

COS433/Math 473: Cryptography

Mark Zhandry

Princeton University

Fall 2020

Announcements/Reminders

Last day to turn in HW4

HW5 due Nov 10

PR2 will be released today

Heads up: Lecture 18 (next lecture) will be pre-recorded

Previously on COS 433...

Integer Factorization

Given an integer **N**, find it's prime factors

Studied for centuries, presumed difficult

- Grade school algorithm: $O(N^{1/2})$
- Better algorithms using birthday paradox: $O(N^{1/4})$
- Even better assuming G. Riemann Hyp.: $O(N^{1/5})$
- Still better heuristic algorithms:
 $\exp(C (\log N)^{1/3} (\log \log N)^{2/3})$
- However, all require super-polynomial time in bit-length of **N**

Quadratic Residues

Definition: y is a quadratic residue mod N if there exists an x such that $y = x^2 \pmod{N}$. x is called a “square root” of y

Ex:

- Let p be a prime, and $y \neq 0$ a quadratic residue mod p . How many square roots of y ?
- Let $N=pq$ be the product of two primes, y a quadratic residue mod N . Suppose $y \neq 0 \pmod{p}$ and $y \neq 0 \pmod{q}$. How many square roots?

Theorem: If the factoring assumption holds, then the QR assumption holds

RSA Problem

Given

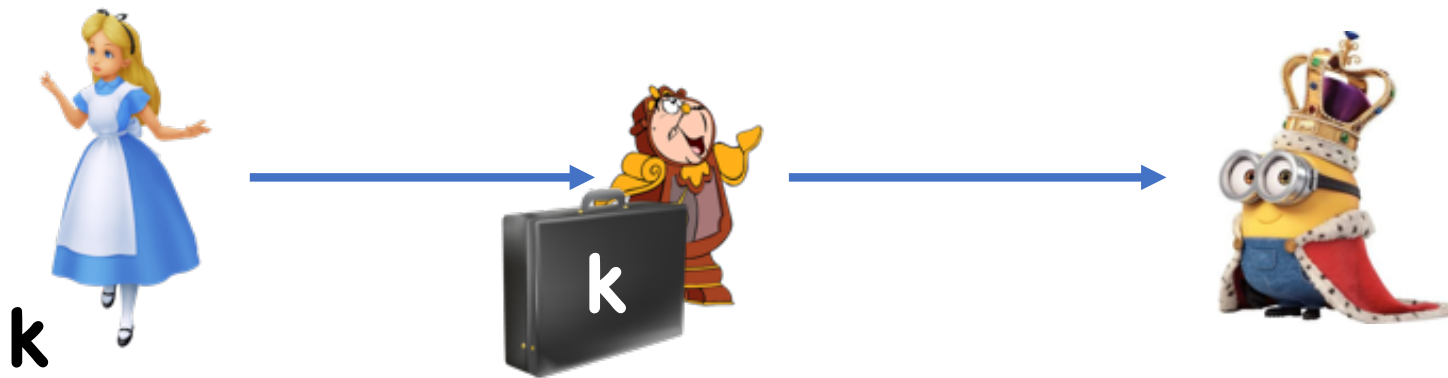
- $N = pq$,
- e such that $\text{GCD}(e, p-1) = \text{GCD}(e, q-1) = 1$,
- $y = x^e \pmod N$ for a random x

Find x

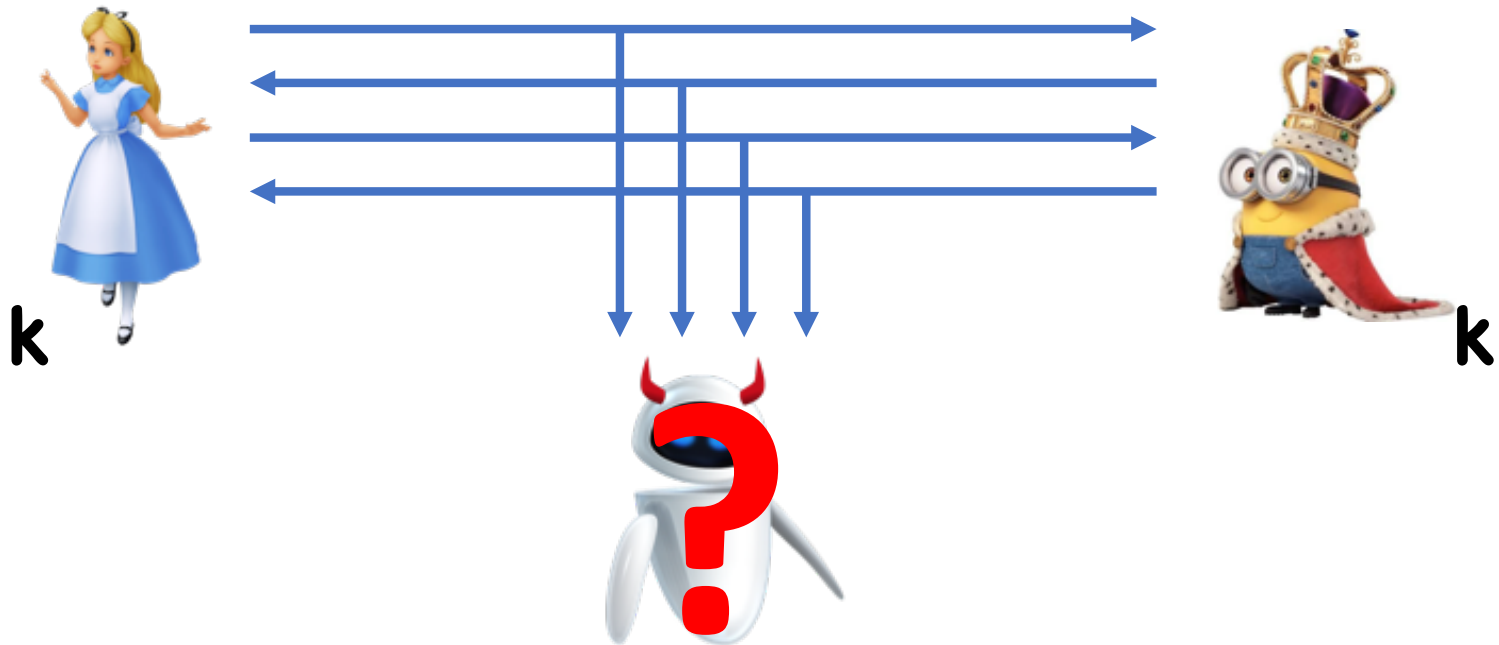
Injectivity means cannot base hardness on factoring,
but still conjectured to be hard

Public Key Cryptography

How do Alice & Bob get **k**?

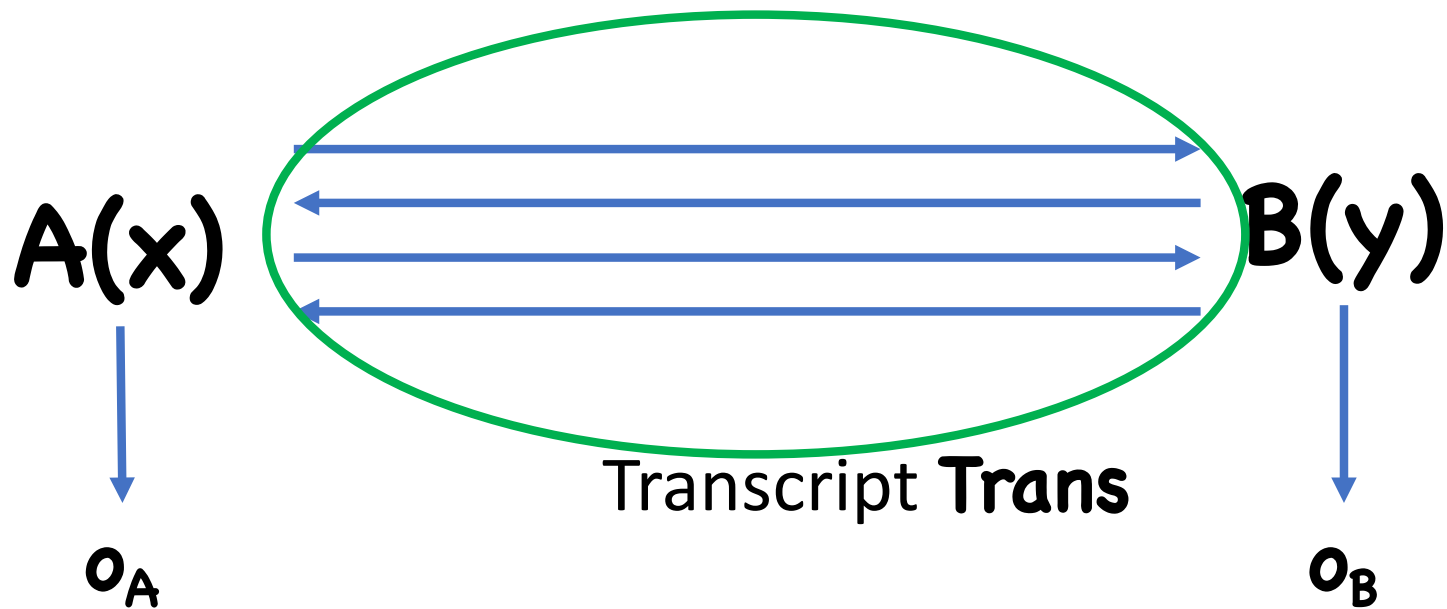


Public Key Distribution



Interactive Protocols

Pair of interactive (randomized) algorithms **A**, **B**



Write $(\text{Trans}, o_A, o_B) \leftarrow (A, B)(x, y)$

Public Key Distribution

Pair of interactive algorithms **A, B**

Correctness:

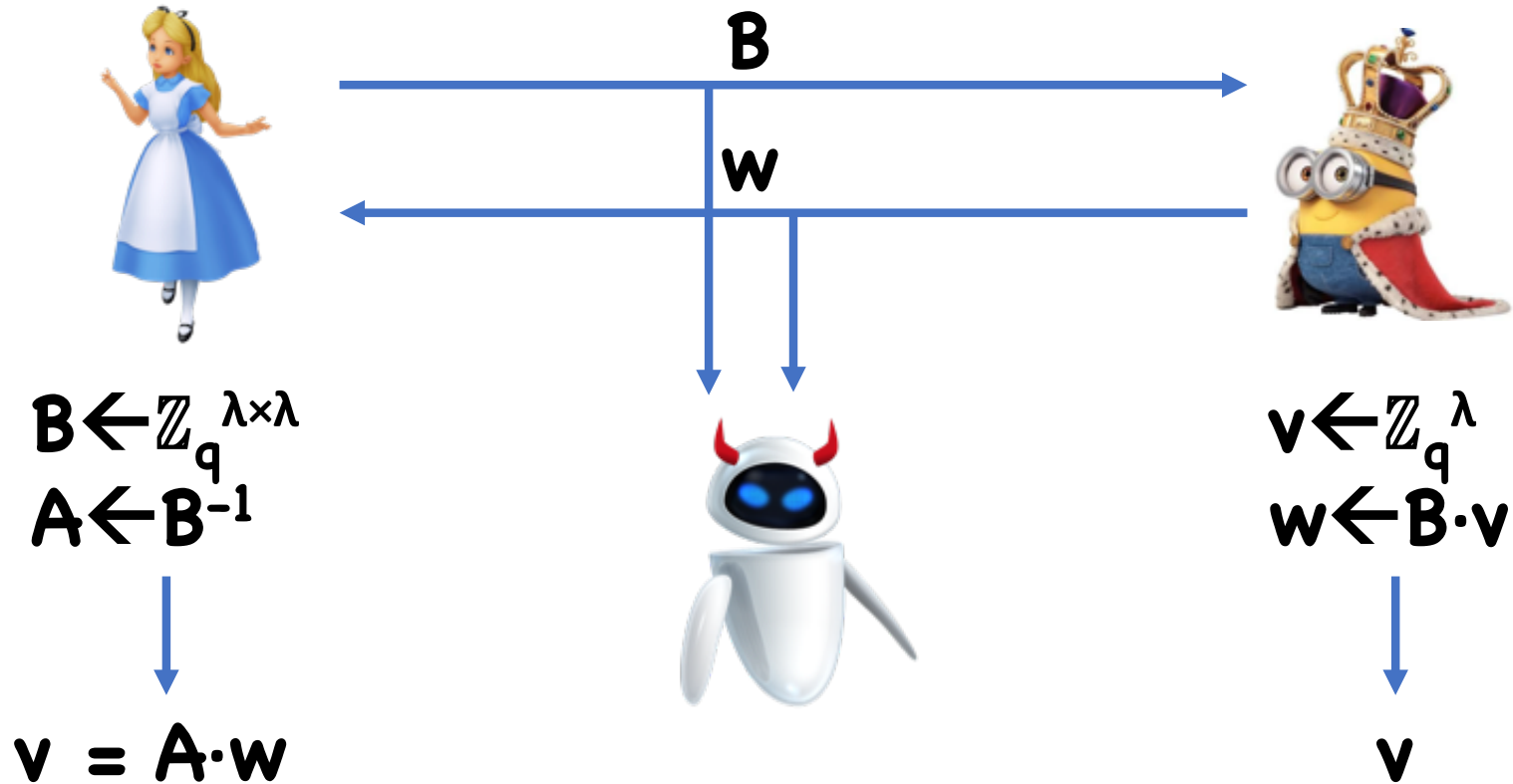
$$\Pr[o_A = o_B : (\text{Trans}, o_A, o_B) \leftarrow (A, B)()] = 1$$

Shared key is $\mathbf{k} := o_A = o_B$

- Define $(\text{Trans}, \mathbf{k}) \leftarrow (A, B)()$

Security: $(\text{Trans}, \mathbf{k})$ is computationally indistinguishable from $(\text{Trans}, \mathbf{k}')$ where $\mathbf{k}' \leftarrow \mathbf{K}$ independent of \mathbf{k}

Matrix Multiplication Approach



Running Times?

Bob: $O(\lambda^2)$

Eve: $O(\lambda^\omega)$ where $\omega \leq 2.373$

Alice: $O(\lambda^\omega)$

Assuming Matrix Multiplication exponent $\omega > 2$,
adversary must work harder than honest users

inverse to **B**

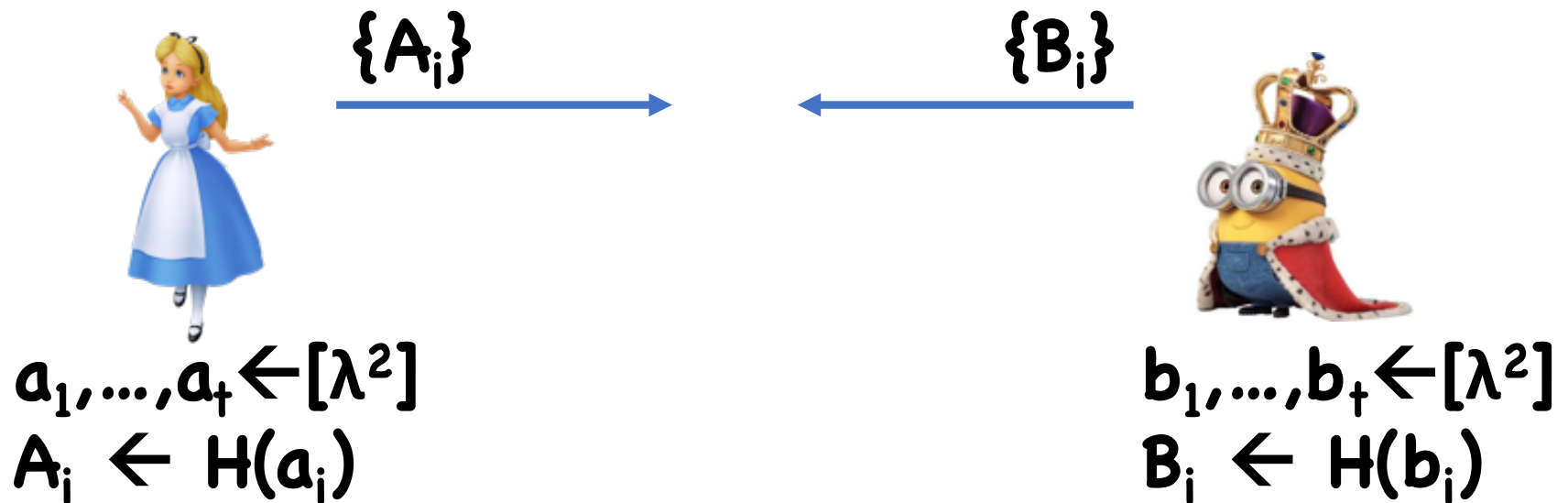
- Output **(A,B)**

Today

Public key cryptography continued

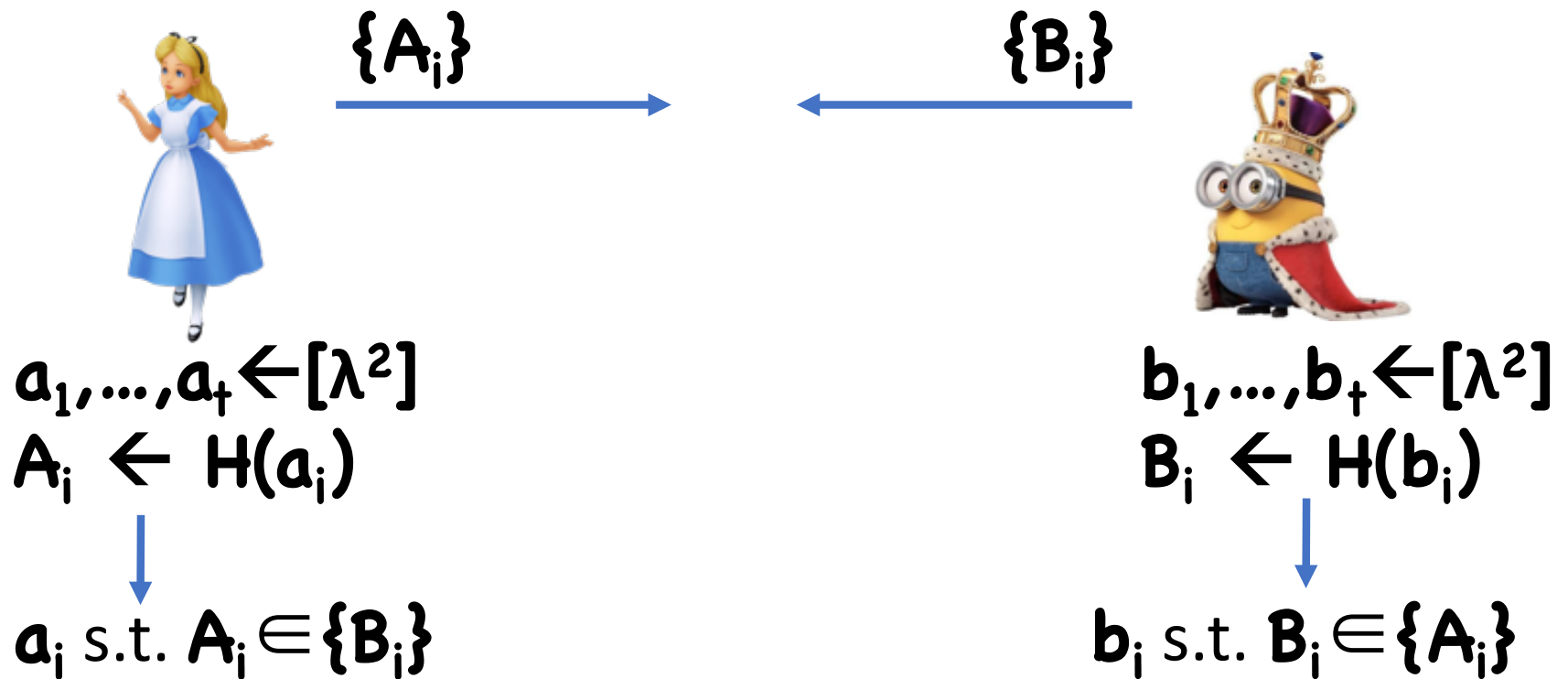
Merkle Puzzles

Let H be some hash function with domain $[\lambda^2]=\{1,\dots,\lambda^2\}$



Merkle Puzzles

Let H be some hash function with domain $[\lambda^2]=\{1,\dots,\lambda^2\}$



Analysis

Protocol succeeds iff:

- H is injective (why?)
- $\{A_i\} \cap \{B_i\} \neq \emptyset$ (equiv, $\{a_i\} \cap \{b_i\} \neq \emptyset$)

What does t need to be to make $\{A_i\} \cap \{B_i\} \neq \emptyset$?

If adversary can only query H on various inputs, how many queries needed?

Limitations

Both matrix multiplication and Merkle puzzle approaches have a polynomial gap between honest users and adversaries

To make impossible for extremely powerful adversaries, need at least $\lambda^2 > 2^{80}$

- Special-purpose hardware means λ needs to be even bigger
- Honest users require time at least $\lambda=2^{40}$
- Possible, but expensive

Limitations

Instead, want want a super-polynomial gap between honest users and adversary

- Just like everything else we've seen in the course

Key Distribution from Obfuscation

Software obfuscation:

- Compile programs into unreadable form
(intentionally)

```
@P=split//, ".URRUU\c8R";@d=split//, "\nrekcah xinU / lreP rehtona tsuJ";sub p{
@p{"r$p", "u$p"}=(P,P);pipe"r$p", "u$p";++$p;($q*=2)+=$f=!fork;map{$P=$P[$f^ord
($p{$_})&6];$p{$_}=/^$P/ix?$P:close$_}keys%p}p;p;p;p;p;p;map{$p{$_}=~/^[P.]/&&
close$_}%p;wait until$?;map{/^r/&&<$_>}%p;$_=$d[$q];sleep rand(2)if/\S/;print
```

Key Distribution from Obfuscation

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



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

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

Key Distribution from Obfuscation

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



$k \leftarrow \{0,1\}^\lambda$
 $P \leftarrow \text{Obf}(F(k, \cdot))$

$r \leftarrow \{0,1\}^\lambda$
 $x \leftarrow P(r)$

Key Distribution from Obfuscation

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



$$k \leftarrow \{0,1\}^\lambda$$
$$P \leftarrow \text{Obf}(F(k, \cdot))$$

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

$$r \leftarrow \{0,1\}^\lambda$$
$$x \leftarrow P(r)$$

$$\downarrow$$
$$r$$

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!

Key Distribution From Obfuscation

Recently (2013), new type of obfuscator has been developed

- Much stronger security guarantees
- Based on mathematical tools
- Many cryptographic applications beyond public key distribution

Downside?

- Extraordinarily impractical (currently)

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

Trapdoor Permutations

Domain X

Gen(): outputs (pk, sk)

F(pk, $x \in X$) = $y \in X$

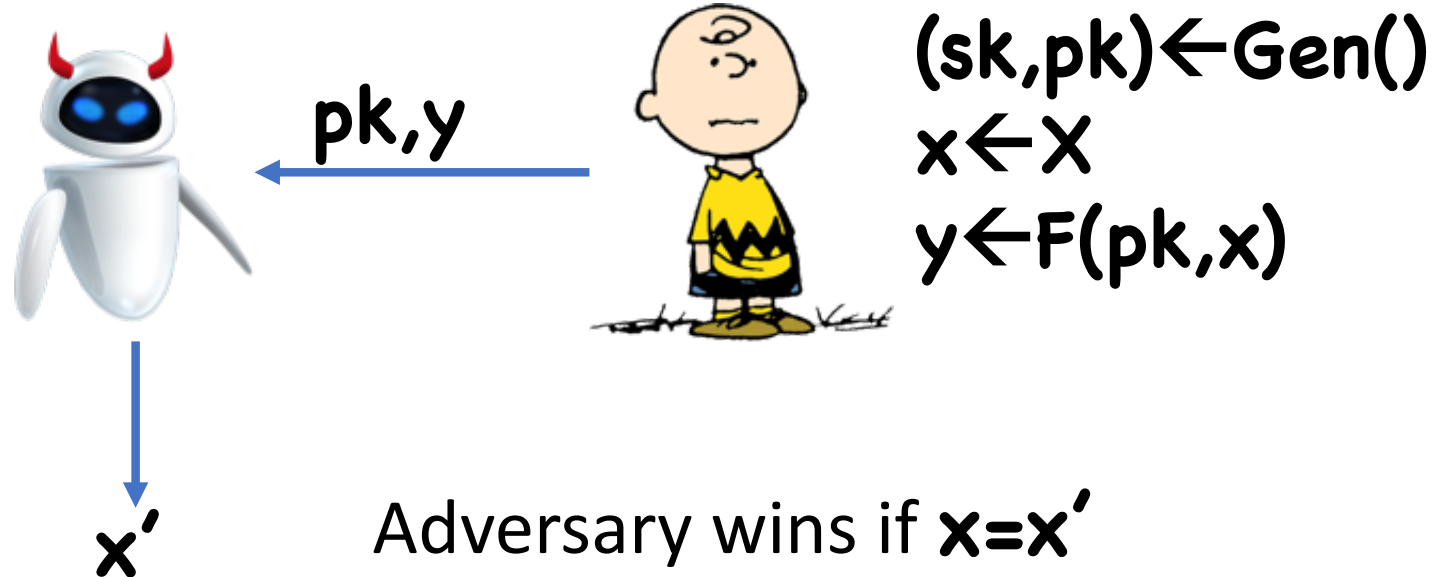
F⁻¹(sk, y) = x

Correctness:

Pr[F⁻¹(sk, F(pk, x)) = x : $(pk, sk) \leftarrow \text{Gen}()$] = 1

Correctness implies **F, F⁻¹** are deterministic,
permutations

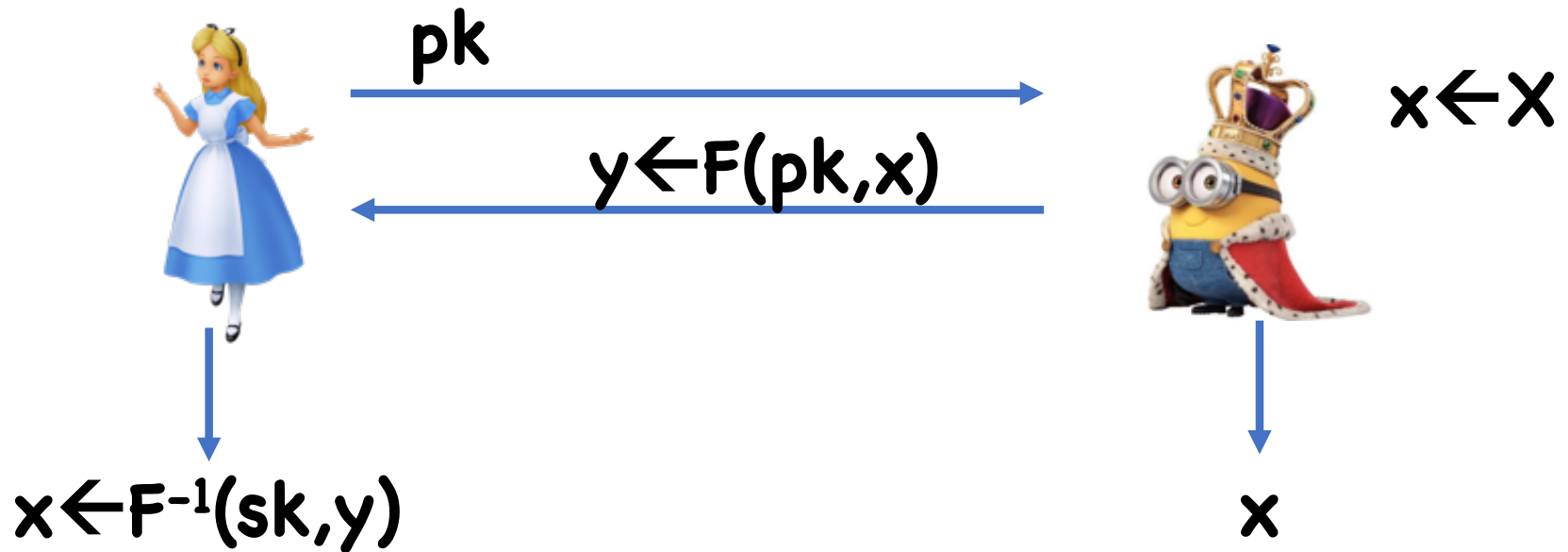
Trapdoor Permutation Security



In other words, $F(pk, \cdot)$ is a one-way function

Key Distribution from TDPs

$(pk, sk) \leftarrow \text{Gen}()$



Analysis


Correctness follows from correctness of TDP

Security:

- By TDP security, adversary cannot compute \mathbf{x}
- However, \mathbf{x} is distinguishable from a random key

Hardcore Bits

Let \mathbf{F} be a one-way function with domain \mathbf{D} , range \mathbf{R}

Definition: A function $\mathbf{h}:\mathbf{D}\rightarrow\{0,1\}$ is a hardcore bit for \mathbf{F} if, for any polynomial time , \exists negligible ϵ such that:

$$| \Pr[1 \leftarrow \text{robot}(F(x), h(x)), x \leftarrow \mathbf{D}]$$

$$- \Pr[1 \leftarrow \text{robot}(F(x), b), x \leftarrow \mathbf{D}, b \leftarrow \{0,1\}] | \leq \epsilon(\lambda)$$

In other words, even given $\mathbf{F}(x)$, hard to guess $\mathbf{h}(x)$

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 $F: \mathbb{Z}_p^* \rightarrow \mathbb{Z}_p^*$ 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 $F: \mathbb{Z}_N^* \rightarrow \mathbb{Z}_N^*$ be $F(x) = x^e \bmod N$ for some e
relatively prime to $(p-1)(q-1)$

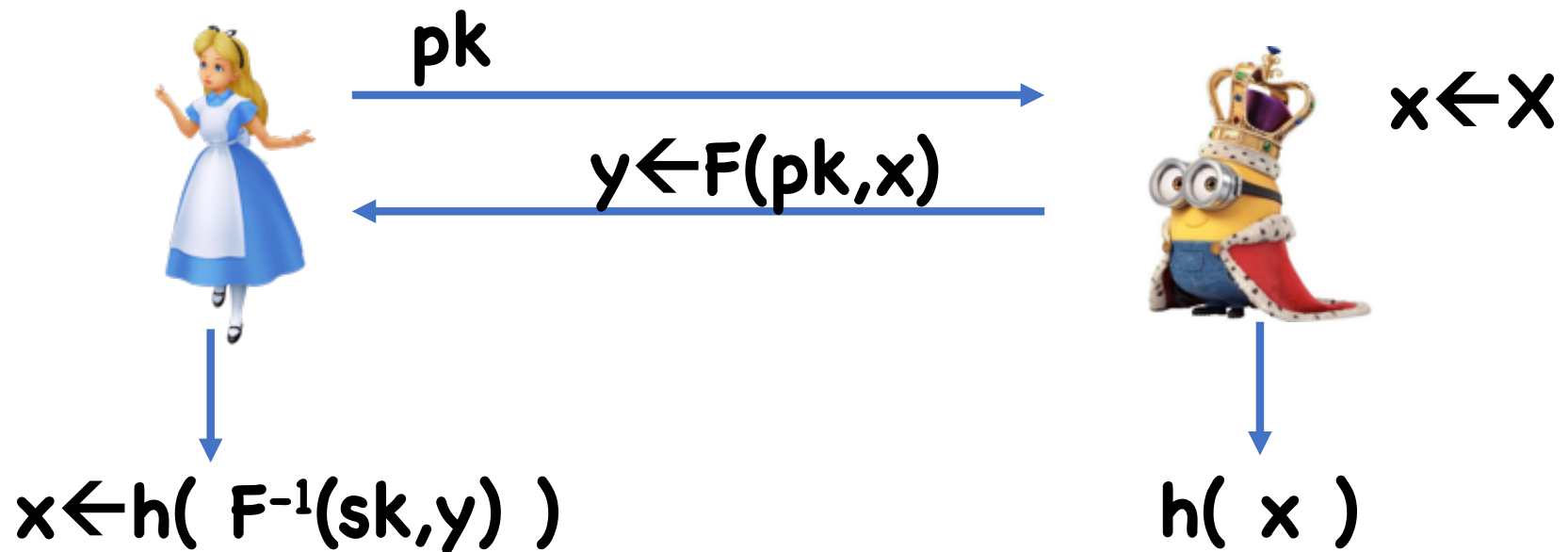
Lsb and Half are hardcore bits for F (assuming RSA)

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

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

Key Distribution from TDPs

$(pk, sk) \leftarrow \text{Gen}()$



h a hardcore bit for $F(pk, \cdot)$

Theorem: If h is a hardcore bit for $F(pk, \cdot)$, 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 $ed=1 \bmod (p-1)(q-1)$
- Output $pk=(N, e), sk=(N, d)$

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

F⁻¹(sk, y): 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

Other TDPs?

For long time, essentially none known

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

Using obfuscation:

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

Key Distribution from DH

Everyone agrees on group \mathbf{G} of prime order \mathbf{p}

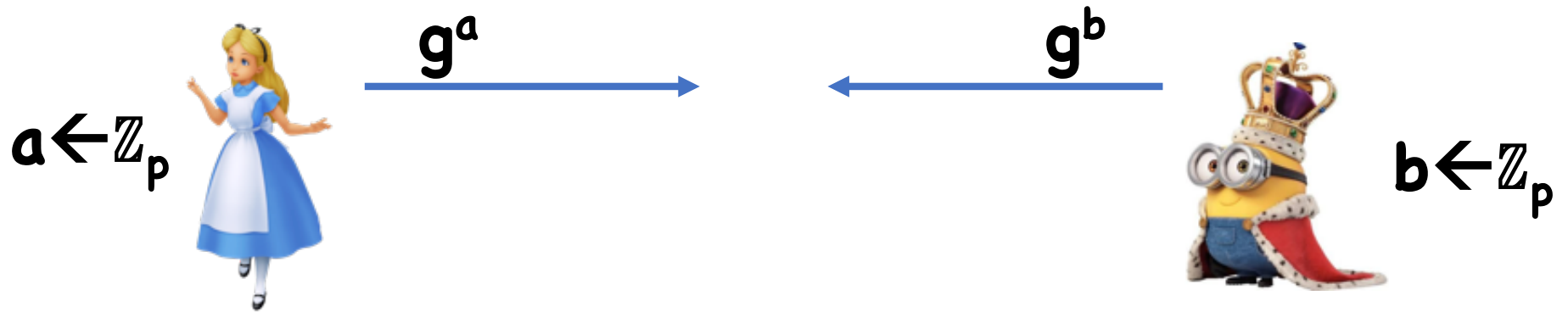
$$a \leftarrow \mathbb{Z}_p$$



$$b \leftarrow \mathbb{Z}_p$$

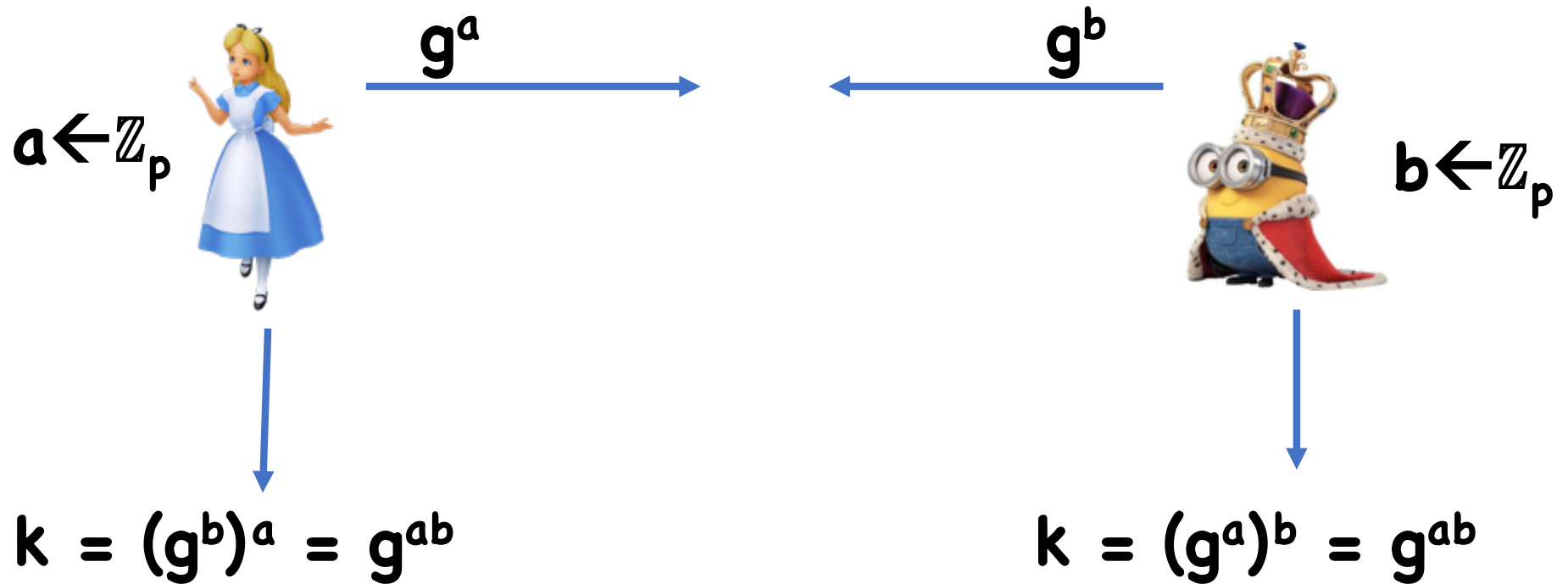
Key Distribution from DH

Everyone agrees on group \mathbf{G} or prime order \mathbf{p}



Key Distribution from DH

Everyone agrees on group \mathbf{G} or prime order \mathbf{p}



Key Distribution from DH

Theorem: If DDH holds on \mathbf{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?