

Lecture 7

Signature Schemes

Stefan Dziembowski
University of Rome
La Sapienza



SAPIENZA
UNIVERSITÀ DI ROMA

BiSS 2009
Bertinoro International
Spring School
2-6 March 2009



Plan



1. The definition of secure signature schemes
2. Signatures based on RSA, “hash-and-sign”, “full-domain-hash”
3. Other constructions

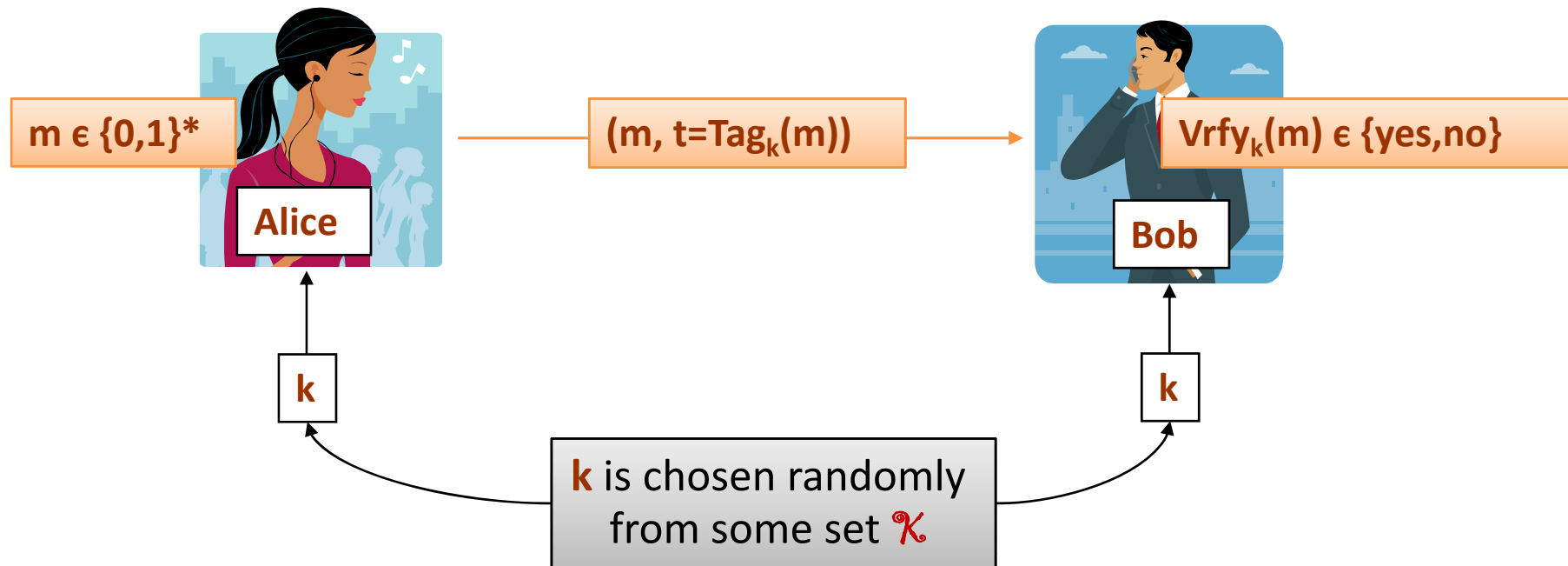
Signature schemes

digital signature schemes

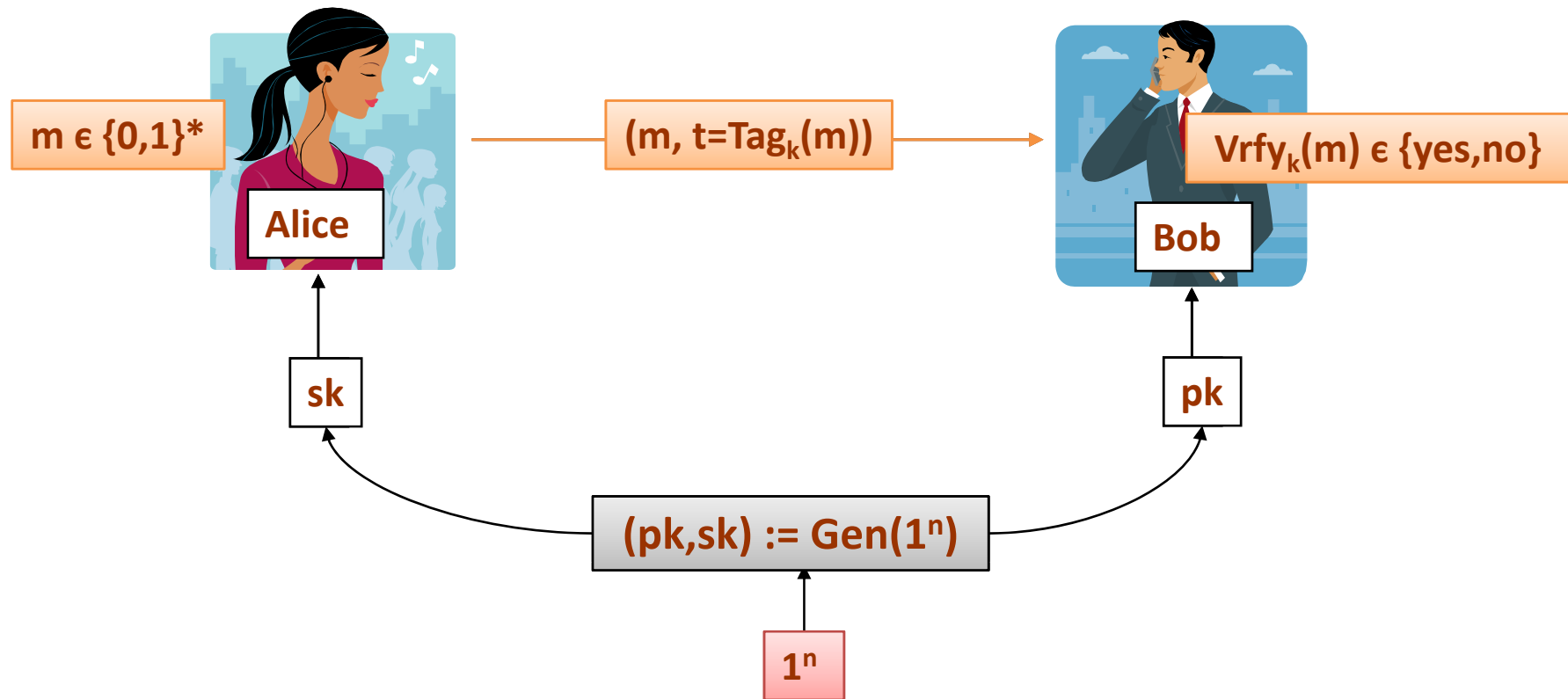


MACs in the public-key setting

Message Authentication Codes – the idea



Signature Schemes



Advantages of the signature schemes

Digital signatures are:

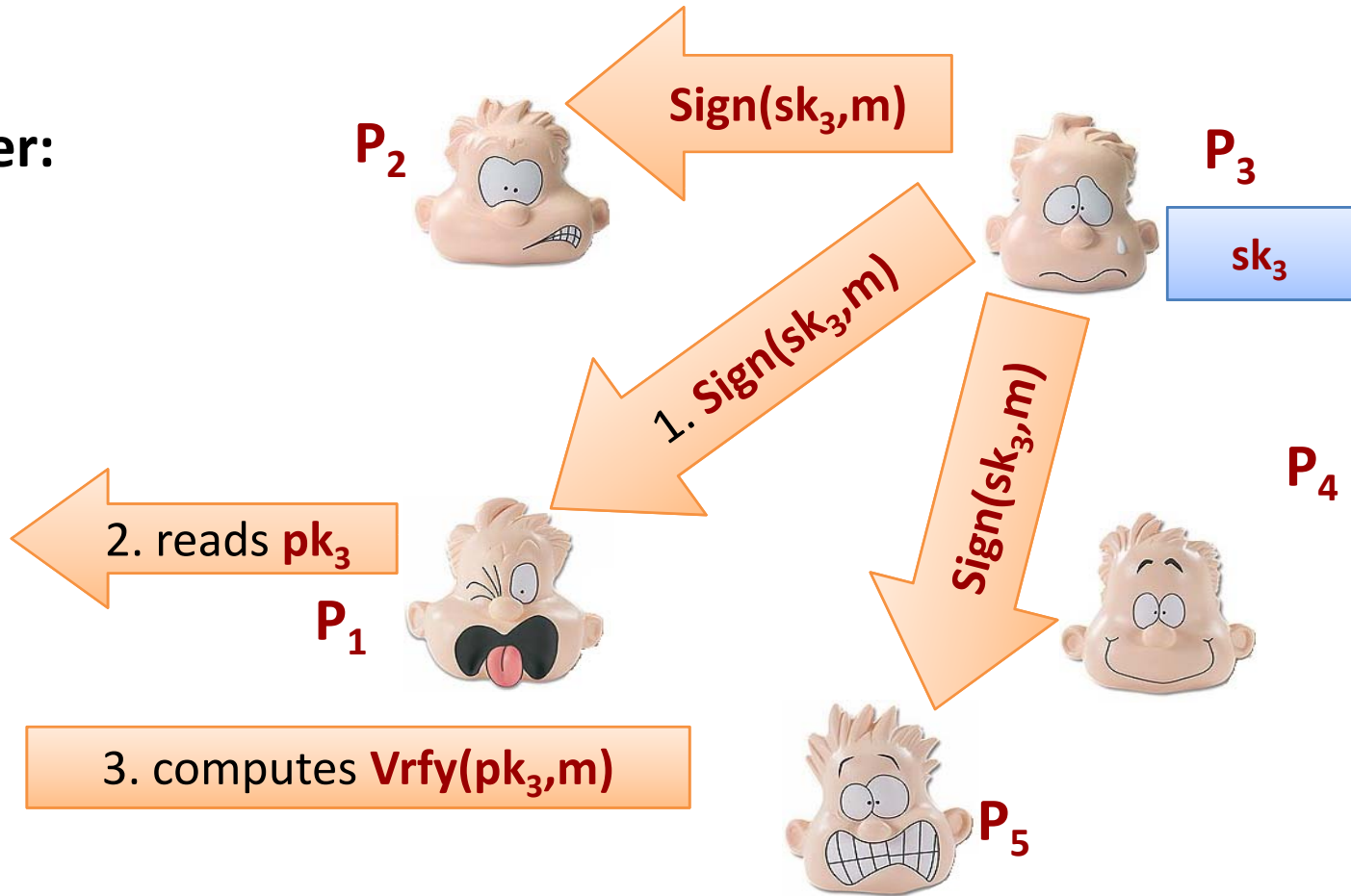
1. **publicly verifiable**
2. **transferable**
3. provide **non-repudiation**

(we explain it on the next slides)

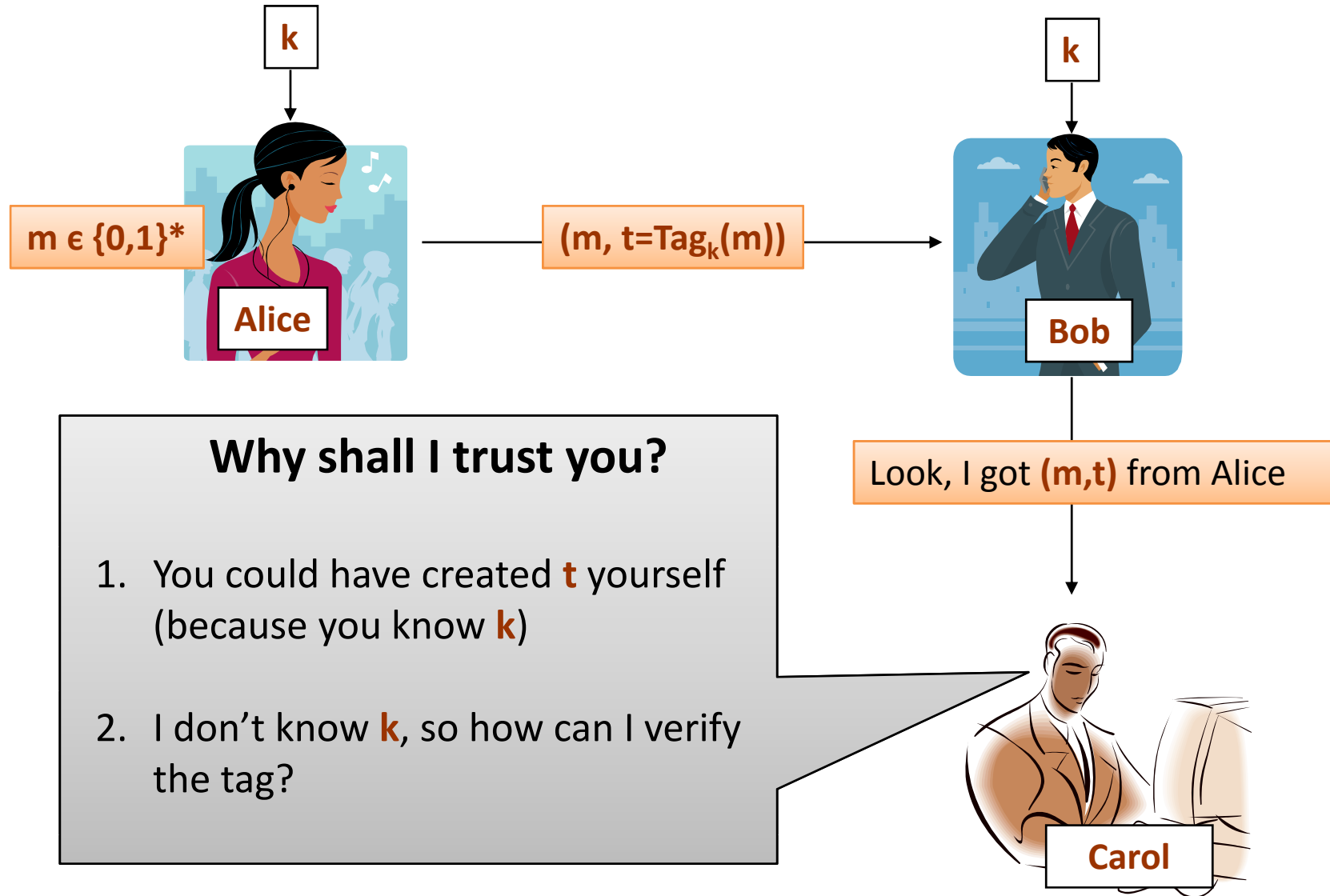
Anyone can verify the signatures

public register:

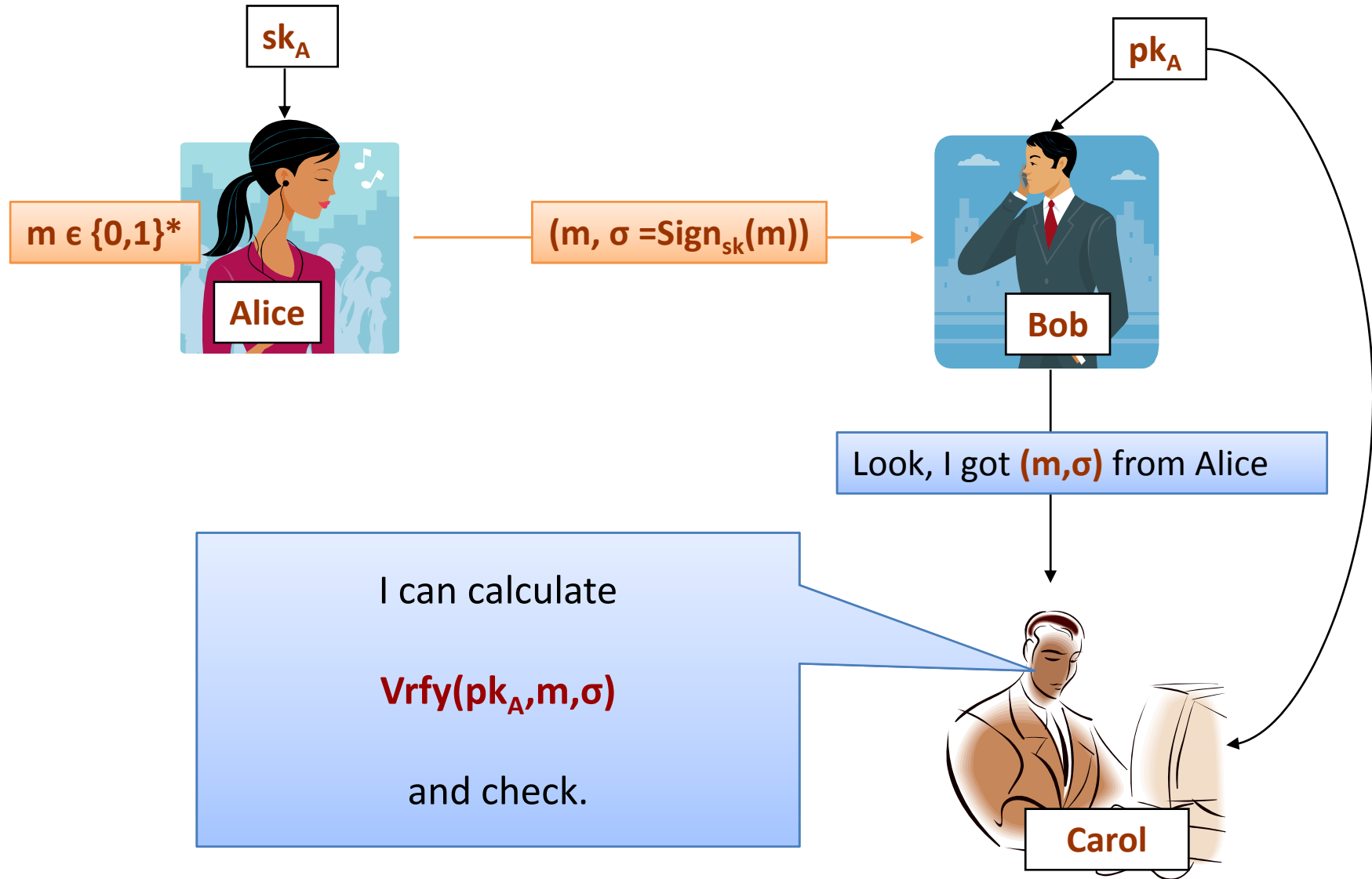
pk_1
pk_2
pk_3
pk_4
pk_5



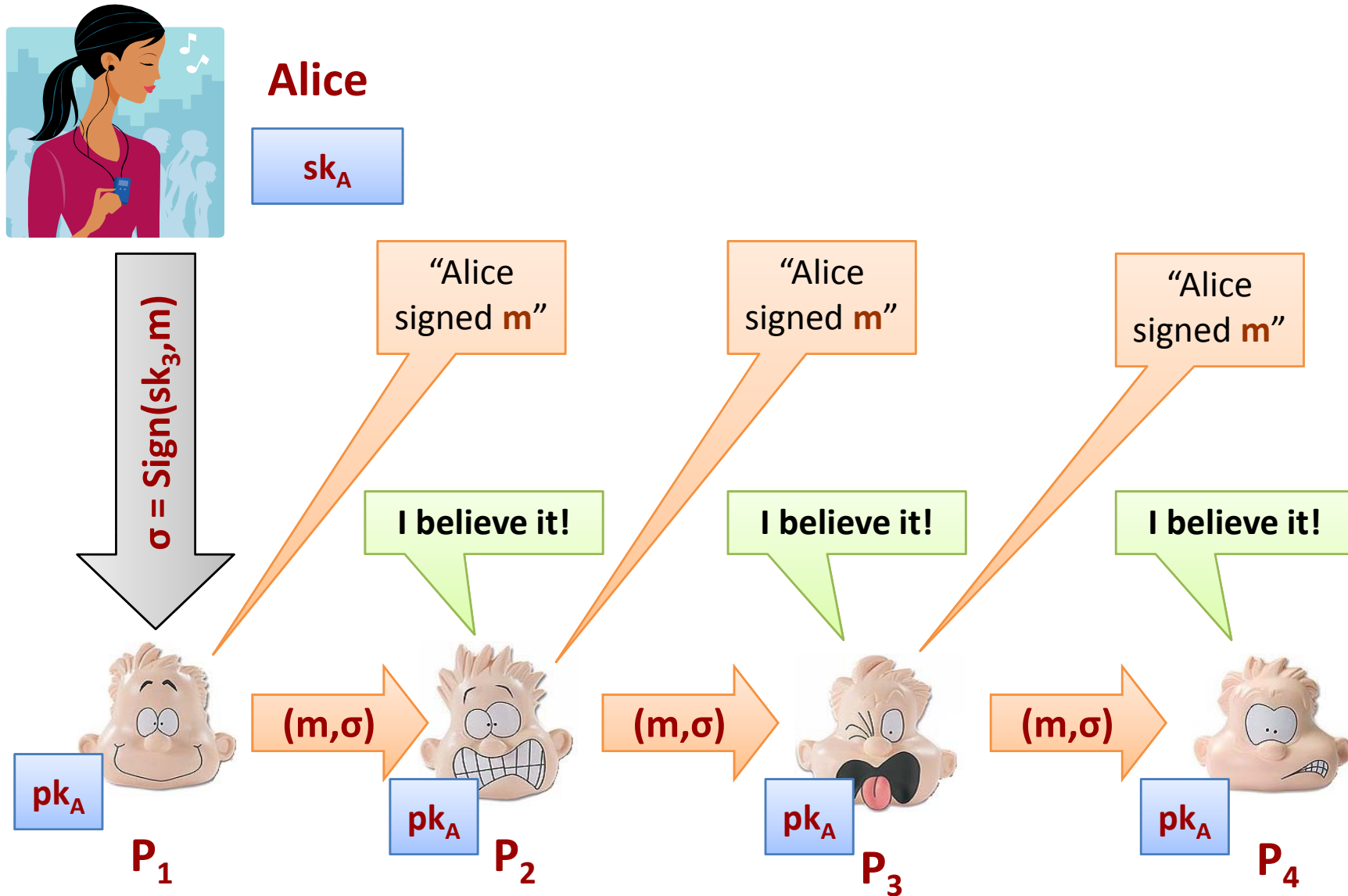
Look at the MACs...



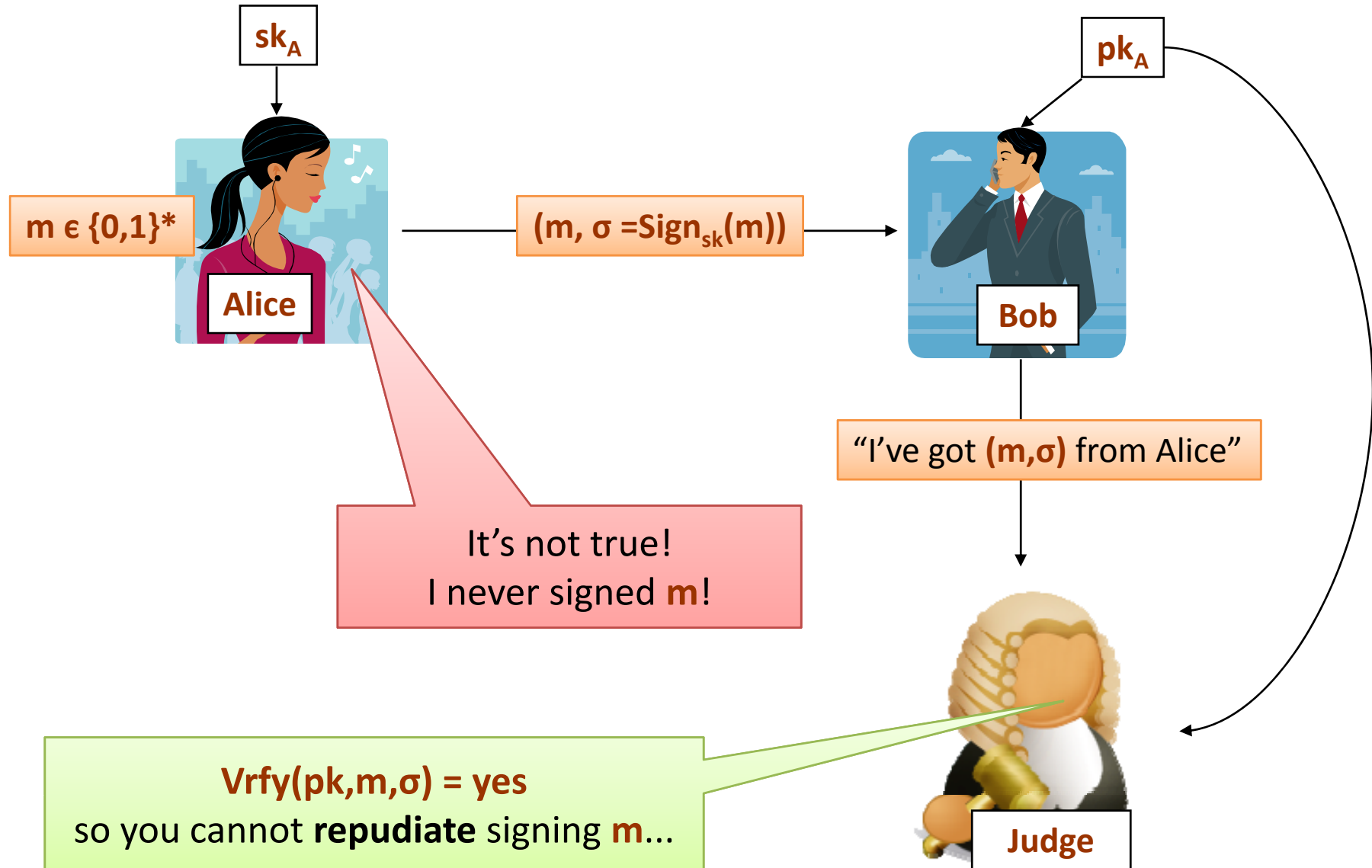
Signatures are publicly-verifiable!



So, the signatures are transferable



Non-repudiation



Digital Signature Schemes

A **digital signature scheme** is a tuple **(Gen, Sign, Vrfy)** of poly-time algorithms, such that:

- the **key-generation** algorithm **Gen** takes as input a security parameter 1^n and outputs a pair **(pk, sk)**,
- the **signing** algorithm **Sign** takes as input a key **sk** and a message $m \in \{0,1\}^*$ and outputs a signature σ ,
- the **verification** algorithm **Vrfy** takes as input a key **pk**, a message **m** and a signature σ , and outputs a bit $b \in \{\text{yes}, \text{no}\}$.

If $\text{Vrfy}_{pk}(m, \sigma) = \text{yes}$ then we say that σ is a **valid signature on the message m**.

Correctness

We require that it always holds that:

$$\text{Vrfy}_{pk}(m, \text{Sign}_{sk}(m)) = \text{yes}$$

What remains is to define **security** of a **MAC**.

How to define security?

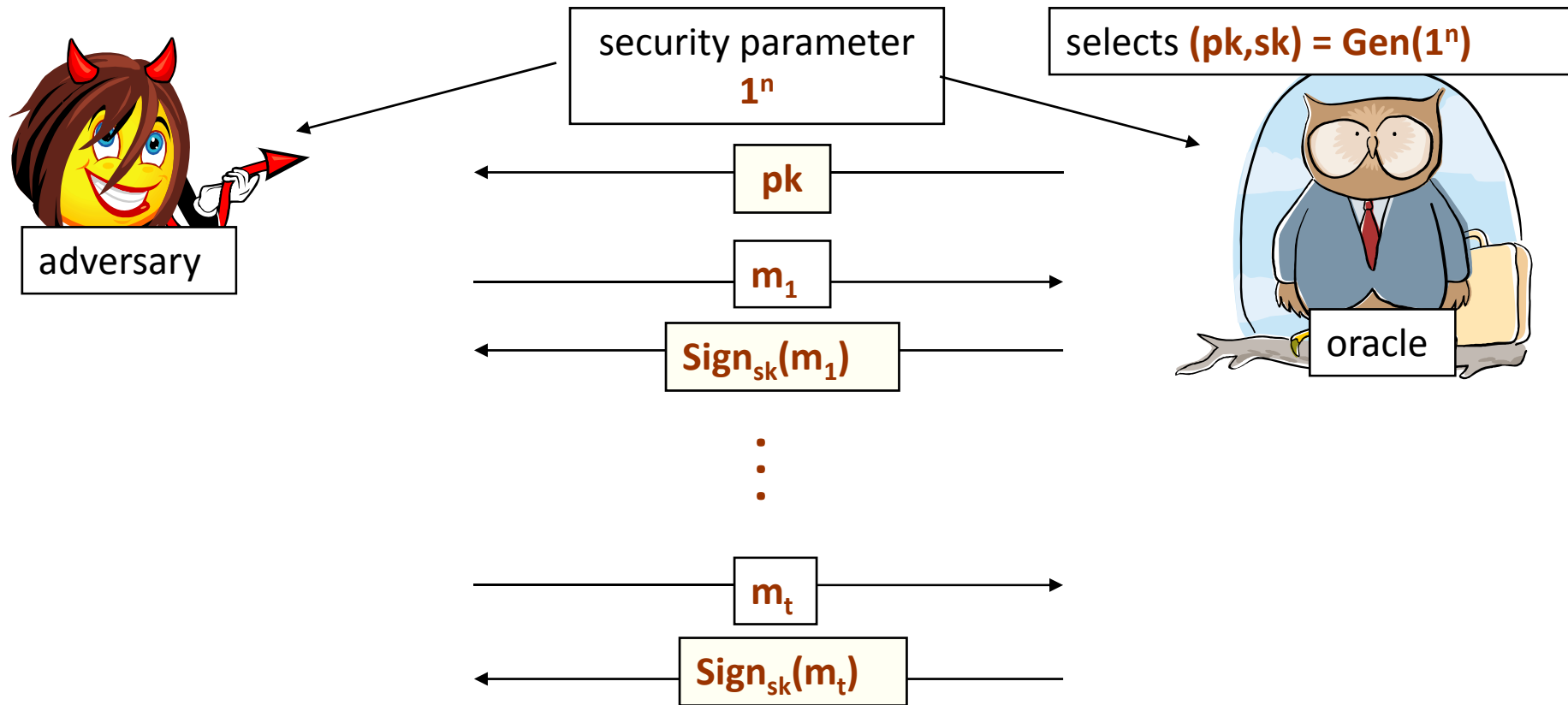
As in the case of MACs, we need to specify:

1. how the messages m_1, \dots, m_t are chosen,
2. what is the goal of the adversary.

Good tradition: be as pessimistic as possible!

Therefore we assume that

1. The adversary is allowed to chose m_1, \dots, m_t .
2. The **goal of the adversary** is to produce a valid signature on some m' such that $m' \neq m_1, \dots, m_t$.



We say that the adversary **breaks the signature scheme** if at the end she outputs (m', σ') such that

1. $\text{Vrfy}(m', \sigma') = \text{yes}$
2. $m' \neq m_1, \dots, m_t$

The security definition

sometimes we just say: **unforgeable** (if the context is clear)

We say that **(Gen, Sign, Vrfy)** is **existentially unforgeable under an adaptive chosen-message attack** if

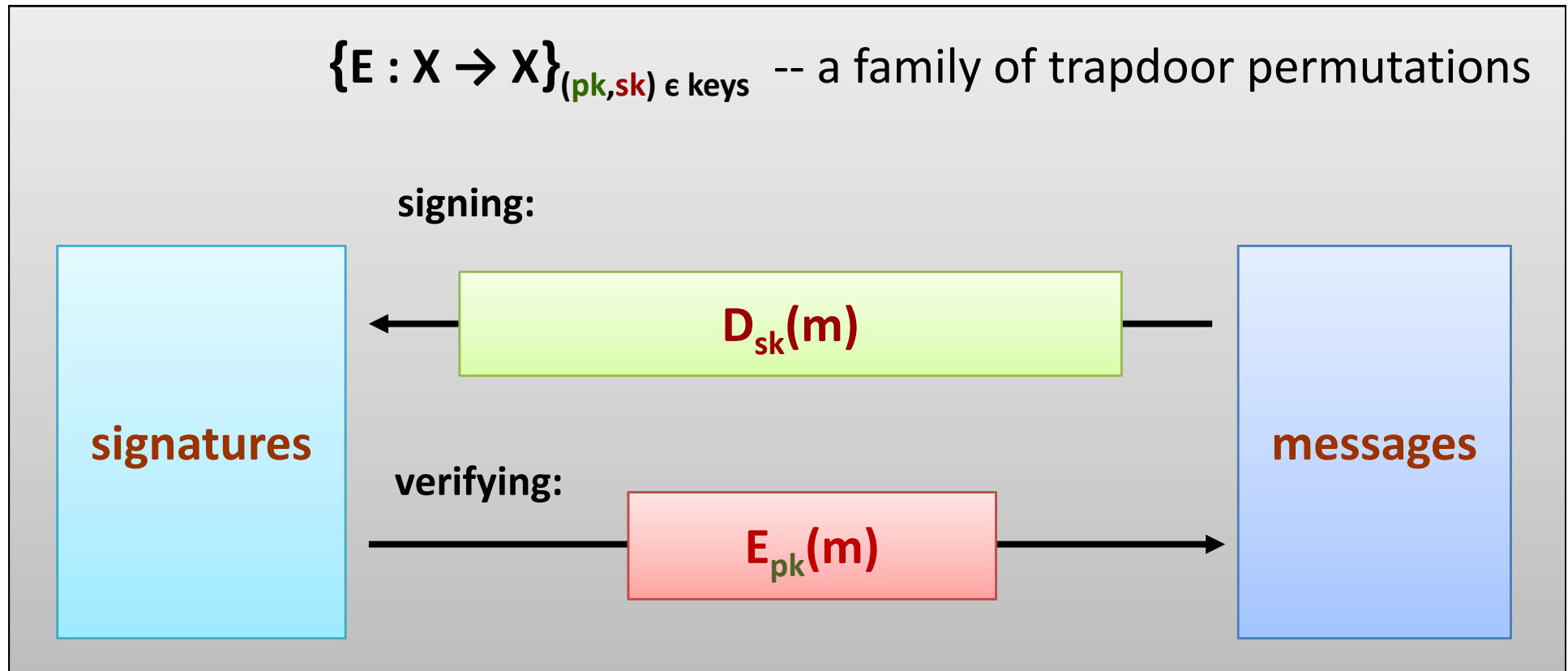


P(A breaks it) is negligible (in **n**)

polynomial-time
adversary **A**

How to design secure signature schemes?

Remember this idea?



We said: In general it's not that simple.

In general it's not that simple

1. Not every trapdoor permutation is OK.

2. There exist other ways to create signature schemes.

3. One can even construct a signature scheme from any one-way function.
(this is a theoretical construction)

Plan

1. The definition of secure signature schemes



2. Signatures based on RSA, “hash-and-sign”, “full-domain-hash”

3. Other constructions

The “handbook RSA signatures”

$N = pq$ - RSA modulus

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

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

$\text{Sign}_{(d,N)}(c) = c^d \pmod{N}$

and

$\text{Vrfy}_{(e,N)}(m, \sigma) = \text{yes}$ iff $\sigma^e = m \pmod{N}$

Correctness:

$$\begin{aligned}\sigma^e &= (m^d)^e \\ &= m^{de} \\ &= m^1 \\ &= m\end{aligned}$$

Problems with the “handbook RSA” [1/2]

A “no-message attack”:

The adversary can forge a signature on a “random” message **m**.

Given the public key **(N,e)**:

he just selects a random **σ** and computes

$$\mathbf{m} = \sigma^e \bmod N.$$

Trivially, **σ** is a valid signature on **m**.

Problems with the “handbook RSA” (2/2)

How to forge a signature on an arbitrary message m ?

Use the homomorphic properties of RSA.



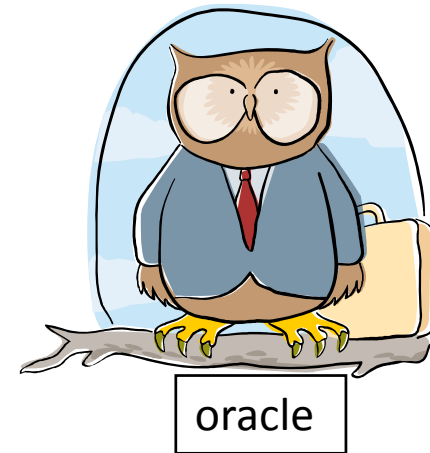
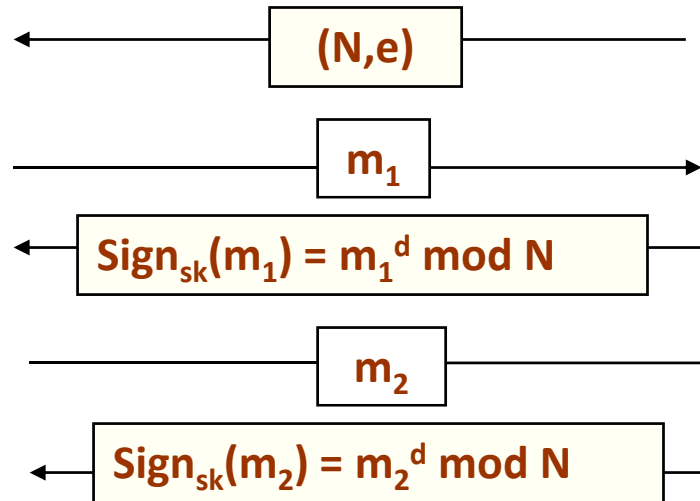
chooses:

1. random m_1
2. $m_2 := m / m_1 \pmod N$

computes ($\pmod N$):

$$\begin{aligned} & m_1^d \cdot m_2^d \\ = & (m_1 \cdot m_2)^d \\ = & m^d \end{aligned}$$

this is a valid signature on m



Is it a problem?

In many applications – probably not.

But we would like to have schemes that are not application-dependent...

Solution

Before computing the RSA function – apply some function **H**.

N = pq, such that **p** and **q** are large random primes
e is such that **gcd(e, φ(N)) = 1**
d is such that **ed = 1 (mod φ(N))**

Sign_d: Z_N^{*} → Z_N^{*} is defined as:
Sign(m) = H(m)^d mod N.

Vrfy_e is defined as:
Vrfy_e(m,σ) = yes iff σ^e = H(m) (mod N)

How to choose such **H**?

A minimal requirement:

it should be collision-resistant.

(because if the adversary can find two messages

m, m' such that

$$\mathbf{H(m) = H(m')}$$

then he can forge a signature on **m'** by asking
the oracle for a signature on **m**)

A typical choice of **H**

Usually **H** is one of the popular **hash functions**.

Additional advantage:

We can sign very long messages keeping the modulus **N** small (it's much more efficient!).

It is called a

hash-and-sign paradigm.

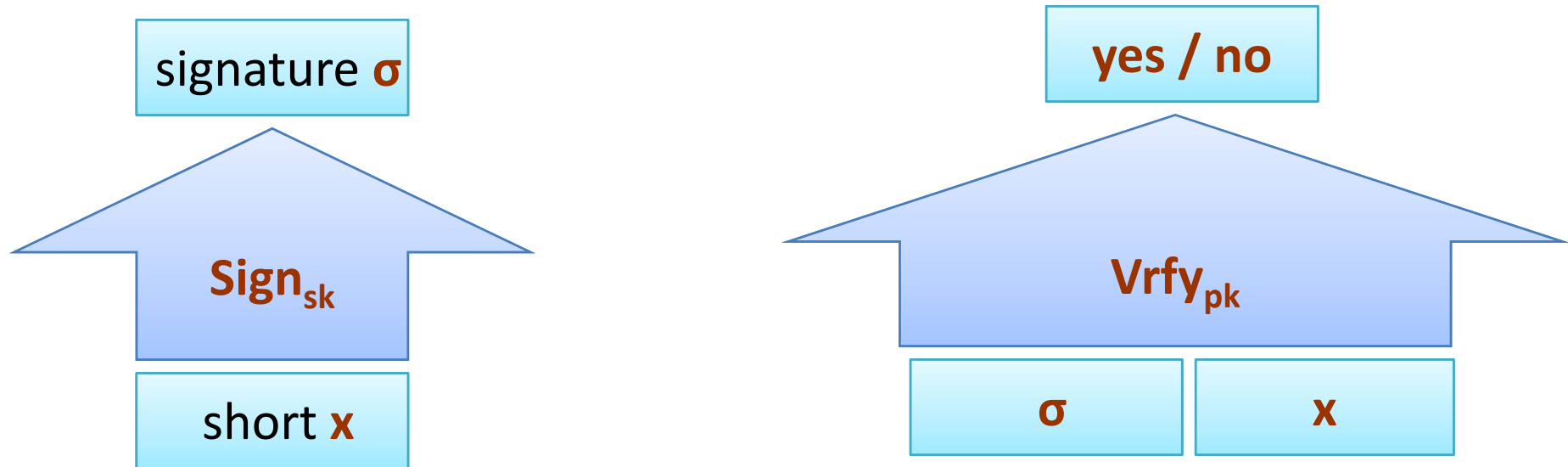
Hash-and-Sign [1/5]

Hash and sign is a generic construction that takes as input:

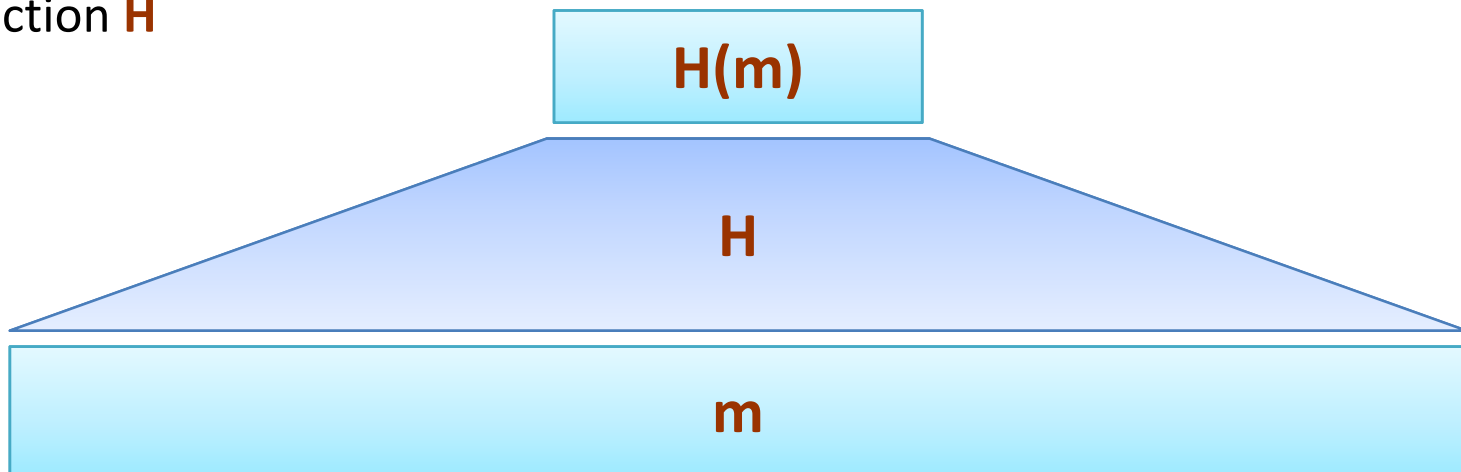
- a signature scheme that works on “**short messages**”, and
 - a **hash function**,
- and transforms it into a
- a signature scheme that works on “**long messages**”.

Hash-and-Sign [2/5]

1. **(Gen, Sign, Vrfy)** – a signature scheme “for short messages”

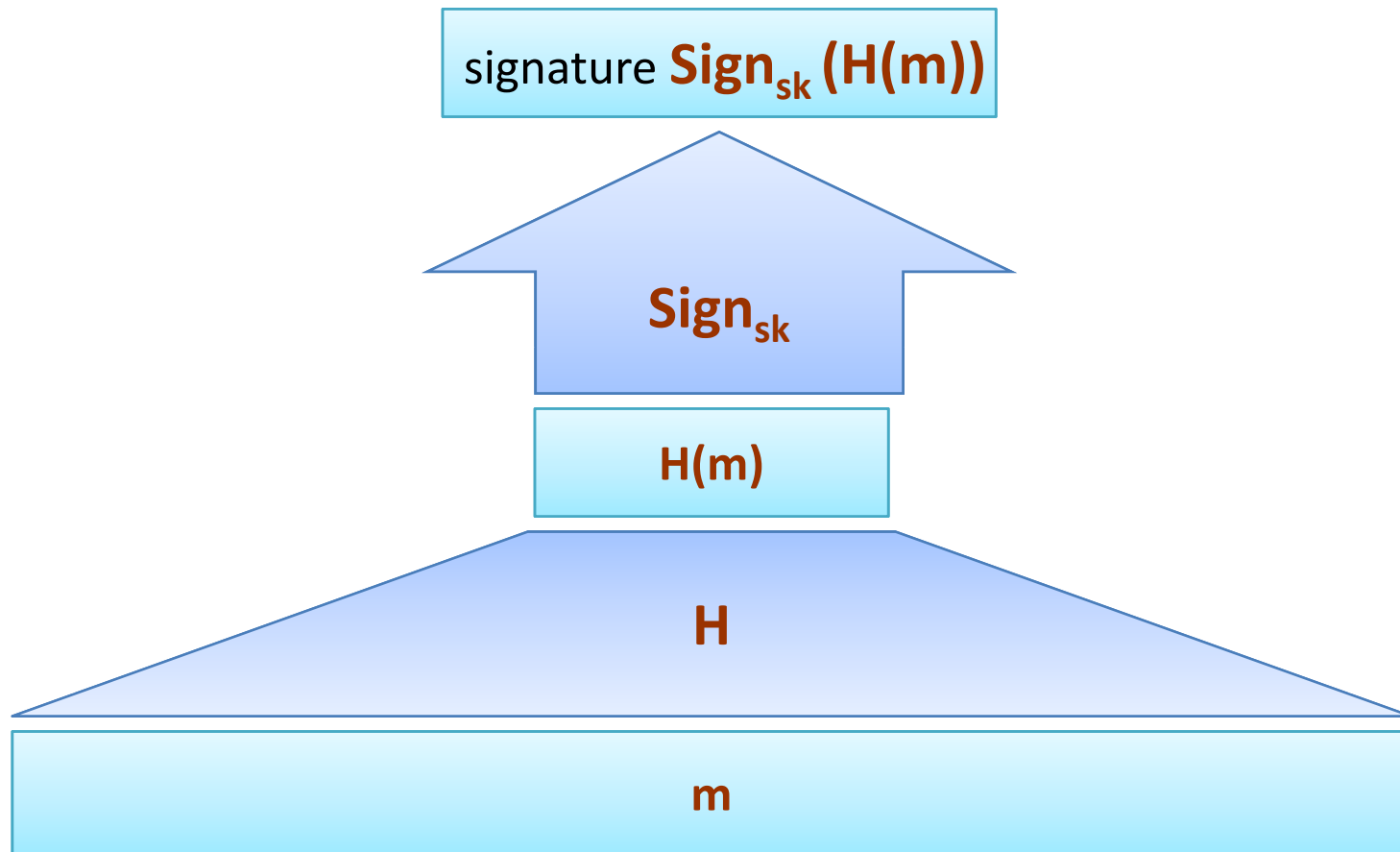


2. a hash function H



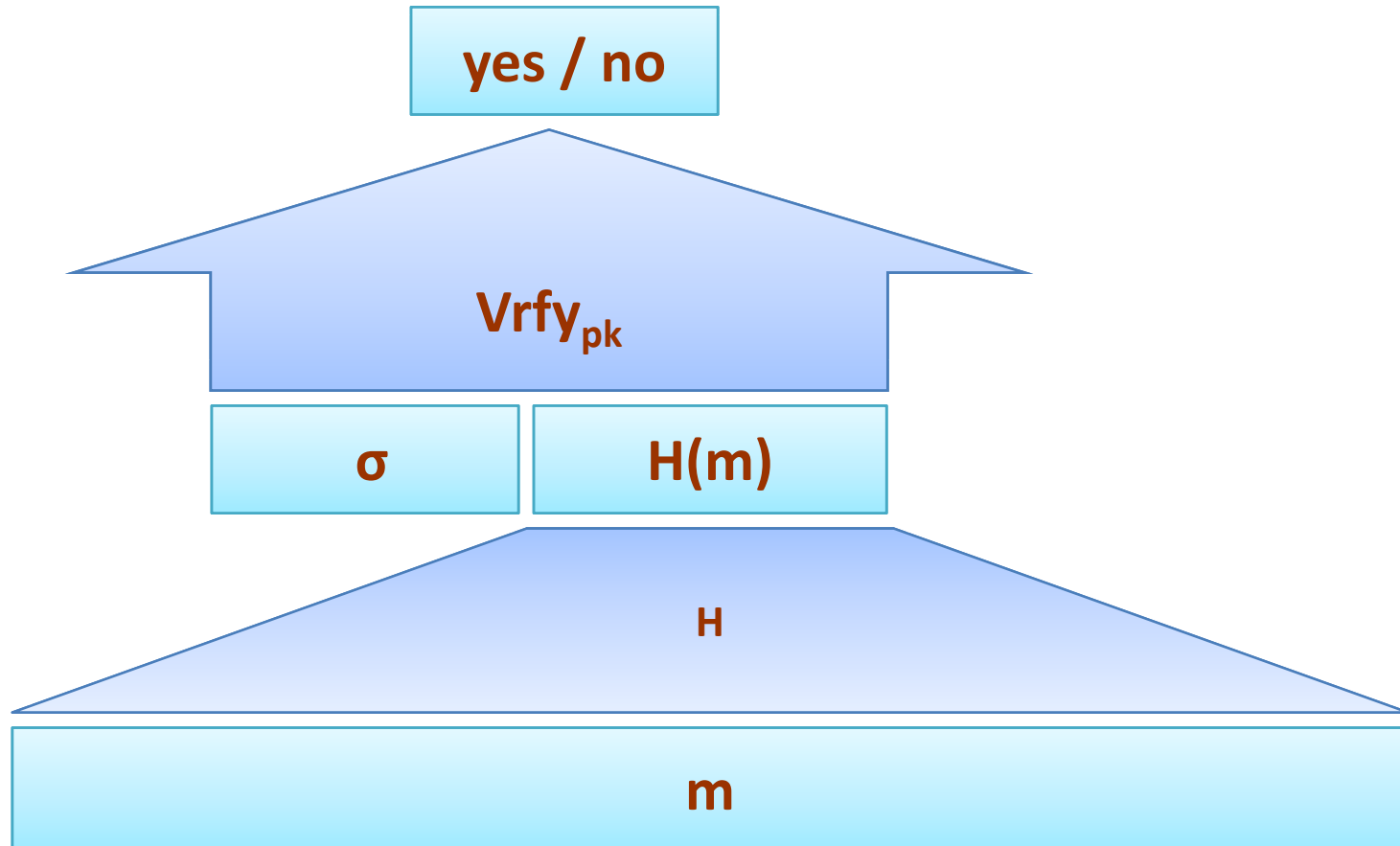
Hash-and-Sign [3/5]

How to sign a message m ?



Hash-and-Sign [4/5]

How to verify?



Hash-and-Sign [5/5]

It can be proven that this construction is secure.

For this we need to assume that **H** is taken from a family of collision-resilient hash functions.

$$\{H^s\}_{s \in \text{keys}}$$

Then **s** becomes a part of the public key and the private key.

What can be proven

Suppose

1. $\{H^s\}_{s \in \text{keys}}$ is a family of collision-resistant hash functions,
2. $(\text{Gen}, \text{Sign}, \text{Vrfy})$ is a secure signature scheme.

Then the signature scheme constructed on the previous slide is secure.

Can anything be proven about the “hashed RSA” scheme?

In the plain model - not really.

But at least the attacks described before “look infeasible”.

1. For the “no message attack”: one would need to invert **H**.
2. The second (“homomorphic”) attack:
Looks impossible because the adversary would need to find messages **m, m₁, m₂** such that

$$H(m) = H(m_1) \cdot H(m_2)$$

Fact (security of the **Full Domain Hash**)

- Let $H : \{0,1\}^* \rightarrow \mathbb{Z}_N^*$ be a hash function modeled as a **random oracle**.
- Suppose the **RSA assumption** holds

Then the “**hashed RSA**” is existentially unforgeable under an adaptive chosen-message attack

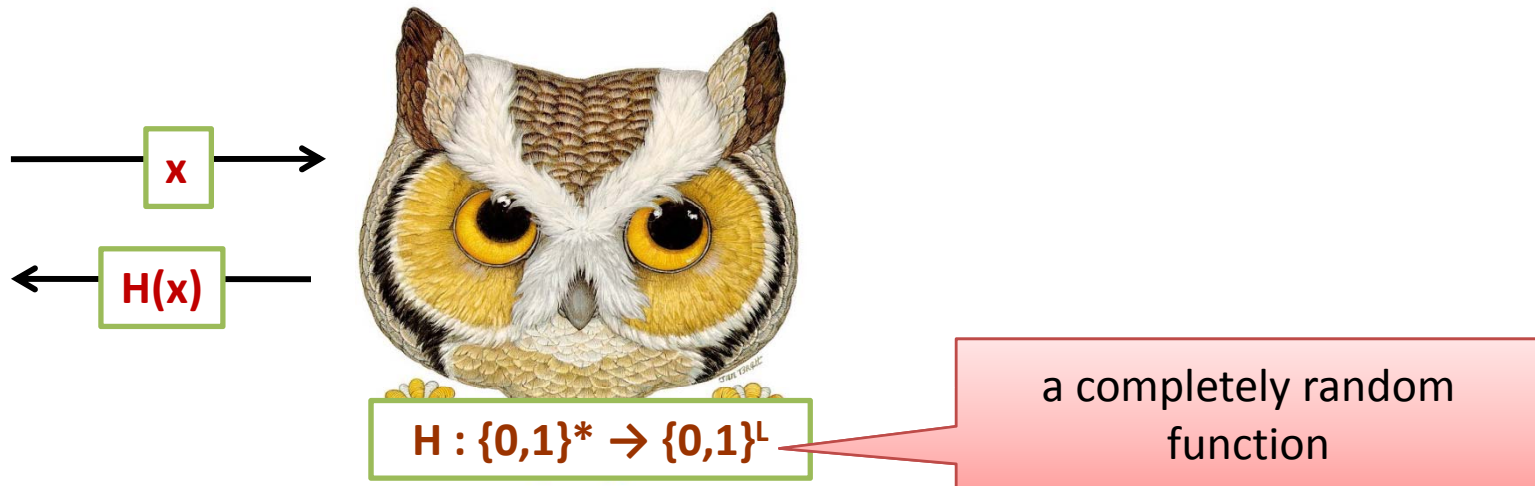
hashed RSA

$N = pq$, such that p and q are large random primes
 e is such that $\gcd(e, \phi(N)) = 1$
 d is such that $ed = 1 \pmod{\phi(N)}$

$\text{Sign}_d: \mathbb{Z}_N^* \rightarrow \mathbb{Z}_N^*$ is defined as:
 $\text{Sign}(m) = H(m)^d \pmod{N}$.

Vrfy_e is defined as:
 $\text{Vrfy}_e(m, \sigma) = \text{yes}$
iff $\sigma^e = H(m) \pmod{N}$

Remember the Random Oracle Model?



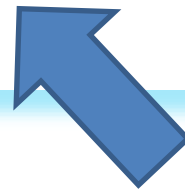
Why does it help?

RSA assumption

For any polynomial time algorithm **A** we have:

$P((A(x,N,e))^e = x \bmod N)$ is negligible

where **$N = pq$** where **p** and **q** are random primes such that **$|p| = |q|$** , and **x** is a random element of **Z_N^*** , and **e** is random element of **$Z_{\phi(N)^*}$**



here we require that **x** is random

Intuition

If we just use a “normal hash function” then the distribution of $H(m_0), H(m_1), H(m_2), \dots$ (for any m_0, m_1, m_2, \dots) can be “complicated”.

If H is a random oracle then $H(m_0), H(m_1), H(m_2), \dots$ are uniform and independent (for pairwise different m_i 's).

This helps a lot in the proof!

Plan

1. The definition of secure signature schemes
2. Signatures based on RSA, “hash-and-sign”, “full-domain-hash”
3. Other constructions



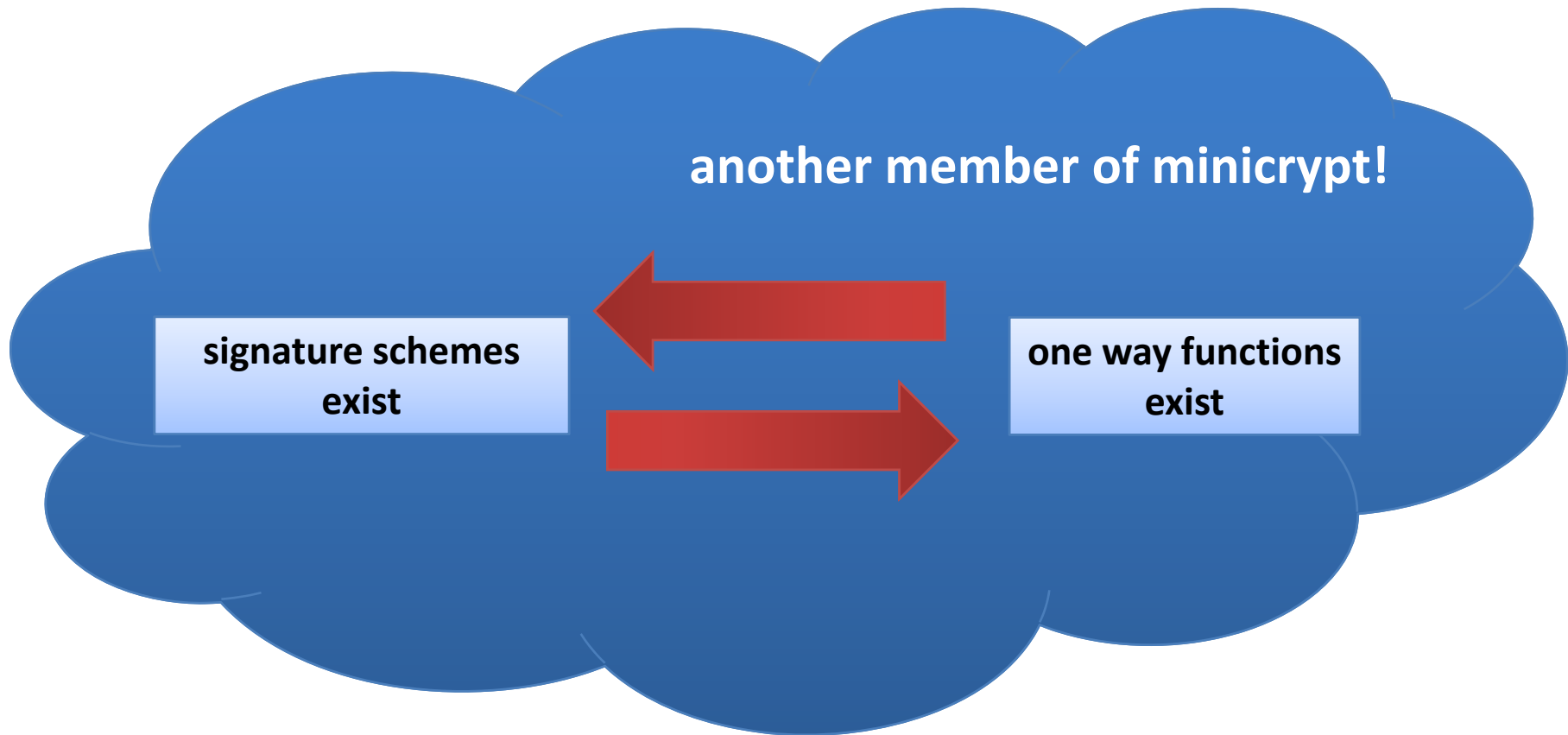
Other popular signature schemes

Based on discrete log:

- **ElGamal** signatures
- Digital Signature Standard (**DSS**)

(also based on other groups – elliptic curves)

Signatures schemes can be constructed
from any one-way function



©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.*