

Lecture 1 - Cryptographic Hash Functions

Rylan Schaeffer and Vincent Yang

March 23, 2016

Note: This lecture is based on Princeton University's BTC-Tech: Bitcoin and Cryptocurrency Technologies Spring 2015 course.

Terms

- Set : group of objects represented as a unit
- Alphabet : finite, non-empty set
- String : finite sequence of characters from common alphabet, including empty string ε
- Language : set of strings over common alphabet

Hash Functions

- $H : \{0, 1\}^* \rightarrow \{0, 1\}^k$, for fixed k e.g. 256
- Should be efficiently computable $O(n)$
- Example: mod operator

Cryptographic Hash Function

Two additional properties

- Collision Resistant: Computationally infeasible to find x, y such that $x \neq y$ and $H(x) = H(y)$
 - mod operator is not collision resistant
 - collisions exist by pigeonhole principle - hence, computationally infeasible
 - birthday paradox reduces difficulty of finding collisions
 - can also call "binding," since once hash is published, you cannot replace input value with another input value without modifying the hash output
- Hiding: Computationally infeasible to find x given H_{given} such that $H(x) = H_{given}$
 - Frequently, cryptographic hash functions will be called one-way hash functions
 - Frequently, message space is too small. Append nonce (i.e. random value) r to grow message space such that computationally infeasible to find x such that $H(x|r) = H_{given}$

Applications

- Message Digest

Create summary (or “digest”) of block of text

Suppose I have msg and H is a cryptographic hash function. Then I know that $H(msg)$ or perhaps $H(msg|r)$ (where r is a random value and is needed because the message is predictable), will produce a hash value that no other block of text will.

Example: cryptographic checksums

- Commitments

Analogous to sealed envelope on the table

Hiding ensures no one can “reverse engineer” the contents. Collision-resistant guarantees to the other party that you are bound to the value you initially put in.

Puzzle Friendliness

- Search Puzzle

Given H , target set Y , and value x

Goal: find r such that $H(x|r) \in Y$

- Puzzle friendly if no solving strategy for puzzle other than trying random guesses at r

- Examples: $0|0,1\}^{k-1}$, $00|0,1\}^{k-1}$, $000|0,1\}^{k-1}$

$P(\text{1 leading zeroes}) = \frac{1}{2^l}$, can use geometric distribution’s cumulative distribution function to model likelihood of observing a “hit” after a given number of failures

- Useful for mining, which we will get to later

Hash Structures

- Hash pointer : hash of data. Gives way to verify information hasn’t changed, much like pointer gives a way to retrieve location of information
- Hash linked list (block chain) : Each block has hash of previous block plus new data. Head is hash of most recent block.

Tamper-evident log

- Hash tree (Merkle Tree) : binary tree of data blocks. Proof of membership and proof of non-membership in $\log(n)$, so faster than hash linked list. Can also sort.
- Can combine. Block chain is usually hash linked list of hash trees