NPTEL ONLINE CERTIFICATION COURSES

Blockchain and its applications

Prof. Shamik Sural
Department of Computer Science & Engineering
Indian Institute of Technology Kharagpur

Lecture 03: Basic Cryptographic Primitives - I

CONCEPTS COVERED

- Cryptographic Primitives useful for Blockchain
- Hash Functions

KEYWORDS

- Hash Function
- SHA-256
- Puzzle Friendly

What You'll Learn

- Basic cryptographic primitives behind blockchain technology
  - **Cryptographically Secure Hash Functions**
  - **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 typically use 256 bits in Blockchain)
  Output H(M): We call this as the message digest

- **Efficiently computable**

Cryptographic Hash Functions: Properties

- **Deterministic**
  Always yields 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 - used to solve the mining puzzle in Bitcoin Proof of Work

Collision Free

- Hash functions are one-way; Given an , it is easy to find . However, given an , **one cannot find**
- It is **difficult to find** and , where , but
- 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 **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 bits of output, an attacker needs to compute only 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 hash operations – this is significantly time consuming
- If every hash computation takes only **1 microsecond**, it will need years

Hash as a Message Digest

- If we observe , it is safe to assume
- 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 and are same, , simply check if
- This is efficient because the **size of the digest is significantly less than the size of the original messages**

Hashing - Illustration

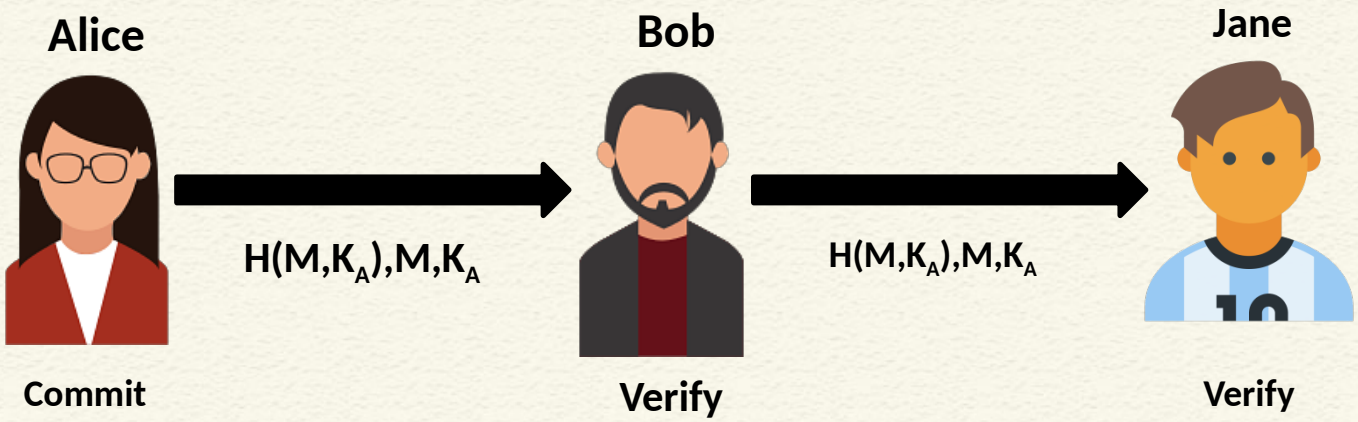
http://www.blockchain-basics.com/HashFunctions.html

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

Information Hiding through Hashing

- Given an , it is “computationally difficult” to find
- 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

Message Commitment through Multiple Parties



K_A is the public key of Alice – A public identity that only Alice can have

Puzzle Friendly

- Say is chosen from a widely spread distribution; it is computationally difficult to find a , such that , where and are known a priori.
- **A Search Puzzle** (Used in Bitcoin Mining)
  and are given, is the search solution
  Note: It might be not exactly a particular value Z, but some properties that Z satisfies, i.e., Z could be a set of possible values
- Puzzle friendly property implies that random searching is the best strategy to solve the above puzzle

CONCLUSIONS

- Discussed what a cryptographic hash function is
- Properties of hash functions
- Uses of hash functions

REFERENCES

- Blockchain Basics: A Non-Technical Introduction in 25 Steps by Daniel Drescher, Apress (2017)
- Cryptography and Network Security – Principles and Practice by William Stallings, Pearson (2017)


Thank
you