Finding collisions for SHA-1

Pierre Karpman

Based on joint work with Ange Albertini, Elie Bursztein, Yarik Markov, Thomas Peyrin and Marc Stevens

Université Grenoble Alpes

Real World Crypto — Zürich

2018–01–11

The near-anniversary of not a birthday search

- On 2017-01-15, the first (public?) SHA-1 collision was found
- ... Coming after the first *freestart* collision in Oct. 2015
- ... Coming after the first "theoretical" attack in 2005
- ... Coming after the first standardization of SHA-1 in 1995

Aim of this talk:

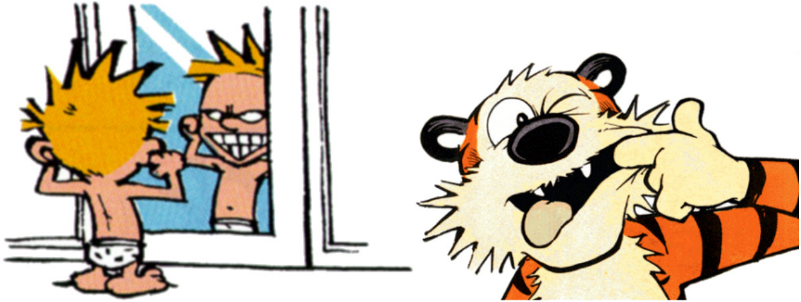
- What's a SHA-1 collision like? How do you compute one?
- How do you measure the "complexity" of such an attack?

A simple collision

h_0	4e a9 62 69 7c 87 6e 26 74 d1 07 f0 fe c6 79 84 14 f5 bf 45
M_1	7f 46 dc 93 a6 b6 7e 01 3b 02 9a aa 1d b2 56 0b
	45 ca 67 d6 88 c7 f8 4b 8c 4c 79 1f e0 2b 3d f6
	14 f8 6d b1 69 09 01 c5 6b 45 c1 53 0a fe df b7
	60 38 e9 72 72 2f e7 ad 72 8f 0e 49 04 e0 46 c2
h_1	8d 64 d6 17 ff ed 53 52 eb c8 59 15 5e c7 eb 34 f3 8a 5a 7b
M_2	30 57 0f e9 d4 13 98 ab e1 2e f5 bc 94 2b e3 35
	42 a4 80 2d 98 b5 d7 0f 2a 33 2e c3 7f ac 35 14
	e7 4d dc 0f 2c c1 a8 74 cd 0c 78 30 5a 21 56 64
	61 30 97 89 60 6b d0 bf 3f 98 cd a8 04 46 29 a1
h_2	1e ac b2 5e d5 97 0d 10 f1 73 69 63 57 71 bc 3a 17 b4 8a c5

h_0	4e a9 62 69 7c 87 6e 26 74 d1 07 f0 fe c6 79 84 14 f5 bf 45
$M_1 \oplus \Delta_1$	73 46 dc 91 66 b6 7e 11 8f 02 9a b6 21 b2 56 0f
	f9 ca 67 cc a8 c7 f8 5b a8 4c 79 03 0c 2b 3d e2
	18 f8 6d b3 a9 09 01 d5 df 45 c1 4f 26 fe df b3
	dc 38 e9 6a c2 2f e7 bd 72 8f 0e 45 bc e0 46 d2
h_1	8d 64 c8 21 ff ed 52 e2 eb c8 59 15 5e c7 eb 36 73 8a 5a 7b
$M_2 \oplus \Delta_2$	3c 57 0f eb 14 13 98 bb 55 2e f5 a0 a8 2b e3 31
	fe a4 80 37 b8 b5 d7 1f 0e 33 2e df 93 ac 35 00
	eb 4d dc 0d ec c1 a8 64 79 0c 78 2c 76 21 56 60
	dd 30 97 91 d0 6b d0 af 3f 98 cd a4 bc 46 29 b1
h_2	1e ac b2 5e d5 97 0d 10 f1 73 69 63 57 71 bc 3a 17 b4 8a c5

A comic application



```
>sha1sum *.pdf
23aa25d9e0449e507a8b4c185fdc86c35bf609bc calvin.pdf
23aa25d9e0449e507a8b4c185fdc86c35bf609bc hobbes.pdf
```

SHA-1 collisions recap

On the way to full practical attacks

What complexity for an attack

Conclusion & Future work

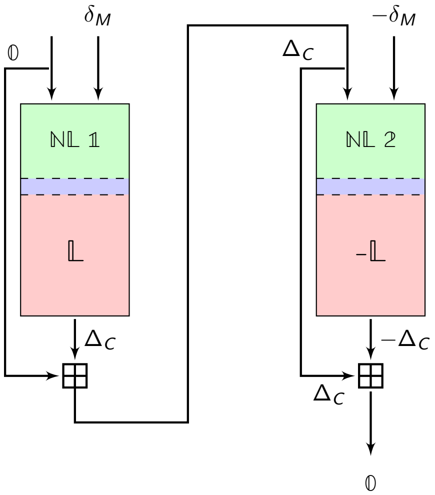
SHA-1 quick history

Secure Hash Standard "SHA-1"

- Standardized by NIST in Apr. 1995
- Similar to MD4/5
  - Merkle-Damgård domain extender
  - Compression function = ad hoc block cipher in Davies-Meyer mode
  - Unbalanced Feistel network, 80 steps
- Quick fix of "SHA-0" (May 1993)
- Hash size is 160 bits ⇒ collision security should be 80 bits

That's nice, but we want to attack it!

A two-block attack in a picture



The result

- SHA-1 is not collision-resistant (Wang, Yin & Yu, 2005)
- Attack complexity $\equiv 2^{69}$ (theoretical)
- Eventually improved to $\equiv 2^{61}$ (ditto, Stevens, 2013)

The attack process

1. Pick a linear path
2. Find a non-linear path (first block)
3. Find accelerating techniques (first block)
4. Compute a *near-collision* (a solution for $(0, \delta_M) \rightarrow \Delta_C$)
    - Possible expected wall time estimation (first block)
5. Find a non-linear path (second block)
6. Find accelerating techniques (second block)
7. Compute a *collision* (a solution for $(\Delta_C, -\delta_M) \rightarrow -\Delta_C$)
    - Possible expected wall time estimation (full attack)

Wall time estimation

Simple approach:

- Implement the attack
- Measure production rate $\#A_{xx}/\mathrm{s}$
- Multiply by probability that a solution A_{xx} extends to A_{80}

Early variant (crude):

- Partial solutions for the differential path up to A_{16} are free
- For $A_{17\ldots??}$, count *path conditions* v. accelerating technique "efficiency"
- Estimate the "critical" step A_{xx} & corresp. production rate
- Multiply by probability that a solution A_{xx} extends to A_{80}

SHA-1 collisions recap

On the way to full practical attacks

What complexity for an attack

Conclusion & Future work

Best practical attack progress (2005-2011)

- 2005 (Biham & al.): 40 steps (cost: "within seconds")
- 2005 (Wang & al.): 58 steps (cost: $\approx 2^{33}$ SHA-1 computations)
- 2006 (De Cannière & Rechberger): 64 (cost: $\approx 2^{35}$)
- 2007 (Rechberger & al.): 70 (cost: $\approx 2^{44}$)
- 2007 (Joux & Peyrin): 70 (cost: $\approx 2^{39}$)
- 2010 (Grechnikov): 73 (cost: $\approx 2^{50.7}$)
- 2011 (Grechnikov & Adinetz): 75 (cost: $\approx 2^{57.7}$)

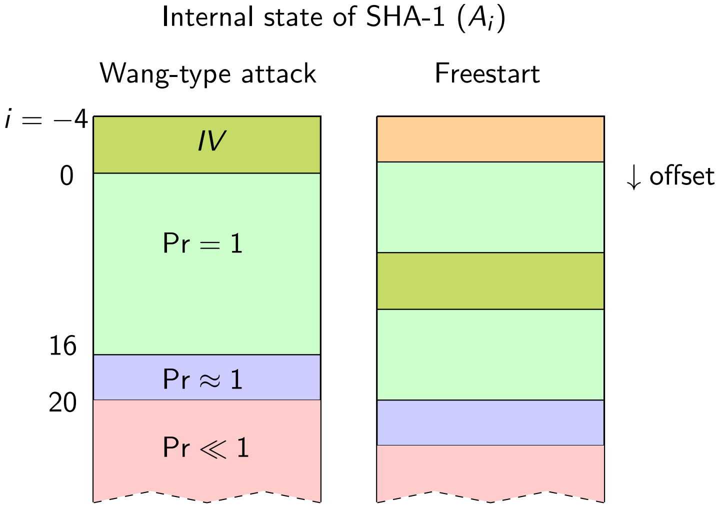
2014: time to improve things again!

- Eventual objective: full practical collision??
- Significant intermediate step: full practical *freestart* collision?
  - Easier in principle, but is it the case?

$\Rightarrow$

- Search for a 76-step freestart collision (lowest # unattacked steps)
- Use the opportunity to develop a GPU framework

The point of freestart (in a picture)



First results

In Dec. 2014: a first 76-step freestart collision (with Peyrin & Stevens)

- Right on time for the ASIACRYPT rump session :P
- Cost: $\approx 2^{50}$ SHA-1 computations *on a GTX-970* $\Rightarrow$ Freestart helps!
- $\Rightarrow$ About 4 days on a single GPU (what we did)
- $\Rightarrow$ About 1 day on a S$ 3000 4-GPU machine

Now what?

Objective: full compression function collision

- Early (optimistic?) estimates: full freestart $\approx 32\times$ more expensive than 76-step
- (Hard to know for sure w/o implementing it)
- $\Rightarrow$ buy (a bit) more GPUs!
- + develop a new attack (“sadly” necessary)
  - Update path search tools
  - Settle on a linear path
  - Generate new attack parameters
  - Program the attack again
  - ...

Let's do this!

Figure: Part of a homemade cluster to be

Second results

In Sep. 2015: a first 80-step (full) freestart collision (with Stevens & Peyrin)

- Right on time for EUROCRYPT submissions :P
- cost: $\approx 2^{57.5}$ SHA-1 computations on a GTX-970
  - A bit more than expected
- $\Rightarrow$ About 680 days on a single GPU
- … or 10 days on a 64-GPU cluster (what we did)
- … or US$ 2000 of the cheapest Amazon EC2 instances

Some early impact

- SHA-1 TLS certificates are *not extended* through 2016 by CA/Browser forum actors
  - Ballot 152 (Oct. 2015!) of the CA/Browser forum is withdrawn
- Some major browsers (Edge, Firefox) sped-up deprecation/security warnings
- But (some) continued use in Git, company-specific certificates (e.g. Facebook until Dec. 2016, Cloudflare), etc.
  - Mostly because of legacy issues

Now what?

Objective: full hash function collision

- Early (optimistic?) estimates: full collision $\approx 50\times$ more expensive than full freestart
- (Hard to know for sure w/o implementing it)
- $\Rightarrow$ buy a lot more GPUs? (No)
- $\Rightarrow$ get help from GPU-rich people/companies? (Yes)
- + develop a new attack
- + add some cool exploitation features!

Let's do this!

A CWI/Google collaboration

1. Prepare a prefix for future colliding PDFs
2. Compute a first (actually two) near-collision block(s)
   - Done on CPU
3. Compute a second near-collision ⇒ the final one!!
   - Done on GPU
4. ~~Profit!~~ Enjoy!

- cost: $\approx 2^{63}$ SHA-1 computations
  - A bit more/less than expected
- ⇒ about 6 500 CPU-year + 100 GPU-year
- … or US$ 100K+ of the cheapest Amazon instances (second block only)

Some more impact

- Finally got Git planning to move away from SHA-1
- Unwittingly broke SVN for a time
- Further deprecation of SHA-1 certificates

SHA-1 collisions recap

On the way to full practical attacks

What complexity for an attack

Conclusion & Future work

Absolute cost v. "complexity"

- Determining the complexity of generic attacks is "easy"
- E.g. $\Theta(2^{n/2})$ for collisions on n-bit hash functions
  - Efficiently parallelizable (van Oorschot & Wiener, 1999)
- What about dedicated attacks?
  - Implement and measure?

A typical metric for cryptanalysis complexity:

1. Estimate the cost of an attack *on some platform*
2. Divide by the cost of computing the attacked function
3. Voilà

A '76 complexity example

Example: 76-step freestart collision
On a GTX-970:

- Expected time to collision = 4.4 days
  - 0.017 solution up to A_{56}/s
- $\approx 2^{31.8}$ SHA-1 compression function/s
- $\Rightarrow 4.4 \times 86400 \times 2^{31.8} \approx 2^{50.3}$

BUT on an Haswell Core i5:

- Expected time to collision = 606 core days
  - 0.000124 solution up to A_{56}/s
- $\approx 2^{23.5}$ SHA-1 compression function/s
- $\Rightarrow 606 \times 86400 \times 2^{23.5} \approx 2^{49.1}$
- Yet much slower & less energy efficient!!

A full hash example

Complexity for the full hash function (second block) collision:

- $2^{62.1}$ on K80, or
- $2^{62.8}$ on K20/40, or
- $2^{63.4}$ on GTX-970

Further code tuning/optimization may again change figures!

Some more issues

- Variation between CPU/GPU and optimized/unoptimized is not so large
  - About $\times 2$–4
- What about reconfigurable/dedicated hardware?
  - FPGA/ASICs are fast and energy efficient
  - $\Rightarrow$ Well-suited to generic attacks!
  - But what about complex ones???
- No reason for a generic attacker to use CPU/GPU over FPGA/ASIC
  - Potential increased development cost well worth it!
- What does a dedicated attack really improve on??

GPU v. ASIC brute force estimates

One generic SHA-1 collision in one year $\approx 2^{80}$ hash computations
On GPU:

- $\approx$ 12.6 million GPUs @ $2^{31.5}$ hashes/s
- $\approx$ 3.1 GW 'round the clock (just the GPUs @ 250 W each)
  - A couple of dedicated nuclear powerplant needed

On ASIC (estimates courtesy of BTC mining hardware)

- $\approx$ 2900 devices @ $2^{43.6}$ hashes/s (Antminer S9-like)
- $\approx$ 4 MW 'round the clock (at 1400 W each)
  - About a large wind turbine needed (with the wind)

An alternative cost measure: The fun calorie

- Introduced by A. Lenstra, Kleinjung & Thomé (2013):

  How much energy is ~~wasted~~ needed by an attack?

- Energy unit: "fun calorie"

  What volume of standard water can you boil (instead)?

- Used to estimate e.g. RSA-768 security

  $\Rightarrow$ 2 olympic pool security (Kleinjung et al., 2010)

Some complexity figures

SHA-0 collision (MP08)	$\lessapprox$ teaspoon sec. (2.5×10^{-3}L)
SHA-1 76' fs.	$\approx$ 4 shower sec. (320L)
SHA-1 fs.	$\approx$ 580 shower sec. (4.5×10^{4}L)
SHA-1 2^{nd} block (ded, GPU)	$\approx$ 1 pool sec. (2.5×10^{6}L)
RSA-768 (K+10)	$\approx$ 2 pool sec. (5×10^{6}L)
SHA-1 1^{st} block (ded, CPU)	$\approx$ 3 pool sec. (7.5×10^{6}L)
DL-768 (K+17)	$\approx$ 6 pool sec. (1.5×10^{7}L)
SHA-0/1 (gen, ASIC)†	$\approx$ 0.004 rain sec.‡ (3.5×10^{8}L)

(Ignoring CPU improvements between 2010 and today)

†: Estimate

‡: *dagelijkse neerslagverdampingenergiebehoeftezekerheid*

In the end...

- Full-GPU dedicated SHA-1 attack: ≈ 1 pool sec.
- $\Rightarrow \approx 100\times$ better than dedicated hardware (conjectured)
- Quite less than $2^{80-63} \approx 130\,000$

SHA-1 collisions recap

On the way to full practical attacks

What complexity for an attack

Conclusion & Future work

Potential future work

- Computing a *chosen-prefix* collision
  - More exploitation
- Computing a collision for the SHA-1||MD5 combiner
  - Wouldn't break SVN?
- Designing a SHA-1-based crypto-currency
  - Get shiny mining hardware!

For more details

- The papers: Eprints 2015/530, 2015/967, 2017/190
- The attack code: `https://github.com/cr-marcstevens/sha1_gpu_nearcollisionattacks`
- Marc's talk @ CRYPTO'17
- Ange's talk @ BlackAlps'17

C'est fini!