ENEE 457: Computer Systems Security
09/28/16

Lecture 8
Introduction to Public Key Systems

Charalampos (Babis) Papamanthou

Department of Electrical and Computer Engineering
University of Maryland, College Park

- Slides adjusted from:
  - http://dziembowski.net/Teaching/BISS09/

How to distribute the cryptographic keys?

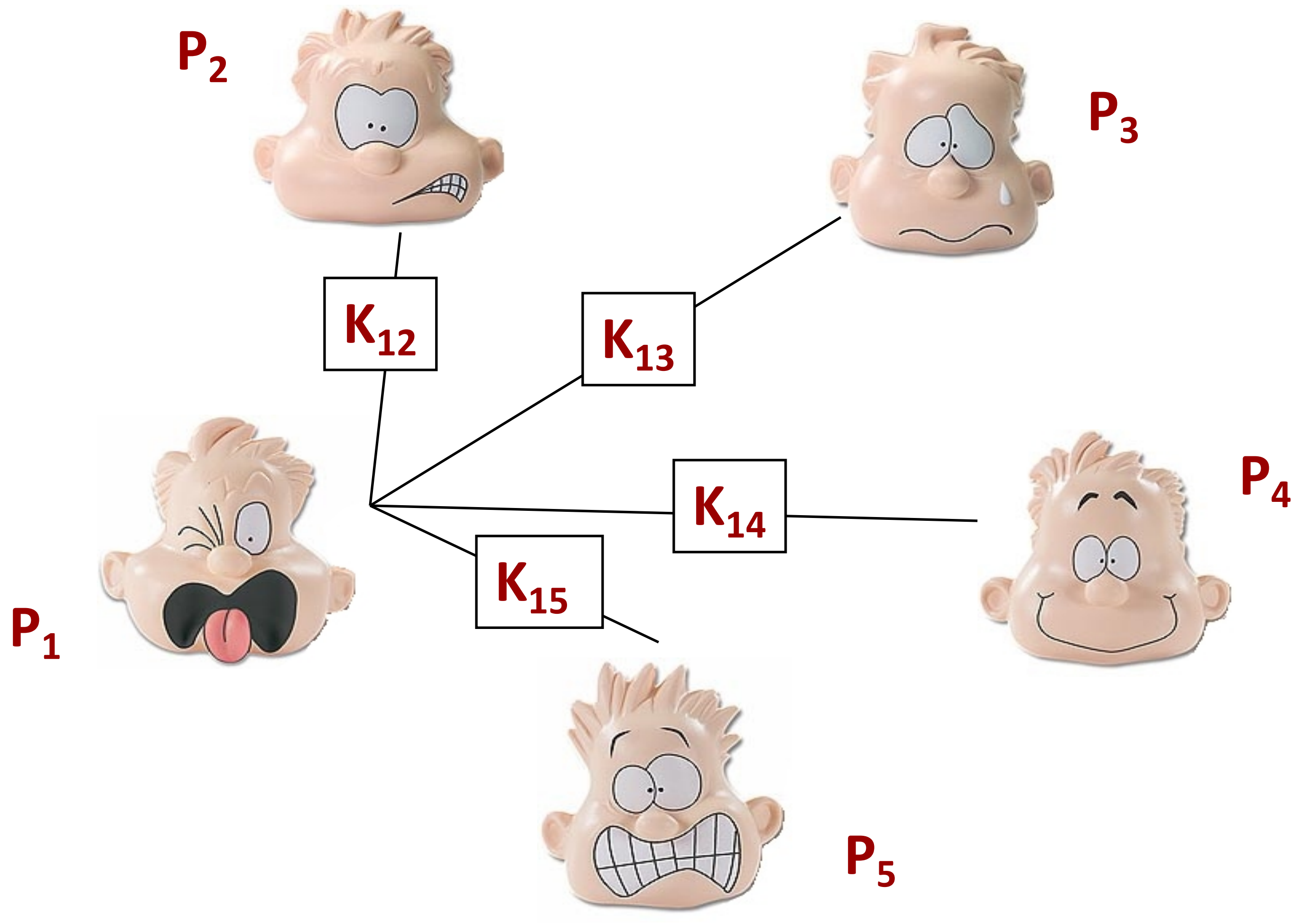
- If the users can meet in person beforehand – **it's simple.**

- But what to do if they **cannot meet**?

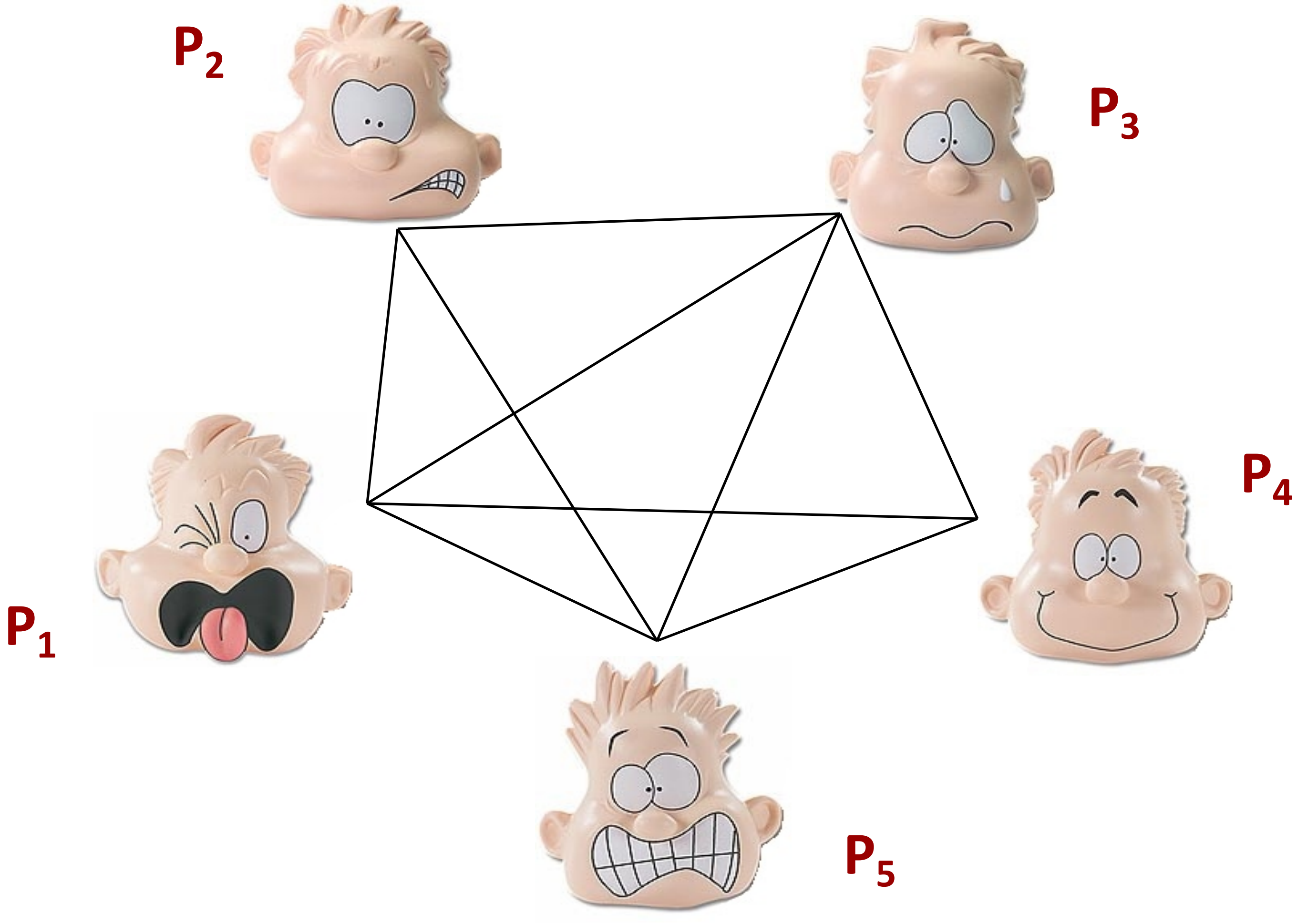
  (a typical example: on-line shopping)

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:

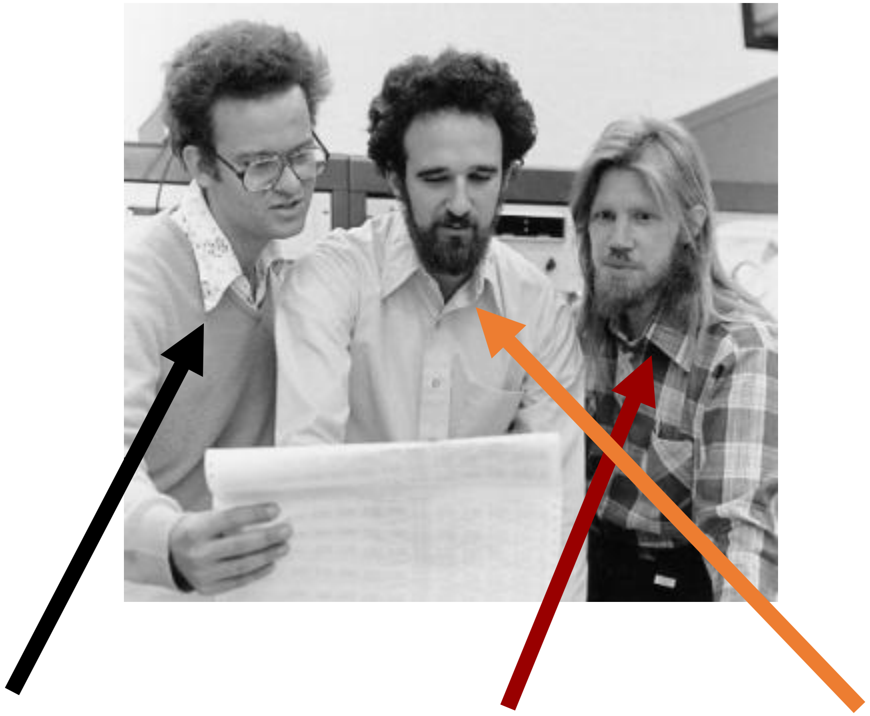
- Someone (a **Key Distribution Center, KDC**) needs to "give the keys"
    - **feasible** if the users are e.g. working in one company
    - **infeasible** on the internet
    - relies on the honesty of **KDC**
    - **KDC** needs to be permanently available
    - ...
- The users need to store large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

<u>Public-Key Cryptography</u>



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**.

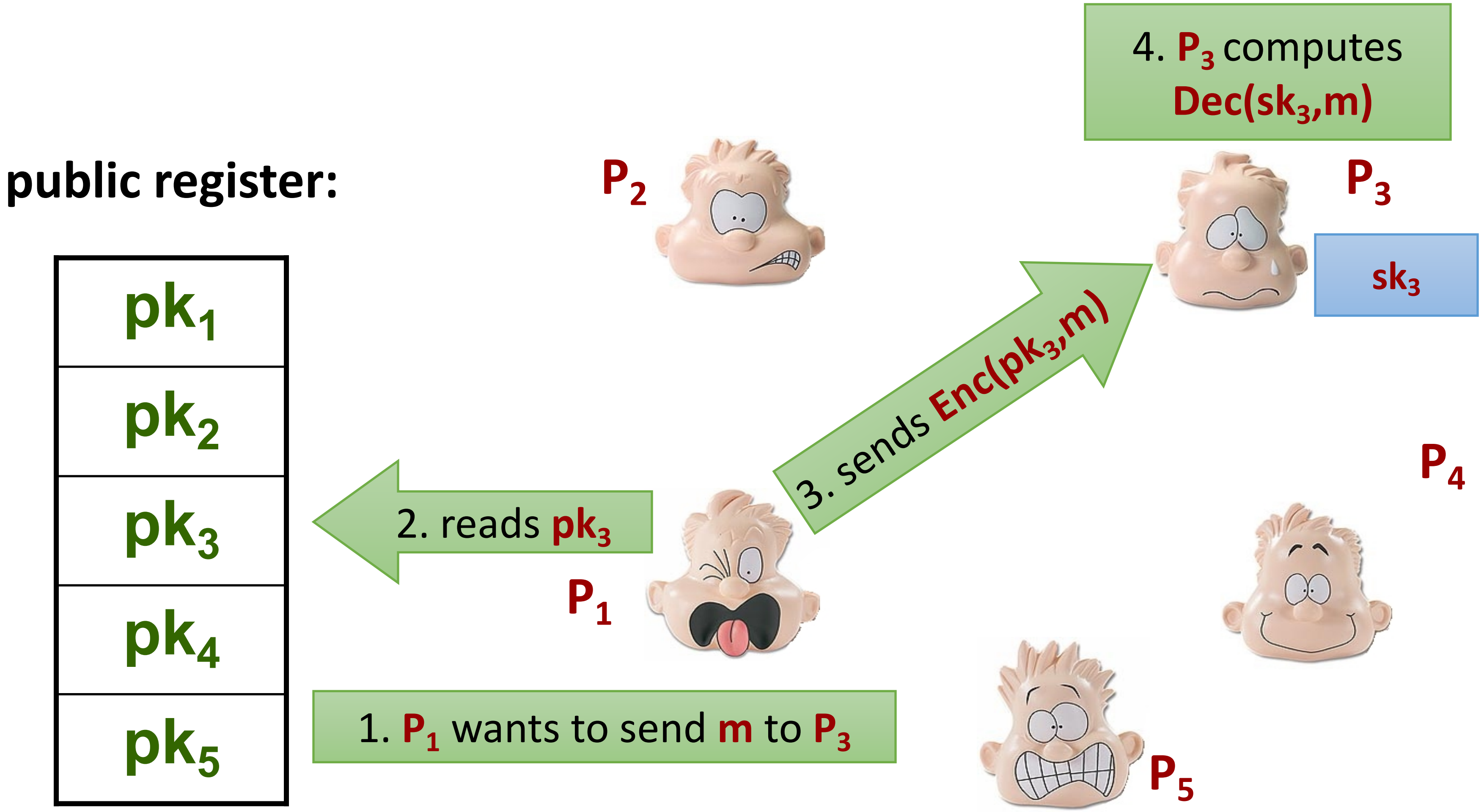
this will be called "signatures"

Sign – the signing algorithm

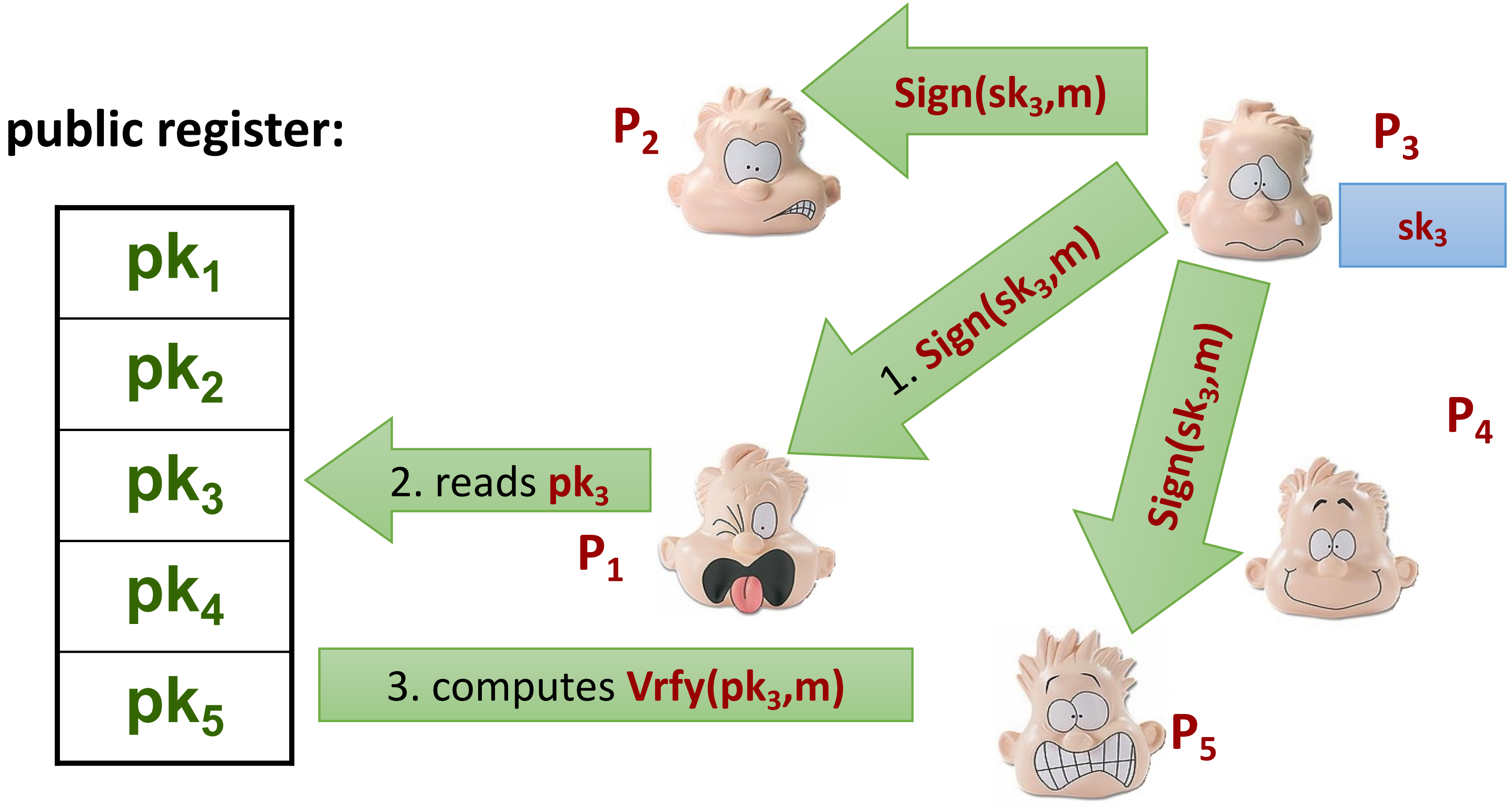
Moreover: **pk** can be public, and only **sk** has to be kept secret!

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

Anyone can send encrypted messages to anyone else



Anyone can verify the signatures



Things that need to be discussed

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

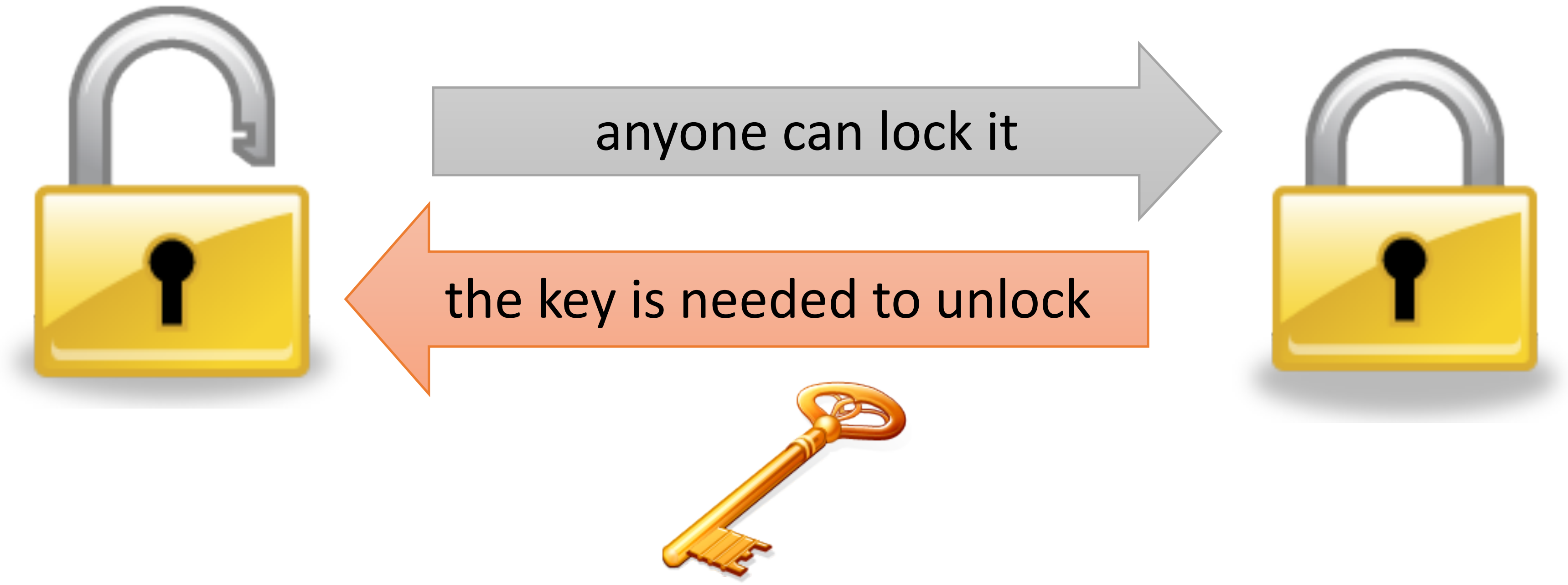
We will discuss this things later
(when we will be talking about the **Public-Key Infrastructure**)

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 won the Turing award for that work in 2016 (Turing award is considered to be the Nobel Prize for Computer Science)

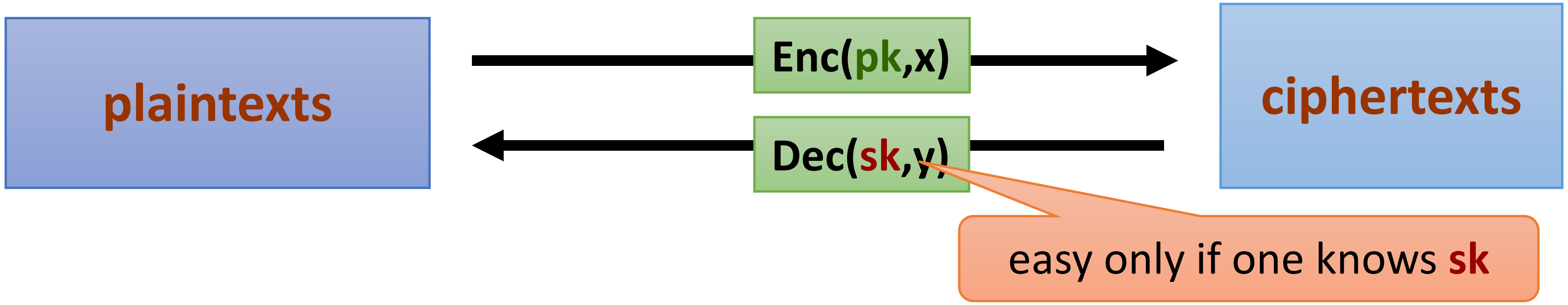
- They just proposed the **concept**, not the **implementation**.

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

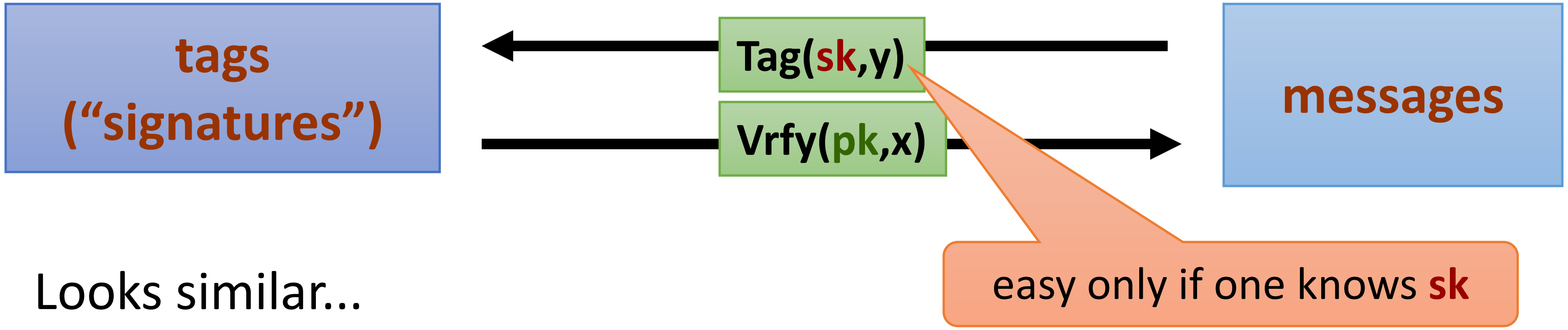
The observation of Diffie and Hellman:

(pk, sk) – the key pair

public-key encryption:



signature schemes:



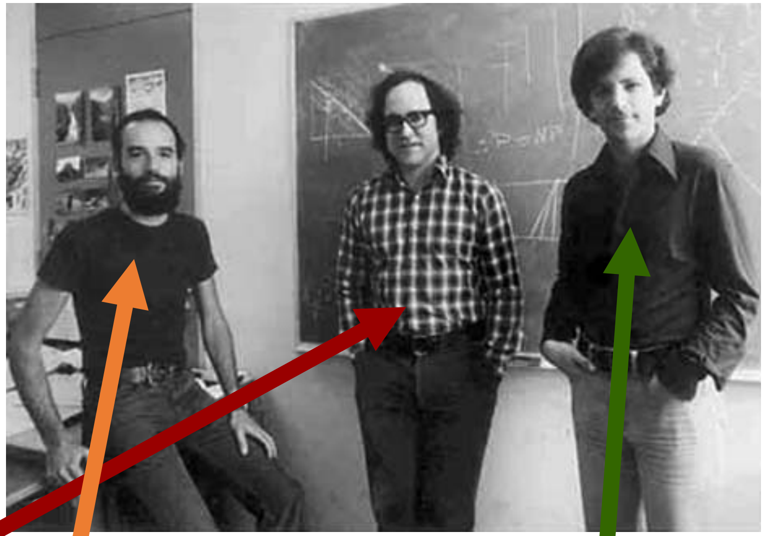
Looks similar...

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)

$Enc_{pk}: Z_N^* \rightarrow Z_N^*$ is defined as:

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

$Dec_{sk}: Z_N^* \rightarrow Z_N^*$ is defined as:

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

An observation

From the previous lecture we know that

- $\mathbf{Enc_{pk}: Z_N^* \rightarrow Z_N^*}$ is a permutation and
- $\mathbf{Dec_{sk}: Z_N^* \rightarrow Z_N^*}$ is its inverse.

Fact

$\mathbf{Enc_{pk}}$ is also a permutation over $\mathbf{Z_N}$ and $\mathbf{Dec_{sk}}$ is its inverse.

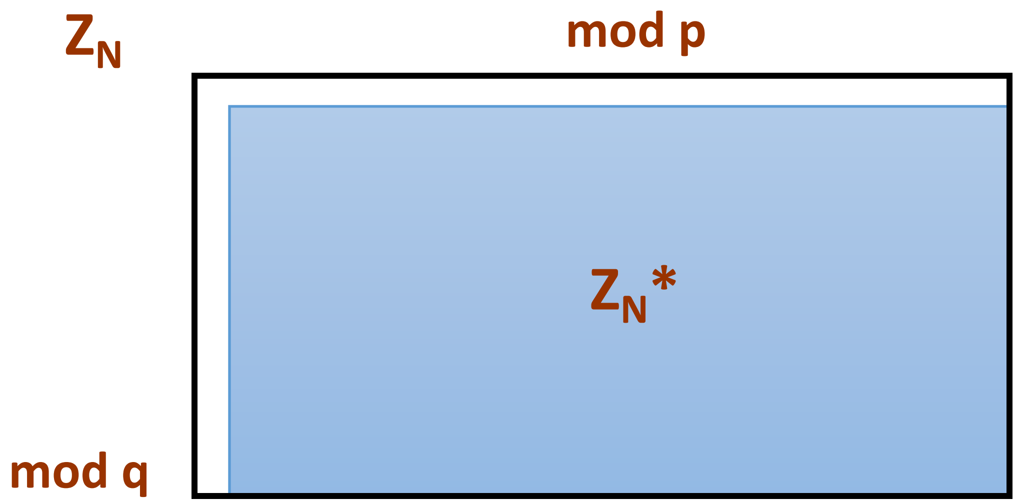
In fact, this doesn't even matter that much because:

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

$$\mathbf{gcd(a, N) > 1.}$$

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

A proof of the fact from the previous slide



Suppose **x = 0 mod p.**

Then (trivially) $(x^e)^d = x \bmod p$

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

CRT → $(x^e)^d = x \bmod N$

because: **ed = 1 mod (p-1)(q-1),**
and therefore **ed = 1 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.