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

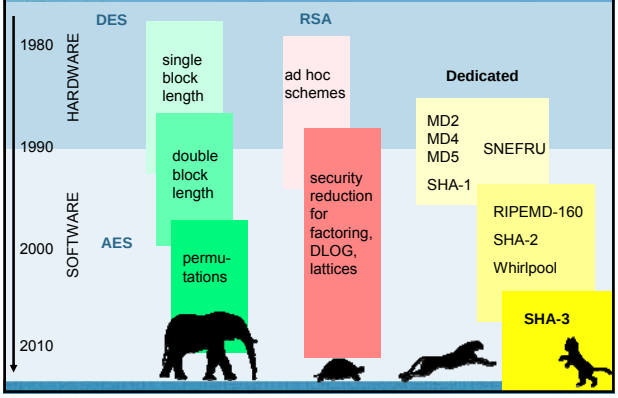
SHA-3

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*

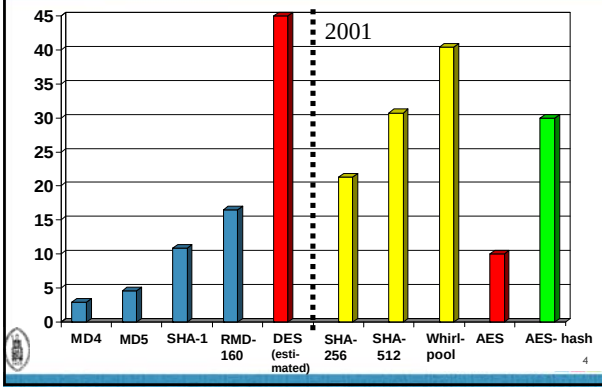
2

Hash function history 101



Performance of hash functions [Bernstein-Lange]

(cycles/byte) AMD Intel Pentium D 2992 MHz (f64)



4

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

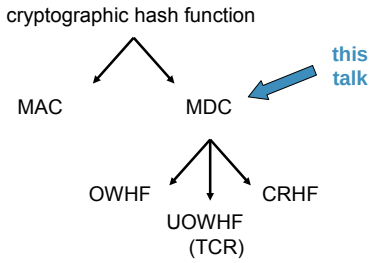
5

Agenda

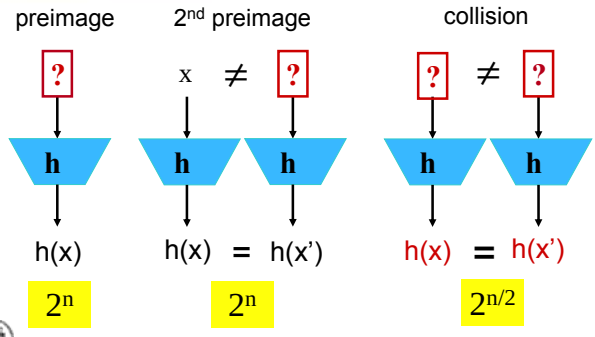
- Definitions
- Iterations (modes)
- Compression functions
- SHA-{0,1,2}
- SHA-3 bits and bytes

6

Hash function flavours



Security requirements (n-bit result)



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
  - 2nd preimage resistance
- Collision Resistant Hash Function (CRHF): OWHF +
  - collision resistant

Formal definition: (2nd) preimage resistance

Notation: $\Sigma = \{0,1\}$, $l(n)>n$

A **one-way hash function (OWFH)** h is a function with domain $D=\Sigma^{l(n)}$ and range $R=\Sigma^n$ that satisfies the following conditions:

- **preimage resistance**: let x be selected uniformly in D and let M be an adversary that on input $h(x)$ uses time $\leq t$ and outputs $M(h(x)) \in D$. For each adversary M,
  $Pr_{x \in D}\{ h(M(h(x)))=h(x) \} < \varepsilon$
  Here the prob. is also taken over the random choices of M.
- **2nd preimage resistance**: let x be selected uniformly in $D=\Sigma^{l(n)}$ and let M' be an adversary who on input x uses time $\leq t$ and outputs $x' \in D$ with $x' \neq x$. For each adversary M',
  $Pr_{x \in D}\{ h(M'(x))=h(x) \} < \varepsilon$
  Here the prob. is taken over the random choices of M'.

Formal definitions: collision resistance

A **collision-resistant hash function (CRHF)** h is a function with domain $D=\Sigma^{l(n)}$ and range $R=\Sigma^n$ that that satisfies the following conditions:

- for each collision string finder F that uses time $\leq t$ and outputs either "?" or a pair $x, x' \in \Sigma^{l(n)}$ with $x' \neq x$ such that $h(x')=h(x)$ *we have that*
  $Pr\{ F(\mathbf{H}) \neq$ "?" $\} < \varepsilon$
  Here the prob. is taken over the random choices of F.

Is this correct?

No! For every h there are many (short) colliding strings, and so there exist many very simple collision string finders F that output a collision with probability 1

Formal definitions: collision resistance

A **collision-resistant hash function family (CRHF)** $\boldsymbol{H}$ is a function family $\{h_S\}$ with domain $D=\Sigma^{l(n)}$ and range $R=\Sigma^n$ and indexed by a "key" S that that satisfies the following conditions:

- collision resistance: let F be a collision string finder that on input $S \in \Sigma^s$ uses time $\leq t$ and outputs either "?" or a pair $x, x' \in \Sigma^{l(n)}$ with $x' \neq x$ such that $h_S(x')=h_S(x)$. For each F,
  $Pr_S\{ F(\mathbf{H}) \neq$ "?" $\} < \varepsilon$
  Here the prob. is also taken over the random choices of F.

Formal definitions - continued

- for collision resistance: formalization requires a family, but see [Stinson'06], [Rogaway'06]: "formalizing human ignorance"
- for (2nd) preimage resistance, one can choose the challenge (x) and/or the key (S) that selects the function. This gives three flavors [Rogaway-Shrimpton'04]:
  - random challenge, random key (Pre and Sec)
  - random key, fixed challenge (ePre and eSec - everywhere) (eSec=UOWHF)
  - fixed key, random challenge (aPre and aSec - always)
- complex relationship (see figure on next slide)

Relation between properties

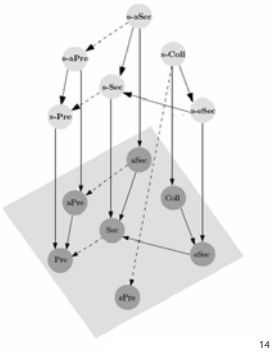
[Rogaway-Shrimpton'04]

[Stinson'06]

[Reyhanitabar-Susilo-Mu'10]

[Andreeva-Stam'10]

Even if Coll ⇒ xSEC/Pre: bound always $2^{n/2} << 2^n$



Brute force (2^{nd}) 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^{nd}) preimage: $\Theta(2^{2n/3})$ [Hellman'80]
  - full cost per (2^{nd}) preimage from $\Theta(2^n)$ to $\Theta(2^{2n/5})$ [Wiener'02] (if $\Theta(2^{3n/5})$ targets are attacked)
- answer: randomize hash function: key, parameter, salt, spice,…

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 $2^{n/2}$)

Brute force attacks in practice

- (2^{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)

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 bypass**
  - UOWHF (TCR, eSec) randomize hash function after choosing the message [Naor-Yung'89]
    - how to enforce this in practice? (insider attacks i.e. attacks by the signer)
  - randomized hashing: RMX mode [Halevi-Krawczyk'05]
    $H(r \| x_1 \oplus r \| x_2 \oplus r \| \dots \| x_t \oplus r)$
    - needs e-SPR (not met by MD5)
    - issues with insider attacks
    - [Gauravaram-Knudsen09]: $2^{n/2}$ complexity but on-line

Properties in practice

- collision resistance is not always necessary
- other properties are needed:
  - pseudo-randomness if keyed (with secret key)
  - 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?

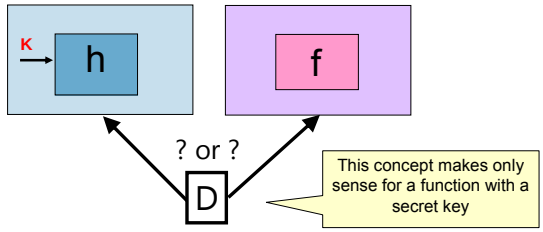
19

Pseudo-random function

computationally indistinguishable from a random function

$Adv_h^{prf} = Pr [K \xleftarrow{\$} \mathcal{K}: A^{h_K(.)} \Rightarrow 1] - Pr [f \xleftarrow{\$} RAND(m,n): A^f \Rightarrow 1]$

RAND(m,n): set of all functions from m-bit to n-bit strings

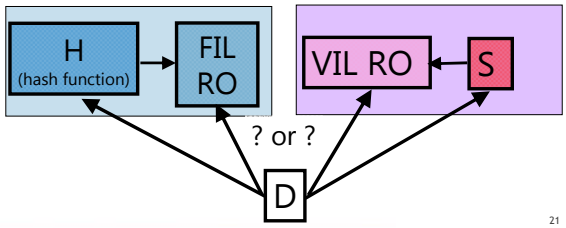


20

Indifferentiability from a random oracle or PRO property [Maurer+04]

variant of indistinguishability appropriate when distinguisher has access to inner component (e.g. building block of a hash function)

∃ Simulator S, ∀ distinguisher D, $Adv^{PRO}(H,S)$ is small

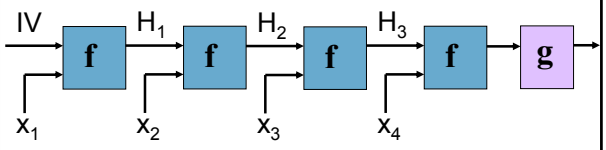


21

Iteration
(mode of compression function)

22

Hash function: iterated structure



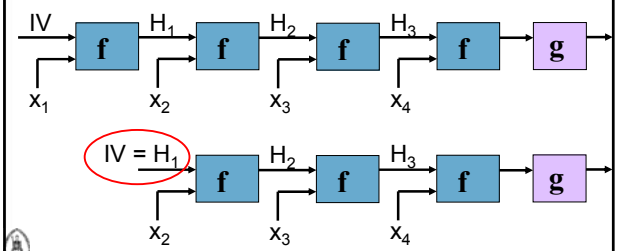
Split messages into blocks of fixed length and hash them block by block with a compression function f

Efficient and elegant
But …

23

Security relation between f and h

- iterating f can degrade its security
  - trivial example: 2nd preimage



24

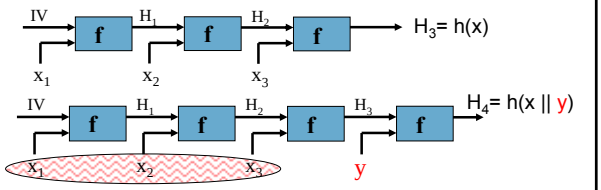
Security relation between f and h (2)

- solution: Merkle-Damgård (MD) strengthening
  - fix IV, use unambiguous padding and insert length at the end
- f is collision resistant ⇒ h is collision resistant
  [Merkle'89-Damgård'89]
- f is ideally 2nd preimage resistant ⇎ h is ideally 2nd preimage resistant [Lai-Massey'92]

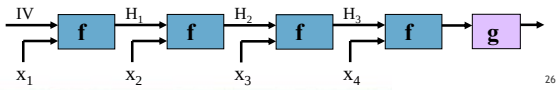
- few hash functions have a strong compression function
- very few hash functions treat x_i and H_{i-1} in the same way

Security relation between f and h (3)

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



solution: output transformation



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					Not applicable			
Envelope MD								
BCM				?				
Haifa								
RMX								
Shoup UOWH								
ROX								

More on property preservation/domain extension

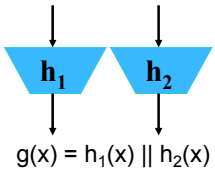
- PRO preservation ⇒ Col, Sec and Pre for ideal compression function
  - but for narrow pipe bounds for Sec and Pre are at most $2^{n/2}$ rather than 2^n
- MD + f = Preimage Aware (PRA) ⇒ PRO
  [Dodis-Ristenpart-Shrimpton'09]
  - Property "in between" CR (collision resistance) and PRO
- MD + split padding + f = csPre (chosen suffix preimage)
  ⇒ Pre (similar for Sec) [Yasuda'08]
- MD with envelope method h(K || x || K)
  ⇒ pseudo-randomness/MAC [Bellare-Canetti-Krawczyk'96]
  - but some problems and HMAC is a better construction

Attacks on MD-type iterations

- multi-collision attack and impact on concatenation [Joux'04]
- long message 2nd preimage attack
  [Dean-Felten-Hu'99], [Kelsey-Schneier'05]
  - Sec security degrades linearly with number 2^t of message blocks hashed: $2^{n-t+1} + t\,2^{n/2+1}$
  - 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-t} + precomputation $2.2^{(n+t)/2}$ + storage 2^t
- recent work [Andreeva-Bouillaget-Dunkelman-Kelsey'09],[Andreeva+09]
  - long message 2nd preimage and herding extended to MD variants (e.g., dither hash, zipper hash, ROX, Shoup hash…)
  - Trojan message attack

How (NOT) to strengthen a hash function?
[Joux'04]

- answer: concatenation
- h_1 (n1-bit result) and h_2 (n2-bit result)
- intuition: the strength of g against collision/(2nd) preimage attacks is the product of the strength of h_1 and h_2
  — if both are "independent"
- but….



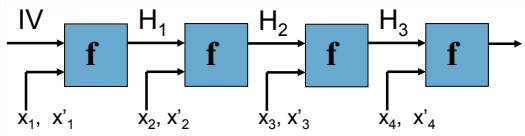
Multiple collisions ≠ multi-collision

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

- $\Theta(2^{n/2})$ evaluations of h (or steps): 1 collision
  - $h(x)=h(x')$
- $\Theta(r.\ 2^{n/2})$ 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)
  - $h(x_1)= h(x_2)= \ldots =h(x_t)$

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 $h(x_1||x_2||x_3||x_4) = h(x'_1||x_2||x_3||x_4) = h(x'_1||x'_2||x_3||x_4) = \ldots$ $= h(x'_1||x'_2||x'_3||x'_4)$ **a 16-fold collision**

32

Multi-collisions [Joux '04]

consider h_1 (n1-bit result) and h_2 (n2-bit result), with n1 ≥ n2.

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
  $n1 \cdot 2^{n2/2} + 2^{n1/2} << 2^{(n1+n2)/2}$
- cost of (2nd) preimage attack against g at most
  $n1 \cdot 2^{n2/2} + 2^{n1} + 2^{n2} << 2^{n1+n2}$
- if either of the functions is weak, the attacks may work better.
- main observation: finding multiple 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)

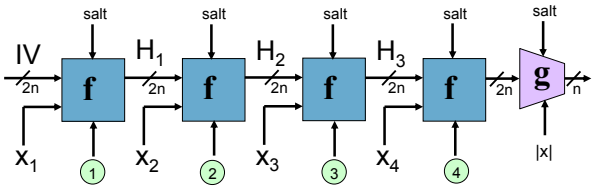
33

Summary

34

Improving MD iteration

salt + output transformation + counter + wide pipe



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

35

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 2nd 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]

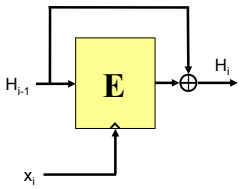
36

Compression functions

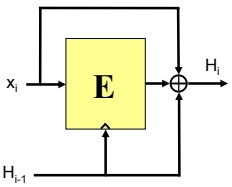
37

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],…

38

Block cipher (E_K) based

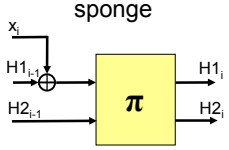
- which assumptions are needed on the block cipher E to prove MD iterated Davies-Meyer secure?
  - standard model: no security results (PRF/PRP is not sufficient)
  - ideal cipher model: ok to prove collision resistance and (second) preimage resistance
    - can this be relaxed?
  - PRO preserving with chop or NMAC/HMAC
  - PRA preserving

39

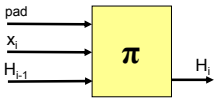
Permutation (π) based

Large permutation

sponge

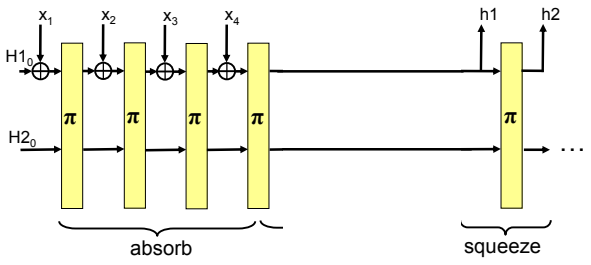


MD6



40

Permutation (π) based: sponge



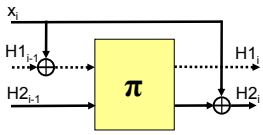
Examples: Panama, RadioGatun, Grindahl, Keccak (no buffer)

41

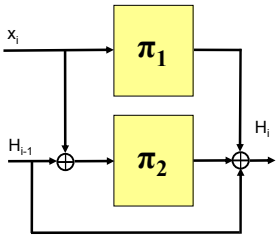
Permutation (π) based

small permutation

JH



Grøstl



42

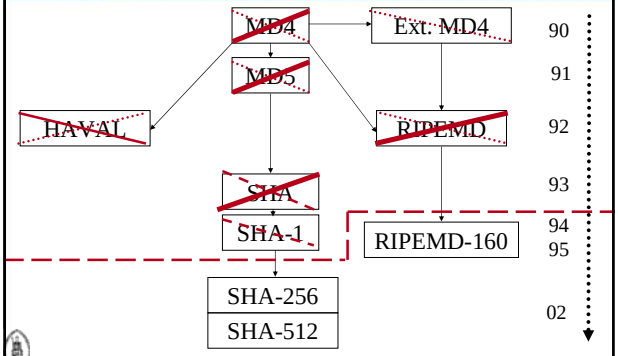
Iteration modes

- 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

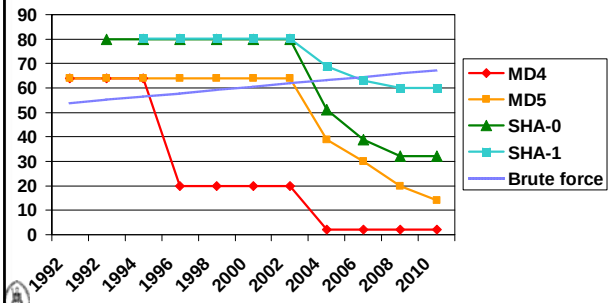
SHA-{0,1,2}

MDx-type hash function history

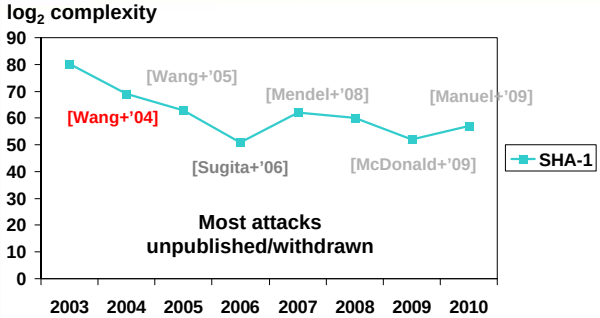


The complexity of collision attacks

brute force: 1 million PCs (1 year) or US$ 100,000 hardware (4 days)

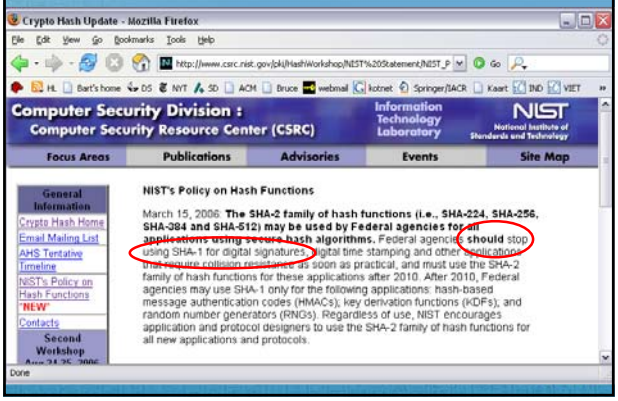


SHA-1



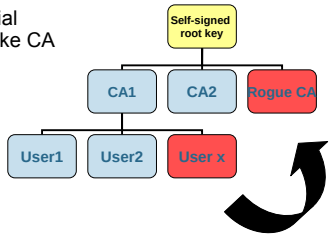
prediction: collision for SHA-1 in the next 12-18 months

NIST and SHA-1



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

- RIPEMD-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**

50

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)

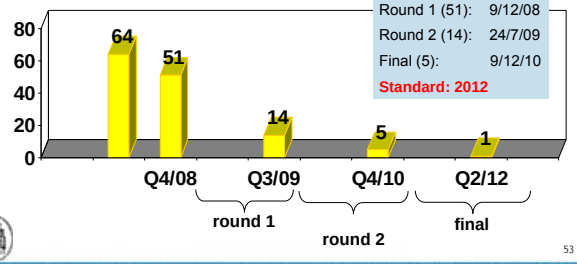
51

SHA-3
(bits and bytes)

52

NIST AHS competition (SHA-3)

- SHA-3 must support 224, 256, 384, and 512-bit message digests, and must support a maximum message length of at least 2^{64} bits



53

The candidates

Slide credit: Christophe De Cannière

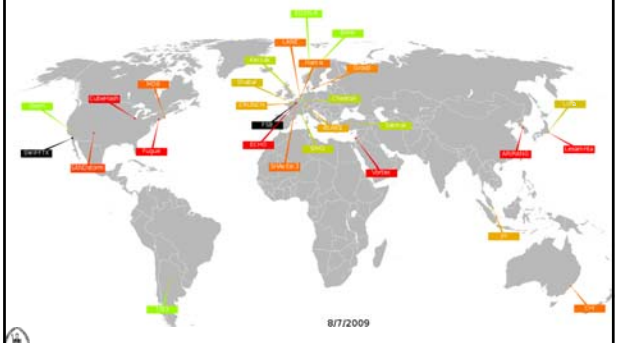
54

Preliminary cryptanalysis

Slide credit: Christophe De Cannière

55

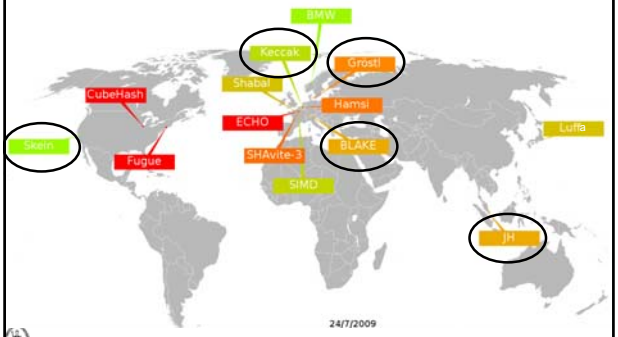
End of Round 1 candidates



Slide credit: Christophe De Cannière

56

Round 2 candidates



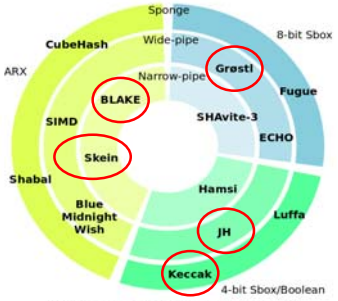
Slide credit: Christophe De Cannière

57

Compression function/iteration

	Block cipher	Permutation	MD/HAIFA
Blake			HAIFA
Grøstl		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

Properties: bits and bytes
[Watanabe'10]



Security reductions
[Andreeva-Mennink-P'10]

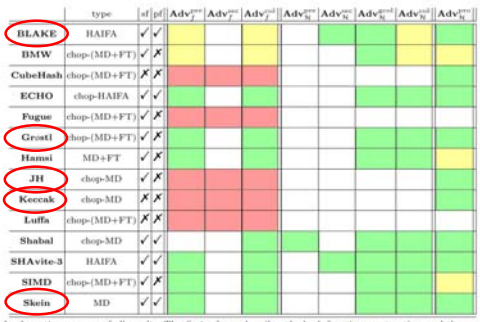
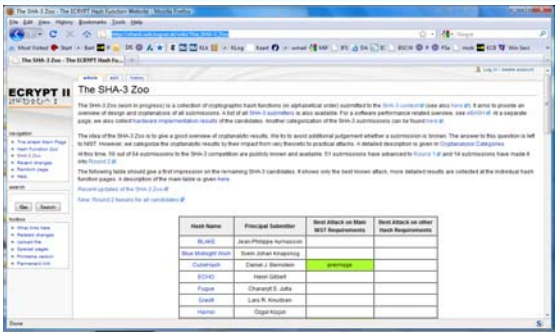


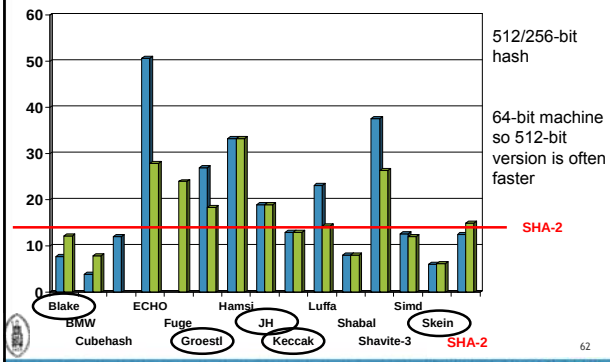
Table 1. A schematic summary of all results. The *first* column describes the hash function construction, and the *second* and *third* column show which hash functions have a suffix-free (sf) or prefix-free (pf) padding. A green box indicates the existence of a non-trivial upper bound, a red box means 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.

Security: SHA-3 Zoo
http://ehash.iaik.tugraz.at/wiki/The_SHA-3_Zoo



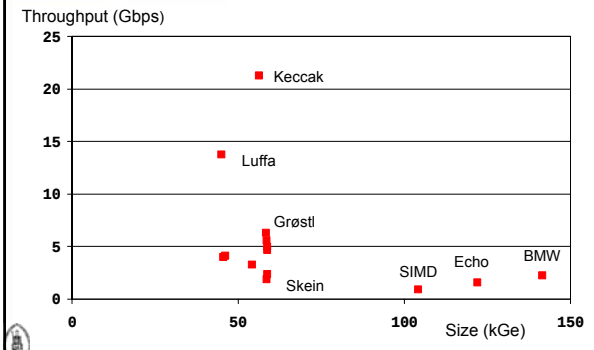
61

Software performance
[Bernstein-Lange10] http://bench.cr.yp.to/ebash.html
cycles/byte on 3.2 GHz, AMD Phenom II X6 1090T (100fa0)



62

Hardware performance
[Tillich+'09] IACR ePrint 2009/510



63

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

64

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

65

Final

- Blake
- JH
- Grøstl
- Keccak
- Skein

66

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

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