SHA-1 is a Shambles

We have computed the very first **chosen-prefix collision for SHA-1**. In a nutshell, this means a complete and practical break of the SHA-1 hash function, with dangerous practical implications if you are still using this hash function. To put it in another way: all attacks that are practical on MD5 are now also practical on SHA-1.

Check our paper **here** for more details. Slides from RWC are also available.

Our Contributions

Complexity Improvements

We have significantly improved the complexity of SHA-1 attacks, with a speedup factor around 10. More precisely, we have reduced the cost of a collision attack from $2^{64.7}$ to $2^{61.2}$, and the cost of a chosen-prefix collision attack from $2^{67.1}$ to $2^{63.4}$ (on a GTX 970 GPU).

Record Computation

We implemented the entire chosen-prefix collision attack with those improvements. This attack is extremely technical, contains many details, various steps, and requires a lot of engineering work. In order to perform this computation with a small academic budget, we rented cheap gaming or mining GPUs from GPUserversrental, rather that the datacenter-grade hardware used by big cloud providers. We have successfully run the computation during two months last summer, using 900 GPUs (Nvidia GTX 1060).

As a side result, this shows that it now costs less than 100k USD to break cryptography with a security level of 64 bits (i.e. to compute 2^{64} operations of symmetric cryptography).

PGP/GnuPG Impersonation

We chose the PGP/GnuPG Web of Trust as demonstration of our chosen-prefix collision attack against SHA-1. The Web of Trust is a trust model used for PGP that relies on users signing each other's identity certificate, instead of using a central PKI. For compatibility reasons the legacy branch of GnuPG (version 1.4) still uses SHA-1 by default for identity certification.

Using our SHA-1 chosen-prefix collision, we have created two PGP keys with different UserIDs and colliding certificates: key B is a legitimate key for Bob (to be signed by the Web of Trust), but the signature can be transferred to key A which is a forged key with Alice's ID. The signature will still be valid because of the collision, but Bob controls key A with the name of Alice, and signed by a third party. Therefore, he can impersonate Alice and sign any document in her name.

Our Chosen-Prefix Collision Example

We have create a chosen-prefix collision with prefixes `99040d047fe81780012000` and `99030d047fe81780011800` (in hexadecimal notation). You can download the two messages below, and verify their hash with the `sha1sum` tool:

- messageA
- messageB

The prefixes have been chosen to build two PGP public keys with colliding SHA-1 certification signatures. You can download two example keys below, with different user names, and examine them with `pgpdump -i` to see that the SHA-1 signatures issued by `0xAFBB1FED6951A956` are the same:

- alice.asc
- bob.asc

In order to avoid malicious usage, the keys have a creation date far in the future; if you want to analyse them with pgp, you can use options `--ignore-time-conflict --ignore-valid-from` (more generally you can prefix arbitrary commands with `faketime @2145920400`).

Responsible Disclosure and Impact

We have tried to contact the authors of affected software before announcing this attack, but due to limited resources, we could not notify everyone.

GnuPG

We have first discussed this attack with the GnuPG developers the 9th of May 2019 and eventually informed them of the newly found chosen-prefix collision the 1st of October 2019. The issue is tracked with CVE number CVE-2019-14855. A countermeasure has been implemented in commit edc36f5, included in GnuPG version 2.2.18 (released on the 25th of November 2019): SHA-1-based identity signatures created after 2019-01-19 are now considered invalid.

CAcert

CAcert is one of them main CAs for PGP keys. We noticed that there is a large number of keys with recent SHA-1 signatures from CAcert on public keyservers. This seems to indicate that they still use SHA-1 to sign user keys. We have first contacted them by email on December 14th, and got an answer on January 6th acknowledging this issue. They are planning a switch to a secure hash function for key certification.

OpenSSL

We have first contacted the OpenSSL developers on December 14th. The next version of OpenSSL will no longer allow X.509 certificates signed using SHA-1 at security level 1 and above (commit 68436f0). Since security level 1 is the default configuration for TLS/SSL, this will prevent SHA-1 usage for certificates. Debian Linux had previously set the default configuration to security level 2 (defined as 112-bit security) in the latest release (Debian Buster); this already prevents dangerous usage of SHA-1.

OpenSSH

The latest version of OpenSSH (8.2) includes a “future deprecation notice” explaining that SHA-1 signatures will be disabled in the near-future.

DNSSEC

After publication of our results, it was pointed that SHA-1 remained used in DNSSEC, with 18% of TLDs using SHA-1 signatures. It is advised that anyone who is using a SHA-1 DNSKEY algorithm (algorithm numbers 7 or less) should upgrade. See related page from Tony Finch or the IETF related discussion.

Q&A

What is a chosen-prefix collision?

A classical collision (or identical-prefix collision) for a hash function H is simply two messages M and M’ that lead to the same hash output: H(M) = H(M’). Even though this security notion is fundamental in cryptography, exploiting a classical collision for attacks in practice is difficult.

A chosen-prefix collision is a more constrained (and much more difficult to obtain) type of collision, where two message prefixes P and P’ are first given as challenge to the adversary, and his goal is then to compute two messages M and M’ such that H(P || M) = H(P’ || M’), where || denotes concatenation.

With such an ability, the attacker can obtain a collision even though prefixes can be chosen arbitrarily (and thus potentially contain some meaningful information). This is particularly impactful when the hash function is used in a digital signature scheme, one of the most common usage of a hash function.

Is SHA-1 really still used?

SHA-1 usage has significantly decreased in the last years; in particular web browsers now reject certificates signed with SHA-1. However, SHA-1 signatures are still supported in a large number of applications. SHA-1 is the default hash function used for certifying PGP keys in the legacy branch of GnuPG, and those signatures were accepted by the modern branch of GnuPG before we reported our results. Many non-web TLS clients also accept SHA-1 certificates, and SHA-1 is still allowed for in-protocol signatures in TLS and SSH. Even if actual usage is low (in the order of 1%), the fact that SHA-1 is allowed threatens the security because a meet-in-the-middle attacker will downgrade the connection to SHA-1. SHA-1 is also the foundation of the GIT versioning system. There are probably a lot of less known or proprietary protocols that still use SHA-1, but this is more difficult to evaluate.

What is affected?

Any usage where collision resistance is expected from SHA-1 is of course at high risk. We identified a few settings that are directly affected by chosen-prefix collisions:

- PGP keys can be forged if third parties generate SHA-1 key certifications
- X.509 certificates could be broken if some Certificate Authorities issue SHA-1 certificates with predictable serial numbers

We note that classical collisions and chosen-prefix collisions do not threaten all usages of SHA-1. In particular, HMAC-SHA-1 seems relatively safe, and preimage resistance (aka ability to invert the hash function) of SHA-1 remains unbroken as of today. Yet, as cryptographers we recommend to deprecate SHA-1 everywhere, even when there is no direct evidence that this weaknesses can be exploited.

What should I do?

Remove any use of SHA-1 in your product as soon as possible and use instead SHA-256 or SHA-3.

SHA-1 has been broken for 15 years, so there is no good reason to use this hash function in modern security software. Attacks only get better over time, and the goal of the cryptanalysis effort is to warn users so that they can deprecate algorithms before the attacks get practical. We actually expect our attack to cost just a couple thousand USD in a few years.

How much does the attack cost?

By renting a GPU cluster online, the entire chosen-prefix collision attack on SHA-1 costed us about 75k USD. However, at the time of computation, our implementation was not optimal and we lost some time (because research). Besides, computation prices went further down since then, so we estimate that our attack costs today about 45k USD. As computation costs continue to decrease rapidly, we evaluate that it should cost less than 10k USD to generate a chosen-prefix collision attack on SHA-1 by 2025.

As a side note, a classical collision for SHA-1 now costs just about 11k USD.

Wasn’t there already a collision attack against SHA-1?

A classical collision has been computed for SHA-1 in late 2017, as you can see here. However, this is very different from a chosen-prefix collision, where any prefix pair can be challenged for the collision, which leads to a much more serious impact in practice.

Wasn’t there already a chosen-prefix collision attack against

SHA-1?

Last year, we announced a new chosen-prefix collision attack, as you can see here (some test code is also available) and this work was published at the Eurocrypt 2019 conference. Here, we further improved these results up to a point where the attack becomes doable for a reasonable amount of money, and we wrote an actual implementation of the attack to compute the chosen-prefix collision against SHA-1.

Can I try it out for myself?

Since our attack on SHA-1 has pratical implications, in order to make sure proper countermeasures have been pushed we will wait for some time before releasing source code that allows to generate SHA-1 chosen-prefix collisions.

Press Coverage

- Ars Technica
- The Register
- Le Monde (in French)
- Tom’s Hardware

Contact

If you have any questions, feel free to contact us:

Gaëtan Leurent: gaetan.leurent@inria.fr
Thomas Peyrin: thomas.peyrin@ntu.edu.sg