
x←X
y←F(pk,x)

x'

Adversary wins if **x=x'**

In other words, **F(pk, ·)** is a one-way function

Key Distribution from TDPs



h a hardcore bit for **F(pk, ·)**

Trapdoor Permutations from RSA

Gen():

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

F(pk,x): Output $y = x^e \mod N$

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

Key Distribution from DH

Everyone agrees on group **G** of prime order **p**

$a \leftarrow \mathbb{Z}_p$

$b \leftarrow \mathbb{Z}_p$

Key Distribution from DH

Everyone agrees on group **G** or prime order **p**

$a \leftarrow \mathbb{Z}_p$

$g^a \longrightarrow$

$\longleftarrow g^b$

$b \leftarrow \mathbb{Z}_p$

Key Distribution from DH

Everyone agrees on group **G** or prime order **p**

$a \leftarrow \mathbb{Z}_p$

g^a →

← g^b

$b \leftarrow \mathbb{Z}_p$

$k = (g^b)^a = g^{ab}$

$k = (g^a)^b = g^{ab}$

Key Distribution from DH

Theorem: If DDH holds on **G**, then the Diffie-Hellman protocol is secure

Proof:

- **(Trans,k) = ((g^a, g^b), g^{ab})**
- DDH means indistinguishable from **((g^a, g^b), g^c)**

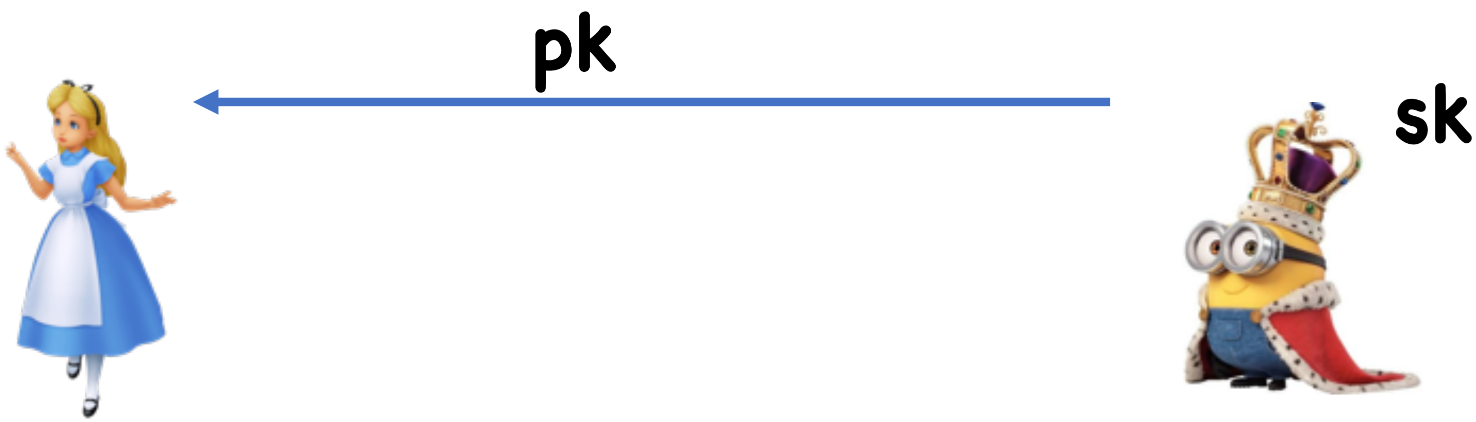
What if only CDH holds, but DDH is easy?

Today

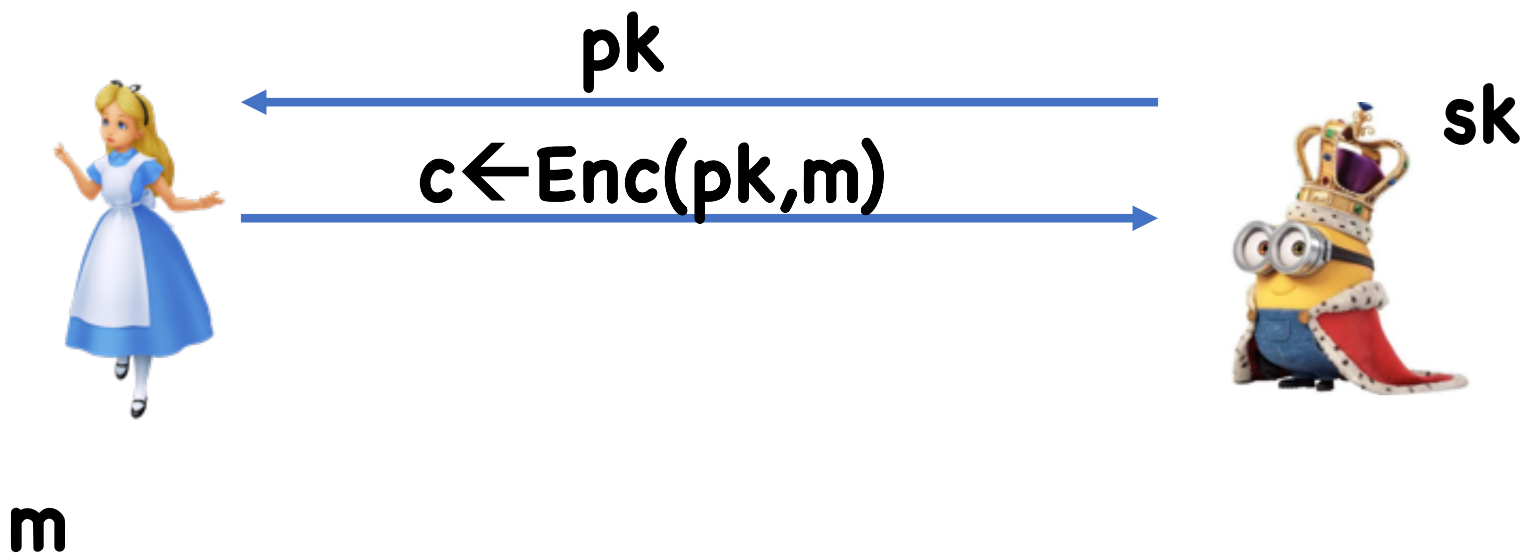
Public Key Encryption
Digital signatures (if time)

Public Key Encryption

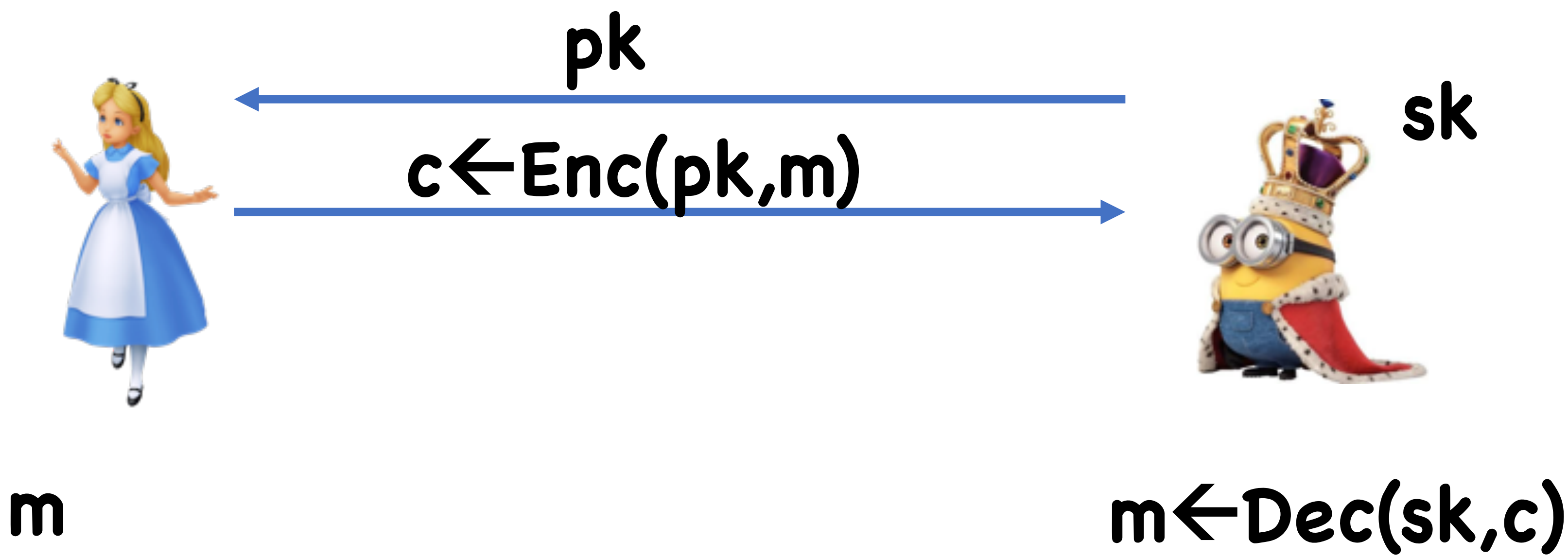
Public Key Encryption



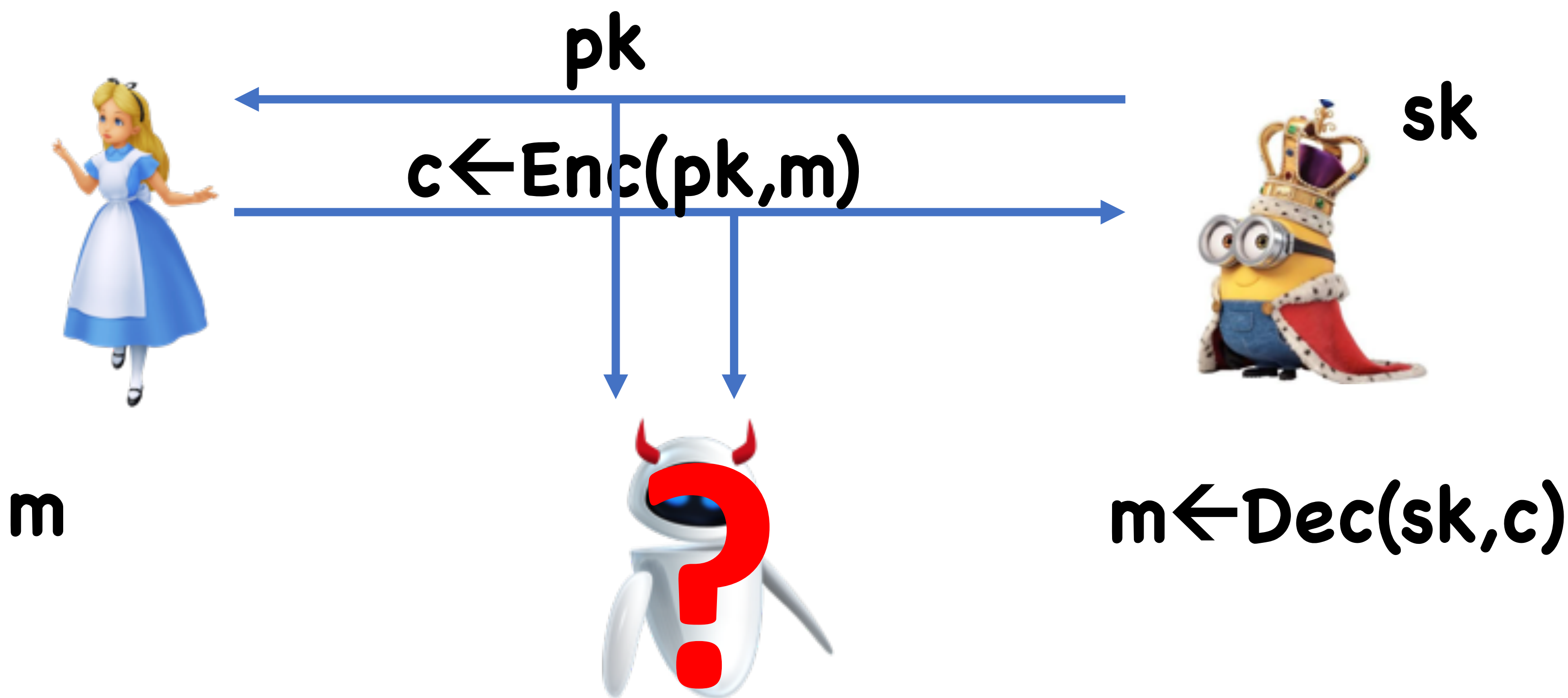
Public Key Encryption



Public Key Encryption



Public Key Encryption



PKE vs Key Agreement

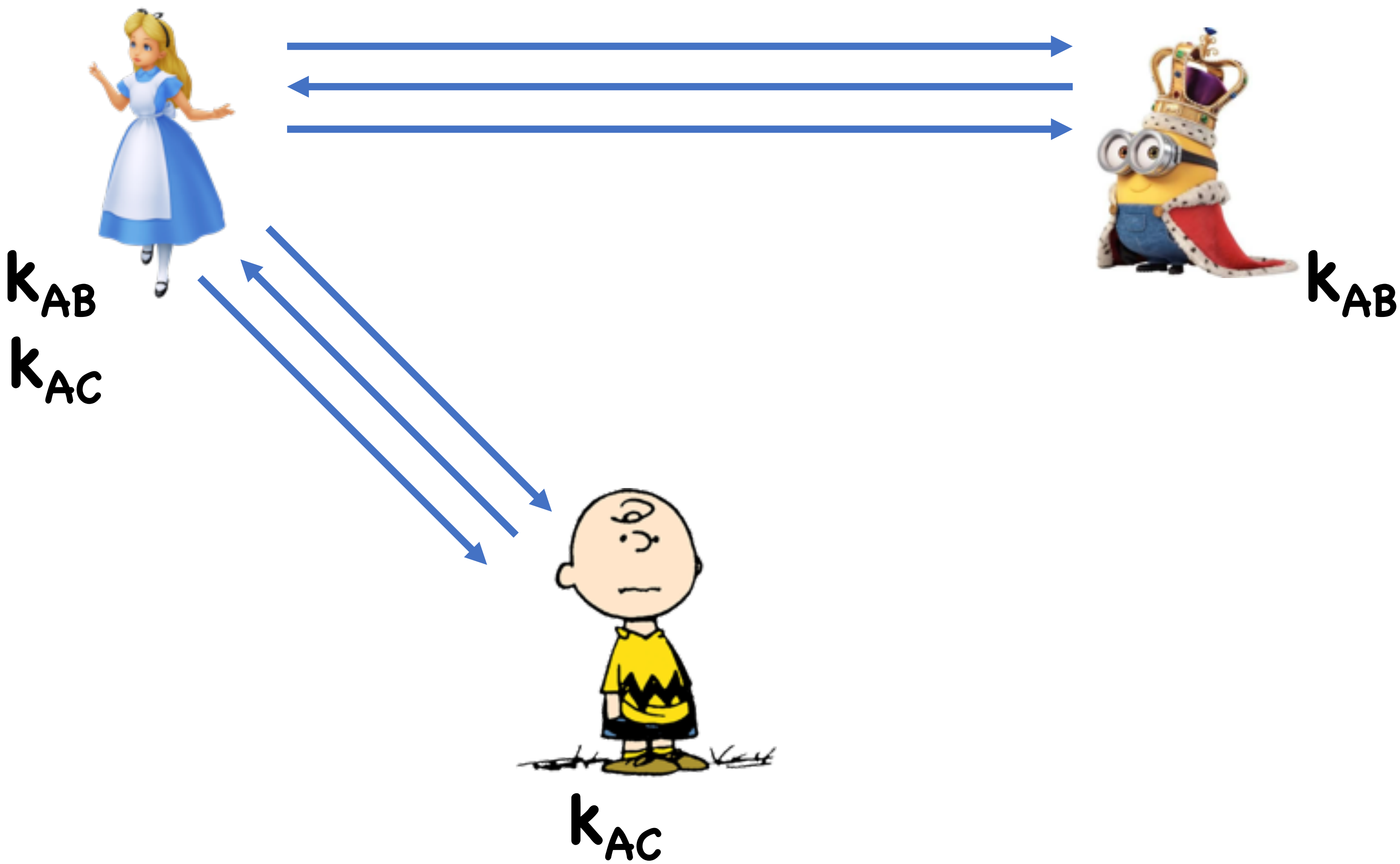
Key agreement:

PKE vs Key Agreement

Key agreement:

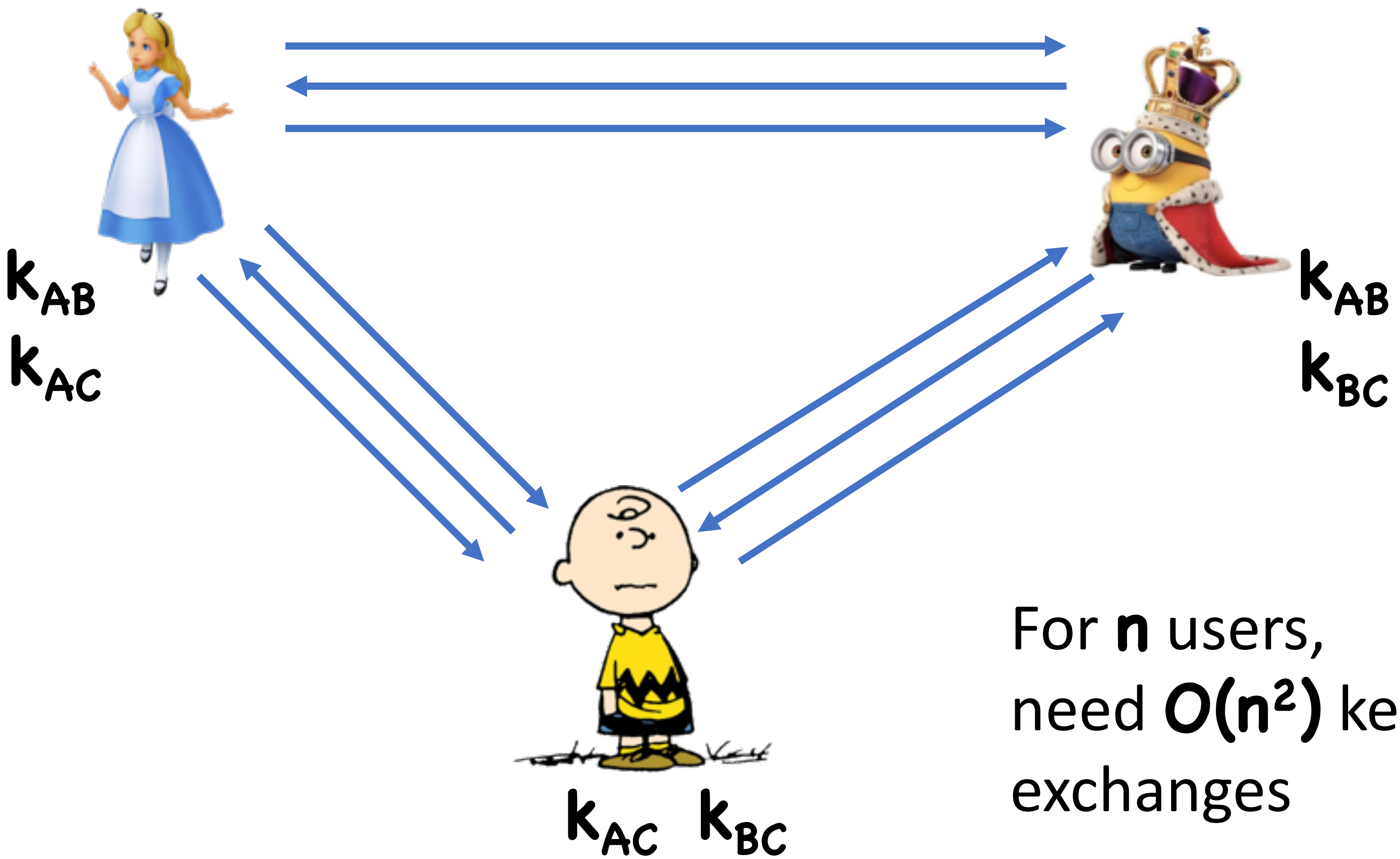
PKE vs Key Agreement

Key agreement:



PKE vs Key Agreement

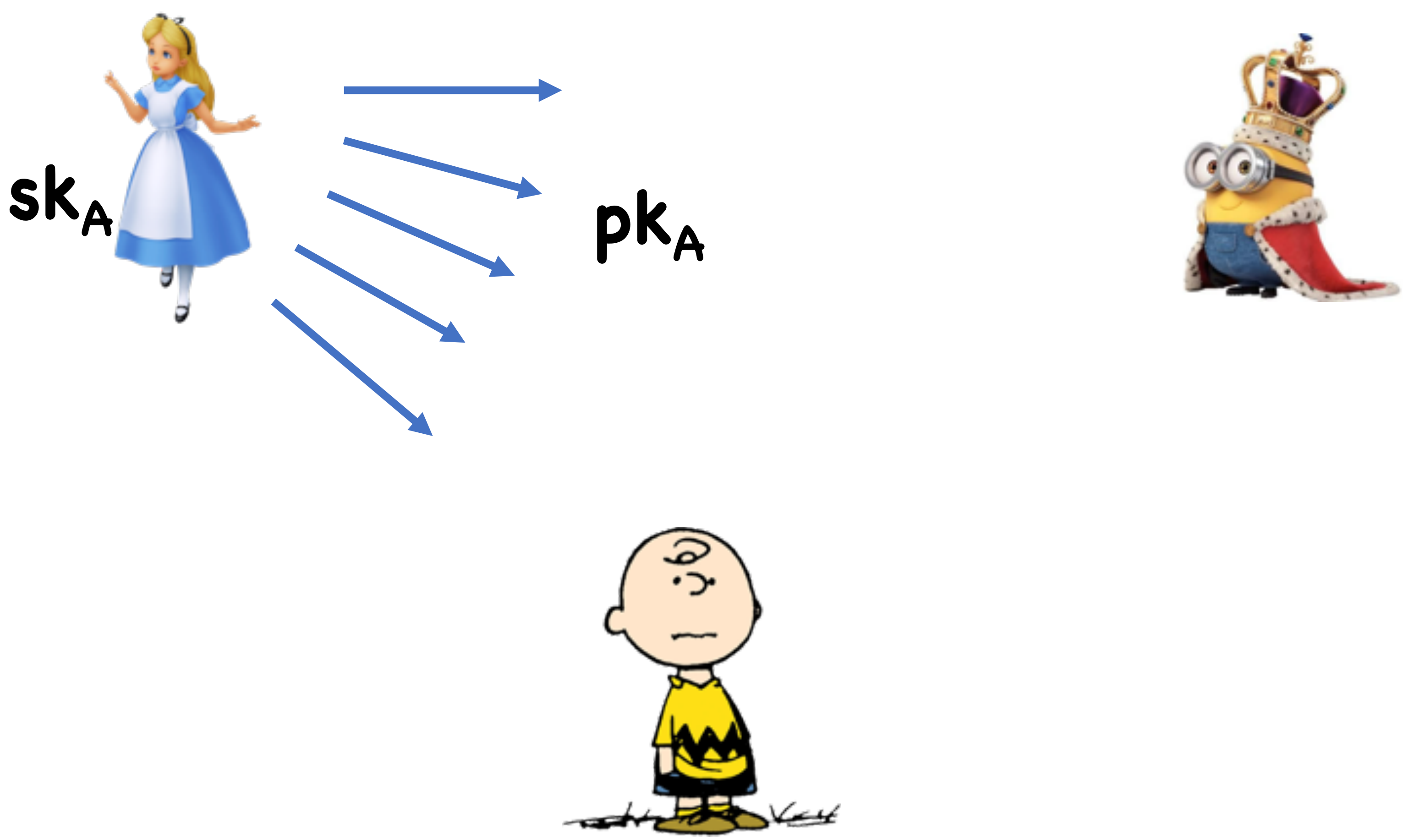
Key agreement:



For **n** users, need **O(n²)** key exchanges

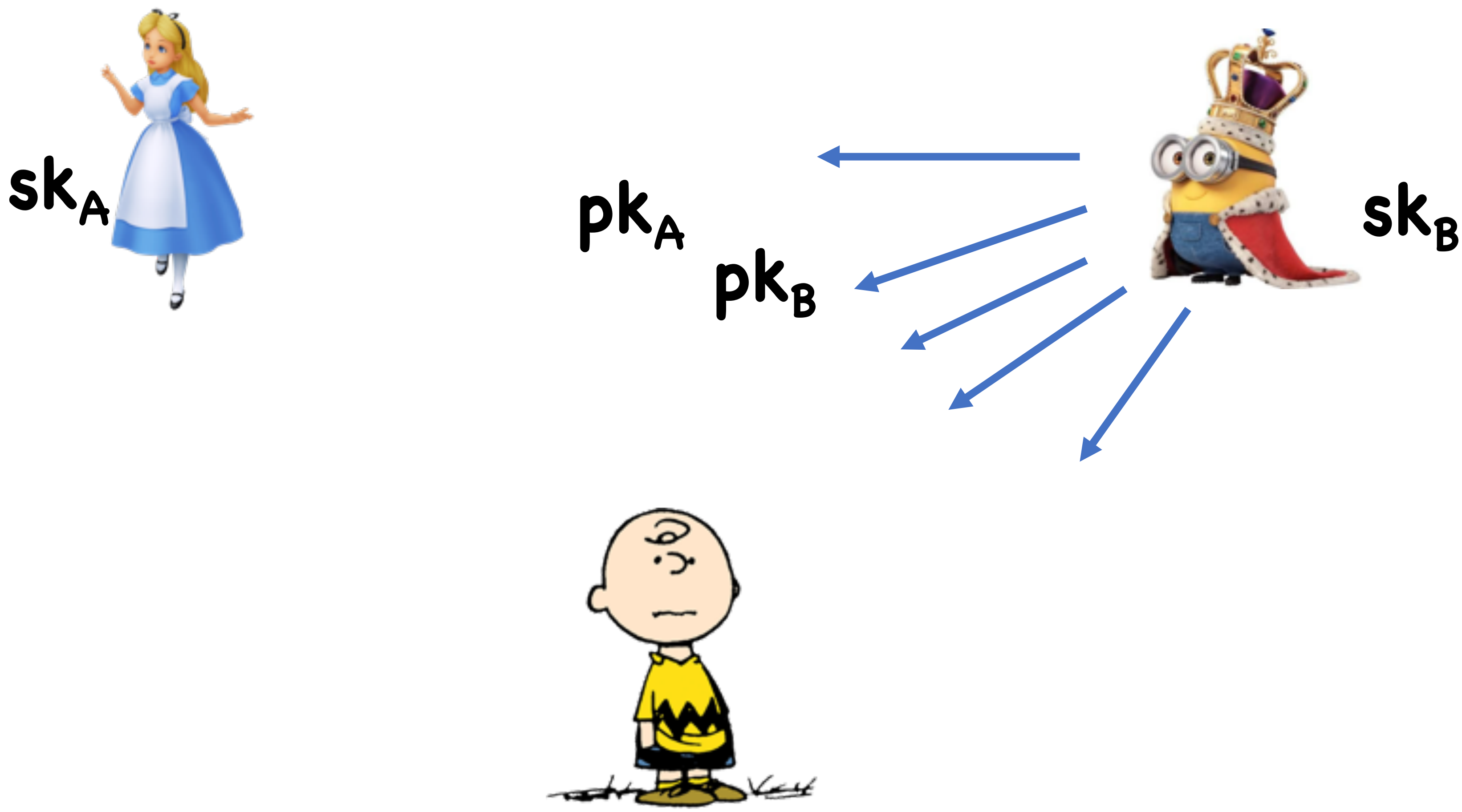
PKE vs Key Agreement

PKE:



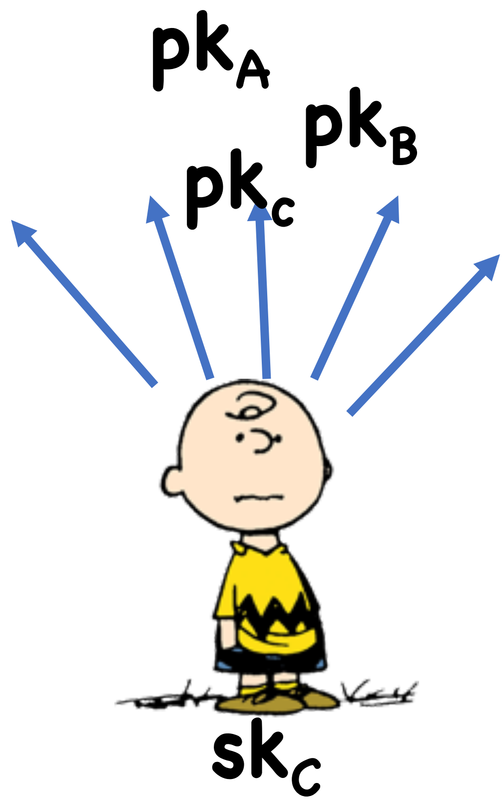
PKE vs Key Agreement

PKE:



PKE vs Key Agreement

PKE:



For **n** users, need **O(n)** public keys

PKE Syntax

Message space **M**

Algorithms:

- **(sk,pk)←Gen(λ)**
- **Enc(pk,m)**
- **Dec(sk,m)**

Correctness:

Pr[Dec(sk,Enc(pk,m)) = m: (sk,pk)←Gen(λ)] = 1

Security

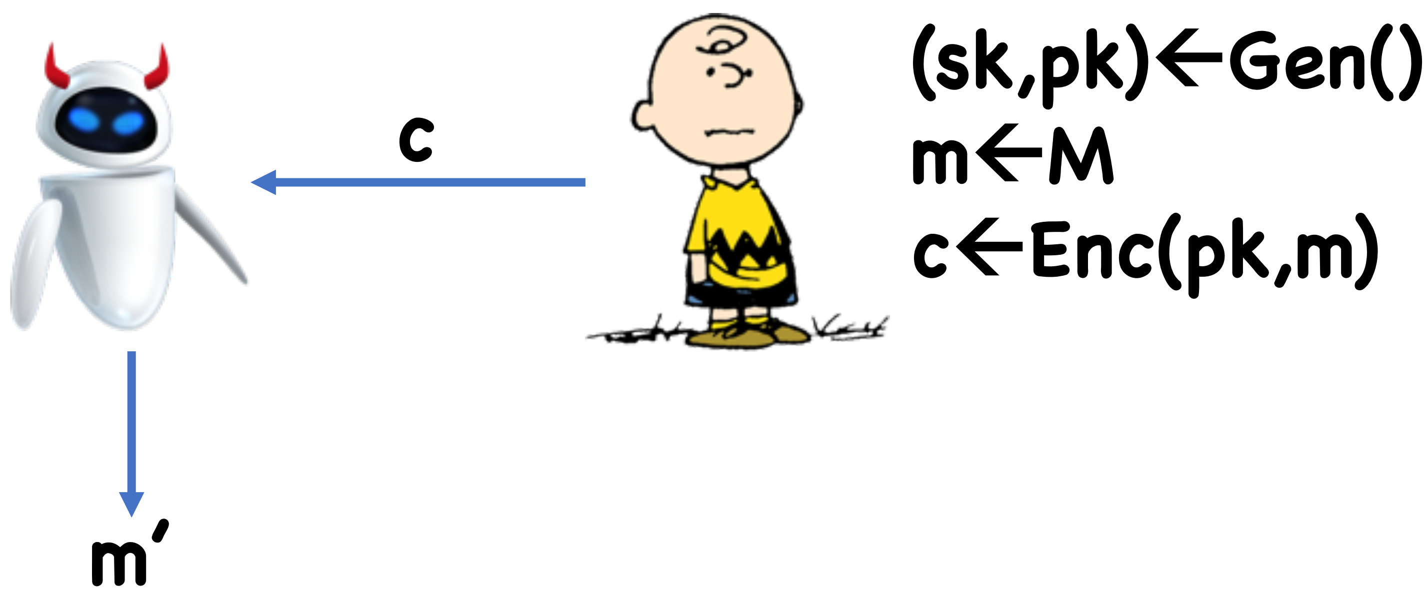
One-way security

Semantic Security

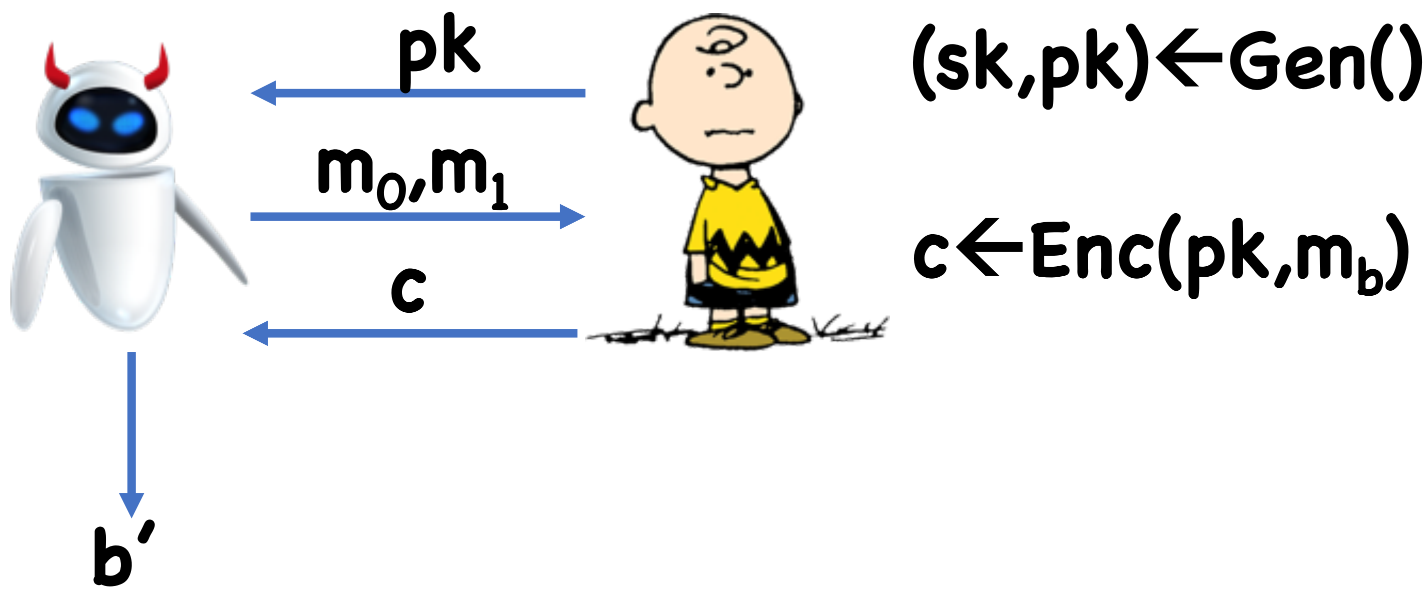
CPA security

CCA Security

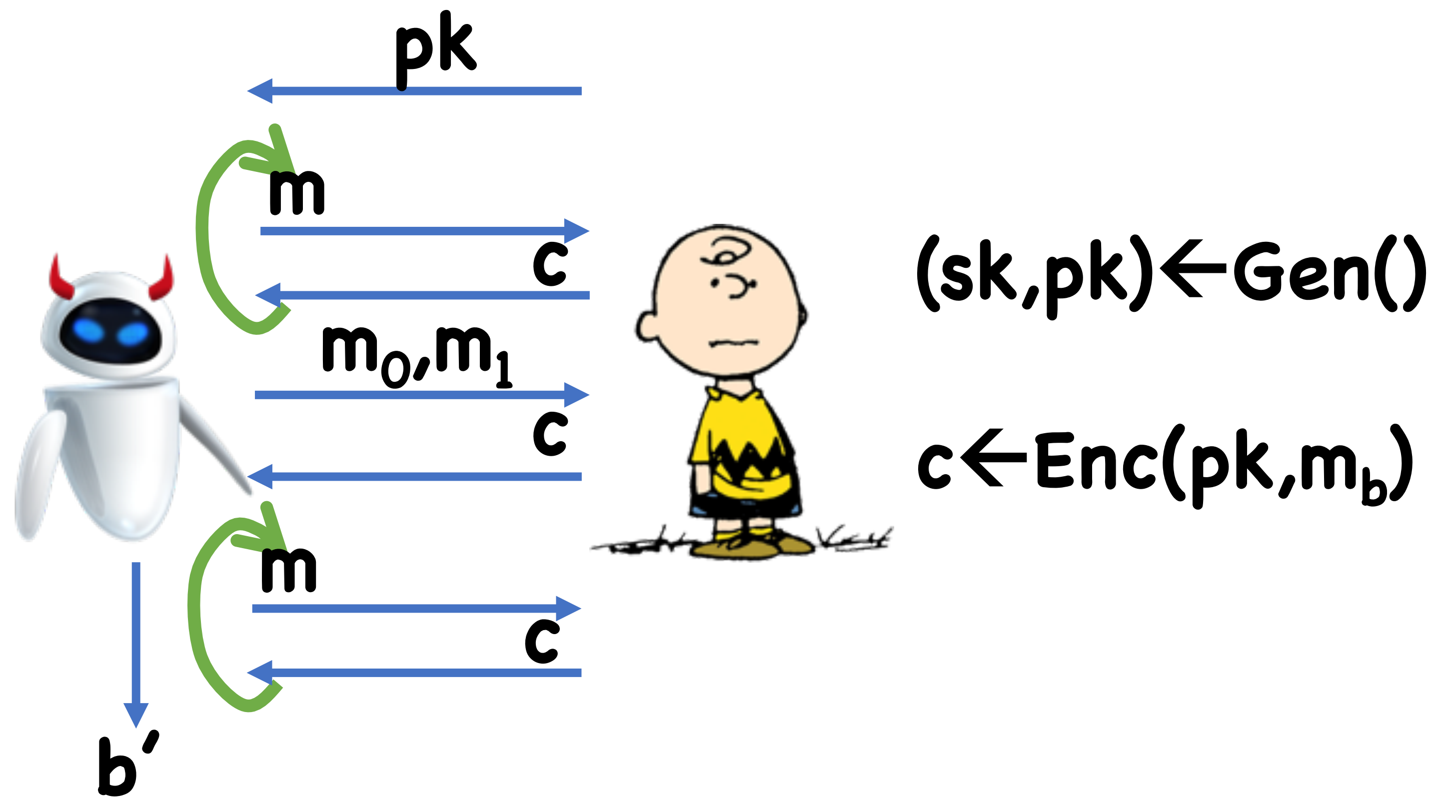
One-way Security



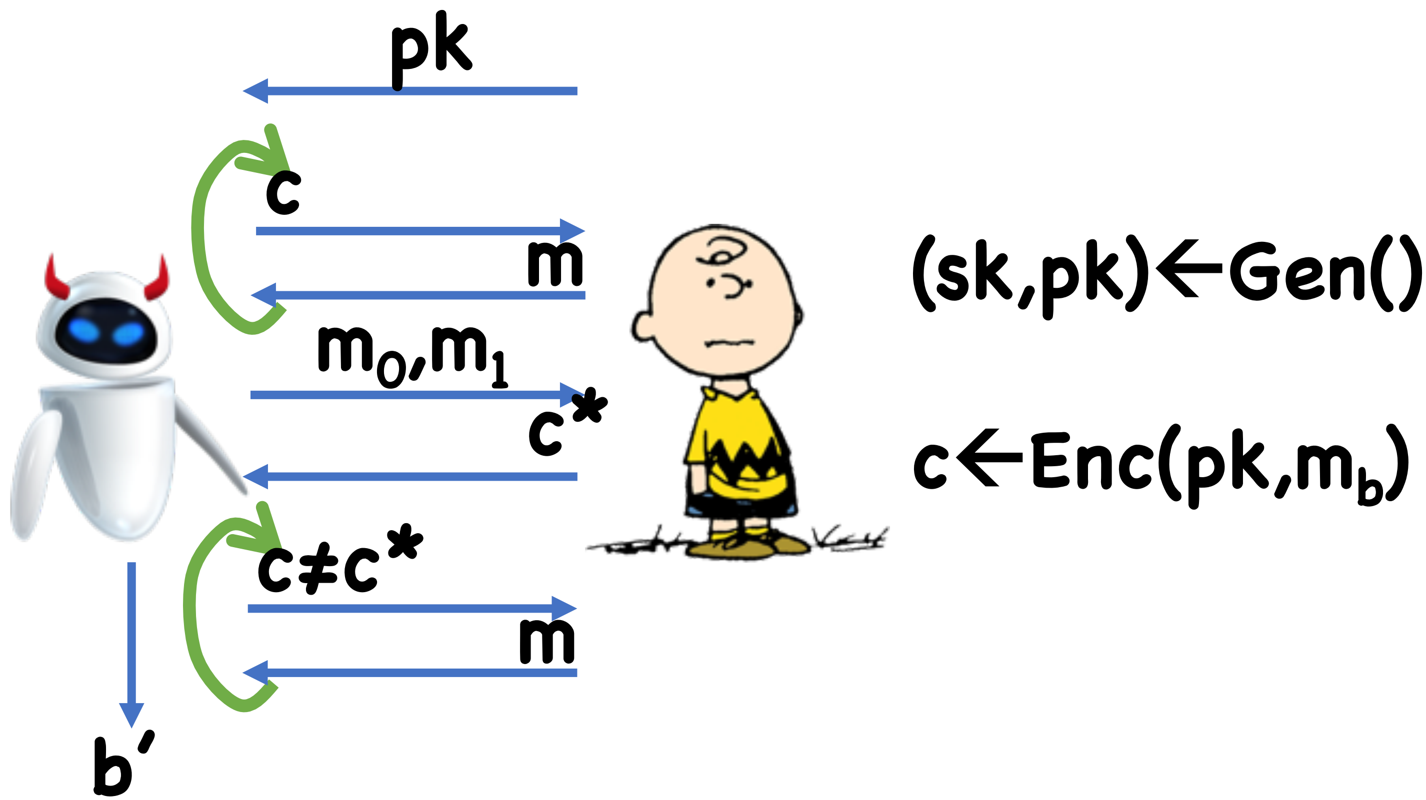
Semantic Security



CPA Security



CCA Security



Question: Which two notions are equivalent?

One-Way Encryption from TDPs

$\mathbf{Gen_E() = Gen_{TDP}()}$

Enc(pk,m): Output $\mathbf{c = F(pk,m)}$

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

Semantically Secure Encryption from TDPs

Ideas?

Considerations

A single server often has to decrypt many ciphertexts, whereas each user only encrypts a few messages

Therefore, would like to make decryption fast

Considerations

Encryption running time:

- $O(\log e)$ multiplications, each taking $O(\log^2 N)$
- Overall $O(\log e \log^2 N)$

Decryption running time:

- $O(\log d \log^2 N)$

(Note that $ed \geq \Phi(N) \approx N$)

Considerations

Possibilities:

- **e** tiny (e.g. **3**): fast encryption, slow decryption
- **d** tiny (e.g. **3**): fast decryption, slow encryption
  - Problem?
- **d** relatively small (e.g. $d \approx N^{0.1}$)
  - Turns out, there is an attack that works whenever $d < N^{.292}$

Therefore, need **d** to be large, but ok taking $e=3$

Considerations

Chinese remaindering to speed up decryption:

- Let $\mathbf{sk=(d_0,d_1)}$ where

$$d_0 = d \bmod (p-1),\ d_1 = d \bmod (q-1)$$

- Let $c_0 = c \bmod p,\ c_1 = c \bmod q$
- Compute $m_0 = c^{d0} \bmod p,\ m_1 = c^{d1} \bmod q$
- Reconstruct $\mathbf{m}$ from $m_0,\ m_1$

Running time:

- $r \log^3 p + r \log^3 q + O(\log^2 N) \approx r(\log^3 N)/4$

ElGamal

Group $\mathbf{G}$ of order $\mathbf{p}$, generator $\mathbf{g}$
Message space = $\mathbf{G}$

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

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

Dec?

Theorem: If DDH is hard in **G**, then ElGamal is CPA secure

Proof:

- Adversary sees $\mathbf{h=g^a, g^r, g^{ar}\times m_0}$
- DDH: indistinguishable from $\mathbf{g^a, g^r, g^c\times m_0}$
- Same as $\mathbf{g^a, g^r, g^c\times m_1}$
- DDH again: indistinguishable from $\mathbf{g^a, g^r, g^{ar}\times m_0}$

CCA-Secure Encryption

Non-trivial to construct with provable security

Most efficient constructions have heuristic security

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

CCA Secure PKE from TDPs

Gen_{PKE}() = Gen()
Enc_{PKE}(pk, m):

- Choose random **r**
- Let **c** ← **F(pk,r)**
- Let **d** ← **Enc_{SKE}(H(r), m)**
- Output **(c,d)**

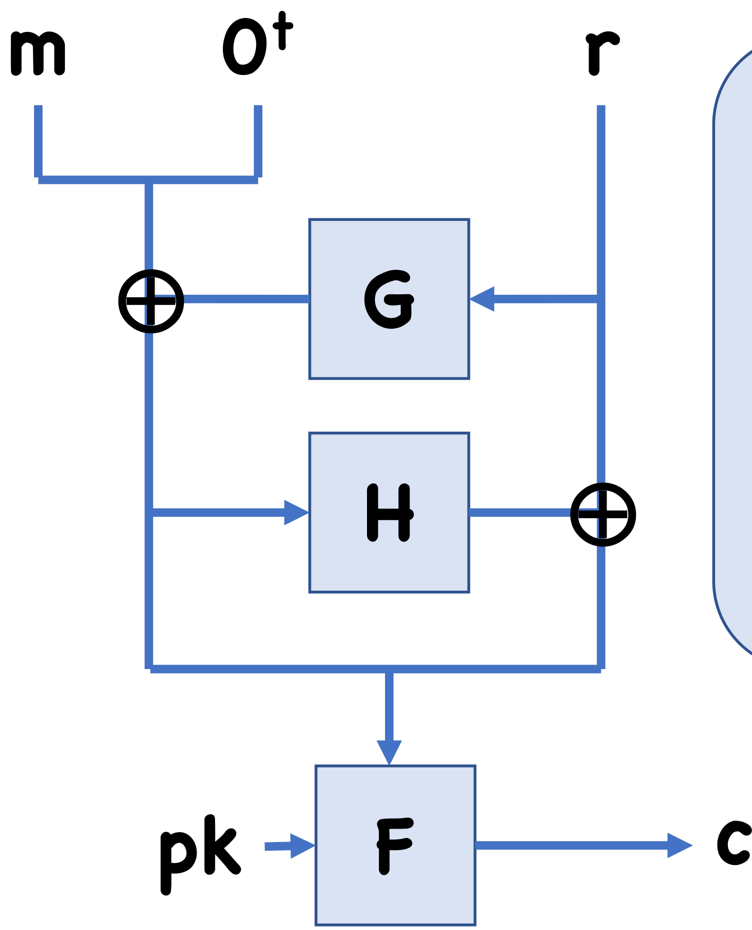
Dec_{PKE}(sk, (c, d)):

- **Let r ← F^{-1}(sk, c)**
- **Let m ← Dec_{SKE}(H(r), d)**

CCA Secure PKE from TDPs

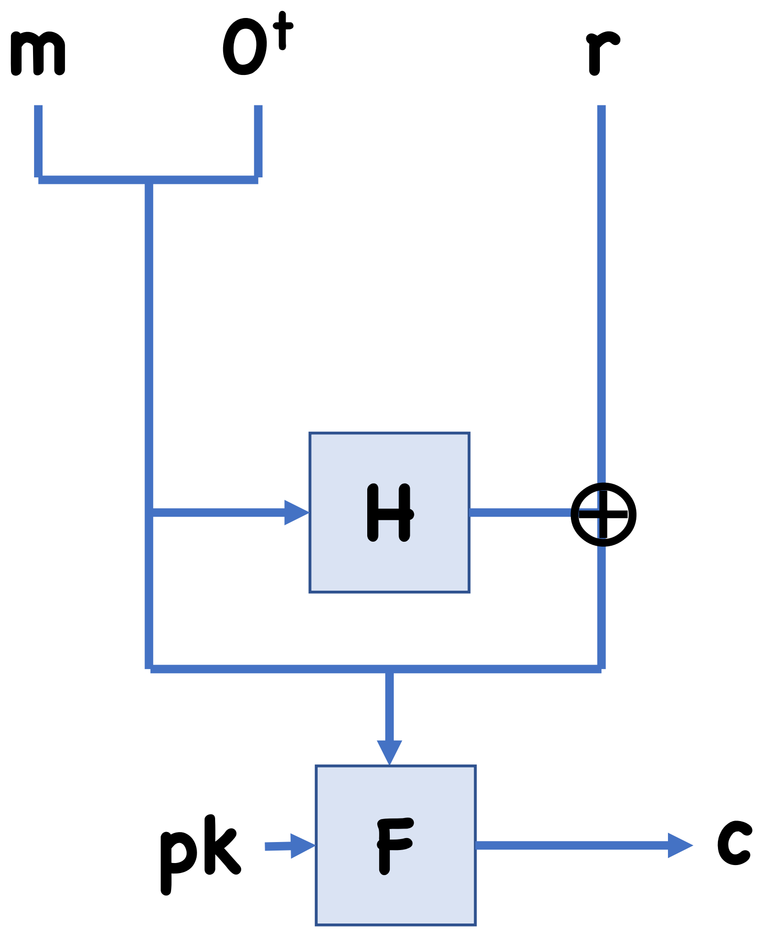
Theorem: If $(\mathbf{Enc}_{SKE},\mathbf{Dec}_{SKE})$ is a CCA-secure secret key encryption scheme, $(\mathbf{Gen},\mathbf{F},\mathbf{F}^{-1})$ is a TDP, and $\mathbf{H}$ is modeled as a random oracle, then $(\mathbf{Gen}_{PKE},\mathbf{Enc}_{PKE},\mathbf{Dec}_{PKE})$ is a CCA secure public key encryption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

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

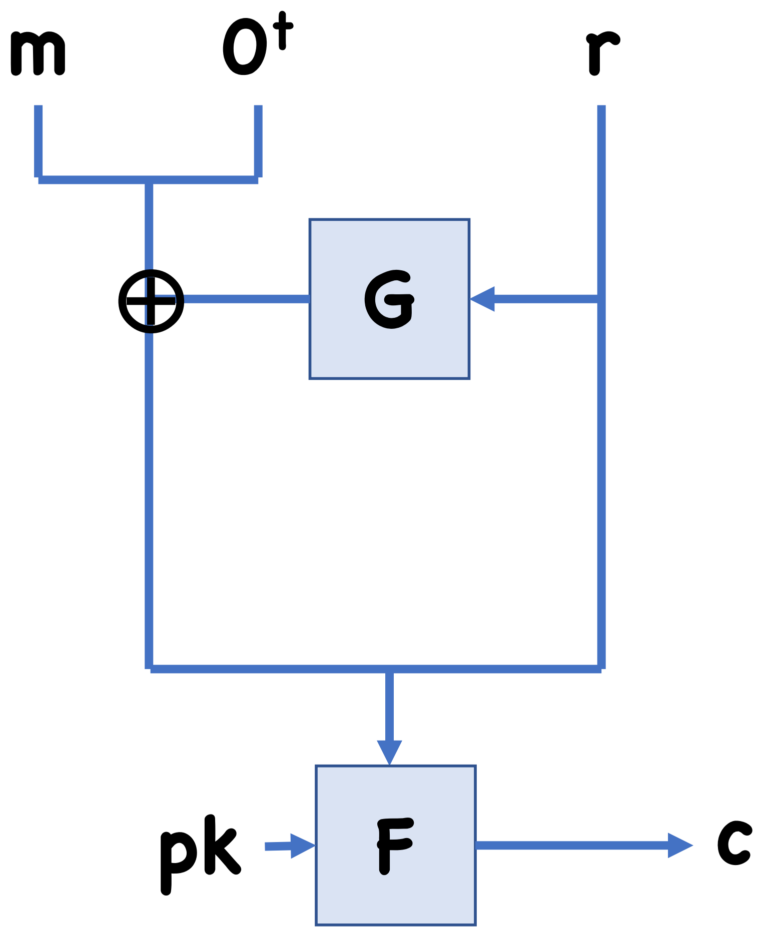


$c = F(pk, (m, 0^t, y))$

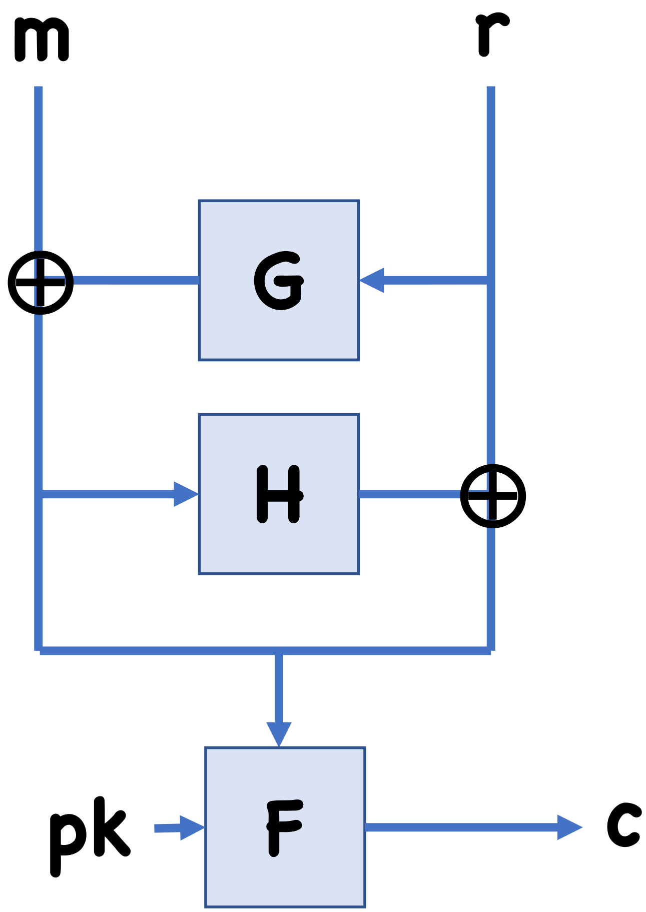
May contain **m** in the clear

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

Insecure OAEP Variants



Why padding?



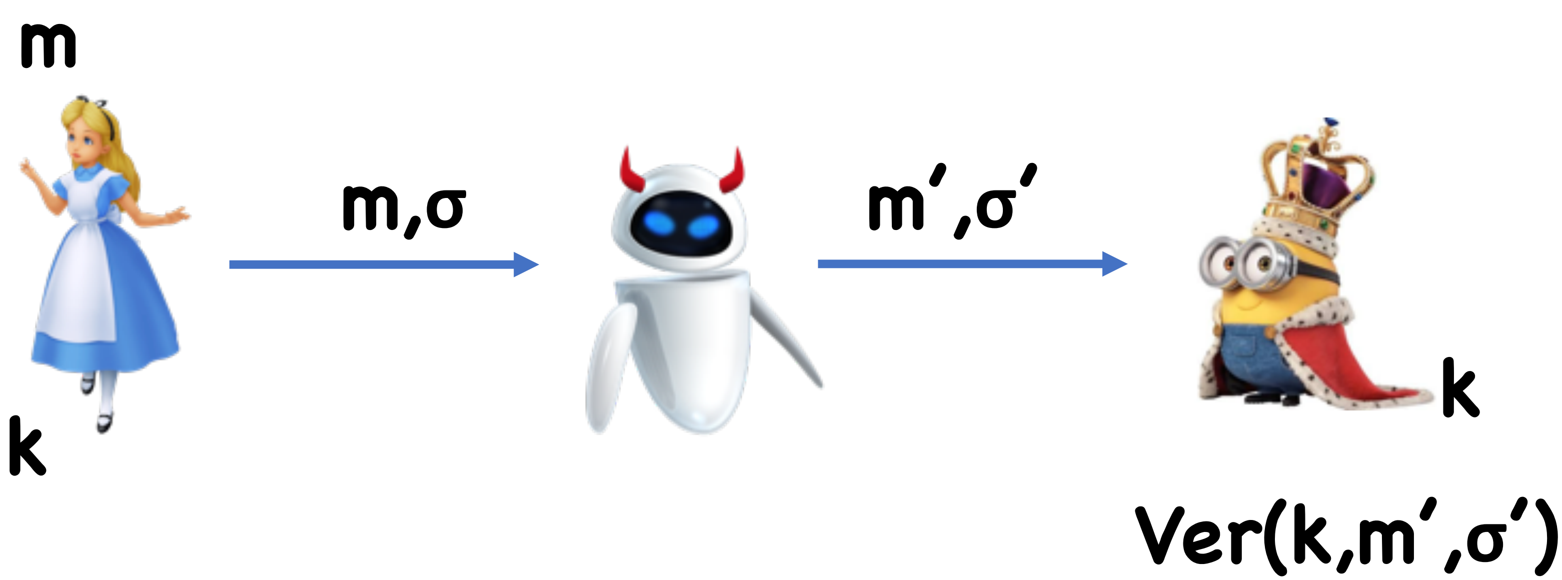
All ciphertexts decrypt to valid messages

- Makes it hard to argue security

Digital Signatures

(aka public key MACs)

Message Authentication Codes



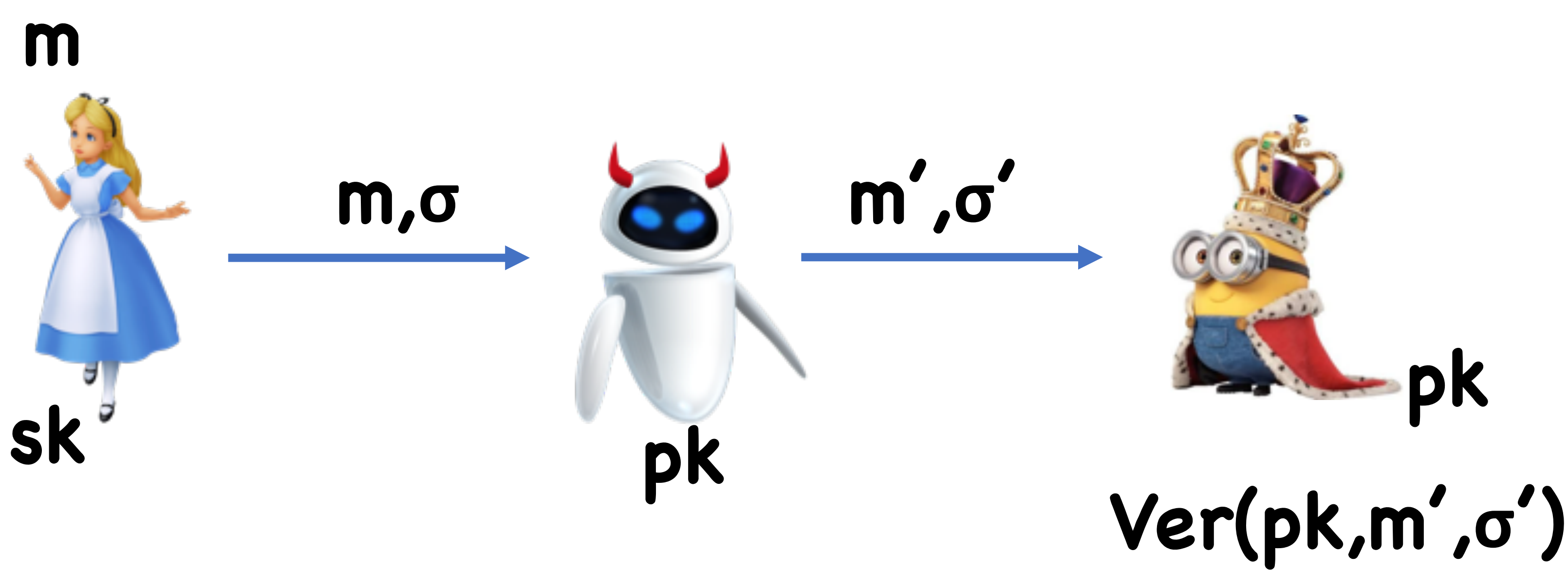
Goal: If Eve changed **m**, Bob should reject

Problem

What if Alice and Bob have never met before to exchange key **k**?

Want: a public key version of MACs where Bob can verify without having Alice's secret key

Message Integrity in Public Key Setting



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

Digital Signatures

Algorithms:

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

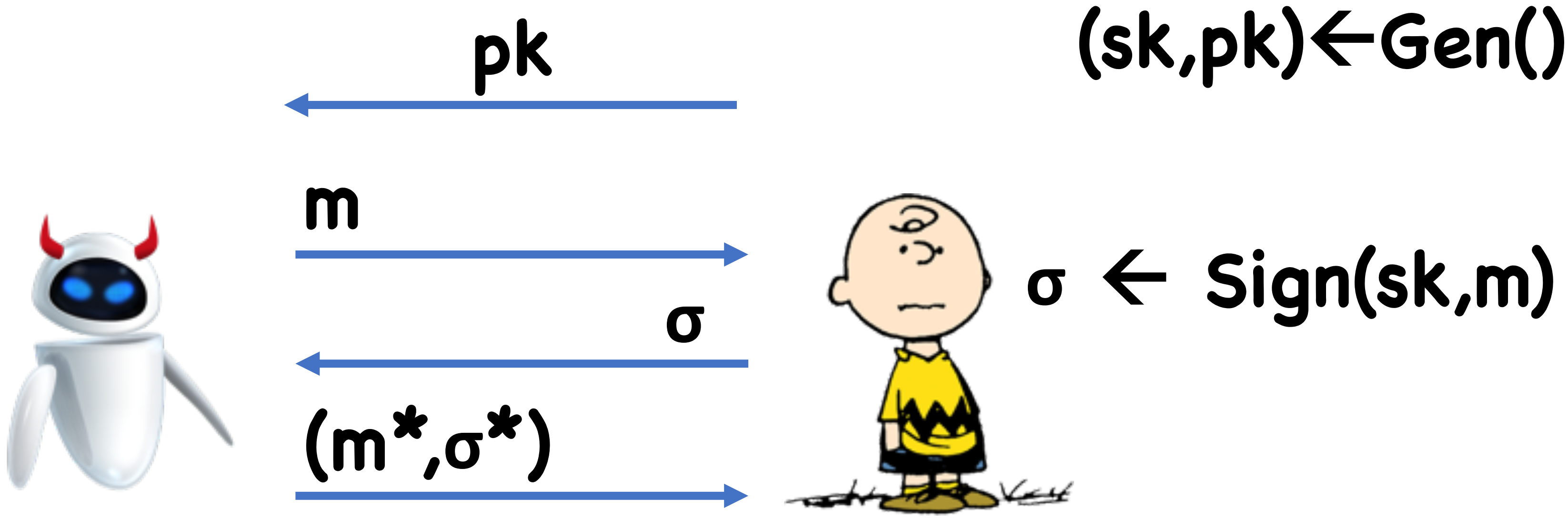
Correctness:

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 Signatures

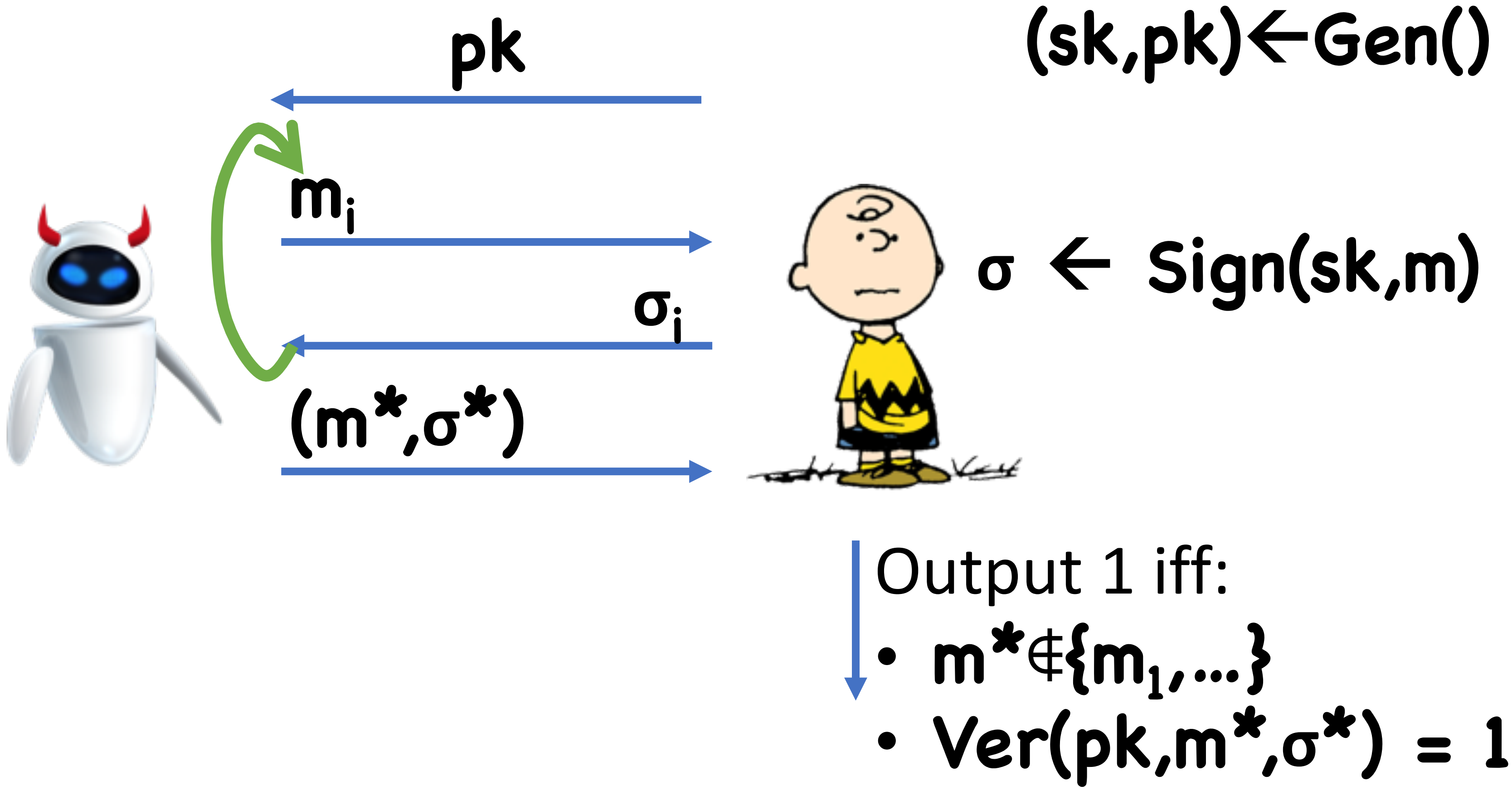


Output 1 iff:

- **m*≠m**
- **Ver(pk,m*,σ*) = 1**

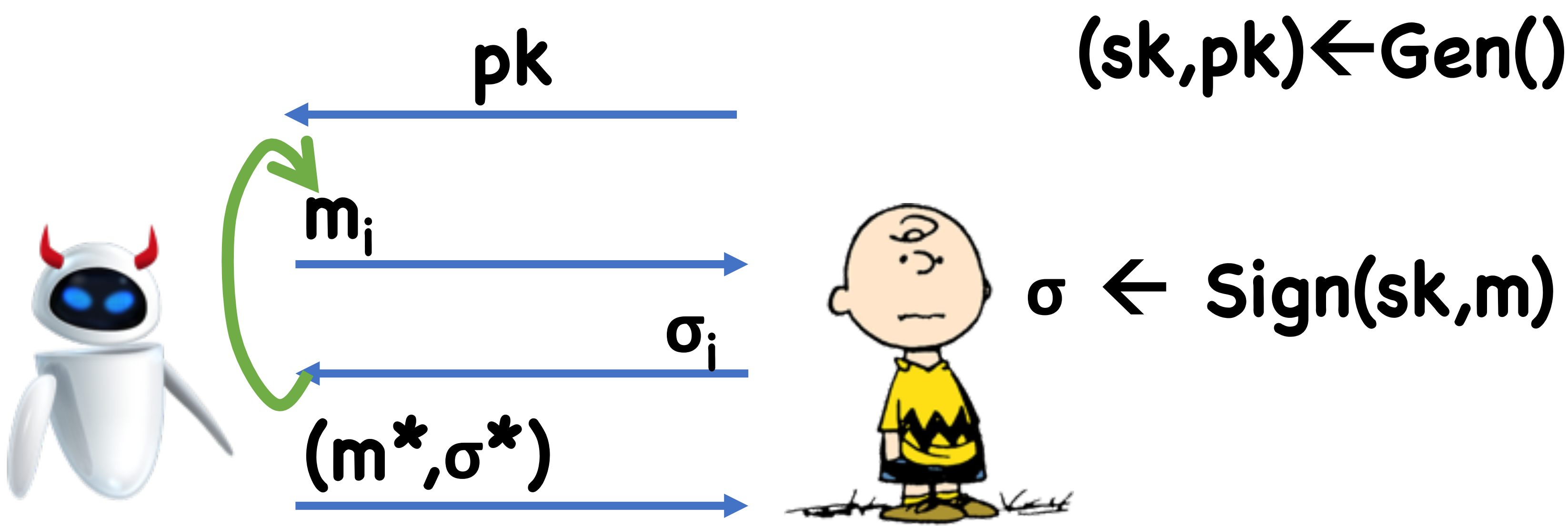
1CMA-Adv() = Pr[ outputs 1]

Many-time Signatures



CMA-Adv() = Pr[ outputs 1]

Strong Security



Output 1 iff:

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

CMA-Adv() = Pr[ outputs 1]

Building Digital Signatures

Non-trivial to construct with provable security

Most efficient constructions have heuristic security

Signatures from TDPs?

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

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

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

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 **H** is "modeled as a random oracle", then $\mathbf{(Gen_{Sig},Sign,Ver)}$ is (strongly) CMA-secure

Basic Rabin Signatures

Gen_{Sig}(): let **p,q** be random large primes
sk = (p,q), pk = N = pq

Sign(sk,m): Solve equation **σ^2 = H(m) mod N** using factors **p,q**

- Output **σ**

Ver(pk,m,σ): σ^2 mod N == H(m)

Problems

H(m) might not be a quadratic residue

Can only sign roughly ¼ of messages

Suppose adversary makes multiple signing queries on the same message

- Receives $\sigma_1, \sigma_2, \ldots$ such that $\sigma_i^2 \bmod N = H(m)$
- After enough tries, may get all 4 roots of **H(m)**
- Suppose $\sigma_1 \neq \pm\sigma_2 \bmod N$
- Then $GCD(\sigma_1 - \sigma_2, N)$ will give a factor

One Solution

Gen_{Sig}(): let **p,q** be primes, **a,b,c** s.t.

- **a** is a non-residue **mod p** and **q**,
- **b** is a residue **mod p** but not **q**,
- **c** is a residue **mod q** but not **p**

$$sk = (p,q,a,b,c),\ pk = (N = pq,\ a,b,c)$$

Sign(sk,m):

- Solve equation $\sigma^2 \in \{1,a,b,c\}\times H(m) \bmod N$
- Output σ

Ver(pk,m,σ): $\sigma^2 \bmod N \in \{1,a,b,c\}\times H(m)$

One Solution

Exactly one of $\{1,a,b,c\}\times H(m)$ is a residue $\bmod N$

$\Rightarrow$ Solution guaranteed to be found

Still have problem that multiple queries on same message will give different roots

One Solution

Possibilities:

- Have signer remember all messages signed

- Choose root that is itself a quadratic residue
  (if **−1** is not a residue mod **p,q**,
  there will be exactly one)

Another Solution

Gen_{Sig}(): let **p,q** be random large primes
sk = (p,q), pk = N = pq

Sign(sk,m): Repeat until successful:

- Choose random $u \leftarrow \{0,1\}^\lambda$
- Solve equation $\sigma^2 = H(m,u) \bmod N$
- Output (u,σ)

Ver(pk,m,(u,σ)): $\sigma^2 \bmod N == H(m,u)$

Another Solution

In expectation, after 4 tries will have success

(Whp) Only ever get a single root of a given **H(m,u)**

Theorem: If factoring is hard and **H** is modeled as a random oracle, then Rabin signatures are (weakly) CMA secure

Another Solution

Sign(sk,m): Repeat until successful:

- Choose random $\mathbf{u} \leftarrow \{0,1\}^\lambda$
- Solve equation $\sigma^2 = H(m,u) \bmod N$ using factors p,q, where $\sigma < (N-1)/2$
- Output (u,σ)

Ver(pk,m,(u,σ)): $\sigma^2 \bmod N == H(m,u) \wedge \sigma < (N-1)/2$

Theorem: If factoring is hard and **H** is modeled as a random oracle, then Rabin signatures are strongly CMA secure