

**ECRYPT II**  
www.ecrypt.eu.org


## Cryptographic Hash Functions: Theory and Practice

Bart Preneel  
Katholieke Universiteit Leuven - COSIC  
firstname.lastname@esat.kuleuven.be




### Hash functions

X.509 Annex D  
MDC-2  
MD2, MD4, MD5  
SHA-1




RIPEMD-160  
SHA-256  
SHA-512




**SHA-3**

*This is an input to a cryptographic hash function. The input is a very long string, that is reduced by the hash function to a string of fixed length. There are additional security conditions: it should be very hard to find an input hashing to a given value (a preimage) or to find two colliding inputs (a collision).*

h




1A3FD4128A198FB3CA345932



### Applications

- short unique identifier to a string
  - digital signatures
  - data authentication
- one-way function of a string
  - protection of passwords
  - micro-payments
- confirmation of knowledge/commitment
- pseudo-random string generation/key derivation
- entropy extraction
- construction of MAC algorithms, stream ciphers, block ciphers,...

2005: 800 uses of MD5 in Microsoft Windows



### Agenda


Definitions

Iterations (modes)

Compression functions

SHA-{0,1,2}

SHA-3 bits and bytes



### Hash function flavours

cryptographic hash function

MAC


MDC

OWHF

UOWHF  
(TCR)


CRHF

this talk



### Informal definitions

- no secret parameters
- input string  $x$  of arbitrary length  $\Rightarrow$  output  $h(x)$  of fixed bitlength  $n$
- computation "easy"
- One Way Hash Function (OWHF)
  - preimage resistance
  - 2<sup>nd</sup> preimage resistance
- Collision Resistant Hash Function (CRHF): OWHF +
  - collision resistant



### Security requirements (n-bit result)

preimage

$2^n$

2<sup>nd</sup> preimage

$x \neq ?$

$2^n$

collision

$? \neq ?$

$2^{n/2}$

### Preimage resistance

$2^n$

- in a password file, one does not store
  - (username, password)
- but
  - (username, hash(password))
- this is sufficient to verify a password
- an attacker with access to the password file has to find a preimage

### Second preimage resistance

2<sup>nd</sup> preimage

$x \neq ?$

$2^n$

$x$  → Channel 1: high capacity and insecure

$h(x)$  → Channel 2: low capacity but secure (= authenticated – cannot be modified)

- an attacker can modify  $x$  but not  $h(x)$
- he can only fool the recipient if he finds a second preimage of  $x$

### Collision resistance (1/2)

- hacker Alice prepares two versions of a software driver for the O/S company Bob
  - $x$  is correct code
  - $x'$  contains a backdoor that gives Alice access to the machine
- Alice submits  $x$  for inspection to Bob
- if Bob is satisfied, he digitally signs  $h(x)$  with his private key
- Alice now distributes  $x'$  to users of the O/S; these users verify the signature with Bob's public key
- this signature works for  $x$  and for  $x'$ , since  $h(x) = h(x')$ !

collision

$x \neq x'$

$2^{n/2}$

### Collision resistance (2/2)

- in many cryptographic protocols, Alice wants to commit to a value  $x$  without revealing it
- Alice picks a secret random string  $r$  and sends  $y = h(x || r)$  to Bob
- in a later phase of the protocol, Alice reveals  $x$  and  $r$  to Bob and he checks that  $y$  is correct
- if Alice can find a **collision**, that is  $(x,r)$  and  $(x',r)$  with  $x' \neq x$  she can cheat
- if Bob can find a **preimage**, he can learn  $x$  and cheat

collision

$x \neq x'$

$2^{n/2}$

### Brute force (2<sup>nd</sup>) preimage

- **multiple target second preimage (1 out of many):**
  - if one can attack  $2^t$  simultaneous targets, the effort to find a single preimage is  $2^{n-t}$
- **multiple target second preimage (many out of many):**
  - time-memory trade-off with  $\Theta(2^n)$  precomputation and storage  $\Theta(2^{2n/3})$  time per (2<sup>nd</sup>) preimage:  $\Theta(2^{2n/3})$  [Hellman'80]
- **answer: randomize hash function with a parameter S (salt, key, spice,...)**

### The birthday paradox

- given a set with  $S$  elements
- choose  $r$  elements at random (with replacements) with  $r \ll S$
- the probability  $p$  that there are at least 2 equal elements (a collision)  $\cong 1 - \exp(-r(r-1)/2S)$
- more precisely, it can be shown that
  - $p \geq 1 - \exp(-r(r-1)/2S)$
  - if  $r < \sqrt{2S}$  then  $p \geq 0.6 r(r-1)/2S$

13

### How to find collisions?

$I$  = space of pairs of messages;  
size  $\approx (2^{64})^2$

$C$  = space of all input messages that collide under  $h$   
 $|C| \approx 2^n |I|$

**Collision search algorithm 1**  
Pick  $2^n$  random message pairs  $(x, x')$   
For each pair,  $\text{Prob}(h(x)=h(x'))=2^{-n}$   
You expect to find a collision, that is, a non-empty intersection with  $C$

14

### How to find collisions?

$I$  = space of pairs of messages;  
size  $\approx (2^{64})^2$

$C$  = space of all input messages that collide under  $h$   
 $|C| \approx 2^n |I|$

**Collision search algorithm 2**  
Pick a set  $R$  of  $2^{n/2}$  random messages  
Find a collision  
You expect to find a collision, that is, a non-empty intersection with  $C$  as there are about  $2^{n/2}$  distinct pairs in  $R$

15

### Collision resistance

- hard to achieve in practice
  - many attacks
  - requires double output length  $2^{n/2}$  versus  $2^n$
- hard to achieve in theory
  - [Simon'98] one cannot derive collision resistance from "general" preimage resistance (there exists no black box reduction)
- hard to formalize: requires
  - family of functions: key, parameter, salt, spice,...
  - "human ignorance" trick [Stinson'06], [Rogaway'06]

16

### Relation between properties

[Rogaway-Shrimpton'04]  
[Stinson'06]  
[ReyhaniTabar-Susilo-Mu'10]  
[Andreeva-Stam'10]

Even if  $\text{Coll} \Rightarrow \text{xSEC/Pre}$ :  
bound always  $2^{n/2} \ll 2^n$

17



### Brute force attacks in practice

- ( $2^{\text{nd}}$ ) preimage search
  - $n = 128$ : 23 B\$ for 1 year if one can attack  $2^{40}$  targets in parallel
- parallel collision search: small memory using cycle finding algorithms (distinguished points)
  - $n = 128$ : 1 M\$ for 8 hours (or 1 year on 100K PCs)
  - $n = 160$ : 90 M\$ for 1 year
  - need 256-bit result for long term security (30 years or more)

18

### Quantum computers

- in principle exponential parallelism
- inverting a one-way function:  $2^n$  reduced to  $2^{n/2}$  [Grover'96]
- collision search:
  - $2^{n/3}$  computation + hardware [Brassard-Hoyer-Tapp'98]
  - [Bernstein'09] classical collision search requires  $2^{n/4}$  computation and hardware (= standard cost of  $2^{n/2}$ )

19

### Properties in practice

- collision resistance is not always necessary
- other properties are needed:
  - PRF: pseudo-randomness if keyed (with secret key)
  - PRO: pseudo-random oracle property
  - near-collision resistance
  - partial preimage resistance (most of input known)
  - multiplication freeness
- how to formalize these requirements and the relation between them?

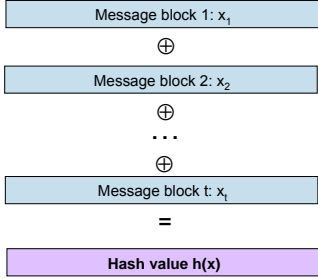
20

## Iteration (mode of compression function)

21

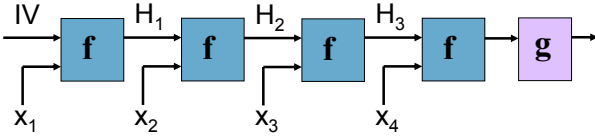
### How not to construct a hash function

- Divide the message into  $t$  blocks  $x_i$  of  $n$  bits each



22

### Hash function: iterated structure



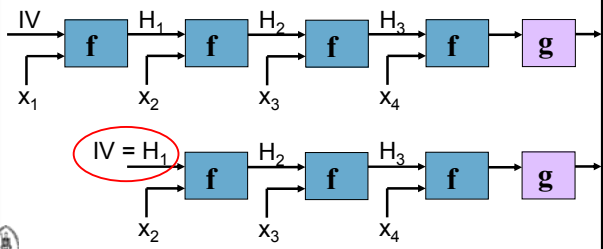
Split messages into blocks of fixed length and hash them block by block with a compression function  $f$

Efficient and elegant  
But ...

23

### Security relation between $f$ and $h$

- iterating  $f$  can degrade its security
  - trivial example:  $2^{\text{nd}}$  preimage



24

### Security relation between f and h (2)

- solution: Merkle-Damgård (MD) strengthening
  - fix IV, use unambiguous padding and insert length at the end
- f is collision resistant  $\Rightarrow$  h is collision resistant [Merkle'89-Damgård'89]
- f is ideally 2<sup>nd</sup> preimage resistant  $\Leftrightarrow$  h is ideally 2<sup>nd</sup> preimage resistant [Lai-Massey'92]
  - few hash functions have a strong compression function
  - very few hash functions treat  $x_i$  and  $H_{i-1}$  in the same way

25

### Security relation between f and h (3)

length extension: if one knows  $h(x)$ , easy to compute  $h(x || y)$  without knowing  $x$  or IV

solution: output transformation

26

### Property preservation

[Andreeva-Mennink-P'10] for overview

Sec/Pre preservation seems to be problematic  
Is Pre preservation meaningful?

	Coll	Sec	Pre	Pro	aSec	eSec	aPre	ePre
Suffix- & Prefix-free MD	Green	Red	Red	Green				
Envelope MD	Green	Red	Red	Green				
BCM	Green	Green	Red	?				
Haifa	Green	Red	Red	Green				
RMX	Green	Red	Red	Red				
Shoup UOWH	Green	Red	Red	Red	Red	Green	Red	Green
ROX	Green	Green	Green	Red	Green	Green	Green	Green

Not applicable

28

### More on property preservation/domain extension

- PRO preservation  $\Rightarrow$  Col, Sec and Pre for ideal compression function
  - but for narrow pipe bounds for Sec and Pre are at most  $2^{n/2}$  rather than  $2^n$
- [...]

28

### Attacks on MD-type iterations

- multi-collision attack and impact on concatenation [Joux'04]
- long message 2<sup>nd</sup> preimage attack [Dean-Felten-Hu'99], [Kelsey-Schneier'05]
  - Sec security degrades linearly with number  $2^l$  of message blocks hashed:  $2^{n+l} + t 2^{n/2+l}$
  - appending the length does not help here!
- herding attack [Kelsey-Kohno'06]
  - reduces security of commitment using a hash function from  $2^n$
  - on-line  $2^{n-1}$  + precomputation  $2.2^{(n+1)/2}$  + storage  $2^l$

29

### How (NOT) to strengthen a hash function?

[Joux'04]

- answer: concatenation
- $h_1$  ( $n_1$ -bit result) and  $h_2$  ( $n_2$ -bit result)

- intuition: the strength of  $g$  against collision/(2<sup>nd</sup>) preimage attacks is the product of the strength of  $h_1$  and  $h_2$ 
  - if both are “independent”
- but....

30

### Multiple collisions $\neq$ multi-collision

Assume "ideal" hash function  $h$  with  $n$ -bit result

- $\Theta(2^{n/2})$  evaluations of  $h$  (or steps): 1 collision  
–  $h(x)=h(x')$
- $\Theta(r \cdot 2^{n/2})$  steps:  $r^2$  collisions  
–  $h(x_1)=h(x'_1)$ ;  $h(x_2)=h(x'_2)$ ; ...;  $h(x_{r^2})=h(x'_{r^2})$
- $\Theta(2^{2n/3})$  steps: a 3-collision  
–  $h(x)=h(x')=h(x'')$
- $\Theta(2^{n(t-1)/t})$  steps: a  $t$ -fold collision (multi-collision)  
–  $h(x_1)=h(x_2)=\dots=h(x_t)$

31

### Multi-collisions on iterated hash function (2)

- for IV: collision for block 1:  $x_1, x'_1$
- for  $H_1$ : collision for block 2:  $x_2, x'_2$
- for  $H_2$ : collision for block 3:  $x_3, x'_3$
- for  $H_3$ : collision for block 4:  $x_4, x'_4$

• now  $h(x_1||x_2||x_3||x_4) = h(x'_1||x_2||x_3||x_4) = h(x'_1||x'_2||x_3||x_4) = \dots = h(x'_1||x'_2||x'_3||x'_4)$  **a 16-fold collision (time: 4 collisions)**

32

### Multi-collisions [Joux '04]

- finding multi-collisions for an iterated hash function is not much harder than finding a single collision (if the size of the internal memory is  $n$  bits)
- algorithm
  - generate  $R = 2^{n/2}$ -fold multi-collision for  $h_2$
  - in  $R$ : search by brute force for  $h_1$
- Time:  $n1 \cdot 2^{n2/2} + 2^{n1/2} \ll 2^{(n1+n2)/2}$

$g(x) = h_1(x) || h_2(x)$

33

### Multi-collisions [Joux '04]

consider  $h_1$  ( $n_1$ -bit result) and  $h_2$  ( $n_2$ -bit result), with  $n_1 \geq n_2$ .  
concatenation of 2 iterated hash functions ( $g(x) = h_1(x) || h_2(x)$ ) is **as most as strong as the strongest** of the two (even if both are independent)

- cost of collision attack against  $g$  at most  
 $n_1 \cdot 2^{n2/2} + 2^{n1/2} \ll 2^{(n1+n2)/2}$
- cost of (2nd) preimage attack against  $g$  at most  
 $n_1 \cdot 2^{n2/2} + 2^{n1} + 2^{n2} \ll 2^{n1+n2}$
- if either of the functions is weak, the attacks may work better

34

### Summary

35

### Improving MD iteration

salt + output transformation + counter + wide pipe

security reductions well understood  
many more results on property preservation  
impact of theory limited

36

### Improving MD iteration

- degradation with use: salting (family of functions, randomization)
  - or should a salt be part of the input?
- PRO: strong output transformation  $g$ 
  - also solves length extension
- long message  $2^{\text{nd}}$  preimage: preclude fix points
  - counter  $f \rightarrow f_i$  [Biham-Dunkelman'07]
- multi-collisions, herding: avoid breakdown at  $2^{n/2}$  with larger internal memory: known as wide pipe
  - e.g., extended MD4, RIPEMD, [Lucks'05]

37

# Compression functions

38

### Block cipher ( $E_k$ ) based

Davies-Meyer

Miyaguchi-Preneel

- output length = block length
- 12 secure compression functions (in ideal cipher model)
- requires 1 key schedule per encryption
- analysis [Black-Rogaway-Shrimpton'02], [Duo-Li'06], [Stam'09],...

39

### Permutation ( $\pi$ ) based: sponge

Examples: Panama, RadioGatun, Grindahl, Keccak (no buffer)

40

### Permutation ( $\pi$ ) based

#### small permutation

JH

Grøstl

41

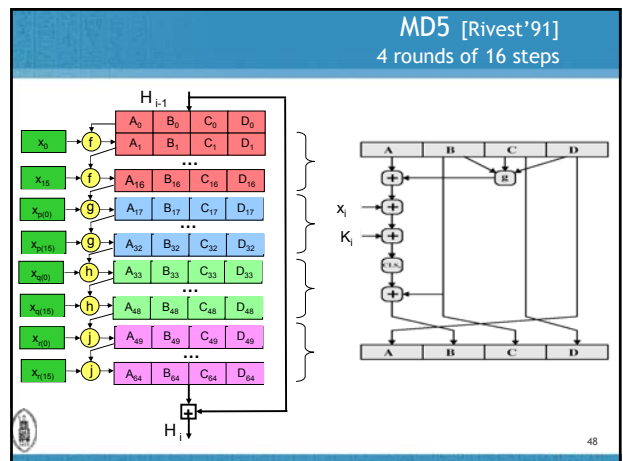
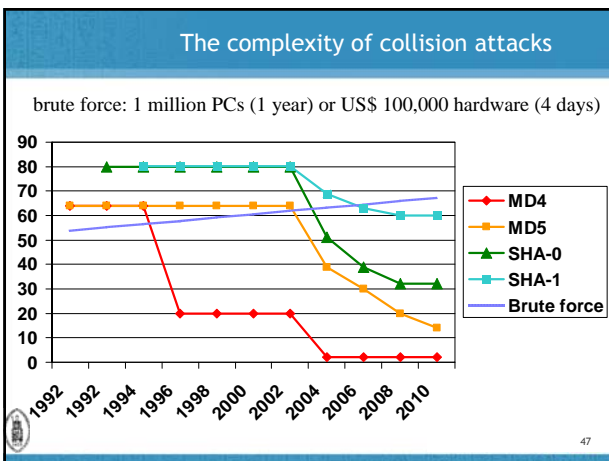
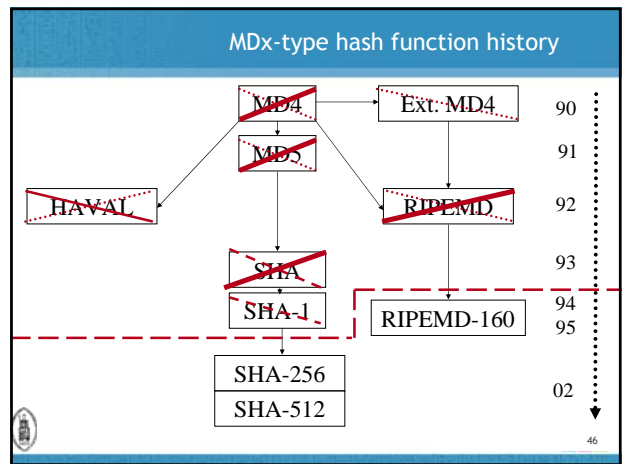
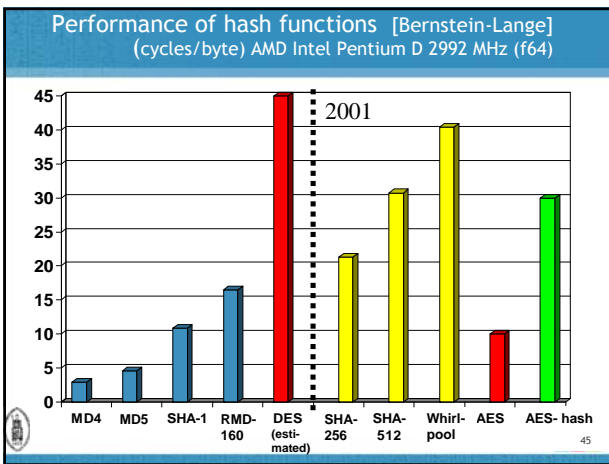
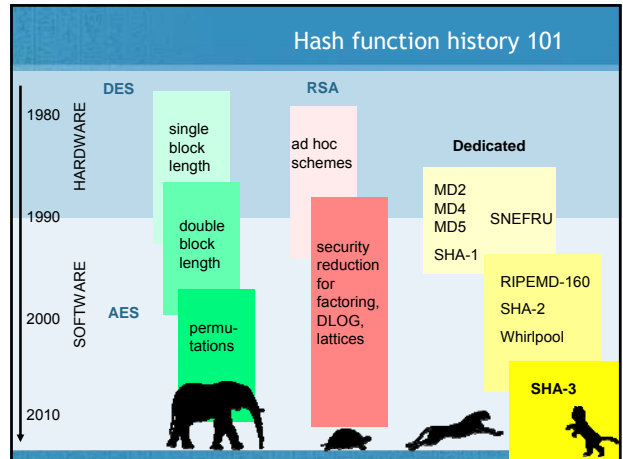
### Iteration modes and compression functions

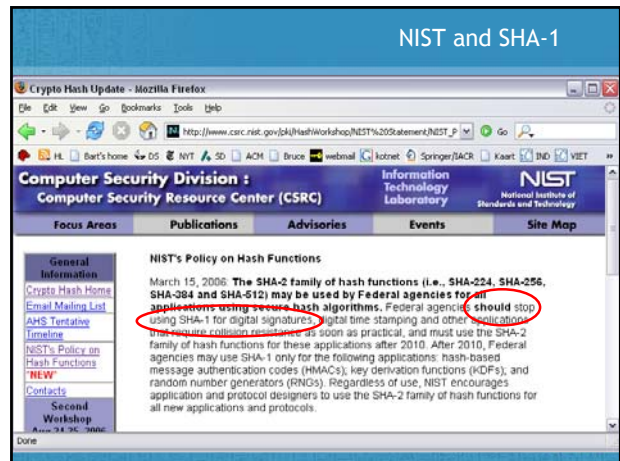
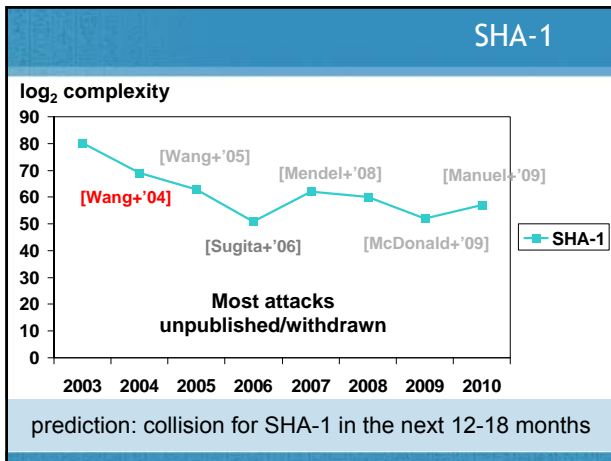
- security of simple modes well understood
- powerful tools available
- analysis of slightly more complex schemes very difficult
- which properties are meaningful?
- which properties are preserved?
- MD versus sponge is still open debate

42

# SHA-{0,1,2}

43





### Rogue CA attack

[Sotirov-Stevens-Appelbaum-Lenstra-Molnar-Osvik-de Weger '08]

- request user cert; by special collision this results in a fake CA cert (need to predict serial number + validity period)

**impact: rogue CA that can issue certs that are trusted by all browsers**

- 6 CAs have issued certificates signed with MD5 in 2008:
  - Rapid SSL, Free SSL (free trial certificates offered by RapidSSL), TC TrustCenter AG, RSA Data Security, Verisign.co.jp

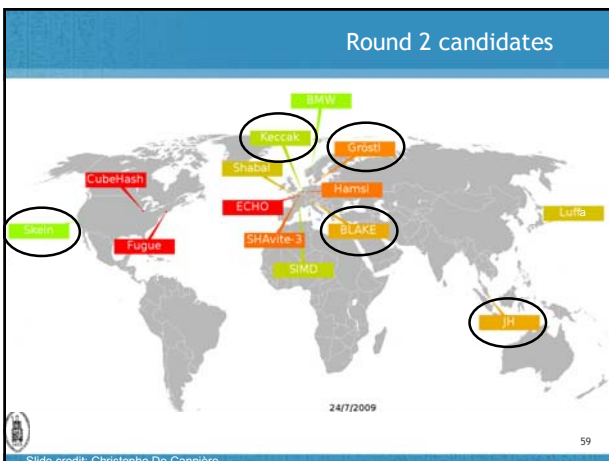
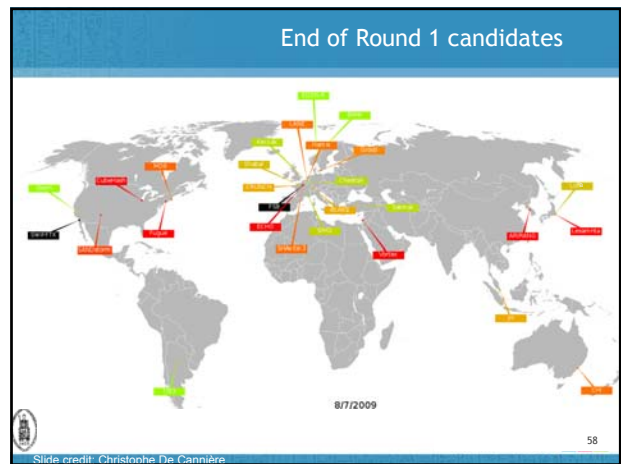
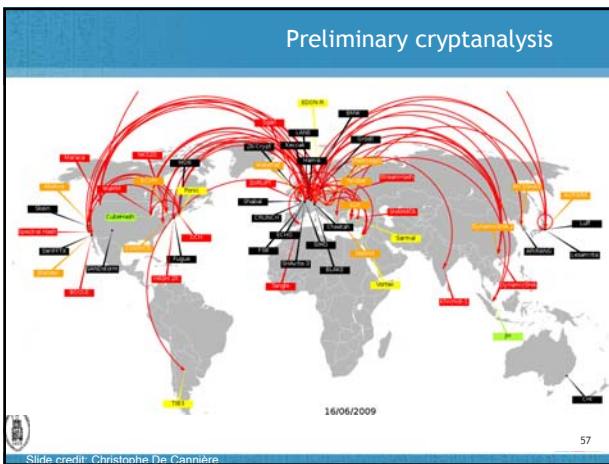
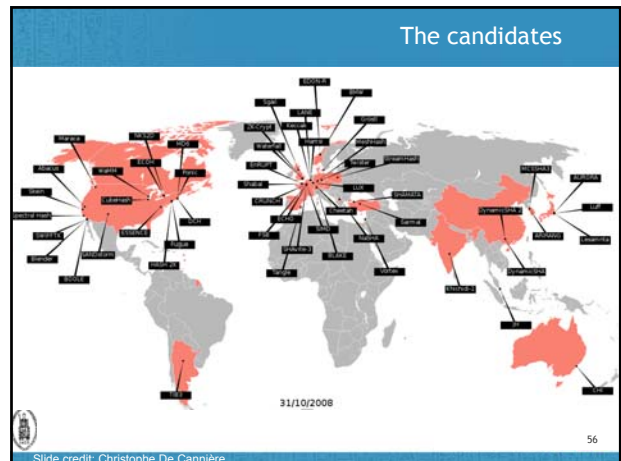
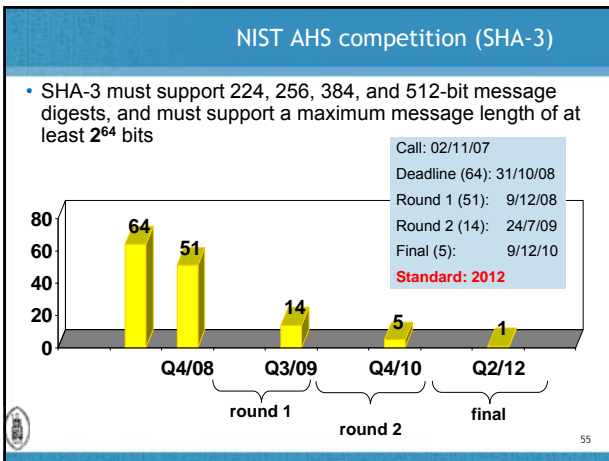
- ### Upgrades
- RIPEND-160 is good replacement for SHA-1
  - upgrading algorithms is always hard
  - TLS uses MD5 || SHA-1 to protect algorithm negotiation (up to v1.1)
  - upgrading negotiation algorithm is even harder: need to upgrade TLS 1.1 to TLS 1.2**
- 52

- ### SHA-2 [NIST'02]
- SHA-224, SHA-256, SHA-384, SHA-512
    - non-linear message expansion
    - more complex operations
    - 64/80 steps
    - SHA-384 and SHA-512: 64-bit architectures
  - SHA-256 collisions: 24/64 steps [Sanadhya-Sarkar'08]
  - SHA-256 preimages: 43/64 steps [Aoki+'09]
  - implementations today faster than anticipated
  - adoption
    - industry may migrate to SHA-2 by 2011 or may wait for SHA-3
    - very slow for TLS/IPsec (no pressing need)
- 53

# SHA-3

(bits and bytes)

54

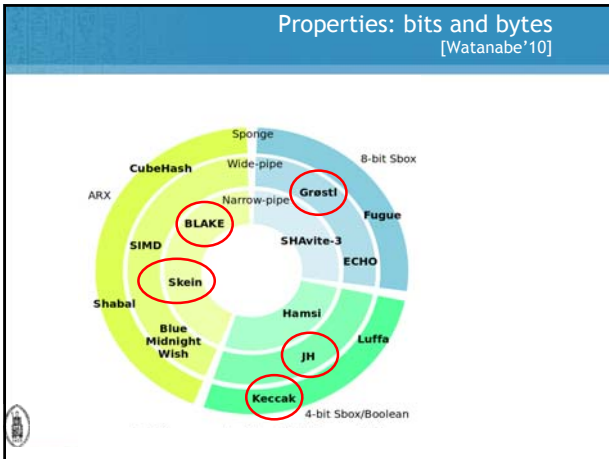


### Compression function/iteration

	Block cipher	Permutation	MD/HAIFA
Blake			HAIFA
Groستl		2-permutation	MD
JH			JH-specific
Keccak		Sponge	
Skein	MMO		MD*/Tree (UBI)
BMW	PGV variant		MD
Cubehash		Sponge-type	
ECHO			HAIFA
Fugue		Spong-type	
Hamsi			
Luffa		Sponge-type	
Shabal		Sponge-type	
Shavite-3	Davies-Meyer		HAIFA
SIMD	PGV variant		MD

Slide credit: Christophe De Cannière

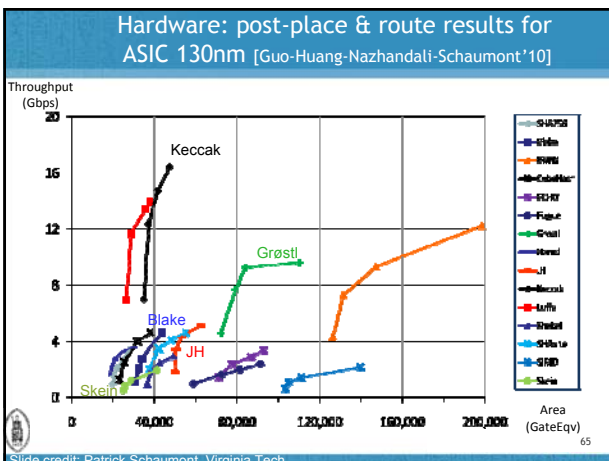
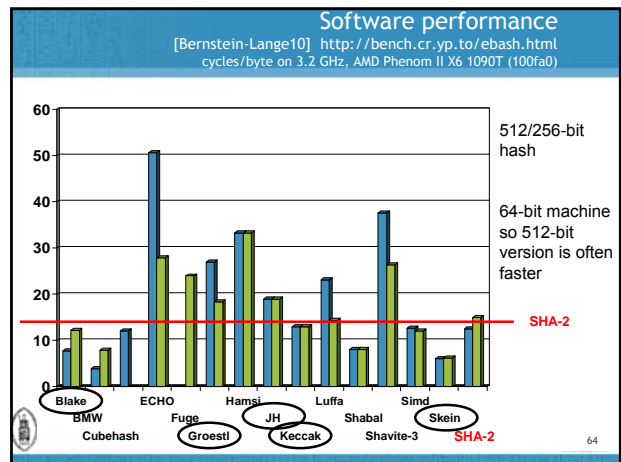
60



### Security reductions [Andreeva-Mennink-P'10]

	type	of	Adv <sup>1</sup>	Adv <sup>2</sup>	Adv <sup>3</sup>	Adv <sup>4</sup>	Adv <sup>5</sup>	Adv <sup>6</sup>	Adv <sup>7</sup>	Adv <sup>8</sup>
BLAKE	HAIFA	✓	✓							
BMW	chop-(MD+FT)	✓	✗							
CubeHash	chop-(MD+FT)	✓	✗							
ECHO	chop-HAIFA	✓	✓							
Fugue	chop-(MD+FT)	✓	✗							
Grostl	hop-(MD+FT)	✓	✗							
Hamsi	MD+FT	✓	✗							
JH	chop-MD	✓	✗							
Keccak	chop-MD	✓	✗							
Luffa	chop-(MD+FT)	✓	✗							
Shabal	chop-MD	✓	✓							
SHAvite-3	HAIFA	✓	✓							
SIMD	chop-(MD+FT)	✓	✗							
Skein	MD	✓	✓							

**Table 1.** A schematic summary of all results. The first column describes the hash function construction, and the second and third columns show which hash functions have a suffix-free (sf) or prefix-free (pf) padding. A green box indicates the existence of a non-trivial upper bound, a red box means that an efficient adversary is known for the security notion, and a yellow box indicates that no result is known, but recent literature gives some confidence in the existence of a non-trivial bound.



- ### Issues arisen during Round 1
- round 1 was very short; several functions received no outside analysis
  - security
    - some controversy on complexity and relevance of attacks
    - proofs have not helped much to survive
  - performance
    - weak performance resulted in elimination
  - 7/14 designs tweaked at the beginning of round 2

## Issues arisen during Round 2

- security
  - few real attacks but some weaknesses
  - new design ideas harder to validate
- performance: roughly as fast or faster than SHA-2
  - SHA-2 gets faster every day
  - widely different results for hardware and software
    - software: large difference between high end and embedded
    - hardware: FGPA and ASIC
  - what about lightweight devices and 128-core machines?
- diversity = third selection criterion
- expect more tweaks before final
- variable number of rounds?
- NIST expects that SHA-2 and SHA-3 will co-exist



67

## Final

- Blake
- JH
- Grøstl
- Keccak
- Skein



68

## SHA-4?

- an open competition such as SHA-3 is bound to result in new insights between 2008-2012
- only few of these can be incorporated using “tweaks”
- the winner selected in 2012 will reflect the state of the art in October 2008
- nevertheless, it is unlikely that we will have a SHA-4 competition before 2030



69

## Hash functions: conclusions

- SHA-1 would have needed 128-160 steps instead of 80
- 2004-2009 attacks: cryptographic meltdown but not dramatic for most applications
  - clear warning: upgrade asap
- half-life of a hash function is < 1 year
- theory is developing for more robust iteration modes and extra features; still early for building blocks
- nirwana: efficient hash functions with security reductions



70