CS61065: Theory and Applications of Blockchain

Basic Crypto Primitives - I

Department of Computer Science and Engineering

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

Sandip Chakraborty
sandipc@cse.iitkgp.ac.in

Shamik Sural
shamik@cse.iitkgp.ac.in

What You'll Learn

- Basic cryptographic primitives behind the blockchain technology
  - **Cryptographically Secure Hash Function**
  - **Digital Signature**

- **Hash Function:** Used to connect the "blocks" in a "chain" in a tamper-proof way

- **Digital Signature:** Digitally sign the data so that no one can "deny" about their own activities. Also, others can check whether it is authentic.

Cryptographic Hash Functions

- Takes any arbitrarily sized string as input
  - Input M: The message
- Fixed size output (We use 256 bits in Blockchain)
  - Output H(M): We call this as the message digest
- Efficiently computable

Cryptographic Hash Function: Properties

Deterministic

Always yield identical hash value for identical input data

Collision-Free

If two messages are different, then their digests also differ

Hiding

Hide the original message; remember about the **avalanche effect**

Puzzle-friendly

Given *X* and *Y*, find out *k* such that $Y = H(X||k)$ - used to solve the mining puzzle in Bitcoin Proof of Work

Collision Free

Hash functions are one - way; Given an x, it is easy to find $H(x)$. However, given an $H(x)$, **no deterministic algorithm** can find x

It is **difficult to find** x and y, where $x \neq y$, but $H(x) = H(y)$

Note the phrase **difficult to find**, collision is **not impossible**

Try with randomly chosen inputs to find out a collision – but it takes too long

Collision Free – How Do We Guarantee

It may be relatively easy to find collision for some hash functions

Birthday Paradox: Find the probability that in a set of n **randomly chosen persons,** some of them will have the same birthday

- By *Pigeonhole Principle*, the probability reaches 1 when number of people reaches 366 (not a leap year) or 367 (a leap year)
- 0.999 probability is reached with just ~70 people, and 0.5 probability is reached with only ~23 people

Collision Free – How Do We Guarantee

Birthday paradox places an upper bound on collision resistance

If a hash function produces N bits of output, an attacker need to compute only $2^{\frac{N}{2}}$ hash operations on a random input to find two matching outputs with probability > 0.98

For a 256 bit hash function, the attacker needs to compute 2^{128} hash operations – this is significantly time consuming
If every hash computation takes only 1 microsecond, it will need $\sim 10^{25}$ years

Hash as A Message Digest

If we observe $H(x) = H(y)$, it is safe to assume $x = y$

We need to remember just the hash value rather than the entire message – we call this as the **message digest**

To check if two messages x and y are same, i.e., whether $x = y$, simply check if $H(x) = H(y)$

This is efficient because the size of the digest is significantly less than the size of the original messages

Hashing - Illustration

- http://www.blockchain-basics.com/HashFunctions.html

Courtesy: Blockchain Basics: A Non-Technical Introduction in 25 Steps by Daniel Drescher

Information Hiding through Hash

Given an $H(x)$, it is “computationally difficult” to find x

The difficulty depends on the size of the message digests

Hiding helps to commit a value and then check it later

- Compute the message digest and store it in a digest store – commit
- To check whether a message has been committed, match the message digest at the digest store