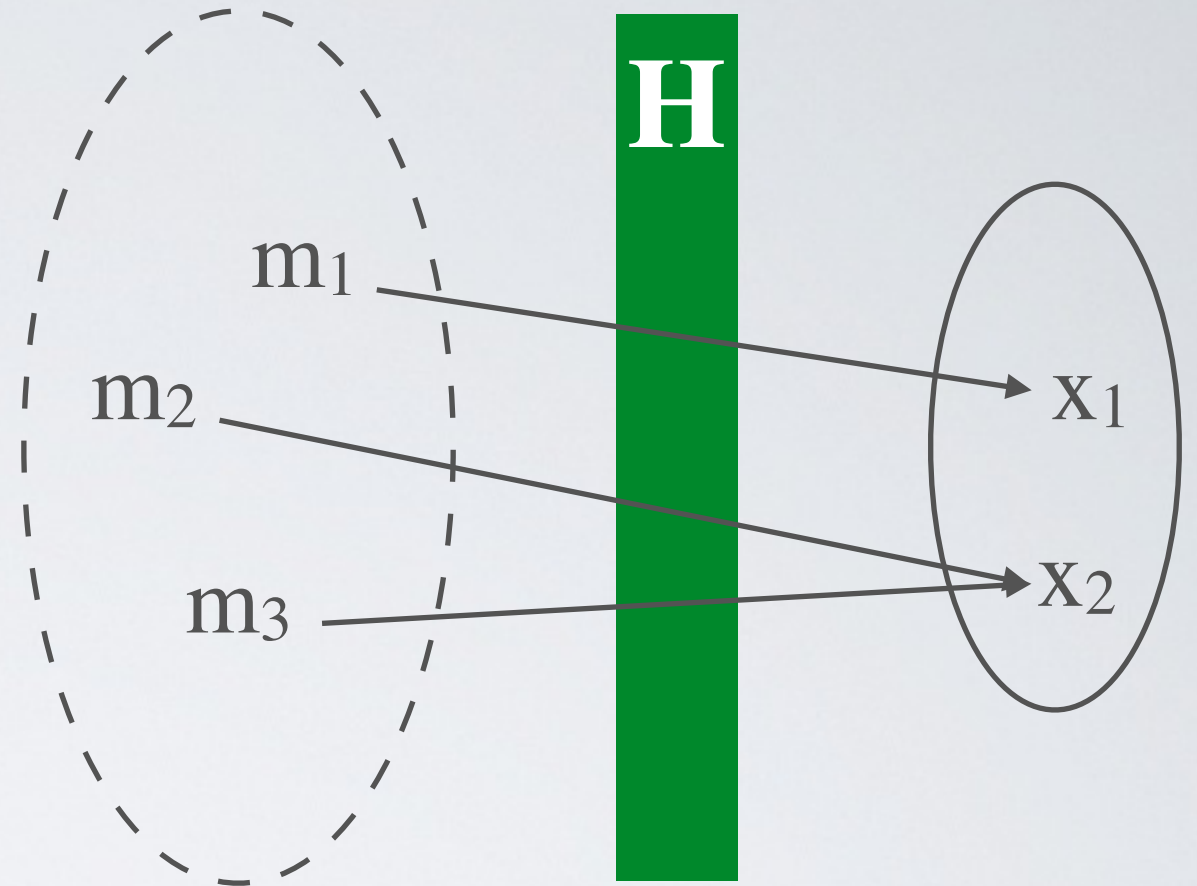


Cryptographic Hash Functions and Message Authentication Code

Thierry Sans

Cryptographic Hashing



$H(m) = x$ is a hash function if

- H is one-way function
- m is a message of any length
- x is a message digest of a fixed length

➡ H is a lossy compression function
necessarily there exists x, m_1 and $m_2 \mid H(m_1) = H(m_2) = x$

Computational Complexity



- Given H and m , computing x is **easy** (polynomial or linear)
- Given H and x , computing m is **hard** (exponential)

➔ H is **not invertible**

Preimage Resistance and Collision Resistance



PR - Preimage Resistance (a.k.a One Way)

- ➡ given H and x , hard to find m
e.g. password storage

2PR - Second Preimage Resistance (a.k.a Weak Collision Resistance)

- ➡ given H , m and x , hard to find m' such that $H(m) = H(m') = x$
e.g. virus identification

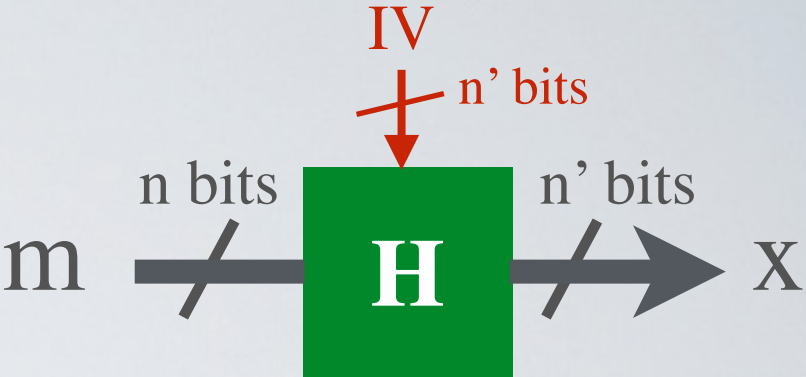
CR - Collision Resistance (a.k.a Strong Collision Resistance)

- ➡ given H , hard to find m and m' such that $H(m) = H(m') = x$
e.g. digital signatures

CR → 2PR and CR → PR

Hash Functions in Practice

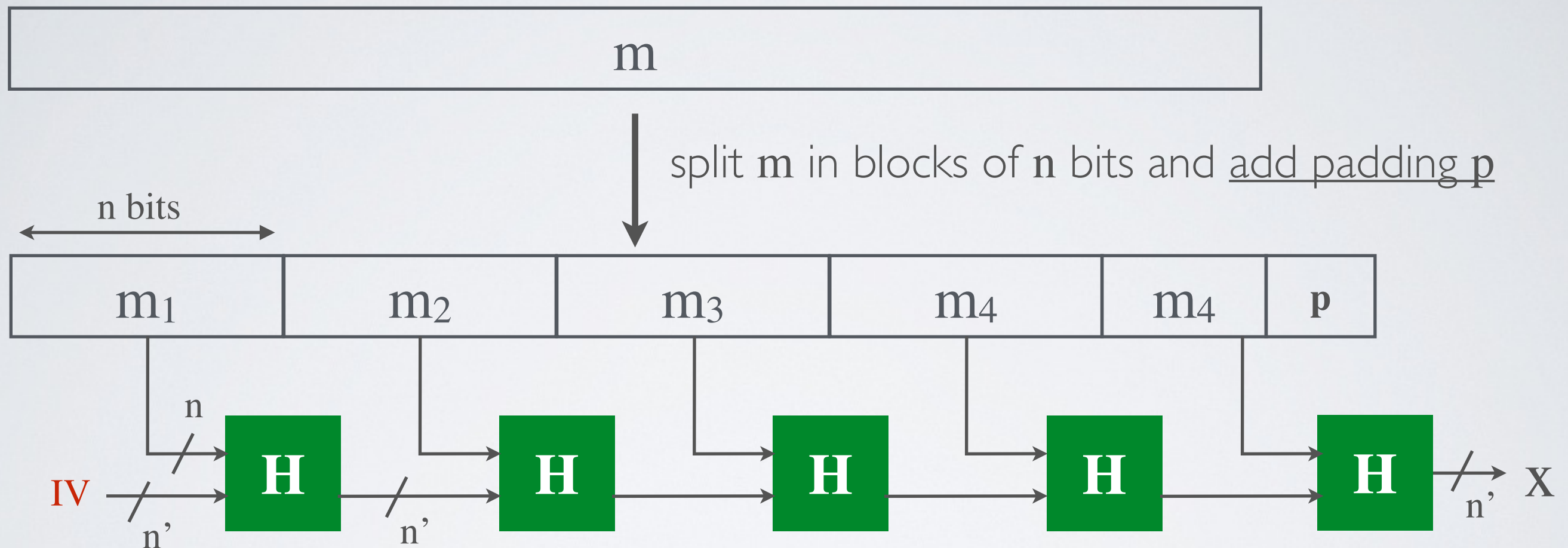
Common Hash Functions



Name	MD5	SHA-1	SHA-2				SHA-3 (Keccak)			
Variant			SHA-224	SHA-256	SHA-384	SHA-512	SHA3-224	SHA3-256	SHA3-384 4	SHA3-512
Year	1992	1993	2001				2012			
Designer	Rivest	NSA	NSA				Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche			
Input n bits	512	512	512	512	1024	1024	1152	1088	832	576
Output n' bits	128	160	224	256	384	512	224	256	384	512
Construction	Merkle–Damgård						Sponge			
Speed cycle/byte	6.8	11.4	15.8		17.7		12.5			
Considered Broken	yes	yes	no				no			

How to hash long messages ?

Merkle–Damgård construction (MD5, SHA-1 and SHA-2)

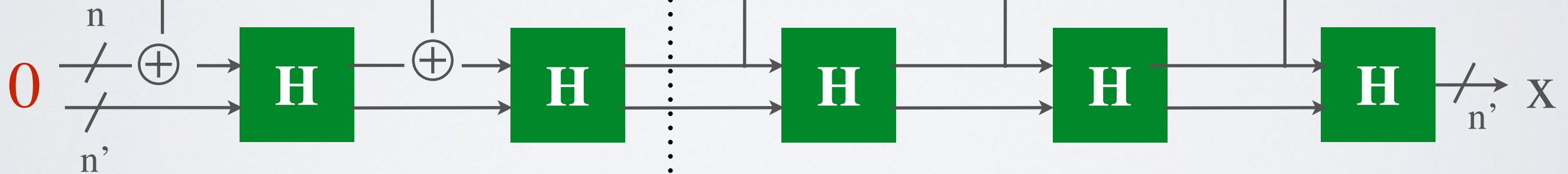
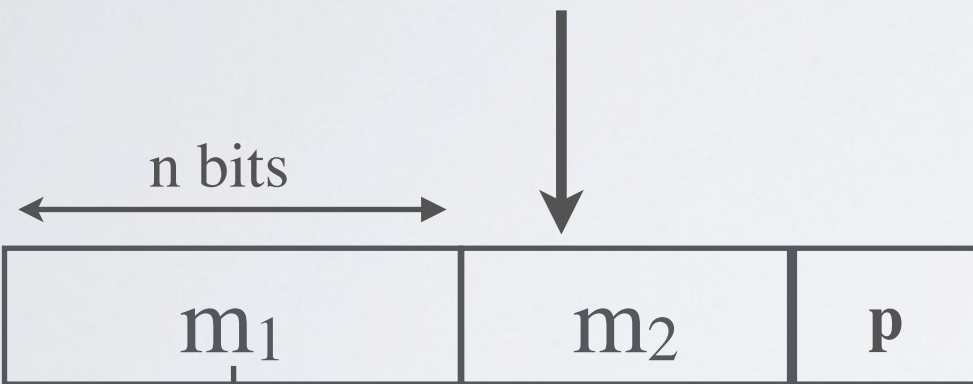


Property : if H is CR then Merkel-Damgard is CR

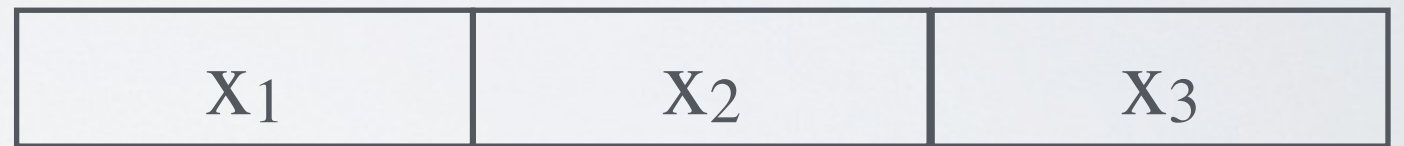
How to hash long messages ?

Sponge construction (SHA-3)

split m in blocks of n bits
and add padding p



assemble the hash



absorbing : *squeezing*

Property : if H is CR then Sponge is CR

Security of hash functions

Brute-forcing a hash function



CR - Collision Resistance

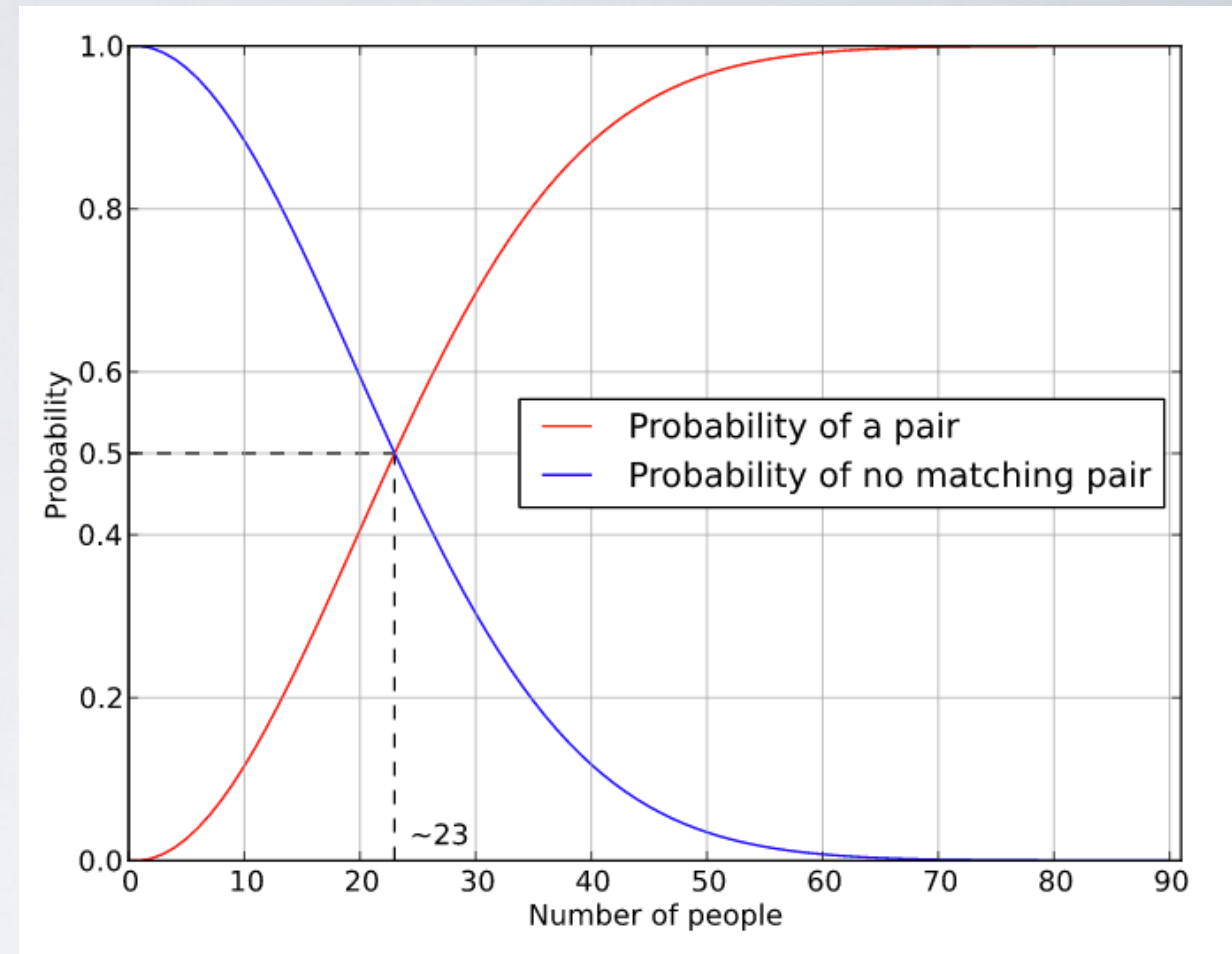
➔ given H , hard to find m and m' such that $H(m) = H(m') = x$

Given a hash function H of n bits output

- Reaching all possibilities 2^n cases
- ~~• On average, an attacker should try half of them 2^{n-1} cases~~

Birthday Paradox

“There are 50% chance that 2 people have the same birthday in a room of 23 people”



N-bits security

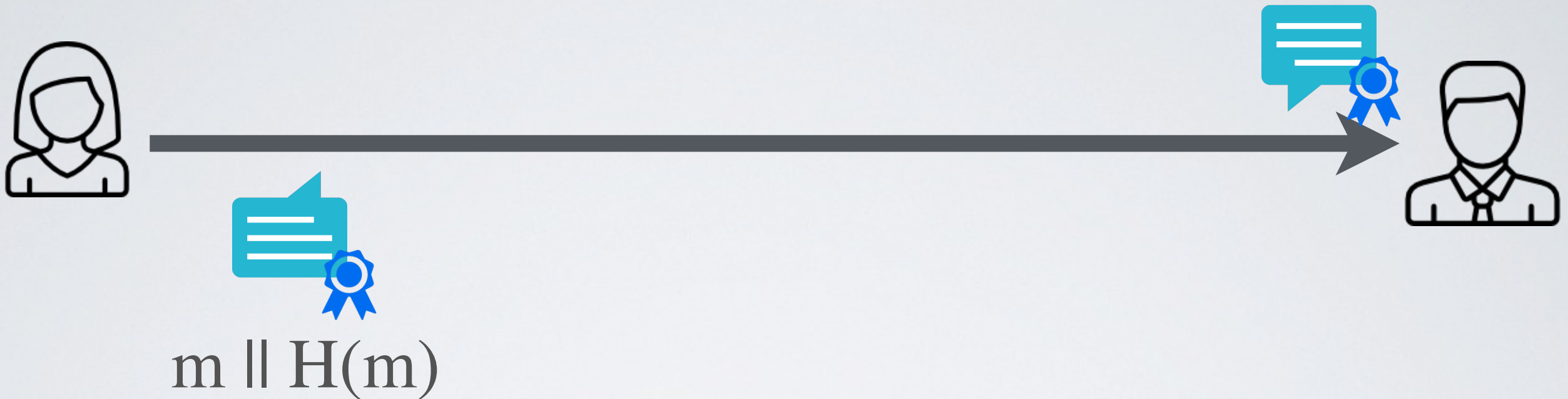
- ➡ Given a hash function **H** of **n** bits output, a collision can be found in around **$2^{n/2}$** evaluations
e.g SHA-256 is 128 bits security

Broken hash functions beyond the birthday paradox

	Year	Collision
MD5	2013	2^{24} evaluations (2^{39} with prefix)
SHA-1	2015	2^{57} evaluations

Using hash functions for Integrity

Hashing



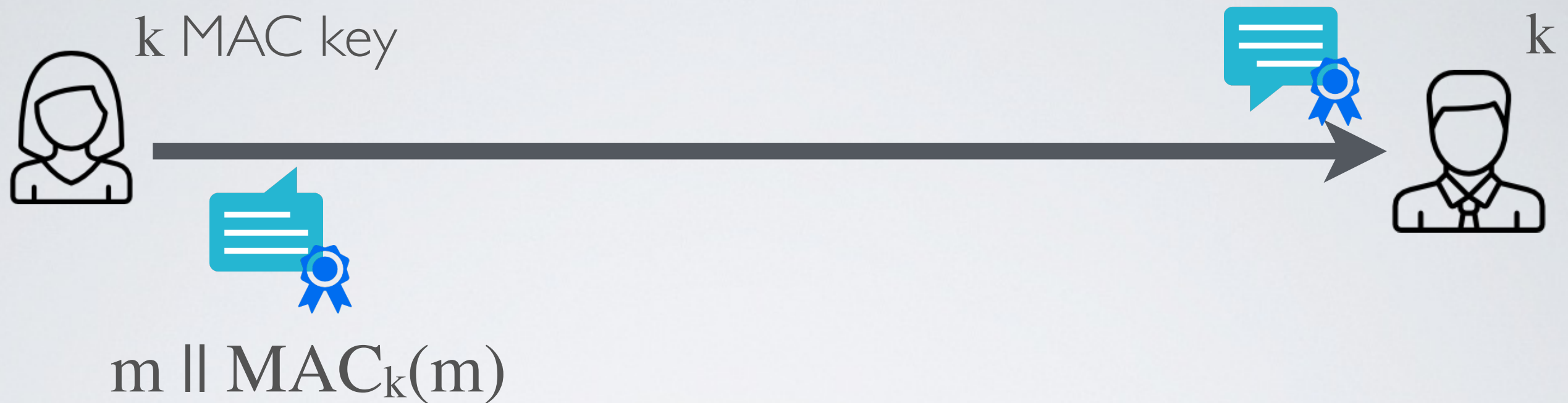
Apache HTTP Server 2.4.23 (httpd): 2.4.23 is the latest available version

The Apache HTTP Server Project is pleased to [announce](#) the release of version 2.4.23 of the Apache HTTP Server ("Apache" and "httpd"). This version of Apache is our latest GA release of the new generation 2.4.x branch of Apache HTTPD and represents fifteen years of innovation by the project, and is recommended over all previous releases!

For details see the [Official Announcement](#) and the [CHANGES_2.4](#) and [CHANGES_2.4.23](#) lists

- Source: [httpd-2.4.23.tar.bz2](http://httpd.apache.org/download.cgi#httpd-2.4.23) [[PGP](#)] [[MD5](#)] [[SHA1](#)]
- Source: [httpd-2.4.23.tar.gz](http://httpd.apache.org/download.cgi#httpd-2.4.23) [[PGP](#)] [[MD5](#)] [[SHA1](#)]

MAC - Message Authentication Code

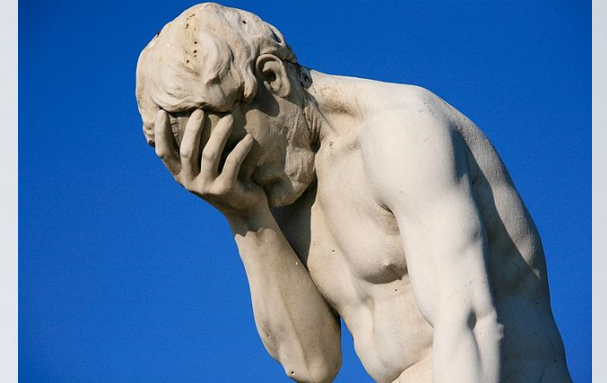


Alice and Bob share a key k

➔ HMAC - use a hash function on the message and the key

$$\text{MAC}_k(m) = H(k \parallel m)$$

Length Extension Attack



Vulnerable : All Merkle–Damgård-based hash functions
so MD5, SHA-1 and SHA-2 (but not SHA-3)

Flickr's API Signature Forgery Vulnerability

Thai Duong and Julian Rizzo

Date Published: Sep. 28, 2009

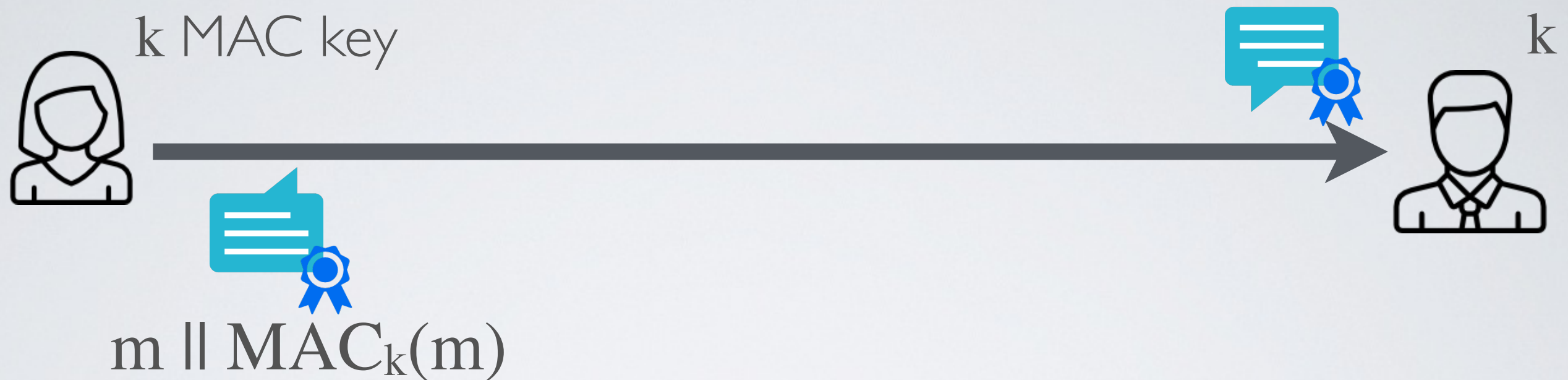
Advisory ID: MOCB-01

Advisory URL: http://netifera.com/research/flickr_api_signature_forgery.pdf

Title: Flickr's API Signature Forgery Vulnerability

Remotely Exploitable: Yes

Good HMAC



Alice and Bob share a key k

➡ Option 1 : envelope method

$$\text{MAC}_k(m) = H(k \parallel m \parallel k)$$

➡ Option 2 : padding method (i.e. HMAC standard)

$$\text{HMAC}_k(m) = H((k \oplus \text{opad}) \parallel H((k \oplus \text{ipad}) \parallel m))$$