COS433/Math 473: Cryptography

Mark Zhandry

Princeton University

Spring 2020

Announcements

PR2 Due April 19th
HW6 Due April 23rd

Previously on COS 433…

Public Key Cryptography

Key Distribution from Obfuscation

Let **F,F^{-1}** be a block cipher

P →

← x

$k \leftarrow \{0,1\}^\lambda$

$P \leftarrow \mathrm{Obf}(\ F(k,\ \cdot)\ )$

↓

$r \leftarrow F^{-1}(k,x)$

$r \leftarrow \{0,1\}^\lambda$

$x \leftarrow P(r)$

↓

r

Key Distribution From Obfuscation

For decades, many attempts at commercial code obfuscators

- Simple operations like variable renaming, removing whitespace, re-ordering operations

Really only a “speed bump” to determined adversaries

- Possible to recover something close to original program (including cryptographic keys)

Don’t use commercially available obfuscators to hide cryptographic keys!

Practical Key Exchange

Instead of obfuscating a general PRP, we will define a specific abstraction that will enable key agreement

Then, we will show how to implement the abstraction using number theory

Today

Trapdoor Permutations

Trapdoor Permutations

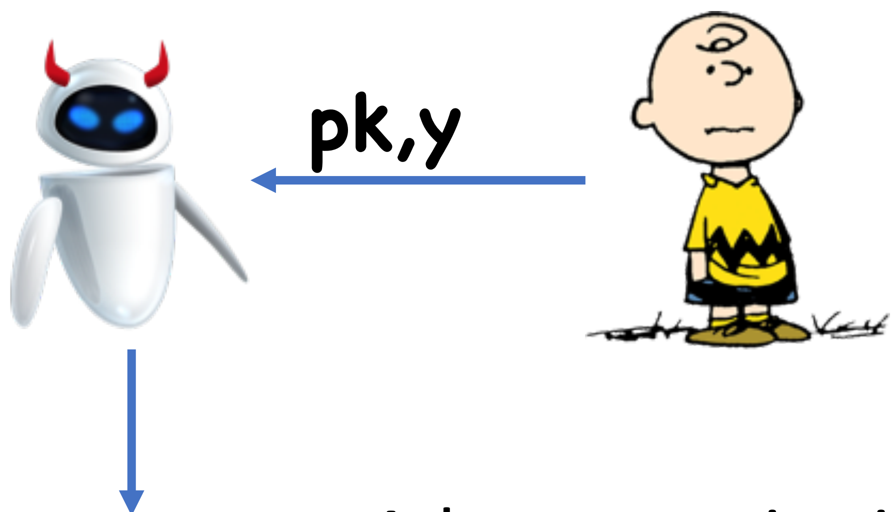
Domain $\mathbf{X}$

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

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

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

Trapdoor Permutation Security



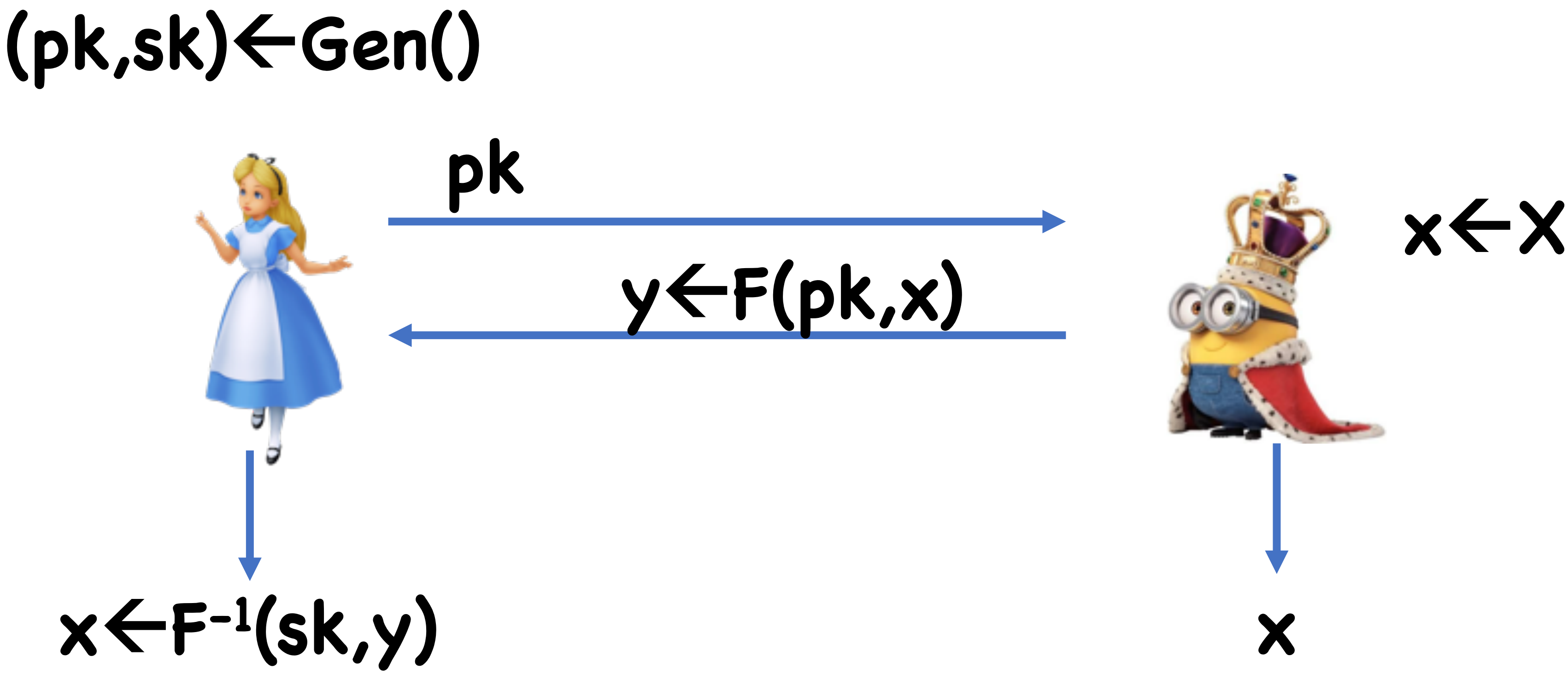
(sk,pk)←Gen()
x←X
y←F(pk,x)

x'

Adversary wins if **x=x'**

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

Key Distribution from TDPs



Analysis

Correctness follows from correctness of TDP

Security:

- By TDP security, adversary cannot compute **x**
- However, **x** is distinguishable from a random key

Hardcore Bits

Let **F** be a one-way function with domain **D**, range **R**

Definition: A function **h:D→{0,1}** is a hardcore bit for **F** if, for any polynomial time , ∃ negligible **ε** such that:

| Pr[1←**(F(x), h(x)), x←D]**

– Pr[1←**(F(x), b), x←D,b←{0,1}] | ≤ ε(λ)**

In other words, even given **F(x)**, hard to guess **h(x)**

Examples of Hardcore Bits

Define **lsb(x)** as the least significant bit of **x**

For $x \in Z_N$, define **Half(x)** as **1** iff $0 \leq x < N/2$

Theorem: Let **p** be a prime, and $F: Z_p^* \rightarrow Z_p^*$ be $F(g,x) = (g, g^x \bmod p)$

Half is a hardcore bit for **F** (assume **F** is one-way)

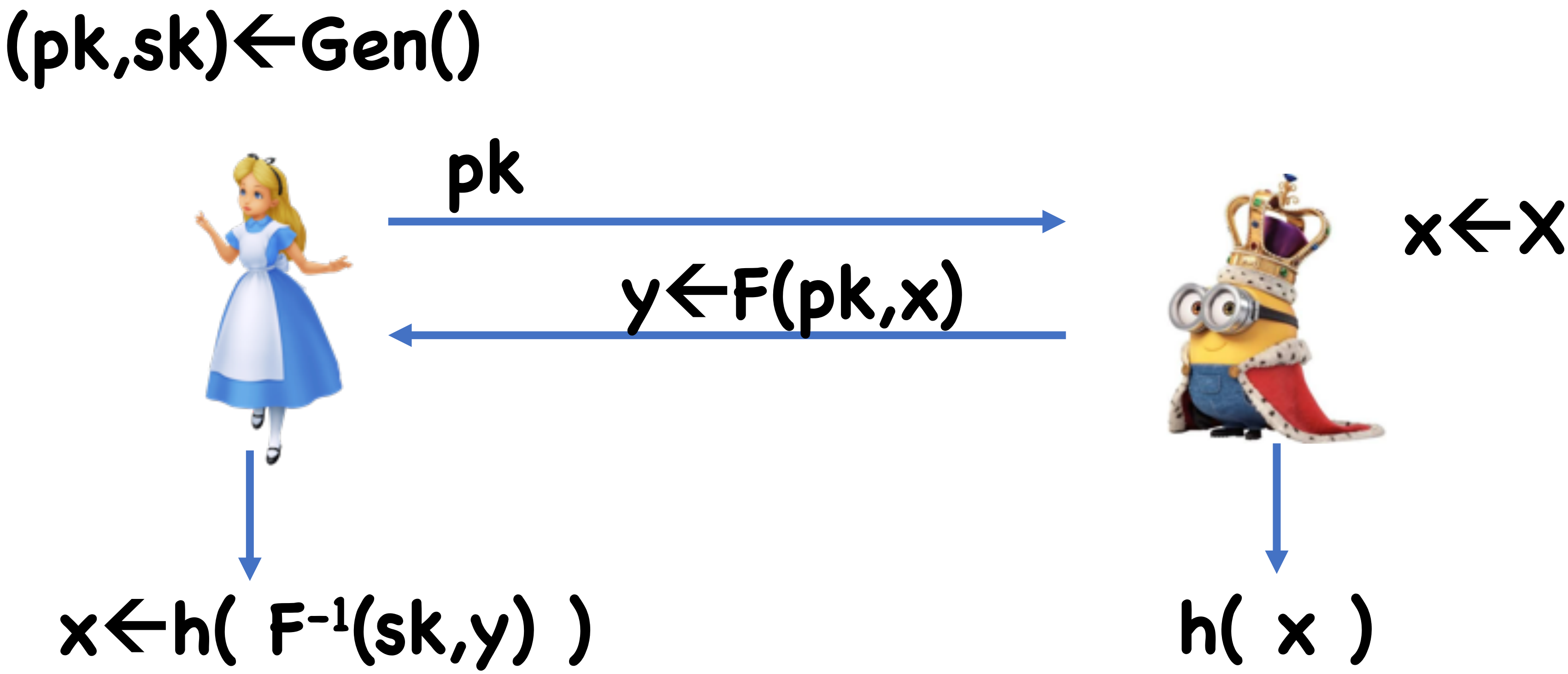
Theorem: Let **N** be a product of two large primes **p,q**, and $F: Z_N^* \rightarrow Z_N^*$ be $F(x) = x^e \bmod N$ for some **e** relatively prime to $(p-1)(q-1)$

Lsb and Half are hardcore bits for **F** (assuming RSA)

Theorem: Let **N** be a product of two large primes **p,q**, and $F: Z_N^* \rightarrow Z_N^*$ be $F(x) = x^2 \bmod N$

Lsb and Half are hardcore bits for **F** (assuming factoring)

Key Distribution from TDPs



h a hardcore bit for **F(pk, ·)**

Theorem: If **h** is a hardcore bit for **F(pk, ·),** then protocol is secure

Proof:

- **(Trans,k) = ((pk,y), h(x))**
- Hardcore bit means indist. from **((pk,y), b)**

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 $y = x^e \bmod N$

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

Caveats

RSA is not a true TDP as defined

- Why???
- What's the domain?

Nonetheless, distinction is not crucial to most applications

- In particular, works for key agreement protocol

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** of 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

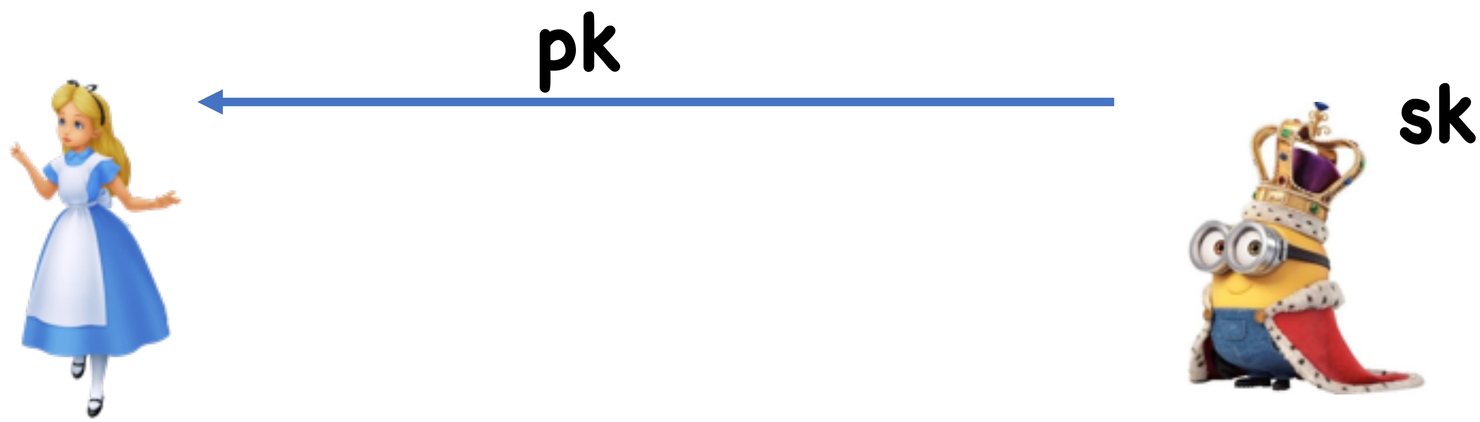
Proof:

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

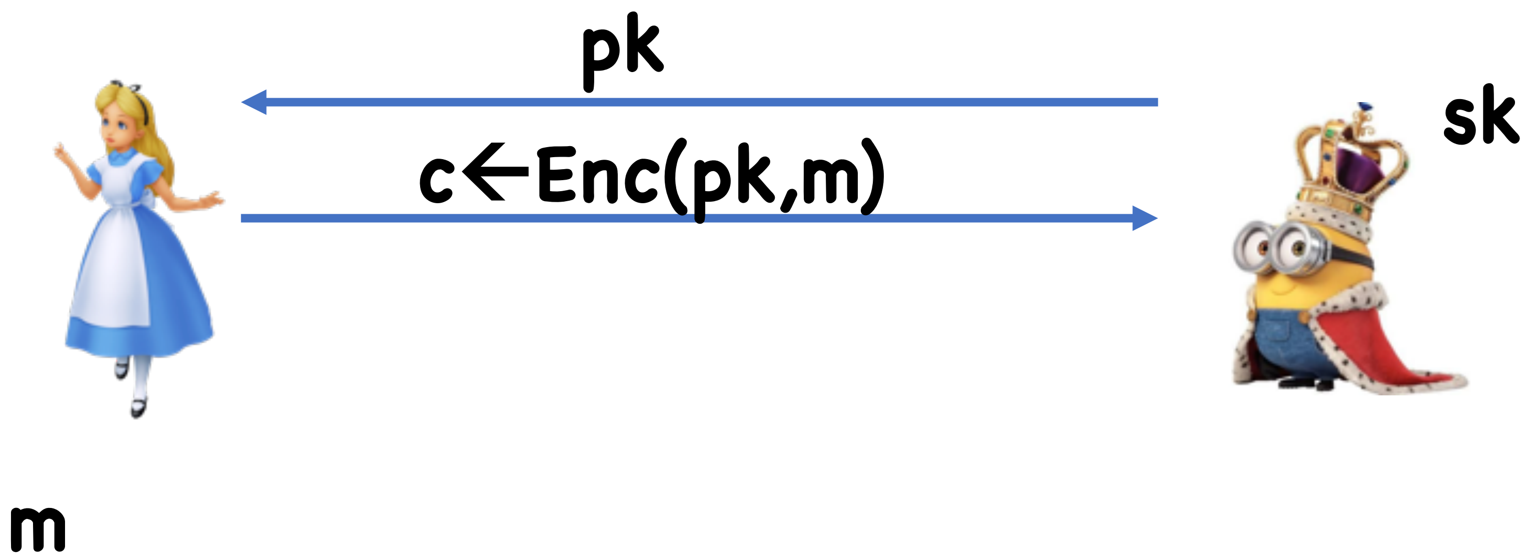
What if only CDH holds, but DDH is easy?

Public Key Encryption

Public Key Encryption



Public Key Encryption



Public Key Encryption

pk

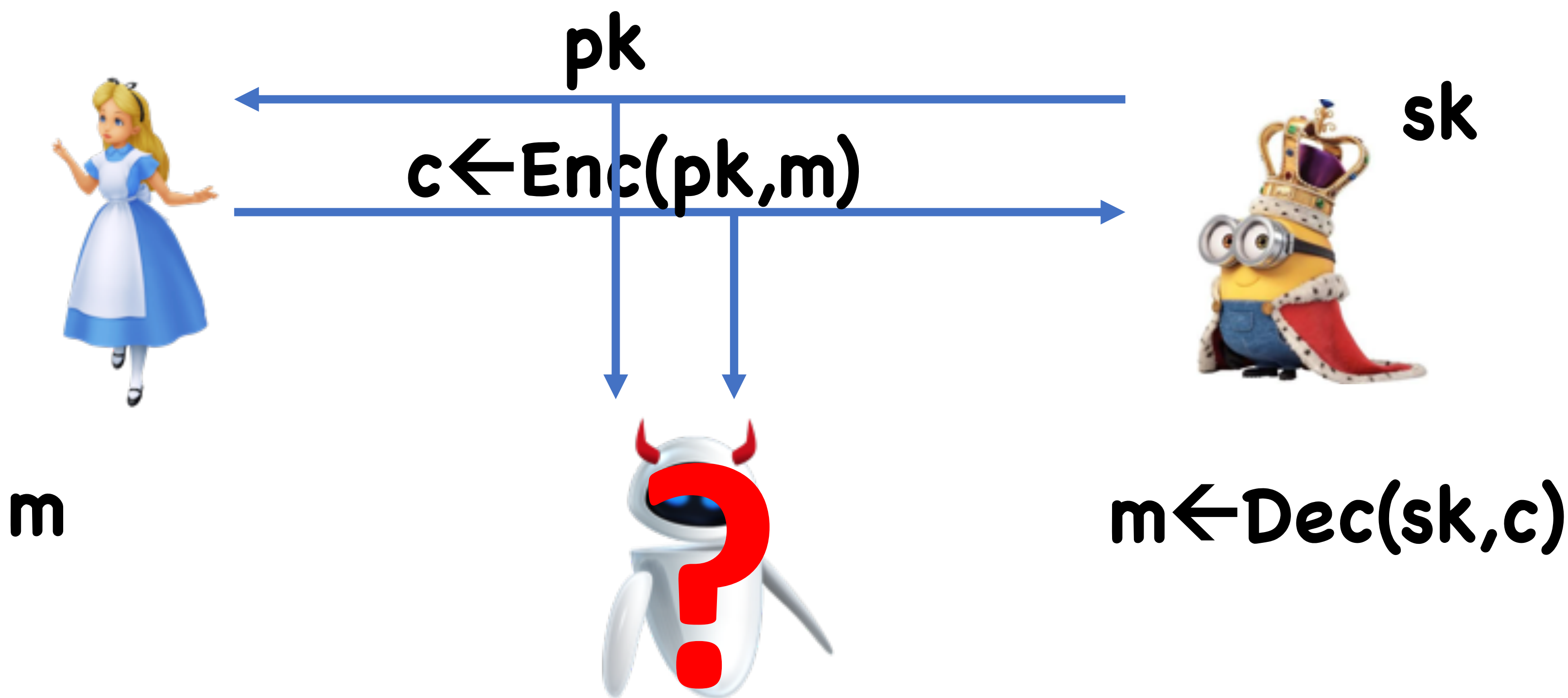
$c \leftarrow \mathrm{Enc}(pk,m)$

sk

m

$m \leftarrow \mathrm{Dec}(sk,c)$

Public Key Encryption

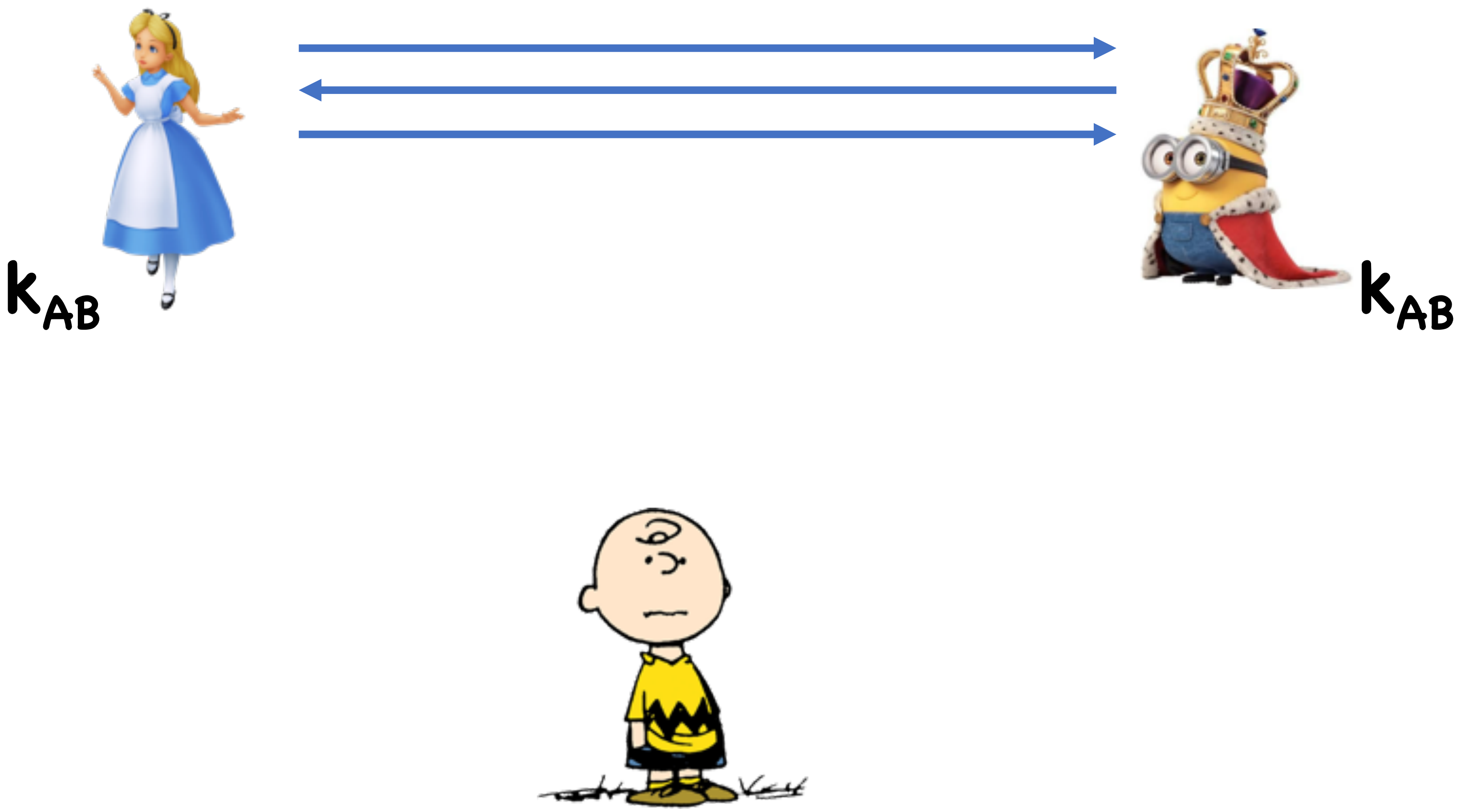


PKE vs Key Agreement

Key agreement:

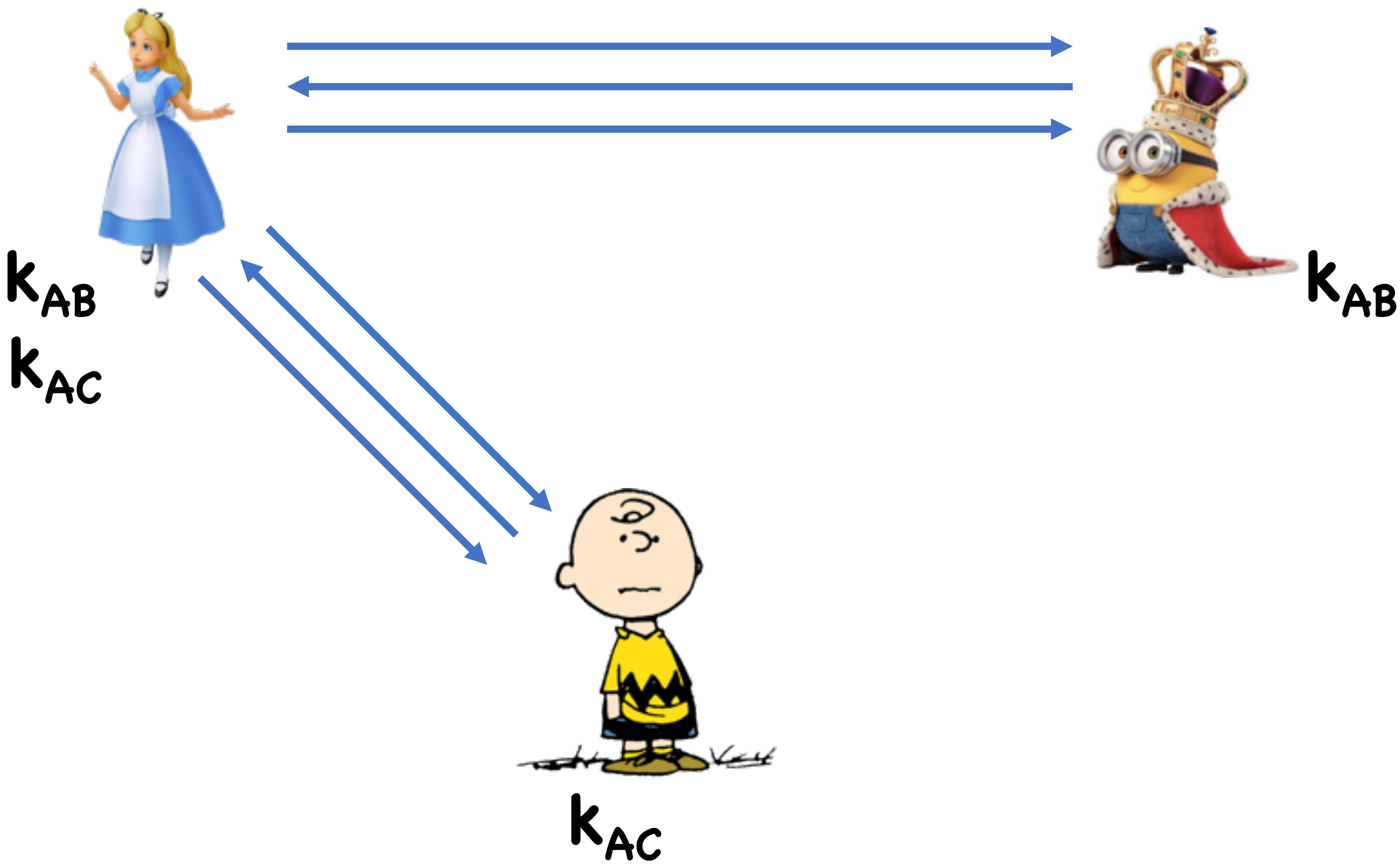
PKE vs Key Agreement

Key agreement:



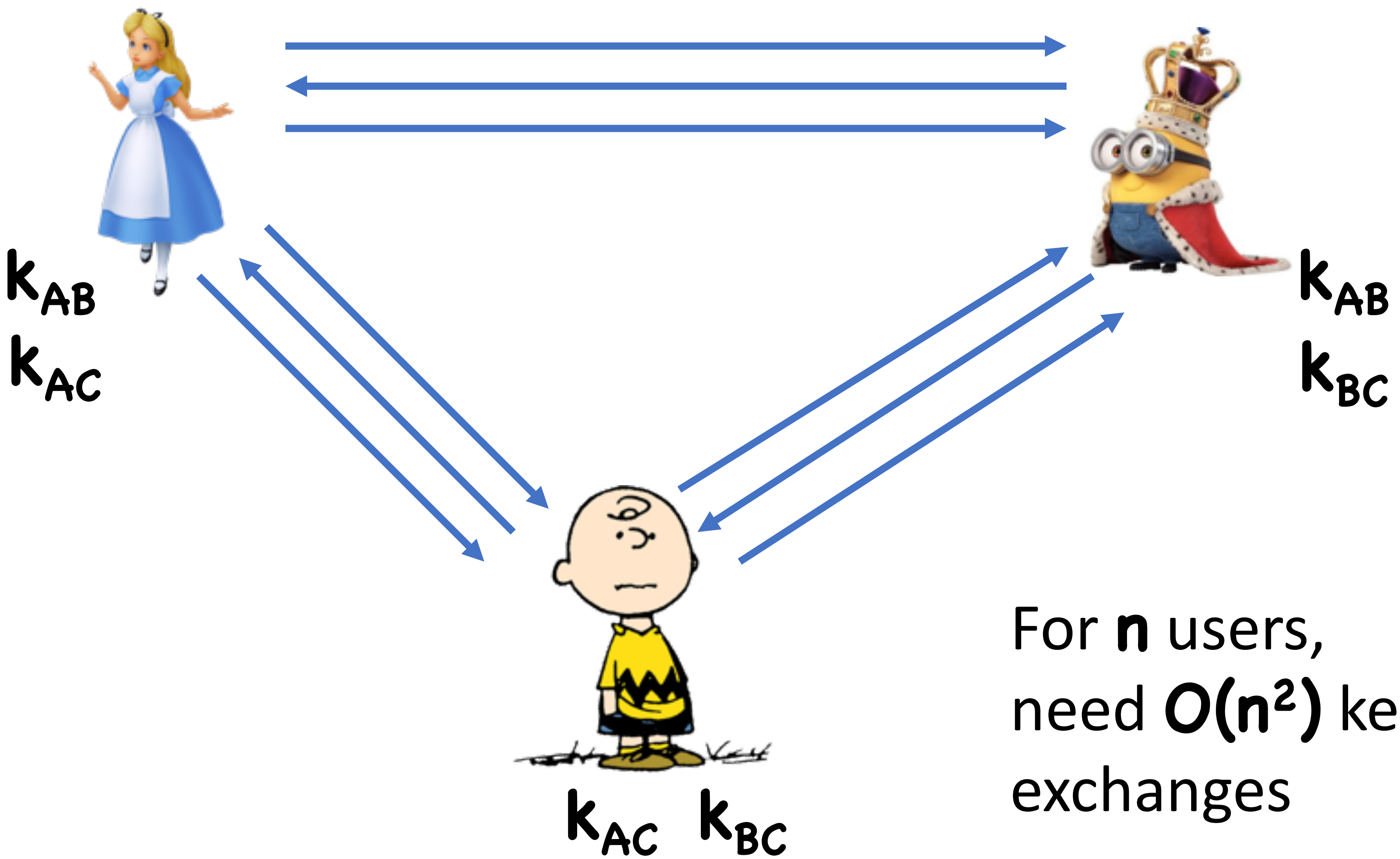
PKE vs Key Agreement

Key agreement:



PKE vs Key Agreement

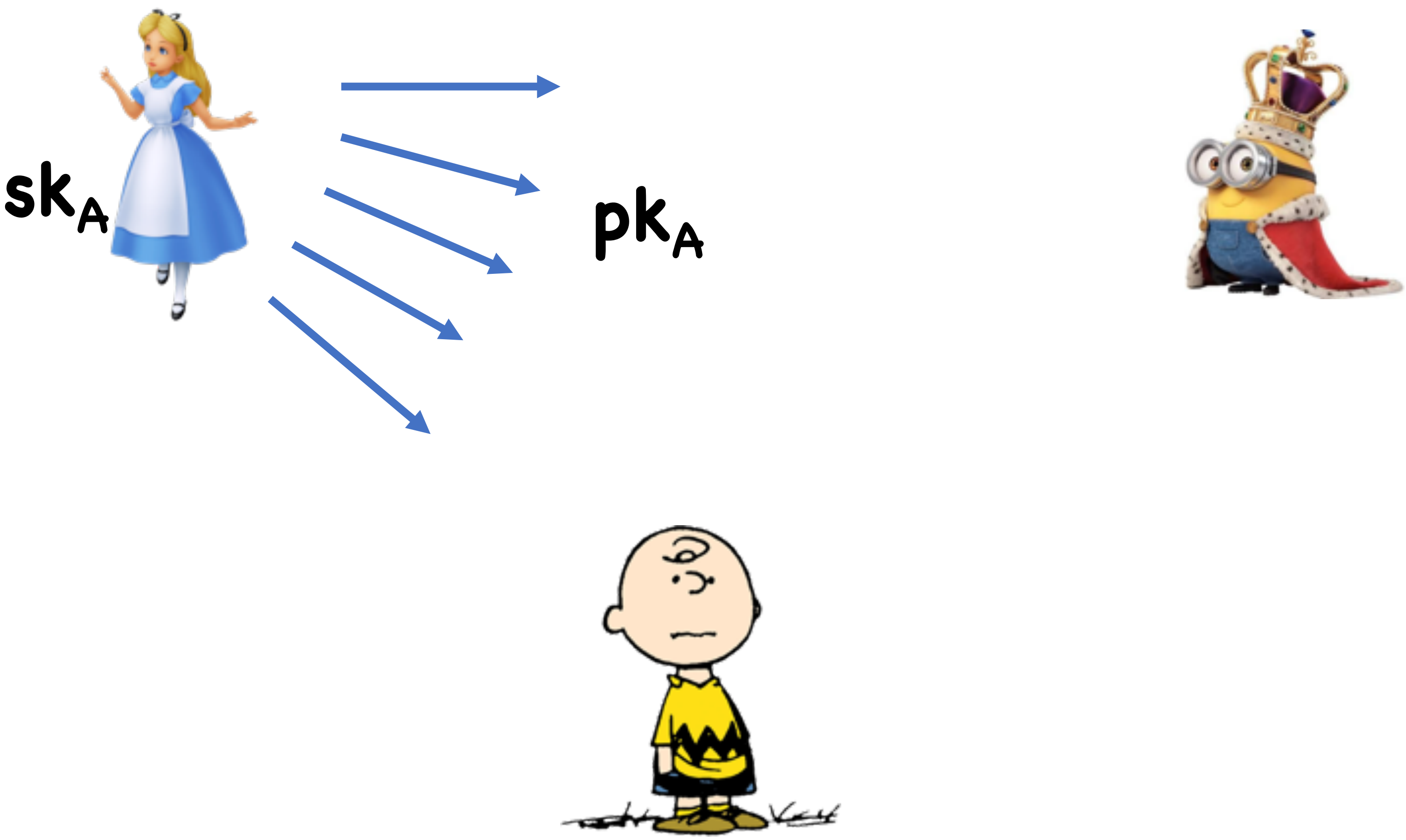
Key agreement:



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

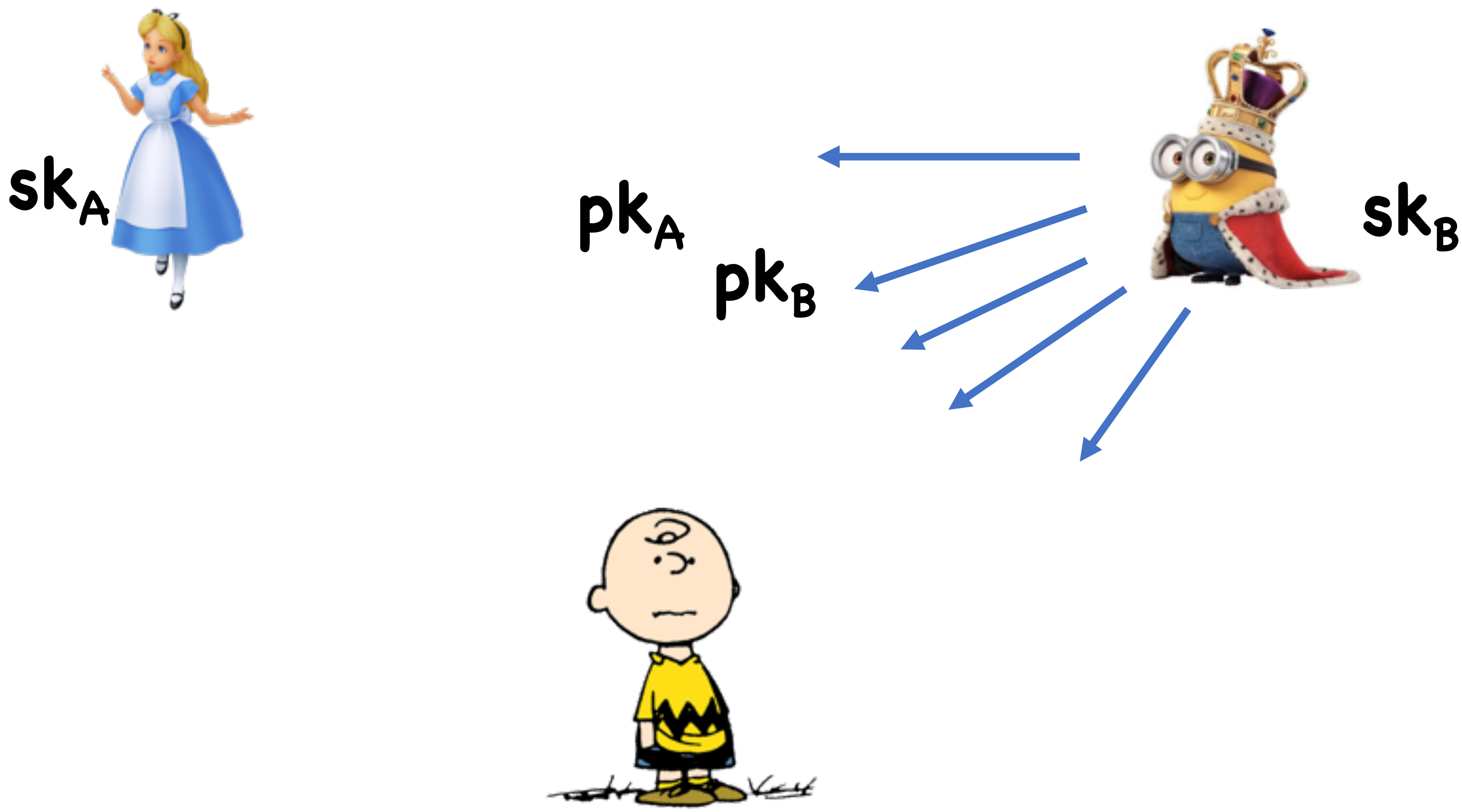
PKE vs Key Agreement

PKE:



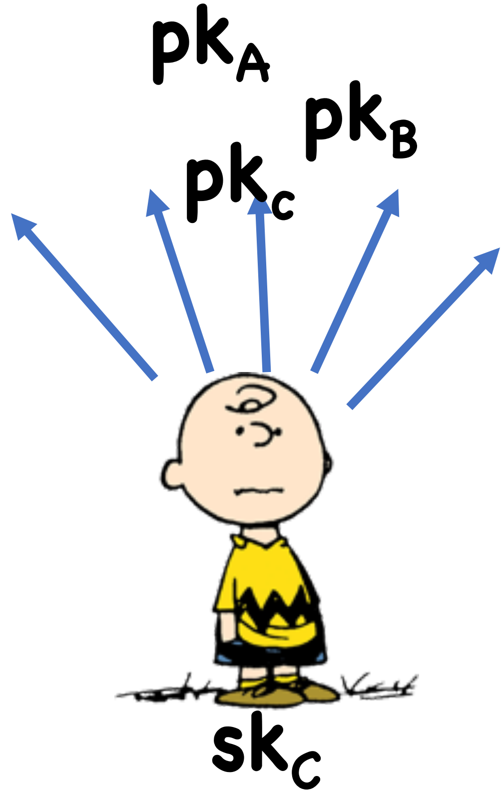
PKE vs Key Agreement

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

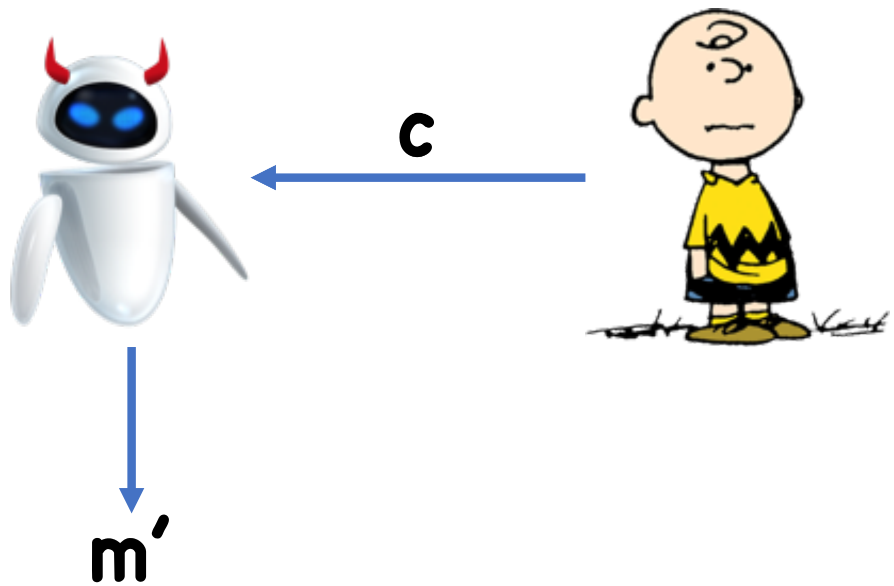
One-way security

Semantic Security

CPA security

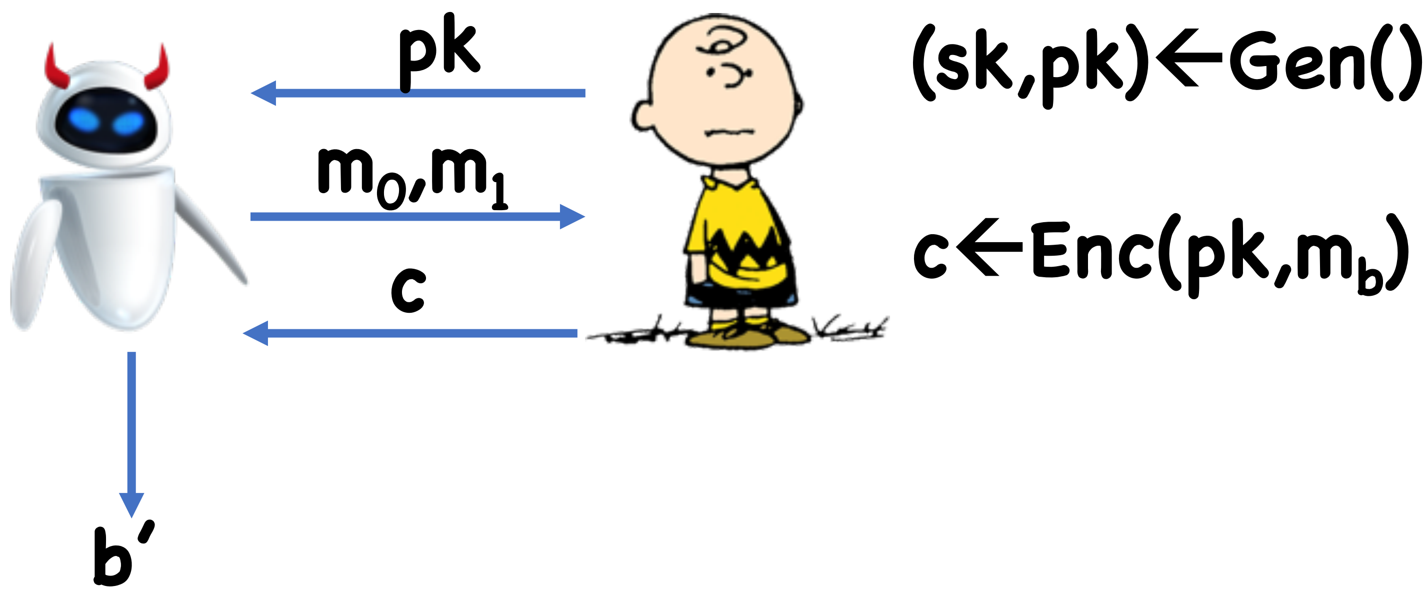
CCA Security

One-way Security

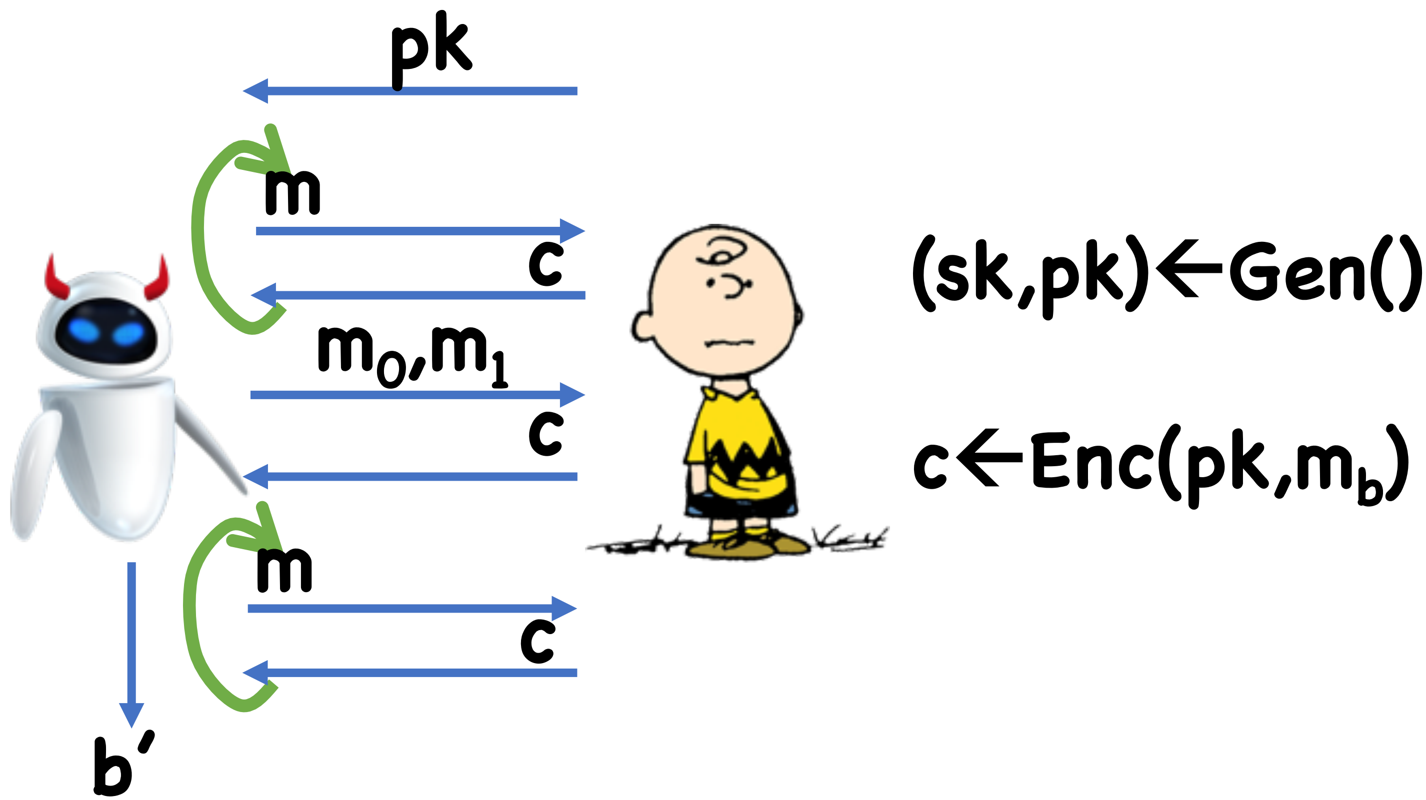


(sk,pk)←Gen()
m←M
c←Enc(pk,m)

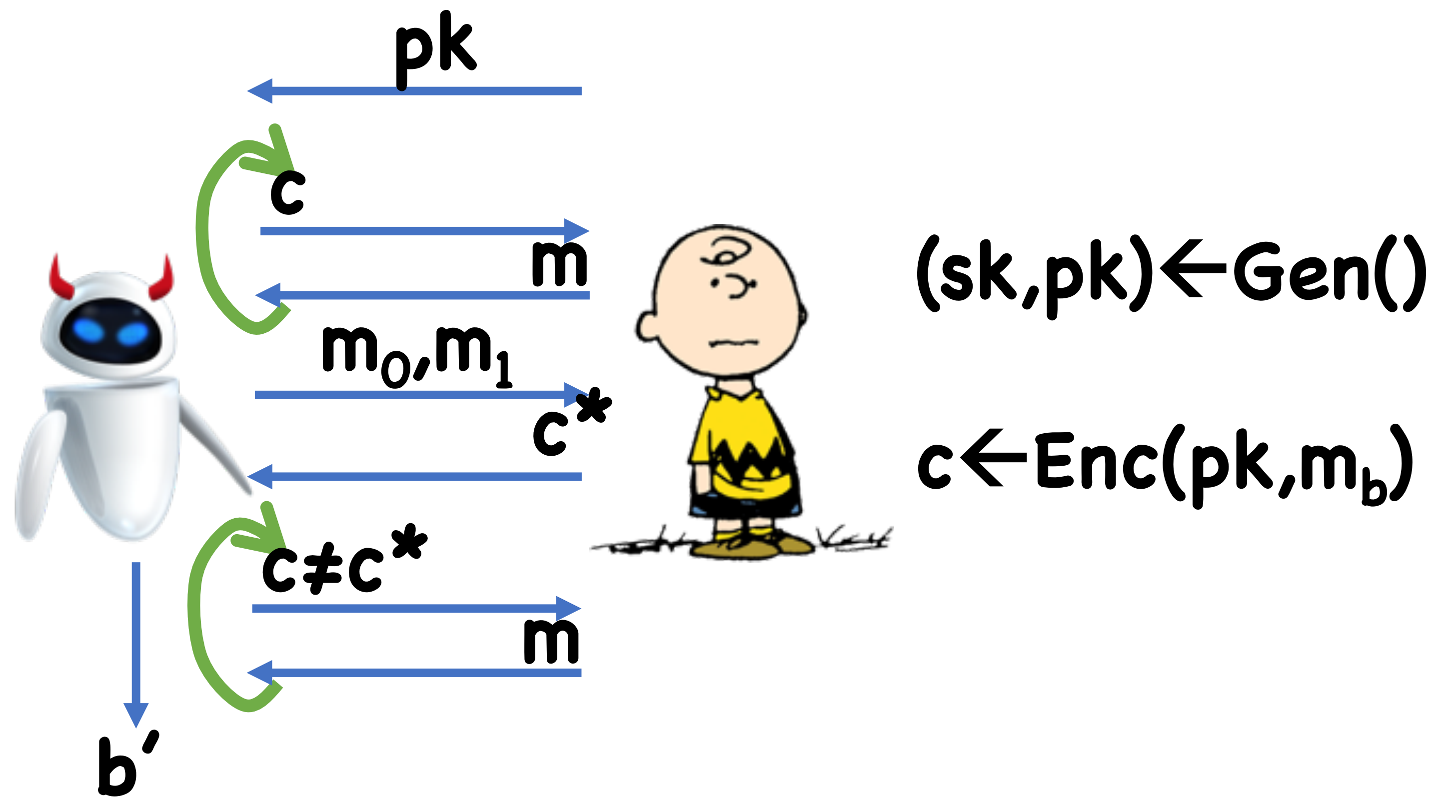
Semantic Security



CPA Security



CCA Security



Question: Which two notions are equivalent?

One-Way Encryption from TDPs

Gen_E() = Gen_{TDP}()

Enc(pk,m): Output **c = F(pk,m)**

Dec(sk,c): Output **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 **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 **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 **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?

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 $\mathbf{g^a, g^r, g^{ar}\times m_0}$

Practical Considerations

Number theory is computationally expensive

- Need big number arithmetic

Symmetric crypto (e.g. block ciphers) much faster

Want to minimize use of number theory, and rely mostly on symmetric crypto

Hybrid Encryption

Let $(\mathbf{Gen}_{PKE}, \mathbf{Enc}_{PKE}, \mathbf{Dec}_{PKE})$ be a PKE scheme, $(\mathbf{Enc}_{SKE}, \mathbf{Dec}_{SKE})$ a SKE scheme

Gen() = Gen$_{PKE}$**()**
Enc(pk, m): k←K, c=(Enc$_{PKE}$**(pk,k), Enc**$_{SKE}$**(k,m))**
Dec(sk, (c$_0$**, c**$_1$**):**

- **k←Dec**$_{PKE}$**(sk,c**$_0$**)**
- **m←Dec**$_{SKE}$**(k,c**$_1$**)**

Now PKE used to encrypt something small (e.g. 128 bits), SKE used to encrypt actual message (say, GB's)

Hybrid Encryption

Theorem: If $(\mathsf{Gen}_{PKE}, \mathsf{Enc}_{PKE}, \mathsf{Dec}_{PKE})$ is CPA secure and $(\mathsf{Enc}_{SKE}, \mathsf{Dec}_{SKE})$ is one-time secure, then $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is CPA secure

Hybrid 0: $(\mathsf{Enc}_{PKE}(pk,k),\ \mathsf{Enc}_{SKE}(k,m_0))$
Hybrid 1: $(\mathsf{Enc}_{PKE}(pk,k'),\ \mathsf{Enc}_{SKE}(k,m_0))$
Hybrid 2: $(\mathsf{Enc}_{PKE}(pk,k'),\ \mathsf{Enc}_{SKE}(k,m_1))$
Hybrid 3: $(\mathsf{Enc}_{PKE}(pk,k),\ \mathsf{Enc}_{SKE}(k,m_1))$

CCA-Secure Encryption

Non-trivial to construct with provable security

Most efficient constructions have heuristic security

CCA Secure PKE from TDPs

Let $(\mathsf{Enc}_{SKE},\mathsf{Dec}_{SKE})$ 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

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

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

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

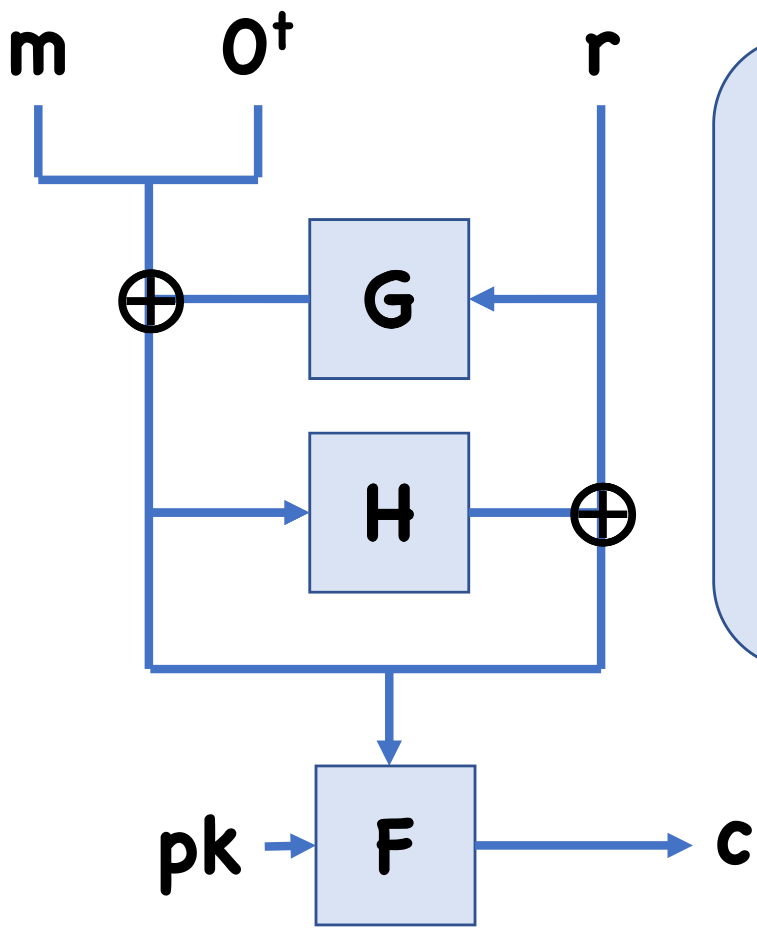
$\mathbf{Dec_{PKE}(sk, (c, d))}$:

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

CCA Secure PKE from TDPs

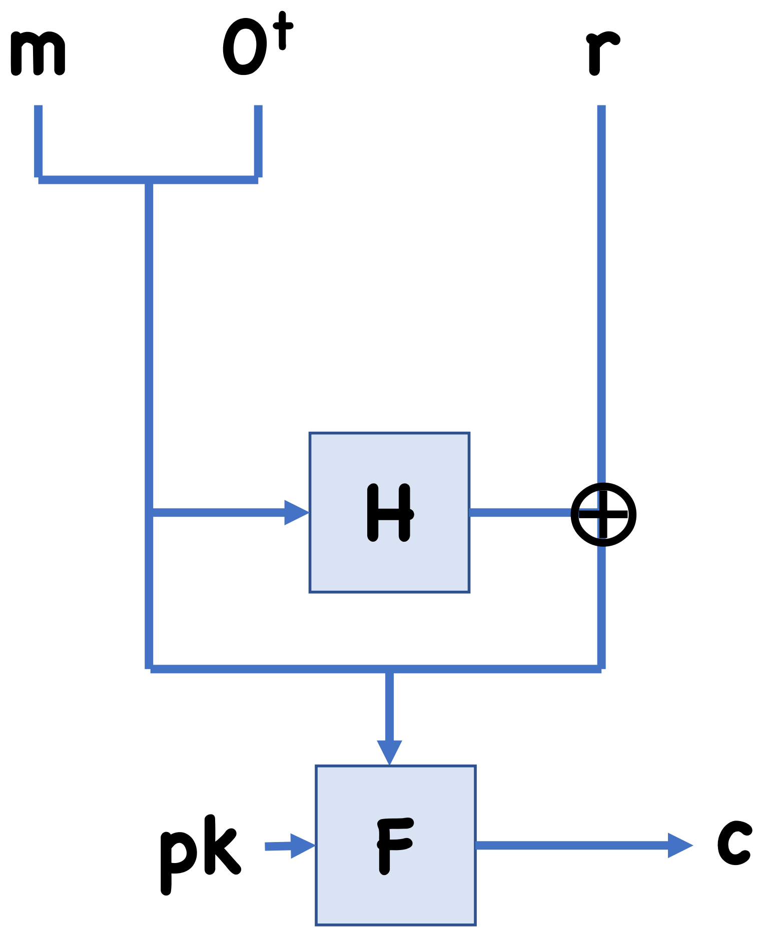
Theorem: If $(\mathsf{Enc}_{SKE},\mathsf{Dec}_{SKE})$ is a CCA-secure secret key encryption scheme, $(\mathsf{Gen},\mathsf{F},\mathsf{F}^{-1})$ is a TDP, and H is modeled as a random oracle, then $(\mathsf{Gen}_{PKE},\mathsf{Enc}_{PKE},\mathsf{Dec}_{PKE})$ is a CCA secure public key encryption scheme

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

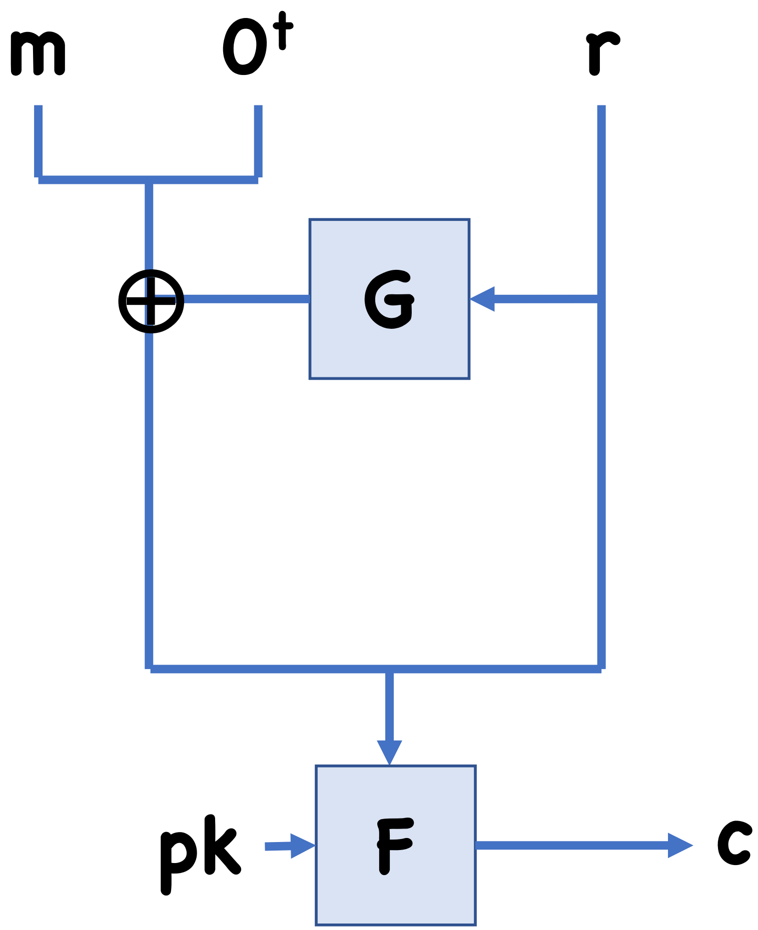


$c = F(pk, (m, 0^t, y))$

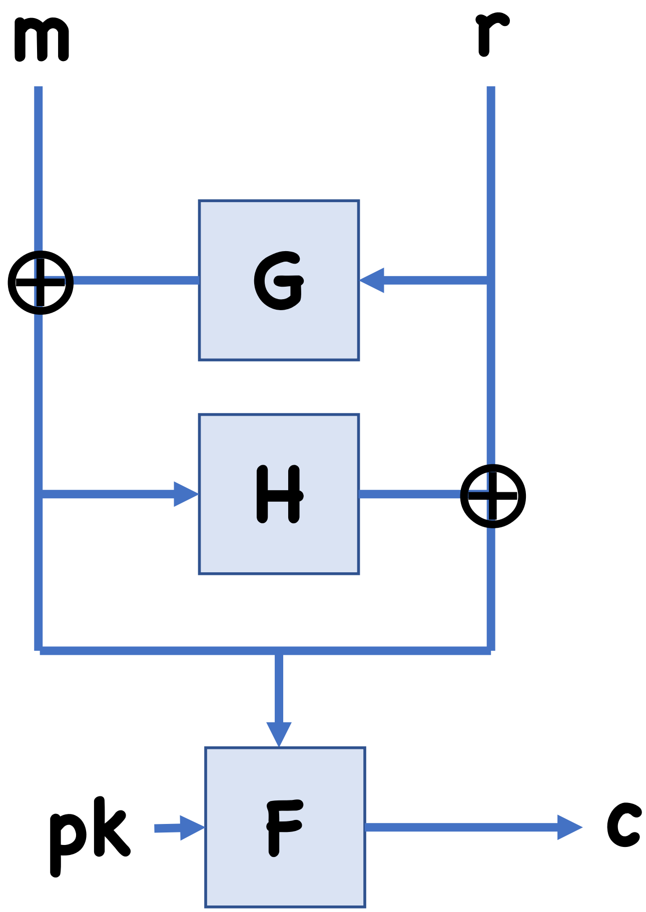
May contain **m** in the clear

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

Insecure OAEP Variants



Why padding?



All ciphertexts decrypt to valid messages

- Makes it hard to argue security

Announcements

PR2 Due April 19th

HW6 Due April 23rd