

COS433/Math 473: Cryptography

Mark Zhandry

Princeton University

Spring 2017

Recap

Diffie-Hellman, Factoring, RSA

Discrete Log


Let p be a large integer (maybe prime)

Given $g \in \mathbb{Z}_p^*$, $a \in \mathbb{Z}$, easy to compute $g^a \bmod p$

- Time **$\text{poly}(\log a, \log p)$**

However, no known efficient ways to recover a from g and $g^a \bmod p$

Hard Problems on Groups

Increasing Difficulty 

DLog:

- Given (g, g^a) , compute a

CDH:

- Given (g, g^a, g^b) , compute g^{ab}

DDH:

- Distinguish (g, g^a, g^b, g^c) from (g, g^a, g^b, g^{ab})

Stronger Assumptions 

Integer Factorization

Given an integer **N**, factor **N** into its prime factors

Studied for centuries, presumed computationally difficult

- Grade school algorithm: $O(N^{1/2})$
- Much better algorithms:
 $\exp(C (\log N)^{1/3} (\log \log N)^{2/3})$
- However, all require super-polynomial time

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

Black Box Separations

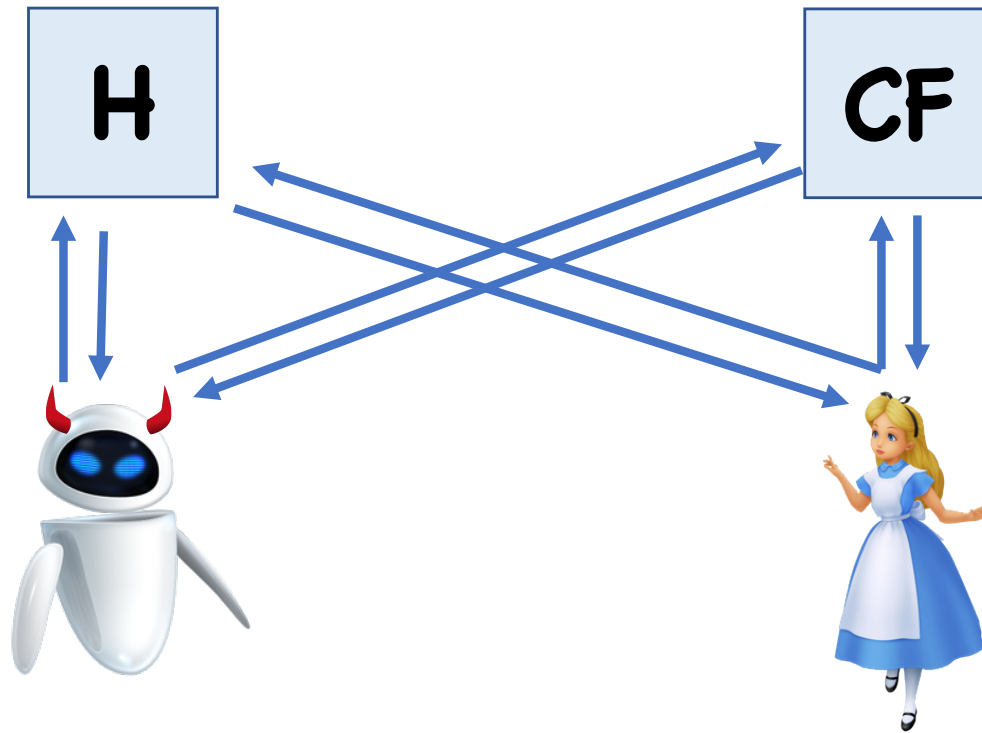
How do we argue that you cannot build collision resistance from one-way functions?

- We generally believe both exist!

Observation: most natural constructions treat underlying objects as black boxes (don't look at code, just input/output)

Maybe we can rule out such natural constructions

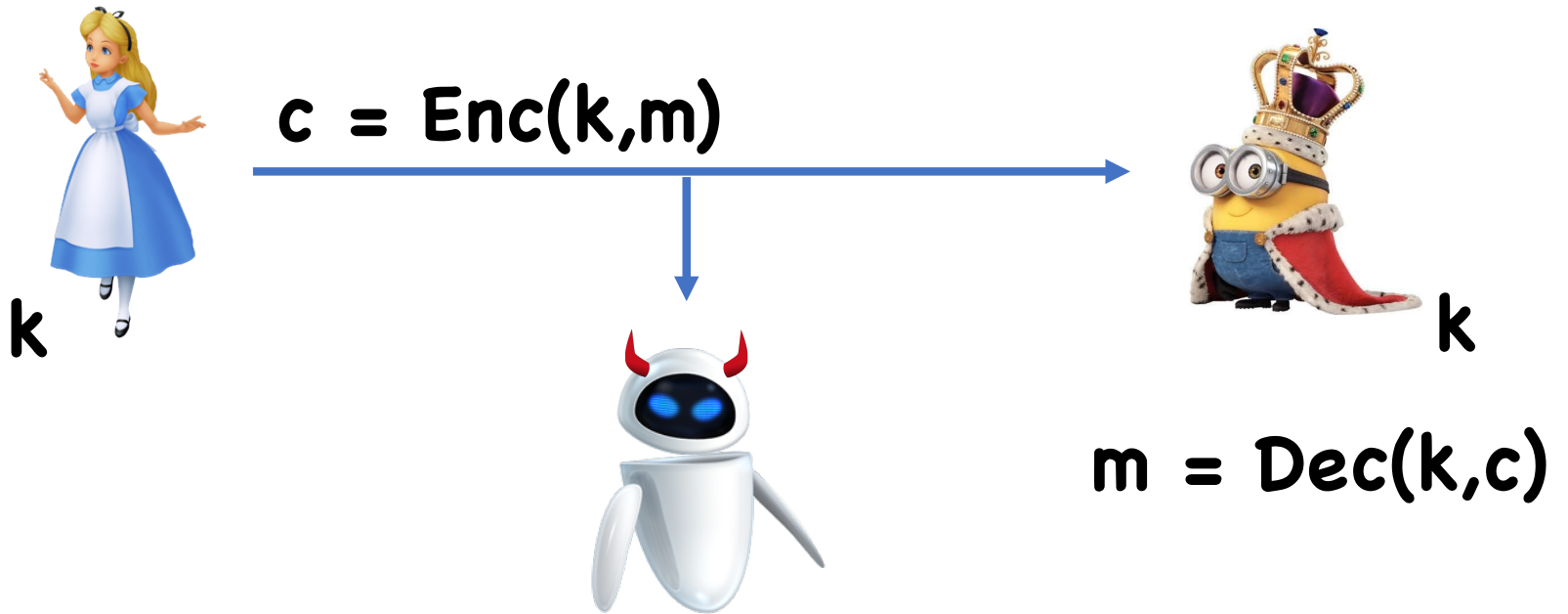
Separating CRH from OWF



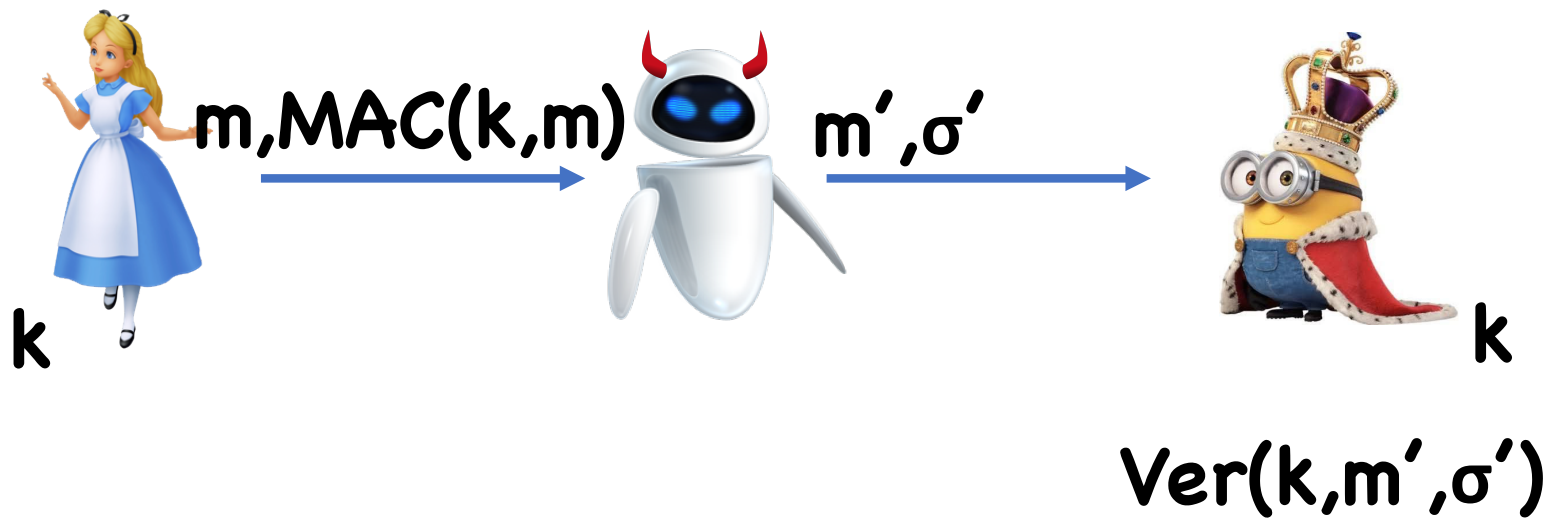
Today

Exchanging keys

Previously



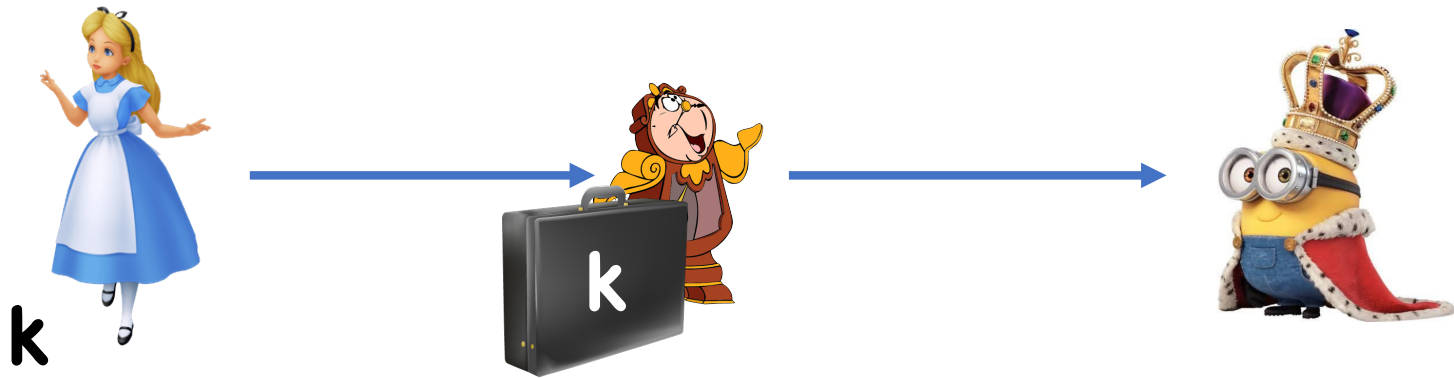
Previously



Today

Where do Alice and Bob get their shared key from?

Traditional Approach



Limitations

Time consuming

Not realistic in many situations

- Do you really want to send a courier to every website you want to communicate with

Doesn't scale well

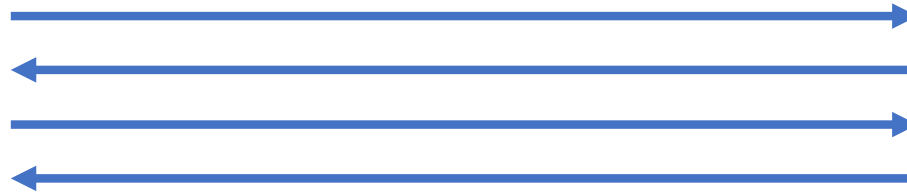
- Imagine 1M people communicating with 1M people

If not meeting in person, need to trust courier

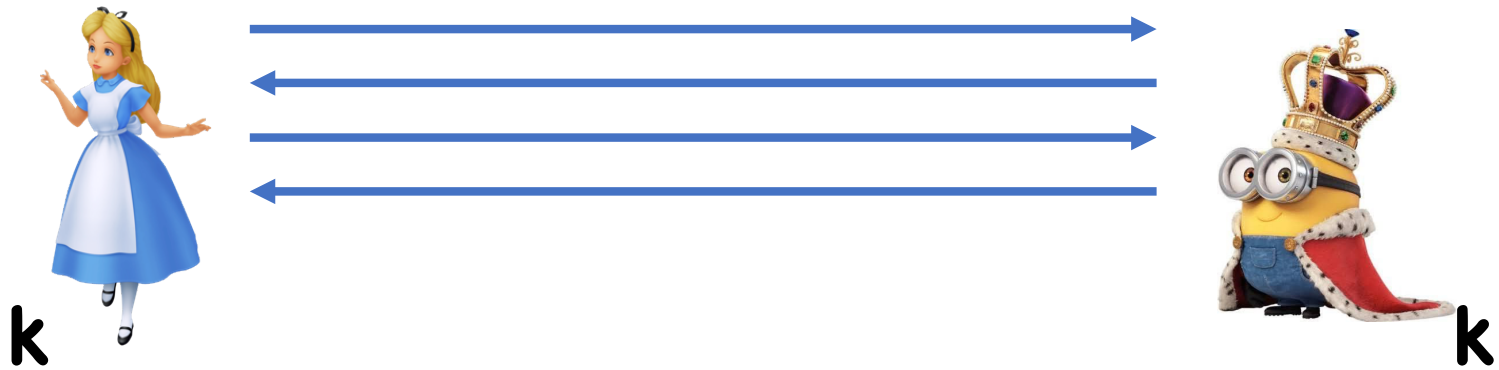
Public Key Distribution



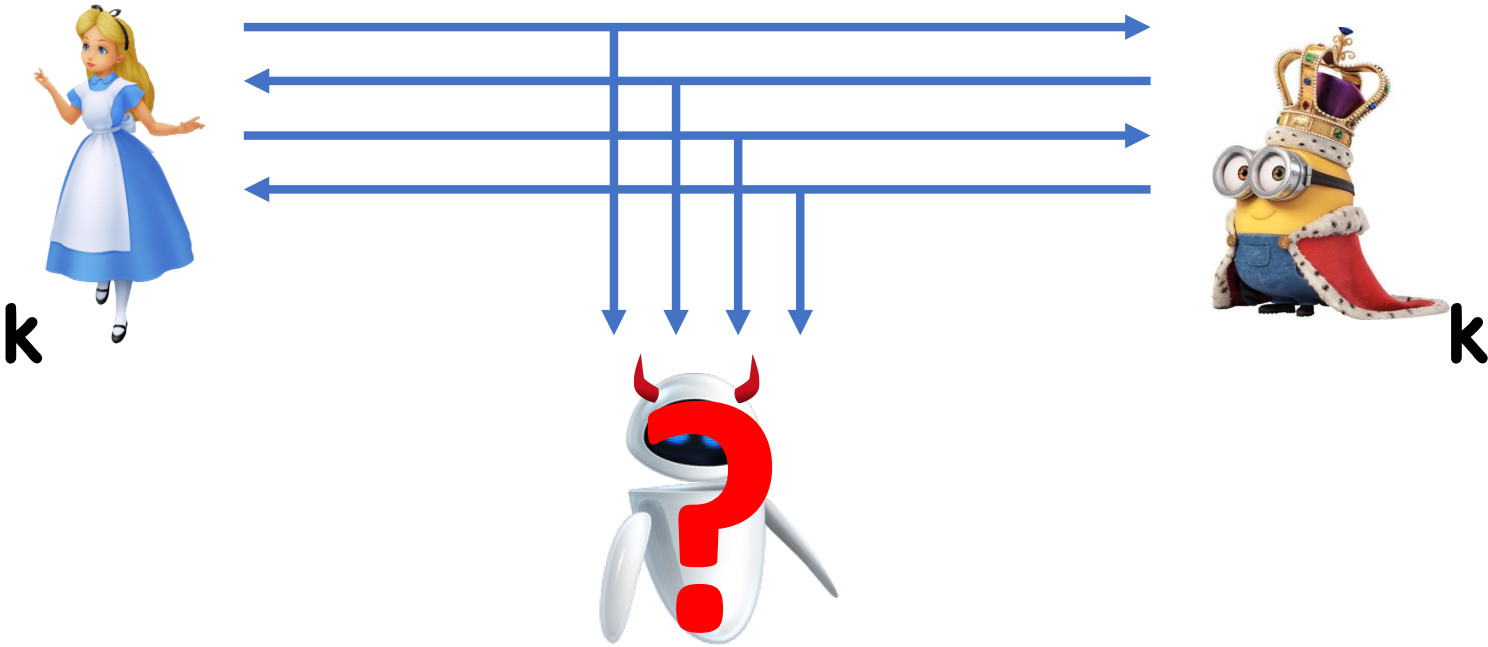
Public Key Distribution



Public Key Distribution

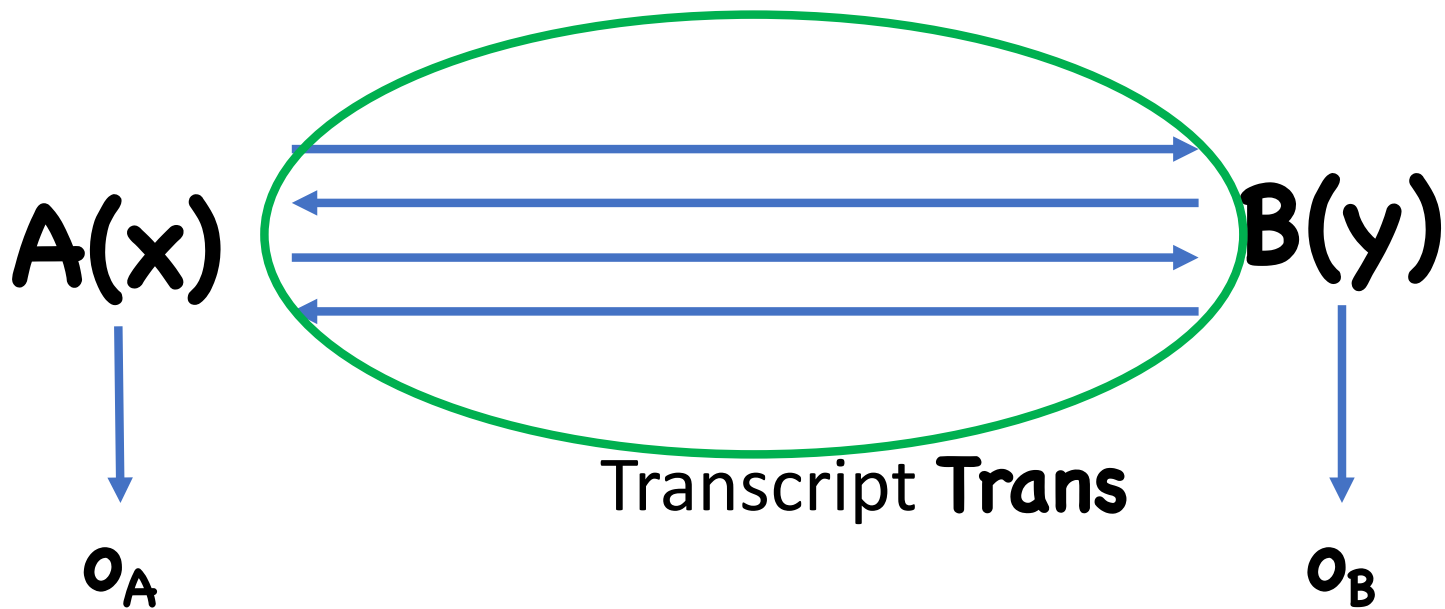


Public Key Distribution



Interactive Protocols

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



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

Public Key Distribution

Pair of interactive algorithms $\mathbf{A}(\lambda), \mathbf{B}(\lambda)$

Correctness:

$$\Pr[\mathbf{o}_A = \mathbf{o}_B : (\mathbf{Trans}, \mathbf{o}_A, \mathbf{o}_B) \leftarrow (\mathbf{A}, \mathbf{B})(\lambda, \lambda)] = 1$$

Shared key is $\mathbf{k} := \mathbf{o}_A = \mathbf{o}_B$

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

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

Matrix Multiplication Approach

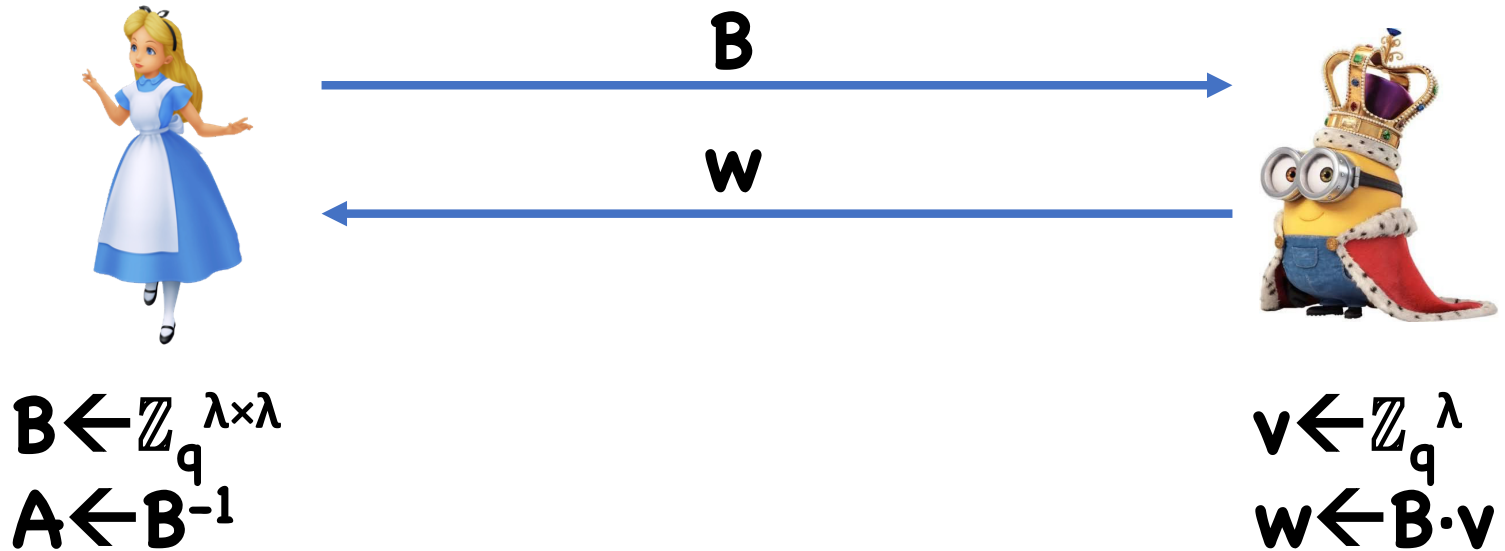


B

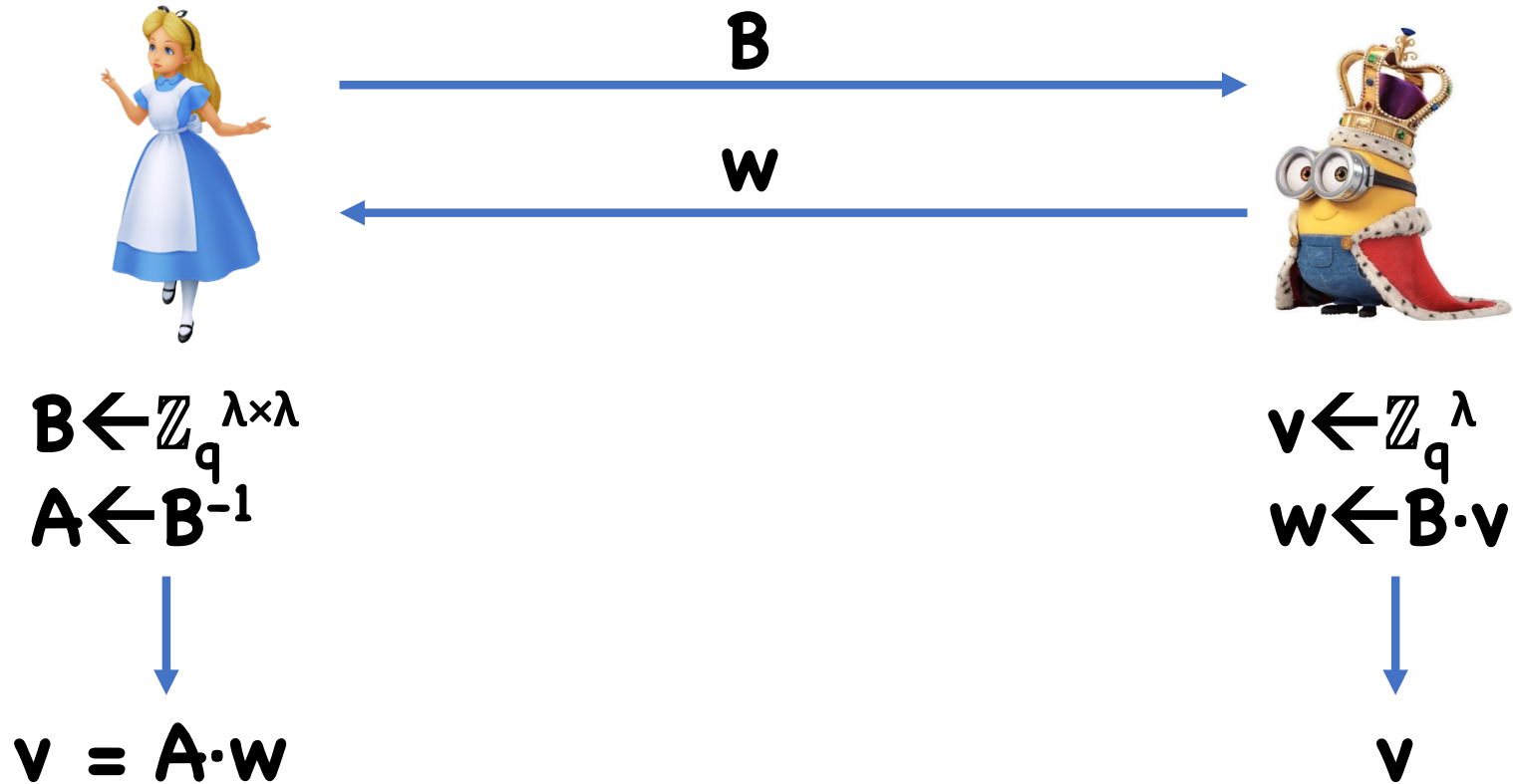


$$\mathbf{B} \leftarrow \mathbb{Z}_q^{\lambda \times \lambda}$$
$$\mathbf{A} \leftarrow \mathbf{B}^{-1}$$

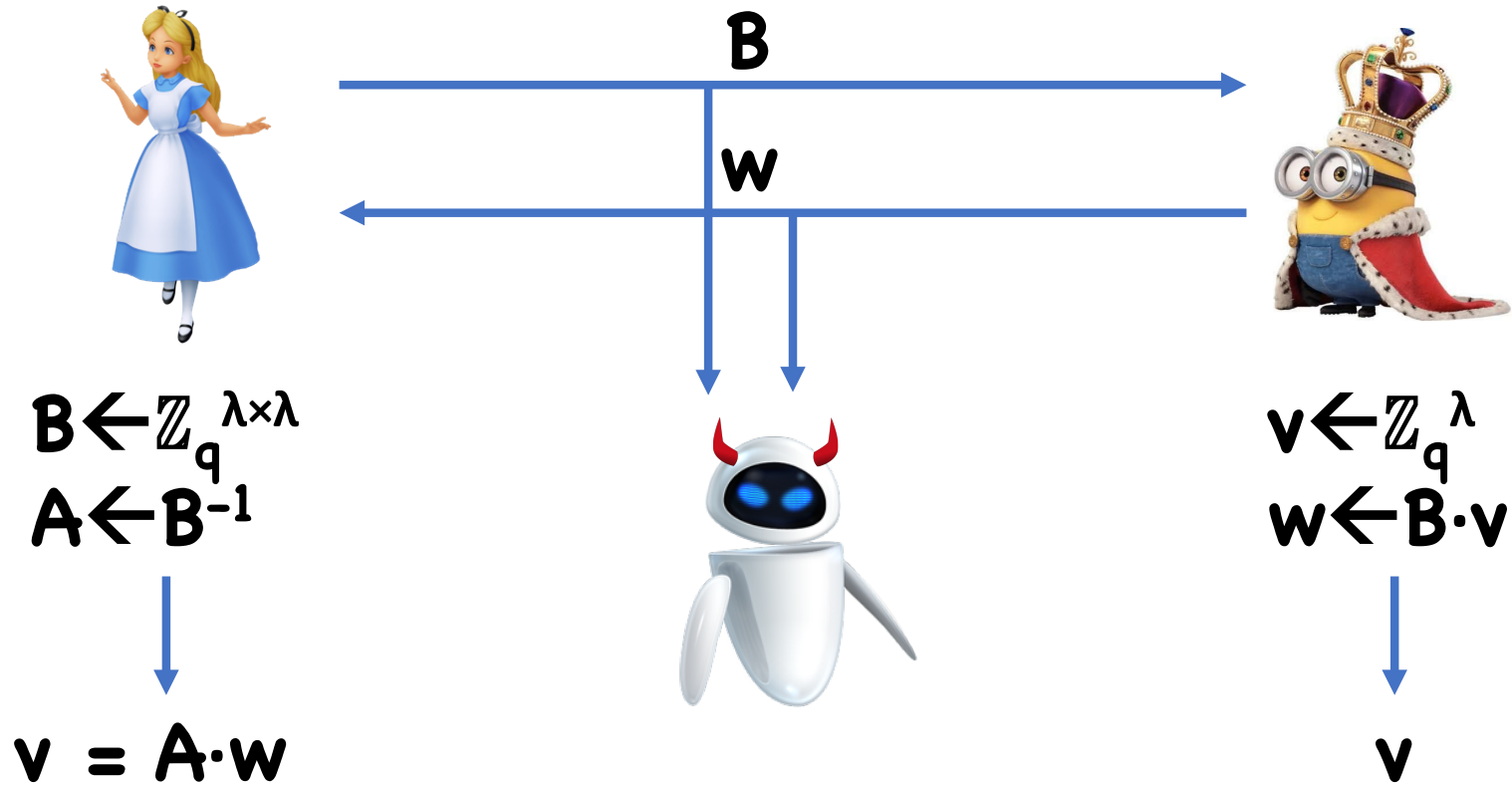
Matrix Multiplication Approach



Matrix Multiplication Approach



Matrix Multiplication Approach



Running Times?

Bob: $O(\lambda^2)$

Eve: $O(\lambda^3)$

Running Times?

Bob: $O(\lambda^2)$

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

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

Different Approach:

- Start with $\mathbf{A} = \mathbf{B} = \mathbf{I}$
- Repeatedly apply random elementary row ops to \mathbf{A} , inverse to \mathbf{B}
- Output (\mathbf{A}, \mathbf{B})

Running Times?

Bob: $O(\lambda^2)$

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

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

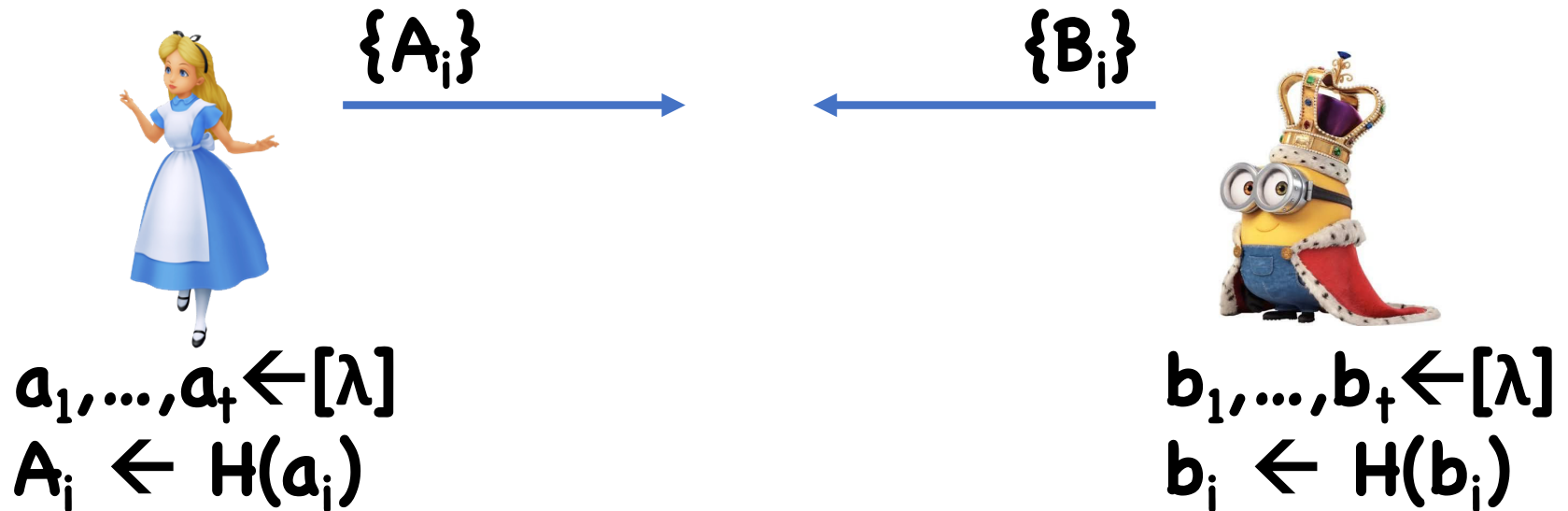
Assuming Matrix Multiplication exponent $\omega > 2$,
adversary must work harder than honest users

inverse to **B**

- Output **(A,B)**

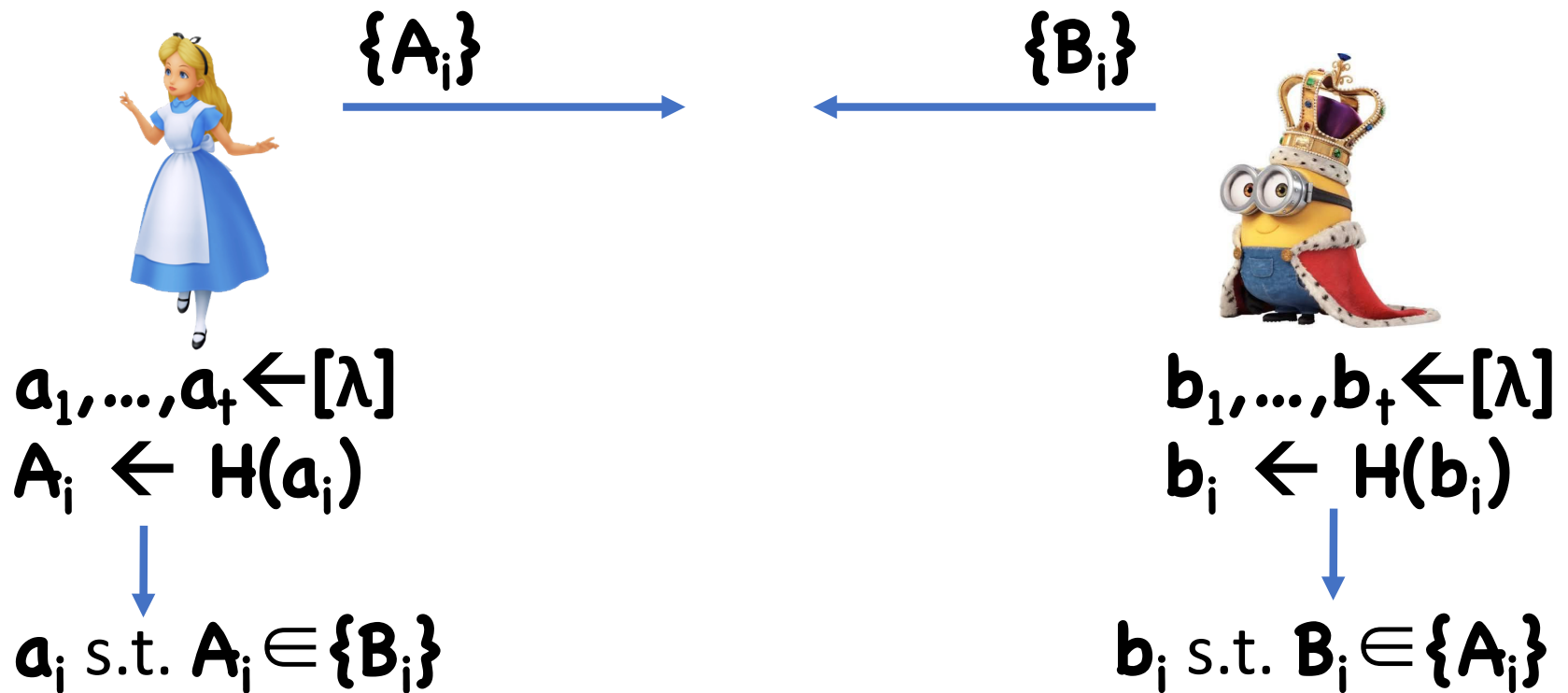
Merkle Puzzles

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



Merkle Puzzles

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



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 $\{A_i\} \cap \{B_i\} \neq \emptyset$

Treating H as ideal hash function (random oracle),
how many queries does adversary need?

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 to rule out all polynomial-time adversaries

- We work very little, but rule out even very hard working adversaries

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)

Key Distribution from RSA

p, q random
primes

$$N = pq$$



N



Key Distribution from RSA

p, q random primes

$N = pq$



$x \leftarrow \mathbb{Z}_N^*$
 $y \leftarrow x^3 \pmod N$



x

Key Distribution from RSA

p, q random primes

$N = pq$



N



$x \leftarrow \mathbb{Z}_N^*$
 $y \leftarrow x^3 \pmod N$



x

$x = y^{1/3} \pmod N$

Analysis

- x uniquely defined as long as $\text{GCD}(3, \Phi(N)) = 1$
- 3 is not a factor of $(p-1)$ or $(q-1)$

How does Alice compute $x = y^{1/3} \bmod N$?

Security:

- Computing cube roots is hard (assuming RSA)
- Adversary cannot compute x
- However, x is distinguishable from a random key

Hardcore Bits

p, q random primes

$$N = pq$$



$$x \leftarrow \mathbb{Z}_N^*$$
$$y \leftarrow x^3 \pmod N$$

$$h(y^{1/3} \pmod N)$$

$$h(x)$$

h a hardcore bit for RSA

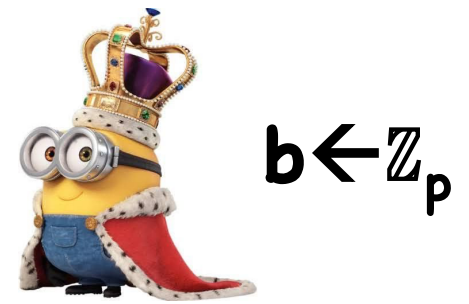
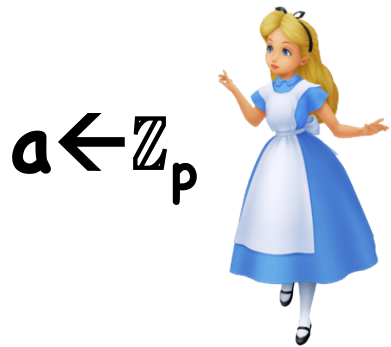
Theorem: If RSA is hard and h is hardcore for RSA, then the protocol is secure

Proof:

- $(\text{Trans}, k) = ((N, x^3), h(x))$
- Hardcore bit means indistinguishable from $((N, x^3), b)$

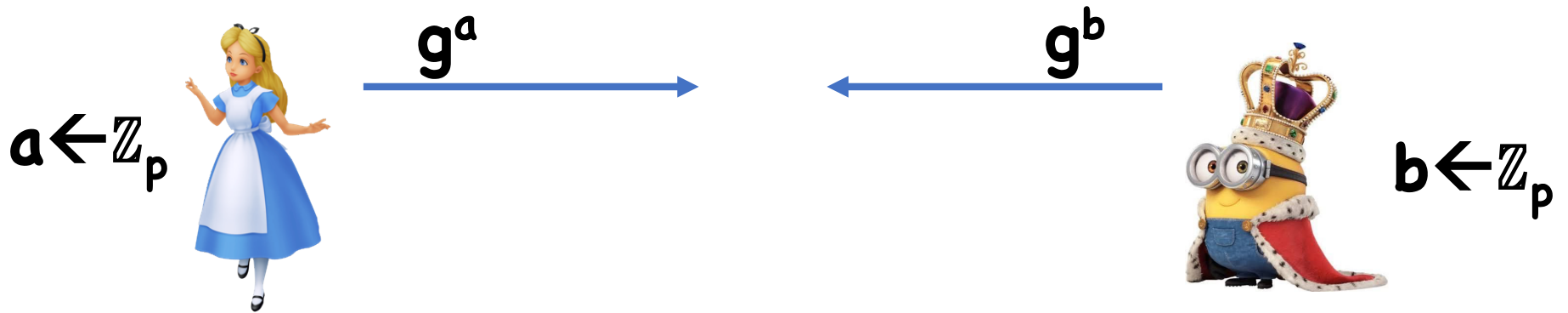
Key Distribution from DH

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



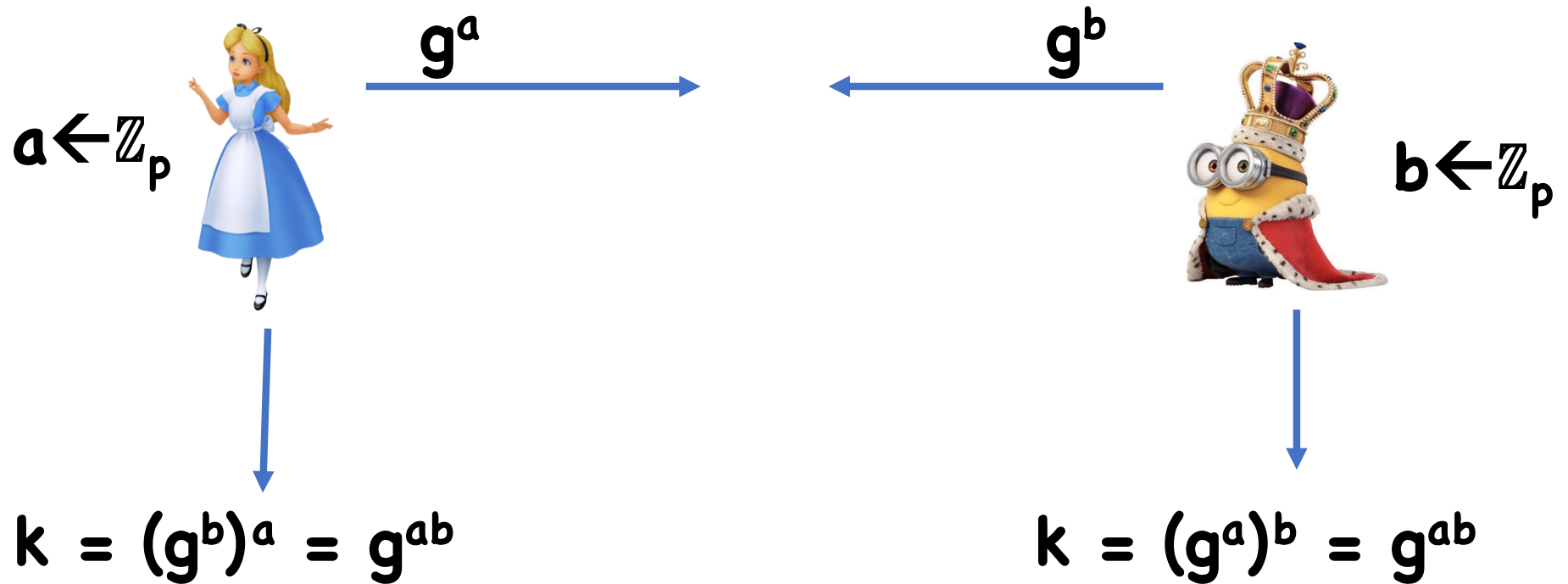
Key Distribution from DH

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



Key Distribution from DH

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:

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

What if only CDH holds, but DDH is easy?

Known Constructions of Public Key Distribution

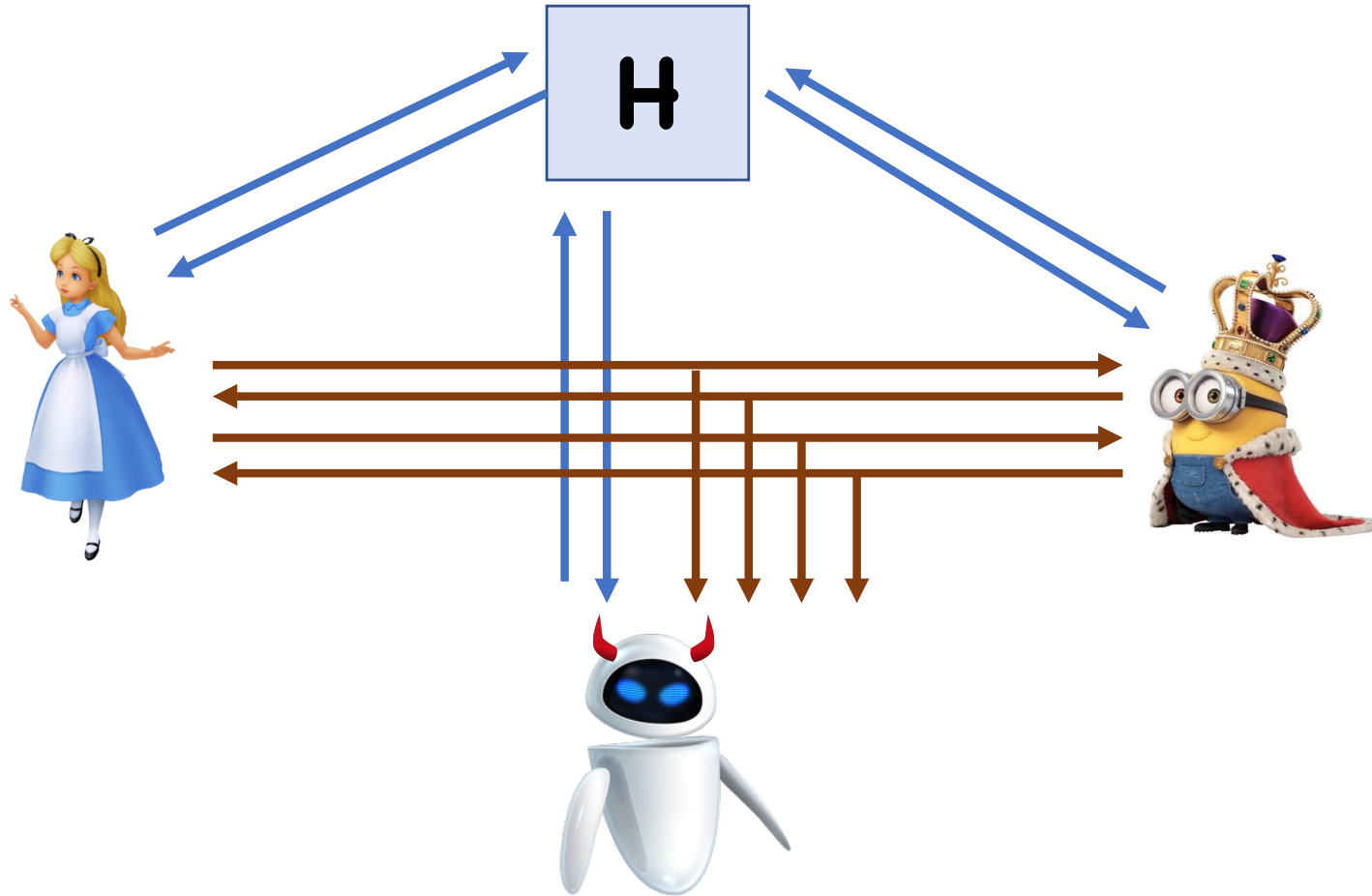
All based on specific number theoretic problems

- RSA, Factoring
- Discrete log, Diffie-Hellman
- ...

No known ways to base (solely) on block ciphers, PRFs, etc.

Is this inherent?

Black Box Separation



Theorem: If H is a random oracle, then for any key agreement protocol in which Alice and Bob make at most n queries, there is an (inefficient) adversary that makes at most $O(n^2)$ queries

Therefore, true public key distribution likely hard to build from one-way functions

If allowing for polynomial hardness gap, then Merkle is likely optimal from one-way functions

History

1974: Merkle invents his puzzles while an undergrad

1976: Diffie and Hellman publish their scheme

- First public mention of public key crypto

1977: RSA publish their scheme

1997: Revealed that public key crypto was developed at GCHQ even earlier

- James H. Ellis: idea for public key crypto
- Clifford Cocks: develops RSA
- Malcolm Williamson: develops Diffie-Hellman

2002: RSA win Turing Award

2015: Diffie-Hellman win Turing Award

Next Time

Public key encryption

- Removes need to key exchange in the first place