

Lecture 5

Introduction to the Public-Key Cryptography

Stefan Dziembowski
University of Rome
La Sapienza



SAPIENZA
UNIVERSITÀ DI ROMA

BiSS 2009
Bertinoro International
Spring School
2-6 March 2009



Plan



1. The problem of key distribution
2. The idea of Merkle, Diffie and Hellman
3. The solution of Rivest, Shamir and Adleman

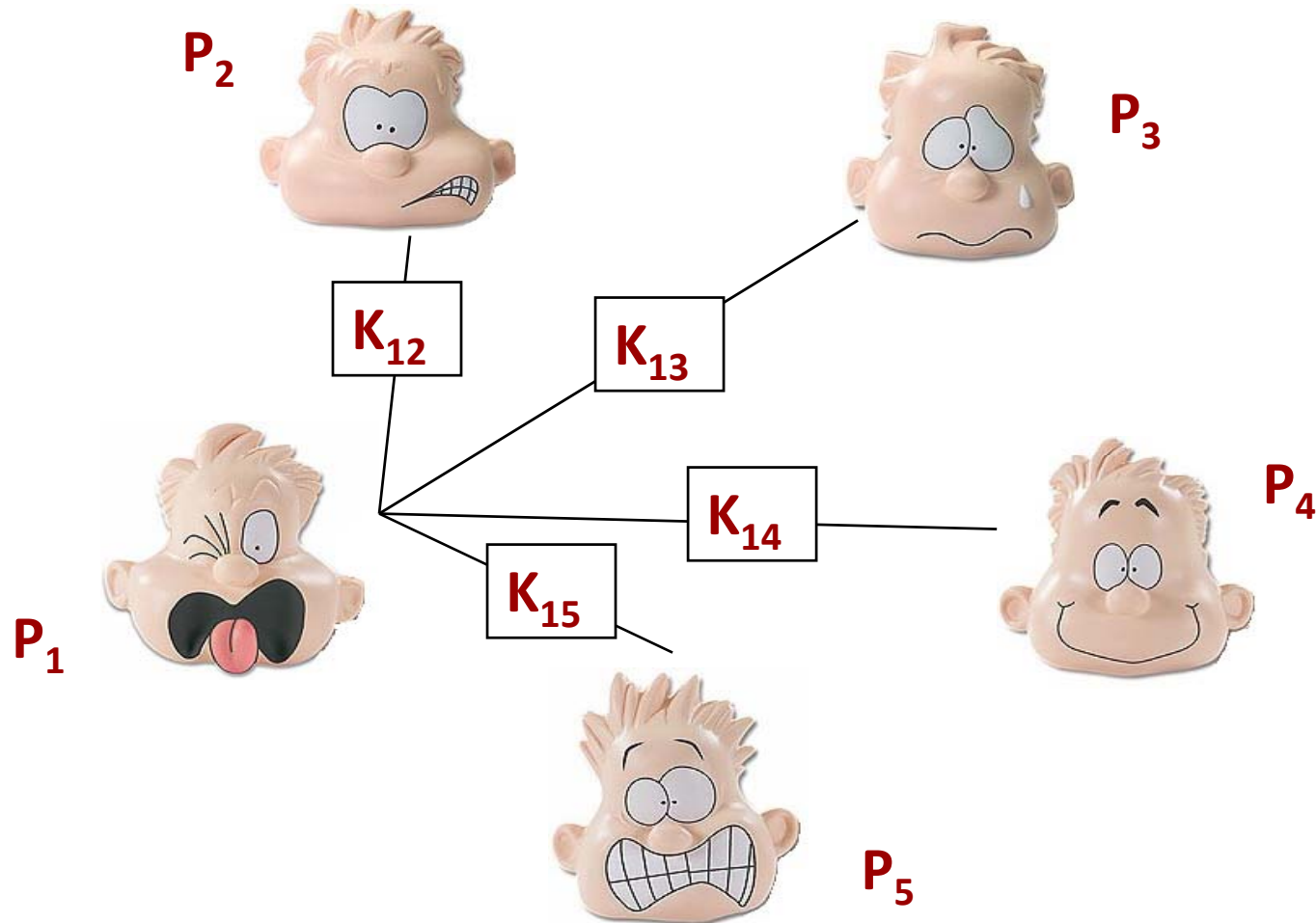
How to distribute the cryptographic keys?

- If the users can meet in person beforehand – **it's simple.**

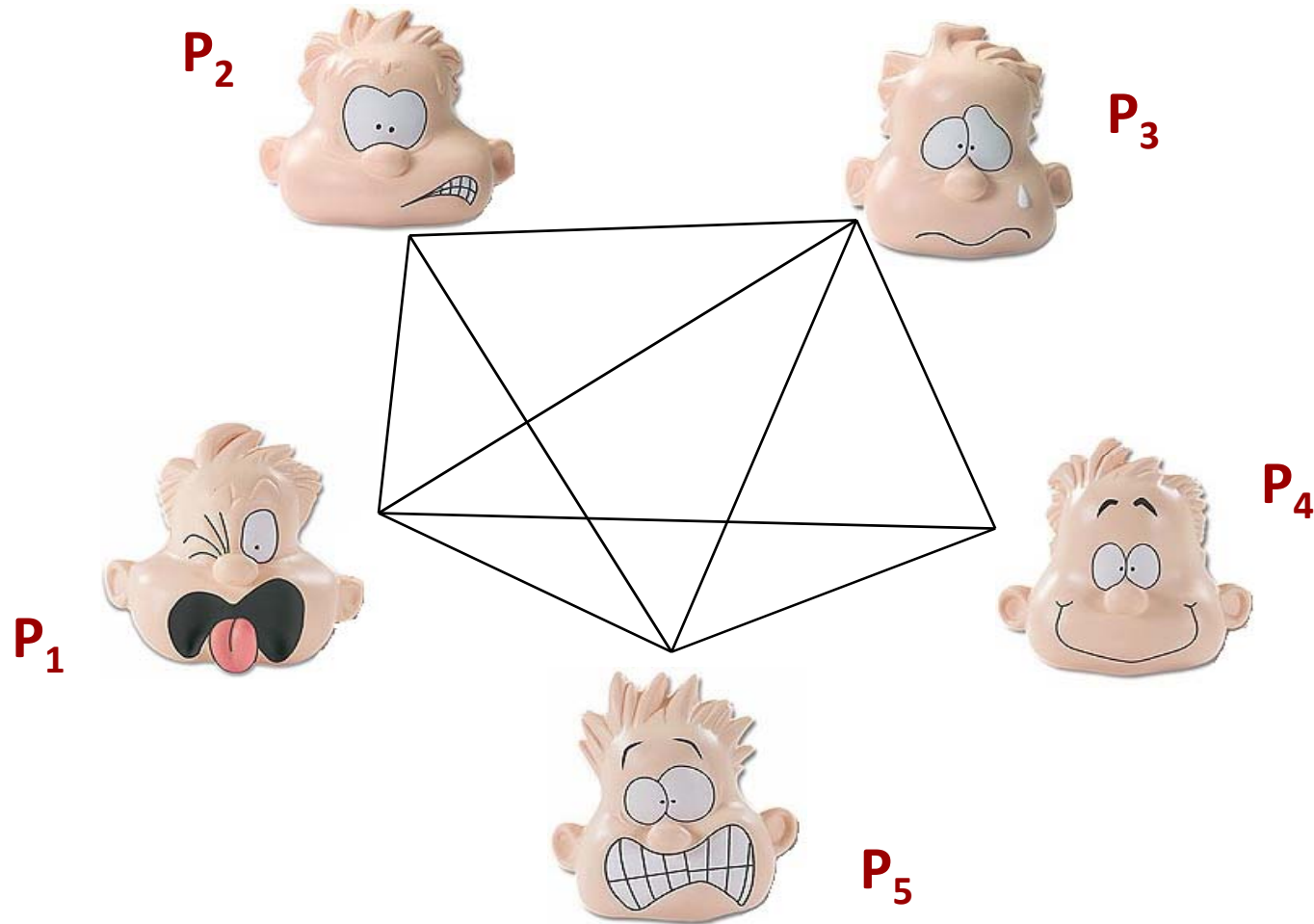
- But what to do if they **cannot meet?**
(a typical example: on-line shopping)

A naive solution:

give to every user P_i a separate key K_{ij} to communicate with every P_j



In general:
a quadratic number of keys is needed

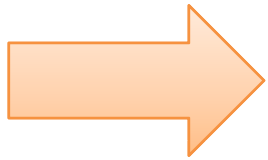


Problems:

- Someone (a **Key Distribution Center, KDC**) needs to “give the keys”
 - **feasible** if the users are e.g. working in one company
 - **infeasible** on the internet
 - relies on the honesty of **KDC**
 - **KDC** needs to be permanently available
 - ...
- The users need to store large numbers of keys **in a secure way**

Plan

1. The problem of key distribution
2. The idea of Merkle, Diffie and Hellman
3. The solution of Rivest, Shamir and Adleman



The solution:

Public-Key Cryptography



Ralph Merkle (1974)

Whitfield Diffie and Martin Hellman (1976)

A little bit of history

- **Diffie and Hellman** were the first to publish a paper containing the idea of the public-key cryptography:

W.Diffie and M.E.Hellman,
New directions in cryptography

IEEE Trans. Inform. Theory, IT-22, 6, **1976**, pp.644-654.

- A similar idea was described by **Ralph Merkle**:
 - in **1974** he described it in a project proposal for a Computer Security course at UC Berkeley (it was rejected)
 - in **1975** he submitted it to the CACM journal (it was rejected)(see <http://www.merkle.com/1974/>)

- It 1997 the GCHQ (the British equivalent of the NSA) revealed that they new it already in **1973**.

The idea

Instead of using one key **K**,

- use **2** keys (**pk,sk**), where
 - **pk** is used for **encryption**,
 - **sk** is used for **decryption**,
or
 - **sk** is used for **computing a tag**,
 - **pk** is used for **verifying correctness of the tag**.

this will be called
“signatures”

Sign – the signing
algorithm

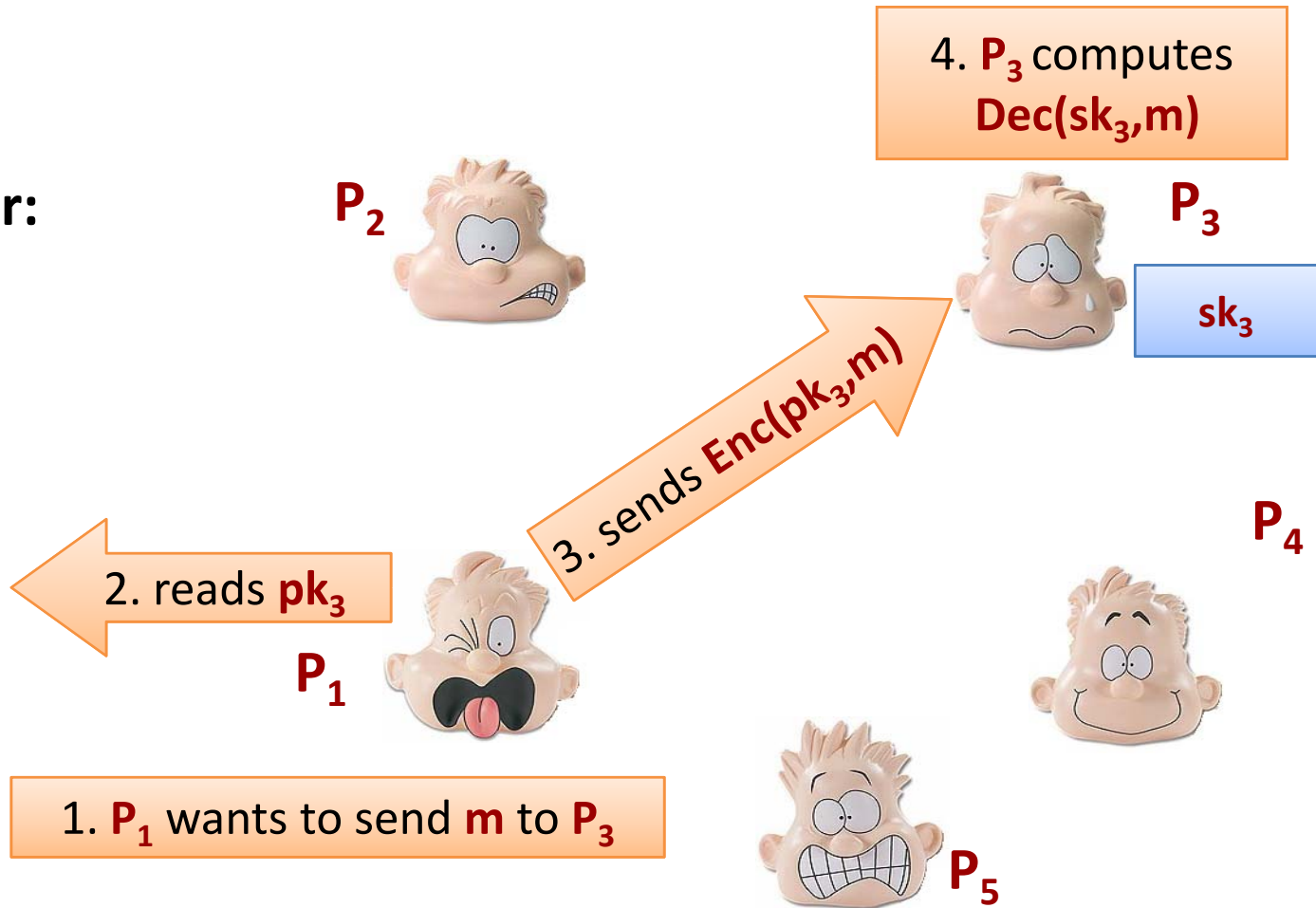
Moreover: **pk** can be public, and only **sk** has to be kept secret!

That's why it's called: **public-key cryptography**

Anyone can send encrypted messages to anyone else

public register:

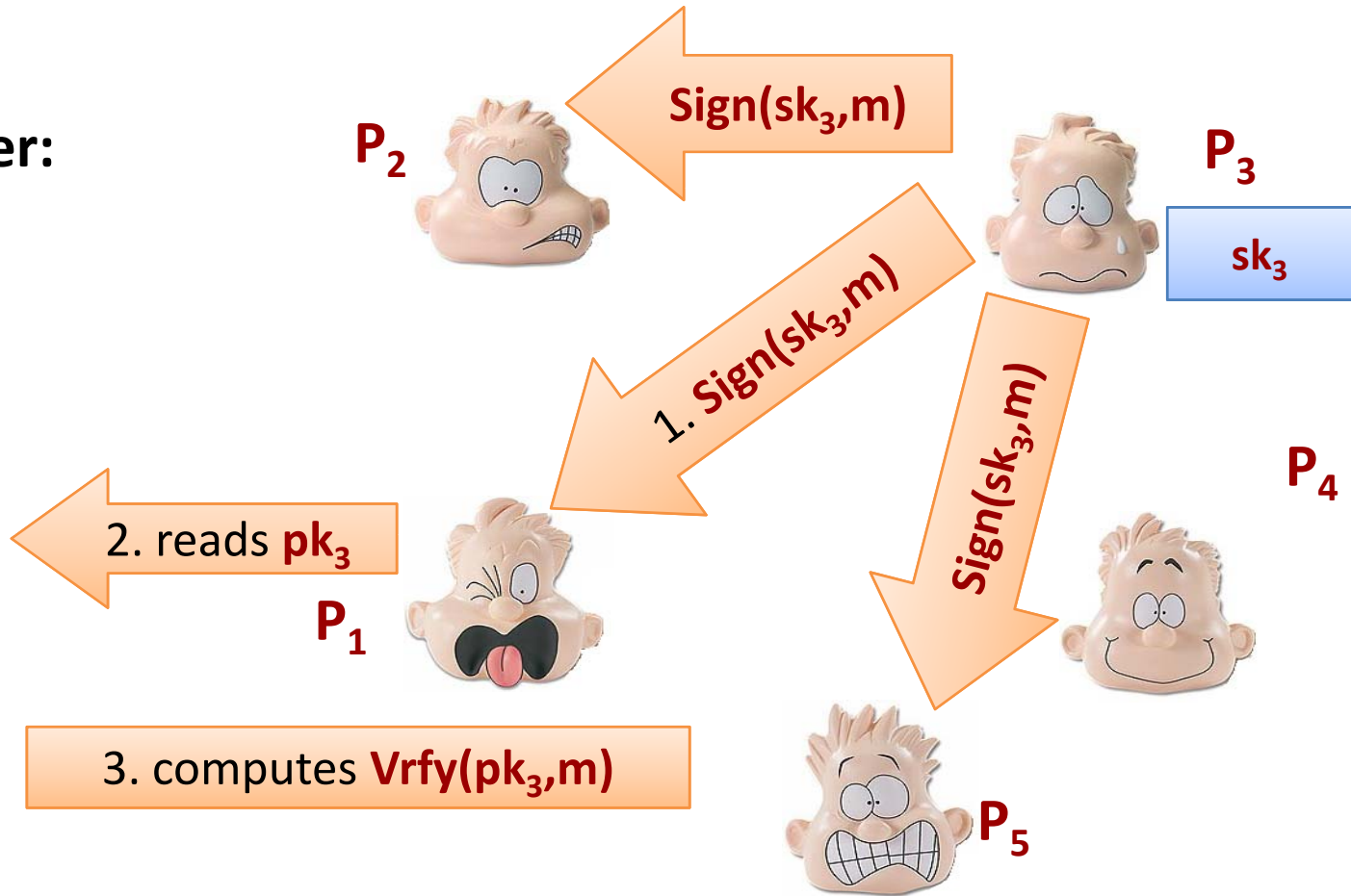
pk_1
pk_2
pk_3
pk_4
pk_5



Anyone can verify the signatures

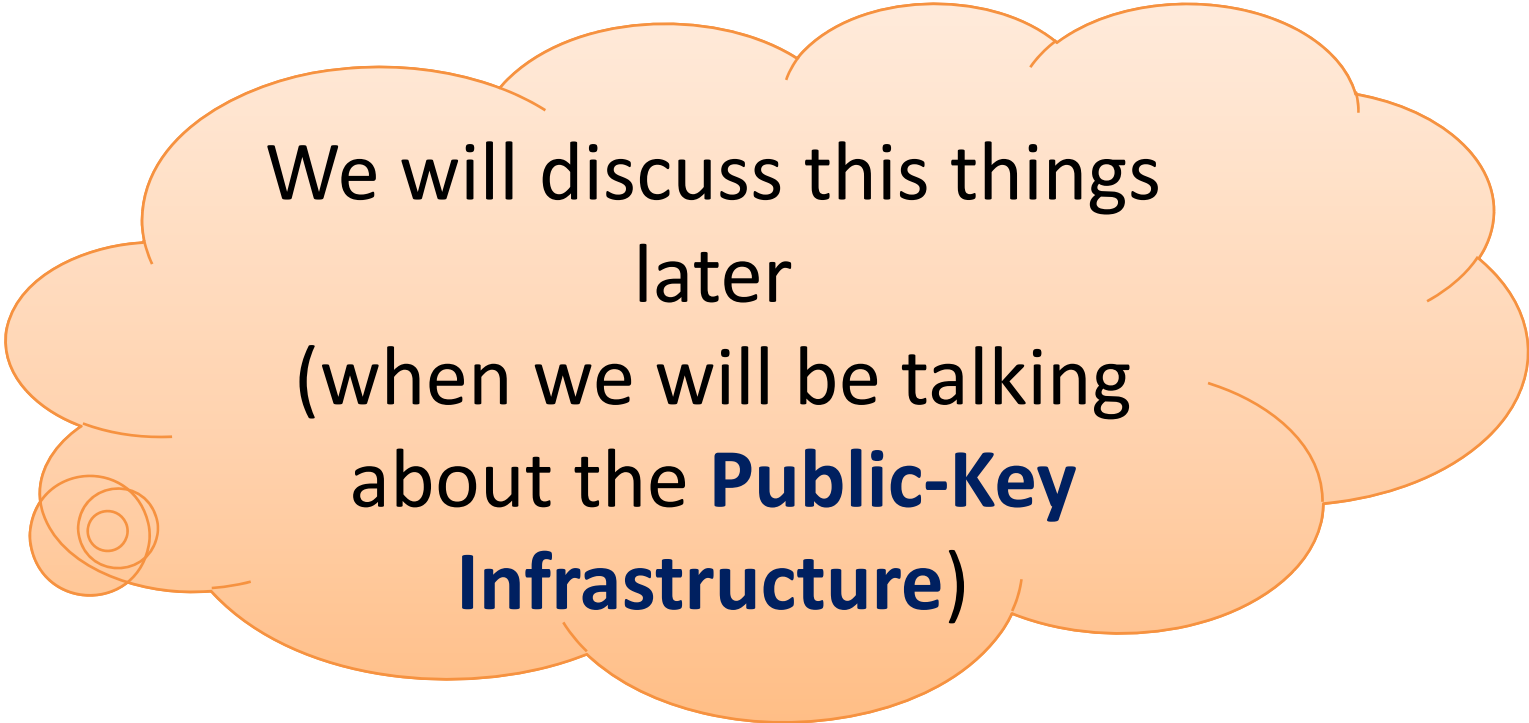
public register:

pk_1
pk_2
pk_3
pk_4
pk_5



Things that need to be discussed

- Who maintains “the register”?
- How to contact it securely?
- How to revoke the key (if it is lost)?
- ...



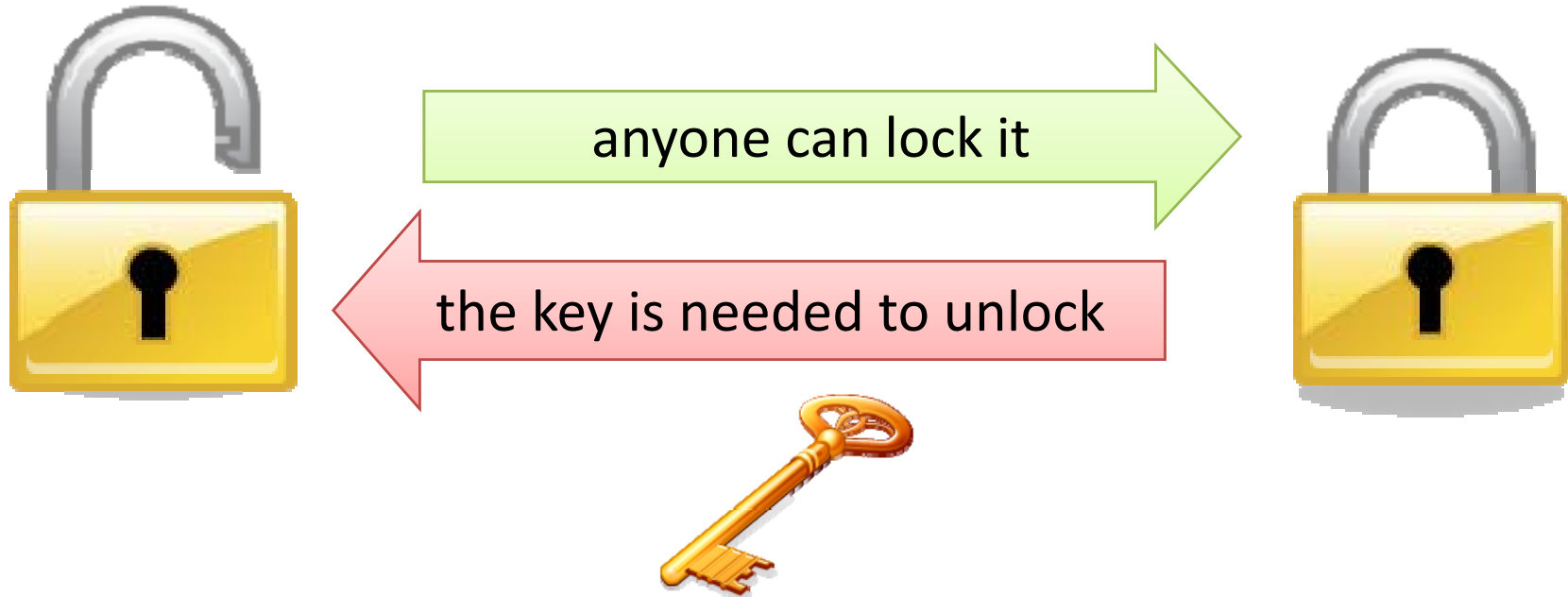
We will discuss these things
later
(when we will be talking
about the **Public-Key
Infrastructure**)

But is it possible?

In “physical world”: yes!

Examples:

1. “normal” signatures
2. padlocks:



Diffie and Hellman (1976)

- Diffie and Hellman proposed the public key cryptography in **1976**.

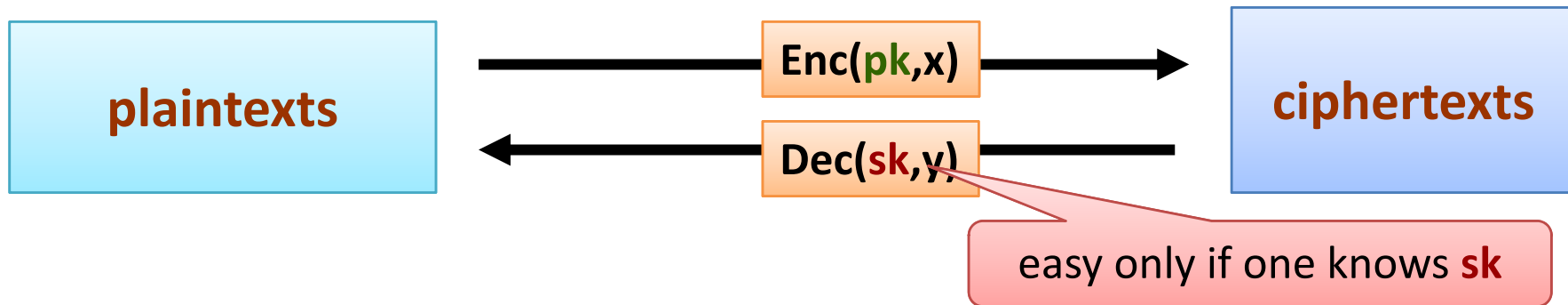
- They just proposed the **concept**, not the **implementation**.

- They have also shown a protocol for **key-exchange**.

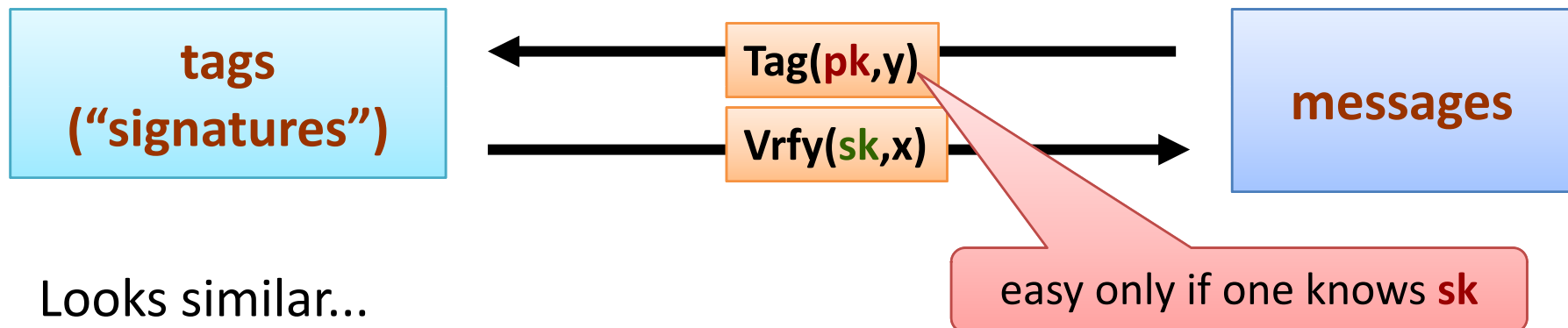
The observation of Diffie and Hellman:

(pk, sk) – the key pair

public-key encryption:



signature schemes:



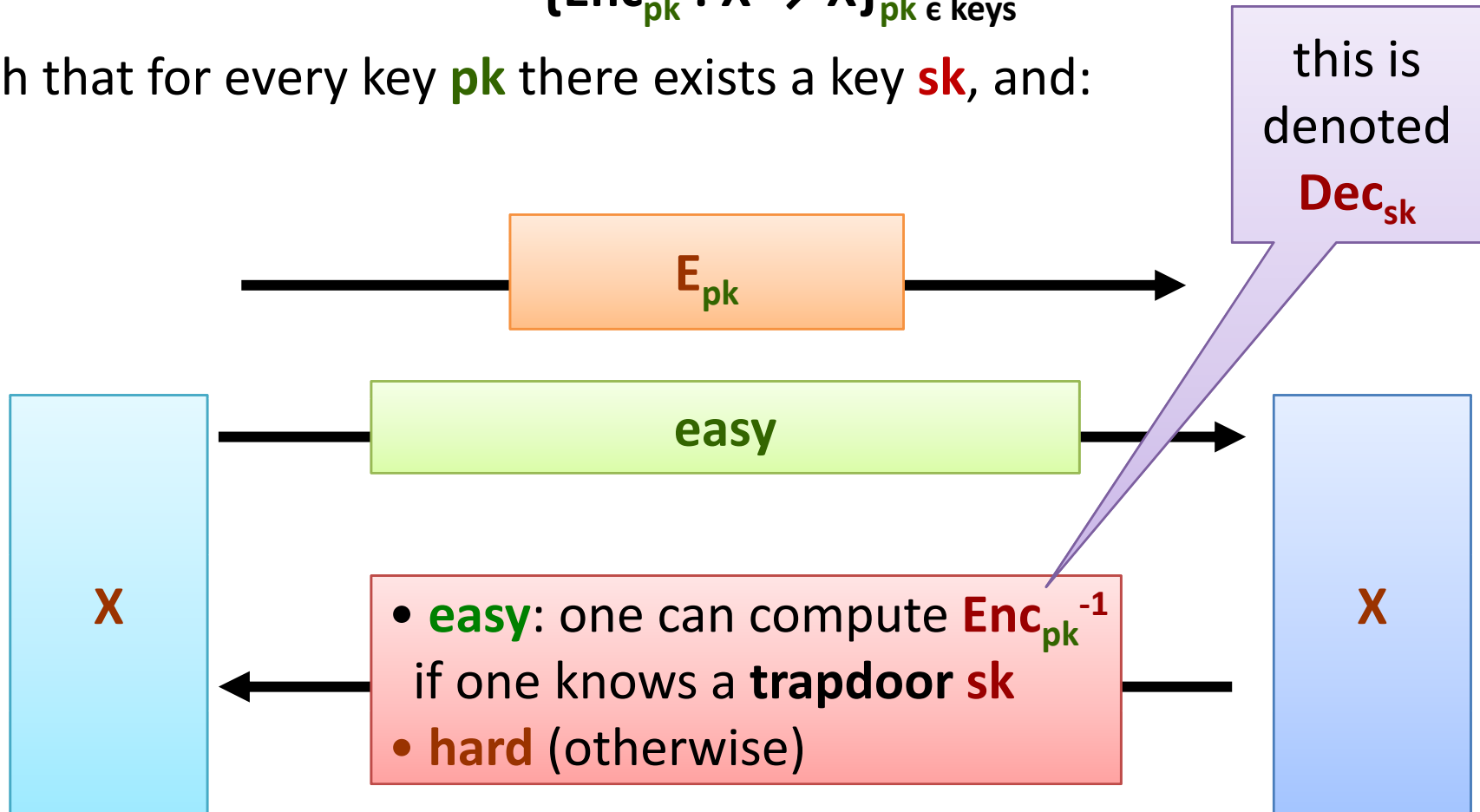
Looks similar...

Trapdoor permutations (informal definition)

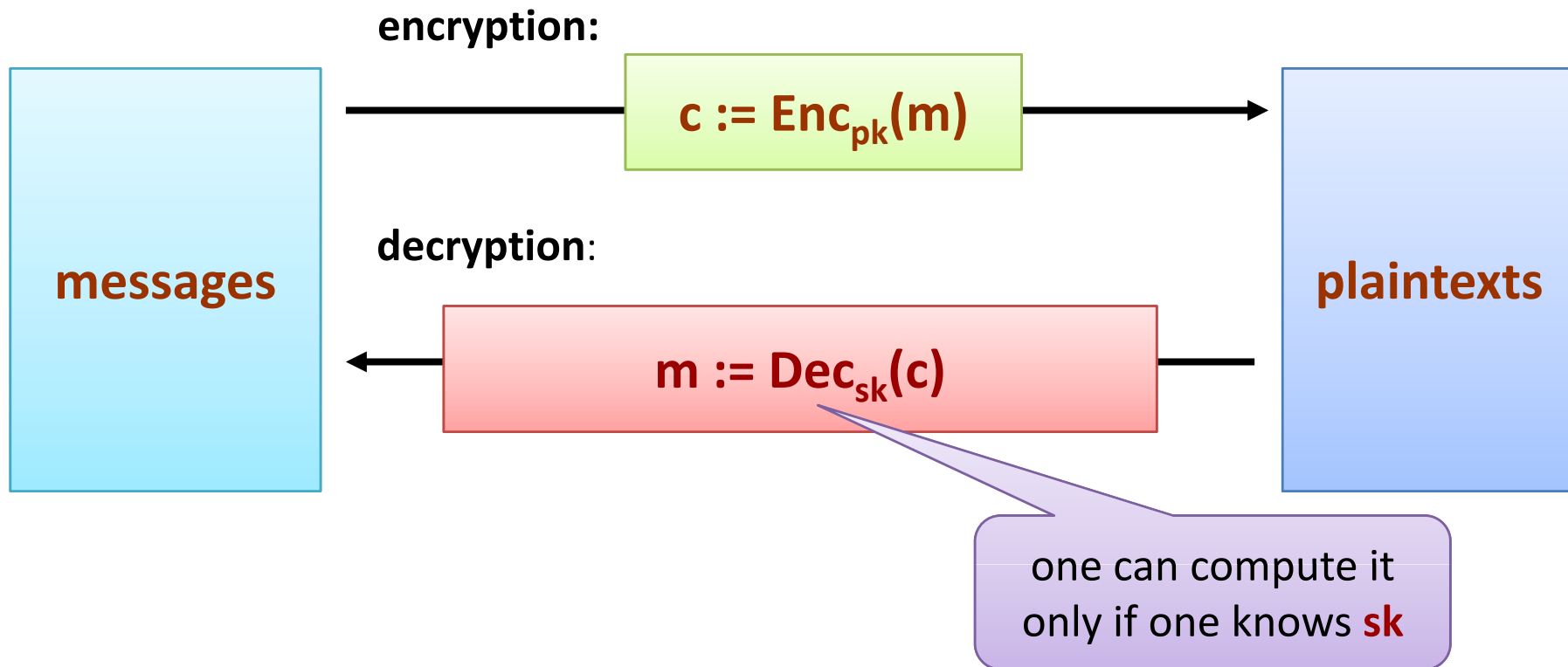
A family of permutations indexed by $pk \in keys$:

$$\{Enc_{pk} : X \rightarrow X\}_{pk \in keys}$$

such that for every key pk there exists a key sk , and:

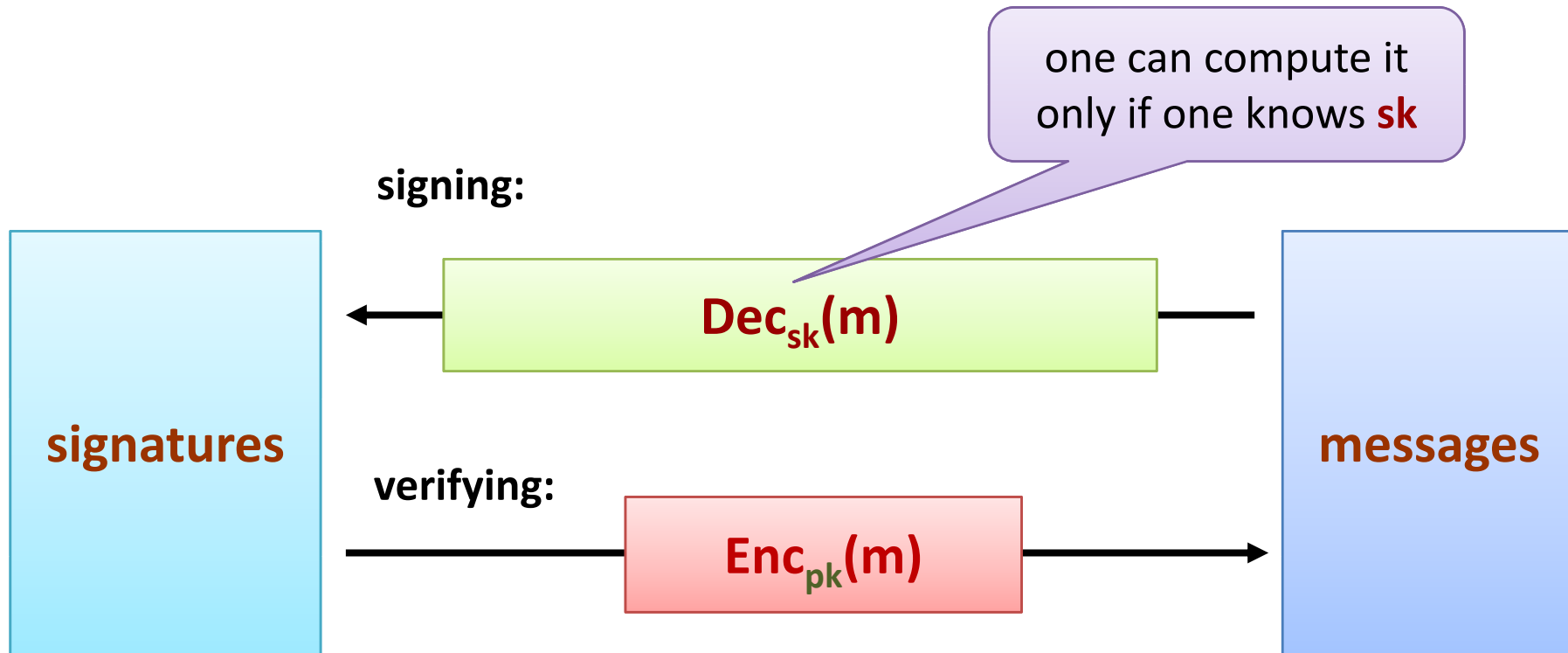


How to encrypt a message **m**



Warning: In general it's not that simple. We will explain it later.

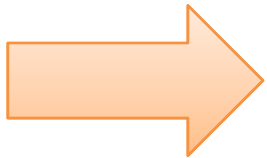
How to sign a message **m**



Warning: In general it's not that simple. We will explain it later.

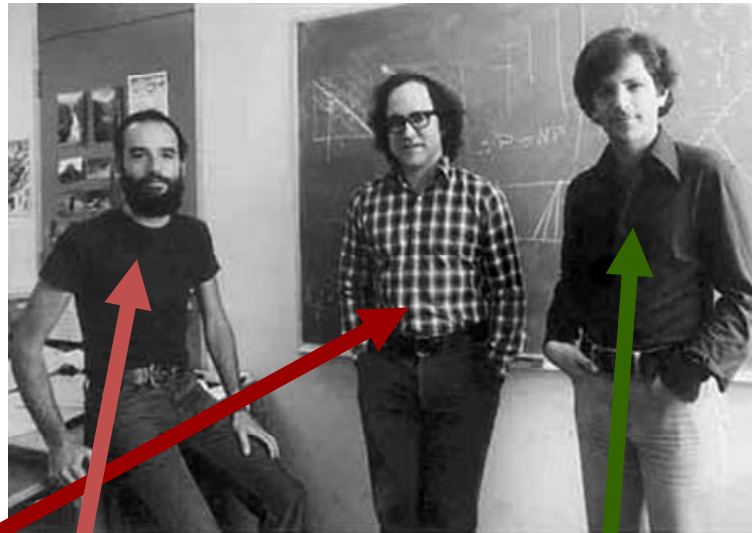
Plan

1. The problem of key distribution
2. The idea of Merkle, Diffie and Hellman
3. The solution of Rivest, Shamir and Adleman



Do such functions exist?

Yes: exponentiation modulo N , where N is a product of two large primes.



Ron Rivest, Adi Shamir, and Leonard Adleman (1977)

RSA function is (conjectured to be) a trapdoor permutation!

The RSA function

$N = pq$, such that p and q are primes,
and $|p| = |q|$

$$\phi(N) = (p-1)(q-1).$$

e is such that $\gcd(e, \phi(N)) = 1$

d is such that $ed = 1 \pmod{\phi(N)}$

$$pk := (N, e)$$

$$sk := (N, d)$$

$Enc_{pk}: \mathbb{Z}_N^* \rightarrow \mathbb{Z}_N^*$ is defined as:

$$Enc_{pk}(m) = m^e \pmod{N}.$$

$Dec_{sk}: \mathbb{Z}_N^* \rightarrow \mathbb{Z}_N^*$ is defined as:

$$Dec_{sk}(c) = c^d \pmod{N}.$$

An observation

From the previous lecture we know that

- $\text{Enc}_{pk}: \mathbb{Z}_N^* \rightarrow \mathbb{Z}_N^*$ is a permutation and
- $\text{Dec}_{sk}: \mathbb{Z}_N^* \rightarrow \mathbb{Z}_N^*$ is its inverse.

Fact

Enc_{pk} is also a permutation over \mathbb{Z}_N and Dec_{sk} is its inverse.

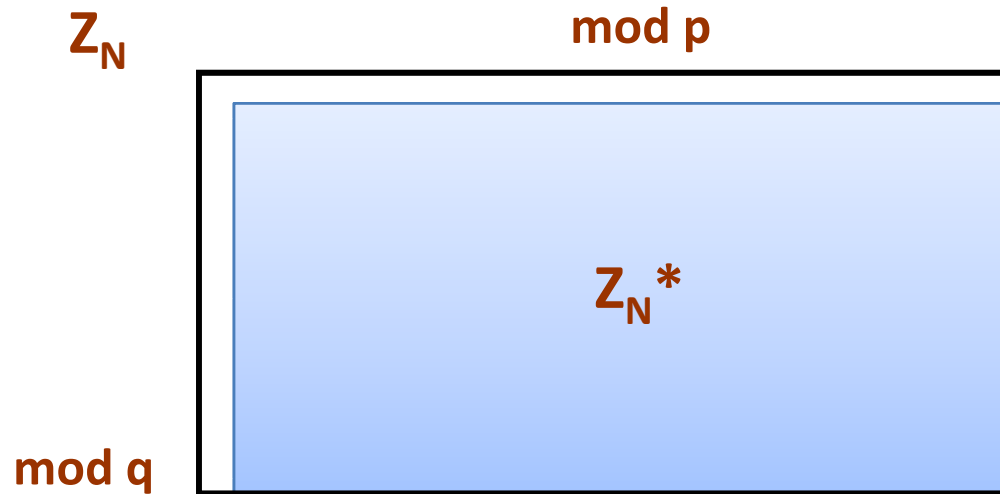
In fact, this doesn't even matter that much because:

if one finds an element $a \in \mathbb{Z}_N \setminus \mathbb{Z}_N^*$ then one can factor N ,
because:

$$\gcd(a, N) > 1.$$

So, finding such an element is as hard as factoring N .

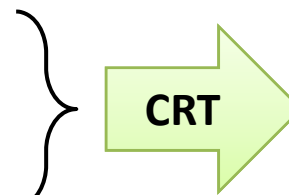
A proof of the fact from the previous slide



Suppose $x = 0 \text{ mod } p$.

Then (trivially) $(x^e)^d = x \text{ mod } p$

On the other hand: $(x^e)^d = x^{ed} = x^1 \text{ mod } q$



$(x^e)^d = x \text{ mod } N$

because: $ed = 1 \text{ mod } (p-1)(q-1)$,
and therefore $ed = 1 \text{ mod } (q-1)$

QED

Is **RSA** secure?

Is **RSA** secure:

1. as an encryption scheme?
2. as a signature scheme?

The answer is not that simple.

First, we need to define security!

We will do it on the next two lectures.

©2009 by Stefan Dziembowski. Permission to make digital or hard copies of part or all of this material is currently granted without fee *provided that copies are made only for personal or classroom use, are not distributed for profit or commercial advantage, and that new copies bear this notice and the full citation.*