Cryptographic Hash Functions

Generally, a hash function is used to compress arbitrary-length strings into shorter strings, in order to achieve O (1) insertion and lookup times for storing a set of elements [17]. However, since the amount of data which are going to compressed is very large, collisions are unavoidable. Therefore, simple hash functions are not good candidates to use in cryptography. A *good* hash function has to supply a unique output for every possible input, therefore minimizing the possibility of collision. Those kind of hash functions are classed as *collision-resistant hash functions*. Designing collision-resistant hash functions are not as easy as creating a regular hash function, where the main purpose of it is to compress files as a data structure. However, in order to use these hash functions in cryptography, collision-resistance is a must and therefore has a more advanced design [17].

According to Katz and Yindell [17], there are three levels of security is considered when considering cryptographic hash functions: collision resistance, second pre-image resistance, pre-image resistance.

- **Collision resistance**: It should be computationally infeasible to find a pair of different input values (m, m') to have the same digest.
- **Second pre-image resistance**: It should be computationally infeasible to find a message m', to hash to the same output as message m.
- **Pre-image resistance**: It should be computationally infeasible to find a message m', which hashes to a specific output, y = H(m).

Here, if a hash function is collision resistant, it is also pre-image resistant; because if there is a 2^{nd} preimage, that means there is a colliding pair. Also, a pre-image resistant function is called a one-way function, since it is difficult to inverse it.

One-way function means that there is no inverse on that function, meaning that you cannot find message m by looking at the digest, y; where H(m) = y. That is why this cryptographic hash functions are used in many information security areas such as digital signatures, message authentication codes, fingerprinting, checksums, and many more. [18-19-20-21]

As stated earlier, regular hash functions have the purpose to compress the data, where cryptographic hash functions may actually increase the size of data. Since cryptographic hash functions has fixed output, a small size input may get bigger in size.

Therefore, a regular hash function can be used as follows to map a string to a value;

Plaintext	**Hash**
Ahmet	00
Mehmet	01
Ayşe	02
Cenk	03

Where a cryptographic hash function would be very different;

Plaintext	**SHA-256 Hash**
A	559aead08264d5795d3909718cdd05abd49572e84fe55590eef31a88a08fdffd
ABC	b5d4045c3f466fa91fe2cc6abe79232a1a57cdf104f7a26e716e0a1e2789df78
ACC	2dde6d03ddaa287e3ac95b4befe8f38e8766a35092fc01e47e28d86c7a29e412
ART	be00235a757aa35d7ca676e16cfb18b870ad3e924df1841e052762e51bc6f8b1

By this table, it can be observed that every SHA-256 hash is 256-bits. Even if the plaintext input changes, the output will be of fixed length. It should also be noted that while there are minor changes in plaintext, corresponding hashes have serious difference between them, which indicates that SHA-256 algorithm has good avalanche effect.

SHA, is an abbreviation of Secure Hash Algorithms and SHA-256 belongs to a set of cryptographic hash functions called SHA-2; designed by National Institute of Standards and Technology (NIST). It is an iterative, one-way hash function producing a digest [22]. Gilbert and Handschuh also showed that SHA-256 provides a better security level than previous hash functions [23].

In Bitcoin blockchain, SHA-256 is used in both creating bitcoin addresses, and also in proof-of-work algorithm.

In Ethereum blockchain, although some sources in the Internet claim they use SHA-3; they actually do not use SHA-3 by NIST standards, but KECCAK-256.