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 $H(x)$, cannot 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

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, $\mathrm{i.e.}$, $\mathrm{whether}\ x = y$, 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

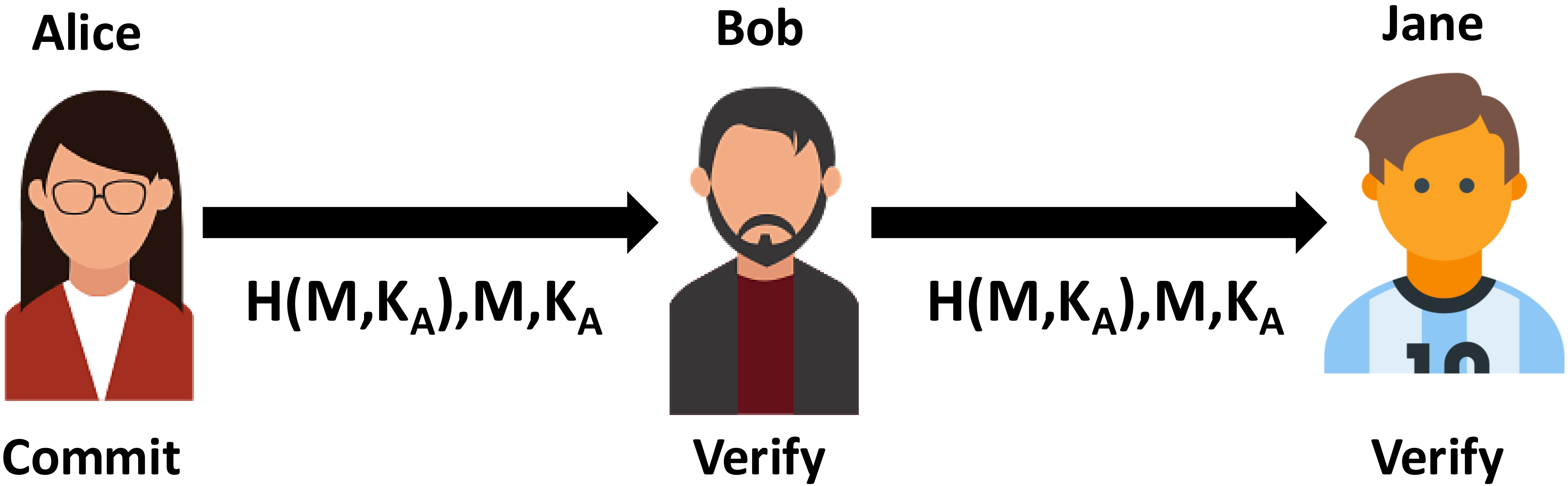
Information Hiding through Hash

- 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

Message Commitment through Multiple Parties



Alice

Bob

Jane

$H(M,K_A),M,K_A$

$H(M,K_A),M,K_A$

Commit

Verify

Verify

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

Puzzle Friendly

- Say M is chosen from a widely spread distribution; it is computationally difficult to compute k, such that $Z = H(M||k)$, where M and Z are known a priori.

- **A Search Puzzle** (Used in Bitcoin Mining)
    - M and Z are given, k 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

Hash Function – SHA256

- SHA256 is used in Bitcoin mining – to construct the Bitcoin blockchain

- Secure Hash Algorithm (SHA) that generates 256 bit message digest

- A part of SHA-2, a set of cryptographic hash functions designed by United States National Security Agency (NSA)

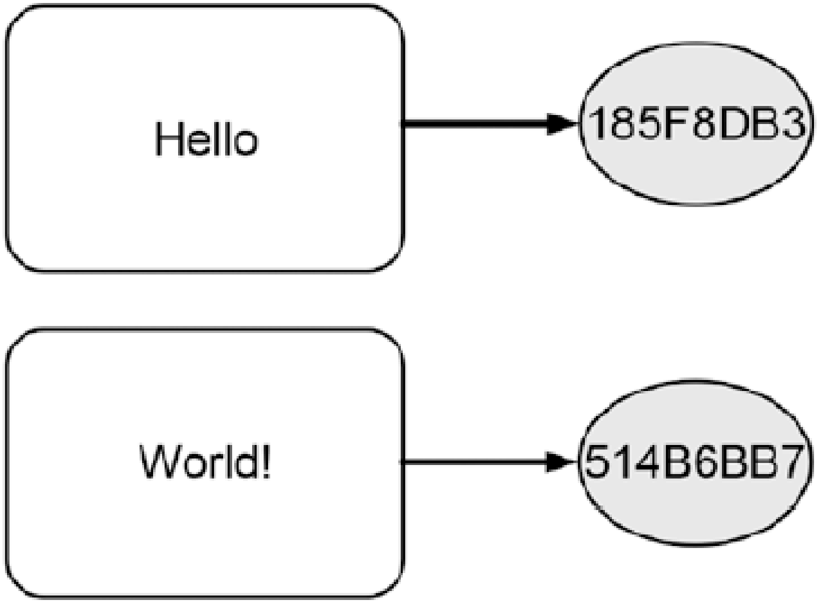
Patterns of Hashing Data

- Independent hashing
- Repeated hashing
- Combined hashing
- Sequential hashing
- Hierarchical hashing

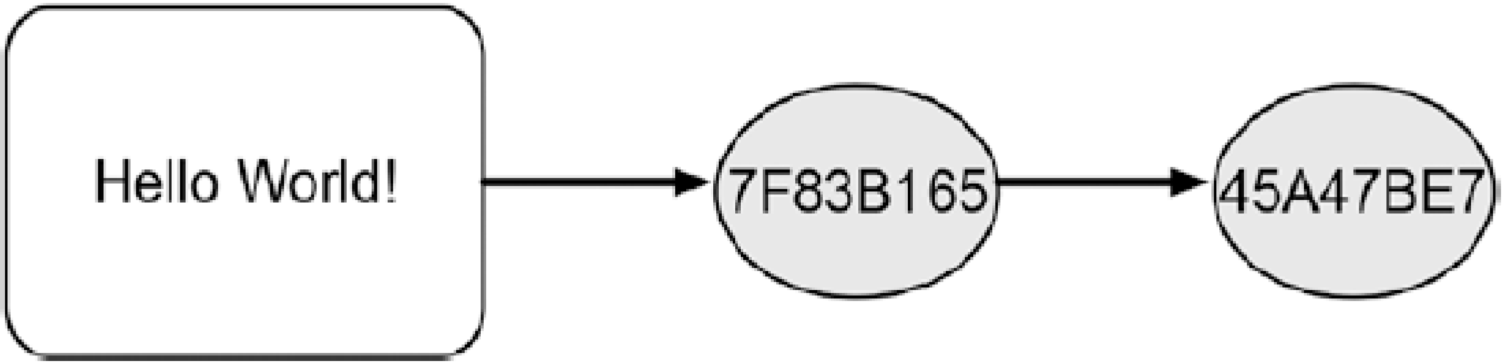
Courtesy: Blockchain Basics: A Non-Technical Introduction in 25 Steps by Daniel Drescher

Types of Hashing

- Independent hashing



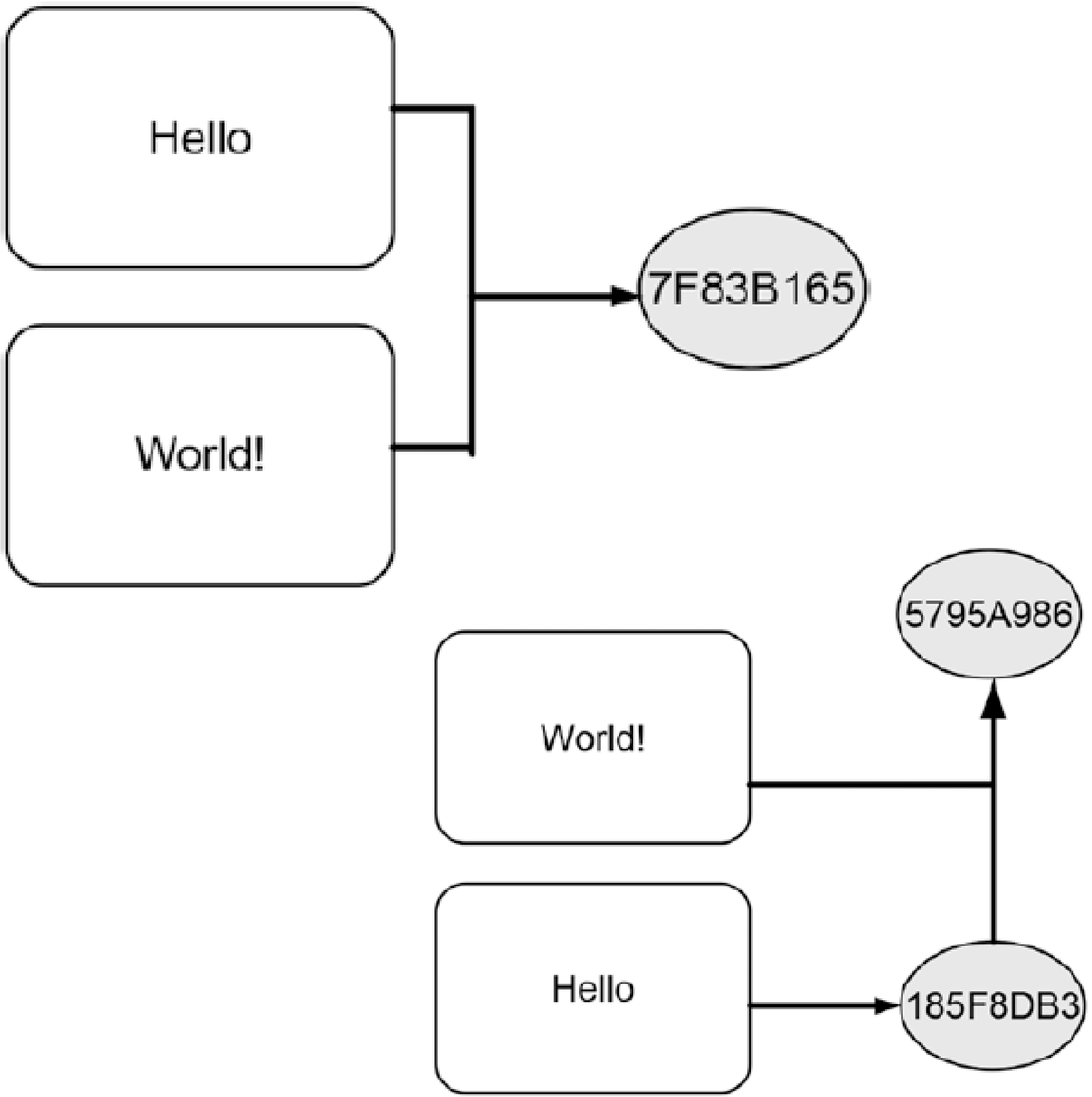
- Repeated hashing



Types of Hashing

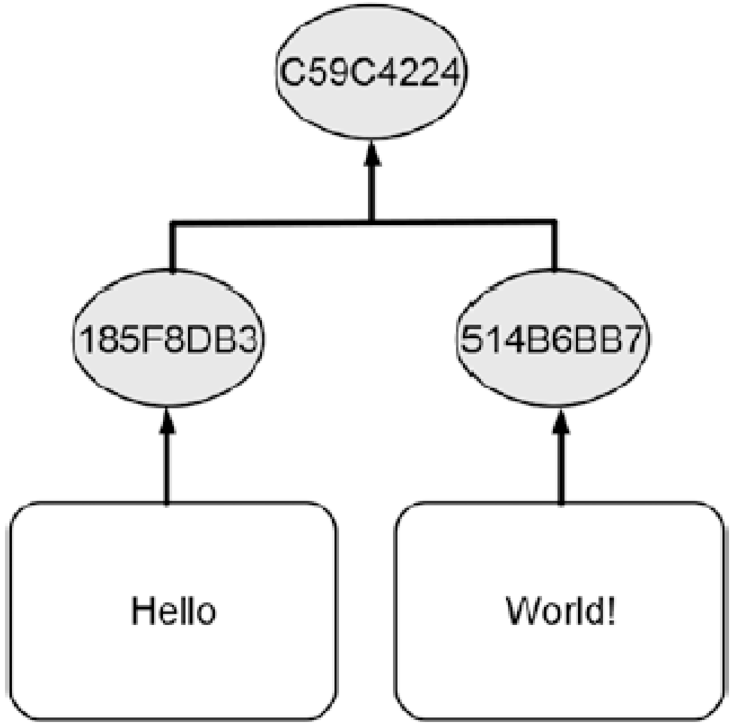
- Combined hashing

- Sequential hashing



Types of Hashing

- Hierarchical hashing

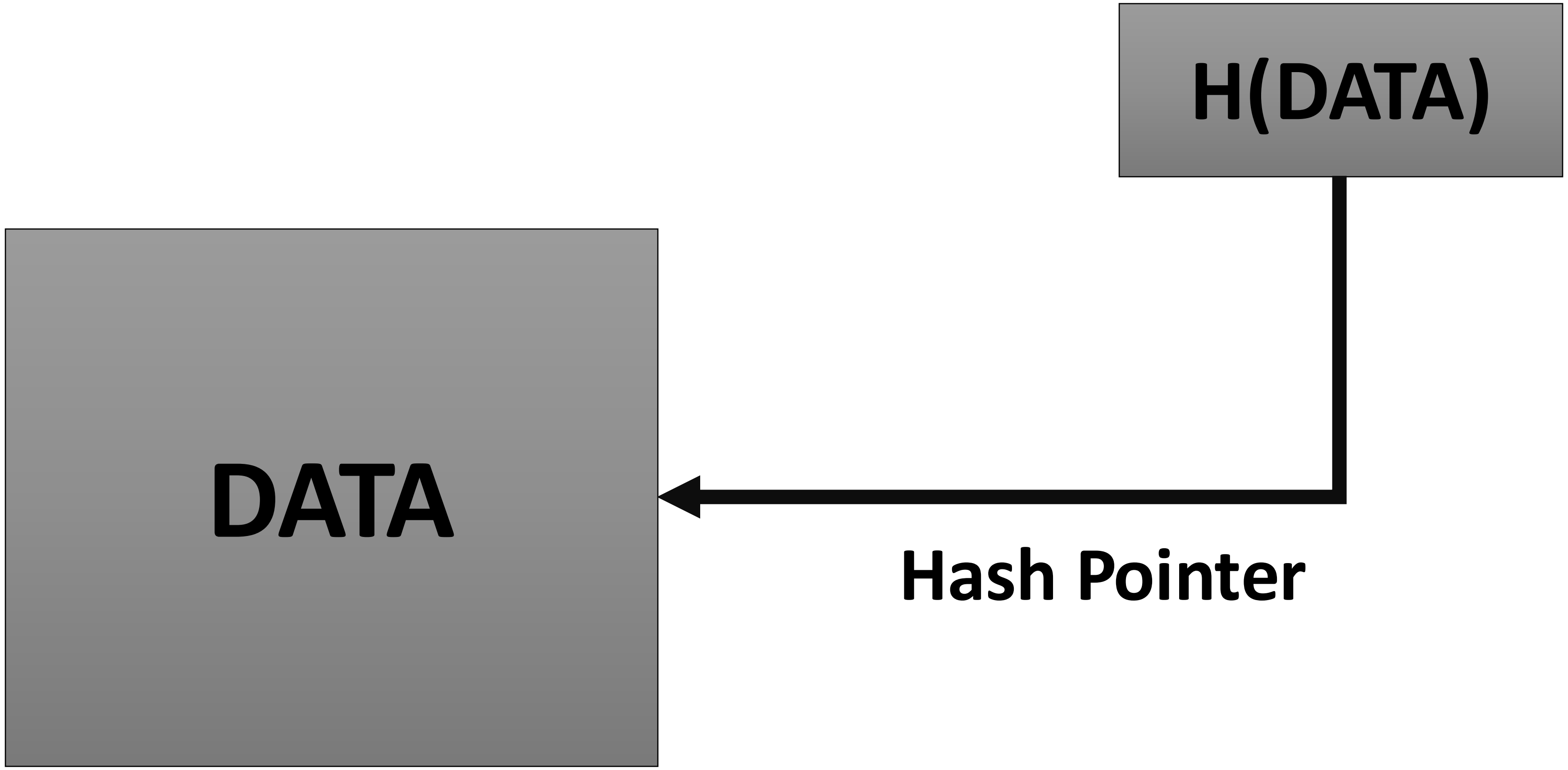


Hash Pointer

- A **Cryptographic Hash Pointer** (Often called Hash Reference) is a pointer to a location where
  - Some information is stored
  - Hash of the information is stored

- With the hash pointer, we can
  - Retrieve the information
  - Check that the information has not been modified (by computing the message digest and then matching the digest with the stored hash value)

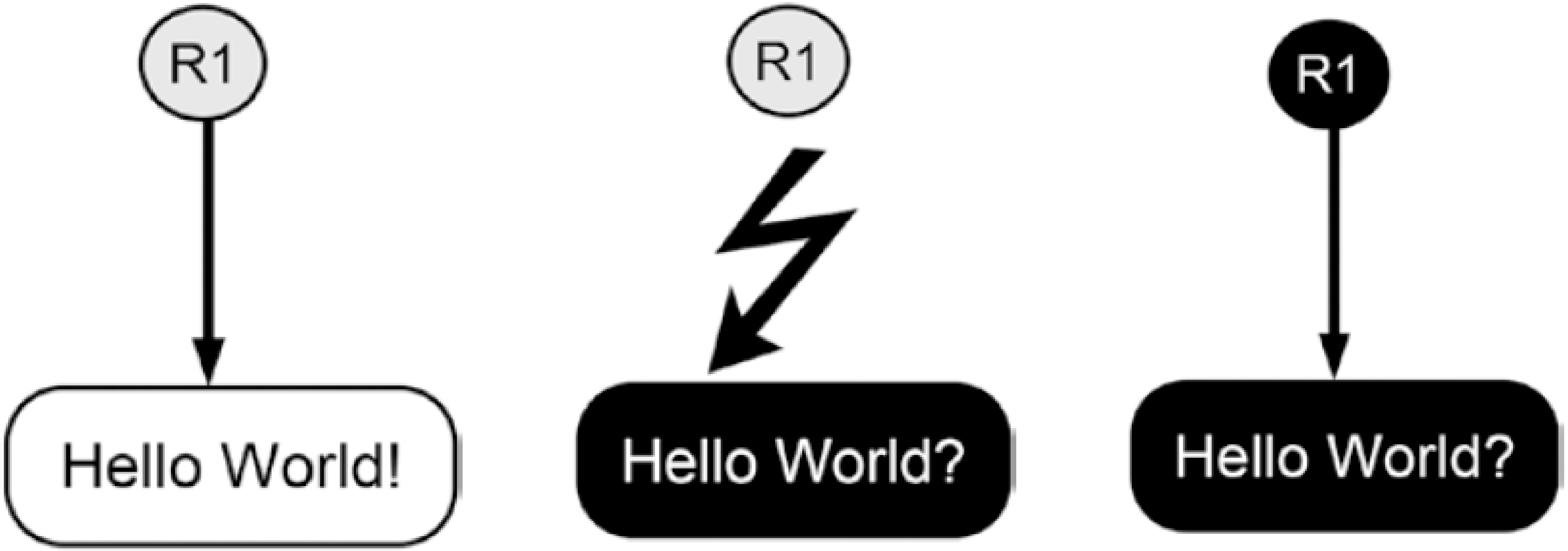
Hash Pointer



Reminds you of a linked list??

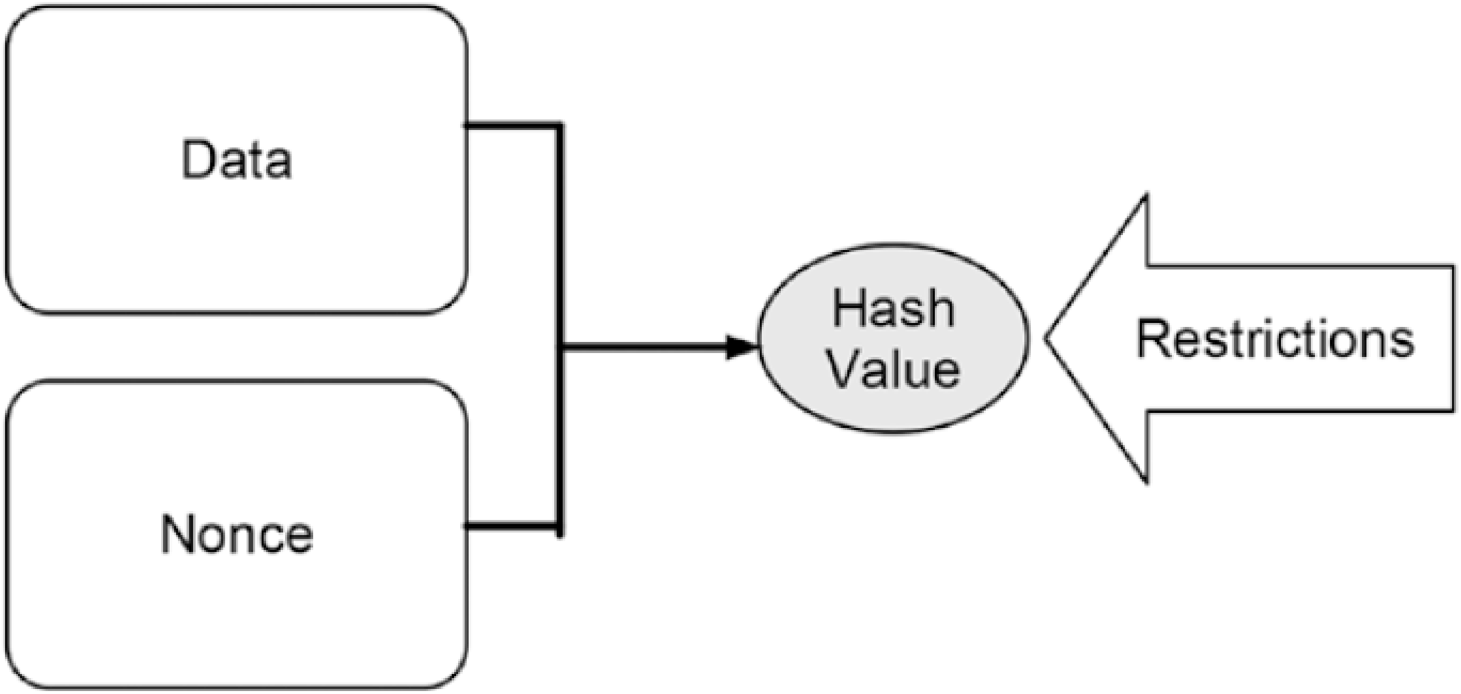
Reference: Coursera course on Bitcoin and Cryptocurrency Technologies

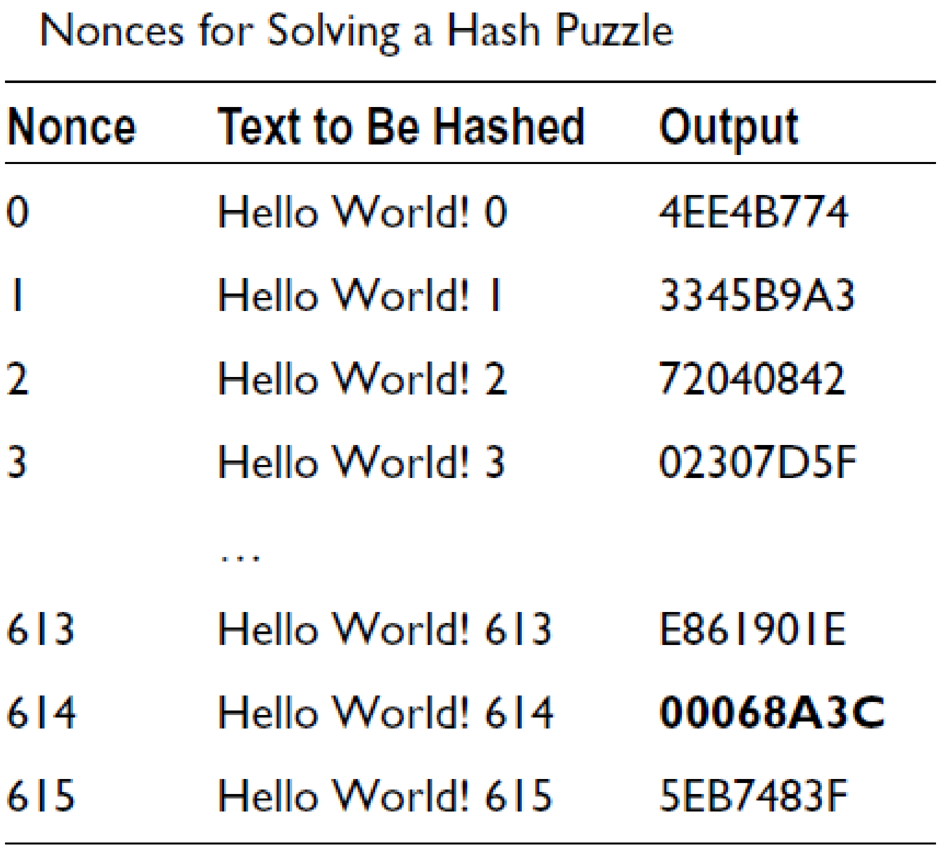
Tamper Detection using Hash Pointer



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

Making Tampering a Hash Chain Computationally Challenging





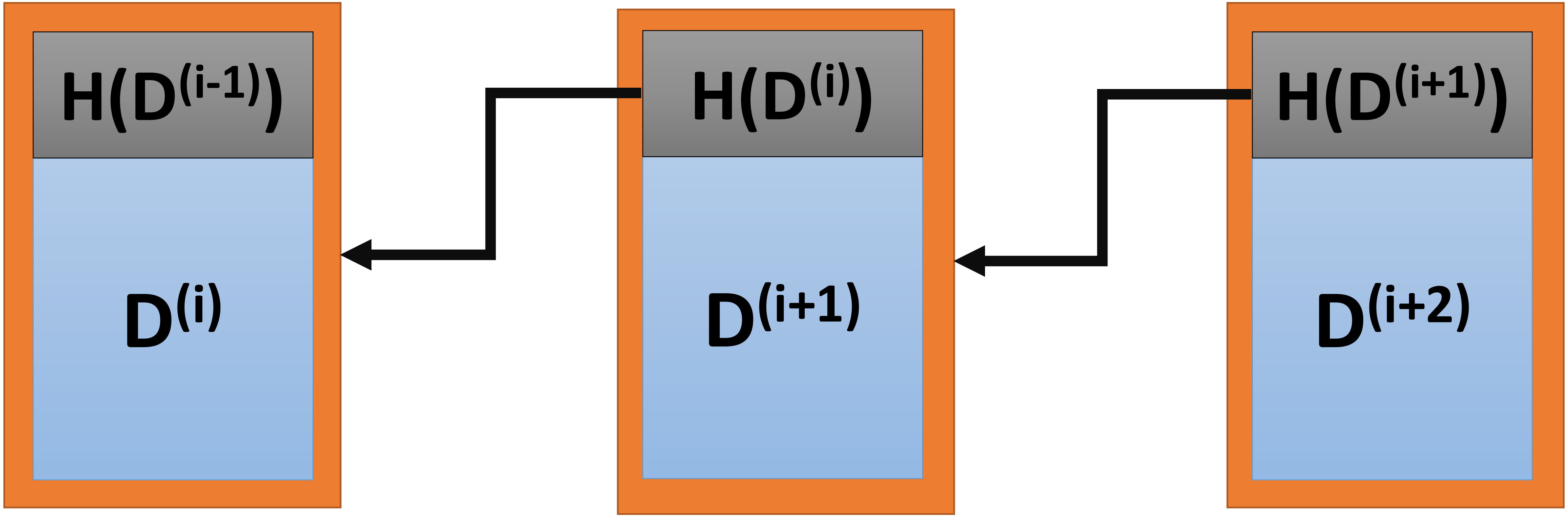
Nonces for Solving a Hash Puzzle

Nonce	Text to Be Hashed	Output
0	Hello World! 0	4EE4B774
1	Hello World! 1	3345B9A3
2	Hello World! 2	72040842
3	Hello World! 3	02307D5F
	...	
613	Hello World! 613	E861901E
614	Hello World! 614	**00068A3C**
615	Hello World! 615	5EB7483F

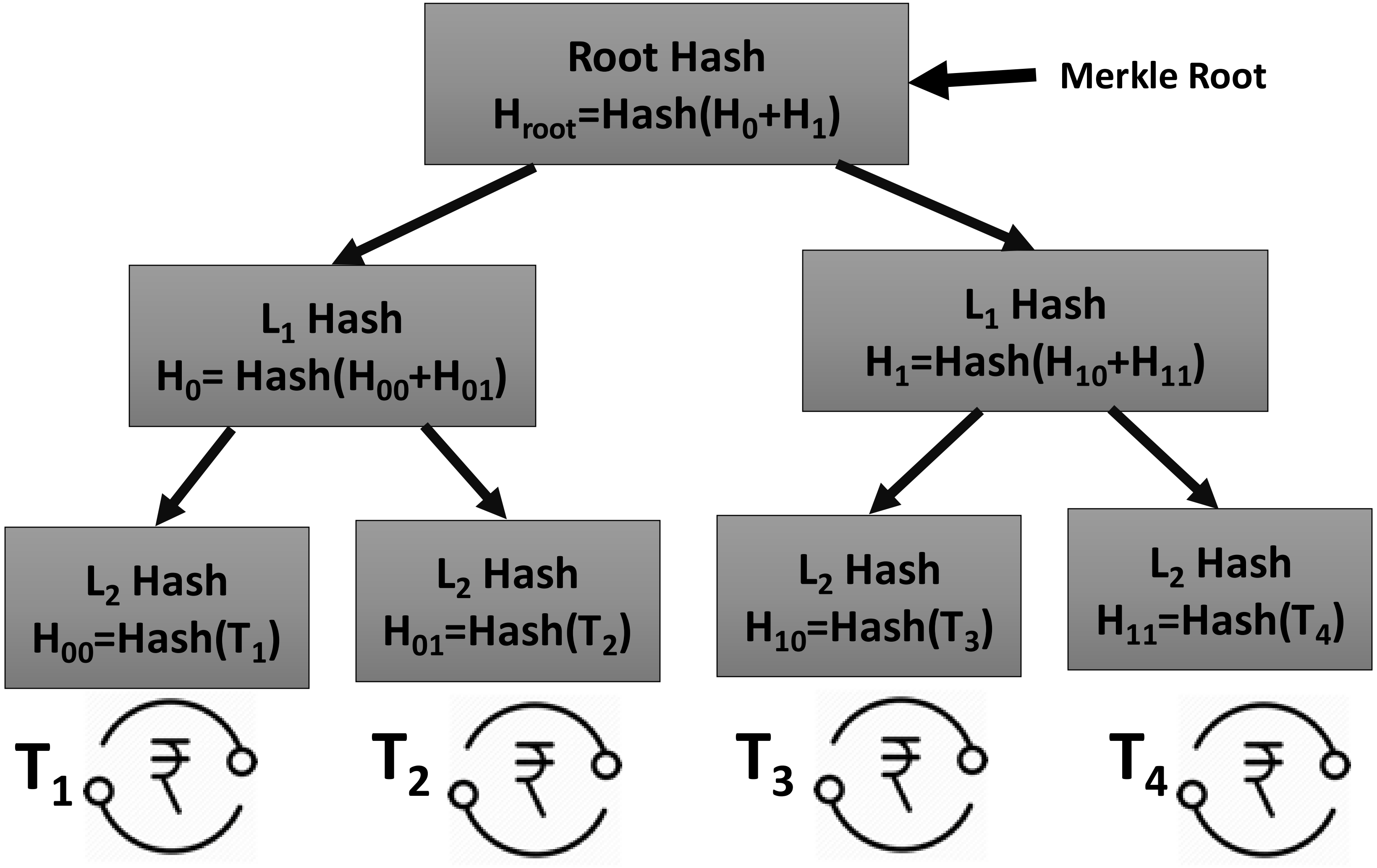
http://www.blockchain-basics.com/HashFunctions.html

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

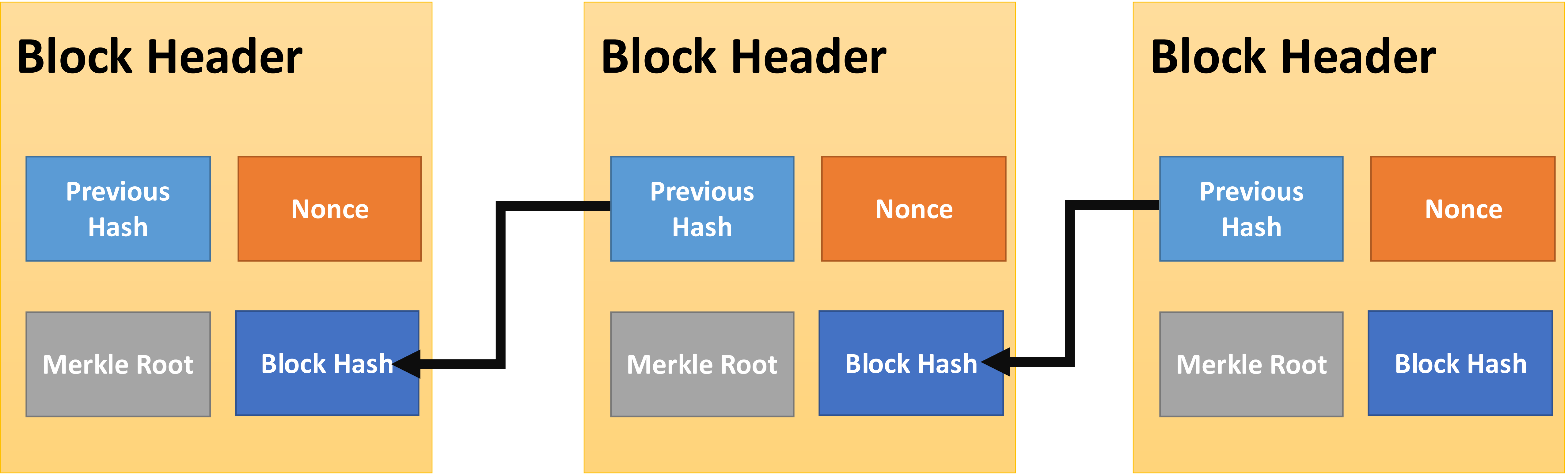
Detect Tampering from Hash Pointers - Hashchain



Merkle Tree – Organization of Hash Pointers in a Tree



Blockchain as a Hashchain



thank you!