

Understanding the computation behind Blockchain

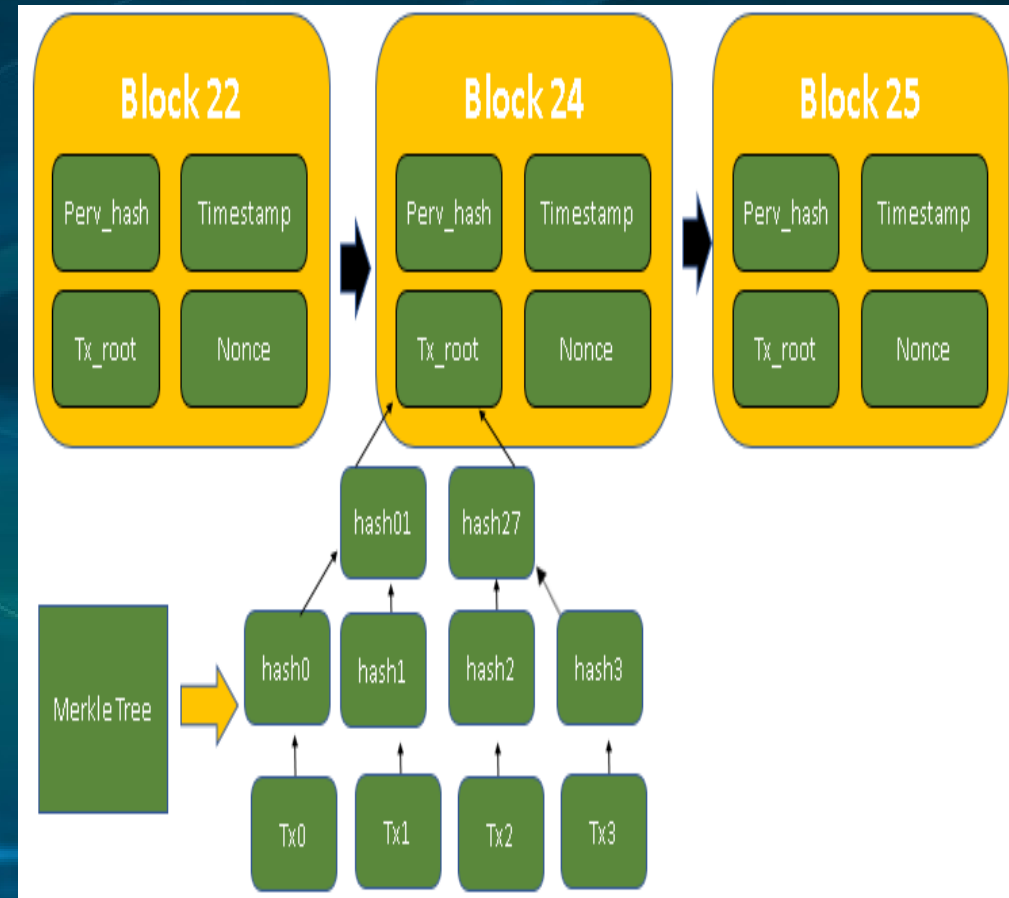
PRESENTED BY BERTONY BORNELUS, GRADUATE STUDENT
CIS4360

Contents Overview

- Overview of Blockchain Technology
- Blockchain Job market boom
- Leading Blockchain companies
- Centralized Vs. Decentralized
- Ledger vs. Blockchain Disturbed Ledger
- Public Key cryptography
- Hash Functions
- How the SHA Algorithm Works
- Blockchain mining
- Merkle Tree
- Conclusion

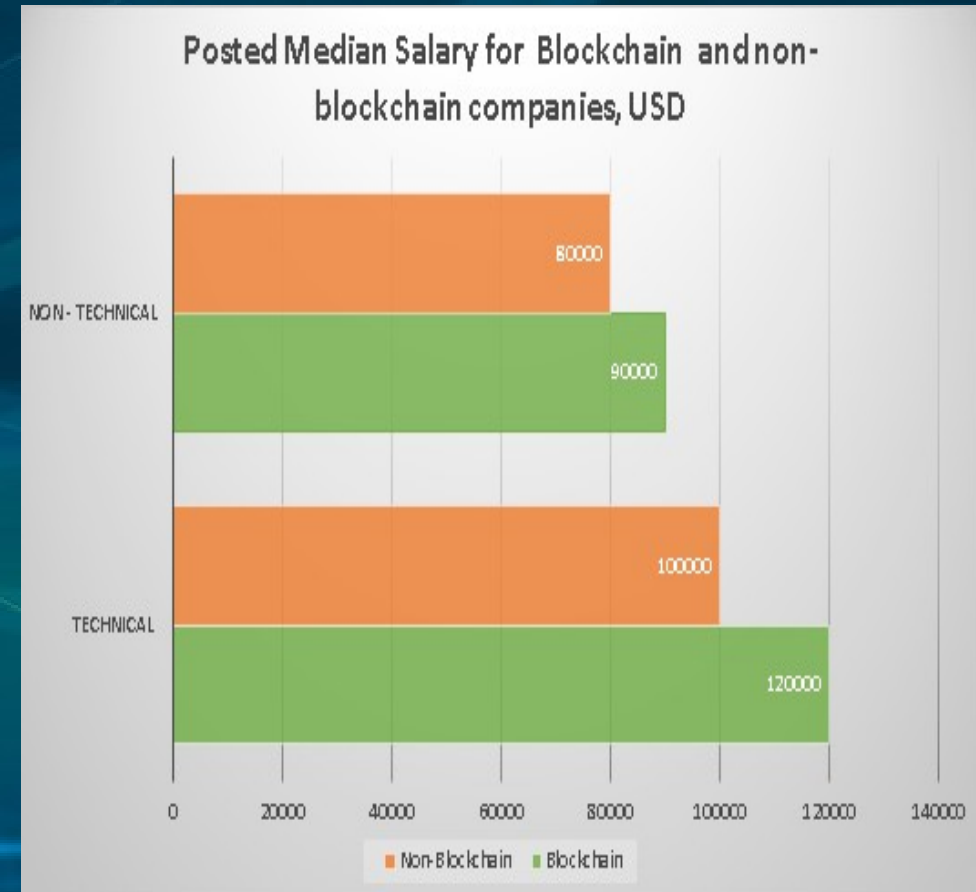
Overview of Blockchain Technology?

- Blockchain – **A growing list of records, called Blocks, which are linked using Cryptography.**
- Blockchain contains a cryptographic hash of the pervious block, a timestamp, and transaction data (Represented as a **merkle tree root hash**)



Blockchain job market boom

- Since the inception of Bitcoin, blockchain technology use cases has grown to several industries: financial, supply chain, and healthcare.
 - Several companies are currently investing in blockchain technology:
 - IBM : IBM Hyperledger
 - Microsoft : Azure Microsoft azure blockchain
 - Amazon: AWS Blockchain
 - **Technical Blockchain median Salary \$120,000 on par with A.I. developers**
- Understanding the computation behind Blockchain



Leading company: Bitcoin

- **First and Most Popular blockchain application.**
- **Cryptocurrency, a form of electronic cash.**
- **No Central Bank or central Authority.**
- **Transactions are verified by network nodes distributed ledger.**
- **Founded in 2009 as of today
1 bitcoin = 4,026 USD**



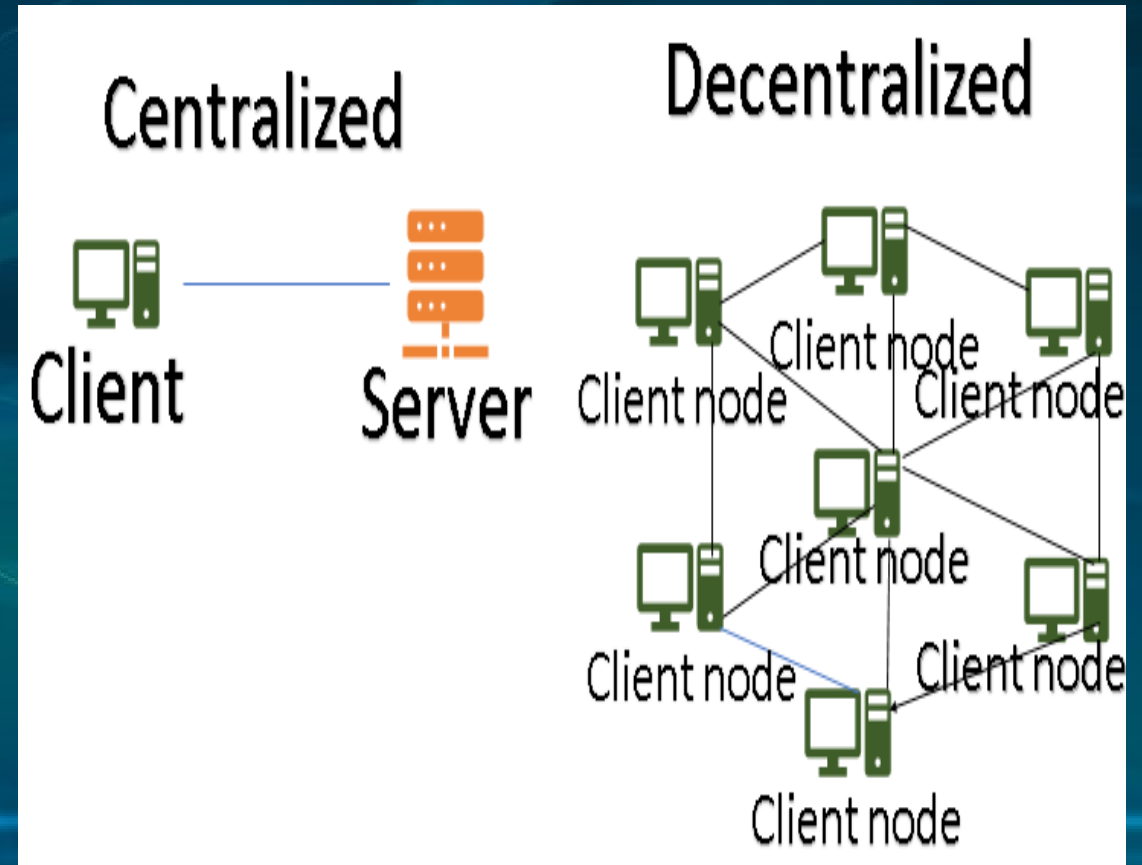
Leading company: Ethereum

- **One of the Leading Platform for Blockchain development.**
- **Decentralized platform that smart contract :applications that run exactly as programmed without possibility of downtime, fraud or third interference.**
- **1 ether = 151 USD**



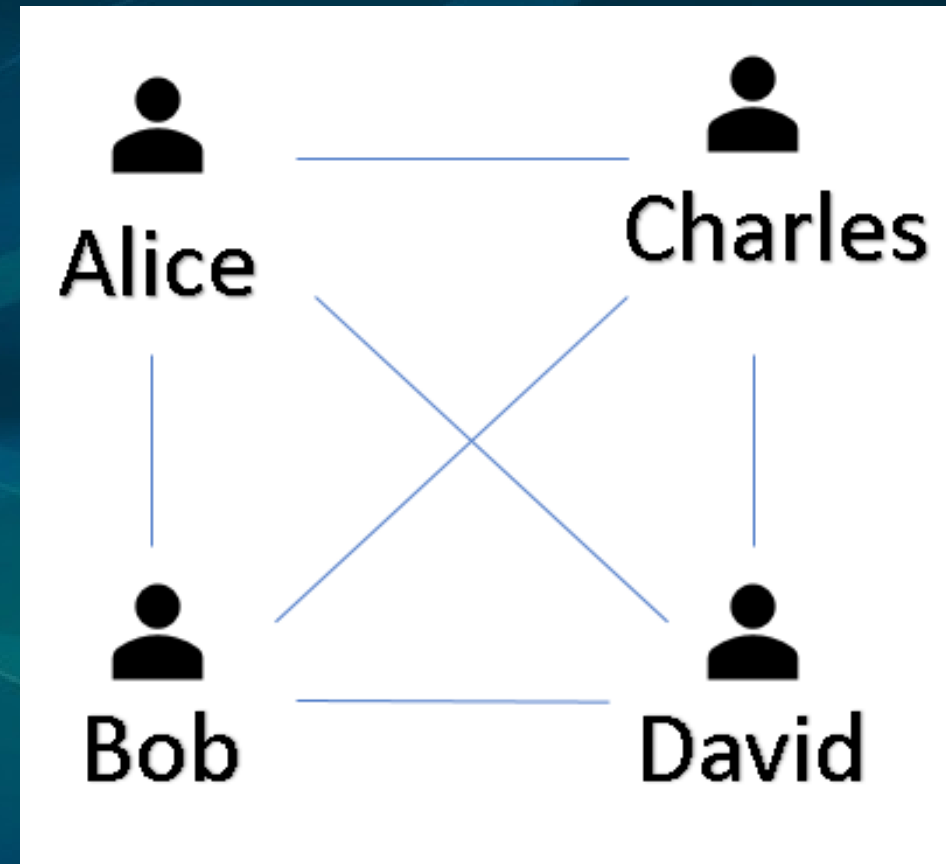
Centralized Vs. Decentralized

- **Centralized network** users prove authentication to gain access to one central authority . Example : Banks - login username/password
- **Decentralized network** every node has a copy of every record and must reach consensus to authenticate transaction.
- **No single point of failure**



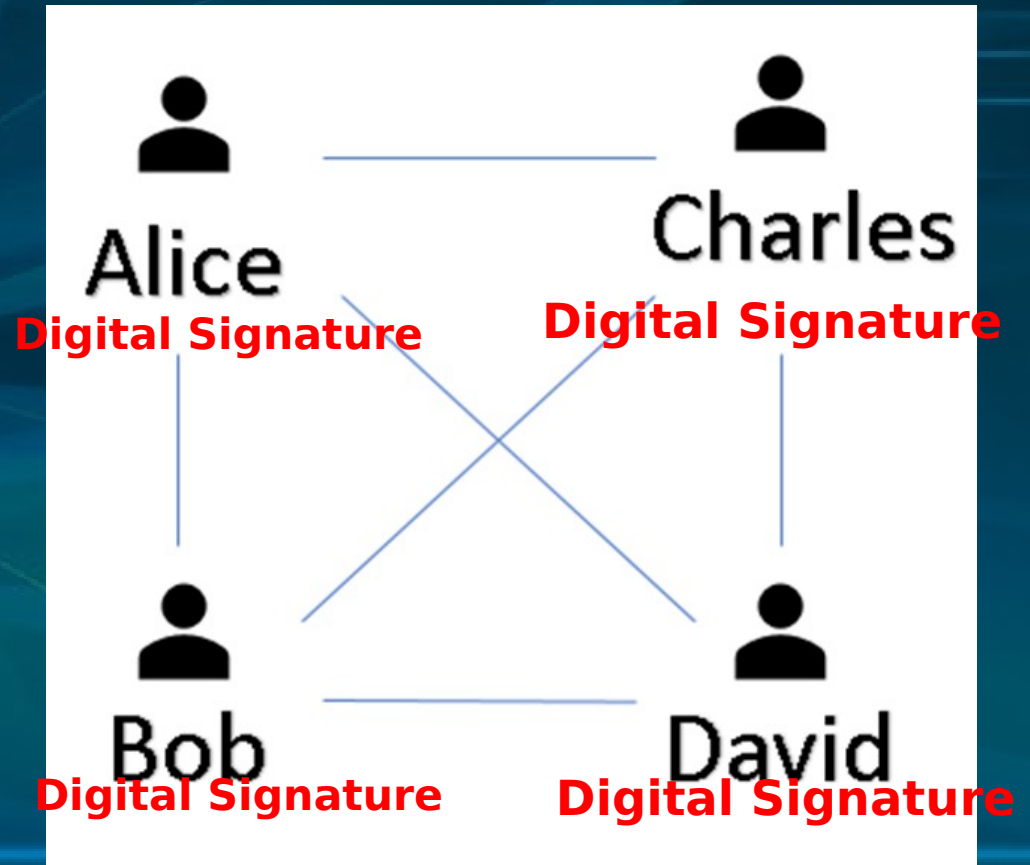
Ledger

- **Alice Paid bob \$100**
- **Bob paid David \$30**
- **David owes Alice \$20**
- **Charles owes Bob \$25**



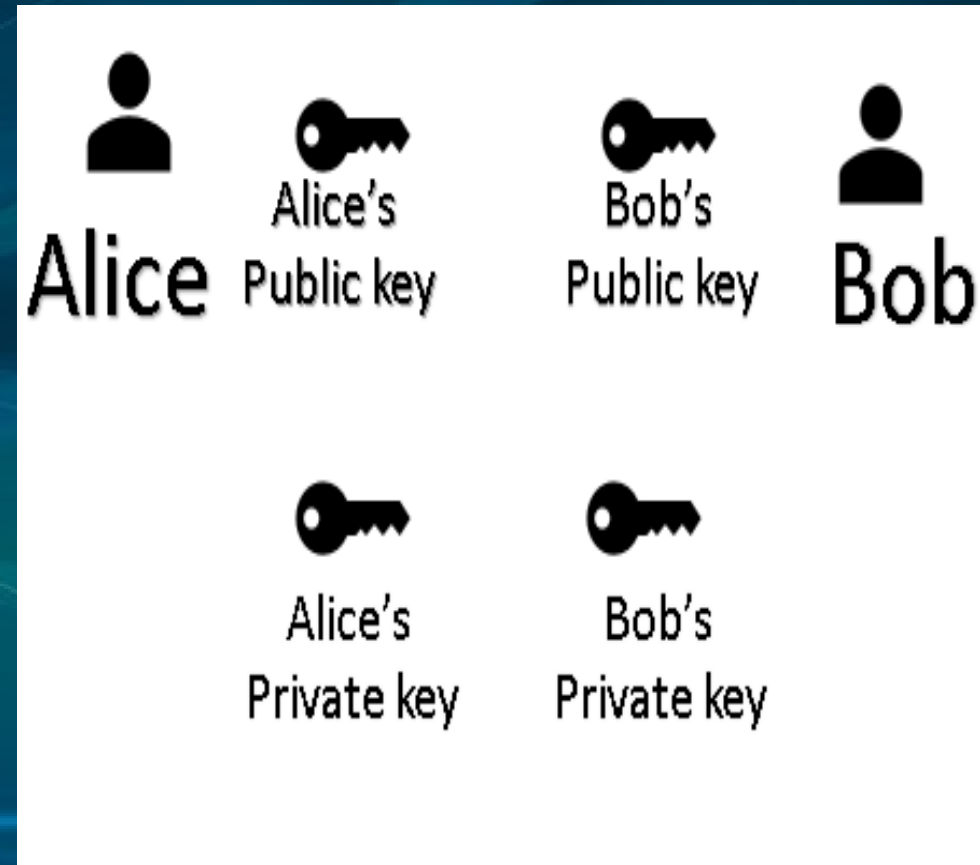
Blockchain Ledger

- Alice Paid bob \$100,
Digital Signature
- Bob paid David \$30,
Digital Signature
- David owes Alice \$20,
Digital Signature
- Charles owes Bob \$25,
Digital Signature



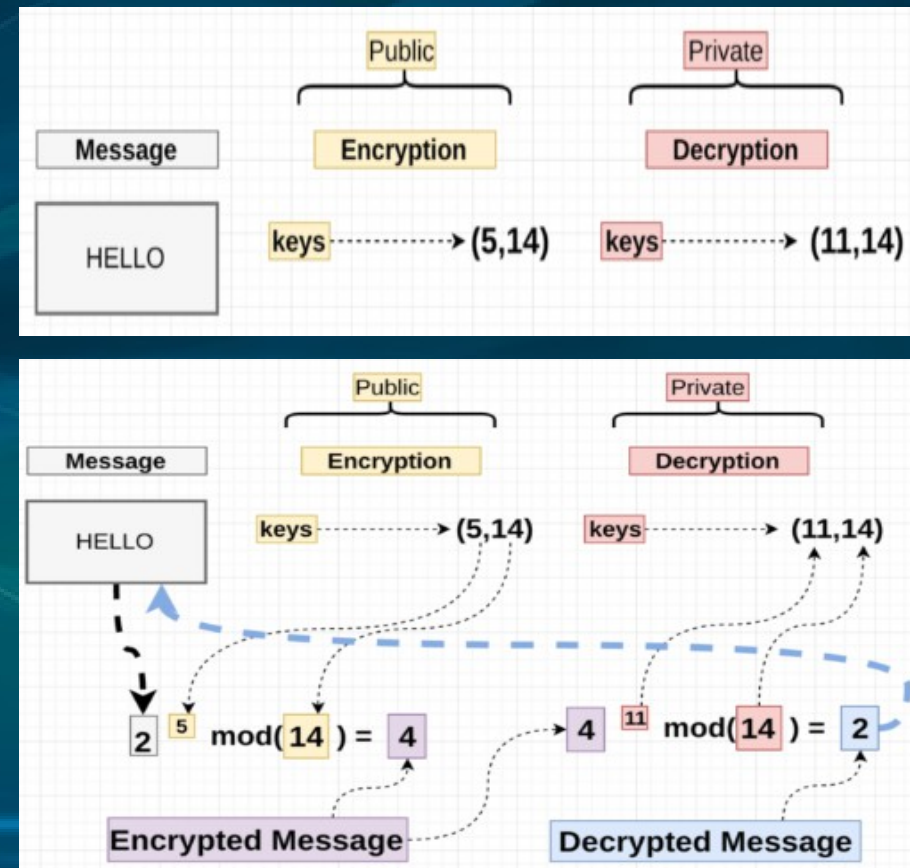
Public Key cryptography

- If two or more people want to secure transaction over the internet they use asymmetric cryptography also referred to as public-private key cryptography.
- This technology allows them to prove their identity with a set of cryptographic keys: private key(secret key) and a public key.
- **Both Alice and Bob have a pair of keys; Alice paid Bob \$100 created a Digital Signature to Verify transaction**
- **Alice uses her private key to sign the transaction Private key is known as secret key sk. Bob use public key to verify transaction.**



RSA Algorithm

- More frequently it is used to encrypt and pass around **asymmetric** keys which can actually deal with encryption at a **faster** speed.



Verifying Transaction

- To Sign is (Message, secret key) = Signature
- To verify is (Message, Signature, public key) = T/F
- RSA Algorithm
- Encryption: $C = M^e \pmod n$
- Decryption: $M = C^d \pmod n$

RSA Algorithm

Key Generation

Select p, q	p and q , both prime; $p \neq q$
Calculate $n = p \times q$	
Calculate $\phi(n) = (p-1)(q-1)$	
Select integer e	$\gcd(\phi(n), e) = 1; 1 < e < \phi(n)$
Calculate d	$de \pmod{\phi(n)} = 1$
Public key	$KU = \{e, n\}$
Private key	$KR = \{d, n\}$

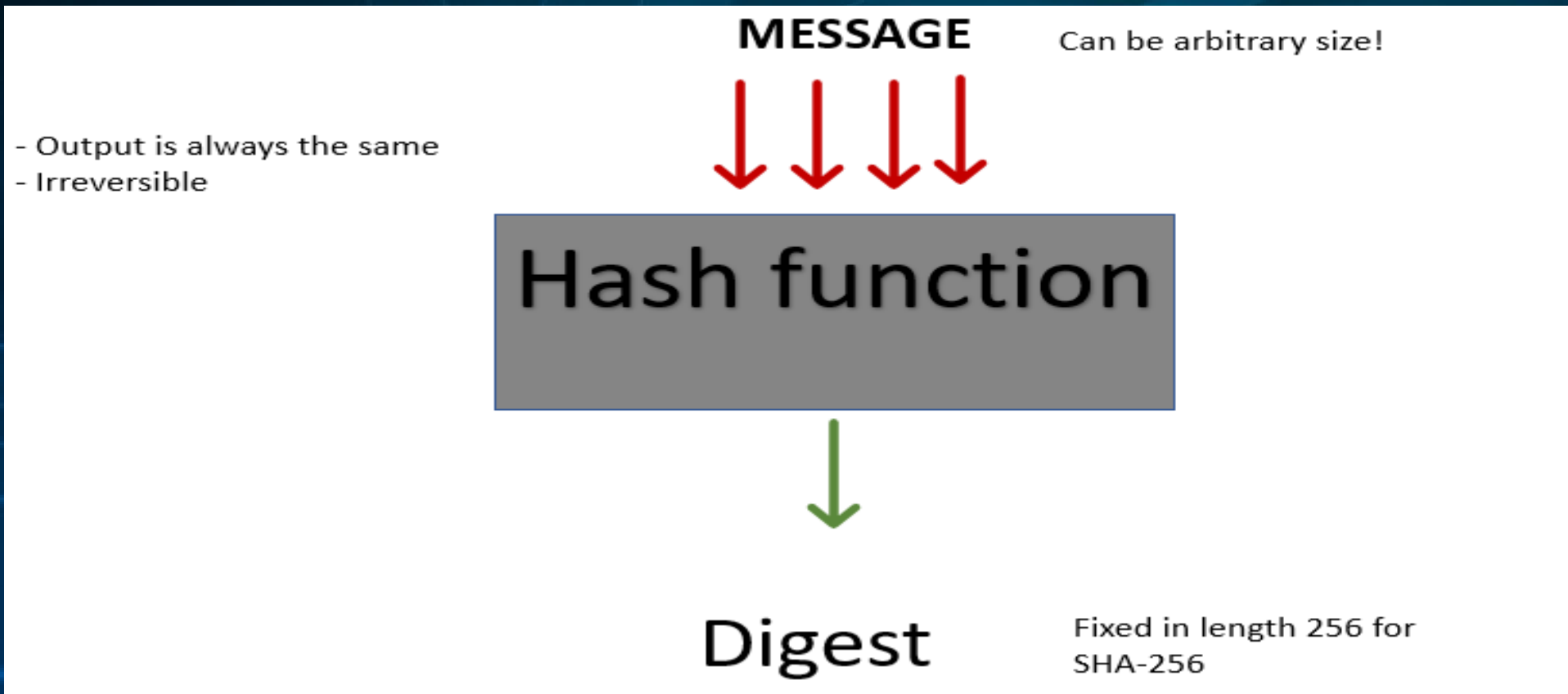
Encryption

Plaintext:	$M < n$
Ciphertext:	$C = M^e \pmod n$

Decryption

Plaintext:	C
Ciphertext:	$M = C^d \pmod n$

Hash Functions



How the SHA Algorithm Works

- **Step 1:** Take your message get the ASCII value, and convert your message (M) into bit.

Message : hello

Become: 01101000 01100101 01101100 01101100 01101111

- **Step 2:** Pad the message to make it 448 mod 512 bits long. Message is 104 bits. We need to add 344. Add 1, then 343 0's, then add a 64 bit value which represents the original message length.

We get:

M=

```
01101000 01100101 01101100 01101100 01101111 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 01101000
```

How the SHA Algorithm Works

- **Step 4:** Start by defining the variables and subfunctions used in the algorithm
 - SHA functions uses some Constant Variables and functions that are always used.
 - If it is longer then 512 bits it is broken up into 512 bit blocks.

$M[1,2,...N] = N$ is every bit block

- **The 4 Processing functions**

- Boolean function applied every block in the blockchain system

$$f(t,B,C,D) = (B \text{ and } C) \text{ OR } ((\text{NOT } B) \text{ AND } D) \quad (0 \leq t \leq 19)$$

$$f(t,B,C,D) = B \text{ XOR } C \text{ XOR } D \quad (20 \leq t \leq 39)$$

$$f(t,B,C,D) = (B \text{ and } C) \text{ OR } (B \text{ AND } D) \text{ OR } (C \text{ AND } D) \quad (40 \leq t \leq 59)$$

$$f(t,B,C,D) = (B \text{ XOR } C \text{ XOR } D) \quad (60 \leq t \leq 79)$$

How the SHA Algorithm Works

Other Variables

- 9 Constant Variables
 - H Variables are the starting point these variables get hash
- | | | |
|---------------------|-----------------------|-------------------|
| $K(t) = 0x5A827999$ | $(0 \leq t \leq 19)$ | $H0 = 0x67452301$ |
| $K(t) = 0x6ED9EBA1$ | $(20 \leq t \leq 39)$ | $H1 = 0xEFCDA89$ |
| $K(t) = 0x8F1BBCDC$ | $(40 \leq t \leq 59)$ | $H2 = 0x98BADCFE$ |
| $K(t) = 0xCA62C1d6$ | $(60 \leq t \leq 79)$ | $H3 = 0xC3D2E1F0$ |
| | | $H4 = 0xC3D2E1F0$ |

Step 6 : Put it through the hashing loop

- For every 512 bit block k in $M[1,2,...N]$
 - Divide $M[K]$ into 16 words ($W0$ to $W15$)
 - Next Define next values $W16$ to $W79$ are defined by some Boolean algorithm
- $W(t) = (W)(t-3) \text{ XOR } W(t-8) \text{ XOR } W(t-14) \text{ XOR } (t-16) < \text{single digit bit shift} << 1$**
- The 80 Step Processing Loop

How the SHA Algorithm Works

Step 7: The Step Process Loop

$A = H_0, B = H_1, C = H_2, E = H_4$

For $t = 0$ to 79 {

$TEMP = A \lll 5 + f(t; B, C, D) + W(t) + K(t)$

$E = D;$

$D = C;$

$C = B \lll 30;$

$B = A;$

$A = TEMP$

}

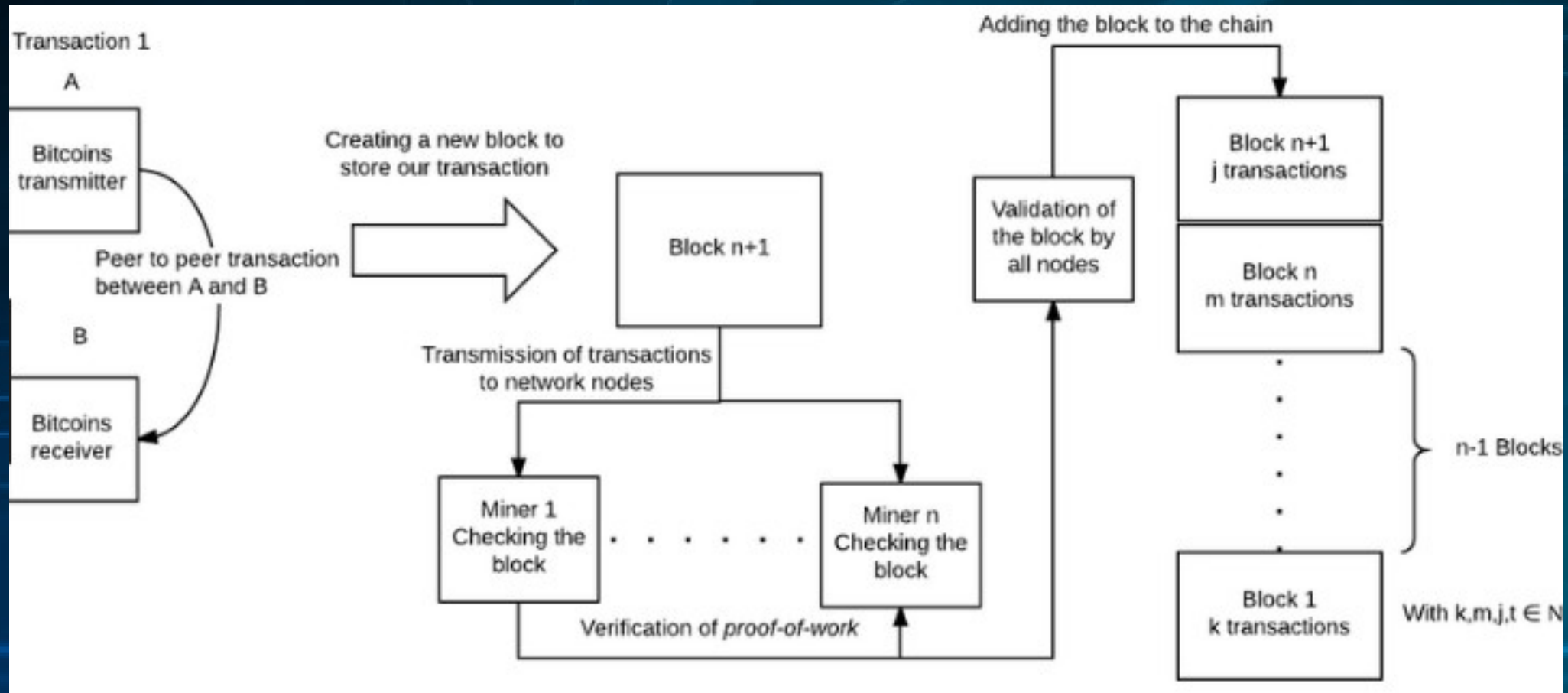
$H_0 = H_0 + A, H_1 = H_1 + B, H_2 = H_2 + C, H_3 = H_3 + D, H_4 = H_4 + E$

- hello hashed is equal to : 2CF24DBA5FB0A30E26E83B2AC5B9E29E1B161E5C1FA7425E73043362938B9824

Hash Function

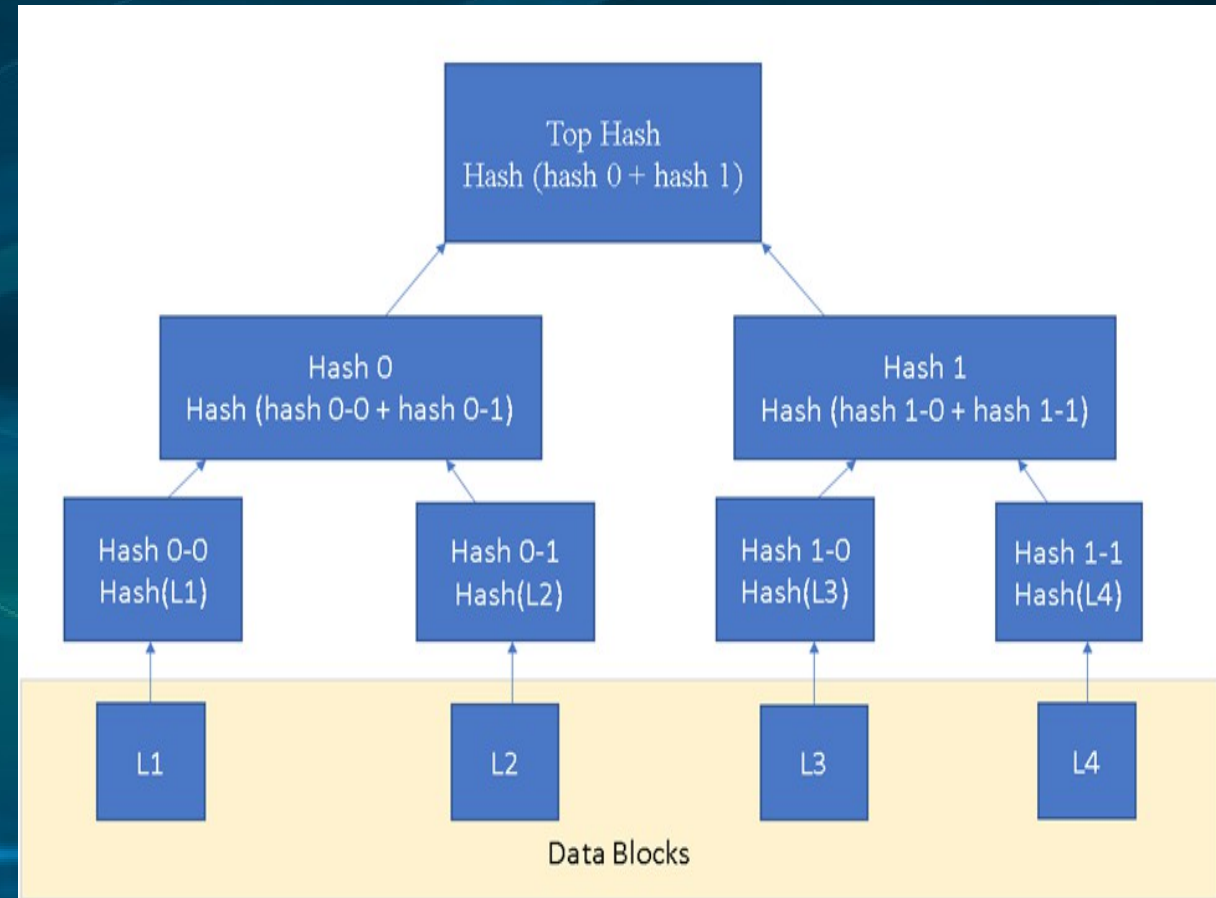
- Verify(Message, 256 bit Signature, pk)
- 2^{256} Possible signature
- Tamper proof
- Miners are incentivized with reward to provide the strongest hash.
- The miner that finds an eligible signature for its block first, broadcasts this block and its signature to all the other miners
- Other miners now verify the signature's legitimacy by taking the string of data of the broadcasted block, and hashing it to see if the output hash indeed matches the included signature.
- If it is valid, the other miners will confirm its validity and agree that the block can be added to the blockchain they reach *consensus*
- <https://passwordsgenerator.net/ha256-hash-generator/>

Blockchain mining



Merkle Tree

- Merkle trees are a fundamental part of what makes blockchains work.
- a way of hashing a large number of "chunks" of data together which relies on splitting the chunks into buckets
- where each bucket contains only a few chunks,
- then taking the hash of each bucket and repeating the same process,
- continuing to do so until the total number of hashes remaining becomes only one: the root hash.



Merkle Proofs in Bitcoin

- Bitcoin blockchain uses Merkle proofs in order to store the transactions in every block
- 80-byte chunks of data for each block that contain only five things
 - A hash of the previous header
 - A timestamp
 - A mining difficulty value
 - A proof of work nonce
 - A root hash for the Merkle tree containing the transactions for that block.

Merkle Proofs in Ethereum

- Every block header in Ethereum contains not just one Merkle tree, but *three* trees for three kinds of objects:
- Transactions
- Receipts (essentially, pieces of data showing the *effect* of each transaction)
- State

Conclusion

Due to Cryptography blockchain ensure the primary goal of information security are fulfilled:

- **Confidentially**
- **Integrity**
- **Availability**

The full encryption of blockchain data ensures that data will not be accessible to unauthorized parties while in transit (so little to no chance for successful man-in-the-middle [MiTM] attacks). Decentralized ledger system helps to protect from distributed denial of service (DDoS) attacks.