Research at Google and CWI

How we created the first SHA-1 collision and what it means for hash security

Elie Bursztein

with the help of Marc Stevens (CWI), Pierre Karpman (INRIA), Ange Albertini, Yarik Markov, Alex Petit-Bianco

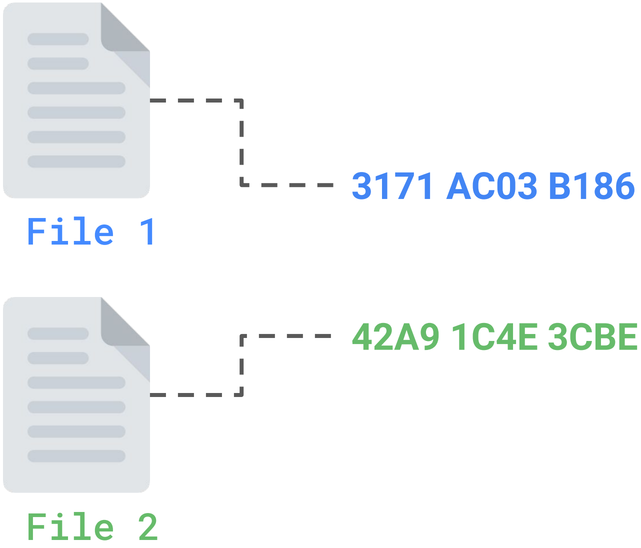
What is a **cryptographic hash function?**

Digest uniqueness

Different files hash to different digests (no collision)

One-way function

Digest reveals no information about the file hashed



What are secure hash **functions used for?**

Document & software signing

HTTPS certificate signing

Version control integrity

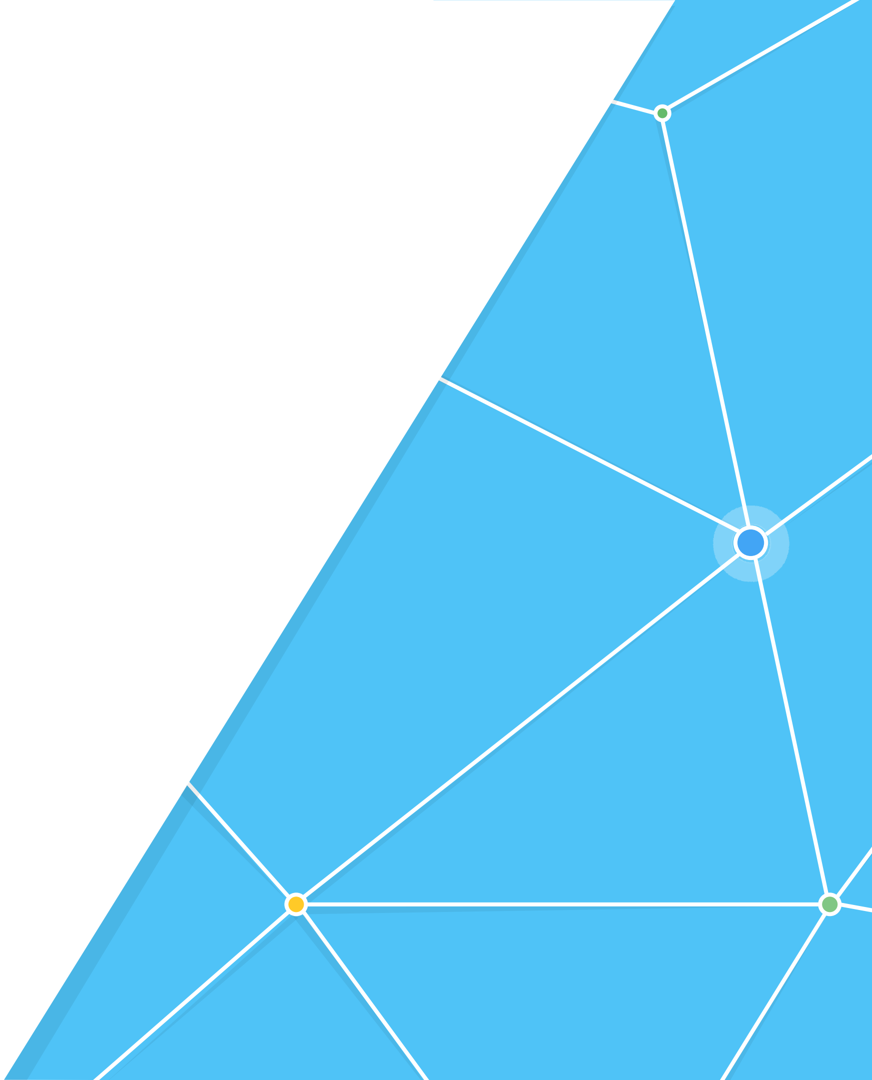
Backup integrity

Agenda

Attacking hash functions
Primer on hash function attacks

Finding a SHA-1 collision
How we did it

Post-collision world
Legacy software & hash function future

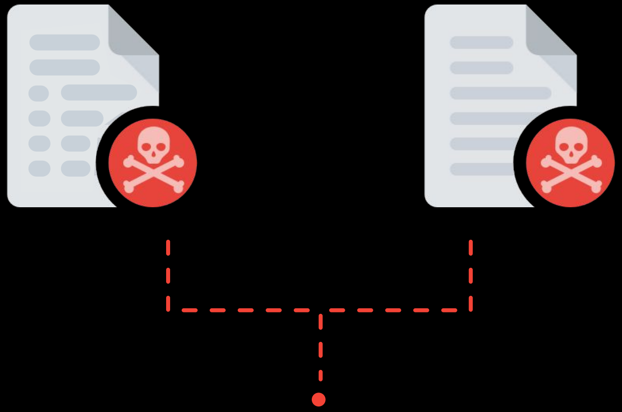


Completing the puzzle

Attacking hash functions

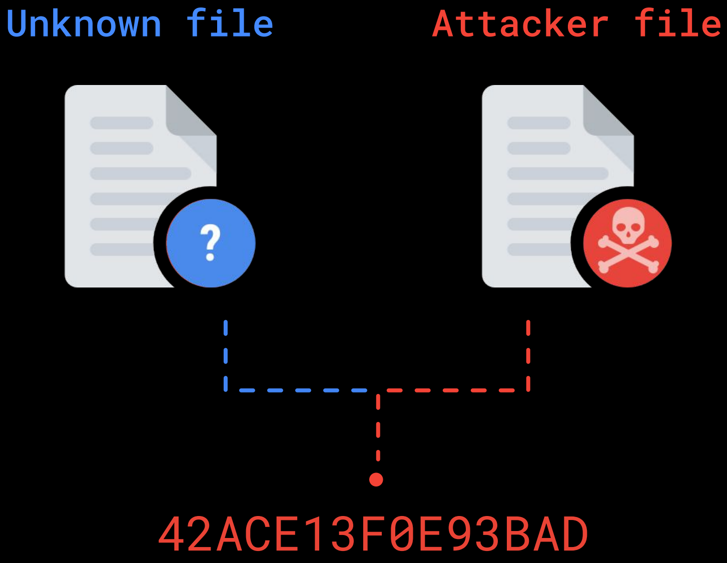
Collision attack

Attacker file 1 Attacker file 2

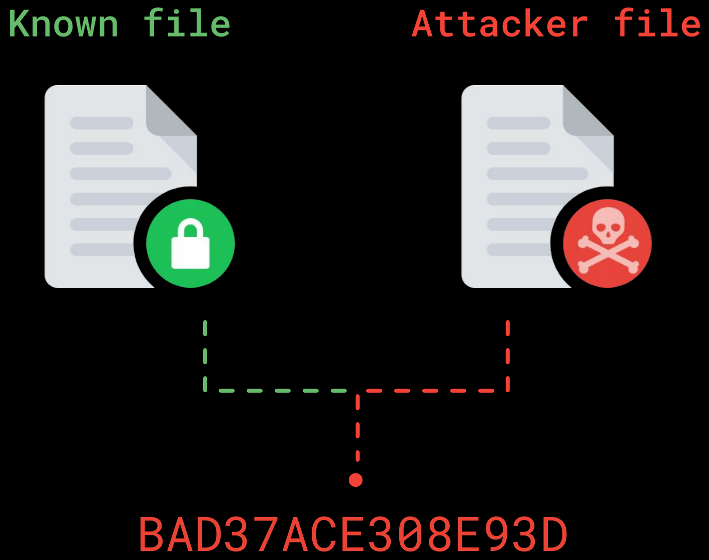


3713ACE30E7ABBA

Preimage attack



Second preimage attack



How to create a collision attack

The need for cryptanalysis

SHA-1 bruteforce

12,000,000 GPU
1 year

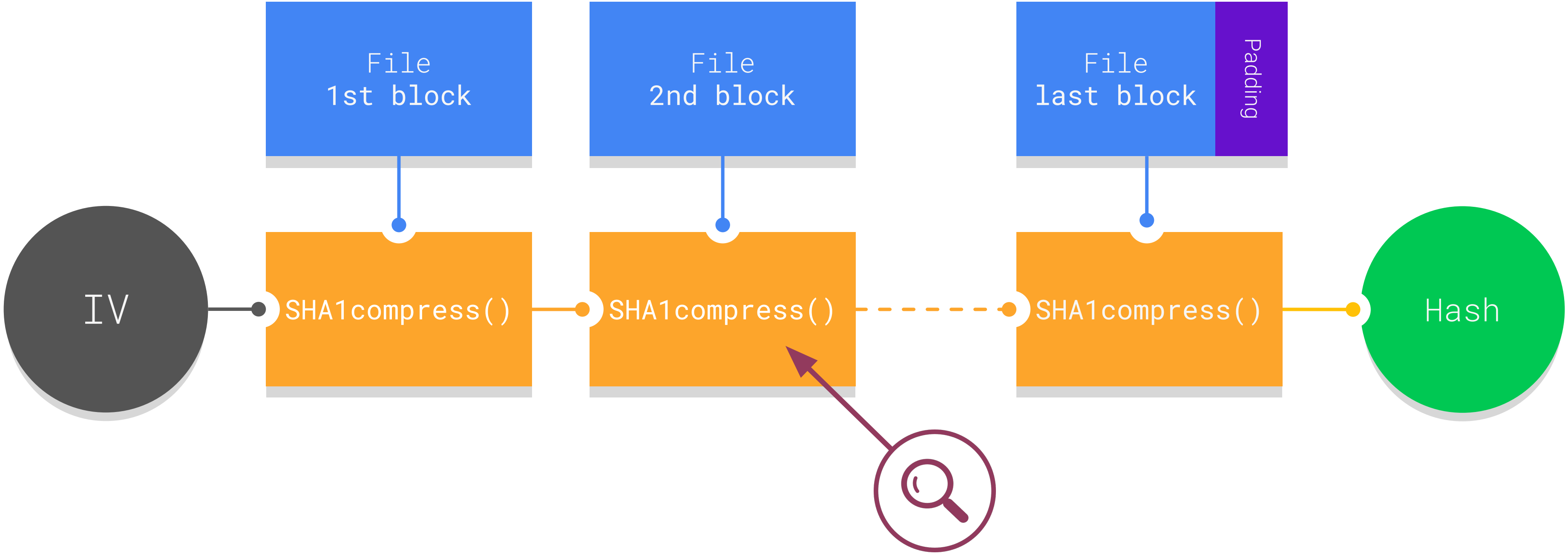
Bruteforce is impractical

Even with GPU you can’t create a collision using bruteforce

Cryptanalysis to the rescue

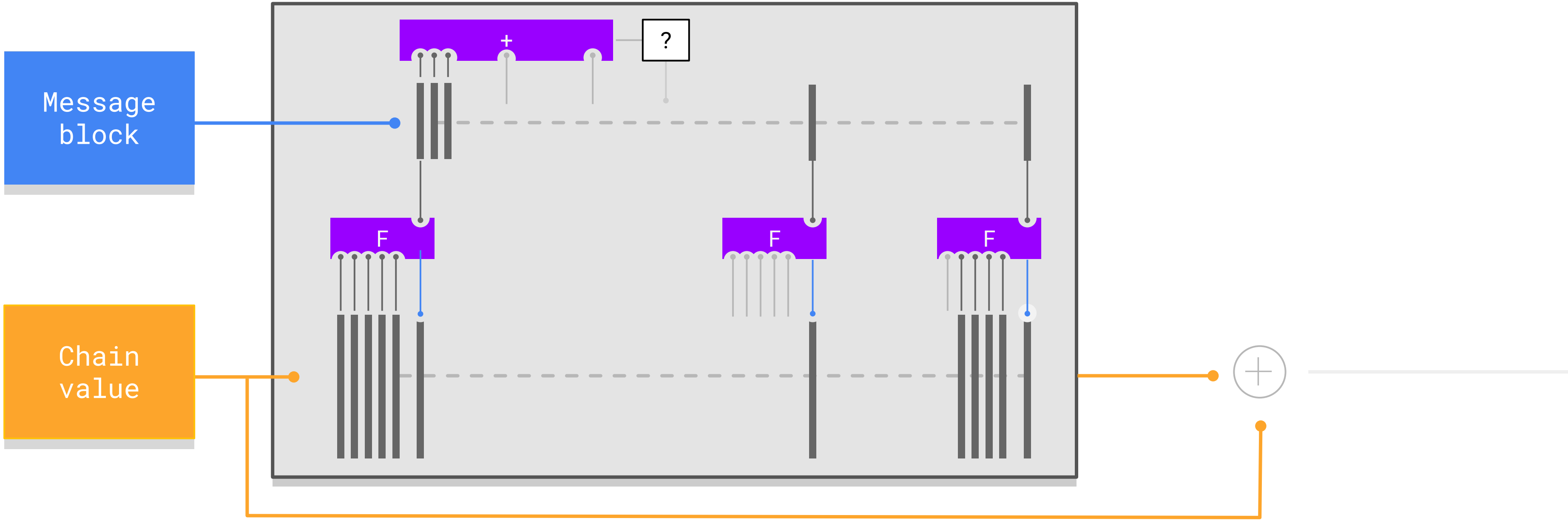
Cryptanalysis techniques are used to reduce the attack complexity to a point where it became feasible

The Merkle–Damgård construction

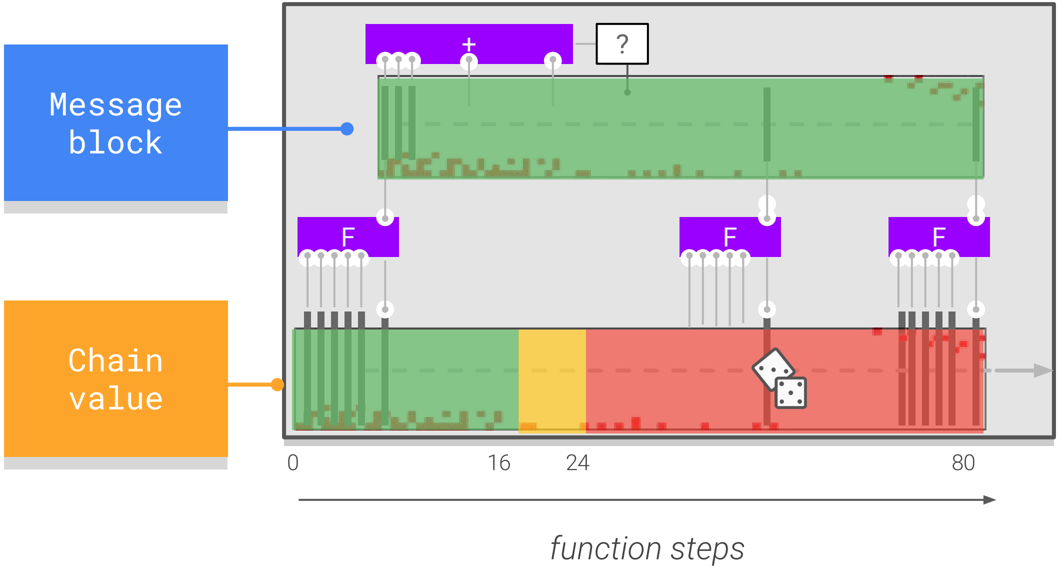


R.C Merkle - Secrecy, authentication, and public key systems (1979)

Unrolled SHA-1 compress function



SHA-1 cryptanalysis in a nutshell



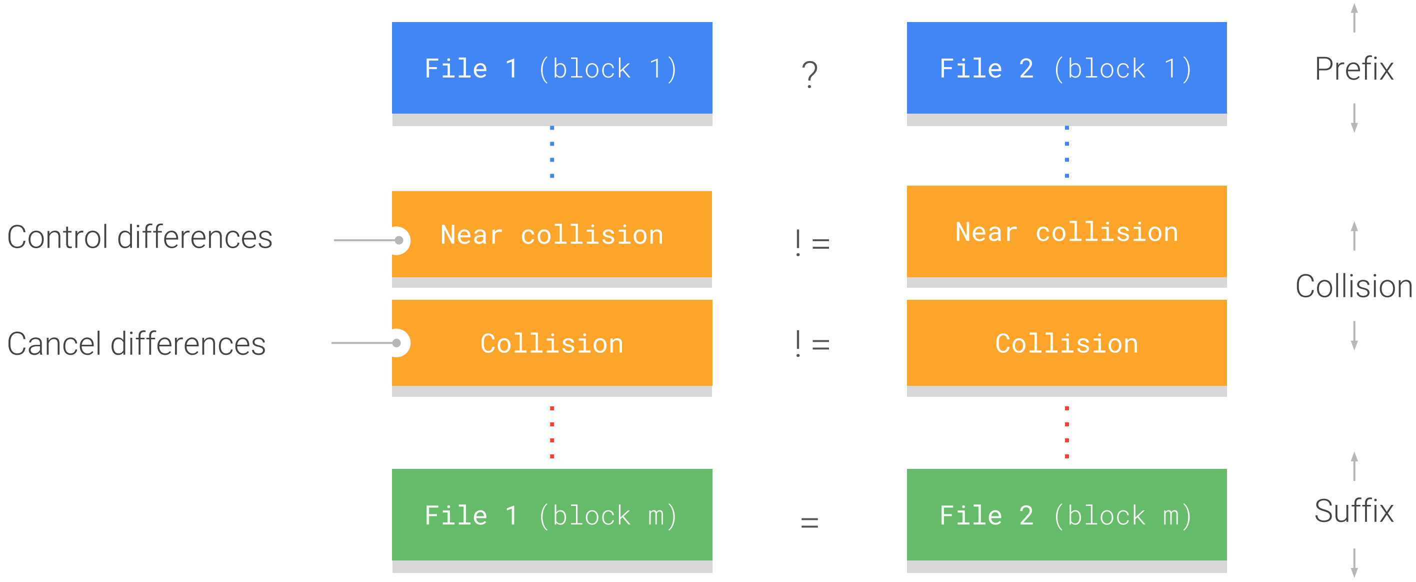
Messages differential path

Describe how differences propagate

Equation system

16 steps solved, predictable till step 24

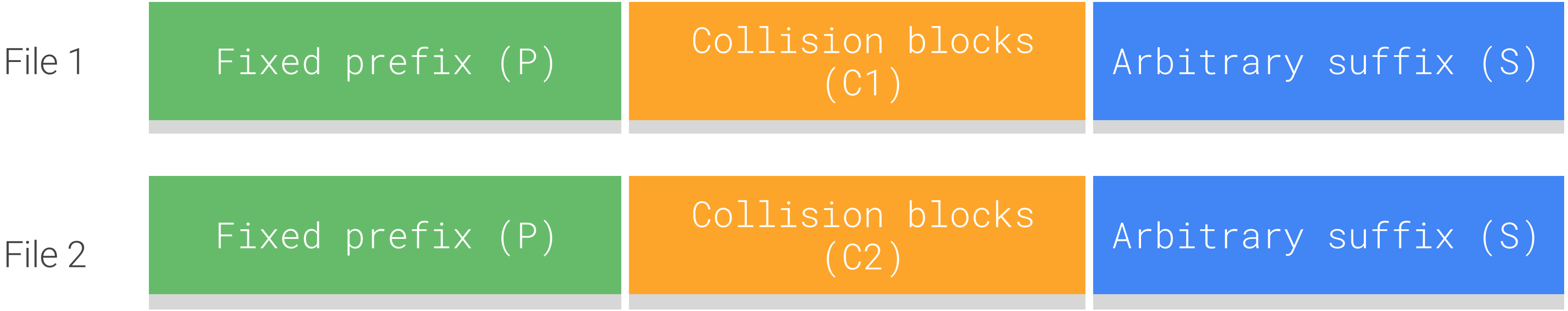
Two block collision



Wang et al. - Finding Collisions in the Full SHA-1 (2005)

Exploiting collisions

Fixed prefix attack (SHA-1)

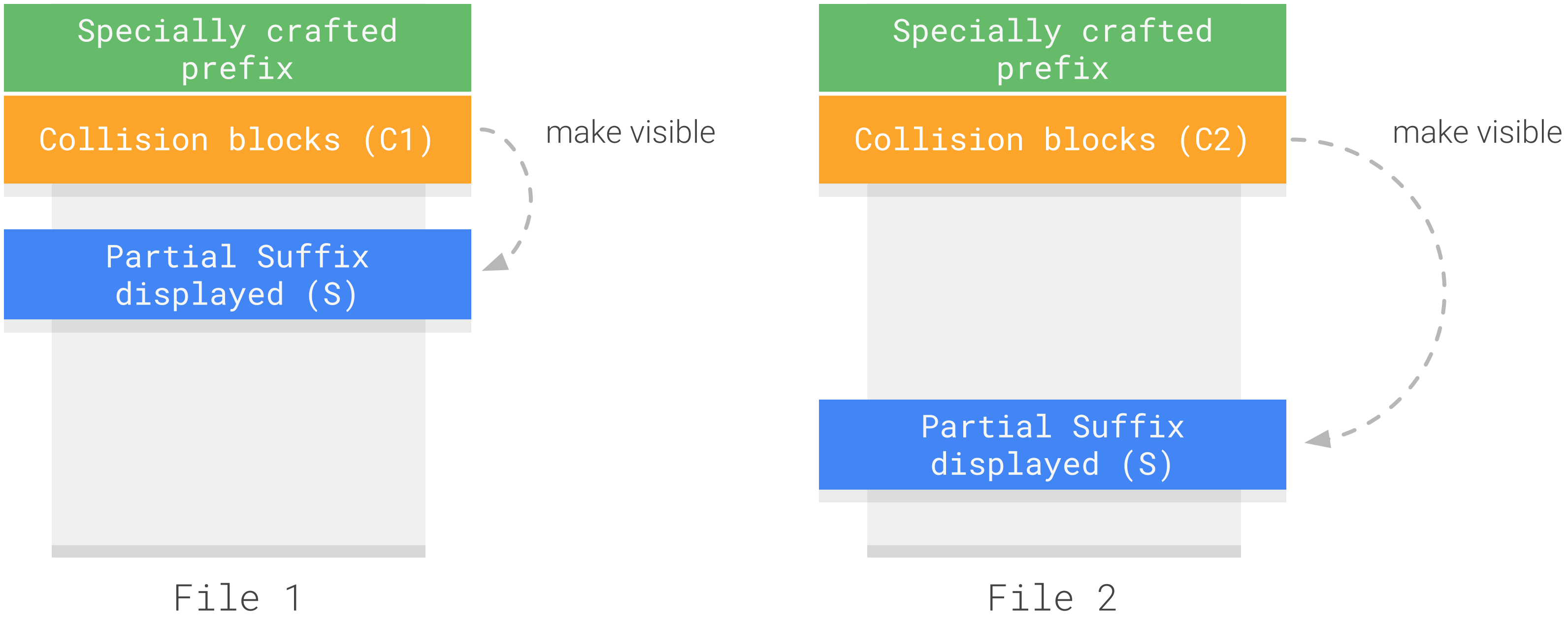


File 1	Fixed prefix (P)	Collision blocks (C1)	Arbitrary suffix (S)
File 2	Fixed prefix (P)	Collision blocks (C2)	Arbitrary suffix (S)

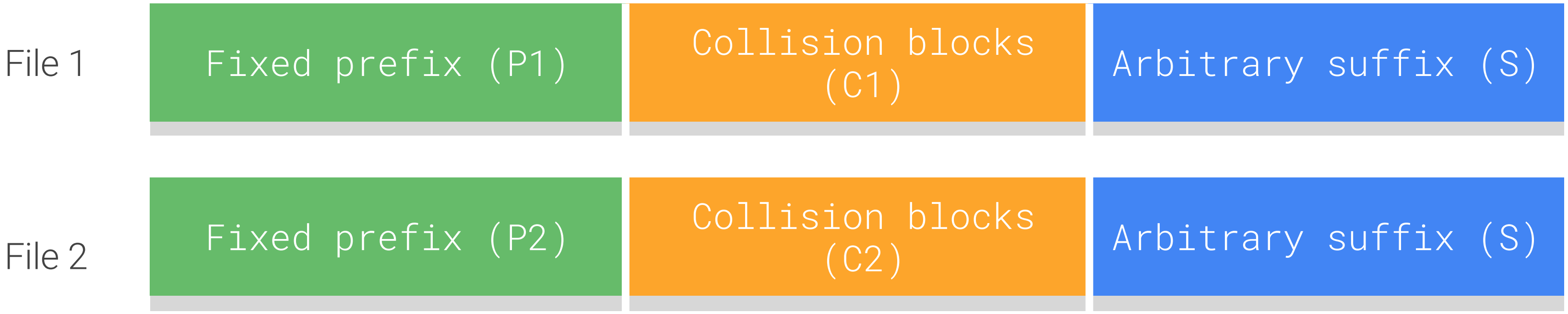
```
P==P and C1!=C2 and S==S
```

Used in the SHA-1 collision

Carefully choosing prefix to improve attack



Chosen-prefix (MD5)



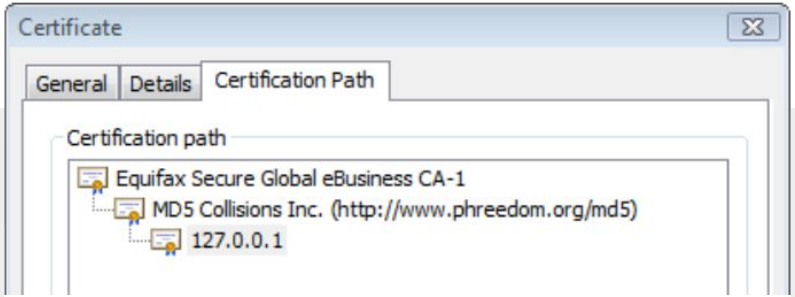
File 1	Fixed prefix (P1)	Collision blocks (C1)	Arbitrary suffix (S)
File 2	Fixed prefix (P2)	Collision blocks (C2)	Arbitrary suffix (S)

P1!=P2 and C1!=C2 and S==S

Used in the MD5 collision attacks

Real world attacks exploiting MD5 collision

Chosen-prefix: **MD5 SSL certificate forgery**

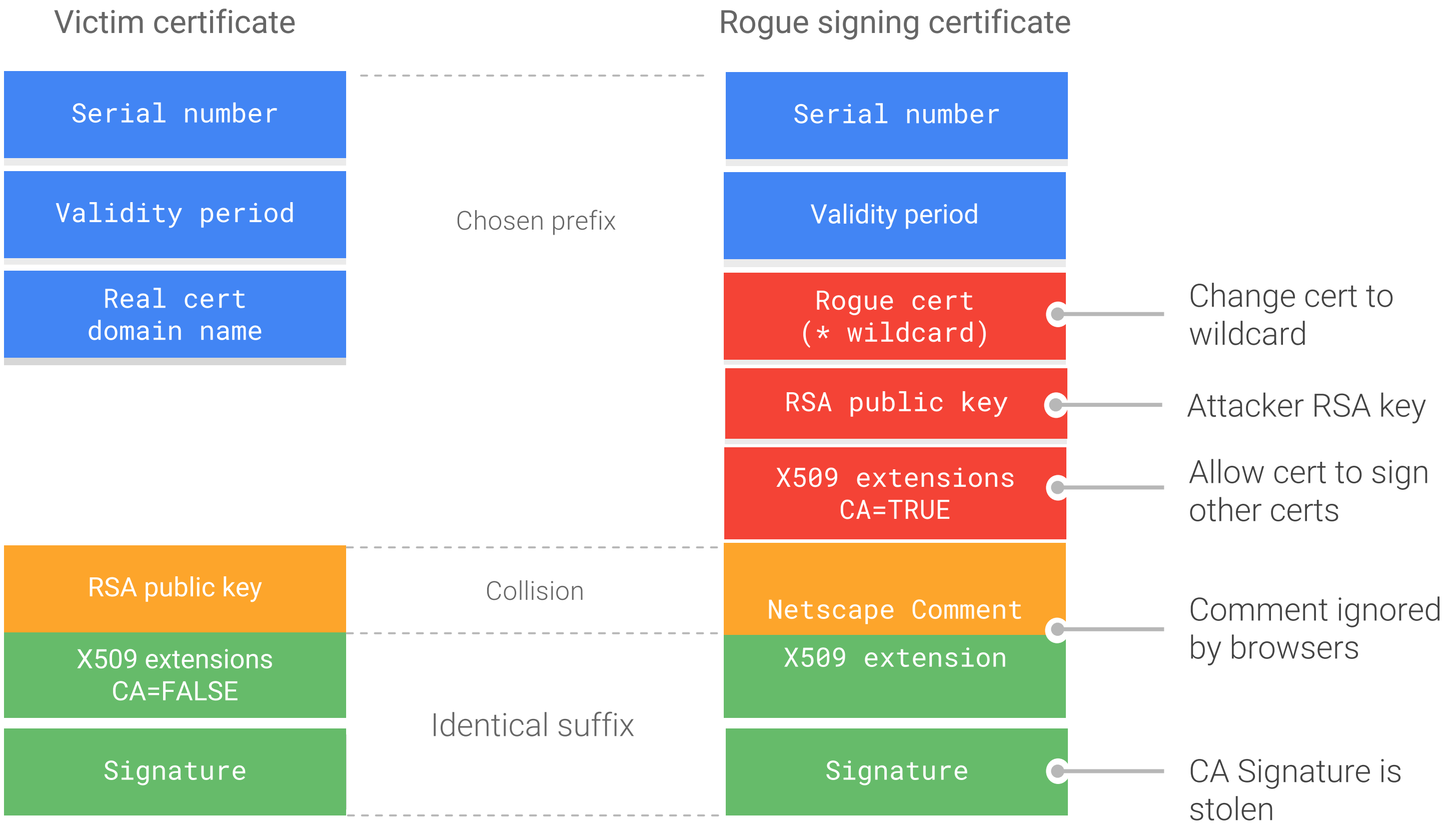


Certification path

Equifax Secure Global eBusiness CA-1
MD5 Collisions Inc. (http://www.phreedom.org/md5)
127.0.0.1

Rogue SSL certificate

Cluster of 200 PlayStation 3 used to compute the MD5 rogue certificate



Victim certificate

Serial number

Validity period

Real cert domain name

RSA public key

X509 extensions CA=FALSE

Signature

Chosen prefix

Collision

Identical suffix

Rogue signing certificate

Serial number

Validity period

Rogue cert (* wildcard)

RSA public key

X509 extensions CA=TRUE

Netscape Comment

X509 extension

Signature

Change cert to wildcard

Attacker RSA key

Allow cert to sign other certs

Comment ignored by browsers

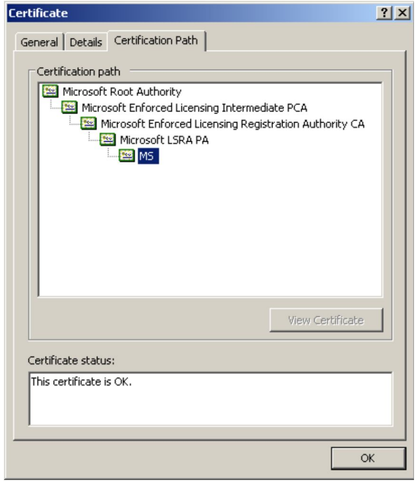
CA Signature is stolen

Stevens et al. - Short Chosen-Prefix Collisions for MD5 and the Creation of a Rogue CA Certificate (2009)

Malware MD5 certificate

MEET 'FLAME,' THE MASSIVE SPY MALWARE INFILTRATING IRANIAN COMPUTERS

WIRED



Certification path

- Microsoft Root Authority
  - Microsoft Enforced Licensing Intermediate PCA
    - Microsoft Enforced Licensing Registration Authority CA
      - Microsoft LSRA PA
        - MS

Certificate status:

This certificate is OK.

Forged windows update certificate

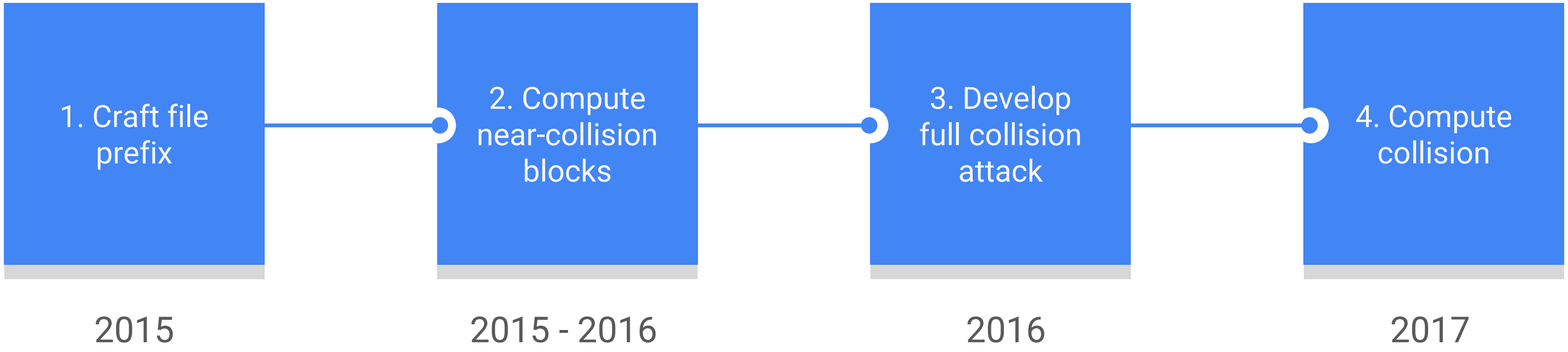
Stevens et al. - Reverse-engineering of the cryptanalytic attack used in the Flame super-malware 2015

Attack feasibility

	Collision resistance			Preimage resistance	
	Security Claim	Fixed prefix	Chosen attack	Security claim	Best attack
MD4	2^{64}	2^{1}		2^{128}	2^{95}
MD5	2^{64}	2^{16}	2^{39}	2^{128}	$2^{123.4}$
SHA-1	2^{80}	$\mathbf{2^{63}}$	2^{77}	2^{160}	-

Finding a SHA-1 collision

Attack overview



1. Craft file prefix

2015

2. Compute near-collision blocks

2015 - 2016

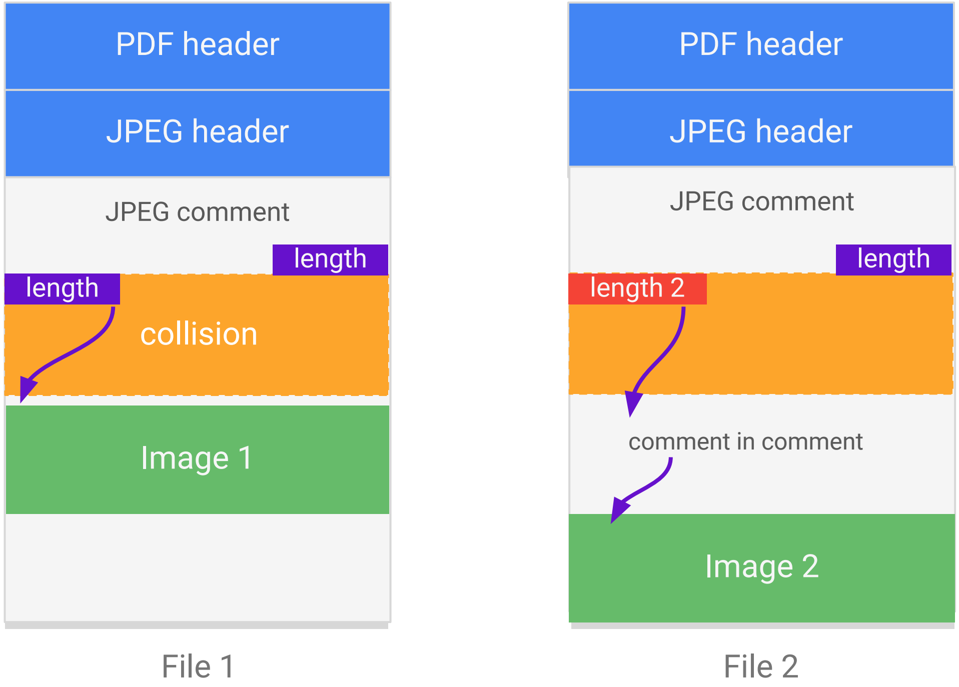
3. Develop full collision attack

2016

4. Compute collision

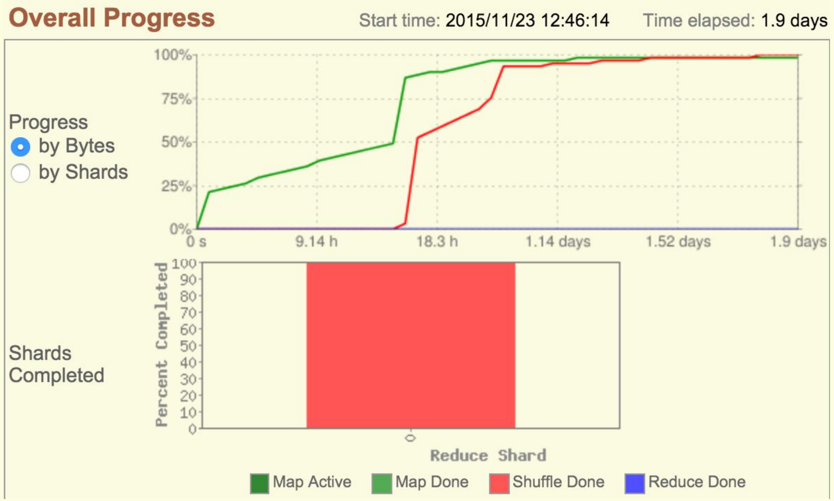
2017

Smart prefix: JPEG embedded in PDF



Some technical details left out for clarity reason

Scaling computation



Overall Progress

Start time: 2015/11/23 12:46:14 Time elapsed: 1.9 days

Progress

● by Bytes

○ by Shards

Work in small batches ~1h

Time is a resilience/performance tradeoff

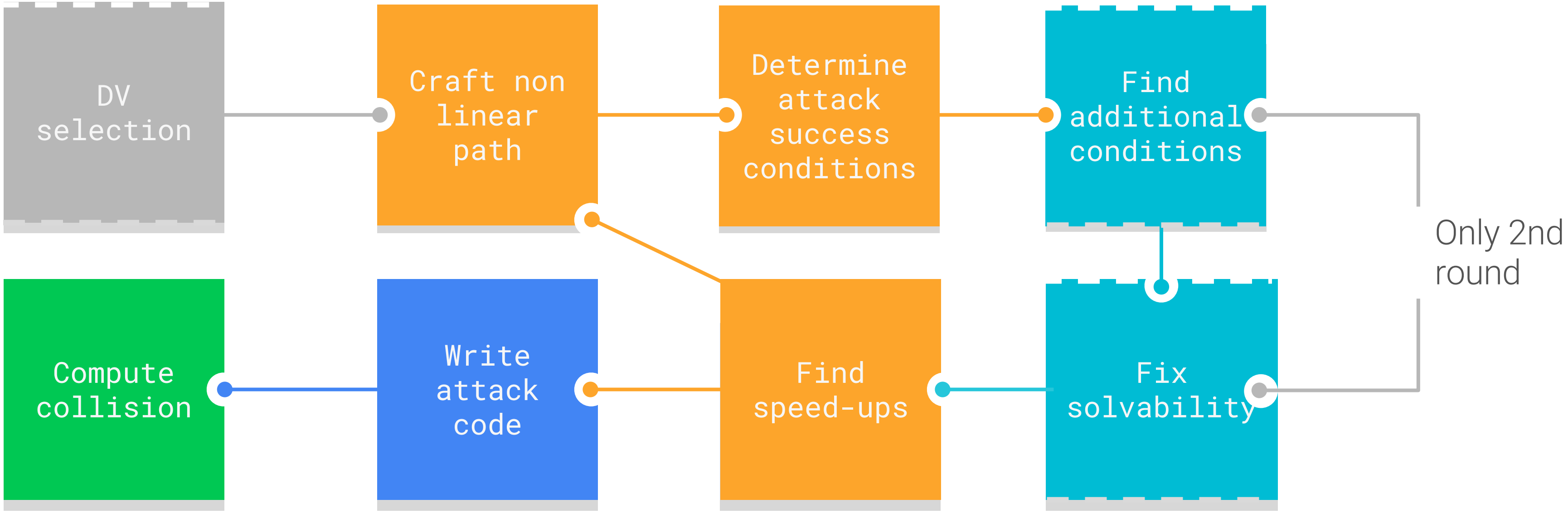
Refactor code to be stateless

Each batch is independent

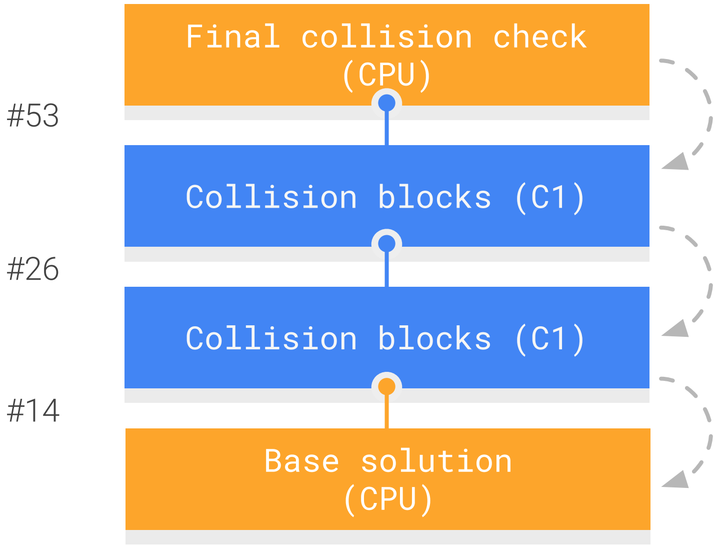
Factory paradigm not map-reduce

Map-reducing causes strangling issues

Developing the full collision attack



Making efficient use of GPUs



Work step by step

Generate enough solutions for next step

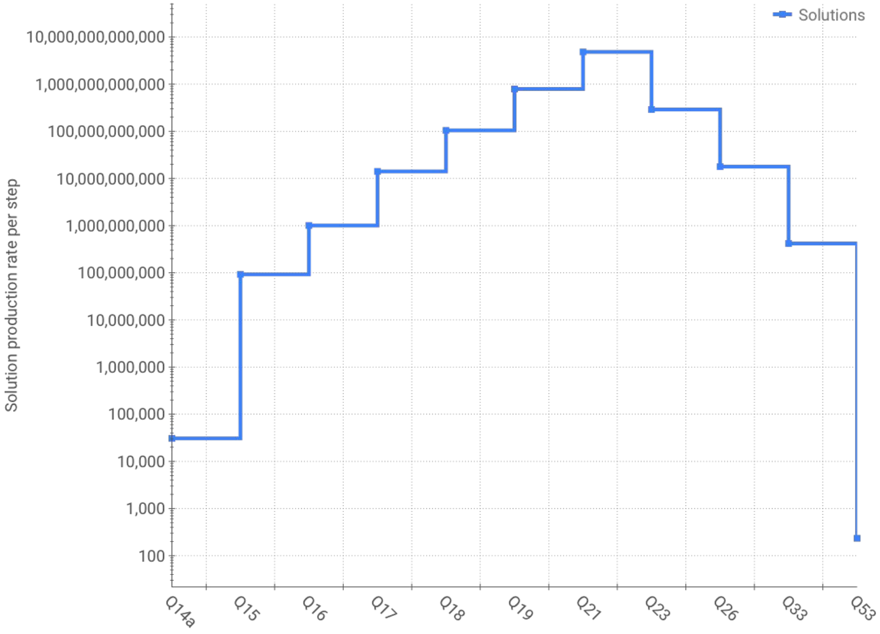
Always try to work at the highest step

Backtrack when pool empty

Parallelized: One thread / one solution

Single instruction - multiple threads

Phase 2 production rate per step



Logarithmic scale

Computational cost comparison

MD5

1 smartphone
30 sec

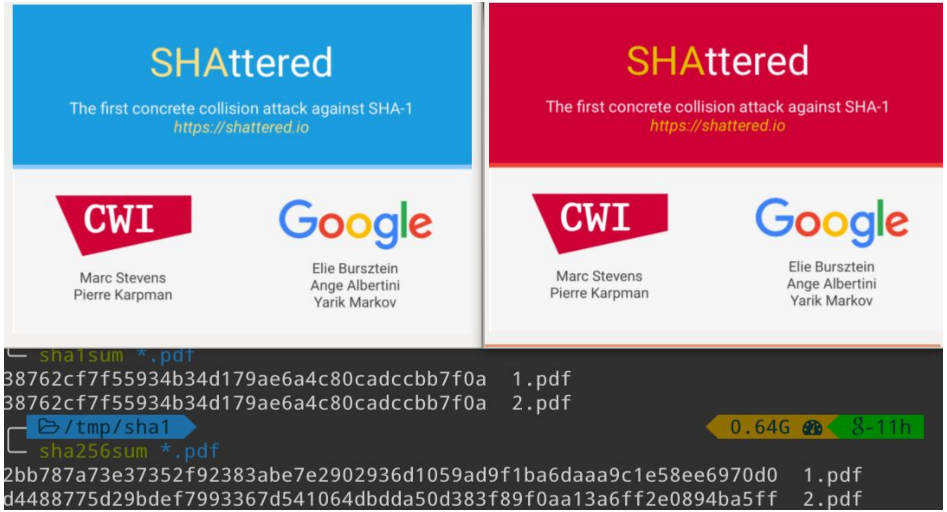
SHA-1 shattered

110 GPU
1 year

SHA-1 bruteforce

12,000,000 GPU
1 year

Colliding PDFs Demo!



SHAttered

The first concrete collision attack against SHA-1
https://shattered.io

CWI

Marc Stevens
Pierre Karpman

Google

Elie Bursztein
Ange Albertini
Yarik Markov

SHAttered

The first concrete collision attack against SHA-1
https://shattered.io

CWI

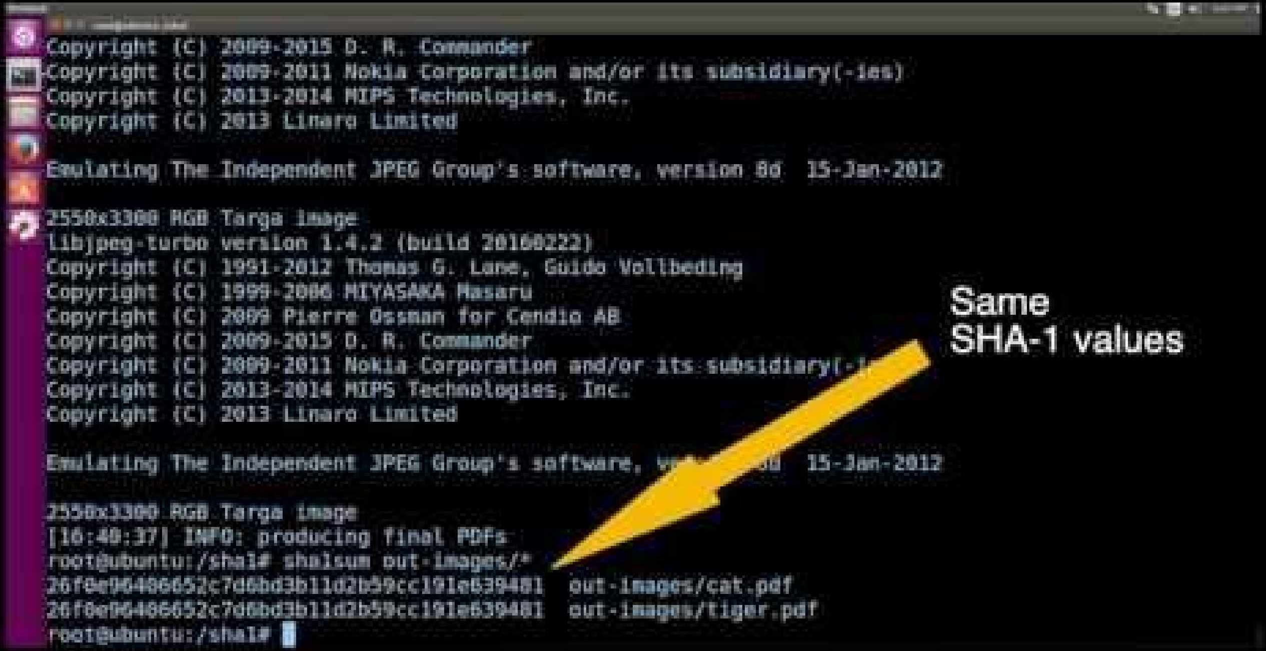
Marc Stevens
Pierre Karpman

Google

Elie Bursztein
Ange Albertini
Yarik Markov

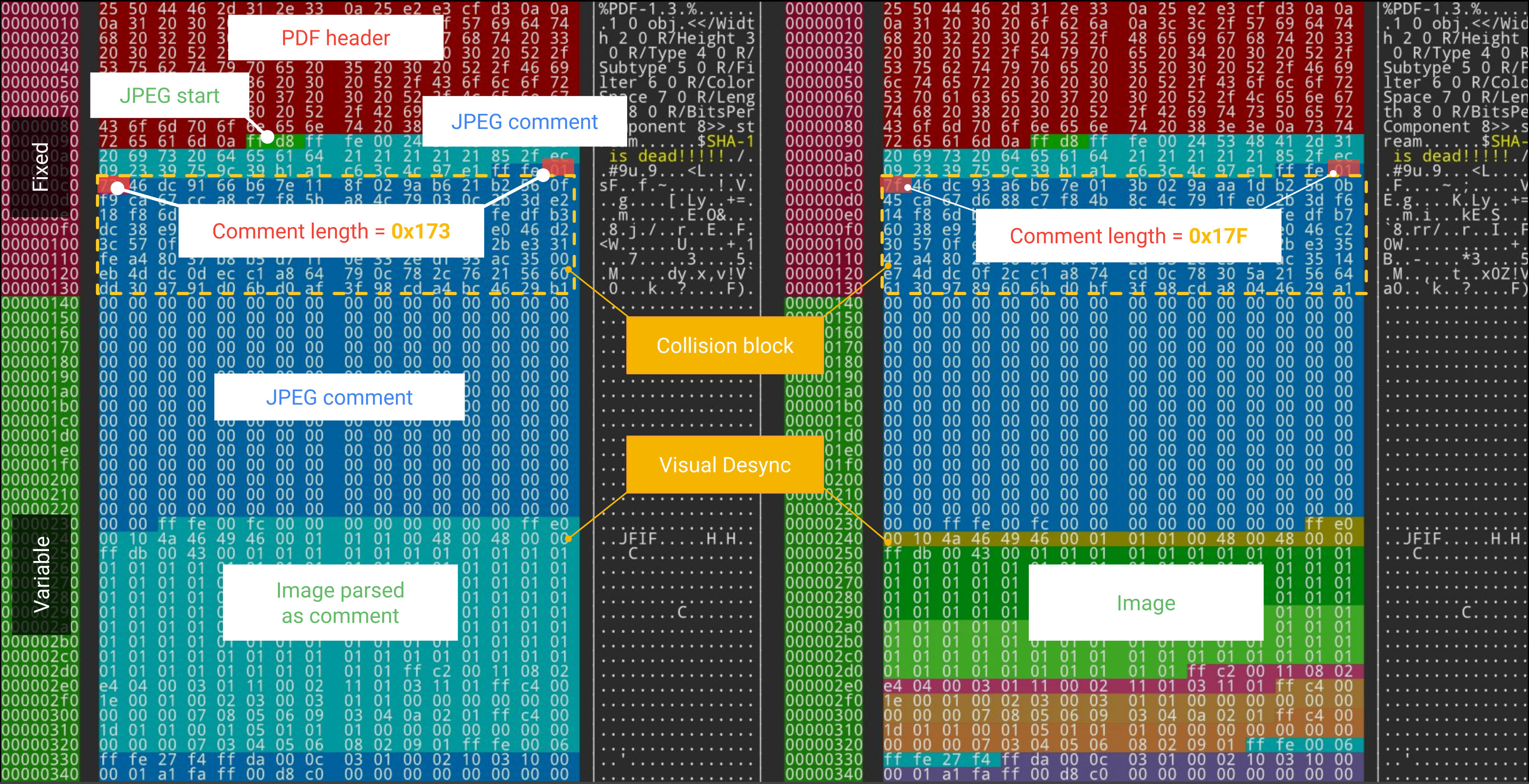
```
└─ sha1sum *.pdf
38762cf7f55934b34d179ae6a4c80cadccbb7f0a  1.pdf
38762cf7f55934b34d179ae6a4c80cadccbb7f0a  2.pdf
┌─ /tmp/sha1                                                  0.64G  8-11h
└─ sha256sum *.pdf
2bb787a73e37352f92383abe7e2902936d1059ad9f1ba6daaa9c1e58ee6970d0  1.pdf
d4488775d29bdef7993367d541064dbdda50d383f89f0aa13a6ff2e0894ba5ff  2.pdf
```

*using dedicated hardware would make the brute-force cheaper



```
Copyright (C) 2009-2015 D. R. Commander
Copyright (C) 2009-2011 Nokia Corporation and/or its subsidiary(-ies)
Copyright (C) 2013-2014 MIPS Technologies, Inc.
Copyright (C) 2013 Linaro Limited

Emulating The Independent JPEG Group's software, version 8d  15-Jan-2012

2550x3300 RGB Targa image
libjpeg-turbo version 1.4.2 (build 20160222)
Copyright (C) 1991-2012 Thomas G. Lane, Guido Vollbeding
Copyright (C) 1999-2006 MIYASAKA Masaru
Copyright (C) 2009 Pierre Ossman for Cendio AB
Copyright (C) 2009-2015 D. R. Commander
Copyright (C) 2009-2011 Nokia Corporation and/or its subsidiary(-ies)
Copyright (C) 2013-2014 MIPS Technologies, Inc.
Copyright (C) 2013 Linaro Limited

Emulating The Independent JPEG Group's software, version 8d  15-Jan-2012

2550x3300 RGB Targa image
[16:40:37] INFO: producing final PDFs
root@ubuntu:/sha1# sha1sum out-images/*
26f0e96406652c7d6bd3b11d2b59cc191e639481  out-images/cat.pdf
26f0e96406652c7d6bd3b11d2b59cc191e639481  out-images/tiger.pdf
root@ubuntu:/sha1# 
```

Same
SHA-1 values



```
00000000  25 50 44 46 2d 31 2e 33  0a 25 e2 e3 cf d3 0a 0a  %PDF-1.3.%......
00000010  0a 31 20 30 2                     f 57 69 64 74  .1 0 obj.<</Widt
00000020  68 20 32 20 3                     7 68 74 20 33  h 2 0 R/Height 3
00000030  20 30 20 52 2                     0 30 20 52 2f   0 R/Type 4 0 R/
00000040  53 75 62 74 79 70 65 20  35 20 30 20 52 2f 46 69  Subtype 5 0 R/Fi
00000050                 36 20 30  20 52 2f 43 6f 6c 6f 72  lter 6 0 R/Color
00000060                 20 37 20  30 20 52                 Space 7 0 R/Leng
00000070                 30 20 52  2f 42 69                 8 0 R/BitsPer
00000080  43 6f 6d 70 6f 6e 65 6e  74 20 38                 ponent 8>>.st
00000090  72 65 61 6d 0a ff d8 ff  fe 00 24                 m......$SHA-1
000000a0  20 69 73 20 64 65 61 64  21 21 21 21 21 85 2f ec   is dead!!!!!./.
000000b0  09 23 39 75 9c 39 b1 a1  c6 3c 4c 97 e1 ff fe 01  .#9u.9...<L.....
000000c0     46 dc 91 66 b6 7e 11  8f 02 9a b6 21 b2 56 0f  sF..f.~......!.V.
000000d0  f9 ca 67 cc a8 c7 f8 5b  a8 4c 79 03 0c 2b 3d e2  ..g....[.Ly..+=.
000000e0  18 f8 6d                             fe df b3  ..m......E.O&...
000000f0  dc 38 e9                             e0 46 d2  .8.j./..r..E..F.
00000100  3c 57 0f                             2b e3 31  <W.......U....+.1
00000110  fe a4 80                 0e 33 2e df 95 ac 35 00  ...7.....3....5.
00000120  eb 4d dc 0d ec c1 a8 64  79 0c 78 2c 76 21 56 60  .M.....dy.x,v!V`
00000130  dd 30 97 91 d0 6b d0 af  3f 98 cd a4 bc 46 29 b1  .0...k..?....F).
00000140  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000150  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000160  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000170  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000180  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000190  00 00 00 00                          00 00 00 00  ................
000001a0  00 00 00 00                          00 00 00 00  ................
000001b0  00 00 00 00                          00 00 00 00  ................
000001c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000001d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000001e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000200  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000210  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000220  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000230  00 00 ff fe 00 fc 00 00  00 00 00 00 00 00 ff e0  ................
00000240  00 10 4a 46 49 46 00 01  01 01 00 48 00 48 00 00  ..JFIF.....H.H..
00000250  ff db 00 43 00 01 01 01  01 01 01 01 01 01 01 01  ...C............
00000260  01 01 01 01                          01 01 01 01  ................
00000270  01 01 01 01                          01 01 01 01  ................
00000280  01 01 01 01                          01 01 01 01  ................
00000290  01 01 01 01                          01 01 01 01  ..........C.....
000002a0  01 01 01 01                          01 01 01 01  ................
000002b0  01 01 01 01                          01 01 01 01  ................
000002c0  01 01 01 01 01 01 01 01  01 01 01 01 01 01 01 01  ................
000002d0  01 01 01 01 01 01 01 01  01 01 ff c2 00 11 08 02  ................
000002e0  e4 04 00 03 01 11 00 02  11 01 03 11 01 ff c4 00  ................
000002f0  1e 00 01 00 02 03 00 03  01 01 00 00 00 00 00 00  ................
00000300  00 00 00 07 08 05 06 09  03 04 0a 02 01 ff c4 00  ................
00000310  1d 01 01 00 01 05 01 01  01 00 00 00 00 00 00 00  ................
00000320  00 00 00 07 03 04 05 06  08 02 09 01 ff fe 00 06  ................
00000330  ff fe 27 f4 ff da 00 0c  03 01 00 02 10 03 10 00  ..'.............
00000340  00 01 a1 fa ff 00 d8 c0  00 00 00 00 00 00 00 00  ................
```

```
00000000  25 50 44 46 2d 31 2e 33  0a 25 e2 e3 cf d3 0a 0a  %PDF-1.3.%.....
00000010  0a 31 20 30 20 6f 62 6a  0a 3c 3c 2f 57 69 64 74  .1 0 obj.<</Wid
00000020  68 20 32 20 30 20 52 2f  48 65 69 67 68 74 20 33  h 2 0 R/Height
00000030  20 30 20 52 2f 54 79 70  65 20 34 20 30 20 52 2f   0 R/Type 4 0 R
00000040  53 75 62 74 79 70 65 20  35 20 30 20 52 2f 46 69  Subtype 5 0 R/F
00000050  6c 74 65 72 20 36 20 30  20 52 2f 43 6f 6c 6f 72  lter 6 0 R/Colo
00000060  53 70 61 63 65 20 37 20  30 20 52 2f 4c 65 6e 67  Space 7 0 R/Len
00000070  74 68 20 38 20 30 20 52  2f 42 69 74 73 50 65 72  th 8 0 R/BitsPe
00000080  43 6f 6d 70 6f 6e 65 6e  74 20 38 3e 3e 0a 73 74  Component 8>>.s
00000090  72 65 61 6d 0a ff d8 ff  fe 00 24 53 48 41 2d 31  ream......$SHA-
000000a0  20 69 73 20 64 65 61 64  21 21 21 21 21 85 2f ec   is dead!!!!!./
000000b0  09 23 39 75 9c 39 b1 a1  c6 3c 4c 97 e1 ff fe 01  .#9u.9...<L....
000000c0  7f 46 dc 93 a6 b6 7e 01  3b 02 9a aa 1d b2 56 0b  .F....~.;.....V
000000d0  45 ca 67 d6 88 c7 f8 4b  8c 4c 79 1f e0 2b 3d f6  E.g....K.Ly..+=
000000e0  14 f8 6d                             e df b7  ..m.i...kE.S...
000000f0  60 38 e9                             0 46 c2  `8.rr/..r..I..F
00000100  30 57 0f                             b e3 35  0W...........+.
00000110  42 a4 80                             c 35 14  B..-.....*3....5
00000120  e7 4d dc 0f 2c c1 a8 74  cd 0c 78 30 5a 21 56 64  .M..,..t..x0Z!V
00000130  61 30 97 89 60 6b d0 bf  3f 98 cd a8 04 46 29 a1  a0..`k..?....F)
00000140  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ...............
00000150  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ...............
00000160  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ...............
00000170  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ...............
00000180  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ...............
00000190  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ...............
000001a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ...............
000001b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ...............
000001c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ...............
000001d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ...............
000001e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ...............
000001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ...............
00000200  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ...............
00000210  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ...............
00000220  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ...............
00000230  00 00 ff fe 00 fc 00 00  00 00 00 00 00 00 ff e0  ...............
00000240  00 10 4a 46 49 46 00 01  01 01 00 48 00 48 00 00  ..JFIF.....H.H
00000250  ff db 00 43 00 01 01 01  01 01 01 01 01 01 01 01  ...C...........
00000260  01 01 01 01 01                          01 01 01  ...............
00000270  01 01 01 01 01                          01 01 01  ...............
00000280  01 01 01 01 01                          01 01 01  ...............
00000290  01 01 01 01 01                          01 01 01  ..........C....
000002a0  01 01 01 01 01                          01 01 01  ...............
000002b0  01 01 01 01                             01 01 01  ...............
000002c0  01 01 01 01 01 01 01 01  01 01 01 01 01 01 01 01  ...............
000002d0  01 01 01 01 01 01 01 01  01 01 ff c2 00 11 08 02  ...............
000002e0  e4 04 00 03 01 11 00 02  11 01 03 11 01 ff c4 00  ...............
000002f0  1e 00 01 00 02 03 00 03  01 01 00 00 00 00 00 00  ...............
00000300  00 00 00 07 08 05 06 09  03 04 0a 02 01 ff c4 00  ...............
00000310  1d 01 01 00 01 05 01 01  01 00 00 00 00 00 00 00  ...............
00000320  00 00 00 07 03 04 05 06  08 02 09 01 ff fe 00 06  ...............
00000330  ff fe 27 f4 ff da 00 0c  03 01 00 02 10 03 10 00  ..'............
00000340  00 01 a1 fa ff 00 d8 c0  00 00 00 00 00 00 00 00  ...............
```

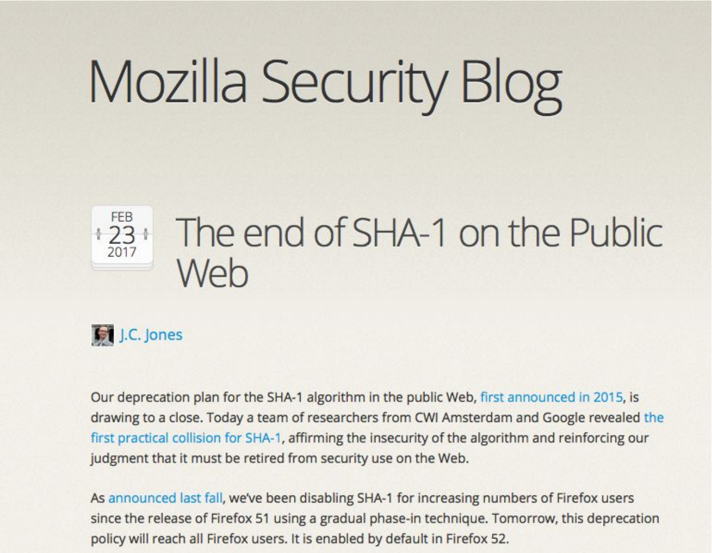
Research at Google

Original diagram by Hector Martin

https://shattered.io

Post-collision world

Firefox gave up on SHA-1 ahead of schedule



Mozilla Security Blog

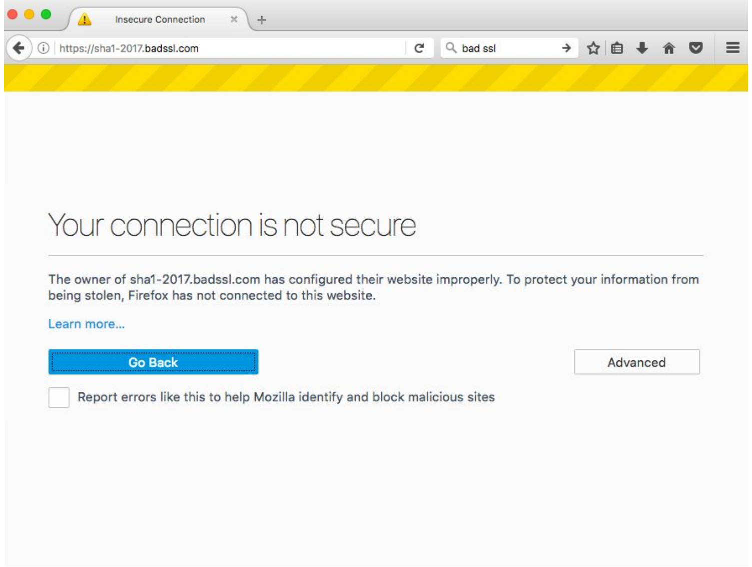
FEB 23 2017

The end of SHA-1 on the Public Web

J.C. Jones

Our deprecation plan for the SHA-1 algorithm in the public Web, first announced in 2015, is drawing to a close. Today a team of researchers from CWI Amsterdam and Google revealed the first practical collision for SHA-1, affirming the insecurity of the algorithm and reinforcing our judgment that it must be retired from security use on the Web.

As announced last fall, we've been disabling SHA-1 for increasing numbers of Firefox users since the release of Firefox 51 using a gradual phase-in technique. Tomorrow, this deprecation policy will reach all Firefox users. It is enabled by default in Firefox 52.



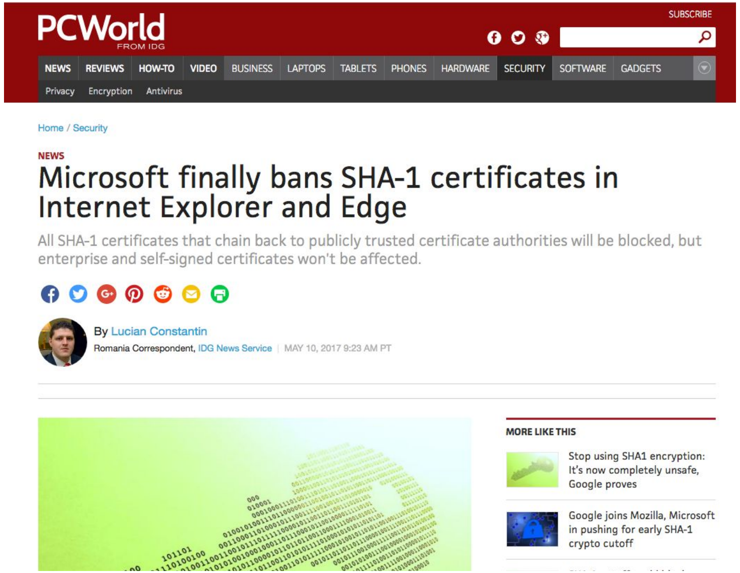
Your connection is not secure

The owner of sha1-2017.badssl.com has configured their website improperly. To protect your information from being stolen, Firefox has not connected to this website.

Learn more…

□ Report errors like this to help Mozilla identify and block malicious sites

Internet Explorer and Edge followed in May



NEWS

Microsoft finally bans SHA-1 certificates in Internet Explorer and Edge

All SHA-1 certificates that chain back to publicly trusted certificate authorities will be blocked, but enterprise and self-signed certificates won't be affected.

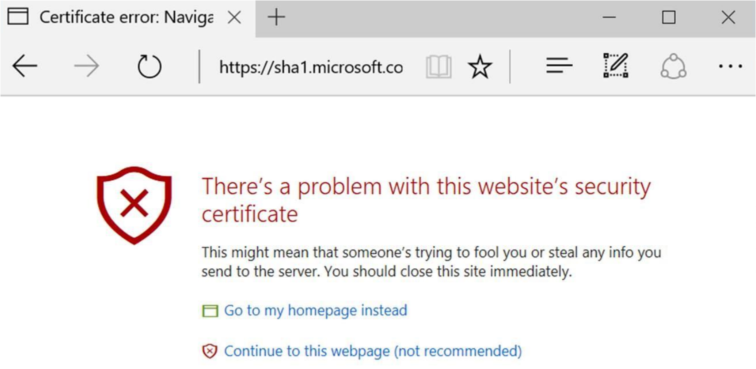
By Lucian Constantin

Romania Correspondent, IDG News Service | MAY 10, 2017 9:23 AM PT

MORE LIKE THIS

Stop using SHA1 encryption: It's now completely unsafe, Google proves

Google joins Mozilla, Microsoft in pushing for early SHA-1 crypto cutoff



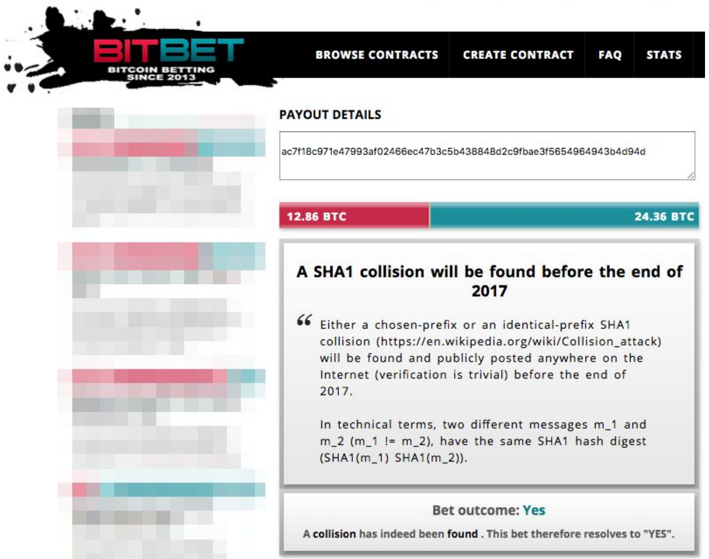
There's a problem with this website's security certificate

This might mean that someone's trying to fool you or steal any info you send to the server. You should close this site immediately.

Go to my homepage instead

Continue to this webpage (not recommended)

We got leaked! Largest bet 48h before release



[redacted]

PAYOUT DETAILS

ac7f18c971e47993af02466ec47b3c5b438848d2c9fbae3f5654964943b4d94d

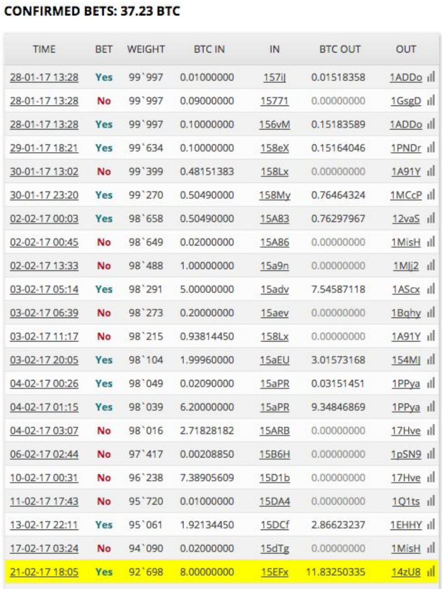
12.86 BTC | 24.36 BTC

A SHA1 collision will be found before the end of 2017

> Either a chosen-prefix or an identical-prefix SHA1 collision (https://en.wikipedia.org/wiki/Collision_attack) will be found and publicly posted anywhere on the Internet (verification is trivial) before the end of 2017.
>
> In technical terms, two different messages m_1 and m_2 (m_1 != m_2), have the same SHA1 hash digest (SHA1(m_1) SHA1(m_2)).

Bet outcome: Yes

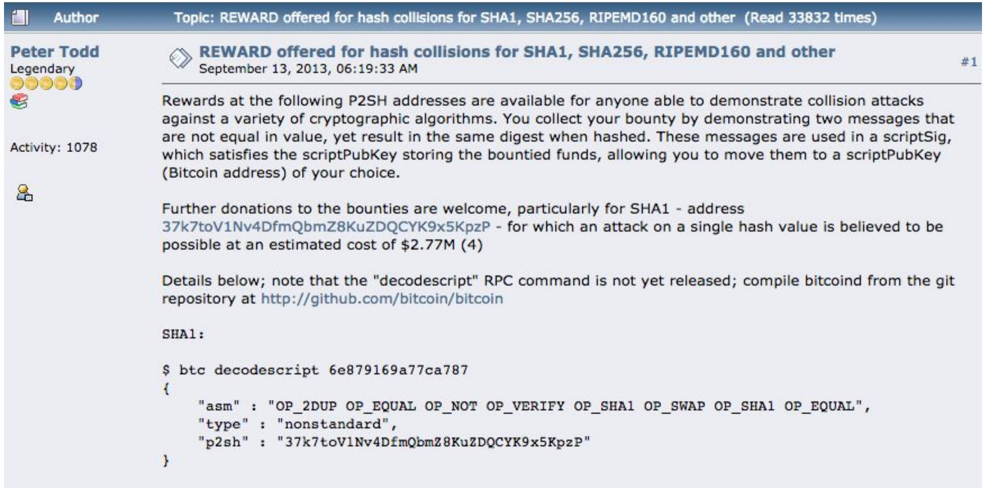
A **collision** has indeed been **found** . This bet therefore resolves to "YES".



CONFIRMED BETS: 37.23 BTC

TIME	BET	WEIGHT	BTC IN	IN	BTC OUT	OUT
28-01-17 13:28	Yes	99`997	0.01000000	157iJ	0.01518358	1ADDo
28-01-17 13:28	No	99`997	0.09000000	15771	0.00000000	1GsgD
28-01-17 13:28	Yes	99`997	0.10000000	156vM	0.15183589	1ADDo
29-01-17 18:21	Yes	99`634	0.10000000	158eX	0.15164046	1PNDr
30-01-17 13:02	No	99`399	0.48151383	158Lx	0.00000000	1A91Y
30-01-17 23:20	Yes	99`270	0.50490000	158My	0.76464324	1MCcP
02-02-17 00:03	Yes	98`658	0.50490000	15A83	0.76297967	12vaS
02-02-17 00:45	No	98`649	0.02000000	15A86	0.00000000	1MisH
02-02-17 13:33	No	98`488	1.00000000	15a9n	0.00000000	1MJj2
03-02-17 05:14	Yes	98`291	5.00000000	15adv	7.54587118	1AScx
03-02-17 06:39	No	98`273	0.20000000	15aev	0.00000000	1Bqhy
03-02-17 11:17	No	98`215	0.93814450	158Lx	0.00000000	1A91Y
03-02-17 20:05	Yes	98`104	1.99960000	15aEU	3.01573168	154MJ
04-02-17 00:26	Yes	98`049	0.02090000	15aPR	0.03151451	1PPya
04-02-17 01:15	Yes	98`039	6.20000000	15aPR	9.34846869	1PPya
04-02-17 03:07	No	98`016	2.71828182	15ARB	0.00000000	17Hve
06-02-17 02:44	No	97`417	0.00208850	15B6H	0.00000000	1pSN9
10-02-17 00:31	No	96`238	7.38905609	15D1b	0.00000000	17Hve
11-02-17 17:43	No	95`720	0.01000000	15DA4	0.00000000	1Q1ts
13-02-17 22:11	Yes	95`061	1.92134450	15DCf	2.86623237	1EHHY
17-02-17 03:24	No	94`090	0.02000000	15dTg	0.00000000	1MisH
21-02-17 18:05	Yes	92`698	8.00000000	15EFx	11.83250335	14zU8

Marc claimed bitcoin bounty just in time



Author | Topic: REWARD offered for hash collisions for SHA1, SHA256, RIPEMD160 and other (Read 33832 times)

Peter Todd
Legendary

Activity: 1078

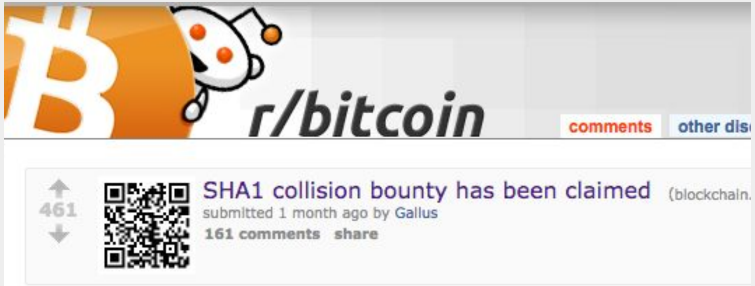
REWARD offered for hash collisions for SHA1, SHA256, RIPEMD160 and other
September 13, 2013, 06:19:33 AM #1

Rewards at the following P2SH addresses are available for anyone able to demonstrate collision attacks against a variety of cryptographic algorithms. You collect your bounty by demonstrating two messages that are not equal in value, yet result in the same digest when hashed. These messages are used in a scriptSig, which satisfies the scriptPubKey storing the bountied funds, allowing you to move them to a scriptPubKey (Bitcoin address) of your choice.

Further donations to the bounties are welcome, particularly for SHA1 - address 37k7toV1Nv4DfmQbmZ8KuZDQCYK9x5KpzP - for which an attack on a single hash value is believed to be possible at an estimated cost of $2.77M (4)

Details below; note that the "decodescript" RPC command is not yet released; compile bitcoind from the git repository at http://github.com/bitcoin/bitcoin

```
SHA1:

$ btc decodescript 6e879169a77ca787
{
    "asm" : "OP_2DUP OP_EQUAL OP_NOT OP_VERIFY OP_SHA1 OP_SWAP OP_SHA1 OP_EQUAL",
    "type" : "nonstandard",
    "p2sh" : "37k7toV1Nv4DfmQbmZ8KuZDQCYK9x5KpzP"
}
```

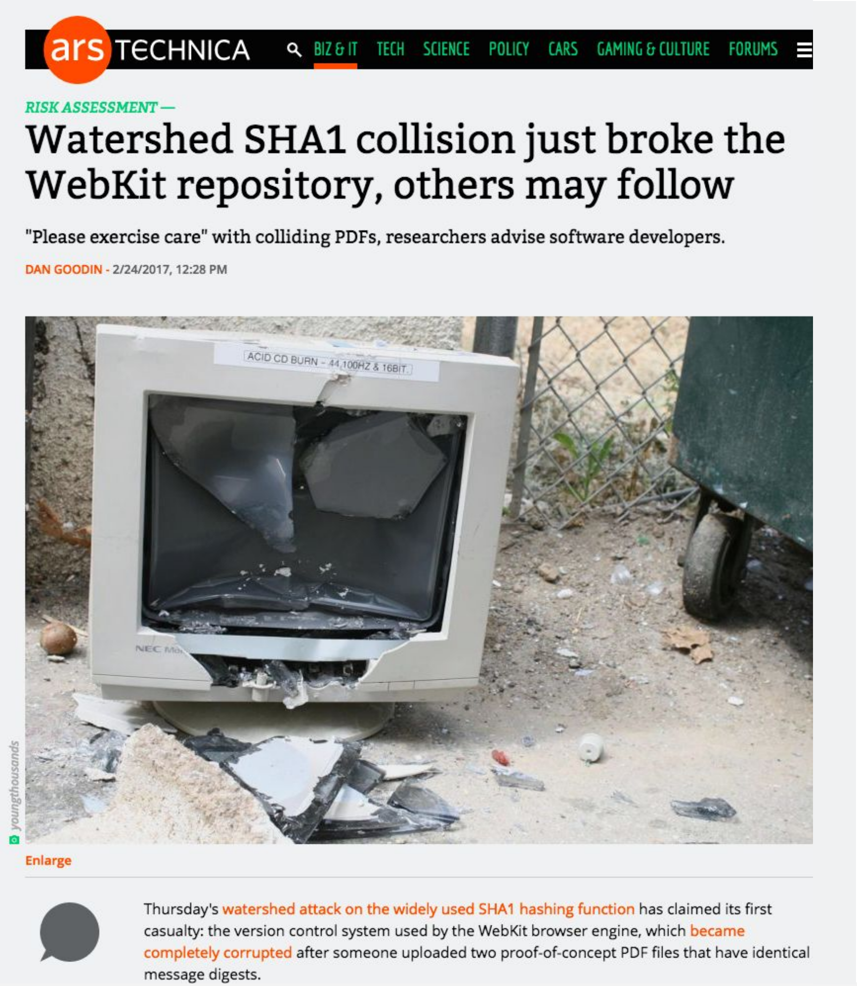


461

SHA1 collision bounty has been claimed (blockchain.

submitted 1 month ago by Gallus

161 comments share



RISK ASSESSMENT —

Watershed SHA1 collision just broke the WebKit repository, others may follow

"Please exercise care" with colliding PDFs, researchers advise software developers.

DAN GOODIN - 2/24/2017, 12:28 PM

Enlarge

Thursday's watershed attack on the widely used SHA1 hashing function has claimed its first casualty: the version control system used by the WebKit browser engine, which became completely corrupted after someone uploaded two proof-of-concept PDF files that have identical message digests.

Scaling computation

WebKit developer submitted a test to prove WebKit is resistant to SHA-1 collision

Due to an unforeseen bug in SVN Webkit SVN is offline for a few hours

SVN issue emergency patch

Legacy software

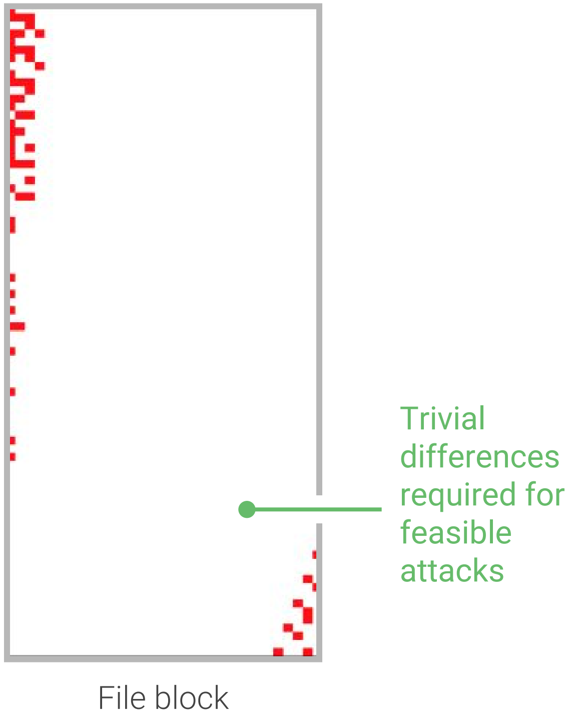
Counter-cryptanalysis to the rescue!

SHA1 deeply integrated into GIT despite early warning made it hard to fix

Transition plan slowly in the making

https://goo.gl/QgqRHp & https://goo.gl/cdvFpP

GIT is using SHA-1 for foreseeable future



Leverage how collisions are created

Works against “unknown” collisions

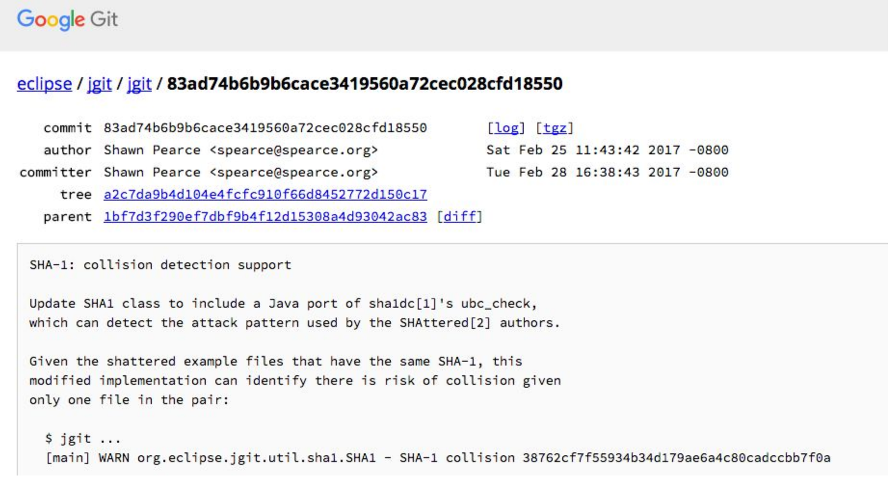
Only requires one file to detect collision

Allows collision forensic (e.g Flame)

Negligible false positives

Provide reliable detection system

Mitigating GIT issues with counter-cryptanalysis

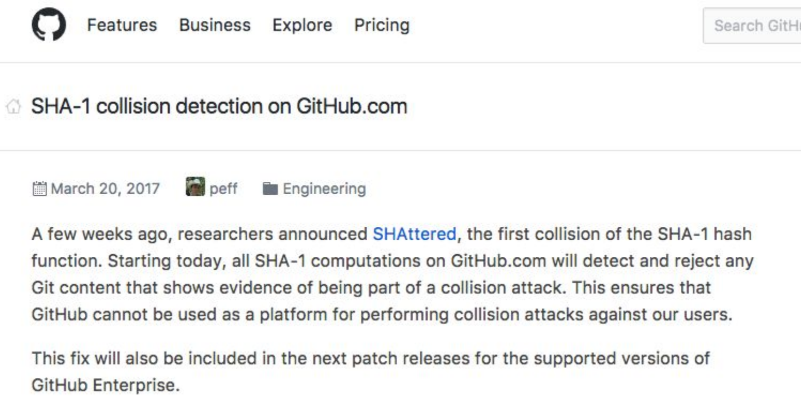


eclipse / jgit / jgit / **83ad74b6b9b6cace3419560a72cec028cfd18550**

```
   commit  83ad74b6b9b6cace3419560a72cec028cfd18550      [log] [tgz]
   author  Shawn Pearce <spearce@spearce.org>            Sat Feb 25 11:43:42 2017 -0800
committer  Shawn Pearce <spearce@spearce.org>            Tue Feb 28 16:38:43 2017 -0800
     tree  a2c7da9b4d104e4fcfc910f66d8452772d150c17
   parent  1bf7d3f290ef7dbf9b4f12d15308a4d93042ac83 [diff]
```

```
SHA-1: collision detection support

Update SHA1 class to include a Java port of sha1dc[1]'s ubc_check,
which can detect the attack pattern used by the SHAttered[2] authors.

Given the shattered example files that have the same SHA-1, this
modified implementation can identify there is risk of collision given
only one file in the pair:

  $ jgit ...
  [main] WARN org.eclipse.jgit.util.sha1.SHA1 - SHA-1 collision 38762cf7f55934b34d179ae6a4c80cadccbb7f0a
```

JGit (Feb 2017)



SHA-1 collision detection on GitHub.com

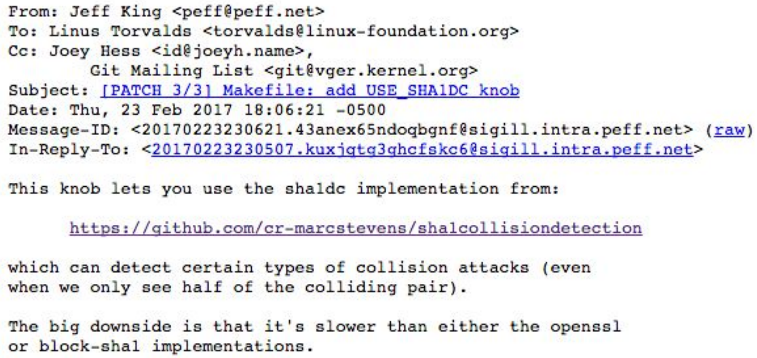
March 20, 2017 peff Engineering

A few weeks ago, researchers announced SHAttered, the first collision of the SHA-1 hash function. Starting today, all SHA-1 computations on GitHub.com will detect and reject any Git content that shows evidence of being part of a collision attack. This ensures that GitHub cannot be used as a platform for performing collision attacks against our users.

This fix will also be included in the next patch releases for the supported versions of GitHub Enterprise.

Github.com (Mar 2017)

GIT got counter-cryptanalysis late march

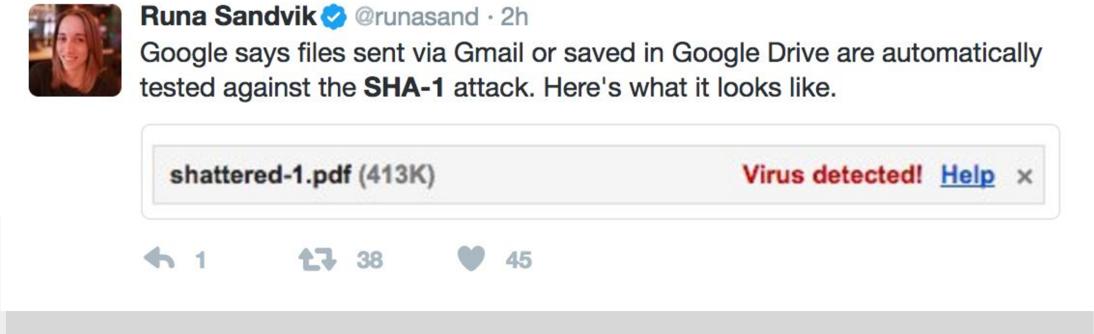


```
From: Jeff King <peff@peff.net>
To: Linus Torvalds <torvalds@linux-foundation.org>
Cc: Joey Hess <id@joeyh.name>,
        Git Mailing List <git@vger.kernel.org>
Subject: [PATCH 3/3] Makefile: add USE_SHA1DC knob
Date: Thu, 23 Feb 2017 18:06:21 -0500
Message-ID: <20170223230621.43anex65ndoqbgnf@sigill.intra.peff.net> (raw)
In-Reply-To: <20170223230507.kuxjqtq3qhcfskc6@sigill.intra.peff.net>

This knob lets you use the sha1dc implementation from:

      https://github.com/cr-marcstevens/sha1collisiondetection

which can detect certain types of collision attacks (even
when we only see half of the colliding pair).

The big downside is that it's slower than either the openssl
or block-sha1 implementations.
```

Git 2.12.2 (Mar 2017)

https://github.com/git/git/tree/master/sha1dc

Google scans incoming documents



Runa Sandvik @runasand · 2h

Google says files sent via Gmail or saved in Google Drive are automatically tested against the **SHA-1** attack. Here's what it looks like.

shattered-1.pdf (413K) **Virus detected!** Help ×

Using Counter-cryptanalysis to prevents old client files reader from being abused

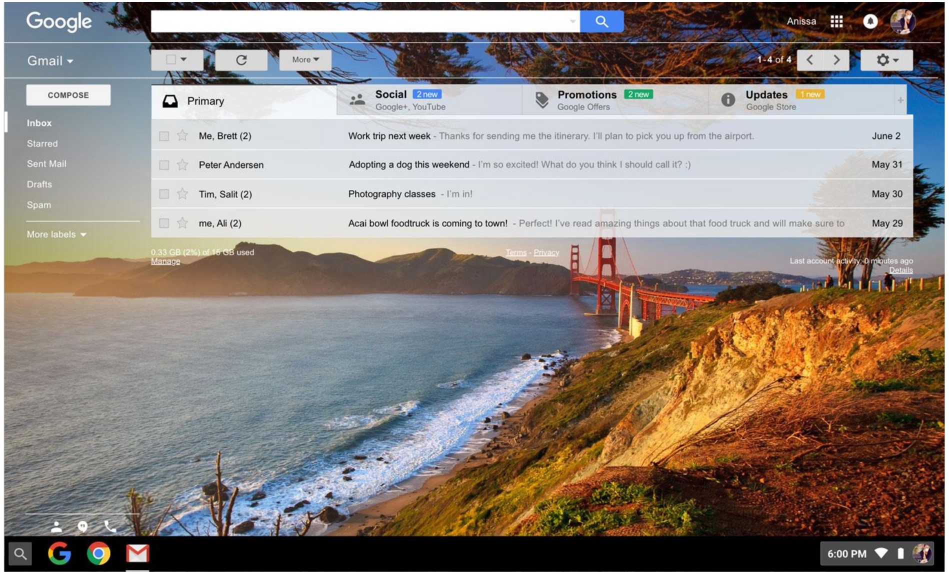
Why scan files for collision?

Crash legacy client software

Colliding document with differents terms

Blackswan

Gmail counter-cryptanalysis cost



Me, Brett (2)	Work trip next week - Thanks for sending me the itinerary. I'll plan to pick you up from the airport.	June 2
Peter Andersen	Adopting a dog this weekend - I'm so excited! What do you think I should call it? :)	May 31
Tim, Salit (2)	Photography classes - I'm in!	May 30
me, Ali (2)	Acai bowl foodtruck is coming to town! - Perfect! I've read amazing things about that food truck and will make sure to	May 29

~4.45%

overhead

Overhead computed on a sample set of 1B PDFs documents scanned in April 2017

The future of hash security is diversity

	Security Claim	Fixed prefix	Chosen attack
~~SHA-1~~	MD		
SHA-256	MD	2^{128}	
SHA-3	Sponge	2^{128}	2^{128}
BLAKE	HAIFA	2^{128}	2^{256}

Takeaways

SHA-1 is dead
long live to
SHA-256 & SHA-3

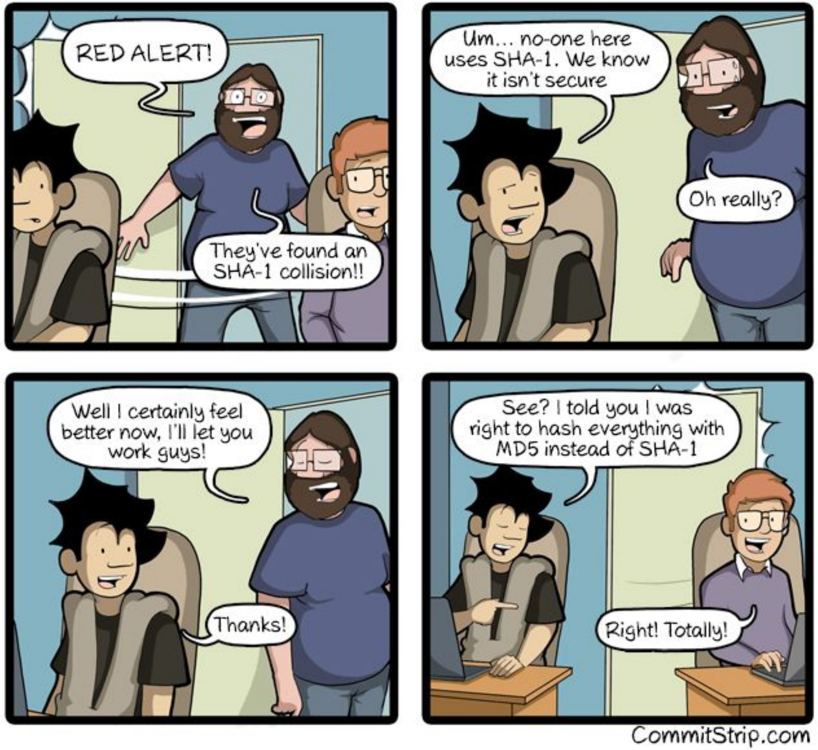
End of an era

Counter-cryptanalysis
as a means of
detection

Hash cryptanalysis as a mean to detect unknown collisions

Hash diversity
as a safeguard for
the years to come

We now have a very diverse set of hash function constructions


RED ALERT!
They've found an SHA-1 collision!!
Um... no-one here uses SHA-1. We know it isn't secure
Oh really?
Well I certainly feel better now, I'll let you work guys!
Thanks!
See? I told you I was right to hash everything with MD5 instead of SHA-1
Right! Totally!
CommitStrip.com

Questions?

Come see our team's other talks

Tracking ransomware end-to end

Today | 5:05pm-5:30pm | Mandalay Bay EF

Attacking encrypted USB keys the hard(ware) way

Tomorrow | 12:10pm-1:00pm | South Seas CD

Research at Google

Thank you

shattered.io