Final Year B. Tech Trimester-XII (AY 2020-2021)

Computer Science and Engineering

Disclaimer:

a. Information included in these slides came from multiple sources. We have tried our best to cite the sources. Please refer to the references to learn about the sources, when applicable.

b. The slides should be used only for preparing notes, academic purposes (e.g. in teaching a class), and should not be used for commercial purposes.

CSO43A: Blockchain Technology

Examination Scheme:

Class Continuous Assessment: 100 Marks Credit: 2

Course Objectives:

- To familiarize the students with functional/operational aspects of cryptocurrency Ecosystem.
- To explain the working of bitcoin and Blockchain Architecture.
- To explore the most prominent smart contract platform - Ethereum and Hyperledger

Course Outcomes:

After completion of this course students will be able to:

- understand the functional/operational aspects of cryptocurrency Ecosystem.
- describe the working of bitcoin and Blockchain Architecture.
- elaborate Ethereum and Hyperledger platforms.

Pre-requisites

- Distributed systems and Networking
- Cryptography
- Data Structures

Syllabus

Unit: I	**Fundamentals** **History:** Traditional financial arrangements, The trouble with credit cards online, From Credit to (Crypto) Cash. **Introduction to Cryptography & Cryptocurrencies:** Cryptographic Hash Functions, Hash Pointers and Data Structures, Digital Signatures, Public Keys as Identities **Bitcoin:** Introduction to Bitcoin, Bitcoin users - Full Client, Light Client, Web Client.	**8 Hrs**
Unit: II	**Bitcoin Mechanics** Centralization vs. Decentralization, Distributed consensus, Byzantine Generals Problem, Implicit Consensus, Bitcoin consensus algorithm, Stealing Bitcoins, Validation Algorithms: Proof of work, Proof of Stake, Proof of Authority, Proof of Activity, Proof of Burn, Proof of Capacity. Block Reward, Transaction fees, Bitcoin transactions, Bitcoin Scripts, Bitcoin blocks, Bitcoin network.	**8 Hrs**
Unit: III	**Blockchain Architecture** Introduction, Structure of a Block, Block Header, Block Identifiers - Block Header Hash and Block Height, The Genesis Block, Linking Blocks in the Blockchain, Types of blockchain, Merkle Trees and Simplified Payment Verification (SPV), Blockchain P2P architecture. Bitcoin Mining- The task of Bitcoin miners, Mining Hardware- CPU mining, GPU mining, FPGA mining, ASIC mining.	**7 Hrs**

Syllabus (Continue)

Unit: IV	**Ethereum & Hyperledger** Ethereum Virtual Machine, Smart contract, wallets for Ethereum, Ethereum Programming Language – Solidity, Mining in Ethereum, uses and benefits of Ethereum Hyperledger architecture, Consensus in Hyperledger, Hyperledger frameworks Bitcoin Security-Security principles, User Security Best Practices.	**7 Hrs**
Books:- **(Text)**	Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller and Steven Goldfeder, “Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction”, Princeton University Press (July 19, 2016)	
Books:- **(Reference)**	Andreas Antonopoulos, “Mastering Bitcoin: Unlocking Digital Cryptocurrencies”, O’Reilly, ISBN-13: 978-1449374044	
Supplementary Reading: 1. E-books 2. Web links 3. MOOCs		

Guidelines for CCA

Examination Scheme

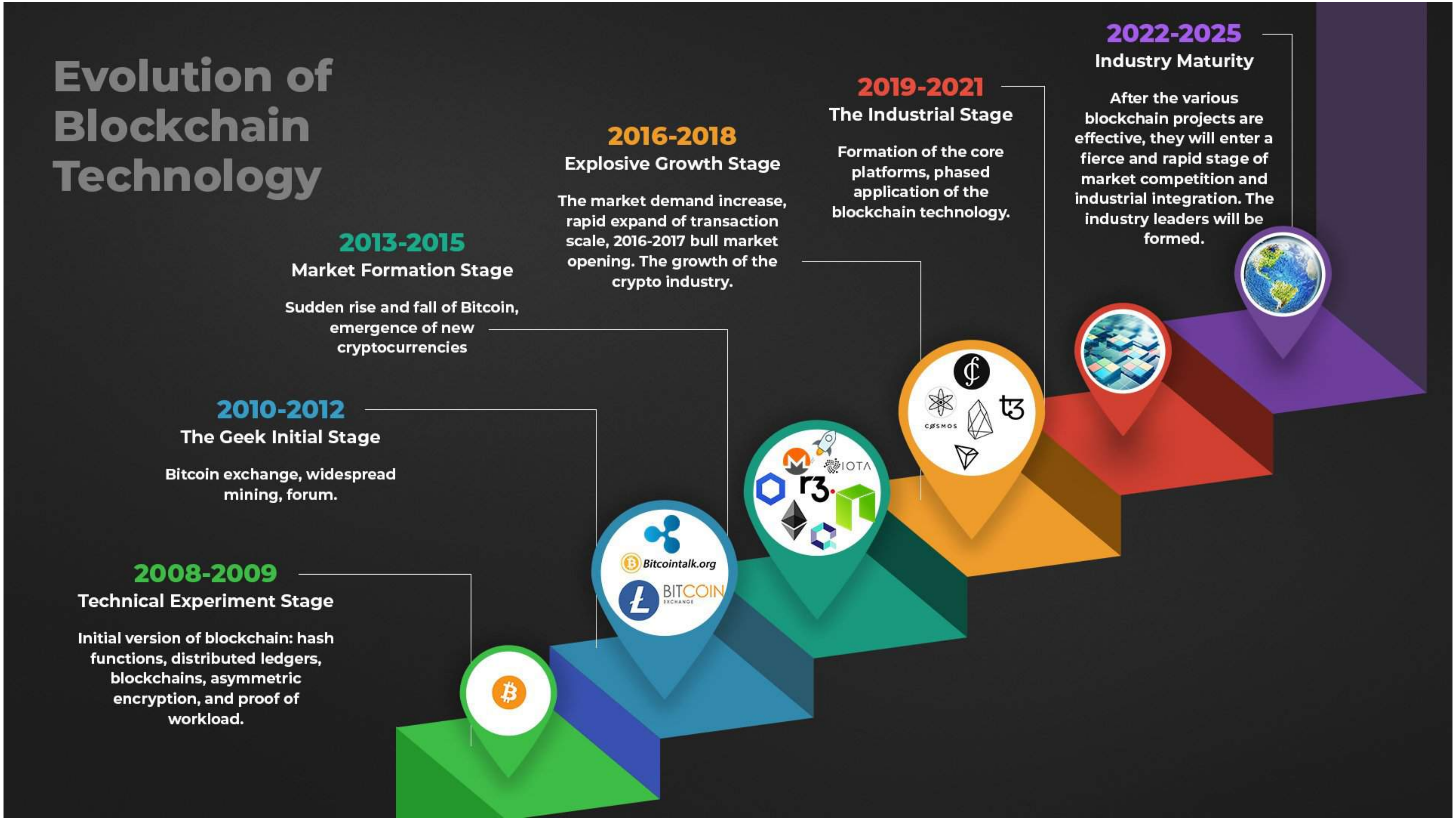
Sr. No.	Examination Scheme	Marks
1.	Class Continuous Assessment (CCA)	100

CCA Marks Distribution

Examination	Weightage	Marks
Theory Assignments	20 %	20
Mid-Term Theory Exam	15 %	15
Active Learning	25 %	25
Practical Assignments / Case Studies Evaluation	40 %	40
	Total	100

Unit-I : Fundamentals

- **History:** Traditional financial arrangements, The trouble with credit cards online, From Credit to (Crypto) Cash.
- **Introduction to Cryptography & Cryptocurrencies:**
- Cryptographic Hash Functions, Hash Pointers and Data Structures, Digital Signatures, Public Keys as Identities
- **Bitcoin:** Introduction to Bitcoin, Bitcoin users - Full Client, Light Client, Web Client.



Evolution of Blockchain Technology

2008-2009
Technical Experiment Stage

Initial version of blockchain: hash functions, distributed ledgers, blockchains, asymmetric encryption, and proof of workload.

2010-2012
The Geek Initial Stage

Bitcoin exchange, widespread mining, forum.

2013-2015
Market Formation Stage

Sudden rise and fall of Bitcoin, emergence of new cryptocurrencies

2016-2018
Explosive Growth Stage

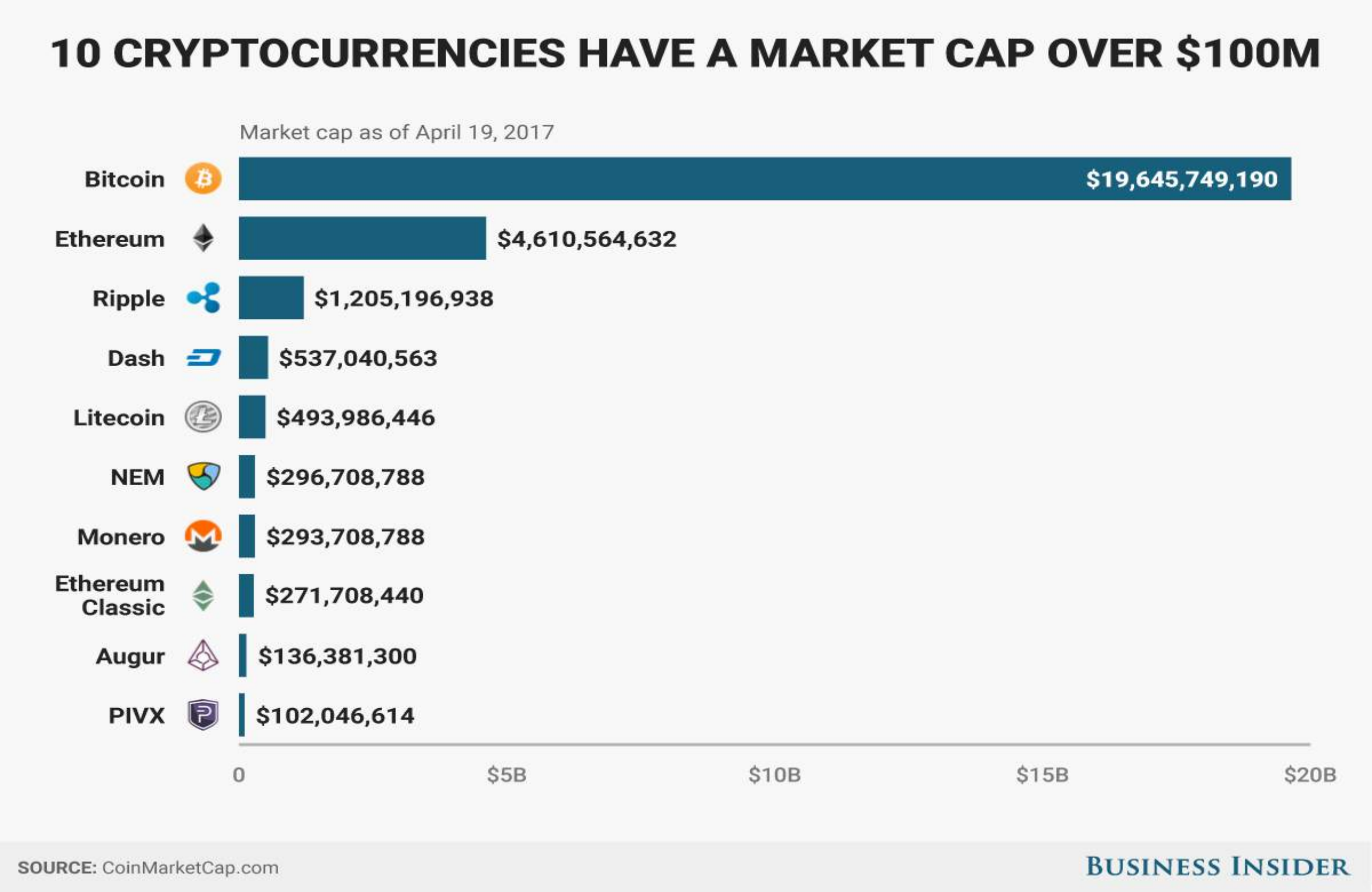
The market demand increase, rapid expand of transaction scale, 2016-2017 bull market opening. The growth of the crypto industry.

2019-2021
The Industrial Stage

Formation of the core platforms, phased application of the blockchain technology.

2022-2025
Industry Maturity

After the various blockchain projects are effective, they will enter a fierce and rapid stage of market competition and industrial integration. The industry leaders will be formed.



10 CRYPTOCURRENCIES HAVE A MARKET CAP OVER $100M

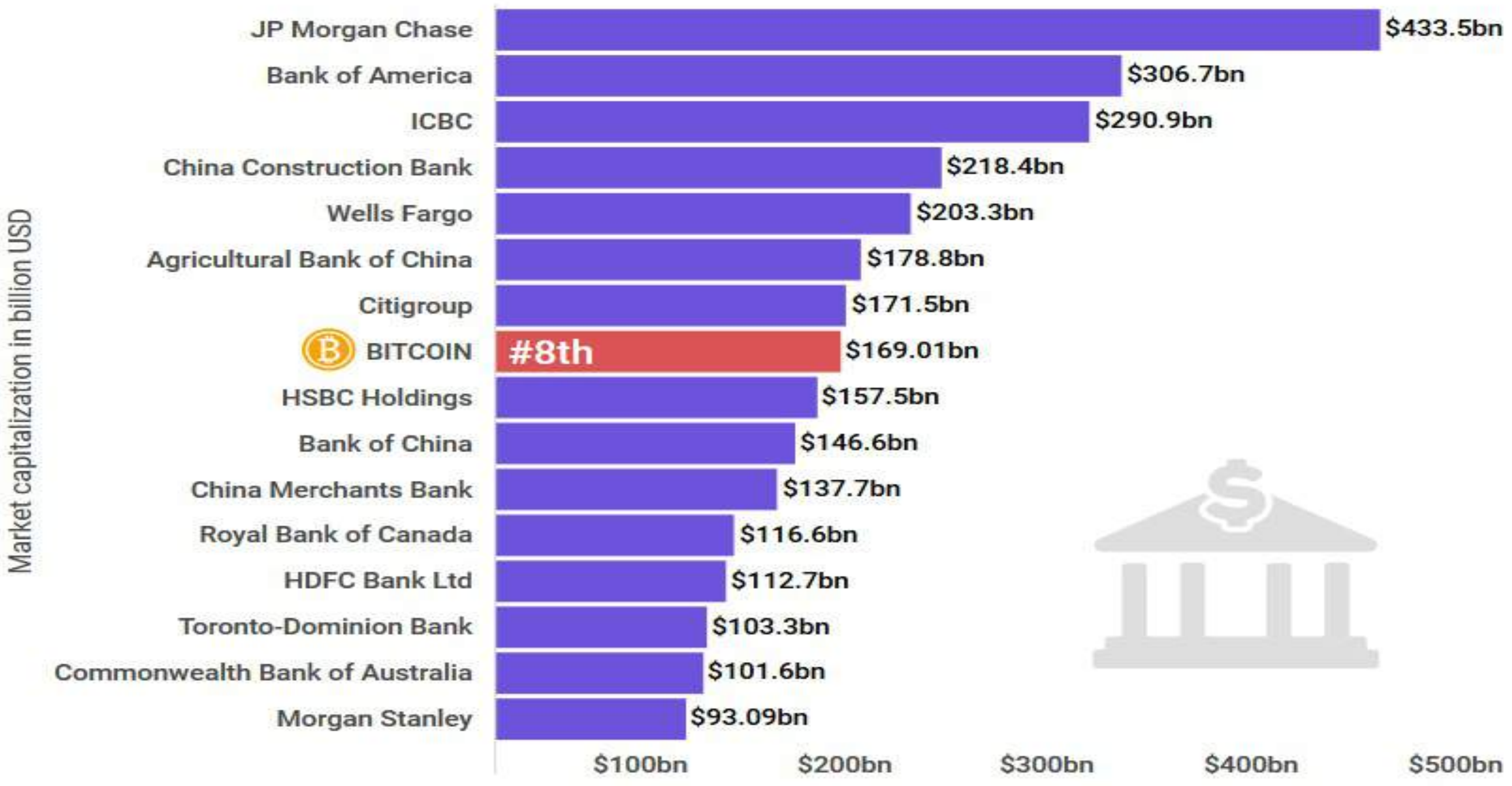
SOURCE: CoinMarketCap.com

15 leading banks worldwide vs. Bitcoin

by market capitalization, as of July 2020, in billion USD

Details: Worldwide; data released in July 2020 (survey as of January 21, 2020).

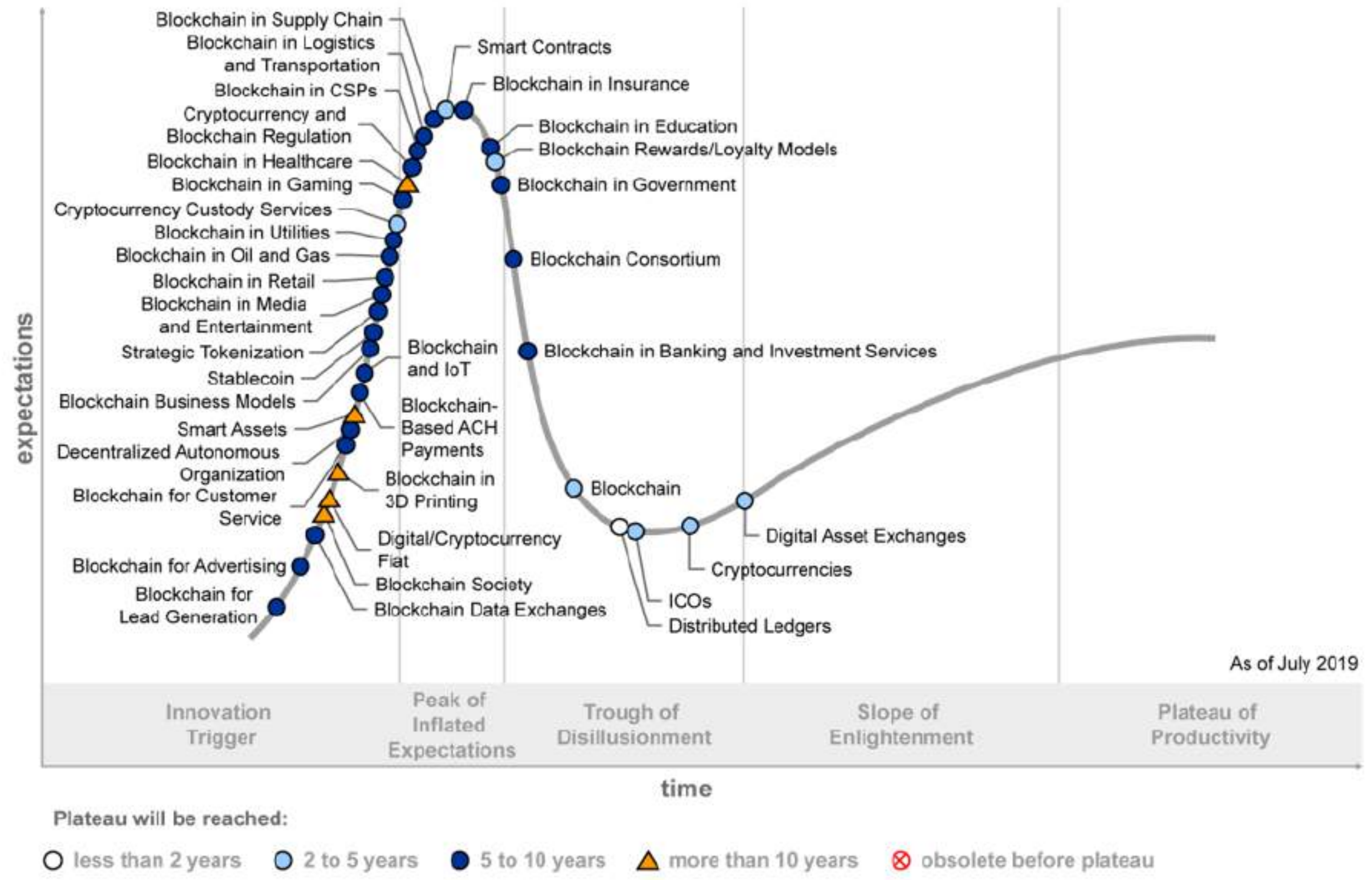
Data: Coinmarketcap.com, Statista



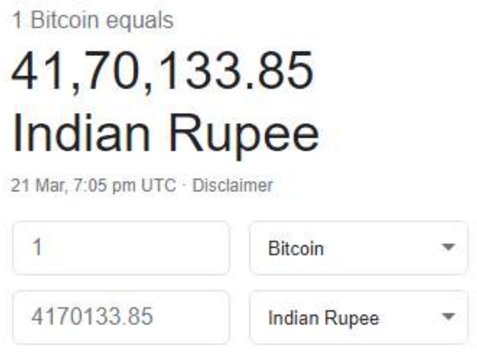
BUY SHARES

The 2019 Gartner, Inc. Hype Cycle for Blockchain Business shows that the business impact of blockchain will be transformational across most industries within five to 10 years.

Hype Cycle for Blockchain Business, 2019



Source: Gartner
ID: 390391



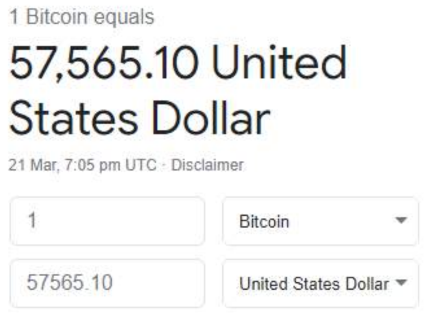
1 Bitcoin equals

41,70,133.85 Indian Rupee

21 Mar, 7:05 pm UTC · Disclaimer

1 | Bitcoin

4170133.85 | Indian Rupee



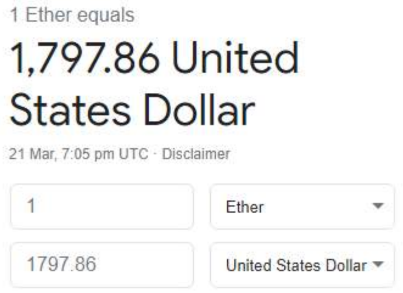
1 Bitcoin equals

57,565.10 United States Dollar

21 Mar, 7:05 pm UTC · Disclaimer

1 | Bitcoin

57565.10 | United States Dollar



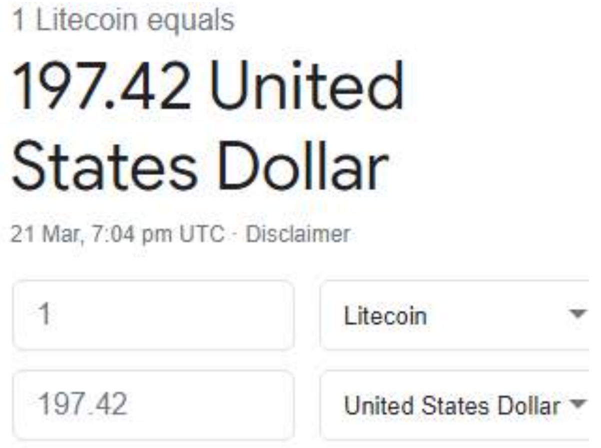
1 Ether equals

1,797.86 United States Dollar

21 Mar, 7:05 pm UTC · Disclaimer

1 | Ether

1797.86 | United States Dollar



1 Litecoin equals

197.42 United States Dollar

21 Mar, 7:04 pm UTC · Disclaimer

1 | Litecoin

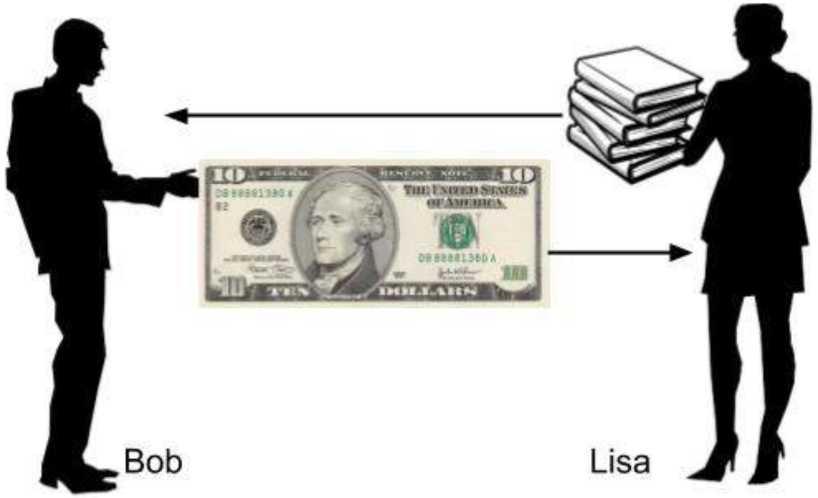
197.42 | United States Dollar

BitCoin value yesterday 21st march 20

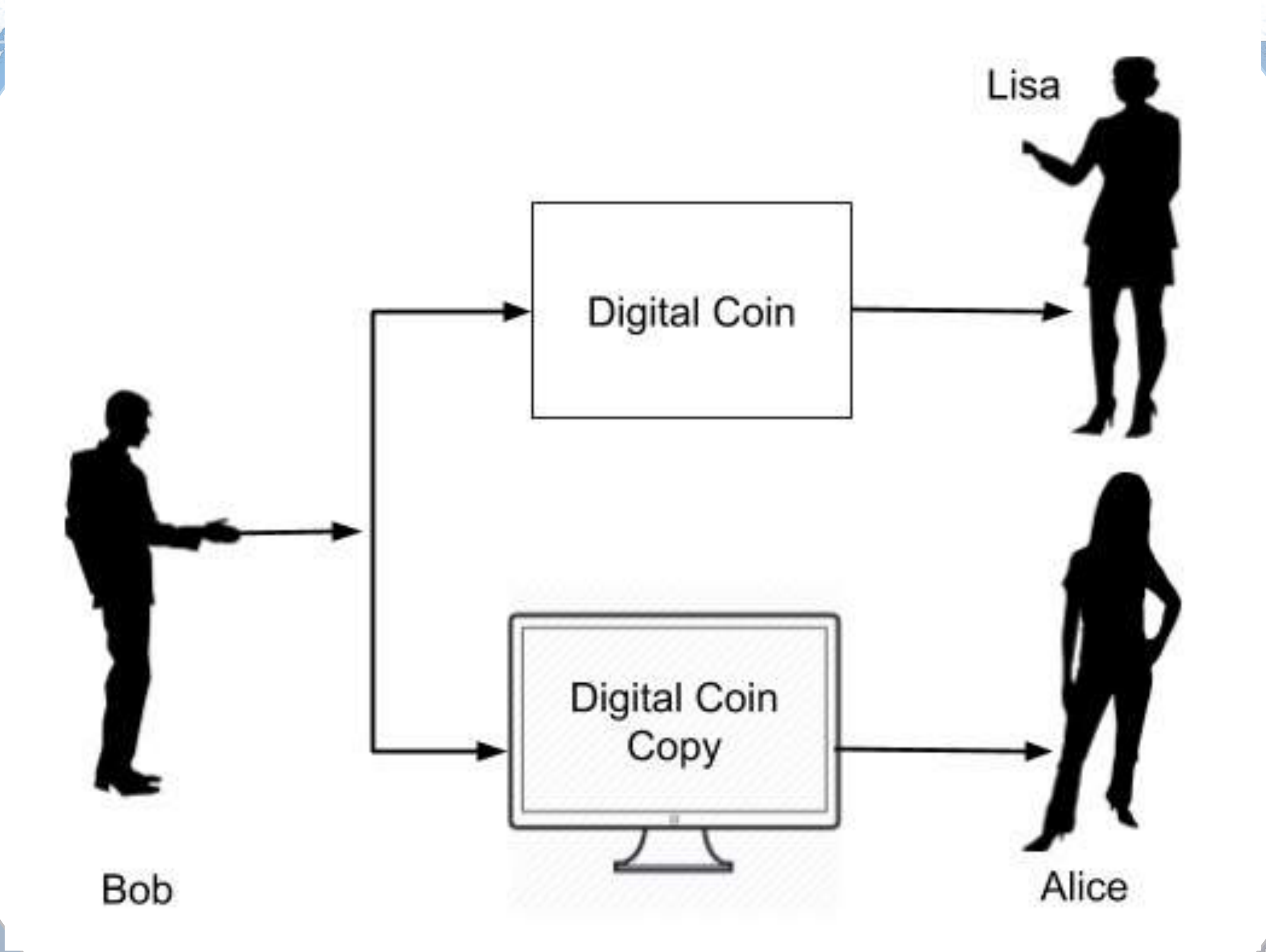
BlockChain Technology

- ❑ A blockchain is a growing list of records, called blocks, which are linked using cryptography.
- ❑ Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data.
- ❑ Blockchain has been in a lot of buzz these days. And that is mainly because it is backbone of the very famous cryptocurrency in the world - the Bitcoin. Many Governments and leading Banks have decided to bring many of their conventional transactions based on Blockchain concept.
- ❑ The applications and potential of this framework is huge and is considered to be changing the way transactions are made in various domains.

- ❑ Ref: Tutorials Point https://www.tutorialspoint.com/blockchain/index.htm



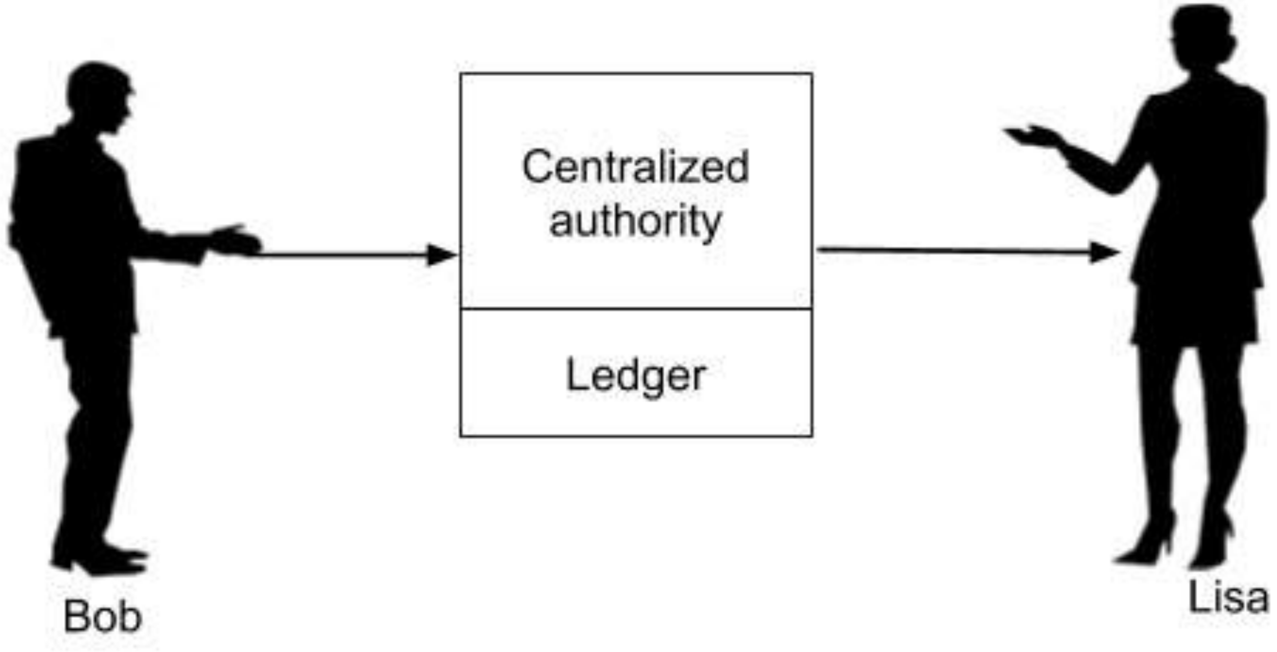
Bob and Lisa transaction ?


Lisa
Digital Coin
Digital Coin Copy
Bob
Alice

Double Spending

- As the format for money exchange is in the digital format, it is essentially a binary physical file stored somewhere on Bob's device.
- After Bob gives this file (digital money) to Lisa, he can also a give a copy of the file to Alice.
- Both now think that they have received the money without having any means of authenticating the digital coin and would thus deliver their respective goods to Bob.
- This is called **double-spending** where the sender spends the same money at more than one place for obtaining services or goods from multiple vendors.

To solve this problem of double-spending, one would employ a centralized authority to monitor all the transactions.



Centralized Authority?

- The introduction of centralized authority though it solves the double-spending problem, introduces another major issue - the cost of creating and maintaining the centralized authority itself.
- As the banks need money for their operations, they start cutting commissions on each currency transaction they do for their clients. This sometimes can become very expensive, especially in overseas transfer of money where multiple agents (banks) may be involved in the entire deal.
- All the above issues are solved by the introduction of digital currency, called Bitcoin.

BitCoin Technology

- The Bitcoin was introduced in this world by Satoshi Nakamoto through a research-style white paper entitled Bitcoin: A Peer-to-Peer Electronic Cash System in the year 2008.
- The Bitcoin not only solved the double-spending problem, but also offered many more advantages, One such advantage worth mentioning here is the **anonymity in the transactions**.
- Satoshi who created the system and did transact few coins on this system is totally anonymous to the entire world.
- Just imagine, in this world of social media, when the privacy of each individual is at stake, the world is not able to trace out so far who is Satoshi?
- In fact, we do not know whether Satoshi is an individual or a group of people. Googling it out also revealed the fact that the bitcoins Satoshi Nakamoto holds is worth about 34.9 billion - that money now remains unclaimed in the Bitcoin system.

BitCoin

- As you saw earlier, the bank maintains a ledger recording each transaction. This ledger is privately held and maintained by the bank.
- Satoshi proposed that let this ledger be public and maintained by the community.
- Since, this ledger is public, it has to be tamper-proof so that nobody can modify its entries.
- As each entry in the ledger is publicly visible, we will have to figure out how to maintain the anonymity - obviously you would not like everybody in the world to know that I paid you one million dollars.

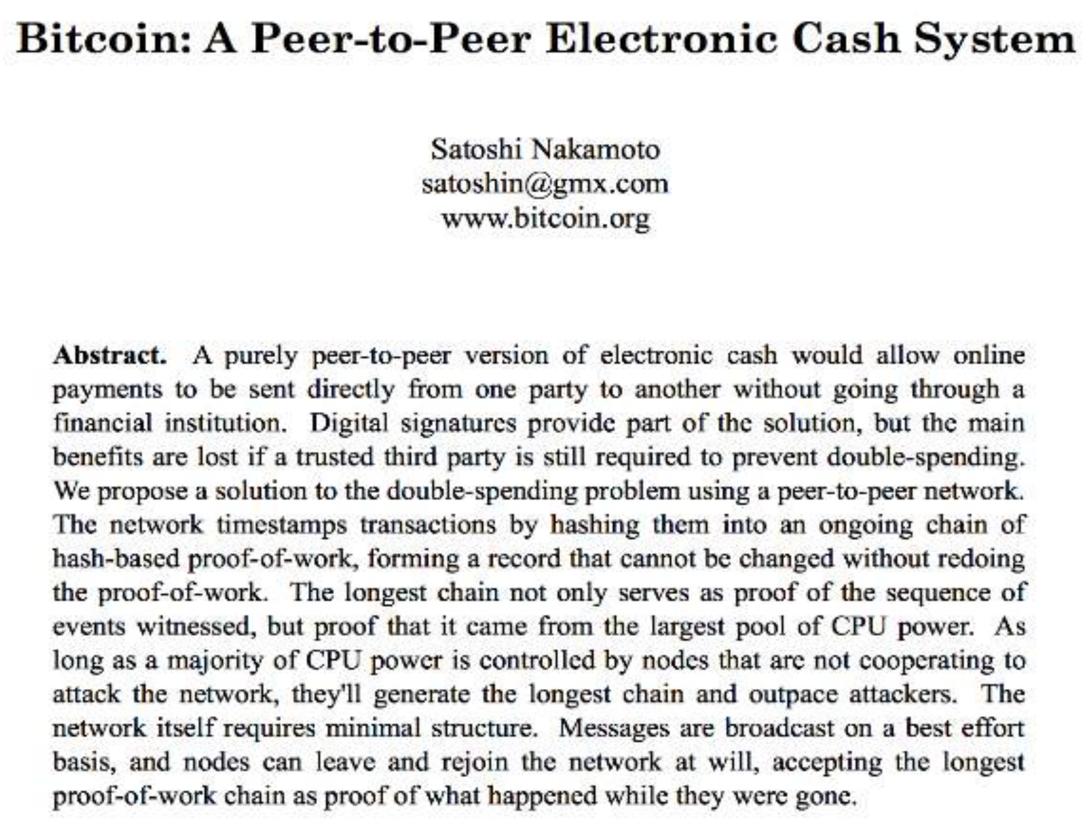
Blockchain

- Blockchain is a system comprised of..
  - Transactions
  - Immutable ledgers
  - Decentralized peers
  - Encryption processes
  - Consensus mechanisms
  - Optional Smart Contracts

- Let’s explore these concepts

Bitcoin Whitepaper: 10/31/2008

https://bitcoin.org/bitcoin.pdf

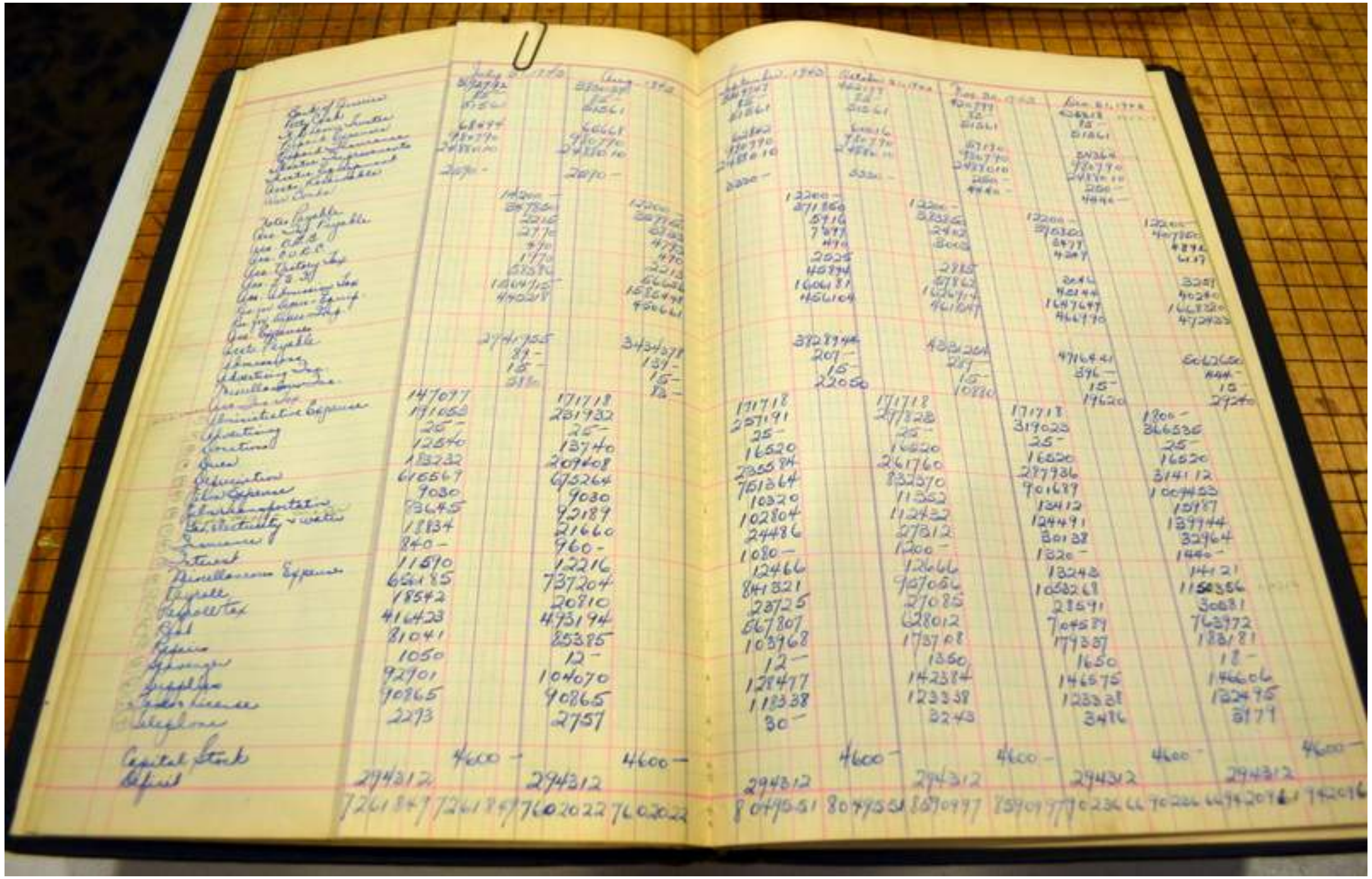


Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

Ledger



Blockchains Unchained Guide

The basic and inter-related goals of blockchain are to:

- Reduce or **eliminate the need for a central authority** (e.g., banks, government)
- Eliminate **central points of failure**
- **Enable trust** among people who don’t know each other to directly conduct transactions

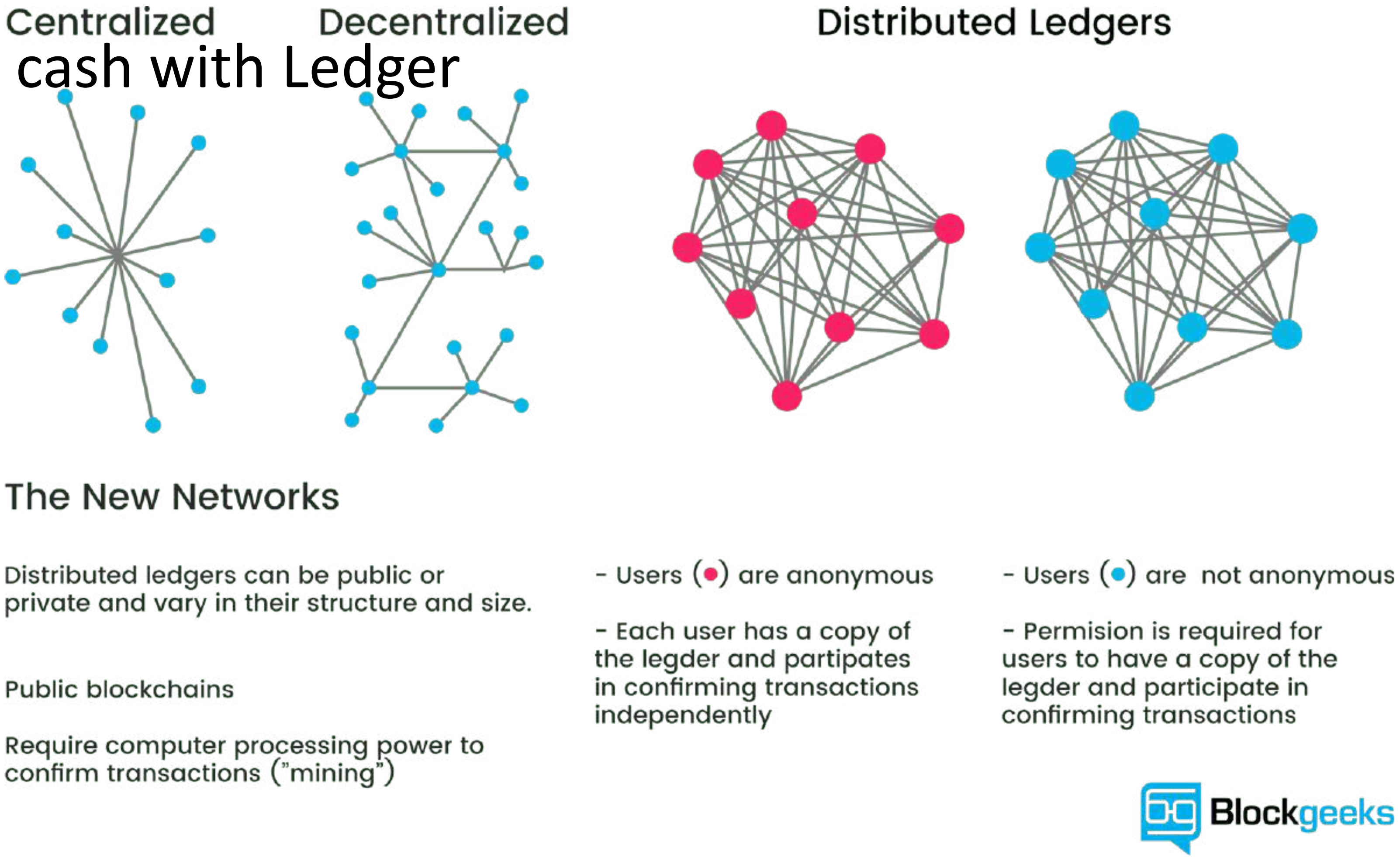
To achieve this, instead of an authority running a central database, every user can have a copy of the full database and can see every transaction that has ever taken place. This is a **distributed ledger.**

Key term: Distributed Ledger
A List of transactions that are spread across many users (not central)

Key term: Node
Another word for a user on a blockchain network running blockchain software and holding a copy of the ledger

Decentralization

- Replace cash with Ledger



The New Networks

Distributed ledgers can be public or private and vary in their structure and size.

Public blockchains

Require computer processing power to confirm transactions (”mining”)

- Users (●) are anonymous

- Each user has a copy of the legder and partipates in confirming transactions independently

- Users (●) are not anonymous

- Permision is required for users to have a copy of the legder and participate in confirming transactions

Transactions

- As with enterprise transactions today, Blockchain is a historical archive of decisions and actions taken
- Proof of history, provides provenance

• **Notable transaction use cases**
• Land registration – Replacing requirements for research of Deeds (Sweden Land Registration)
• Personal Identification – Replacement of Birth/Death certificates, Driver’s Licenses, Social Security Cards (Estonia)
• Transportation – Bills of Lading, tracking, Certificates of Origin, International Forms (Maersk/IBM)
• Banking – Document storage, increased back office efficiencies (UBS, Russia’s Sberbank)
• Manufacturing – Cradle to grave documentation for any assembly or sub assembly
• Food distribution – Providing location, lot, harvest date Supermarkets can pin point problematic food (Walmart)
• Audits – Due to the decentralized and immutable nature of Blockchain, audits will fundamentally change.

Validating Transactions

Ok, so everyone can have a copy of the ledger and see all of the transactions. **But how can they be sure these transactions are valid?**

- To submit a transaction, a user must digitally sign it using a “cryptographic **key**”.
- When a user submits a transaction, it propagates throughout the network in seconds or minutes. **Every node checks** to ensure the transaction is feasible and was properly signed. If yes, they continue to propagate, if not, they discard the transaction.
  - If **more than 50%** agree that is it valid, it is considered a valid transaction.
  - *But… these are not part of the blockchain yet.*

Key term: Cryptographic Key

Old decryption technology. All users have a “public key” and a ***linked*** “private key”. The public key is widely known. The private key must never be shared. A user signs their transactions with the private key, and then **all users can verify** that it was truly the right person by checking it against their public key.

Mining Transactions

After transactions are validated, they wait in a queue until they are “mined” by a “mining node”.

- A mining node will **validate a set of transactions** from the queue and group them into a “block”.
- The mining node then **publishes the block** to the chain and begins to broadcast the new block across the network.
- The mining node discards any invalid transactions

Although this is complex, it is all done automatically with blockchain software.

Key term: Mining
The act of again validating a group of transactions from the queue and publishing them as new block to a chain. The agreed-upon “consensus model” (a very complicated concept to be explained shortly) for the blockchain determines who can do this. Sometimes it’s competitive. Sometimes it’s based on user permissions.

"Immutability"

As with existing databases, Blockchain retains data via transactions

The difference is that once written to the chain, the blocks can be changed, but it is extremely difficult to do so. Requiring rework on all subsequent blocks and consensus of each.

The transaction is, immutable, or indelible

In DBA terms, Blockchains are Write and Read only

Like a ledger written in ink, an error would be be resolved with another entry

In addition to they key principles of:

1. Distributed: everyone holding a copy of the ledger and these copies are automatically synchronised
2. Shared: all transactions being transparent to everyone

There is a key third principle: ***Immutability***

Key term: Immutability

Once data has been written to a Blockchain, no one, not even a system administrator, can change it. This helps to ensure trustworthiness.

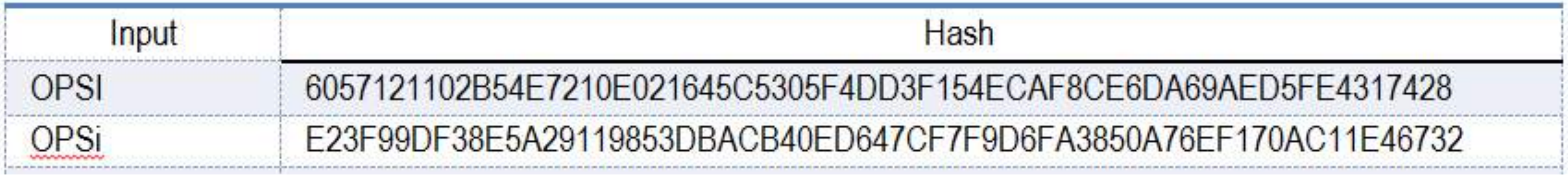
Immutability is a result of how blockchain technology is designed.

How Blockchains are Designed

A “blockchain” is literally a chain of blocks

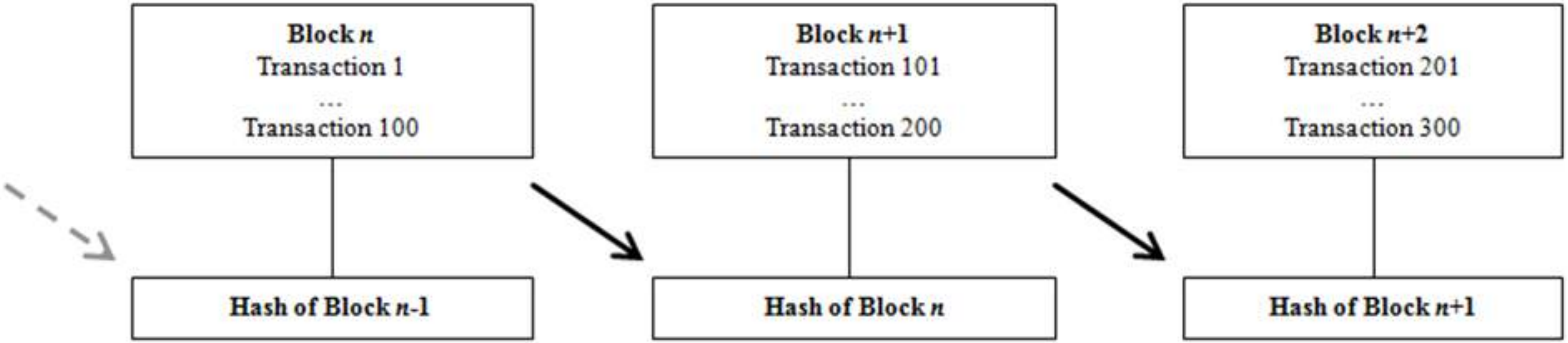
- As discussed, each block contains a group of validated transactions.
- These blocks are added one-by-one to the chain in a **linear, chronologica**l manner.
- *Critically*, every block is ***inextricably linked*** to the previous blockchain using a process called “**hashing**”.
- Each block’s contents are “hashed”, and each block gets a unique “hash code”, which ***links blocks together***

Key term: Hashing
An encryption function that converts any input (text, image, etc.) into a fixed-length code. The same input will always result in the same code. However, even the most minor change will entirely change the code.



Input	Hash
OPSI	6057121102B54E7210E021645C5305F4DD3F154ECAF8CE6DA69AED5FE4317428
OPSi	E23F99DF38E5A29119853DBACB40ED647CF7F9D6FA3850A76EF170AC11E46732

Linking of Blocks



If anyone tried to alter even the smallest piece of a transaction, it would **completely change** the hash code for the transaction and the block. This would cause **cascading effects** for all of the connected blocks.

This would **immediately be noticed** by the nodes and discarded.

Difference Between Blockchain and Bitcoin

One of the biggest challenges for blockchain is that it is **conflated with Bitcoin**.

- Blockchain was born with Bitcoin and remains the largest blockchain *platform*
- However, hundreds or **thousands of other platforms** now exist
- The underlying technology has uses and implications that go **far beyond Bitcoin** or cryptocurrencies in general

Some platforms have developed innovative new features. Most notably, “**smart contracts**”

Key term: Smart Contracts

Self-executing contracts where the terms are written directly in software code on the blockchain.

Each smart contract is an automated “if/then” scenario that executed when a specific trigger occurs.

Public versus Private Blockchains

Blockchain ledgers can be public (“permissionless”), or private (“permissioned”). The distinction between the two is much like the *internet* versus an *intranet*.

- **Permissionless ledgers** (e.g., Bitcoin) allow anyone to make transactions and to hold identical copies of the full ledger.

- **Permissioned ledgers** limit contributions to a limited set of users who have been given permission. Access to view records can be restricted or public, depending on the settings of the ledger. In fact, many different aspects of the blockchain can be customized to meet different needs. **These are likely to be the most useful for public sector use.**

The types of “**consensus models**”, which are the rules that determine who has the right to publish the next block, vary depending on type of blockchain at hand.

Consensus Models

There are a growing number of consensus models that determine **which node has the right to publish the next block**. The two below are examples.

Proof of Work – Always for permissionless ledgers (e.g., Bitcoin)

- Since no trust exists, in order to keep one or a few users from taking control, a complicated process exists to help even the playing field.
- Each user competes to solve a puzzle that is intentionally resource intensive (e.g., processing power, and by extension, electricity) to solve.
- The winner publishes the next block and earns financial reward.

Proof of Authority – Usually for permissioned ledgers

- User identities must be known and verified.
- Ability to publish new blocks is dictated by user permissions (not unlike a traditional database).
- No issues related to processing power or electricity.

Notable transaction use cases

Land registration – Replacing requirements for research of Deeds

Personal Identification – Replacement of Birth/Death certificates, Driver's Licenses, Social Security Cards

Transportation – Bills of Lading, tracking, Certificates of Origin, International Forms (Maersk/IBM)

Banking – Document storage, increased back office efficiencies

Manufacturing – Cradle to grave documentation for any assembly or sub assembly

Food distribution – Providing location, lot, harvest date Supermarkets can pin point problematic food

Audits – Due to the decentralized and immutable nature of Blockchain, audits will fundamentally change

Introduction to Cryptography & Cryptocurrencies

- All currencies need some way to control supply and enforce various security properties to prevent cheating
- In fiat currencies, organizations like central banks control the money supply and add anticounterfeiting features to physical currency.
- Ultimately, law enforcement is necessary for stopping people from breaking the rules of the system.

Introduction to Cryptography & Cryptocurrencies

- Cryptocurrencies too must have security measures that prevent people from tampering with the state of the system
- If Alice convinces Bob that she paid him a digital coin, for example, she should not be able to convince Carol that she paid her that same coin.
- But unlike fiat currencies, the security rules of cryptocurrencies need to be enforced purely technologically and without relying on a central authority.

Introduction to Cryptography & Cryptocurrencies

- As the word suggests, cryptocurrencies make heavy use of cryptography.
- Cryptography provides a mechanism for securely encoding the rules of a cryptocurrency system in the system itself.
- We can use it to prevent tampering and equivocation (making mutually inconsistent statements to different people) , as well as to encode, in a mathematical protocol, the rules for creation of new units of the currency.

Introduction to Cryptography & Cryptocurrencies

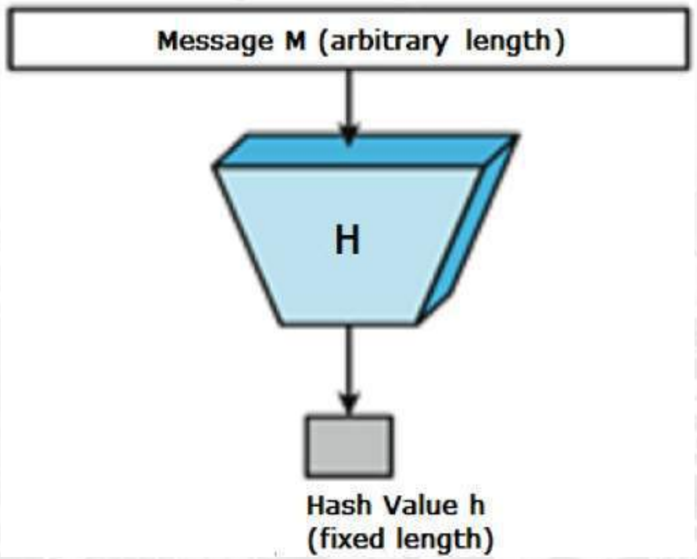
- Cryptography is a deep academic research field using many advanced mathematical techniques that are notoriously subtle and complicated
- We specifically study cryptographic hashes and digital signatures, two primitives that prove to be useful for building cryptocurrencies

Cryptographic Hash Functions

- The first cryptographic primitive that we need to understand is a *cryptographic hash function*
- A *hash function* is a mathematical function with the following three properties:
- Its input can be any string of any size
- It produces a fixed-sized output
- we will assume a 256-bit output size
- It is efficiently computable.

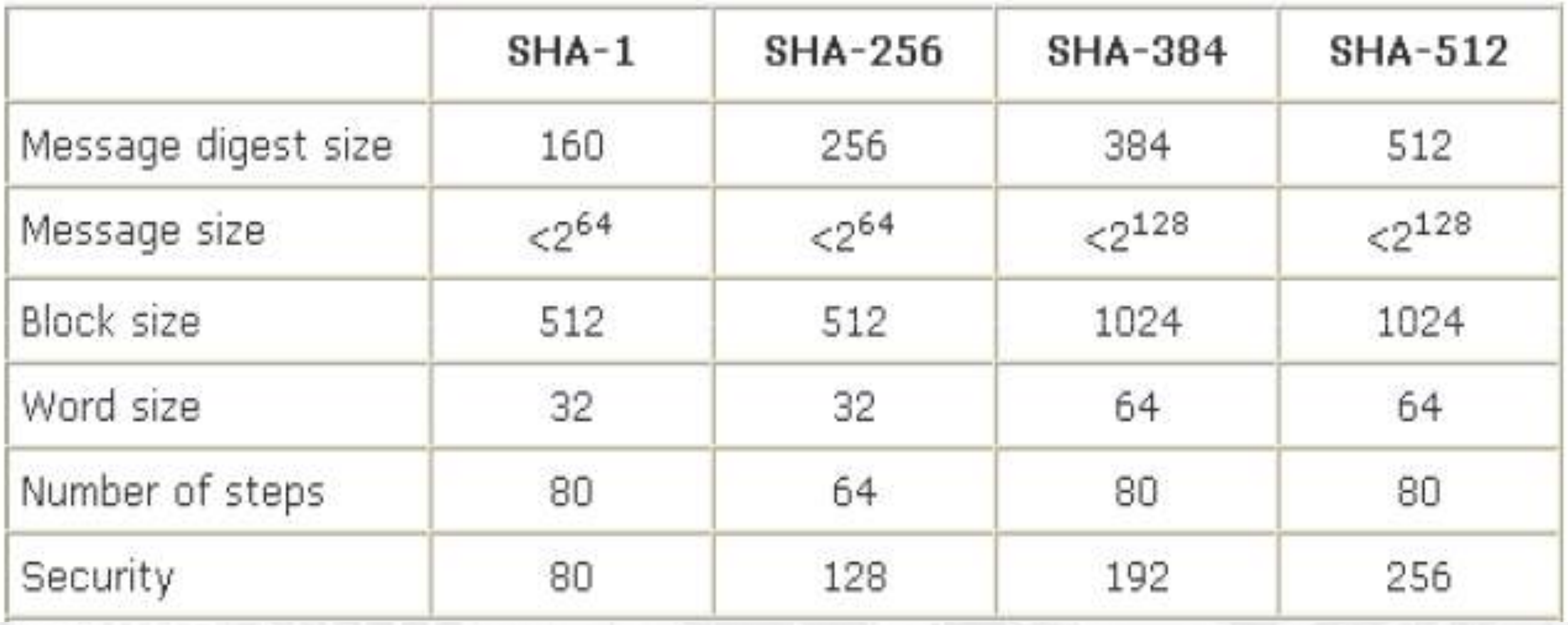
Message Digest: MD 5 and SHA -1

- The digest is sometimes called the "hash" or "fingerprint" of the input.
- Hash value is used to check the integrity of the message
- MD5 processes a variable-length message into a fixed-length output of 128 bits.



Secure Hash Algorithm (SHA)

- ❖ SHA is a modified version of MD5. (Published in 1993)
- ❖ SHA works any input message less than 2^{64} bits and produces a hash value of 160 bits.
- ❖ SHA is designed to be computationally infeasible to:
  - Obtain the original message
  - Find two message producing the same MD.



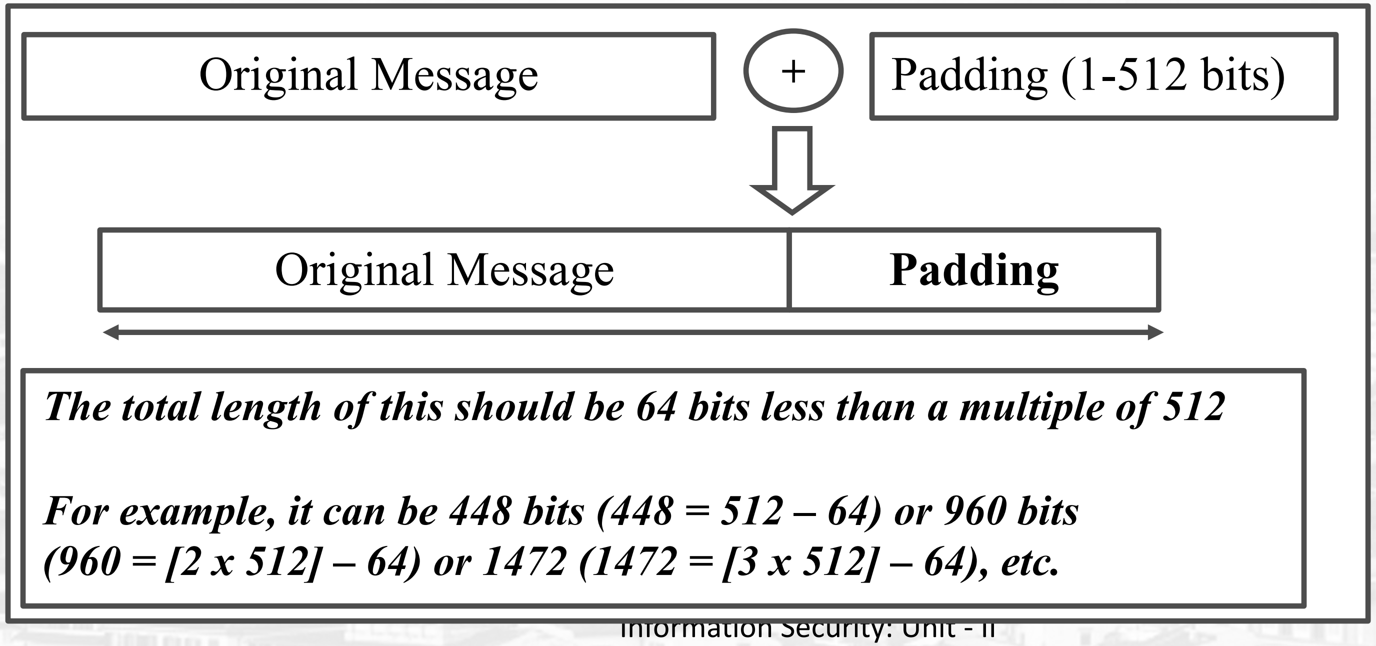
	SHA-1	SHA-256	SHA-384	SHA-512
Message digest size	160	256	384	512
Message size	$<2^{64}$	$<2^{64}$	$<2^{128}$	$<2^{128}$
Block size	512	512	1024	1024
Word size	32	32	64	64
Number of steps	80	64	80	80
Security	80	128	192	256

Algorithm:

- Step -1: Padding
- Step - 2: Append length
- Step - 3: Divide the input into 512-bit blocks.
- Step - 4: Initialize chaining variables (4 variables)
- Step - 5: Process blocks

Step 1: Padding

- To make the length of the original message equal to a value, which is 64 bits less than an exact multiple of 512
- **Note:** Padding is always added, even if the original message is already 64 bits less than a multiple of 512

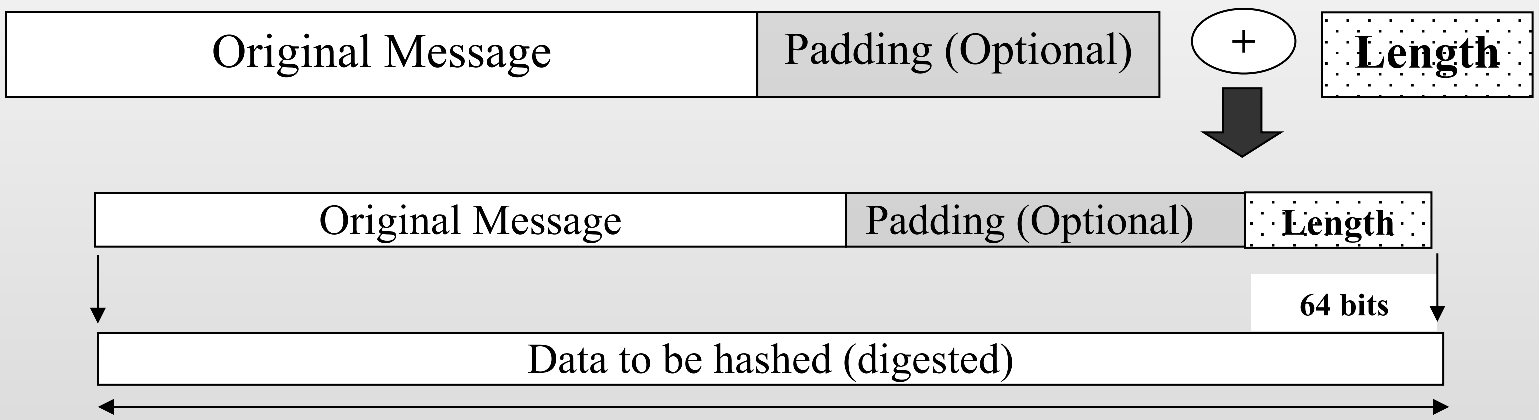


The total length of this should be 64 bits less than a multiple of 512

For example, it can be 448 bits (448 = 512 – 64) or 960 bits (960 = [2 x 512] – 64) or 1472 (1472 = [3 x 512] – 64), etc.

Step 2: Append length

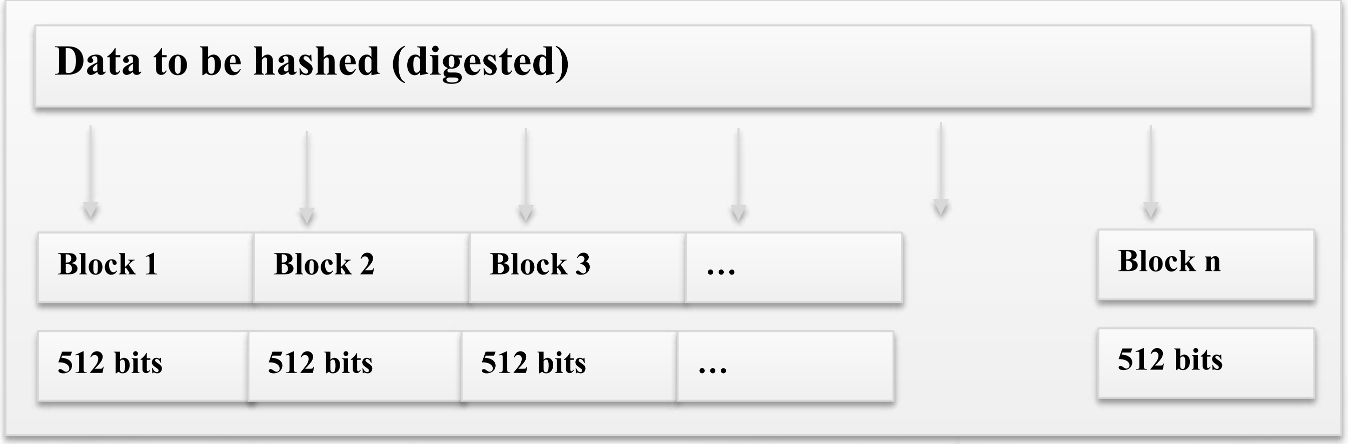
- Add a 64-bit binary-string which is the representation of the message’s length
- If the original length is greater than 2^{64}, then only **the low-order 64** bits of the length are used.
- Thus, field contains the length of the original message, modulo 2^{64}.



The length of the original message (excluding the padding) is calculated, and is appended to the end of the message block + padding. This length is expressed in 64 bits. If the length of the message exceeds 64 bits (i.e. it is greater than 2^{64}, then only the last 64 bits of the length are used, i.e. a modulus 64 of the length is taken. After the 64-bit length of the message is appended, this becomes the final message (i.e. the message to be hashed).

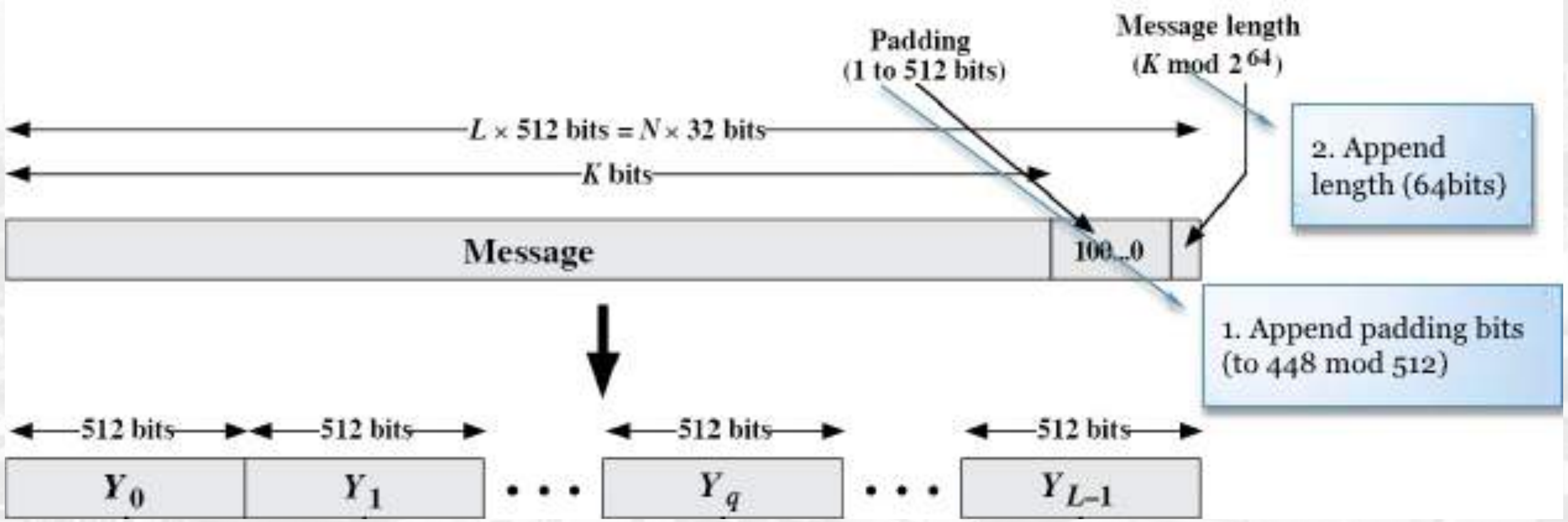
The length of the message is now a multiple of 512 bits.

Step 3: Divide the input into 512-bit blocks



Data to be hashed (digested)

Block 1	Block 2	Block 3	…	Block n
512 bits	512 bits	512 bits	…	512 bits

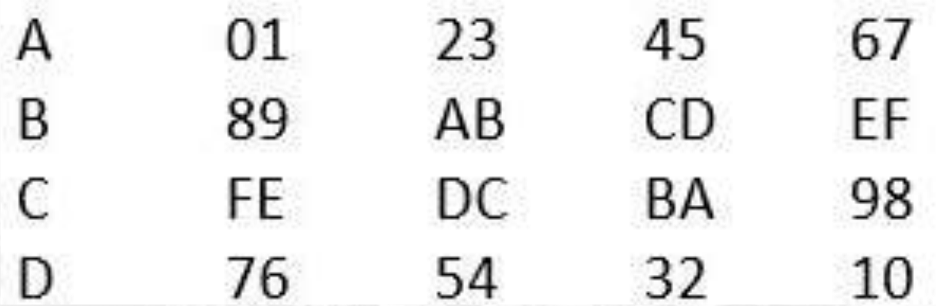


2. Append length (64bits)

1. Append padding bits (to 448 mod 512)

Step 4: Initialize MD buffer

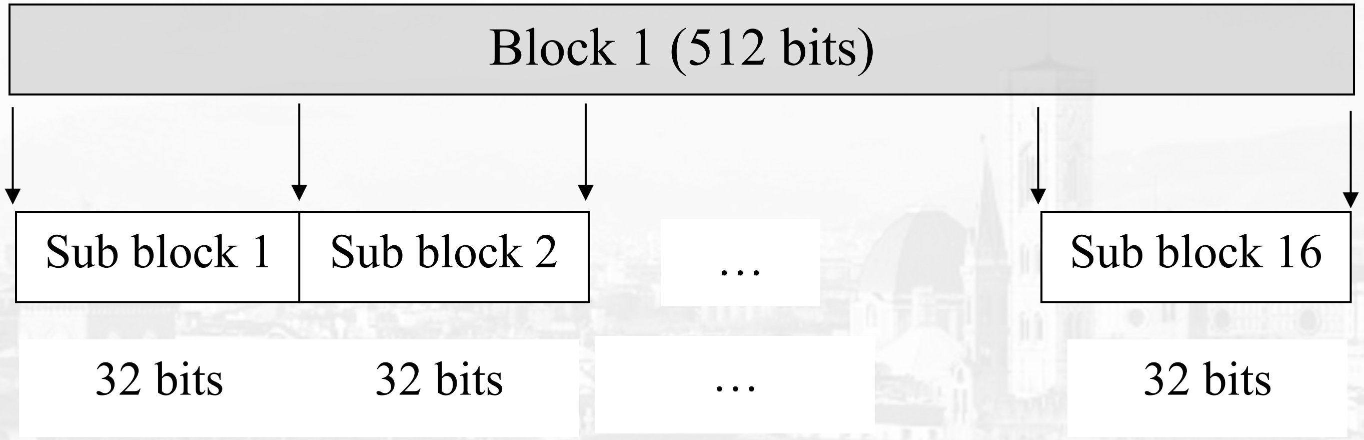
- ❖ A four-word buffer (A, B, C, D) is used to compute the message digest.
- ❖ Here each of A, B, C, D is a 32 bit register.



A	01	23	45	67
B	89	AB	CD	EF
C	FE	DC	BA	98
D	76	54	32	10

Step 5: Process Blocks (or message)

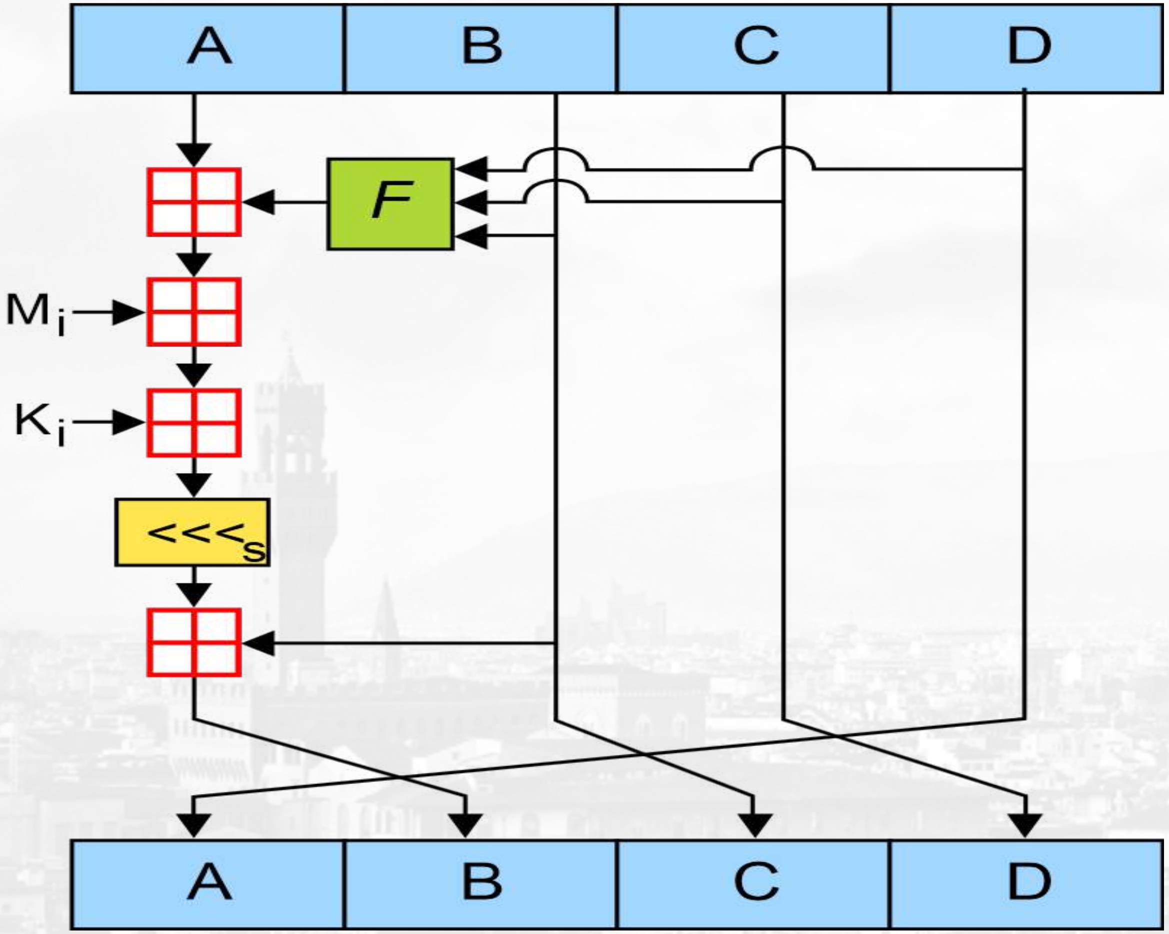
- ❖ Divide the 512- bit block into 16 sub-blocks.
- ❖ Each sub-block undergoes 4 rounds of operations. Total 16 operations are performed.



Block 1 (512 bits)

Sub block 1	Sub block 2	…	Sub block 16
32 bits	32 bits	…	32 bits

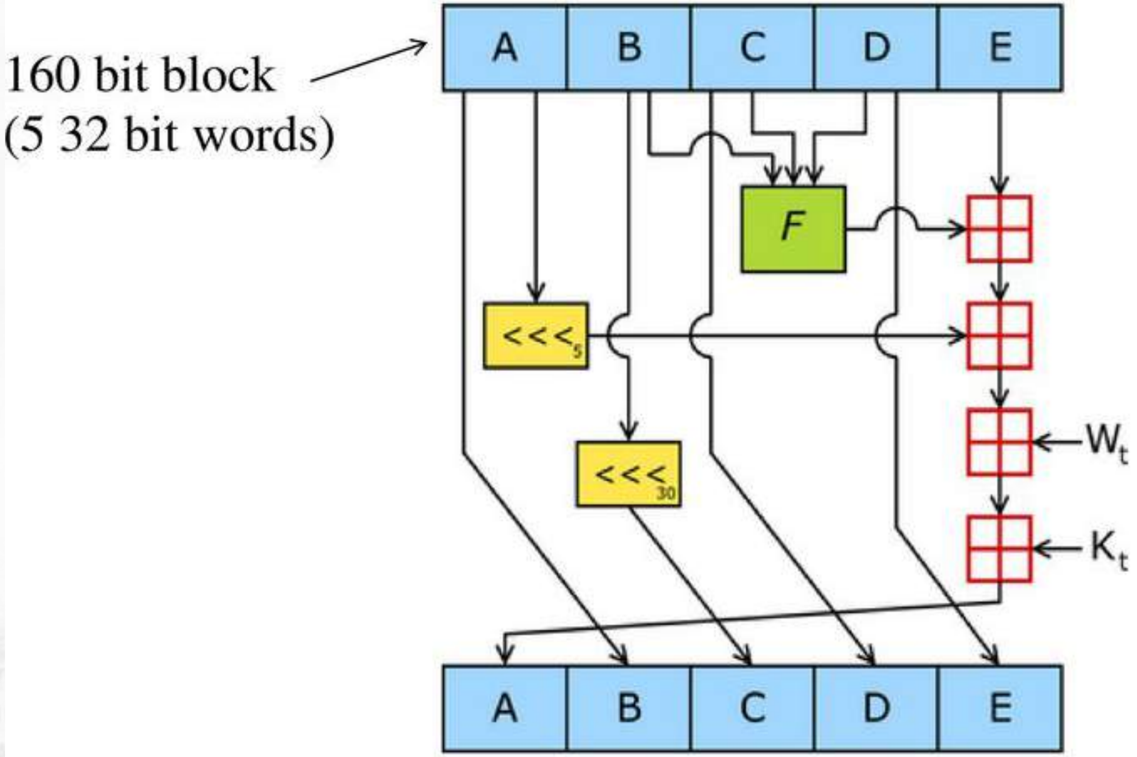
$$A = B + ((A + \text{Process } F(B, C, D) + M_i + K_i) <<< s)$$



- ❖ There are four possible functions F; a different one is used in each round:

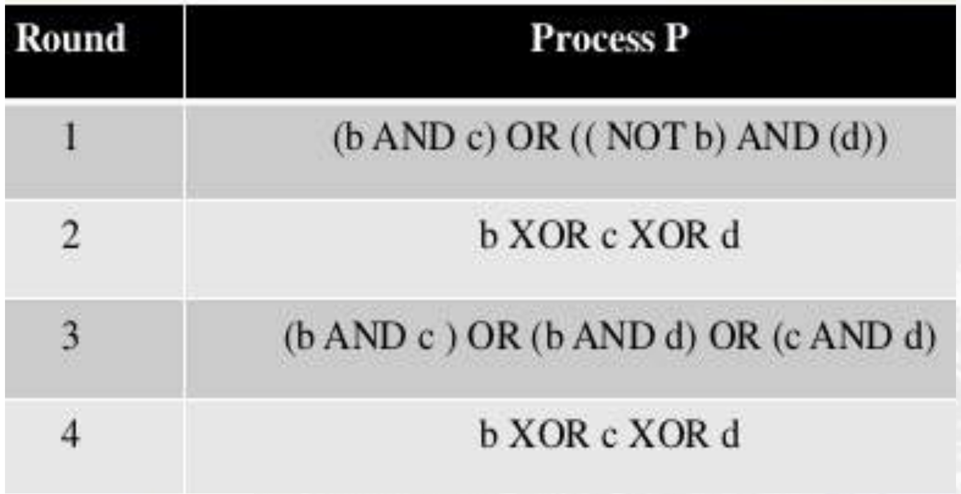
Round	Process F
1	(B AND C) OR ((NOT B) AND (D))
2	(B AND D) OR (C AND (NOT D))
3	B XOR C XOR D
4	C XOR (B OR (NOT D))

Process each block with A, B, C, D, E



Last round:
A-E is the digest

```
temp = (A <<<5) + F + E + Kt + wt
E = D
D = C
C = B <<<30
B = A
A = temp
```



Round	Process P
1	(b AND c) OR ((NOT b) AND (d))
2	b XOR c XOR d
3	(b AND c) OR (b AND d) OR (c AND d)
4	b XOR c XOR d

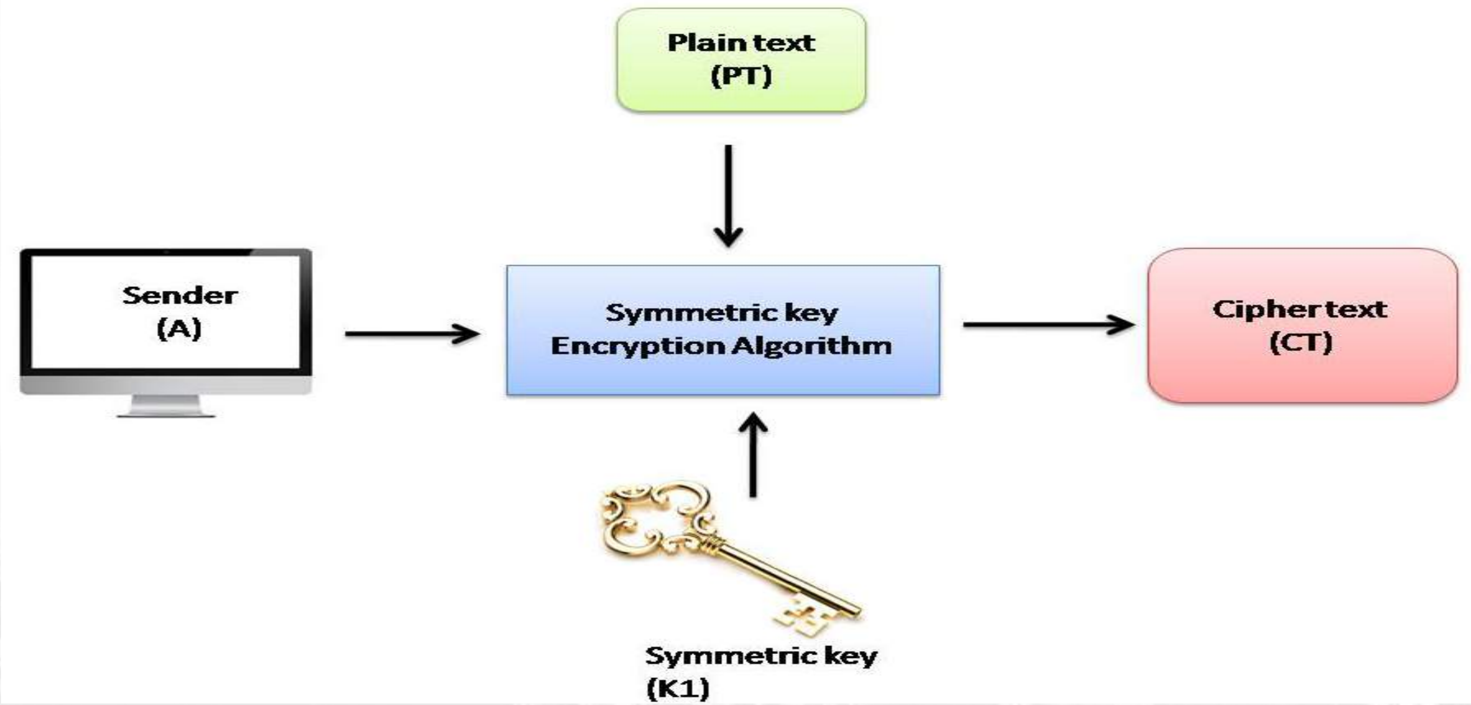
Types of Attack on Hashes

- **Preimage:** An attacker has an output and finds an input that hashes to that output
- **2nd Preimage:** An attacker has an output and an input x and finds a 2nd input that produces the same output as x
- **Collision:** An attacker finds two inputs that hash to the same output
- **Length Extension:** An attacker, knowing the length of message M and a digest of M signed by a sender can extend M with an additional message N and can compute the digest of M || N even without the key used to sign the digest of M

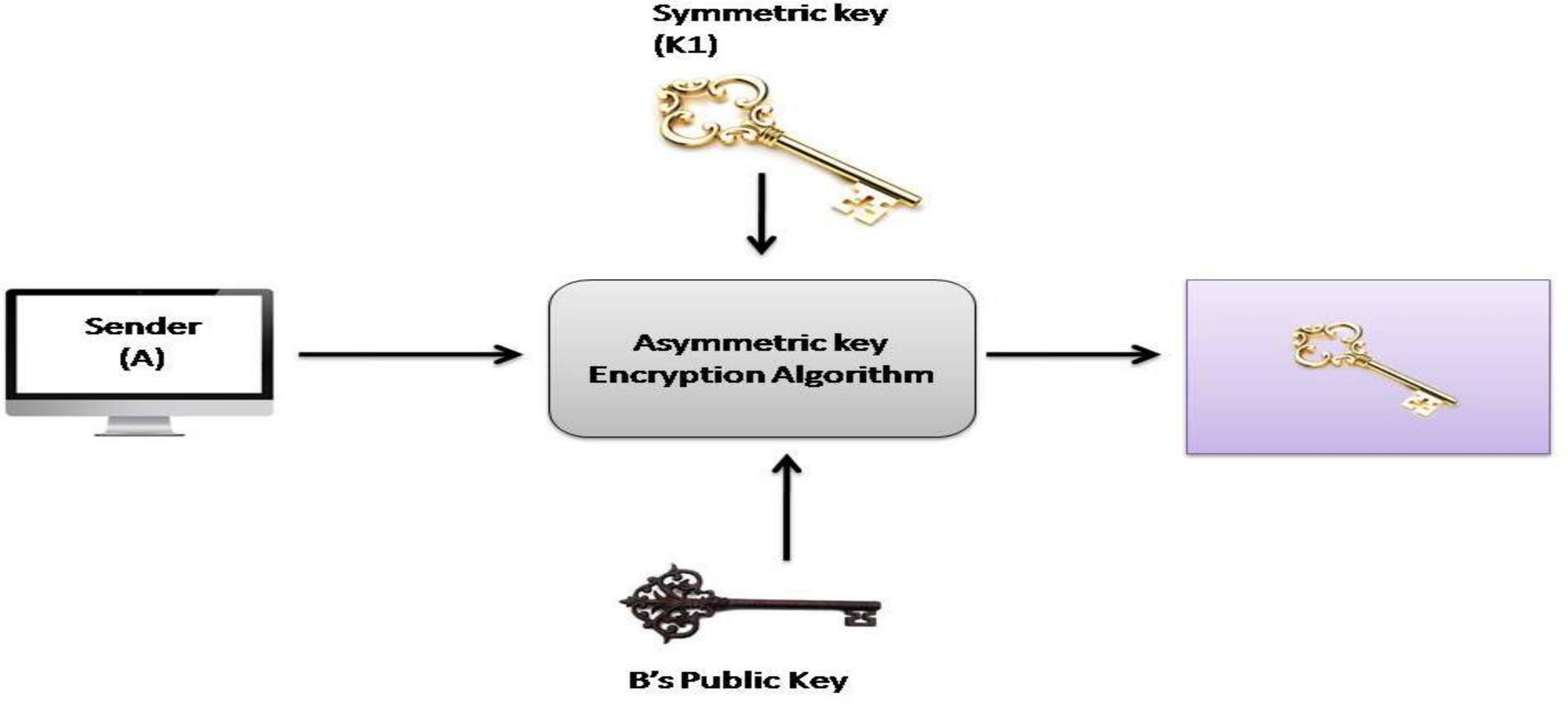
Comparison of MD5 and SHA

Point of discussion	MD5	SHA
Message digest length in bits	128	160
Attack to try and find the original message given a message digest	Requires 2^{128} operations to break in	Requires 2^{160} operations to break in, therefore more secure
Attack to try and find two messages producing the same message digest	Requires 2^{64} operations to break in	Requires 2^{80} operations to break in
Successful attacks so far	There have been reported attempts to some extent	No such claims so far
Speed	Faster (64 iterations, and 128-bit buffer)	Slower (80 iterations, and 160-bit buffer)
Software implementation	Simple, does not need any large programs or complex tables	Simple, does not need any large programs or complex tables

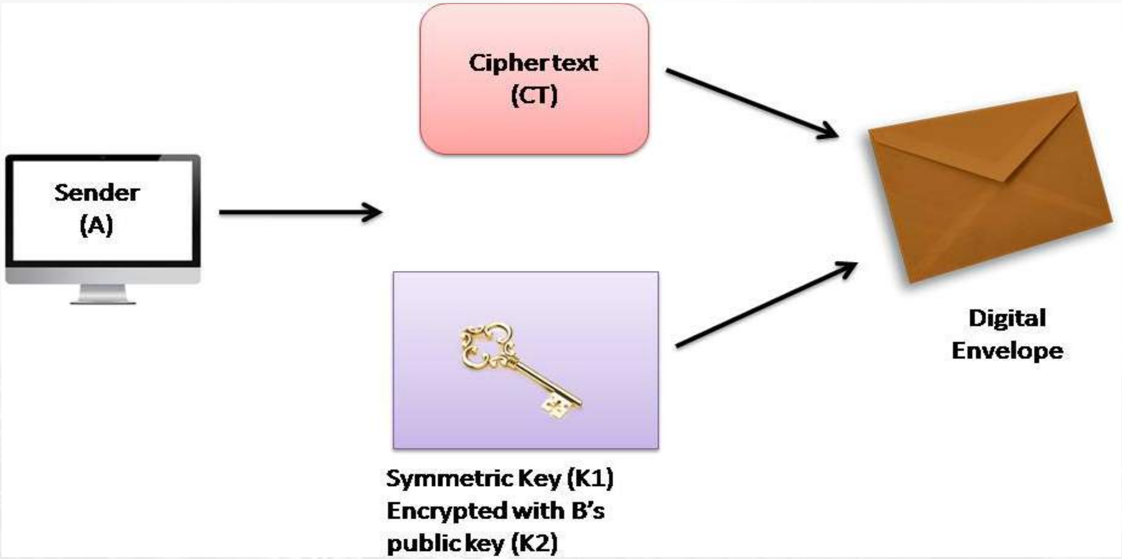
Symmetric and Asymmetric Key Cryptography Together

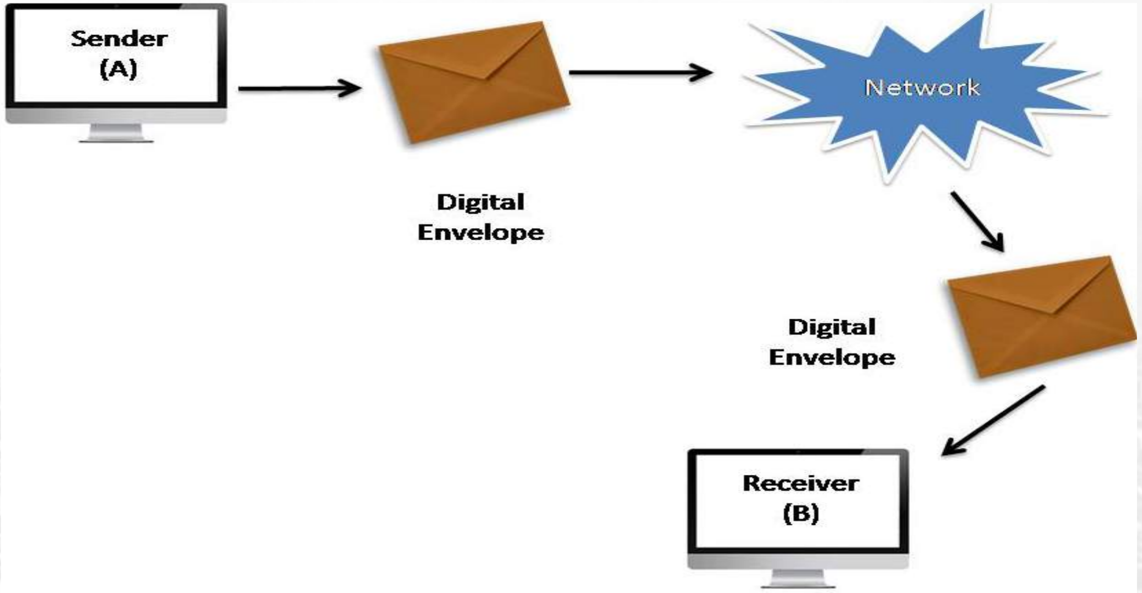


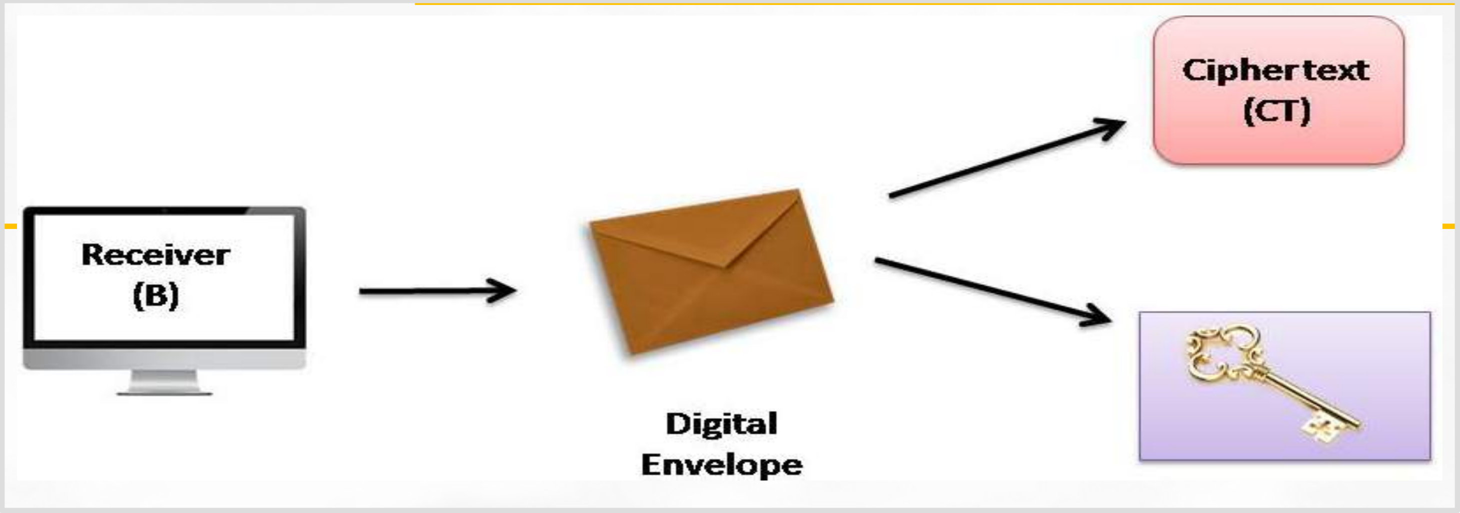
- Takes the one-time symmetric key (i.e. K1), and encrypts K1 with B’s public key (K2). This process is called **key wrapping** of the symmetric key

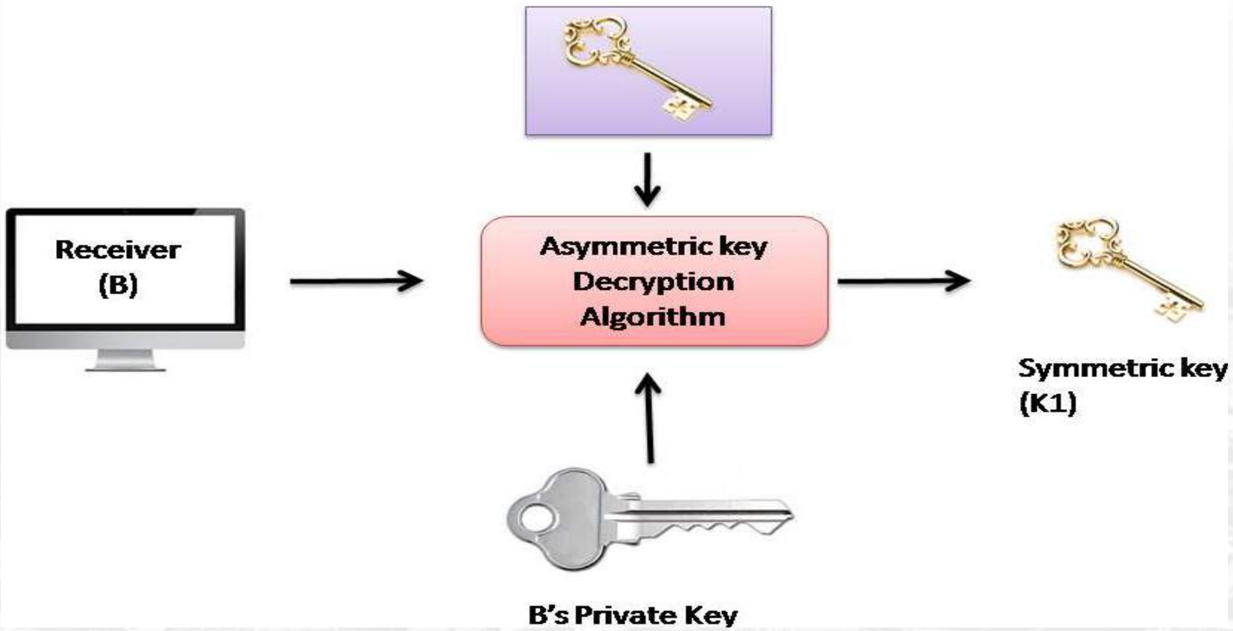


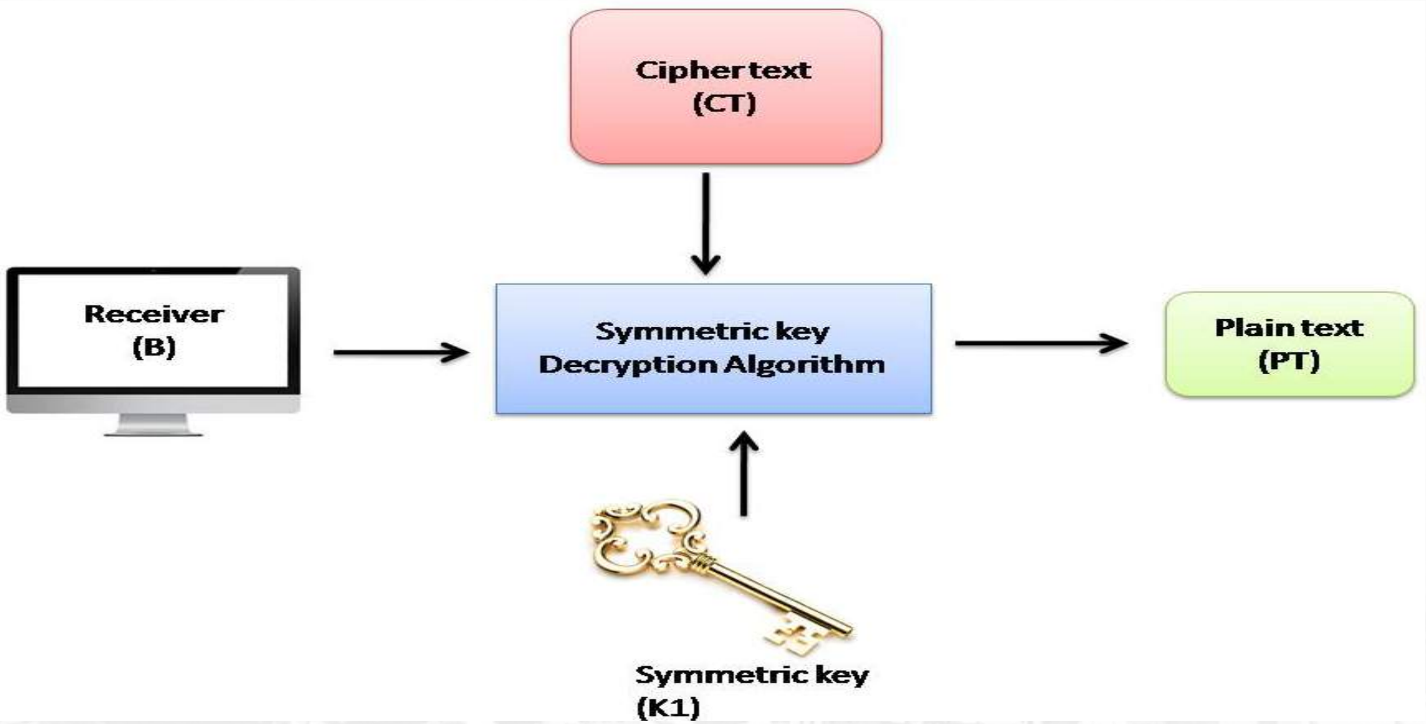
- A puts the cipher text CT and the encrypted symmetric key together inside a digital envelope.



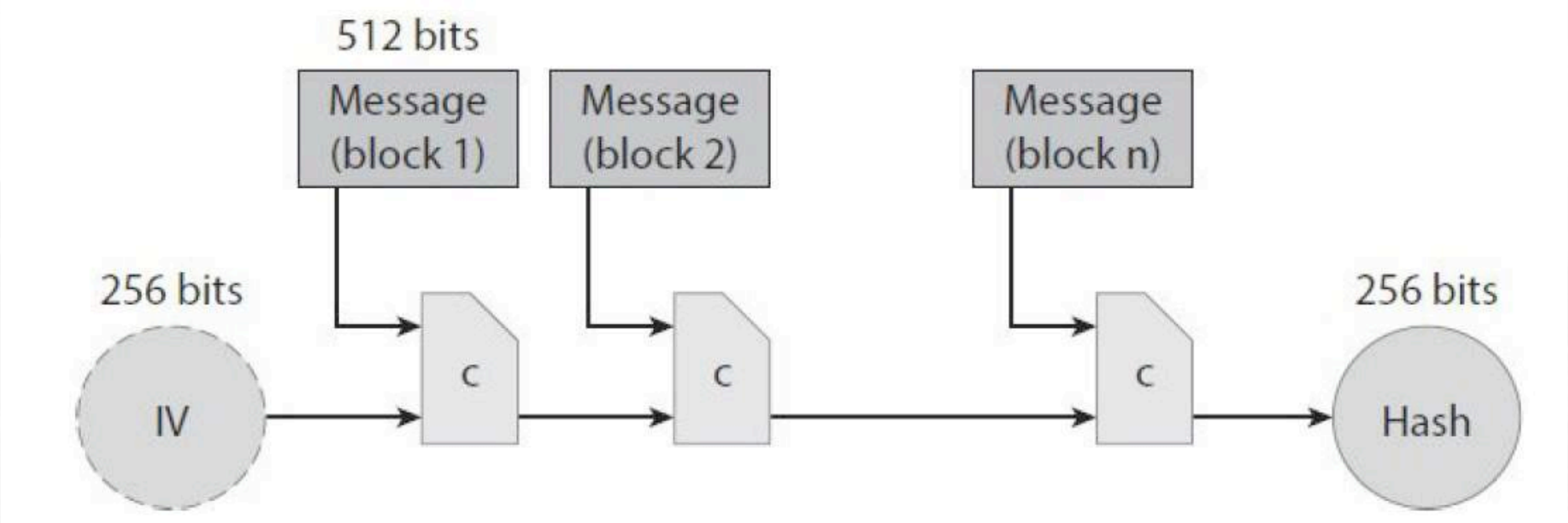



Receiver
(B)
Digital
Envelope
Cipher text
(CT)


Receiver
(B)
Asymmetric key
Decryption
Algorithm
Symmetric key
(K1)
B's Private Key


Cipher text
(CT)
Receiver
(B)
Symmetric key
Decryption Algorithm
Plain text
(PT)
Symmetric key
(K1)

SHA-256 hash function (simplified)



SHA-256 hash function (simplified)

- SHA-256 uses the Merkle- Damgård transform to turn a fixed-length collision-resistant compression function into a hash function that accepts arbitrary-length inputs

- The input is padded, so that its length is a multiple of 512 bits. IV stands for initialization vector.

Modeling Hash Functions

- Hash functions are the Swiss Army knife of cryptography: they find a place in a spectacular variety of applications.
- The flip side to this versatility is that different applications require slightly different properties of hash functions to ensure security.
- It has proven notoriously hard to pin down a list of hash function properties that would result in provable security across the board.

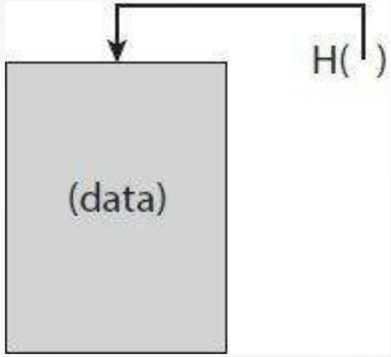
Modeling Hash Functions

- SHA-256 uses a compression function that takes 768-bit input and produces 256-bit outputs

- The block size is 512 bits

Hash Pointers and Data Structures

- A hash pointer is a data structure that turns out to be useful in many of the systems that we consider.

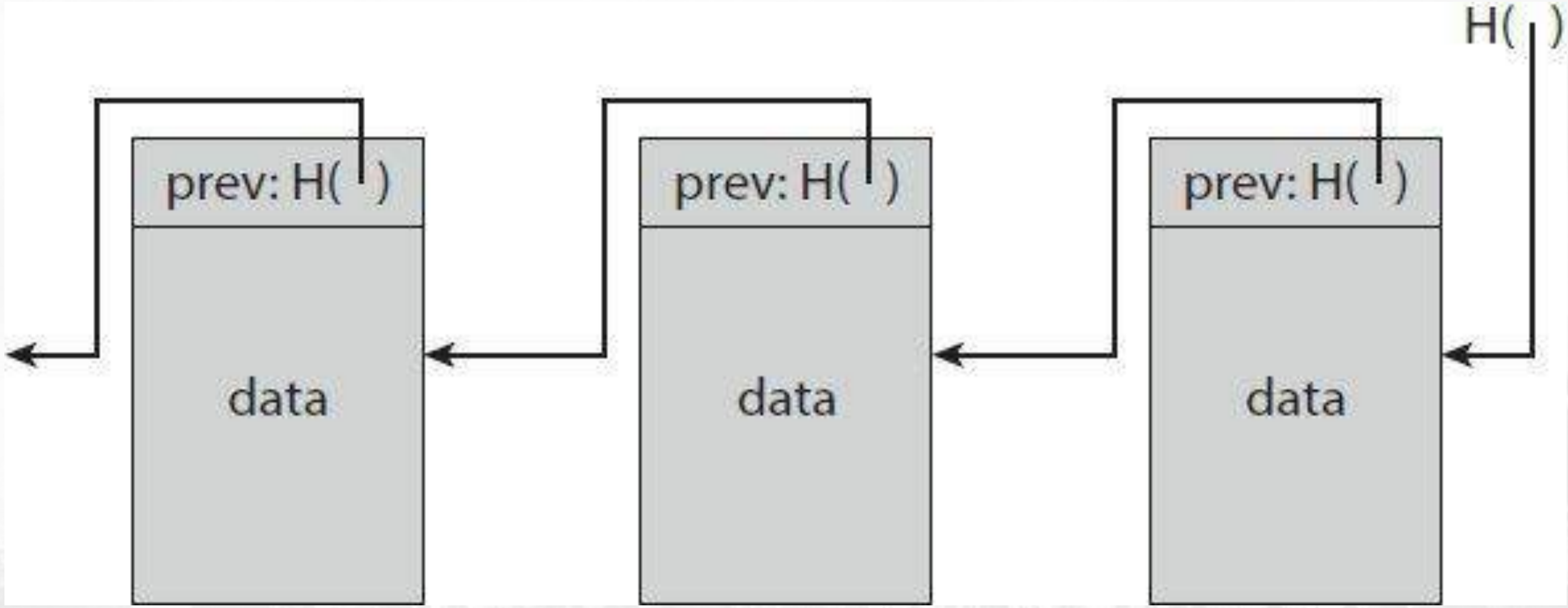


A hash pointer is simply a pointer to where some information is stored together with a cryptographic hash of the information.

Hash Pointers and Data Structures

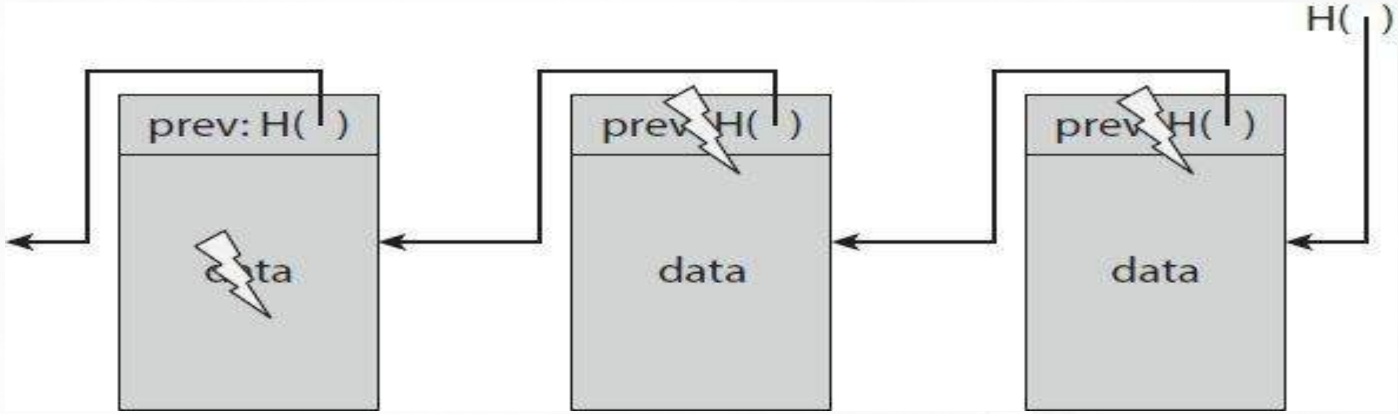
- a regular pointer gives you a way to retrieve the information, a hash pointer also allows you to verify that the information hasn’t been changed

Block Chain



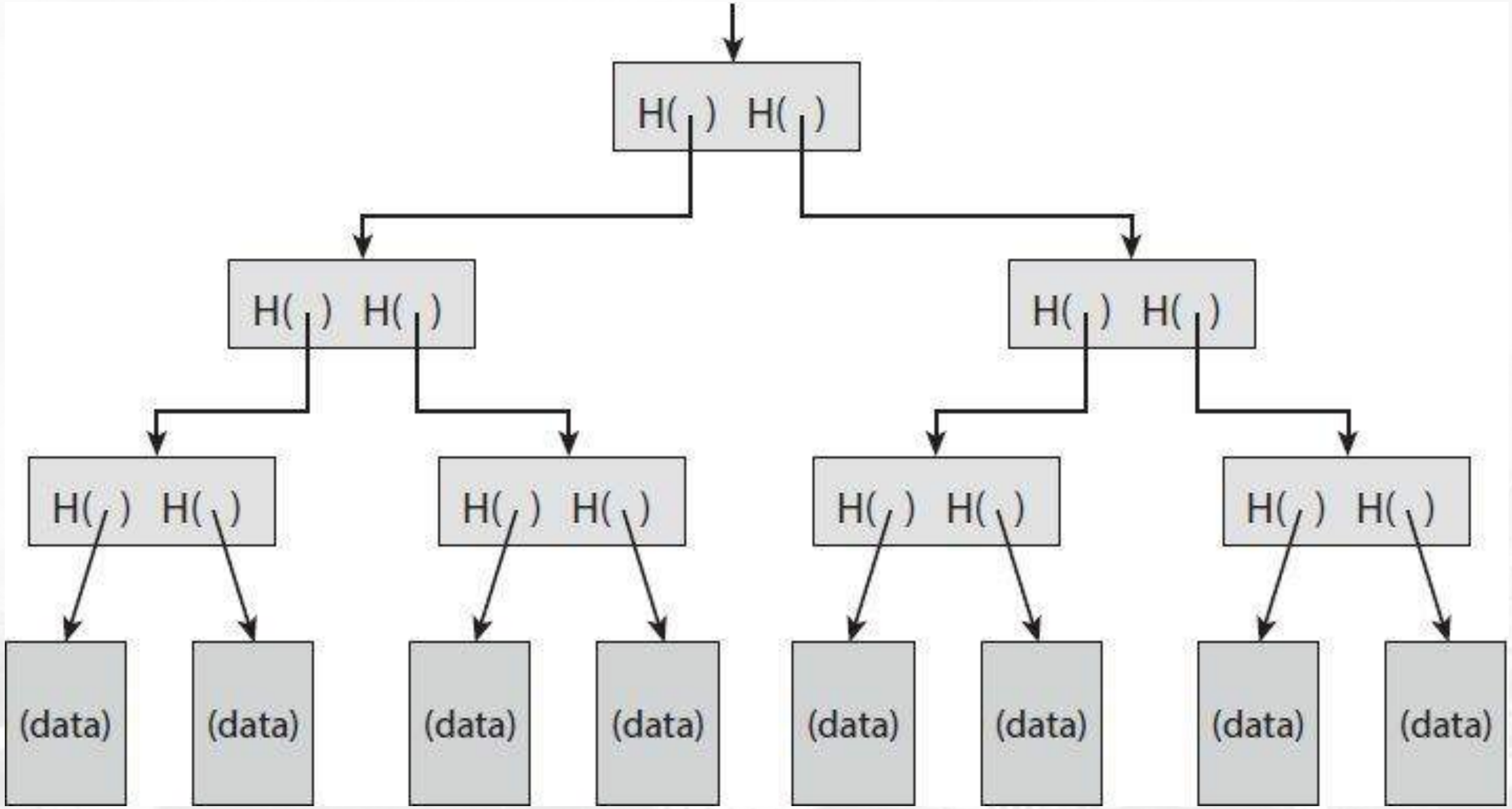
A block chain is a linked list that is built with hash pointers instead of pointers.

Block Chain

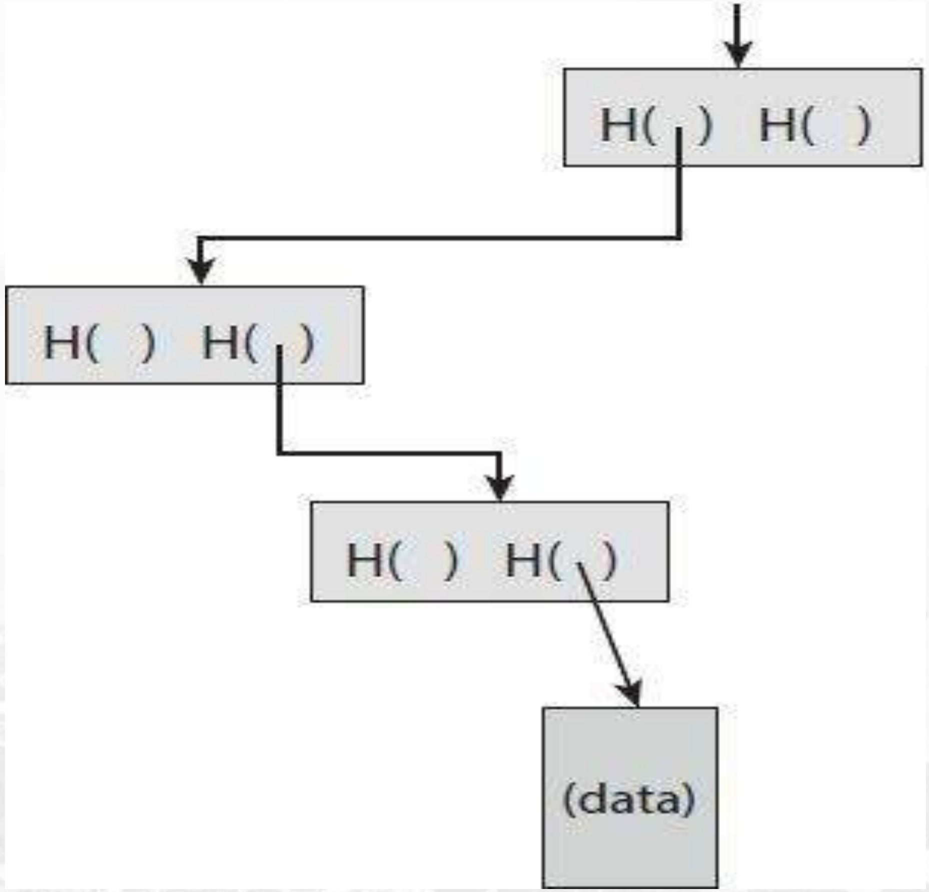


Tamper-evident log

Merkle Tree



Proof of Membership

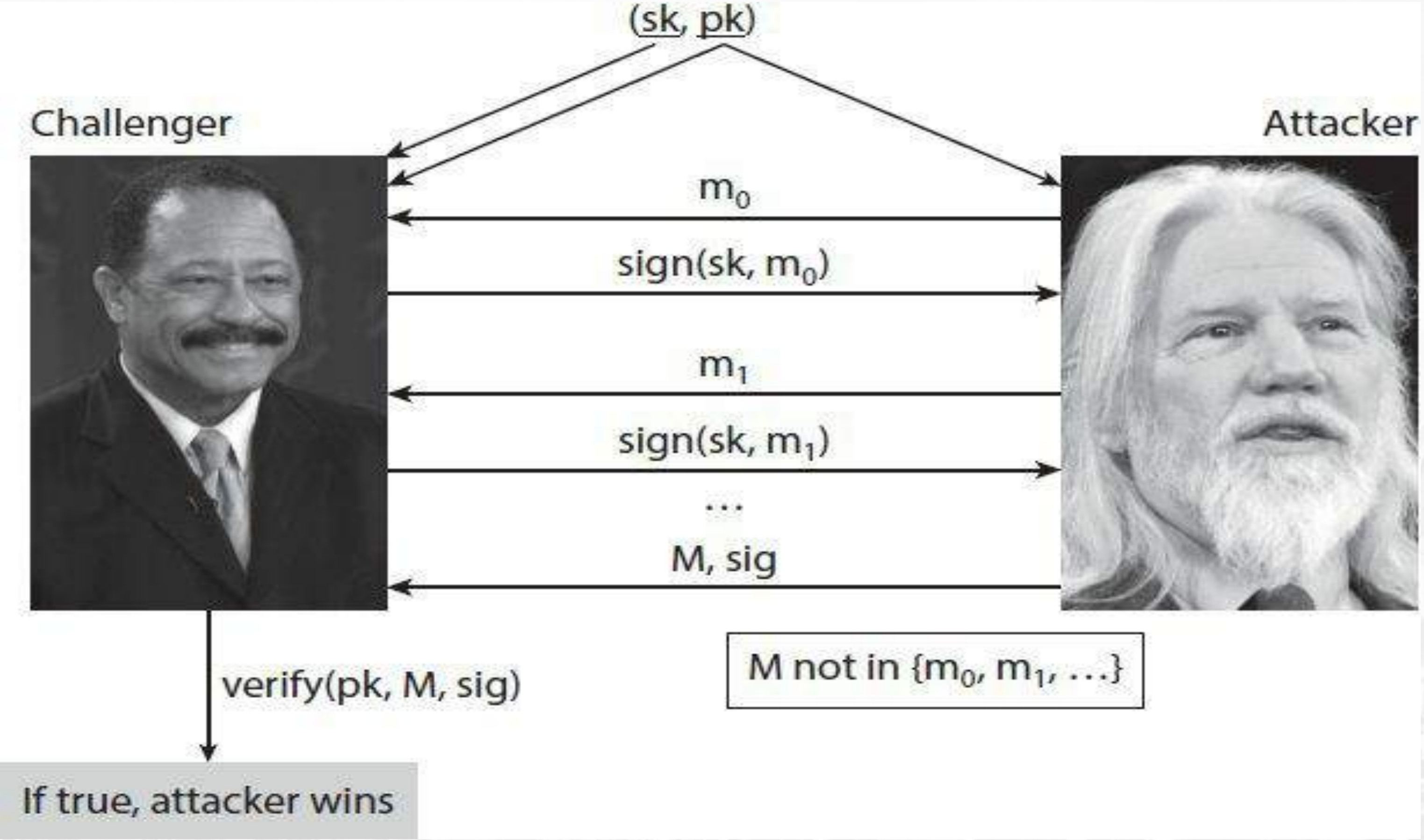


Proof of Nonmembership

- Using a sorted Merkle tree, it becomes possible to verify nonmembership in logarithmic time and space

- We can prove that a particular block is not in the Merkle tree.

Unforgeability Game



Practical Concerns

- Several practical things must be done to turn the algorithmic idea into a digital signature mechanism that can be implemented

- For example, many signature algorithms are randomized (in particular, the one used in Bitcoin)

- we therefore need a good source of randomness

PUBLIC KEYS AS IDENTITIES

- Cryptocurrencies and Encryption

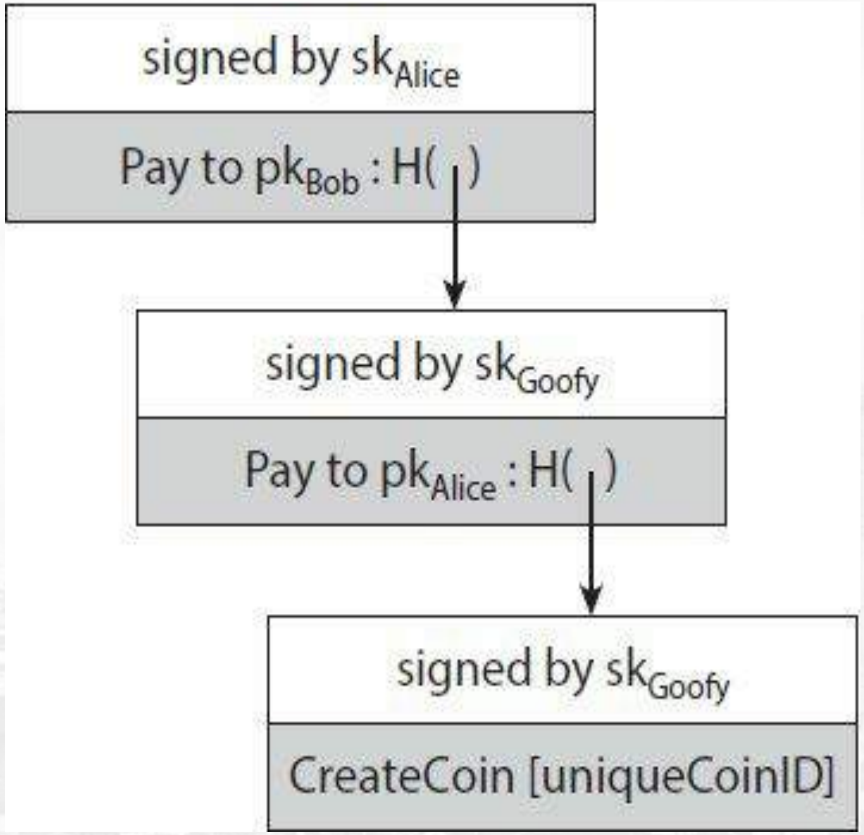
- Decentralized Identity Management
- Security and Randomness

TWO SIMPLE CRYPTOCURRENCIES

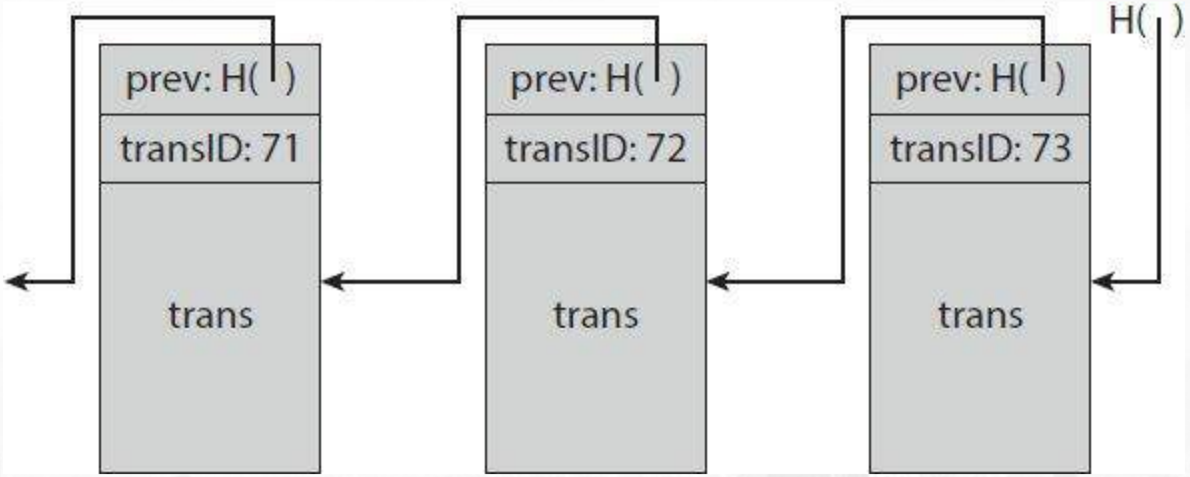
- Goofycoin

- Goofy can create new coins by simply signing a statement that he's making a new coin with a unique coin ID.
- Whoever owns a coin can pass it on to someone else by signing a statement that says, "Pass on this coin to X" (where X is specified as a public key).
- Anyone can verify the validity of a coin by following the chain of hash pointers back to its creation by Goofy, verifying all signatures along the way.

TWO SIMPLE CRYPTOCURRENCIES



Scroogecoin



To solve the double-spending problem, we'll design another cryptocurrency, called *Scroogecoin*. Scroogecoin is built off of Goofycoin, but it's a bit more complicated in terms of data structures.

How Bitcoin Achieves Decentralization

Centralization Versus Decentralization

- Decentralization is an important concept that is not unique to Bitcoin
- The notion of competing paradigms of centralization versus decentralization arises in a variety of different digital technologies
- Decentralization is not all or nothing; almost no system is purely decentralized or purely centralized.

How Bitcoin Achieves Decentralization

With this in mind, let’s break down the question of how the Bitcoin protocol achieves decentralization into five more specific questions:

1. Who maintains the ledger of transactions?
2. Who has authority over which transactions are valid?
3. Who creates new bitcoins?
4. Who determines how the rules of the system change?
5. How do bitcoins acquire exchange value?

How Bitcoin Achieves Decentralization

The first three questions reflect the technical details of the Bitcoin protocol

—these three questions are the focus

DISTRIBUTED CONSENSUS

- Distributed consensus has various applications, and it has been studied for decades in computer science
- The traditional motivating application is reliability in distributed systems

Distributed consensus protocol. There are *n* nodes that each have an input value. Some of these nodes are faulty or malicious. A distributed consensus protocol has the following two properties:

- It must terminate with all honest nodes in agreement on the value.
- The value must have been generated by an honest node.

Bitcoin

- Bitcoin is a collection of concepts and technologies that form the basis of a digital money ecosystem.

- Units of currency called bitcoins are used to store and transmit value among participants in the bitcoin network.

- Bitcoin users communicate with each other using the bitcoin protocol primarily via the Internet, although other transport networks can also be used.

Bitcoin

- The bitcoin protocol stack, available as open source software, can be run on a wide range of computing devices, including laptops and smartphones, making the technology easily accessible.

- Users can transfer bitcoin over the network to do just about anything that can be done with conventional currencies, such as buy and sell goods, send money to people or organizations, or extend credit.

Bitcoin

- Bitcoin technology includes features that are based on encryption and digital signatures to ensure the security of the bitcoin network.

- Bitcoins can be purchased, sold and exchanged for other currencies at specialized currency ex- changes.

- Bitcoin in a sense is the perfect form of money for the Internet because it is fast, secure, and borderless.

Bitcoin

- Unlike traditional currencies, bitcoins are entirely virtual

- There are no physical coins or even digital coins per se.

- Users of bitcoin own keys which allow them to prove owner- ship of transactions in the bitcoin network, unlocking the value to spend it and transfer it to a new recipient.

Digital Currencies Before Bitcoin

- Can I trust the money is authentic and not counterfeit?

- Can I be sure that no one else can claim that this money belongs to them and not me?

- When cryptography started becoming more broadly available and understood in the late 1980s, many researchers began trying to use cryptography to build digital curren- cies. These early digital currency projects issued digital money, usually backed by a national currency or precious metal such as gold.

History of Bitcoin

- Bitcoin was invented in 2008 by Satoshi Nakamoto with the publication of a paper titled "Bitcoin: A Peer-to-Peer Electronic Cash System". Satoshi Nakamoto combined several prior inventions such as b-money and HashCash to create a completely de-centralized electronic cash system that does not rely on a central authority for currency issuance or settlement and validation of transactions.

Bitcoin Uses, Users and Their Stories

- Bitcoin is a technology, but it expresses money which is fundamentally a language for exchanging value between people. Let’s look at the people who are using bitcoin and some of the most common uses of the currency and protocol through their stories.

Bitcoin Uses, Users and Their Stories

- *North American Low Value Retail*
- *North American High Value Retail*
- *Offshore Contract Services*
- *Charitable Donations*

bitcoin clients

Full Client

A full client, or “full node” is a client that stores the entire history of bitcoin trans- actions (every transaction by every user, ever), manages the user’s wallets and can initiate transactions directly on the bitcoin network. This is similar to a standalone email server, in that it handles all aspects of the protocol without relying on any other servers or third party services.

bitcoin clients

Light Client

A lightweight client stores the user’s wallet but relies on third-party owned servers for access to the bitcoin transactions and network. The light client does not store a full copy of all transactions and therefore must trust the third party servers for transaction validation. This is similar to a standalone email client that connects to a mail server for access to a mailbox, in that it relies on a third party for interactions with the network.

bitcoin clients

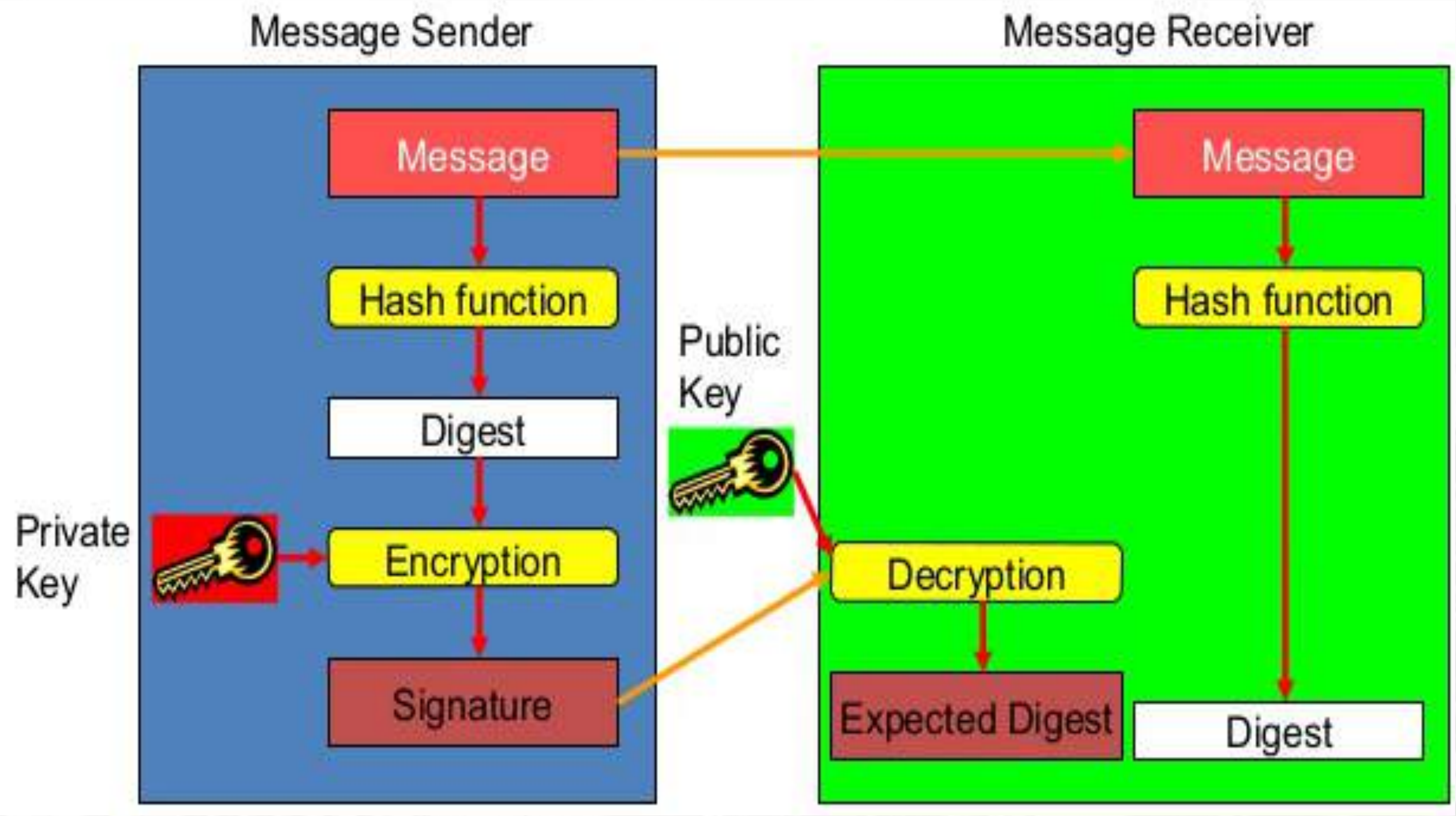
Web Client

Web-clients are accessed through a web browser and store the user's wallet on a server owned by a third party. This is similar to webmail in that it relies entirely on a third party server.

Digital Signature Techniques

- ❖ DSS (Digital Signature Standard) uses SHA-1
- ❖ RSA and DSA

RSA and Digital Signature



Digital Signature Algorithm (DSA)

- creates a 320 bit signature with 512-1024 bit security
- smaller and faster than RSA
- a digital signature scheme only
- security depends on difficulty of computing discrete logarithms
- variant of ElGamal & Schnorr schemes

DSA Key Generation

- have shared global public key values (**p, q, g**):
  - choose a large prime **p** with $2^{L-1} < p < 2^{L}$
    - where L= 512 to 1024 bits and is a multiple of 64
  - choose 160-bit prime number q
    - such that q is a 160 bit prime divisor of (p-1)
  - choose $g = h^{(p-1)/q}$
    - where $1 < h < p-1$ and $h^{(p-1)/q} \bmod p > 1$
- users choose **private** & compute **public key**:
  - choose random private key: $x < q$
  - compute public key: $y = g^{x} \bmod p$

Note: L will be one member of the set {512, 576, 640, 704, 768, 832, 896, 960, 1024}

DSA Signature Creation

- to **sign** a message M the sender:
  - generates a random signature key k, $k < q$
- then computes signature pair:

$$\mathbf{r = (g^k \bmod p) \bmod q}$$

$$\mathbf{s = [k^{-1}(SHA(M) + x * r)] \bmod q}$$

- sends **signature (r, s)** with message M

DSA Signature Verification

- having received M & signature (r, s)
- to **verify** a signature, recipient computes:

  $w = s^{-1} \bmod q$

  $u_1 = [SHA(M) * w] \bmod q$

  $u_2 = (r * w) \bmod q$

  $v = [(g^{u1} * y^{u2}) \bmod p] \bmod q$

- if $v = r$ then signature is verified

DSA Signature Verification

- having received M & signature (r, s)
- to **verify** a signature, recipient computes:

$w = s^{-1} \bmod q$

$u_1 = [SHA(M) * w] \bmod q$

$u_2 = (r * w) \bmod q$

$v = [(g^{u1} * y^{u2}) \bmod p] \bmod q$

- if $v = r$ then signature is verified