Real World Cryptanalysis

Marc Stevens
Cryptology Group
CWI Amsterdam
Real World Cryptanalysis
Real World Cryptanalysis

Cryptographic Standards crucial for secure Internet

Gain confidence in security over time through extensive scrutiny (Before & After Standardization)
Real World Cryptanalysis

Cryptographic Standards crucial for secure Internet

Occasionally leap in cryptanalysis exposes unknown weaknesses

😢 Deprecate Standard!
```c
uint32_t w14_q20_nb = 0;
for (unsigned l = 0; l < (1<<2); ++l)
{
    NEXT_NB(w14_q20_nb, W14NBQ20M);
    m14 &= ~W14NBQ20M;
    m14 |= w14_q20_nb;
    q15 += rotate_left(w14_q20_nb, 5);
    m17 ^= rotate_left(m14, 1);
    m20 ^= rotate_left(m17, 1);
    uint32_t w15_q20_nb = 0;
    for (unsigned k = 0; k < (1<<5); ++k)
    {
        NEXT_NB(w15_q20_nb, W15NBQ20M);
        m15 &= ~W15NBQ20M;
        m15 |= w15_q20_nb;
        m18 ^= rotate_left(m15, 1);
    }
}
```
Cryptographic hash functions
Cryptographic hash functions

A hash function is a deterministic mapping from arbitrary length inputs to a fixed length output.

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).

<table>
<thead>
<tr>
<th>MDC-2</th>
<th>MD2, MD4, MD5</th>
<th>RIPEMD-160</th>
<th>SHA-3-{256,512}</th>
</tr>
</thead>
<tbody>
<tr>
<td>old &amp; weak</td>
<td></td>
<td>SHA-2-{256,512}</td>
<td>New Std in 2015</td>
</tr>
<tr>
<td></td>
<td></td>
<td>secure</td>
<td>NIST: use SHA-2</td>
</tr>
</tbody>
</table>

4e1243bd22c66e76c2ba9eddc1f9  
hash / digest / fingerprint
Collision resistance

Find \( m \neq k \) such that \( H(m) = H(k) \)

Only max. \((n/2)\)-bit security!

128-bit hash \( \Rightarrow \) 64-bit security
160-bit hash \( \Rightarrow \) 80-bit security
256-bit hash \( \Rightarrow \) 128-bit security
512-bit hash \( \Rightarrow \) 256-bit security

Note:

**Bitcoin** network computes
- \( 2^{64} \) SHA-2 / sec
- \( 2^{80.5} \) SHA-2 / day
- \( 2^{84} \) SHA-2 / 12 days
- \( 2^{89} \) SHA-2 / year

\( \Rightarrow \)
breaks 80-bit security
brute-force in 1 day!
Design of hash functions

- **Merkle-Damgård Construction**
  - Splits message into 512-bit blocks
  - Processes blocks iteratively using compression function

- **Security reduction**
  - Collision hash function $\Rightarrow$ collision compression function
MD5 / SHA-1 / SHA-256 compression function

MD5
very weak
permutation
message expansion
16 x 32-bit → {64,80} x 32-bit

SHA-1
theoretically weak
linear recurrence

SHA-2
secure
non-linear
Differential cryptanalysis

- Consider two different instances
  \[ \text{Compress}(CV, M) \]
  \[ \text{Compress}(CV', M') \]
- Analyze differences

Differential path

- Precise description of all differences propagating through compression function
- Translate differential path into \textbf{system of equations} to solve to find \( M, M' \)
Differential cryptanalysis

**System of equations**
- Sufficient system of equations: Applying the input differences guarantees diff.path
- Simple equations on message and state bits

**Solve**
- First 16 steps easily solved \( \Rightarrow \) all message bit equations fulfilled
- Make predictable small changes to solve up to step 24 (amortizes cost of earlier steps)
- Probabilistically fulfill remaining steps (with many solutions up to step 24)
Deprecating MD5 in 2008

(known to be practically broken since 2004)

Joint work with:
Alexander Sotirov, Jacob Applebaum, Arjen Lenstra, David Molnar, Dag Arne Osvik, Benne de Weger
2008 [SSALMOdW]: Breakthrough on MD5
- Practical *chosen-prefix collision* attack on MD5
- Arbitrary different prefixes made to collide
Example chosen-prefix collision between

- Our website:  
  https://i.broke.the.internet.and.all.i.got.was.this.t-shirt.phreedom.org

- A hand-crafted sub-C.A. certificate 😈

- Valid signature for both from Verisign!
Rogue C.A.

- Realistic Man-in-the-Middle attack against any secure website
- Responsible disclosure
  - Pre-informed Browsers and C.A.s
  - Rogue C.A. purposely crippled: only valid in August 2004
- MD5 deprecated within hours
- Software released for research
  - Anyone can create chosen-prefix collisions
  - \( \approx 1 \) day on quadcore machine
  - https://github.com/cr-marcstevens/hashclash
What happened since?

- 2009: CABforum: MD5 deprecated for signatures

- 2012: supermalware Flame uses forged MD5 signature to push fake Windows Updates
  Discovery of yet-unknown variant MD5 collision attack

- 2016: SLOTH: Transcript collision attacks against TLS, IKE, SSH

- 2017: Oracle JRE rejects MD5 signatures
  Originally planned for January, was postponed till April

- 2018: US SWGDE (Scientific Working Group on Digital Evidence)
  Publication “explains that the use of the MD5 and SHA1 hash algorithms remains acceptable”
Deprecating SHA-1 in 2017

(known to be weak since 2005)

Joint work with
Ange Albertini, Elie Bursztein, Pierre Karpman, Yarik Markov
Weaknesses

SHA-1 is not collision resistant

Collision attack with complexity $2^{69}$ (4M core-years) [WangYY 2005]
Later improved to $2^{61}$ (15,300 core-years) [Stevens 2012]

Projected costs of SHA-1 collisions [Schneier 2012]

$2.77M$ in 2012
$700K$ by 2015
$173K$ by 2018 $\Rightarrow$ “we can postpone 5 years..”
$43K$ by 2021

(based on [Stevens 2012], Amazon EC2 rates & Moore’s Law)

Practical SHA-1 collision remained open problem
GPU >> CPU

- [§13]: SHA-1 collision attack with complexity $\approx 2^{61}$
  - $\Rightarrow$ CPU attack: 15.3K coreyears

- [SPK16]: attack complexity $\approx 2^{62.2}$ on GTX-970
  - $\Rightarrow$ GPU attack: 112 GPUyears
  - $\approx$ $100k$ renting fee (on Amazon EC2)
  - $\times 7$ lower cost in 2015 than predicted earlier by Schneier
  - Initiated collaboration with Google
First SHA-1 Collision

- **Collaboration with Google [SBKAM17]**
  - Google Infrastructure:
    - Large heterogenous cluster of CPUs & GPUs
  - But: no direct access, proprietary Compile & Job system
  - ‘Blind’ adaptation source-code by Google

- **First near-collision attack**
  - Took 3583 core years \(\approx 2^{60}\) SHA-1 compressions
  - Run on 100k+ PCs in several weeks

- **Second near-collision attack**
  - Tailored to 1st NC output
  - Using NVIDIA Tesla K20, K40, K80
  - Took \(\approx 114\) K20years \(\approx 71\) K80years \(\approx 2^{62.8}\) SHA-1
  - Run on >3000 GPUs in just 8 calendar days

- Collision on [https://shattered.io/](https://shattered.io/)
**Reusable meaningful** SHA-1 collision:
- 1 collision: infinite colliding PDF-pairs with distinct embedded JPGs
- Use JPG for page-content ⇒ arbitrary distinct page contents
- Use PDF image cropping ⇒ arbitrary distinct multi-page contents

- DIY: https://github.com/nneonneo/sha1collider
Impact

• Project Webpage, Google Drive & Gmail check for SHA-1 collisions

• Unexpectedly collision can break Subversion repositories
  • Webkit developer submitted test to prove WebKit resistant to SHA-1 collisions
  • Broke Webkit repository
  • Internal deduplication uses SHA-1 and keeps only 1 colliding file
  • MD5 is used to check integrity ⇒ will always fail on checkout

• Git started moving away from SHA-1
• Git & GitHub now using strengthened SHA-1 implementation by default

• CA/Browser Forum: Ballot 152
  • Extend issuance SHA-1 certificates up to 1 Jan. 2017 (before: 1 Jan. 2016)
  • (unaltered: deprecate SHA-1 certificates after 1 Jan. 2017)
  • Our recommendations on 8 Oct. ensured Ballot did not pass on 16 Oct.

• TLS 1.3 draft 9
  • Deprecated all uses of SHA-1 digital signatures
From attacks to toys
Instant Collisions

• Instant collision scripts for many file formats
  • Instant, re-usable and generic collisions
  • Take any pair of files, run script, get colliding files

• SHA-1: PDF, HTML
• MD5
  • PDF
  • PNG, JPG, JP2
  • MP4, GIF
  • PE (windows executable)
Toys

PoC || GTFO = PDF + HTML + ZIP

same file

AN INSTANT COLLISION OF:
- A DOCUMENT
- AN EXECUTABLE
- AN IMAGE
- A VIDEO.

generates

All these files have the same MD5

https://github.com/angea/pocorgtfo#0x19
Hashquines: documents that show their own MD5 hash

Toys

Hashquines: documents that show their own MD5 hash

Toys

Hashquines: documents that show their own MD5 hash

Toys
Thank you!