

Cryptographic Hash Functions Revisited: The NIST SHA-3 Competition

Bart Preneel
 COSIC – Katholieke Universiteit Leuven, Belgium
 bart.preneel(AT)esat.kuleuven.be

CASED
 May 2009



Outline

- definitions
- applications
- generic attacks
- attacks on iterated constructions: the rise and fall of MD
- SHA-0, SHA-1
- SHA-2
- SHA-3: the NIST AHS competition
- SHA-4
- conclusions

2

Hash functions

- MDC (manipulation detection code)
 - (MDC-2)
 - (MD5)
 - (SHA-1)
 - RIPEMD-160
 - SHA-256, SHA-512
- Protect short hash value rather than long text

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

3

Hash function flavours

```

    graph TD
      CH[cryptographic hash function] --> MAC
      CH --> MDC
      MDC --> OWHF
      MDC --> UOWHF_TCR[UOWHF (TCR)]
      MDC --> CRHF
      style MDC stroke:#00aaff,stroke-width:2px
      style this_talk[this talk] stroke:#00aaff,stroke-width:2px
      this_talk --> MDC
    
```

4

Informal definitions (1)

- 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

5

Security requirements (n-bit result)

preimage	2 nd preimage	collision
$?$	$x \neq ?$	$? \neq ?$
\downarrow	\downarrow	\downarrow
\downarrow	\downarrow	\downarrow
\downarrow	\downarrow	\downarrow
$h(x)$	$h(x) = h(x')$	$h(x) = h(x')$
2^n	2^n	$2^{n/2}$

6

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 ,

$$\Pr_{x \in D} \{ h(M(h(x))) = h(x) \} < \epsilon$$
 Here the probability 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' ,

$$\Pr_{x \in D} \{ h(M'(x)) = h(x) \} < \epsilon$$
 Here the probability is taken over the random choices of M' .

7

Formal definitions: collision resistance

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

- the functions h_s are preimage resistant and second preimage resistant
- 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 ,

$$\Pr_S \{ F(H) \neq "?" \} < \epsilon$$
 Here the probability is also taken over the random choices of F .

8

Formal definitions - continued

- For collision resistance: considering a **family** of hash functions indexed by a parameter ("key") is essential for formalization (but see [Rogaway'06]: "formalizing human ignorance")
- For (2nd) preimage resistance, one can choose the challenge (x) and/or the key 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).

9

Relation between formal definitions

[Rogaway-Shrimpton'04]

Figure 1: Summary of the relationships among seven notions of hash-function security. Solid arrows represent conventional implications, dotted arrows represent provisional implications (their strength depends on the relative size of the domain and range), and the lack of an arrow represents a separation.

10

Informal definitions (2)

- preimage resistant \Leftrightarrow 2nd preimage resistant
 - take a preimage resistant hash function; add an input bit b and replace one input bit by the sum modulo 2 of this input bit and b
- 2nd preimage resistant \Leftrightarrow preimage resistant
 - if h is OWHF, \underline{h} is 2nd preimage resistant but not preimage resistant:

$$\underline{h}(x) = \begin{cases} 0 \parallel x & \text{if } |x| \leq n \\ 1 \parallel h(x) & \text{otherwise} \end{cases}$$
- collision resistant \Rightarrow 2nd preimage resistant
- [Simon'98] one cannot derive collision resistance from "general" preimage resistance (there exists no black box reduction)

11

Applications

- digital signatures: OWHF/CRHF, 'destroy algebraic structure'
- information authentication: protect authenticity of hash result
- protection of passwords: preimage resistant
- confirmation of knowledge/commitment: OWHF/CRHF
- pseudo-random string generation/key derivation
- micropayments (e.g., micromint)
- construction of MAC algorithms, stream ciphers, block ciphers
- (redundancy: hash result appended to data before encryption)

Until recently: 800 uses of MD5 in Windows

12

Applications (2)

- Collision resistance is not always necessary
- Other properties are needed:
 - pseudo-randomness if keyed (with secret key)
 - near-collision resistance
 - partial preimage resistance
 - multiplication freeness
 - indistinguishable from random oracle
- how to formalize these requirements and the relation between them?

13

Summary



14

Brute force (2^{nd}) preimage

- If one can attack 2^t simultaneous targets, the effort to find a single preimage is 2^{n-t}
 - note for $t = n/2$ this is $2^{n/2}$
- [Hellman'80] if one has to find (second) preimages for many targets, one can use a time-memory trade-off with $\Theta(2^n)$ precomputation and storage $\Theta(2^{2n/3})$
 - inversion of one message in time $\Theta(2^{2n/3})$
- [Wiener'02] if $\Theta(2^{3n/5})$ targets are attacked, the full cost per (2^{nd}) preimage decreases from $\Theta(2^n)$ to $\Theta(2^{2n/5})$
- answer: randomize hash function
 - salt, spice, "key": parameter to index family of functions

15

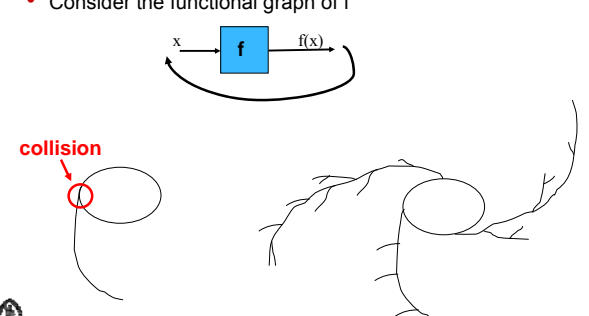
The birthday paradox for collisions

- 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) is $1 - \exp(-r(r-1)/2S)$
- S large, $r = \sqrt{S}$, $p = 0.39$: finding a collision takes computation and memory \sqrt{S}
 - for birthdays: $S = 365$, $r = 23$, $p = 0.50$

16

Brute force collision search: low memory

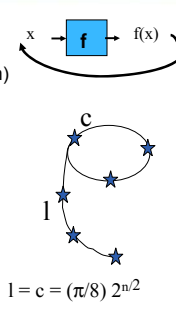
- Consider the functional graph of f



17

Brute force collision search: low memory

- Efficient implementation of the birthday attack [Pollard'78][Quisquater'89]
 - very little memory (cycle finding algorithm)
 - full parallelism [Wiener-van Oorschot'94]
- Distinguished point (d bits)
 - $\Theta(e^{2n/2} + e^{2^{d+1}})$ steps
 - $\Theta(n^{2^{n/2-d}})$ memory
 - with e the cost of one function evaluation
- [Wiener'02] full cost: $\Theta(e n^{2^{n/2}})$



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

18

Brute force attacks in practice

- (2nd) preimage search
 - n = 128: 90 M\$ for 1 year if one can attack 2⁴⁸ targets in parallel
 - n = 128: 90 B\$ for 1 year if one can attack 2³⁸ targets in parallel
- parallel collision search
 - n = 128: 1 M\$ for 12 hours (or 1 year on 60K PCs)
 - n = 160: 90 M\$ for 1 year
 - need 256-bit result for long term security (25 years or more)

19

Can we get rid of collision resistance?

- collision resistance
 - requires double output lengths
 - requires family of functions for formalization
 - is hard to achieve (e.g., not by black box reduction from one-wayness)
- UOWHF (TCR, eSec) **randomize** hash function after choosing the message
- [Halevi-Krawczyk'05] randomized hashing = RMX mode:

$$H(r \parallel x_1 \oplus r \parallel x_2 \oplus r \parallel \dots \parallel x_t \oplus r)$$
 - needs e-SPR (not met by MD5 and SHA-1 reduced to 53 steps)
 - issues with **insider attacks** (i.e. attacks by the signer)
 - birthday attack for signatures [Gauravaram-Knudsen+09]

20

Outline

- definitions
- applications
- generic attacks
- attacks on iterated constructions: the rise and fall of MD
- SHA-0, SHA-1
- SHA-2
- SHA-3: the NIST AHS competition
- SHA-4
- conclusions

21

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 many problems...

22

Security relation between f and h

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

23

Security relation between f and h

- Solution: Merkle-Damgard (MD) strengthening (popular!)
 fix IV, use unambiguous padding and insert length at the end
- [MD'89] f is collision resistant \Rightarrow h is collision resistant
- [Lai-Massey'92] f is 2nd preimage resistant \Leftrightarrow h is 2nd preimage resistant ?

24

Construction: relation between f and h (2)

[Damgård-Merkle'89]

Let f be a collision resistant function mapping l to n bits (with $l > n$).

- If the padding contains the length of the input string, and if f is preimage resistant, the iterated hash function h based on f will be a CRHF.
- If an unambiguous padding rule is used, the following construction will yield a CRHF ($l-n > 1$):

$$H_t = f(H_0 \parallel 0 \parallel x_t)$$

$$H_i = f(H_{i-1} \parallel 1 \parallel x_i) \quad i=2,3,\dots,t.$$

25

Comment: tree structure

already suggested by Damgård in 1989; further work by Sarkar et al.

26

Construction: relation between f and h (3)

[Lai-Massey'92]

Assume that the padding contains the length of the input string, and that the message x (without padding) contains at least two blocks.

Then finding a second preimage for h with a fixed IV requires 2^n operations **iff** finding a second preimage for f with arbitrarily chosen H_{i-1} requires 2^n operations.

- this theorem is not quite right (see below)
- very few hash functions have a strong compression function
- very few hash functions are designed based on a strong compression function in the sense that they treat x_i and H_{i-1} in the *same way*.

27

Security relation between f and h (4)

- MD with envelope method (prepend and append secret key, no output transformation) works for pseudo-randomness/MAC [BCK'96]
 - but there are some issues and HMAC is a better construction

- MD preserves Preimage Awareness [Dodis-Ristenpart-Shrimpton'09]

28

Security relation between f and h (5)

- MD does **not** work for UOWHF [Bellare-Rogaway'97]
- MD without output transformation is vulnerable to extension attack: if one knows $h(x)$, easy to compute $h(x \parallel y)$ without knowing x

hence does not preserve pseudo-random oracle (PRO) property [Coron+05]

29

Attacks on MD: 1999-2006

- long message 2^{nd} preimage attack [Dean-Felten-Hu'99], [Kelsey-Schneier'05]
 - if one hashes 2^t message blocks with an iterated hash function, the effort to find a second preimage is only $2^{n+t-1} + t \cdot 2^{n/2+t-1}$
- multi-collision attack and impact on concatenation [Joux'04]
 - the concatenation of 2 iterated hash functions ($g(x) = h_1(x) \parallel h_2(x)$) is **as most as strong as the strongest** of the two (even if both are independent)
 - cost of collision attack against g at most $n_1 \cdot 2^{n_1/2} + 2^{n_1/2} \ll 2^{(n_1 + n_2)/2}$
- herding attack [Kelsey-Kohno'06]
 - reduces security of commitment using a hash function from 2^n
 - on-line $2^{n_1} + \text{precomputation } 2 \cdot 2^{(n_1+1)/2} + \text{storage } 2^t$

30

Defeating MD for 2nd preimages [Dean-Felten-Hu'99] and [Kelsey-Schneier 05]

[Merkle'79]: if one hashes **2^t messages**, the average effort to find a second preimage for one of them is 2^{n-t}

New: if one hashes **2^t message blocks** with an iterated hash function, the effort to find a second preimage is only 2^{n-t+1} + t 2^{n/2+1}

- idea: create expandable message using fixed points
 - Finding fixed points can be easy (e.g., Davies-Meyer)
- find 2nd preimage that hits any of the 2^t chaining values in the calculation
- stretch the expandable message to match the length (and thus the length field)
- But still very long messages for attack to be meaningful
 - n=128, t=32, complexity reduced from 2¹²⁸ to 2⁹⁷, length is 256 Gbyte

31

Defeating MD for 2nd preimages (2)

expandable message: $H_0 \xrightarrow{f, x'_1} H_1 \xrightarrow{f, x'_2} H'_1 \xrightarrow{f, x'_3} \dots$

target message: $H_2 \xrightarrow{f, x_1} H_3 \xrightarrow{f, x_2} H_4 \xrightarrow{f, x_3} \dots \xrightarrow{f, x_{2t-1}} H_{2t-1} \xrightarrow{f, x_{2t}} H_{2t}$

success probability $\approx 2^t$

$h(x'_1 || x'_2 || x'_3 || \dots || x'_{2t}) = h(x_1 || x_2 || x_3 || \dots || x_{2t-1} || x_{2t})$

32

How to find fix points?

- Davies-Meyer: $E_{x_i}(H_{i-1}) \oplus H_{i-1}$
- Fix point $H_{i-1} = D_{x_i}(0)$ for any x_i
 - Proof: $E_{x_i}(H_{i-1}) \oplus H_{i-1} = H_{i-1}$ implies $E_{x_i}(H_{i-1}) = 0$
- Expandable message using meet-in-the-middle
 - Generate 2^{n/2} values x_2 and compute $H_1 = D_{x_2}(0)$
 - Generate 2^{n/2} values x_1 and compute $H_1 = E_{x_1}(H_0) \oplus H_0$
 - Find a match with high probability
- For non-Davies-Meyer: use the trick of Joux

33

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...

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

34

Multi-collisions [Joux'04]

Consider h_1 (n1-bit result) and h_2 (n2-bit result), with $n_1 \geq n_2$. The concatenation of two 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 $n_1 \cdot 2^{n_2/2} + 2^{n_1/2} \ll 2^{(n_1+n_2)/2}$
- Cost of (2nd) preimage attack against g at most $n_1 \cdot 2^{n_2/2} + 2^{n_1} + 2^{n_2} \ll 2^{n_1+n_2}$
- 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)

35

Multi-collisions (2) [Joux'04]

- 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**

36

Other issues with MD iteration: herding

- Herding attack [Kelsey-Kohno'06]
 - reduces security of commitment using a hash function
 - on-line 2^{n-t} + precomputation $2 \cdot 2^{(n+t)/2}$ + storage 2^t
 - example ($n=128, t=42$): with a storage of 100 Terabyte and a precomputation of 2^{86} steps, a 128-bit commitment computed using an iterated hash function can be spoofed with effort 2^{86} steps

37

Herding attack (2)

- protocol: publish $h(x)$, reveal x at later date
- find second preimage $x' = z || y || x$ with z and y selected in 2020
- approach: generate collision tree (diamond structure) of 2^t values H_{i-1} and x_i , hashing to the same value (cost $2 \cdot 2^{t/2} \cdot 2^{n/2}$)
 - work factor for first layer: $x^2/2^{n+1} = 2^t$ or $x = \sqrt{2 \cdot 2^{n/2} \cdot 2^{n/2}}$
- z = result of all Champions League finals between 2010 and 2020
- try in 2020 random strings y until $h(z || y) = H_{j-1}$ for some j (cost 2^{n-t})
- then $h(z || y || x_j) = h(x)$, so you can claim that you "knew" z in 2008

38

Herding attack (3)

new message

success probability $\approx 2^{-1}$

$h(z || y || x) =$ committed value

39

Improving MD iteration

- degradation with use: salting (family of functions, randomization)
- extension attack + PRO preservation: strong output transformation g (which includes total length and salt)
- long message 2^{nd} preimage: preclude fix points
 - counter $f \rightarrow f_i$ [Biham-Dunkelman] or dithering [Rivest]
- multi-collisions, herding: avoid breakdown at $2^{n/2}$ with larger internal memory: known as wide pipe
 - e.g., extended MD4, RIPEMD, [Luks'05]

40

Many more ideas....

- [Biham-Dunkelman'06] Haifa: bit counter and salt input to f
- [Bellare-Ristenpart'06] EMD transform (envelope MD): preserves CR, PRF, RO

- [Andreeva+06] analysis of preservation of CR, (e/a)-PR, (e/a)-SPR, (RO, PRF)
- [Yasuda08] split padding MD variant: preserves SPR, OW

41

Many more ideas....

- LANE (SHA-3 submission by [Indestege+], COSIC

C_i , number of bits hashed so far
 ϕ flag that indicates presence/absence of salt S
 n output length
 l total message length in bits

42

More details..

- [Andreeva+06] analysis of preservation of CR, (e/a/-)PR, (e/a/-)SPR, (RO, PRF)
 - Most constructions do not preserve other properties
 - ROX construction preserves 7 properties with a logarithmic number of random masks and a randomized padding (both a function of key and salt)

43

Sponge functions

Examples

- Panama
- RadioGatun
- Grihndahl
- Keccak

44

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

45

Outline

- definitions
- applications
- generic attacks
- attacks on iterated constructions: the rise and fall of MD
- SHA-0, SHA-1
- SHA-2
- SHA-3: the NIST AHS competition
- SHA-4
- conclusions

46

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
- some constructions use codes over $GF(2^n)$

based on algebraic constructions with security proof

- factoring, discrete log, ECC: very slow
- additive: lattices
- multiplicative: matrices

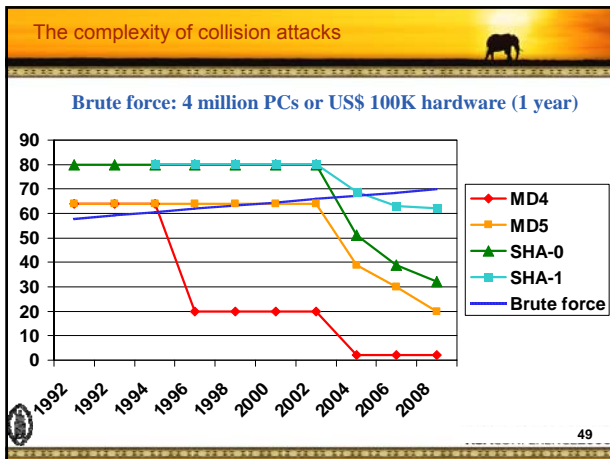
dedicated hash functions

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

47

MDx-type hash function history

48



SHA(-0)

- SHA designed by NIST (NSA) in '92
 - now called SHA-0, because of '94 of publication SHA-1
- very similar to MD5:
 - 16 extra steps (from 64 to 80)
 - message expansion uses bitwise code rather than repetition

$$W_j \leftarrow (W_{j-3} \oplus W_{j-8} \oplus W_{j-14} \oplus W_{j-16}) \ggg j > 15$$
 - quasicyclic code with $d_{\min} = 23$
- 1994: withdrawn by NIST for unidentified flaw
- 2004: collisions for in 2^{51} [Joux+'04]
- 2005: collisions in 2^{39} [Wang+'05]
- 2007: collisions in 2^{32} [Joux+'07]
- 2008: collisions in 1 hour [Manuel-Peyrin'08]
- 2008: preimages for 49 steps in 2^{159} [DeCannière+'08]

50

SHA-1

- fix after 2 years ('94) to SHA-0
- add rotation to message expansion: quasicyclic code with $d_{\min} = 25$

$$W_j \leftarrow (W_{j-3} \oplus W_{j-8} \oplus W_{j-14} \oplus W_{j-16}) \ggg j > 15$$

collisions

- 53 steps [Oswald-Rijmen'04 and Biham-Chen'04]
- 58 steps [Wang+'05]
- 64 steps in 2^{35} – highly structured [De Cannière-Rechberger'06-'07]:
- 70 steps in 2^{44} – highly structured [De Cannière-Rechberger'06-'07]:
- 70 steps 2^{39} (4 days on a PC) [Joux-Peyrin'07]
- 2^{69} [Wang+'05]
- 2^{63} [Wang+'05 - unpublished]
- 2^{62} [Mendel+'08 - unpublished]
- 2^{52} [McDonald+'09 - unpublished]

- preimages for 49/80 steps in 2^{157} [DeCannière-Rechberger'08]

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

Impact of collisions (1)

- collisions for MD5, SHA-0, SHA-1
 - 2 messages differ in a few bits in 1 to 3 512-bit input blocks
 - limited control over message bits in these blocks
 - but arbitrary choice of bits before and after them

- what is achievable for MD5?
 - 2 colliding executables/postscript/gif/... [Lucks-Daum'05]
 - 2 colliding RSA public keys – thus with colliding X.509 certificates [Lenstra+'04]
 - chosen prefix attack: different IDs, same certificate [Stevens+'07]
 - 2 arbitrary colliding files (no constraints) in 12 hours for 1 M\$

53

Impact of collisions (2)

- [Sotirov-Stevens-Appelbaum,-Lenstra-Molnar-Osvik-de Weger '08] MD5 considered harmful today
 - fake CA certificate.
 - results in a rogue CA: its certificates are trusted by all common browsers
 - need to predict serial number + validity period
- 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

54

Impact of collisions (3)

- digital signatures: only an issue if for **non-repudiation**
- none** for signatures computed before attacks were public (1 August 2004)
- ~~none for certificates if public keys are generated at random in a controlled environment~~
- substantial** for signatures after 1 August 2005 (cf. traffic tickets in Australia)

55

And (2nd) preimages?

- security degrades with number of applications
- for large messages even with the number of blocks (cf. supra)
- specific results:
 - MD2: 2^{73} [Knudsen+09]
 - MD4: 2^{102} [Leurent'08]
 - MD5: 2^{123} [Asaki-Aoki'09]
 - SHA-0: 49 of 80 steps in 2^{159} [De Cannière-Rechberger'08]
 - SHA-1: 44 of 80 steps in 2^{157} [De Cannière-Rechberger'08]

56

HMAC?

- HMAC-MD4: 2^{72} chosen plaintexts & 2^{77} time
- HMAC-MD5: 2^{51} chosen plaintexts & 2^{100} time in a related key setting
- HMAC-SHA-1 seems fine for now

57

Fixes/Alternatives

- RIPEMD-160 seems more secure than SHA-1 ☺
- message precoding for SHA-1
- small patches to SHA-1

58

SHA-2 [01]

- 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
- collisions: 24 steps [Sanadhya-Sarkar'08]
- adoption
 - in spite of pedigree and analysis, industry migrates in the next 2-3 years to SHA-2
 - very slow for TLS/IPsec (no pressing need)
- alternative: Whirlpool (AES-based), in ISO standard

59

Performance of hash functions - Bernstein (cycles/byte) AMD Intel Pentium D 2992 MHz (f64)

Function	Performance (cycles/byte)
MD4	4
MD5	5
SHA-1	11
RMD-160	17
SHA-256	22
SHA-512	31
Whirlp.	41
AES-BC	30
AES	10

60

Outline

- definitions
- applications
- generic attacks
- attacks on iterated constructions: the rise and fall of MD
- SHA-0, SHA-1
- SHA-2
- SHA-3: the NIST AHS competition
- SHA-4
- conclusions

61

NIST Advanced Hash Function competition (AHS)

- 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
- standard will be published in 2012

Call: 02/11/07
Deadline (64): 31/10/08
Phase 1 (51): 9/12/08

62

Accepted submission to AHS competition (1)

- About 13 out of 51 have been broken and other 8 "damaged"
— 10 designers have conceded so far
- 13 not accepted, but only 5 are public (4 have been broken)

Analysis of 30/51 selected designs

63

Accepted submission to AHS competition (2)

Analysis of 30/51 selected designs

- all wide pipe + sponge designs have an output transformation
- 5 narrow designs do not have an output transformation
- most narrow designs have a counter (3 do not)

64

Accepted submission to AHS competition (3)

Analysis of 30/51 selected designs

Reference platform: Intel Core Duo

65

Selected designs (highly subjective)

- ARIRANG [KO] – J. Lim
- Blake [CH] – J.-P. Aumasson
- Cheetah [LU] – D. Khovratovich
- Cubehash [US] – D.J. Bernstein
- Echo [FR] – H. Gilbert
- FSB [FR] – M. Finiasz
- Fugue [US] – C. Jutla
- Grøstl [DK/AT/PO] – L.R. Knudsen
- JH [Singapore] – H. Wu
- Keccak [BE/IT] – J. Daemen
- LANE [BE] – S. Indestege
- Lesamnta [JP] – H. Yoshida
- Luffa [JP] – D. Watanabe
- MD6 [USA] – R.L. Rivest
- Shabal [FR] J.-F. Misarsky
- SHAvite-3 [IL] – O. Dunkelman
- SIMD [FR] – G. Leurent
- SKEIN [USA] – B. Schneier
- SWIFFTX [USA] – D. Micciancio

66

SWIFFTX

[Arbitman-Dogon-Lyubashevsky-Micciancio-Peikert-Rosen'08]

- compression function:
 - SWIFFT: FFT-like operation from $(\mathbb{Z}_2^{32})^{64}$ to \mathbb{Z}_{257}^{64}
 - sandwich: 3xSWIFFT - S-boxes - 1xSWIFFT
- asymptotic proof of security: *"it can be formally proved that finding a collision in a randomly-chosen compression function from the SWIFFTX family is at least as hard as finding short vectors in cyclic/ideal lattices over the ring $\mathbb{Z}[\alpha]/(\alpha^n+1)$ is in the worst case."*
- note: SWIFFT mapping is linear and some heuristics are needed to "kill" the linearity
- speed: 57 cpb

67

FSB [Augot-Finiasz-Gaborit-Manuel-Sendrier'08]

- compression function: multiplication of vector of Hamming weight w with a truncated quasi-cyclic binary matrix
 - can be interpreted as a syndrome computation of an error pattern with weight w
- MD iteration with Whirlpool as output transformation
- security can be reduced to:
 - (Computational Syndrome Decoding) *Given a binary $r \times n$ matrix H , a word $e \in \{0,1\}^r$ and an integer $w > 0$, find a word $e \in \{0,1\}^n$ of Hamming weight $\leq w$ such that $eH^T = e$.*
 - (Codeword Finding) *Given a binary $r \times n$ matrix H and an integer $w > 0$, and a non-zero word $e \in \{0,1\}^n$ of Hamming weight $\leq w$ with an all zero H -syndrome.*
- 324 cpb (can be optimized)

68

Advertisement: LANE

Designer: S. Indestege (COSIC)

H_i 256 bit and X_i 512 bit

Expanded linearly to 6 256-bit words

P_i/Q_i consist of 6/3 AES parallel rounds

- AddRoundKey: add round constant and counter
- SwapColumn to mix two 128-bit halves

69

Issues

- security
 - how to define an attack, e.g. pseudo-near collision, attacks with huge memory?
 - importance of proofs
- performance
 - designs with tunable security/performance tradeoff: how important are the nominal parameters?
 - do we care about a very large memory (500-700 bytes) which may be a problem for small devices?
 - can we exploit 64 or 128 cores? Intel AES instruction?

70

SHA-4

- an open competition such as SHA-3 is bound to result in new insights between 2009-2012
- 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
- maybe everyone will use SHA-2 in 2020?
- for which applications will security proofs be relevant?

71

ZesT: a SHA-4 candidate?

- Zémor-Tillich: consider the two generators of the group $SL(2; \mathbb{F}_{2^n})$

$$A_0 = \begin{pmatrix} x & 1 \\ 1 & 0 \end{pmatrix} \quad A_1 = \begin{pmatrix} x & x+1 \\ 1 & 1 \end{pmatrix}$$
 the hash value of a string x with elements $x[i]$ is $\prod_{i=1}^n A_{x[i]}$
- ZesT = vectorial version of the Zémor-Tillich function iterated $2x$
- security: ZesT is collision resistant if and only if the balance problem is hard and in particular if the representation problem is hard for the group $SL(2; \mathbb{F}_{2^n})$ and the generators A_0 and A_1
- performance: 10-20 times slower than SHA-512 but parallelism

More details: PhD thesis of Christophe Petit, UCL, May 2009

72

Hash functions: conclusions

- SHA-1 would have needed 128-160 steps instead of 80
- recent attacks are not dramatic for all applications, but they form a clear warning: upgrade asap
- theory is developing for more robust iteration modes and extra features; still early for building blocks
- use weaker security assumptions if possible (UOWHF?)
- Nirwana: efficient hash functions with security reduction

73

Hash functions: further reading

- NIST <http://csrc.nist.gov/groups/ST/hash/index.html>
 - first SHA-3 candidate conference: February 25-28, 2009, Leuven
 - workshop October 31-November 1, 2005 and August 24-25, 2006
- ECRYPT: <http://www.ecrypt.eu.org>
 - SHA-3 Zoo http://ehash.iaik.tugraz.at/wiki/The_SHA-3_Zoo
 - workshops in May 2007 and June 2005 + statement on hash functions
- The IACR eprint server <http://eprint.iacr.org>
- My 1993 PhD thesis <http://homes.esat.kuleuven.be/~preneel>
- Overview paper from 1998 (LNCS 1528)
<http://www.cosic.esat.kuleuven.be/publications/article-246.pdf>

Thank you for your attention

74