CS61065: Theory and Applications of Blockchain

Basic Crypto Primitives

Department of Computer Science and Engineering

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

Sandip Chakraborty
sandipc@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, $\mathrm{i.e.}$, 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

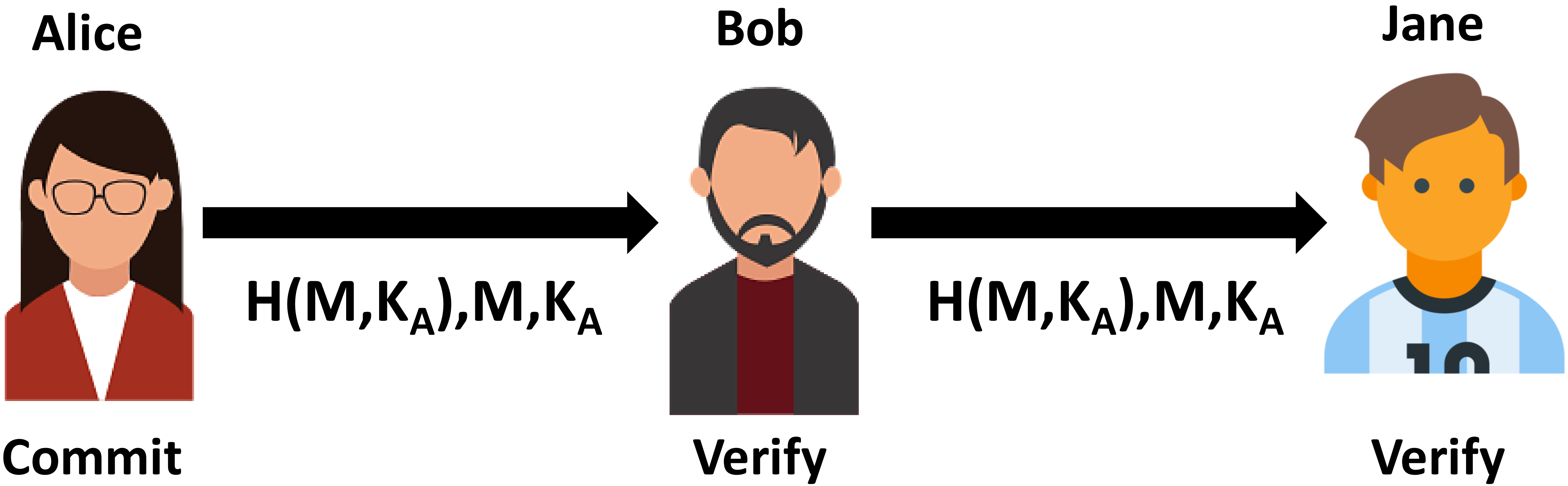
- http://www.blockchain-basics.com/HashFunctions.html

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

Message Commitment through Multiple Parties



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)

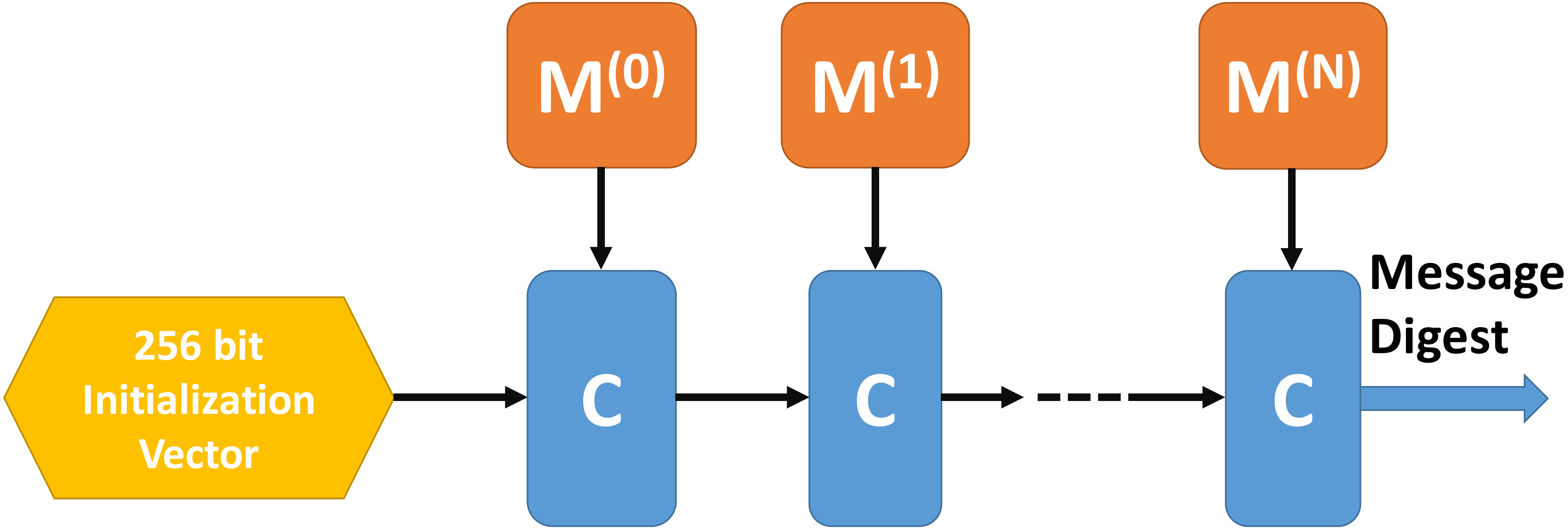
SHA256 Algorithm - Preprocessing

- **Pad the message such that the message size is a multiple of 512**
  - Suppose that the length of the message M is l; and $l \ mod \ 512 \neq 0$
  - Append the bit "1" at the end of the message
  - Append k zero bits, where k is the smallest non-negative solution to the equation $l + 1 + k \equiv 448 \ mod \ 512$
  - Append the 64-bit block which is equal to the number l written in binary
  - The total length gets divisible by 512
- **Partition the message into N 512-bit blocks $M^{(1)}, M^{(2)}, \ldots, M^{(N)}$**
- **Every 512 bit block is further divided into 32 bit sub-blocks $M_0^{(i)}, M_1^{(i)}, \ldots, M_{15}^{(i)}$**

SHA-256 Algorithm

- The message blocks are processed one at a time
- Start with a fix initial hash value $H^{(0)}$
- Sequentially compute $H^{(i)} = H^{(i-1)} + C_{M^{(i)}}(H^{(i-1)})$; C is the SHA-256 *compression function* and + means mod 2^{32} addition. $H^{(N)}$ is the hash of M.

SHA-256 Algorithm



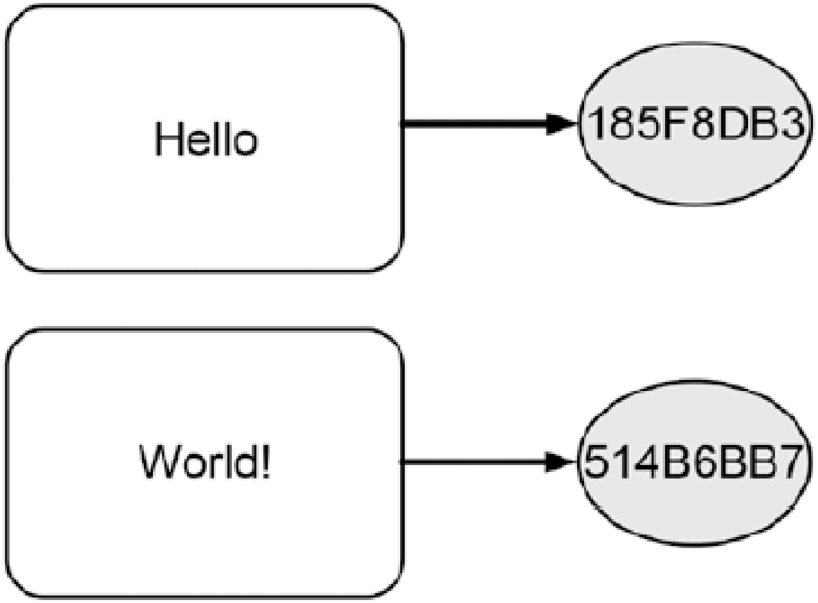
Patterns of Hashing Data

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

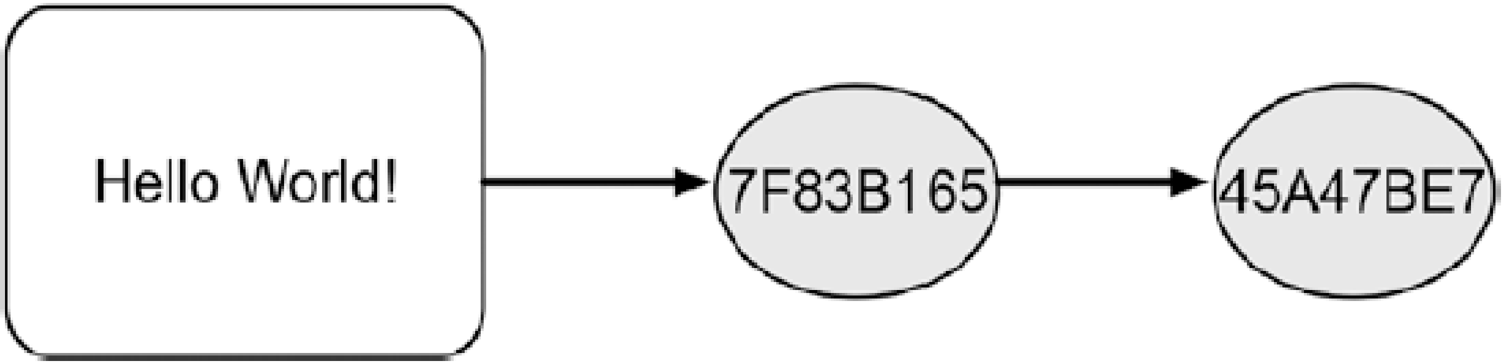
Courtesy: Blockchain Basics: A Non-Technical Introduction in 25 Steps by Daniel Drescher

Types of Hashing

- Independent hashing

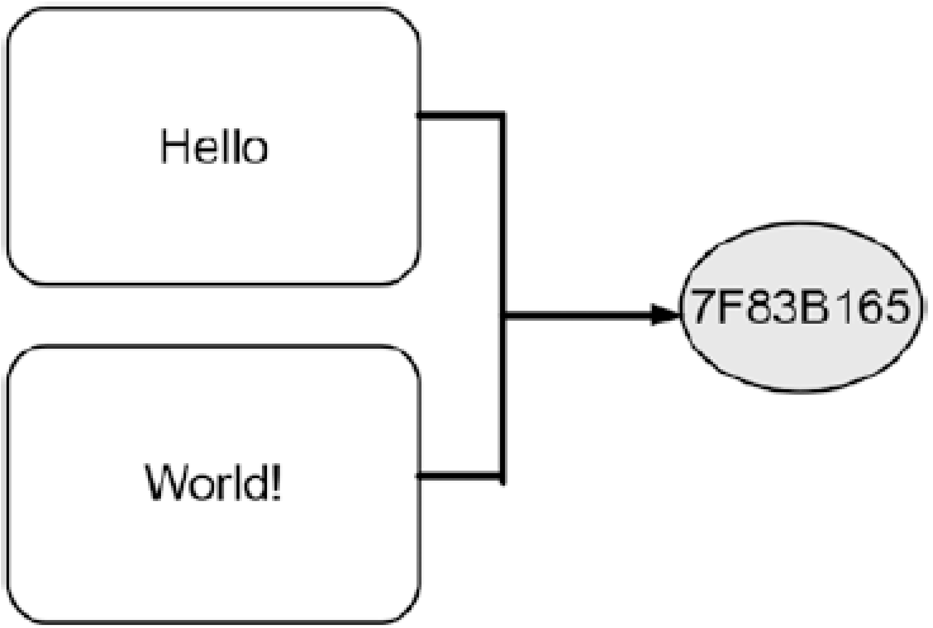


- Repeated hashing

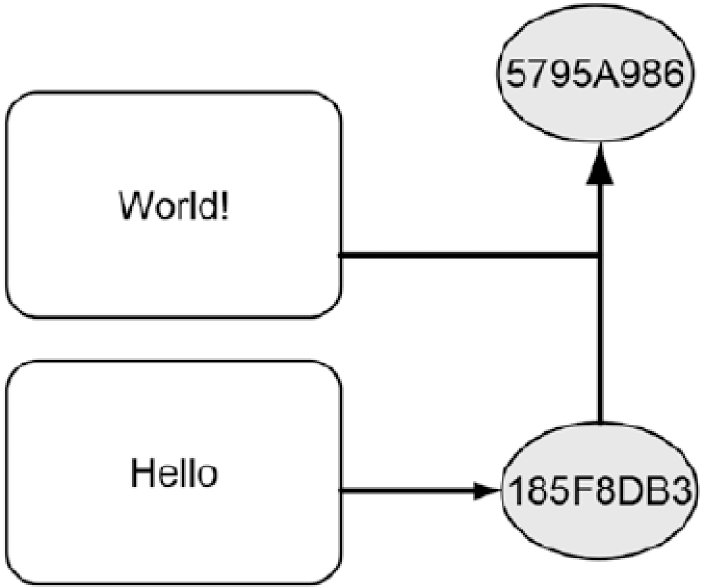


Types of Hashing

- Combined hashing

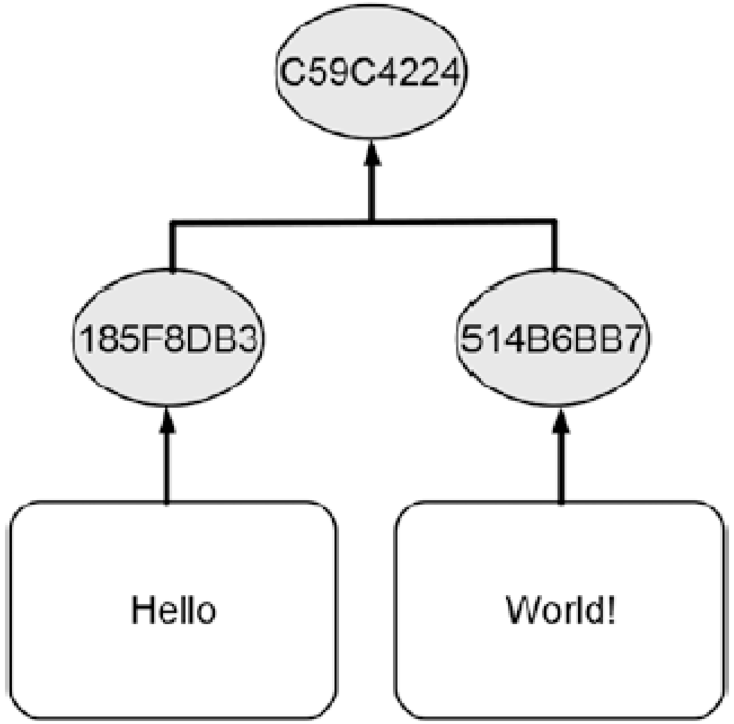


- Sequential hashing



Types of Hashing

- Hierarchical hashing

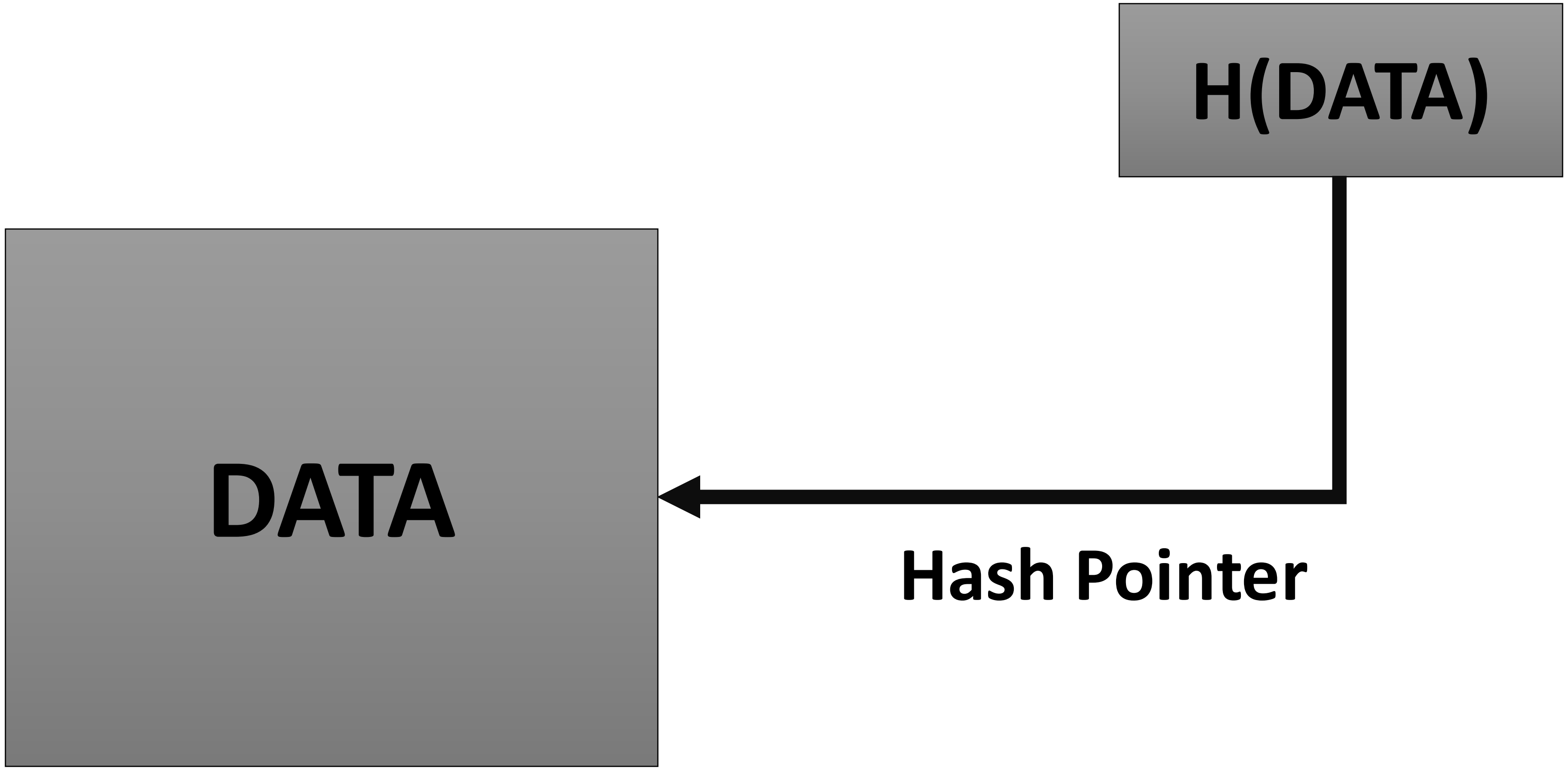


Hash Pointer

- 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)

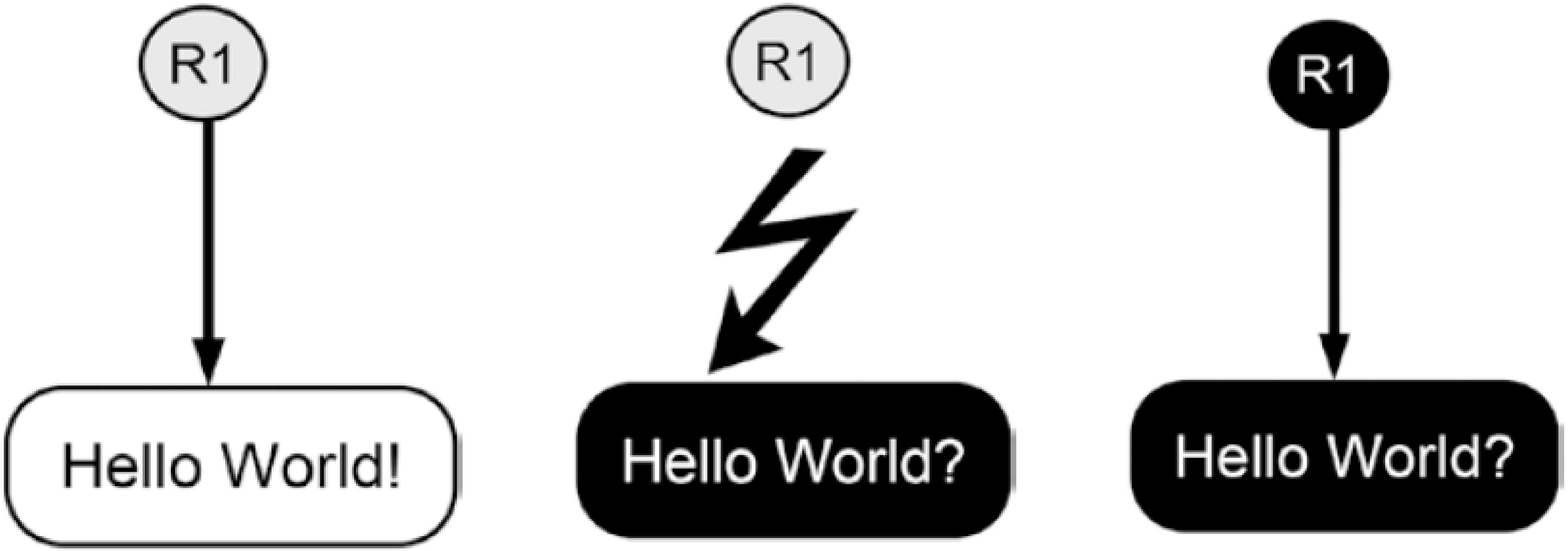
Hash Pointer



Reminds you of a linked list??

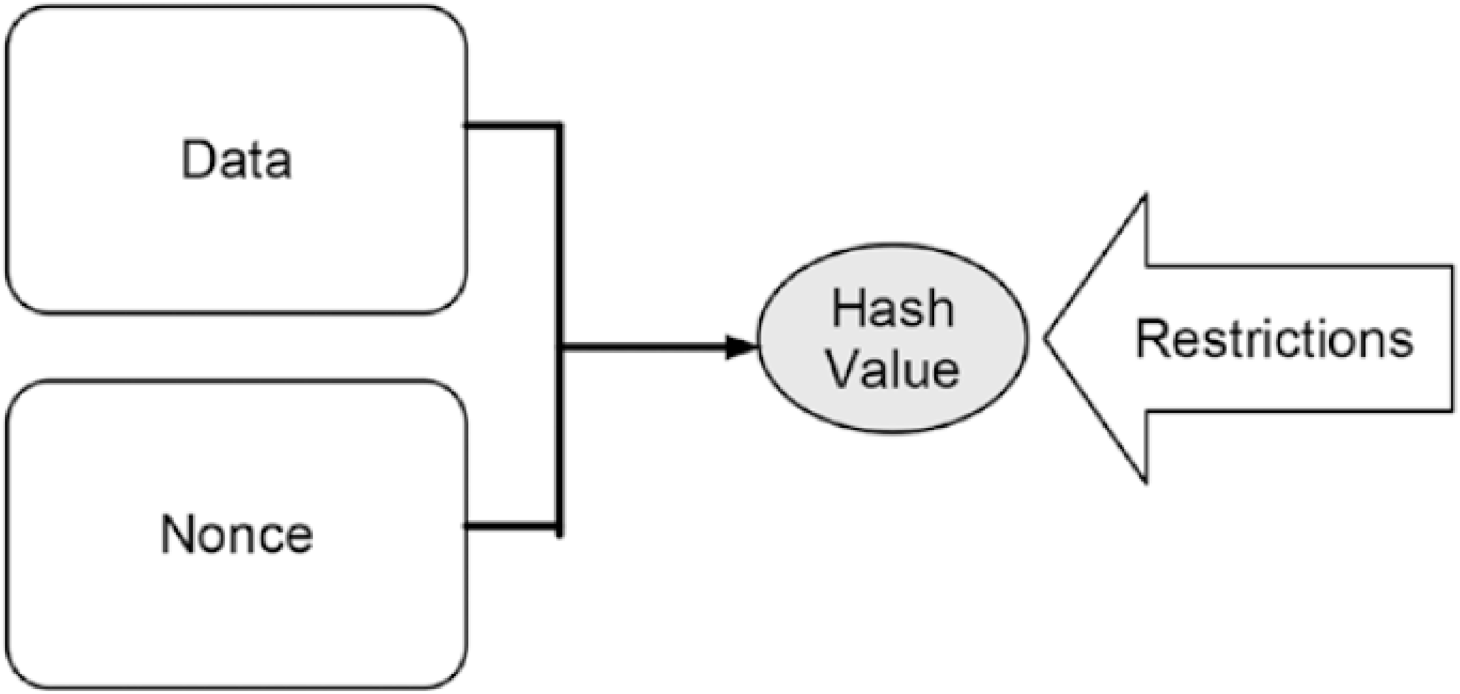
Reference: Coursera course on Bitcoin and Cryptocurrency Technologies

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



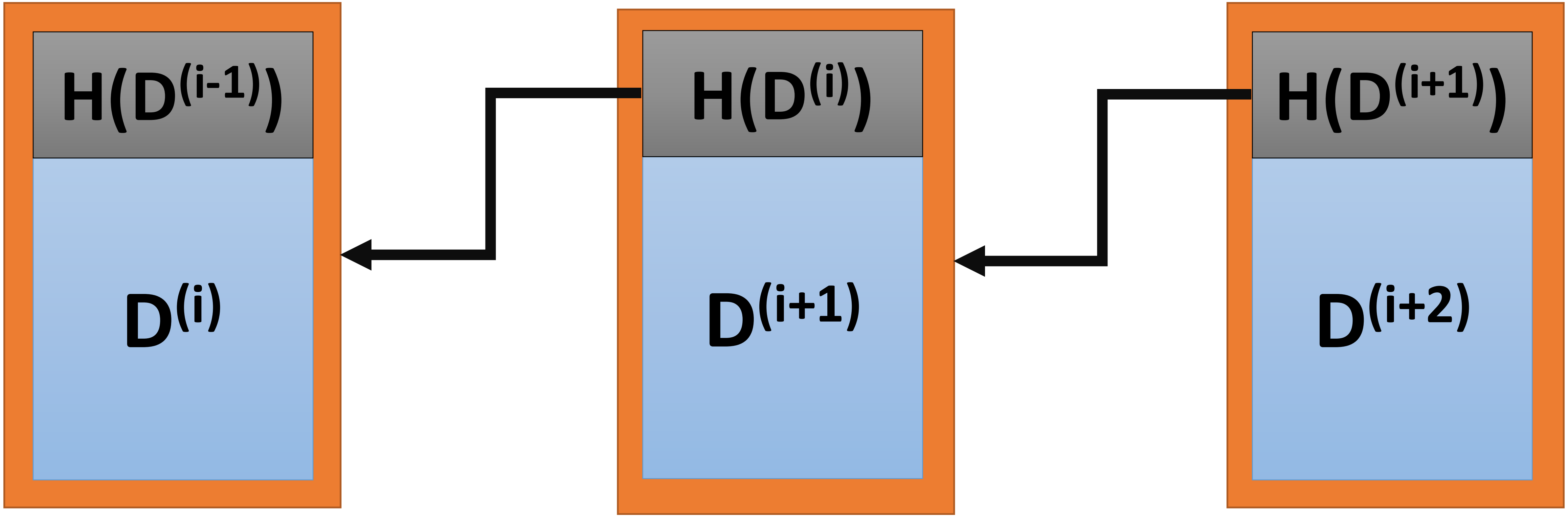
http://www.blockchain-basics.com/HashFunctions.html

Nonces for Solving a Hash Puzzle

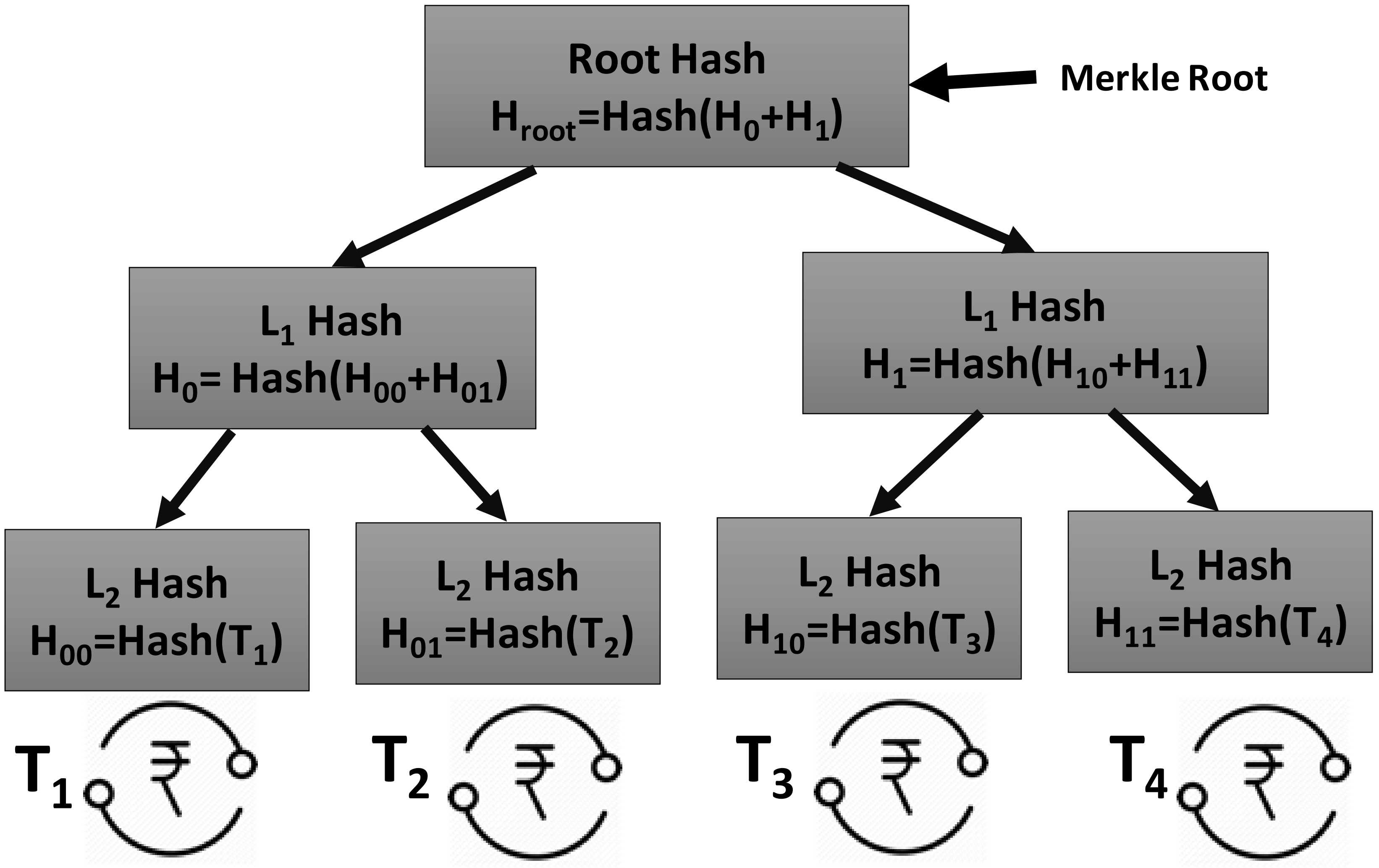
Nonce	Text to Be Hashed	Output
0	Hello World! 0	4EE4B774
1	Hello World! 1	3345B9A3
2	Hello World! 2	72040842
3	Hello World! 3	02307D5F
	...	
613	Hello World! 613	E861901E
614	Hello World! 614	**00068A3C**
615	Hello World! 615	5EB7483F

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

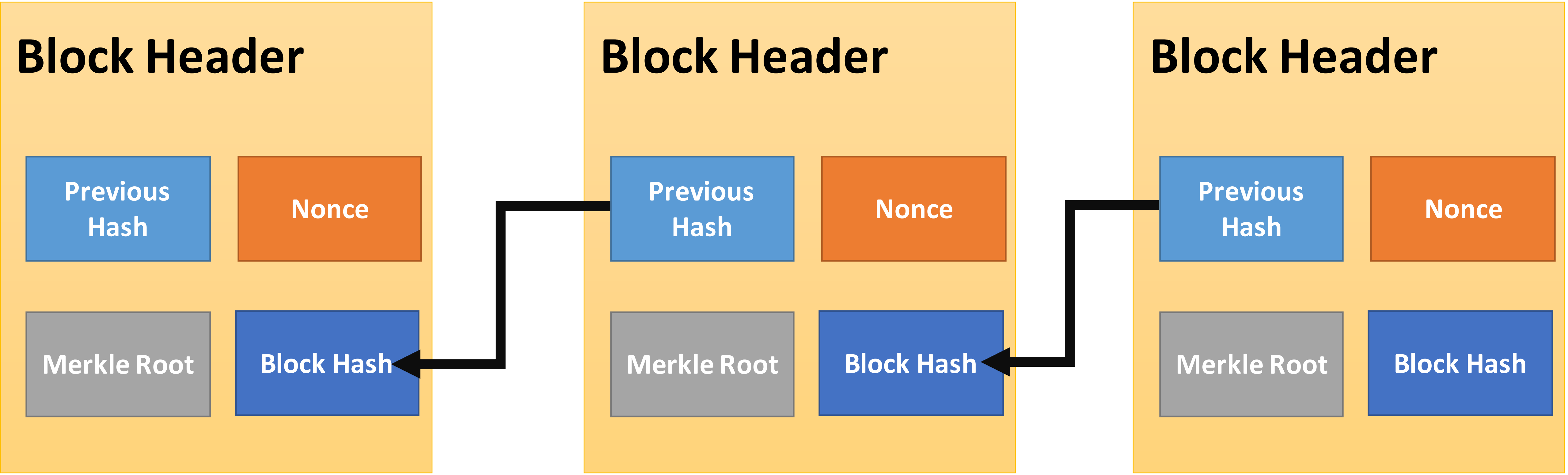
Detect Tampering from Hash Pointers - Hashchain



Merkle Tree – Organization of Hash Pointers in a Tree



Blockchain as a Hashchain



Digital Signature

- A **digital code**, which can be included with an electronically transmitted document to verify
  - The content of the document is authenticated
  - The identity of the sender
  - Prevent *non-repudiation* – sender will not be able to deny about the origin of the document

Purpose of Digital Signature

- Only the **signing authority** can sign a document, but everyone can verify the signature

- Signature is **associated with** the particular document
  - Signature of one document cannot be transferred to another document

Public Key Cryptography

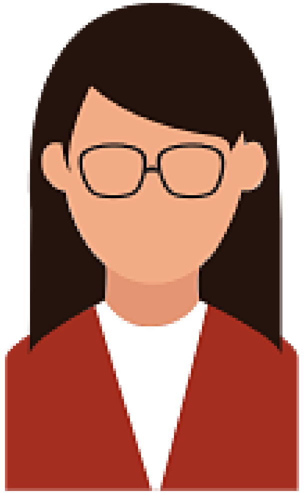
- Also known as **asymmetrical cryptography** or **asymmetric key cryptography**
- **Key:** A parameter that determines the functional output of a cryptography algorithm
  - **Encryption:** The key is used to convert a plain-text to a cypher-text; $M' = E(M, k)$
  - **Decryption:** The key is used to convert the cypher-text to the original plain text; $M = D(M', k)$

Public Key Cryptography

- Properties of a cryptographic key (you need to prevent it from being guessed)
  - Generate the key truly randomly so that the attacker cannot guess it
  - The key should be of sufficient length – increasing the length makes the key difficult to guess
  - The key should contain sufficient entropy, all the bits in the key should be equally random

Public Key Cryptography

- Two keys are used
  - **Private key**: Only Alice has her private key
  - **Public key:** “Public” to everyone – everyone knows Alice’s public key



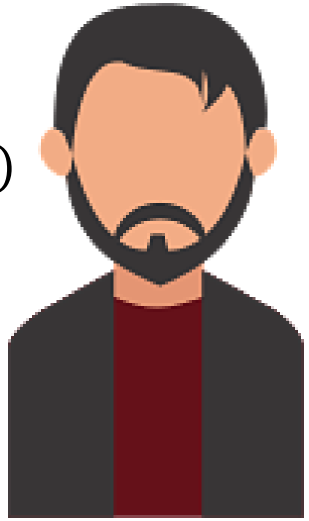
Encrypt the message with Bob’s public key

$M' = E(M, K^{B}_{pub})$

M'

Decrypt the message with his private key

$M = E(M', K^{B}_{pri})$



Public Key Encryption - RSA

- Named over (Ron) Rivest – (Adi) Shamir – (Leonard) Adleman – inventors of the public key cryptosystem
- The encryption key is public and decryption key is kept secret (private key)
    - Anyone can encrypt the data
    - Only the intended receiver can decrypt the data

RSA Algorithm

- Four phases
  - Key generation
  - Key distribution
  - Encryption
  - Decryption

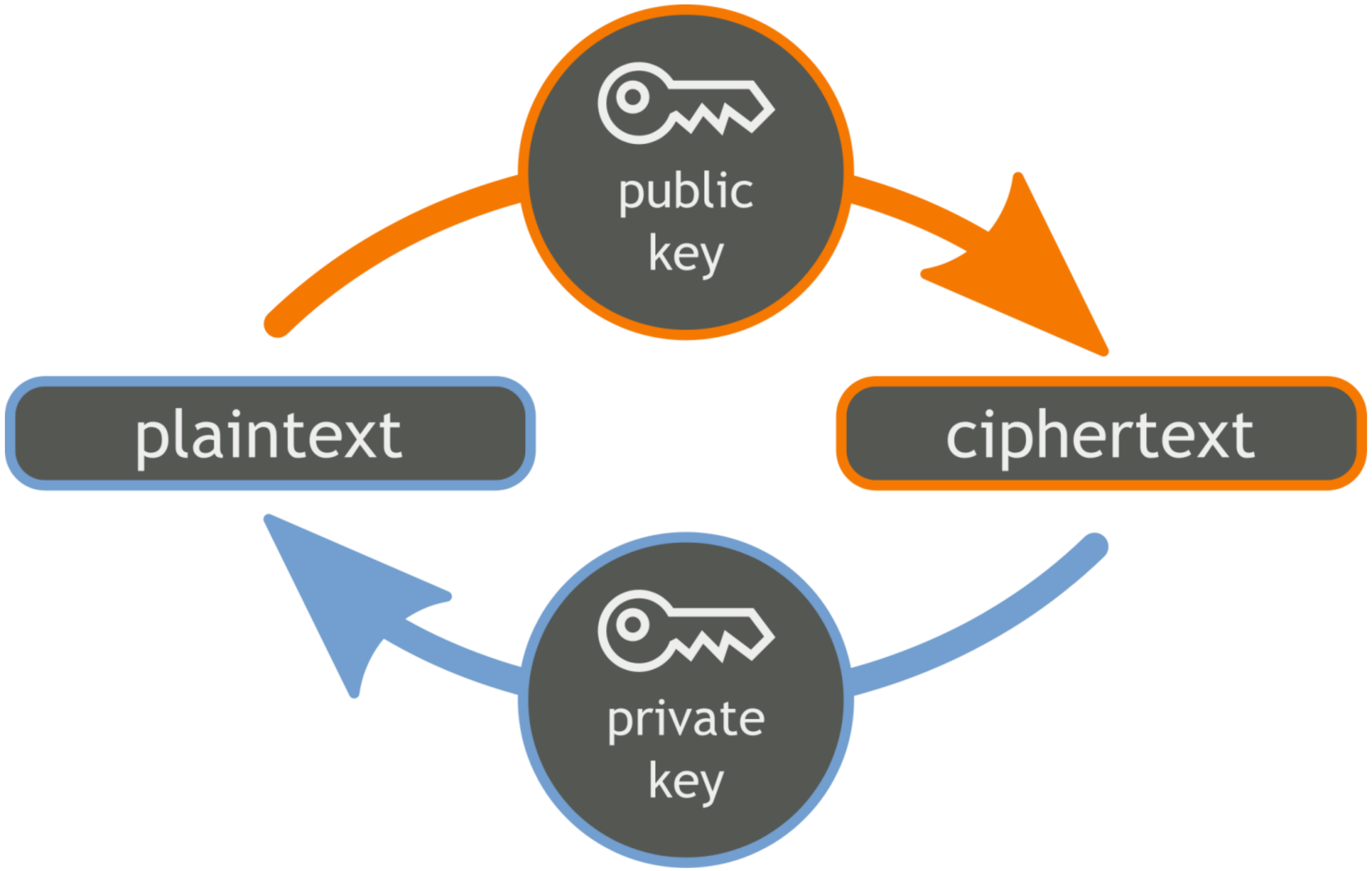


Image source:
https://commons.wikimedia.org/

Public and Private Keys in RSA

- It is feasible to find **three very large positive integers** e, d and n; such that *modular exponentiation* for integers m $(0 \le m < n)$:
$$(m^e)^d \equiv m \ (mod\ n)$$
- Even if you know e, n and m; it is extremely difficult to find d
- Note that
$$(m^e)^d \equiv m \ (mod\ n) = \left(m^d\right)^e \equiv m \ (mod\ n)$$
- (e, n) is used as the public key and (d, n) is used as the private key. m is the message that needs to be encrypted.

RSA Key Generation and Distribution

- Chose two distinct prime integer numbers p and q
  - p and q should be chosen at random to ensure tight security
- Compute $n = pq$; n is used as the modulus, the length of n is called the key length
- Compute $\phi(n) = (p-1)(q-1)$ – *Euler totient function*
- Choose an integer e such that $1 < e < \phi(n)$ and $\gcd\big(e, \phi(n)\big) = 1$; e and $\phi(n)$ are co-prime
- Determine $d = e^{-1}(mod\ \phi(n))$: d is the *modular multiplicative inverse* of $e(mod\ \phi(n))$ [Note $d.e = 1(mod\ \phi(n))$]

RSA Encryption and Decryption

- Let m be the integer representation of a message M.

- **Encryption with public key $(\boldsymbol{e}, \boldsymbol{n})$**

$$c \equiv m^e \ (mod\ n)$$

- **Decryption with private key $(\boldsymbol{d}, \boldsymbol{n})$**

$$m \equiv c^d \ (mod\ n) \equiv (m^e)^d (mod\ n)$$

RSA Encryption and Decryption - Example

Key Selection

- Select 2 prime numbers: p=17, q=11
- Calculate $n=pq=17\times11=187$
- Calculate $\phi(n)=(p-1)(q-1)=16\times10=160$
- Select e such that e is relatively prime to $\phi(n)=160$ and less than $\phi(n)$; Let e=7
- Determine d such that $d.e \equiv 1 \bmod 160$ and $d<160$; Can determine d = 23 since $23\times7 = 161 = 1\times160+1$

Encryption of Plaintext M = 88

- $C=88^7 \bmod 187$
- $= [(88^4 \bmod 187)\times(88^2 \bmod 187)\times(88^1 \bmod 187)] \bmod 187 = (88\times77\times132) \bmod 187 = 11$

Decryption of Ciphertext C = 11

- $M=11^{23} \bmod 187$
- $=[(11^1 \bmod 187)\times(11^2 \bmod 187)\times(11^4 \bmod 187)\times(11^8 \bmod 187)\times(11^8 \bmod 187)] \bmod 187$
- $=(11\times121\times55\times33\times33) \bmod 187 = (79720245) \bmod 187 = 88$

Courtesy: Cryptography and Network Security by William Stallings

RSA Encryption and Decryption - Demo

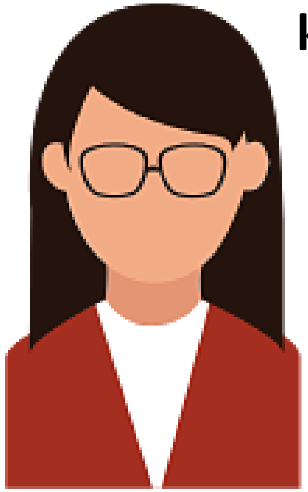
- https://www.devglan.com/online-tools/rsa-encryption-decryption

Digital Signature using Public Key Cryptography

- **Sign the message using the Private key**
  - Only Alice can know her private key
- **Verify the signature using the Public key**
  - Everyone has Alice's public key and they can verify the signature

Sign the message with her private key

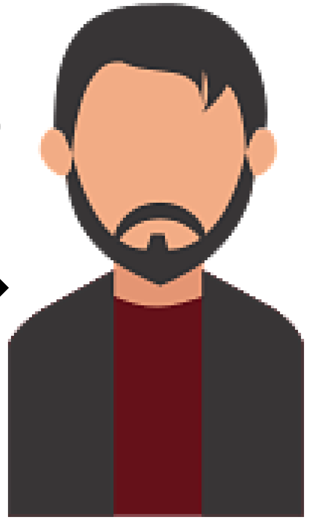
$M' = E(M, K^{A}_{pri})$



Verify the signature using Alice's public key

$M = E(M', K^{A}_{pub})$

M, M'



Reduce the Signature Size

- Use the message digest to sign, instead of the original message

Sign the message with her private key

$$S = E(H(M), K^A_{pri})$$

Verify the signature using Alice's public key

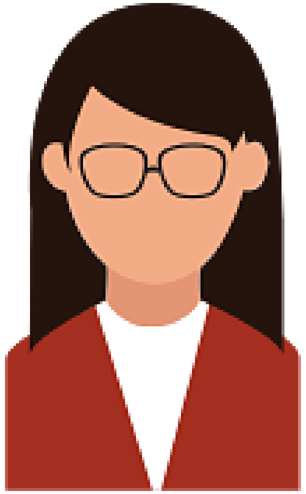
$$H(M) = E(S, K^A_{pub})$$

Digital Signature in Blockchain

- Used to validate the origin of a transaction
  - Prevent non-repudiation
    - **Alice cannot deny her own transactions**
    - **No one else can claim Alice's transaction as his/her own transaction**

- Bitcoin uses *Elliptic Curve Digital Signature Algorithm* ***(ECDSA)***
  - Based on elliptic curve cryptography
  - Supports good randomness in key generation

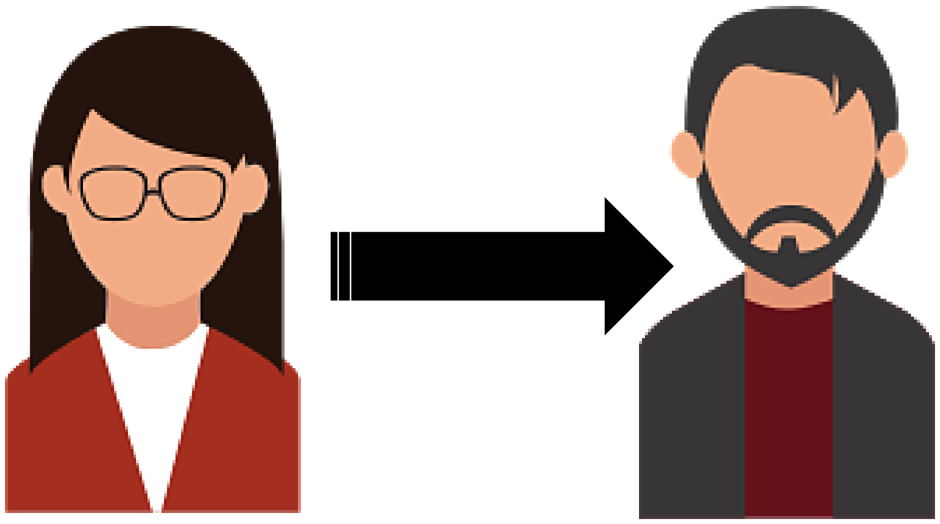
A Cryptocurrency using Hashchain and Digital Signatures

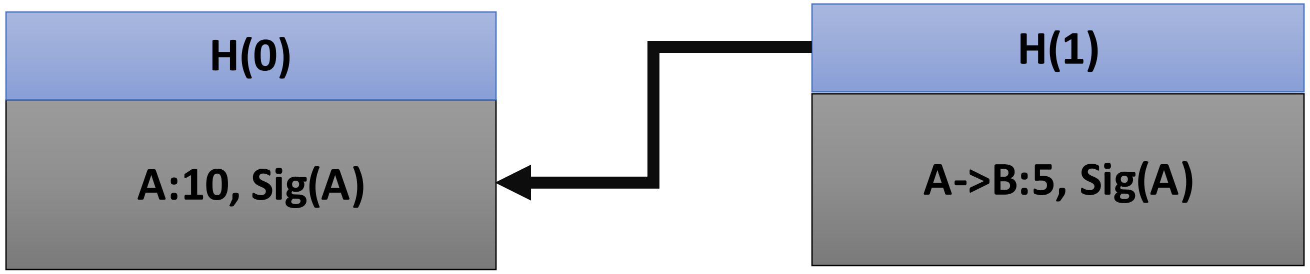


A:10, Sig(A)

- Alice generates 10 coins
- Sign the transaction A:10 using Alice's private key and put that in the blockchain

A Cryptocurrency using Hashchain and Digital Signatures





- Alice transfers 5 coins to Bob
- Sign the transaction A-B:5 using Alice's private key and put that in the blockchain

A Cryptocurrency using Hashchain and Digital Signatures

- Maintain the economy
  - Generate new coins with time
  - Delete old coins with time

- A central authority like bank can create and destroy coins based on economic policies

- **Crucial Question:** How can we distribute coin management (creation and destroy) thank you!