Hash algorithms

Hash algorithms helps people to transfer files from one computer to another and makes sure that while it was traveling it was not altered or damaged on the way. So the sender and the receiver can know that the file was not altered by a hacker during the transfer.

Hash algorithm take a big files and encrypts it in to a smaller string of characters, but if only one simbol is changed in the original file, then it would produce a completely different hash code. These types of algorithms have to have 3 main features:

1.Be fast to encrypt a message, few seconds at most. But being too quick is considered that algorithm is too simple.

2.If at least one bit of the original message is change, the algorithm has to be completely different.

3.It can not produce hash collision. When two different messages generate the same hash code.

As everything in technology there were and is a lot of different hash algorithms. And as technology is getting faster and better security algorithms have to be at least two steps a head.

One of the agencies that are looking after internet security is National Institute of Standards and Technology (NIST). This institute basically is helping to standartize different types of technology so that it would be easier to combine them to achieve the next milestone in technology.

And in 2006 NIST announced a competition for creating the next hash function standart - SHA-3. They saw a rising need for a better hash algorithm because new cases of breaking MD5 and SHA-0 have proven to be true (most of MD5 hash codes can now be solved just by entering it to google), and successful attacks were carried out on SHA-1. Luckily at this time SHA-2 was an international standard, but it was build using the same engine as SHA-1. It was only a matter of time till SHA-2 becomes unsafe.

And finally on August 5, 2015 NIST announced that Keccak algorithm designed by Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche has become a hashing standard - SHA-3.

Keccak uses an innovative "sponge engine" to hash the message text. This algorithm has 2 phases. First one to absorb data and second to squeeze it out. First phase is a lot like SHA-1 and SHA-2 Merkle-Damgard engine. With the biggest difference in the state size of 1600 bits. It also handles inputs differently. Instead of compression functions with 2 inputs (state and message) SHA-3 uses exclusive 'OR' gate to send the massage block to the state and then apply a block permutation to the state. Second state works opposite of first phase. Pieces of blocks are squeezed out by the state with block permutation being applied in between.

Keccak is fast, with a reported average speed of 12.5 cycles per byte on an Intel Core 2 processor. Keccak can resists known attacks with a minimum complexity of 2n (n is the hash size).

SHA-3 comes with 4 security levels at 224, 256, 384, 512 bits. The more hash algorithm uses bits, the more secure and longer hash code is created.

Keccak has a wide safety margin and to date, third-party cryptanalysis has shown no serious weaknesses in Keccak.