Volume 1, Issue 2 | 23 October, 2018

Three Things you should know about: Secure Hash Algorithms

Secure Hash Algorithms (SHA) are cryptographic hash functions first developed by the NSA. A hash function is a mathematical algorithm that can take any amount of data and map it onto a chunk of data of a fixed size. A cryptographic hash function is a hash function designed for cryptography.

SHA functions are deterministic: Applying the hash algorithm to a message (a chunk of data) will always produce the same value. SHA is used to authenticate data by computing the hash value of the data and comparing it to the expected hash value—if the hash values don't match, the data has been modified or tampered with.

To be effective for cryptography, **hash functions must be collision resistant:** A collision occurs when two messages produce the same hash value. Remember, a hash takes any size message and converts it to a fixed-size value, so it's theoretically possible for two messages to produce the same hash value (think monkeys and typewriters). However, it is not easy. SHA-1 was first published in 1995; in 2017 Google engineers published proof of the first collision, publishing two dissimilar PDF files that produced the same SHA-1 hash value. In order to force the collision, they ran 2^{63} (nine quintillion) SHA-1 computations, which took 6,500 years of CPU time and 110 years of GPU time—take our word for it, this is A LOT of computational power. The SHA-1 algorithm has been considered non-secure since 2005 and has now largely been replaced by the next-generation algorithms, SHA-2 and SHA-3.

SHA algorithms are preimage resistant: The other criteria for a cryptographic hash function is that it must be preimage resistant. A preimage attack is the ability to identify what message produced a hash value—in other words, undoing the hash. If it took nine quintillion computations to find two messages that produced the same hash, imagine how much it would take to undo a particular hash. The best available algorithms are theoretically estimated to require 2^{188} computations–that's a quintillion times more complex than the collision attack algorithm.

So that gives us our bonus fourth thing about SHA: they are **not vulnerable to quantum computing.** While quantum computing (think billion-fold processing power) does pose threats to certain types of cryptography (such as RSA, a cryptokey system we'll address in the next issue), SHA attacks require so much computing power that even quantum computing makes attacks practically impossible.