BLOCKCHAIN

EXPLAINER

Hashing

#BLOCKCHAIN

#HASHING

A hashing function is a unidirectional function that transforms an arbitrarily sized data input into a unique, fixed size representation.

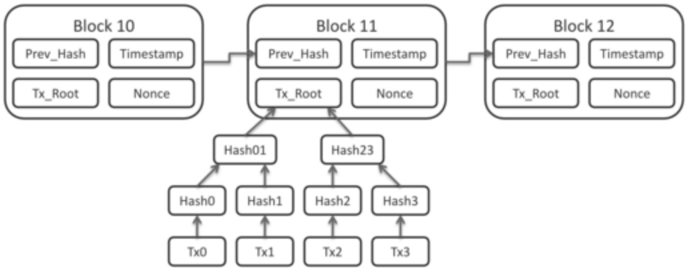
Features of Hashing Functions

- **Inputs:** Files of varying types such as text, audio, video, or program
- **Output:** Fixed string of numbers called a hash. Note: slight changes to an input will produce a completely different hash
- **Pre-Image Resistance:** Secure cryptographic hashing functions are one-way, which means that it is mathematically improbable to determine the input from the output and infeasible to learn about the input data form the hash
  - *For example:* The hash for data “MattBen01” should be very different than the hash “MattBen02” and very different than hashing “Matt” or “Ben”
- **Collision Resistance:** Collisions describe when two different inputs result in the same hash. Collision resistance means that collisions are hard to find or rare
- **Deterministic:** Hashing the same value twice will result in the same hash both times
- **Speed:** Computing a hash should be a fast computation while determining the input for a specific hash
- should take a very long time
  - *Note:* A key difference between encryption and hashing is that encrypted strings can be reversed, or decrypted to their original form with the right key

Popular Hashing Algorithms

- **SHA-256:** Secure Hashing Algorithm 2 generates an almost-unique, 256-bit hash (32 byte). SHA-2 consists of a set of 6 hashing algorithms where SHA-256 or above is recommended for high security applications

- **MD5:** The Message-Digest 5 algorithm is a hash function that produces a 128-bit (16-byte) hash value



Basic Applications of Hashing

Indexing

Used to index and retrieve items in a data base. It is faster to find an item using the short hash than it is to find the original value.

Authentication

Rather than storing a sensitive data object, such as a password, a hash is stored. Once the data object is inputted, it is hashed and then cross referenced. For example, rather than storing a user’s password, an application will store the SHA-256 hash of the password; when the user inputs the password, the system will first hash it and will grant access only if the hashes match.

Verifying Transactions

Rather than describe a transaction as “Transfer of $100 between Ben and Matt on Nov 1 2016 at 14:00”, the transactions are referred to by their hash. Any change to the transaction will cause a change to the hash. In blockchain application, one cannot change a past transaction as it will change the hash and thus further blocks in the chain.