Hash function design and cryptanalysis: basic topics

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

Ice Break 2013

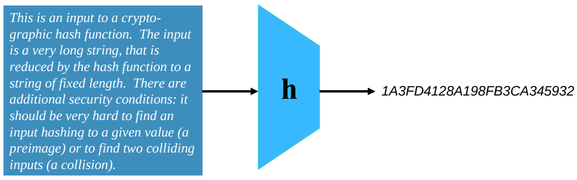
June 2013

Hash functions

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

RIPEMD-160
SHA-256
SHA-512

SHA-3



Applications

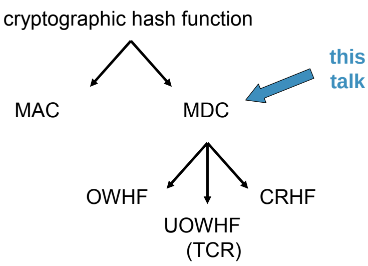
- short unique identifier to a string
  - digital signatures
  - data authentication
- one-way function of a string
  - protection of passwords
  - micro-payments
- confirmation of knowledge/commitment
- pseudo-random string generation/key derivation
- entropy extraction
- construction of MAC algorithms,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

2005: 800 uses of MD5 in Microsoft Windows

Agenda

- Definitions
- Iterations (modes)
- Compression functions
- Constructions
- SHA-3
- Conclusions

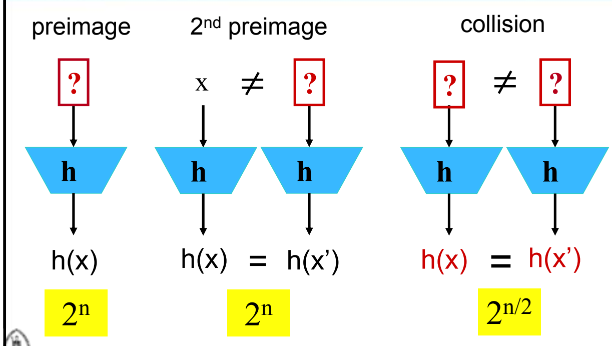
Hash function flavours



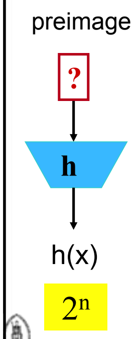
Informal definitions

- no secret parameters
- input string *x* of arbitrary length ⇒ 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

Security requirements (n-bit result)

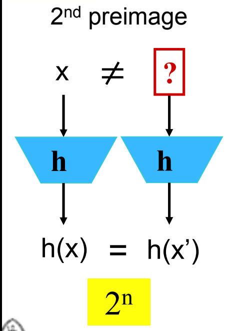


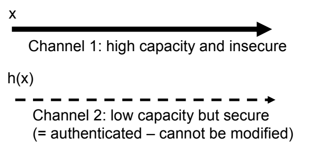
Preimage resistance



- 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

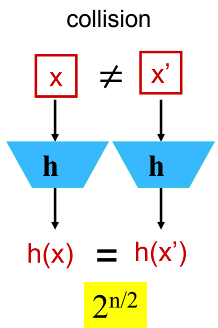




- an attacker can modify x but not h(x)
- he can only fool the recipient if he finds a second preimage of x

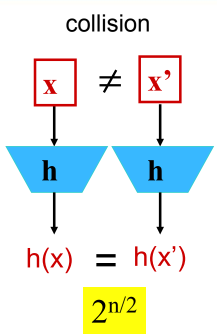
Collision resistance (1/2)

- 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 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' ≠ x she can cheat
- if Bob can find a preimage, he can learn x and cheat

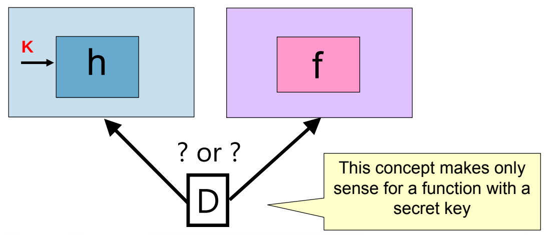


Pseudo-random function

computationally indistinguishable from a random function

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

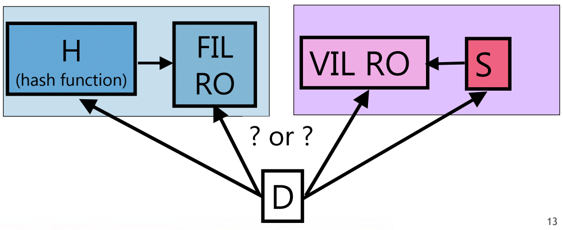
RAND(m,n): set of all functions from m-bit to n-bit strings



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



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]
- answer: randomize hash function with a parameter S (salt, key, spice,…)

The birthday paradox

how many people r do I need to have in a room to have a probability of p=50% to have at least 2 people with the same birthday?

answer: 23

what is the probability that the birthdays of r people are distinct?

$q = 1 - p = \underbrace{1 \,.\, 364/365 \,.\, 363/365 \,.\, 362/365 \ldots (365-(r-1))/365}_{r \text{ terms}}$

$q = 1-p \approx 0.5$ for $r = 23$

intuition: number of distinct pairs of people is 23.22/2 = 253; each pair has probability 1/365 to have the same birthday

exercise: how many people do you need in a room to have a probability of 0.50 to have 3 people with the same birthday?

The birthday paradox (2)

- given a set with S elements
- choose r elements at random (with replacements) with r « 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 $p \geq 0.6\, r(r-1)/2S$

⇒ for a hash function with an n-bit result, a collision can be found in time $2^{n/2}$ and memory $2^{n/2}$

- the number of collisions follows a Poisson distribution with $\lambda = r(r-1)/2S$
  - the expected number of collisions is equal to λ
  - the probability to have c collision is $e^{-\lambda} \lambda^c / c!$

The birthday paradox - proof

$q = 1-p = \underbrace{1 \,.\, ((S-1)/S) \,.\, ((S-2)/S) \ldots ((S-(r-1))/S)}_{r \text{ terms}}$

or $q = \Pi_{k=1}^{r-1} (S-k/S)$

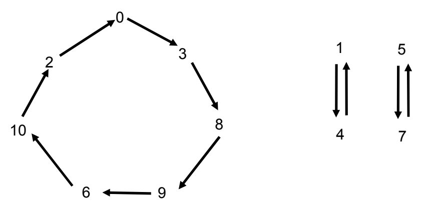
$\ln q = \Sigma_{k=1}^{r-1} \ln(1-k/S) \cong \Sigma_{k=1}^{r-1} -k/S = -r(r-1)/2S$

Taylor: if x « 1: $\ln(1-x) \cong x$

summation: $\Sigma_{k=1}^{r-1} k = r(r-1)/2$

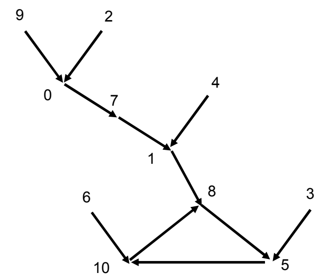
hence $p = 1 - q = 1 - \exp(-r(r-1)/2S)$

Functional graph of $f(x) = x^3 + 3 \bmod 11$



Exercise: find the functional graph of $f(x) = x^3 + 7 \bmod 11$

Functional graph of $f(x) = x^2 + 7 \bmod 11$



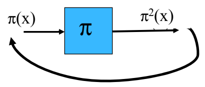
Done!

- Exercise: why is the indegree of 5 nodes equal to 0 resp. 2?

19

Functional graph of a permutation π

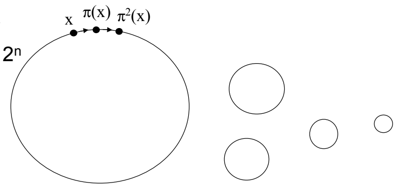
permutation π



every permutation of a finite set can be written as a product of disjoint cycles

expected length of largest cycle: $0.62\ 2^n$

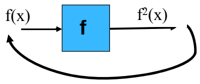
expected number of cycles of length at most $m \approx \ln m$



20

Functional graph of a random function f

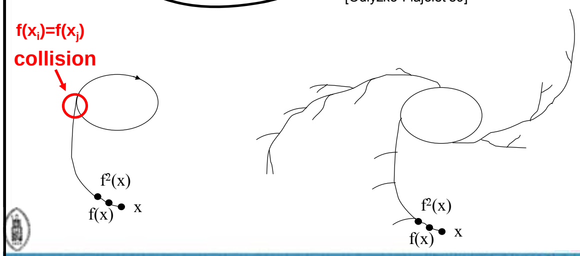
random function f



Expected length of largest cycle: $(\pi/8)\ 2^{n/2}$

Expected length from a point to the cycle: $(\pi/8)\ 2^{n/2}$

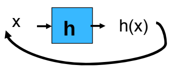
[Odlyzko-Flajolet'89]



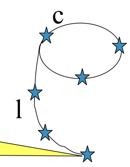
21

Brute force collision search

- low memory and parallel implementation of the birthday attack
  [Pollard'78][Quisquater'89][Wiener-van Oorschot'94]



- distinguished point (d bits)
  - $\Theta(e 2^{n/2} + e\ 2^{d+1})$ steps with e the cost of one function evaluation
  - $\Theta(n 2^{n/2-d})$ memory
  - full cost: $\Theta(e\ n 2^{n/2})$



$l = c = (\pi/8)\ 2^{n/2}$

M. Wiener: The Full Cost of Cryptanalytic Attacks, J. of Cryptology, 2002

22

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]

23

23

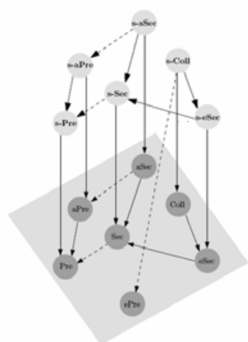
Relation between properties

[Rogaway-Shrimpton'04]

[Stinson'06]

[Reyhanitabar-Susilo-Mu'10]

[Andreeva-Stam'10]



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

24

Brute force attacks in practice

- (2nd) 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)

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 $2^{n/2}$)


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 (indifferentiable from a random oracle) – but see [Ristenpart-Shacham-Shrimpton'11]
  - near-collision resistance
  - partial preimage resistance (most of input known)
  - multiplication freeness
- how to formalize these requirements and the relation between them?

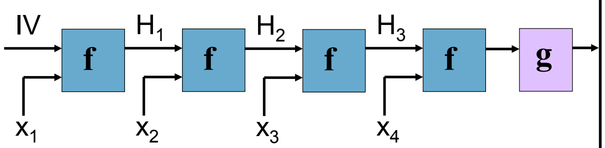
Iteration
(mode of compression function)

How not to construct a hash function

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

Message block 1: x_1
⊕
Message block 2: x_2
⊕
…
⊕
Message block t: x_t
=
Hash value h(x)

Hash function: iterated structure

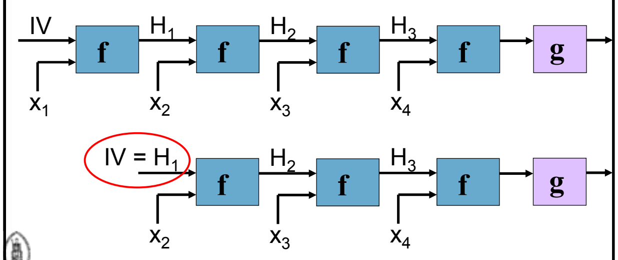


- split messages into blocks of fixed length and hash them block by block with a compression function f
- need padding at the end

efficient and elegant…. but …

Security relation between f and h

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



31

Security relation between f and h (2)

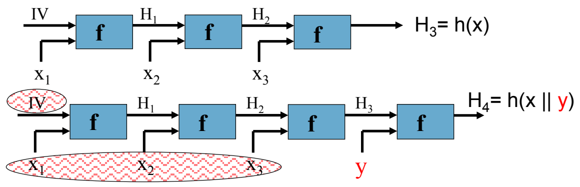
- solution: Merkle-Damgård (MD) strengthening
  - fix IV, use unambiguous padding and insert length at the end
- f is collision resistant ⇒ h is collision resistant [Merkle'89-Damgård'89]
- f is ideally 2nd preimage resistant ⇐? h is ideally 2nd preimage resistant [Lai-Massey'92]

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

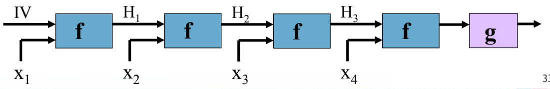
32

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



33

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

many more results

34

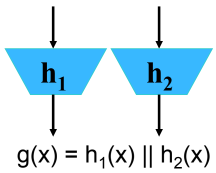
Attacks on MD-type iterations

- **long message 2nd preimage attack** [Dean-Felten-Hu'99], [Kelsey-Schneier'05]
  - Sec security degrades lineary with number 2^t of message **blocks** hashed: $2^{n-t+1} + t\ 2^{n/2+1}$
  - appending the length does not help here!
- **multi-collision attack and impact on concatenation** [Joux'04]
- **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

35

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….

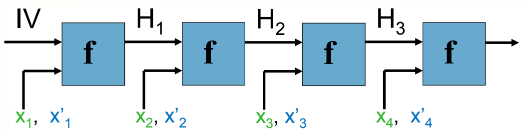
36

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)
  - $h(x_1)= h(x_2)= \ldots =h(x_t)$

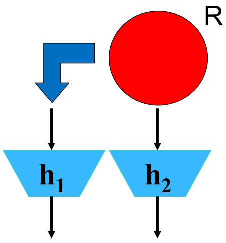
Multi-collisions on iterated hash function (2)



- 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 (time: 4 collisions)**

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 = $2^{n1/2}$-fold multi-collision for h_2
    - in R: search by brute force for h_1
  - Time: n1. $2^{n2/2} + 2^{n1/2}$ $<< 2^{(n1 + n2)/2}$



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

Multi-collisions [Joux '04]

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

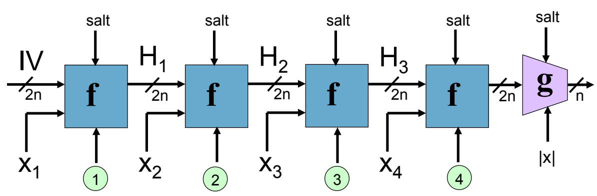
concatenation of 2 iterated hash functions ($g(x)= h_1(x) \,||\, h_2(x)$) is as most as strong as the strongest of the two (even if both are independent)

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

Summary

Improving MD iteration

salt + output transformation + counter + wide pipe



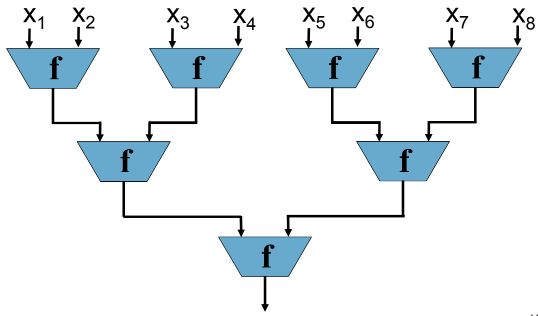
security reductions well understood
many more results on property preservation
impact of theory limited

Improving MD iteration

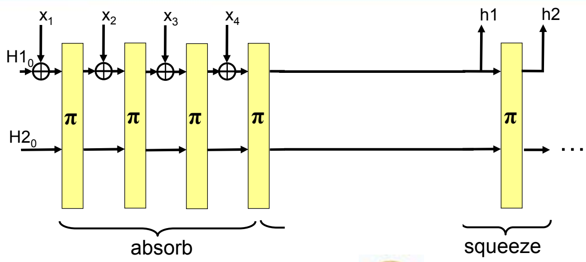
- degradation with use: salting (family of functions, randomization)
  - or should a salt be part of the input?
- PRO: strong output transformation g
  - also solves length extension
- long message 2^{nd} preimage: preclude fix points
  - counter f → f_i [Biham-Dunkelman'07]
- multi-collisions, herding: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

Tree structure: parallelism

[Damgård'89], [Pal-Sarkar'03]



Permutation (π) based: sponge

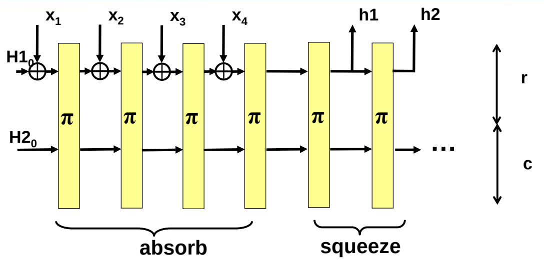


example: RadioGatun

generalization ("Parazoa")

JH, Cubehash, Fuge, Grindahl, Hamsi, Luffa

Permutation (π) based: sponge



if H1 has r bits (rate), H2 has c bits (capacity) and the permutation π is "ideal", then a sponge function has security $O(2^c)$ against (2^{nd}) preimage attacks and $O(2^{c/2})$ against collision attacks

Summary

- growing theory to reduce security properties of hash function to that of compression function (MD) or permutation (sponge)
  - preservation of large range of properties
  - relation between properties
- it is very nice to assume multiple properties of the compression function f, but unfortunately it is very hard to verify these
- still no single comprehensive theory

Agenda

- Definitions
- Iterations (modes)
- Compression functions
- Constructions
- SHA-3
- Conclusions

Compression functions

Block ciphers

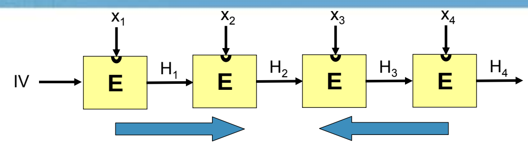
- E: $\{0,1\}^n \times \{0,1\}^k \to \{0,1\}^n$ or $E_K: \{0,1\}^n \to \{0,1\}^n$
- family of permutations on the domain $\{0,1\}^n$
- every key selects one permutation
  - block length n: there exist $2^n! \approx 2^{(n-1)2^n}$ permutations
  - key length k: 2^k selectable permutations only

	year	n	k
DES	1977	64	56
3-DES	1978	64	112, 168
IDEA	1991	64	128
AES	1997	128	128, 192, 256

Hash functions based on block ciphers

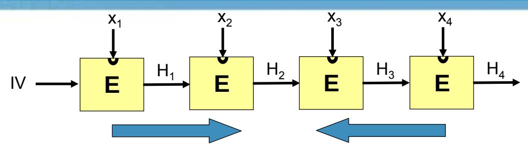
- why
  - trust
  - reduce design, evaluation, and implementation effort
  - compact implementation
  - a nice research problem
- why not
  - slow (one key schedule per encryption)
  - weaknesses which are not relevant to encryption (AES-256, weak keys, fixed points)
  - block-oriented output: structural problems
  - export restrictions
- rate = # blocks hashed per encryption

Single block length: [Rabin'78]



- Merkle's meet in the middle: (2^{nd}) preimage in time $2^{n/2}$
  - Select $2^{n/2}$ values for (x_1,x_2) and compute forward H'_2
  - Select $2^{n/2}$ values for (x_3,x_4) and compute backward H''_2
  - By the birthday paradox expect a match and thus a (2^{nd}) preimage
- extensions
  - [Quisquater+89] low memory version (distinguished points)

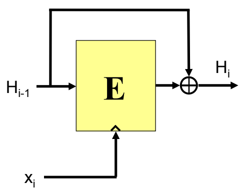
Single block length: [Rabin'78]



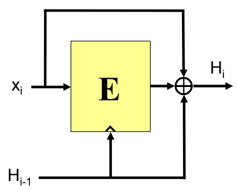
- consider a meet in the middle attack where it takes 1 step to compute forward and 2^s step to compute backwards
- how long does it take to find a 2^{nd} preimage?
- answer $2^{1+(n+s)/2}$ steps [Lai-Massey'92]

Block cipher (E_K) based: single block length

Davies-Meyer



Miyaguchi-Preneel

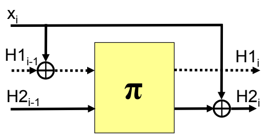


- output length = block length m; rate 1; 1 key schedule per encryption
- 12 secure compression functions (in ideal cipher model)
  - lower bounds: collision $2^{m/2}$, (2nd) preimage 2^m
- [Preneel+'93], [Black-Rogaway-Shrimpton'02], [Duo-Li'06], [Stam'09],…

Permutation (π) based

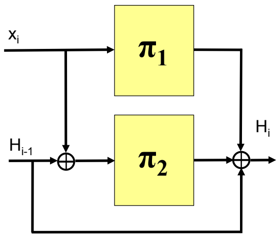
parazoa

JH



small permutation

Grøstl



55

Single Block Length (3)

- Secure schemes have proof in the ideal cipher model [Winternitz'82] and [Black-Rogaway-Shrimpton'02]
- Ideal cipher?
- Define $B_{k,n}$ the set of all block ciphers with k-bit keys and n-bit block
  - The cardinality of this set is $|B_{k,n}| = \binom{2^n!}{2^k}$
- And ideal (block) cipher is a block cipher selected according to the uniform distribution from the set $B_{k,n}$
- These proofs protect against generic attacks. But small deviations from being ideal can result in devastating attacks on the hash function
  - DES: weak and semi-weak keys
  - SHACAL-1 (based on SHA-1): best known attack on SHACAL 2^{500} but collisions for SHA-1 in 2^{69}
  - AES-128 has special structure up to 7 out of 10 rounds [Rijmen-Knudsen'07]; even worse for AES-192 and AES-256 (related key attacks!)

56

Iteration modes and compression functions

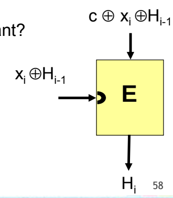
- security of simple modes well understood
- powerful tools available
- analysis of slightly more complex schemes very difficult
- which properties are meaningful?
- which properties are preserved?
- MD versus sponge is still open debate

57

Exercise: analyze the security

- Block cipher E with block length and key length equal to n = 128 bits
- Compression function $H_i = f(H_{i-1}, x_i)$
- Hash function h: starts with fixed IV, Merkle-Damgaard iteration; pad at the end with zeroes; fill the last block with the 88-bit string 1000…000 followed by the message length in a field of 40 bits
- C is the 128-bit constant 0xAAAAAA…A
- H_0 is the 128-bit constant 0x000000..0

1. Is the compression function f preimage resistant?
2. Is the compression function f 2nd preimage resistant?
3. Is the compression function f collision resistant?
4. Is the hash function h preimage resistant?
5. Is the hash function h 2nd preimage resistant?
6. Is the hash function h collision resistant?

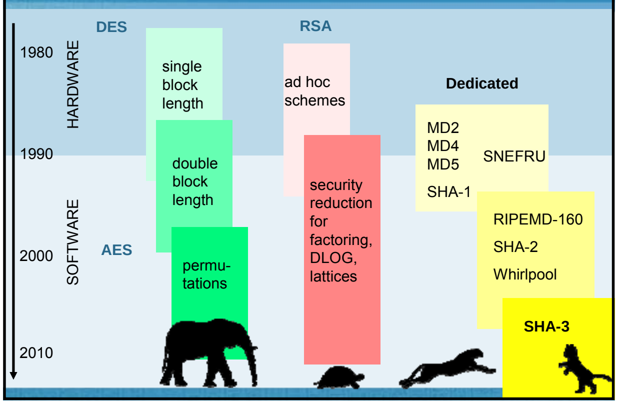


58

Hash function constructions

59

Hash function history 101



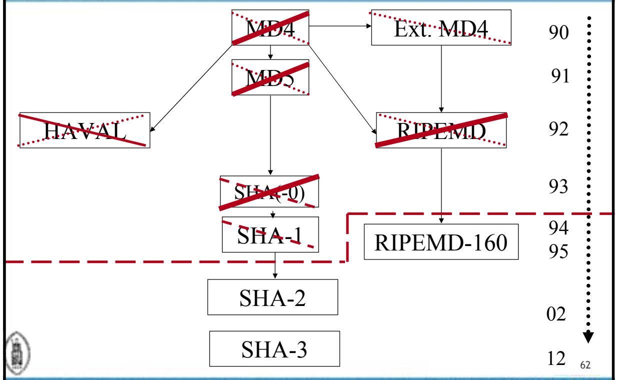
Hash function constructions

block cipher based
- well studied but need very strong assumption on block cipher
- due to key schedule for every encryption at least 3-4 times slower than AES
- 30 proposals, more than half broken
- progress in proofs steady but slowly

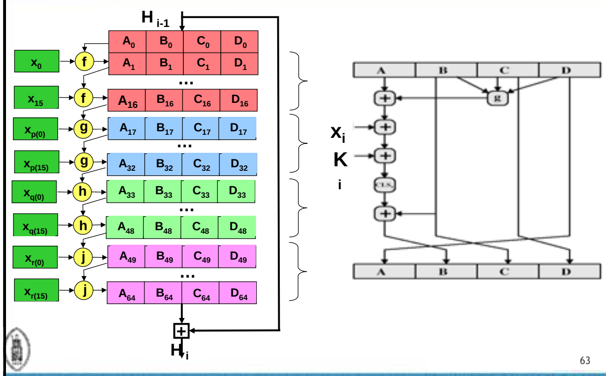
based on algebraic constructions with security reduction
- factoring, discrete log, ECC: very slow
- additive: lattices/knapsacks
- multiplicative: matrices

dedicated hash functions
- >40 designs until 2008
- about 30 broken: X.509 Annex D, FFT-hash I,II, N-hash, Snefru, MD2, …

MDx-type hash function history

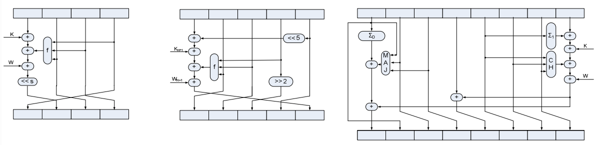


MD5 [Rivest'91]: 4 rounds of 16 steps



State updates in the MD4 family

MD4 SHA/SHA-1 SHA-256

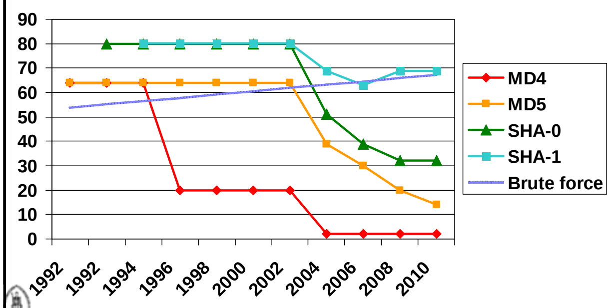


Design principles copied in MD5, RIPEMD, HAVAL, SHA, SHA-1, SHA-256, ...
- All hash functions in use today

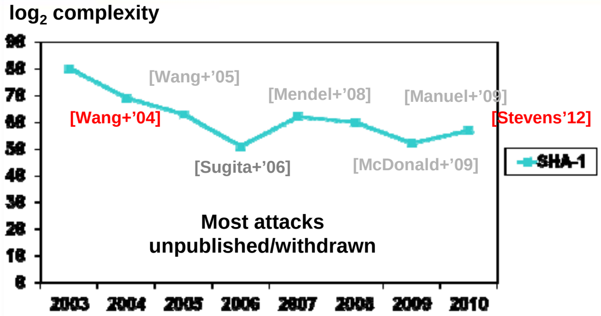
Slide credit: C. Rechberger

The complexity of collision attacks

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



SHA-1 designed by NIST (NSA) in '94

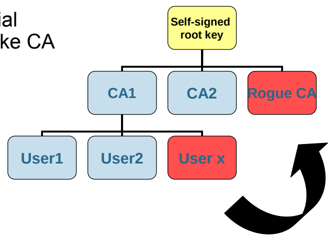


prediction: collision for SHA-1 in the next 12 months

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**

68

SHA-2 [NIST'02]

- SHA-224, SHA-256, SHA-384, SHA-512
  - non-linear message expansion
  - 64/80 steps
  - SHA-384 and SHA-512: 64-bit architectures
- SHA-256 collisions: 31/64 steps $2^{65.5}$ [Mendel+'13]
  - free start collision: 52/64 steps (2^{12x}) [Li+12]
  - non-randomness 47/64 steps (practical) [Biryukov+11][Mendel+11]
- SHA-256 preimages: 45/64 steps (2^{25x}) [Khovratovich+'12]
- implementations today faster than anticipated
- adoption
  - industry slow in migrating; may be now implementing SHA-3
  - very slow for TLS/IPsec (no pressing need)

69

Agenda

- Definitions
- Iterations (modes)
- Compression functions
- Constructions
- SHA-3
- Conclusions

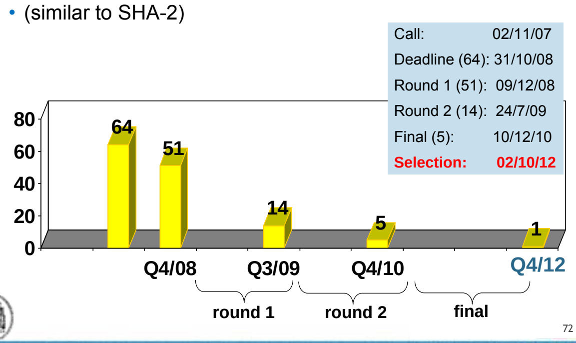
70

SHA-3
(bits and bytes)

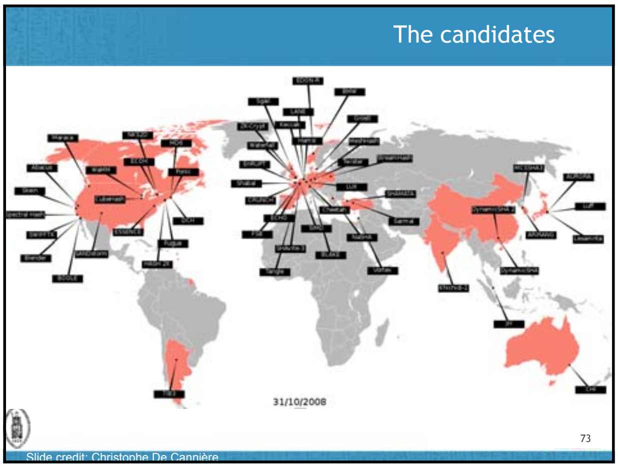
71

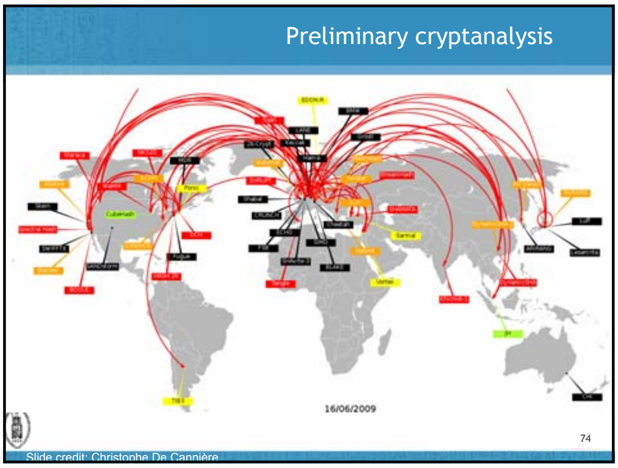
NIST AHS competition (SHA-3)

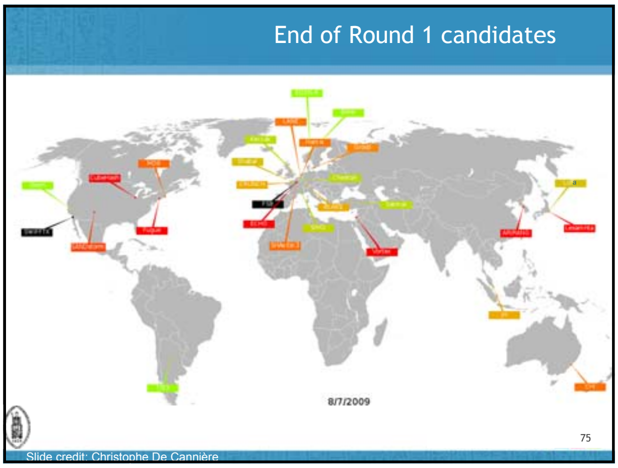
- SHA-3: 224, 256, 384, and 512-bit message digests
- (similar to SHA-2)

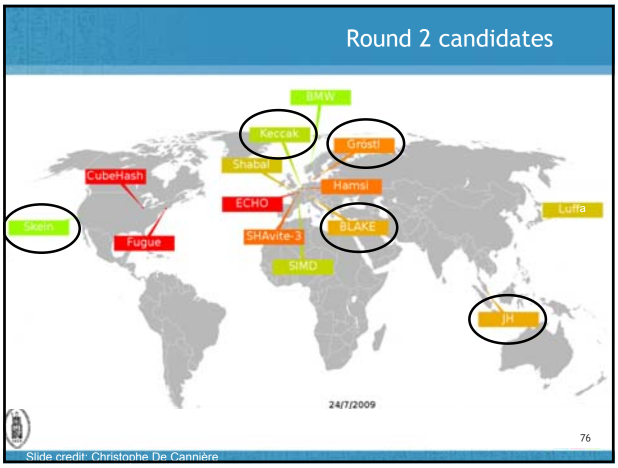


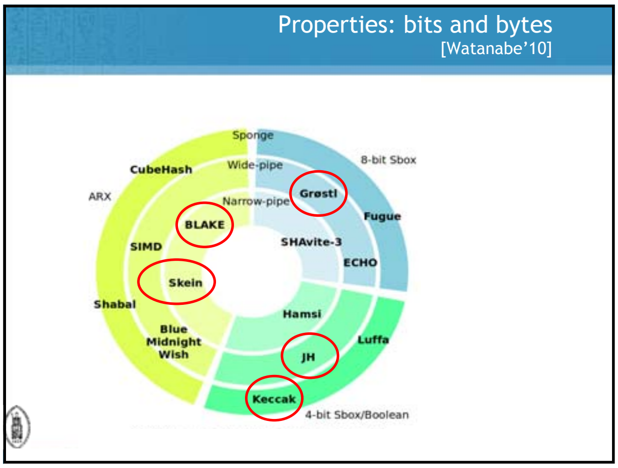
72

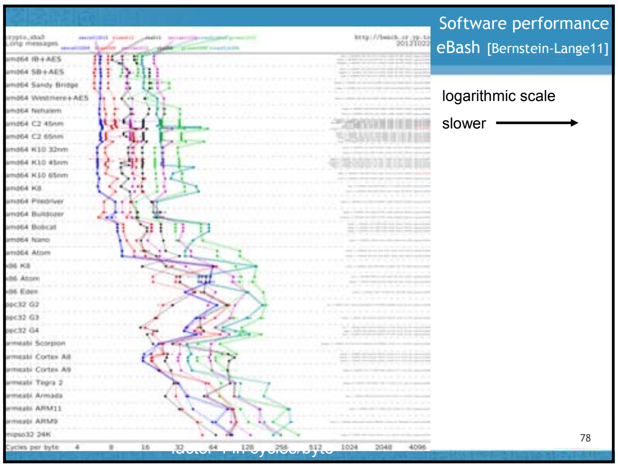

The candidates
31/10/2008
73
Slide credit: Christophe De Cannière


Preliminary cryptanalysis
16/06/2009
74
Slide credit: Christophe De Cannière

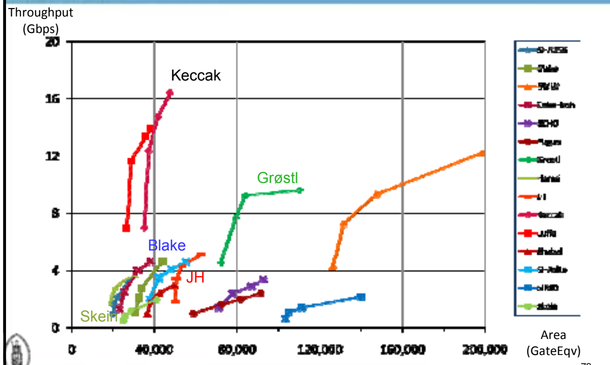

End of Round 1 candidates
8/7/2009
75
Slide credit: Christophe De Cannière


Round 2 candidates
BMW
Keccak
Grøstl
Shabal
CubeHash
Hamsi
ECHO
Skein
BLAKE
Luffa
SHAvite-3
Fugue
SIMD
JH
24/7/2009
76
Slide credit: Christophe De Cannière


Properties: bits and bytes
[Watanabe'10]
Sponge
8-bit Sbox
CubeHash
Wide-pipe
ARX
Narrow-pipe
Grøstl
BLAKE
Fugue
SIMD
SHAvite-3
ECHO
Skein
Shabal
Hamsi
Blue Midnight Wish
Luffa
JH
Keccak
4-bit Sbox/Boolean

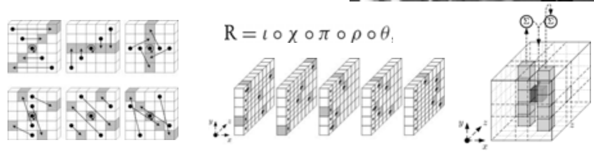

Software performance
eBash [Bernstein-Lange11]
logarithmic scale
slower
78

Hardware: post-place & route results for ASIC 130nm [Guo-Huang-Nazhandali-Schaumont'10]



Slide credit: Patrick Schaumont, Virginia Tech

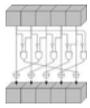
Keccak



permutation: 25, 50, 100, 200, 400, 800, 1600

nominal version:

- 5x5 array of 64 bits
- 18 rounds of 5 steps

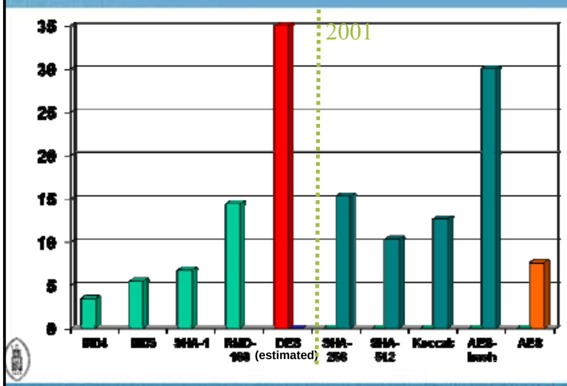


Keccak: FIPS

- new number (not 180-x)
- flexible output length and tree structure (Sakura) allowed by additional encoding
- six versions
  - n=256; c = 256; r = 1344 (84%)
  - n=256; c = 256; r = 1344 (84%)
  - n=384; c = 512; r = 1088 (68%)
  - n=512; c = 512; r = 1088 (68%)
  - n=x; c = 256; r = 1344 (84%)
  - n=x; c = 512; r = 1088 (68%)

If H1 has r bits (rate), H2 has c bits (capacity) and the permutation π is "ideal", then a sponge function has security $O(2^c)$ against (2nd) preimage attacks and $O(2^{c/2})$ against collision attacks

Performance of hash functions - Bernstein
(cycles/byte) Intel Core 2 Quad Q9550; 4 x 2833MHz (2008)



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
- theory is developing for more robust iteration modes and extra features; still early for building blocks
- Nirwana: efficient hash functions with security reduction