Google Security Blog

The latest news and insights from Google on security and safety on the Internet

Announcing the first SHA1 collision

February 23, 2017

Posted by Marc Stevens (CWI Amsterdam), Elie Bursztein (Google), Pierre Karpman (CWI Amsterdam), Ange Albertini (Google), Yarik Markov (Google), Alex Petit Bianco (Google), Clement Baisse (Google)

Cryptographic hash functions like SHA-1 are a cryptographer's swiss army knife. You'll find that hashes play a role in browser security, managing code repositories, or even just detecting duplicate files in storage. Hash functions compress large amounts of data into a small message digest. As a cryptographic requirement for wide-spread use, finding two messages that lead to the same digest should be computationally infeasible. Over time however, this requirement can fail due to attacks on the mathematical underpinnings of hash functions or to increases in computational power.

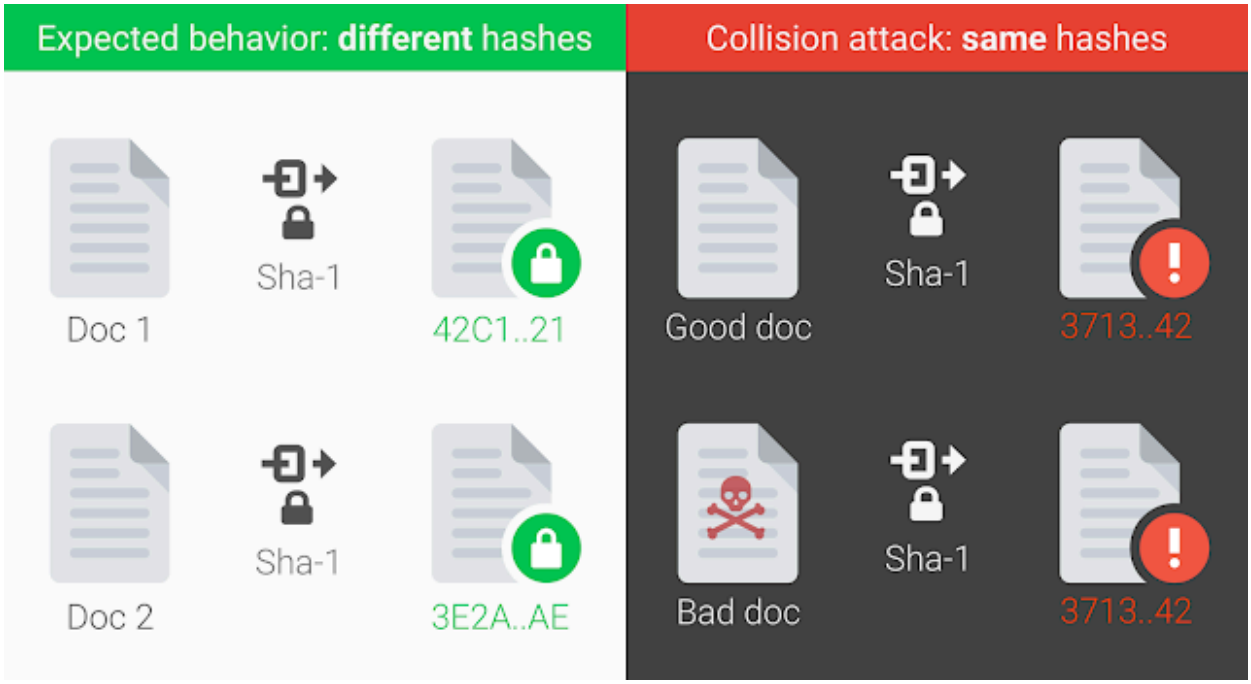
Today, more than 20 years after of SHA-1 was first introduced, we are announcing the first practical technique for generating a collision. This represents the culmination of two years of research that sprung from a collaboration between the CWI Institute in Amsterdam and Google. We've summarized how we went about generating a collision below. As a proof of the attack, we are releasing two PDFs that have identical SHA-1 hashes but different content.

For the tech community, our findings emphasize the necessity of sunsetting SHA-1 usage. Google has advocated the deprecation of SHA-1 for many years, particularly when it comes to signing TLS certificates. As early as 2014, the Chrome team announced that they would gradually phase out using SHA-1. We hope our practical attack on SHA-1 will cement that the protocol should no longer be considered secure.

We hope that our practical attack against SHA-1 will finally convince the industry

that it is urgent to move to safer alternatives such as SHA-256.

What is a cryptographic hash collision?



A collision occurs when two distinct pieces of data—a document, a binary, or a website's certificate—hash to the same digest as shown above. In practice, collisions should never occur for secure hash functions. However if the hash algorithm has some flaws, as SHA-1 does, a well-funded attacker can craft a collision. The attacker could then use this collision to deceive systems that rely on hashes into accepting a malicious file in place of its benign counterpart. For example, two insurance contracts with drastically different terms.

Finding the SHA-1 collision

In 2013, Marc Stevens published a paper that outlined a theoretical approach to create a SHA-1 collision. We started by creating a PDF prefix specifically crafted to allow us to generate two documents with arbitrary distinct visual contents, but that

would hash to the same SHA-1 digest. In building this theoretical attack in practice we had to overcome some new challenges. We then leveraged Google's technical expertise and cloud infrastructure to compute the collision which is one of the largest computations ever completed.

Here are some numbers that give a sense of how large scale this computation was:

- Nine quintillion (9,223,372,036,854,775,808) SHA1 computations in total
- 6,500 years of CPU computation to complete the attack first phase
- 110 years of GPU computation to complete the second phase

While those numbers seem very large, the SHA-1 shattered attack is still more than 100,000 times faster than a brute force attack which remains impractical.

Mitigating the risk of SHA-1 collision attacks

Moving forward, it's more urgent than ever for security practitioners to migrate to safer cryptographic hashes such as SHA-256 and SHA-3. Following Google's vulnerability disclosure policy, we will wait 90 days before releasing code that allows anyone to create a pair of PDFs that hash to the same SHA-1 sum given two distinct images with some pre-conditions. In order to prevent this attack from active use, we've added protections for Gmail and GSuite users that detects our PDF collision technique. Furthermore, we are providing a free detection system to the public.

You can find more details about the SHA-1 attack and detailed research outlining our techniques here.

About the team

This result is the product of a long-term collaboration between the CWI institute and Google's Research security, privacy and anti-abuse group.

Marc Stevens and Elie Bursztein started collaborating on making Marc's cryptanalytic attacks against SHA-1 practical using Google infrastructure. Ange Albertini developed the PDF attack, Pierre Karpman worked on the cryptanalysis and the GPU implementation, Yarik Markov took care of the distributed GPU code, Alex Petit Bianco implemented the collision detector to protect Google users and Clement Baisse oversaw the reliability of the computations.


SHAttered
The first concrete collision attack against SHA-1
https://shattered.io
A collision is when two different documents have the same hash fingerprint
Doc 1
SHA-1
42C1..21
bad doc 1
SHA-1
3713..42
Doc 2
SHA-1
3E2A..AE
bad doc 2
SHA-1
3713..42
Normal behavior - different hashes
Collision - same hashes
Potentially Impacted Systems
https
Document signature
HTTPS certificate
Version control (git)
Backup System
Attack complexity
9,223,372,036,854,775,808
SHA-1 compressions performed
Shattered compared to other collision attacks


MD5
1 smartphone
30 sec
SHA-1 Shattered
110 GPU
1 year
SHA-1 Bruteforce
12,000,000 GPU
1 year
Defense
Use SHA-256 or SHA-3 as replacement
Use shattered.io to test your PDF
Google products are already protected
Use collision detection code
Team
CWI
Marc Stevens
Pierre Karpman
Google
Elie Bursztein
Ange Albertini
Yarik Markov
learn more at https://shattered.io

No comments :

Post a Comment