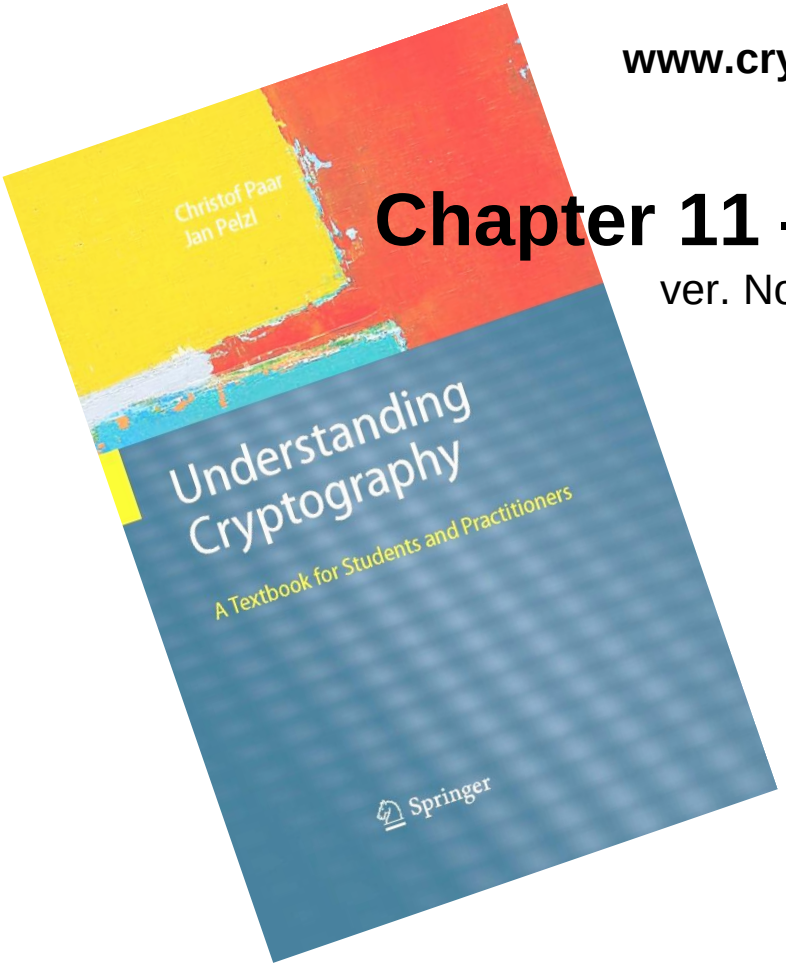
Understanding Cryptography – A Textbook for Students and Practitioners

by Christof Paar and Jan Pelzl

www.crypto-textbook.com



Chapter 11 – Hash Functions

ver. November 17, 2024

These slides were originally prepared by Stefan Heyse and Christof Paar and Jan Pelzl. Later, they were modified by Tomas Fabsic for purposes of teaching I-ZKRY at FEI STU.

Homework

- Read Sections 10.1.-10.2.
- Read Sections 11.1.-11.3.
- Solve problems from the exercise set no. 9 and submit them to AIS by **25.11.2024 23:59**.

Some legal stuff (sorry): Terms of Use

- The slides can used free of charge. All copyrights for the slides remain with Christof Paar and Jan Pelzl.
- The title of the accompanying book “Understanding Cryptography” by Springer and the author’s names must remain on each slide.
- If the slides are modified, appropriate credits to the book authors and the book title must remain within the slides.
- It is not permitted to reproduce parts or all of the slides in printed form whatsoever without written consent by the authors.

Content of this Chapter

- Why we need hash functions
- How it works
- Security properties
- Algorithms

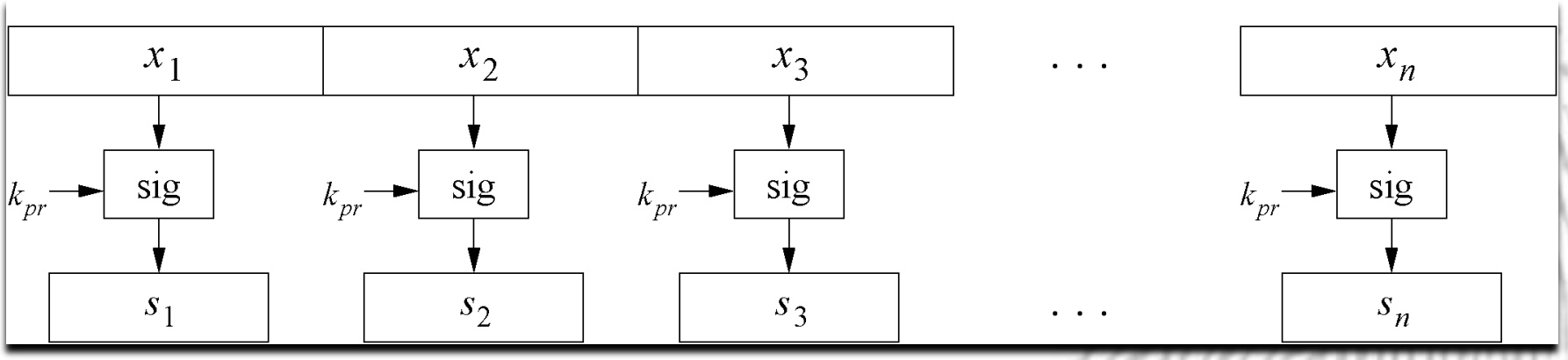
Content of this Chapter

- **Why we need hash functions**
- How it works
- Security properties
- Algorithms

Motivation

Problem:

Naive signing of long messages generates a signature of same length.



Three Problems:

- Computational overhead
- Message overhead
- Security limitations

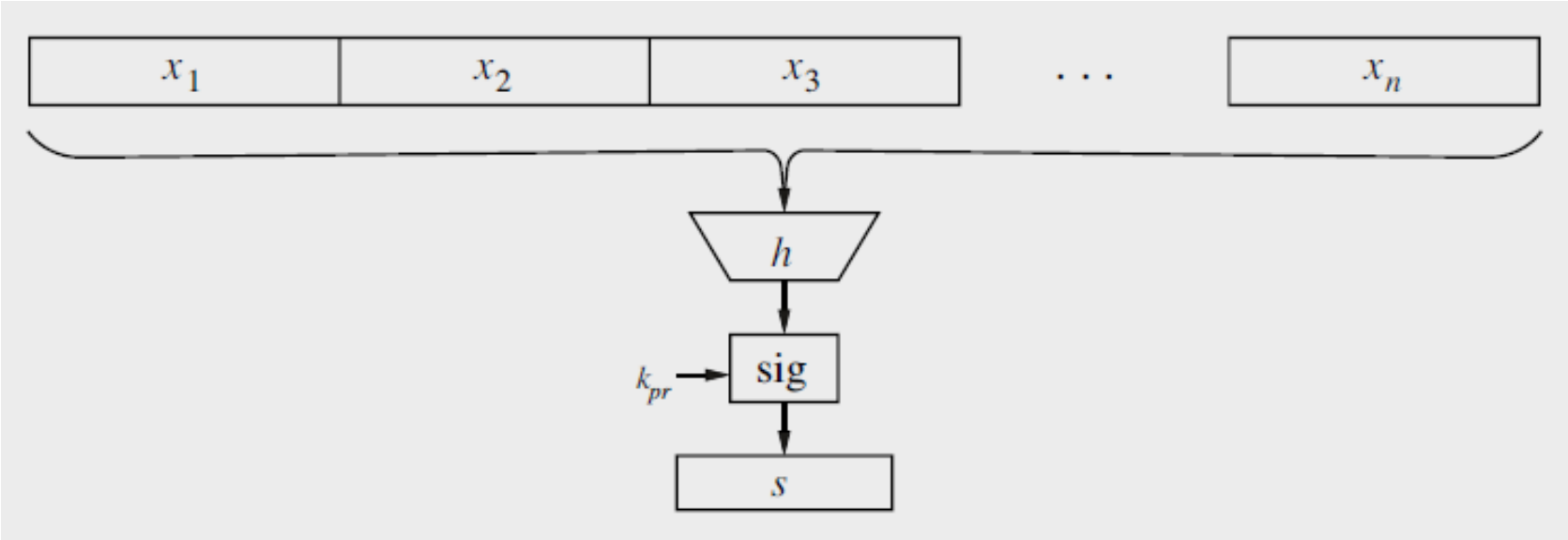
Solution:

Instead of signing the whole message, sign only a digest (=hash)

Needed:

Hash Functions

Digital Signature with a Hash Function



Notes:

- *h* does not require a key.
- *h* is public.

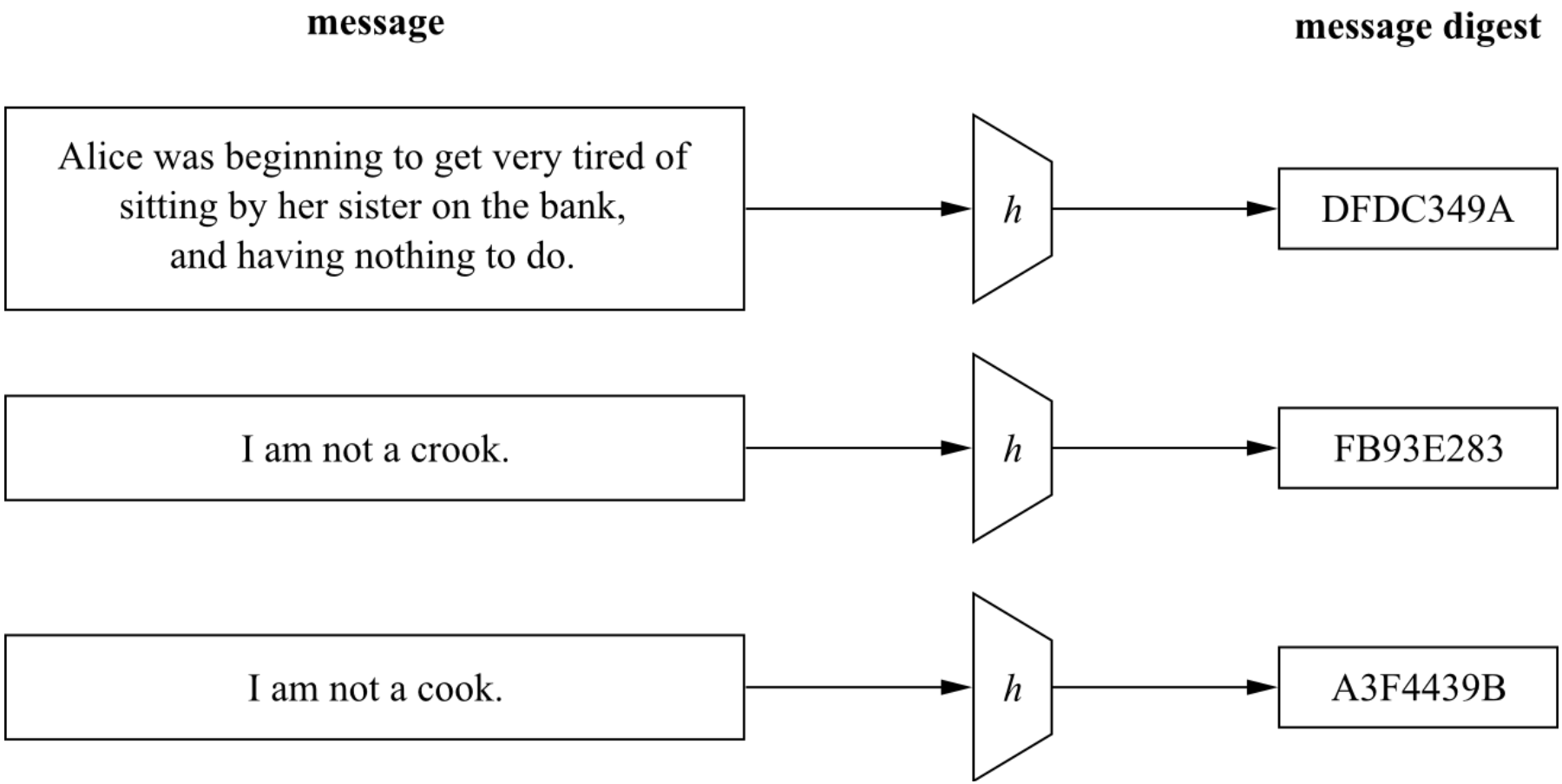
Basic Protocol for Digital Signatures with a Hash Function:

Alice		Bob
	$\xleftarrow{K_{pub}}$	
		$z = h(x)$
		$s = sig_{K_{pr}}(z)$
	$\xleftarrow{(x,\ s)}$	
$z' = h(x)$		
$ver_{K_{pub}}(s,z')=true/false$		

Content of this Chapter

- Why we need hash functions
- **How it works**
- Security properties
- Algorithms

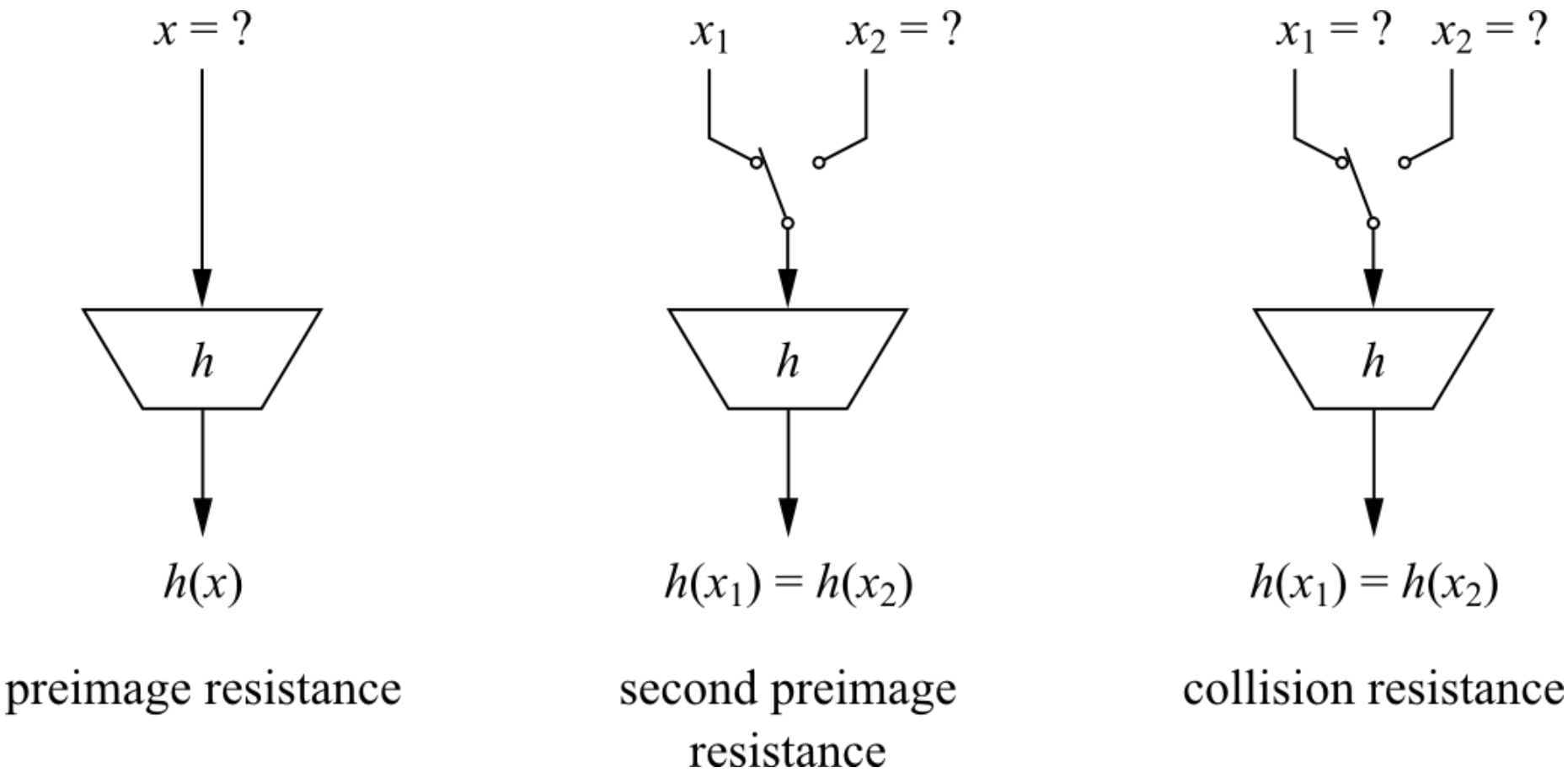
Principal input–output behavior of hash functions



Content of this Chapter

- Why we need hash functions
- How it works
- **Security properties**
- Algorithms

The three security properties of hash functions



Hash Functions: Security Properties

- **Preimage resistance** *(odolnost voci najdeniu vzoru):*
  For a given output z, it is computationally infeasible to find any input x such that h(x) = z, i.e., **h(x) is one-way** *(jednosmerna).*

- **Second preimage resistance** *(odolnost voci najdeniu druheho vzoru):*
  Given x_1, and thus $h(x_1)$, it is computationally infeasible to find any x_2 such that $h(x_1) = h(x_2)$.

- **Collision resistance** *(odolnost voci koliziam):*
  It is computationally infeasible to find any pairs $x_1 \neq x_2$ such that $h(x_1) = h(x_2)$.

Hash Functions: Security

It turns out that collison resistance causes most problems

- How hard is it to find a collision with a probability of 0.5 ?
- Related Problem: How many people are needed such that two of them have the same birthday with a probability of 0.5 ?
- No! Not 365/2=183. 23 are enough ! This is called **the birthday paradox** *(narodeninovy paradox)*.
- Generalization: If the output of a hash function has n bits, then a search for a collision takes $\approx\sqrt{2^n}$.
- To deal with this paradox, hash functions need an output size of at least 256 bits (the output size of 256 bits gives the security level of 128 bits).
- For more info see Chapter 11.2.3 in *Understanding Cryptography*.

Content of this Chapter

- Why we need hash functions
- How it works
- Security properties
- **Algorithms**

Popular hash functions used today

- SHA-2
- SHA-3

Lessons Learned: **Hash Functions**

- Hash functions are keyless.
- The three security requirements for hash functions are one-wayness, second preimage resistance and collision resistance.
- Hash functions should have at least 256-bit output length in order to withstand collision attacks.