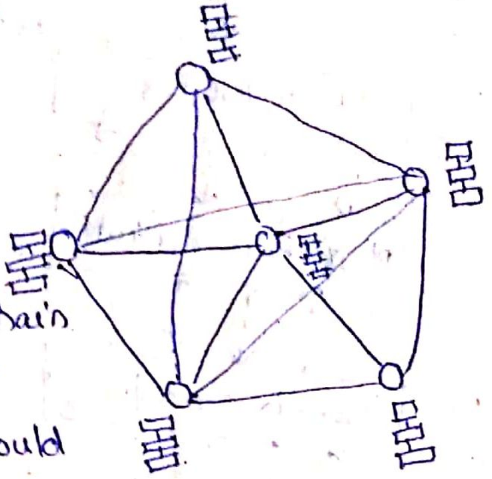
24/07/17 Distributed System.



Ensure the following!

→ Protocol for commitment:
- all valid transactions they should be present in blockchain.

→ Consensus : [Agreement].
- The block at each node should be consistant

→ Security :
- need to provide security ie, if malicious users are there they shouldn't add blocks.

→ Privacy & Authenticity :
- shouldn't reveal the private info, to the public
- before any transaction whether the sender have the ownership to do transaction to y, is to be checked.

$$\begin{tikzcd} x \arrow[r] \arrow[d] & y \\ \text{Owner or (not)} & \end{tikzcd}$$

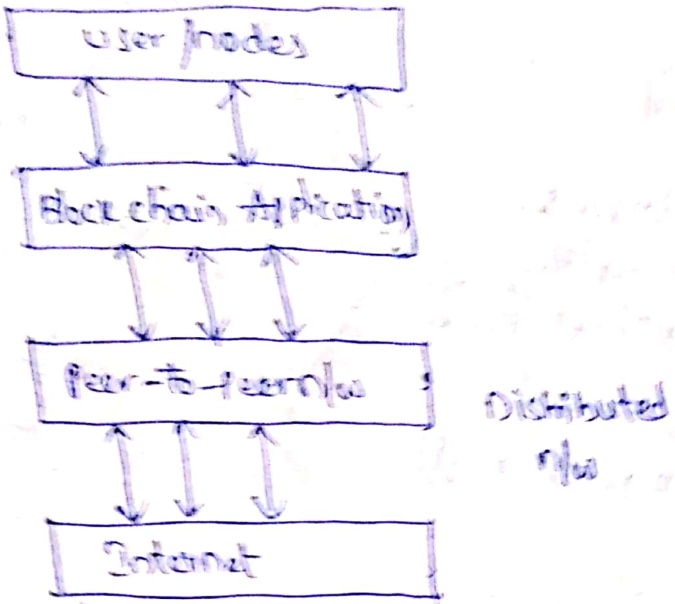
Block Chain:-

- Block chain is a peer-to-peer network, distributed ledger that is cryptographically secure, append-only, immutable and updatable only via consensus.

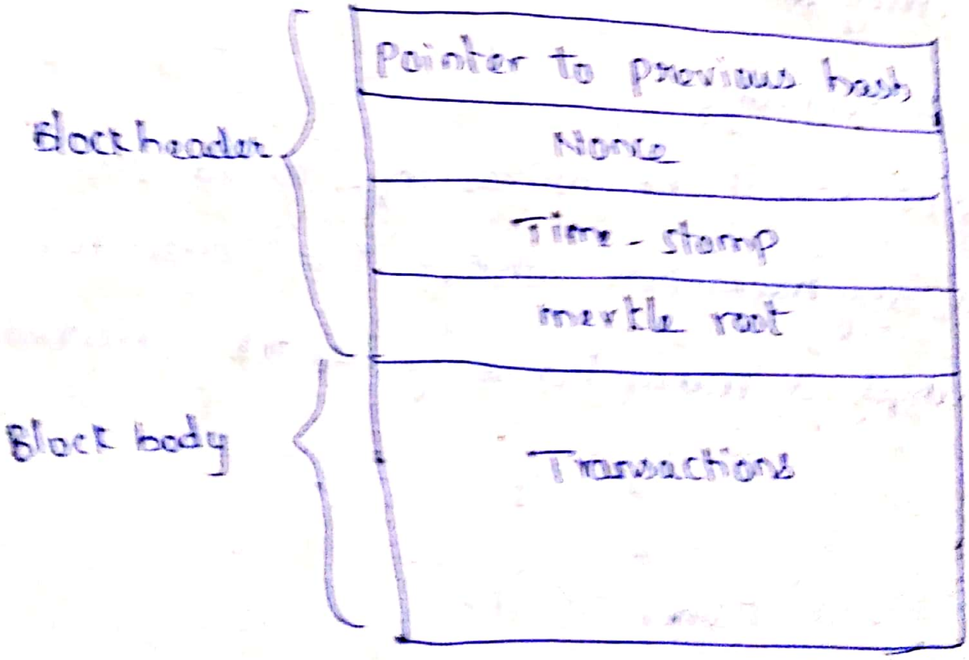
ledger = Record | Book keeping

Scanned by CamScanner

N/w view of a block chain:



→ **Block - structure**



Nonce - is a random number [arbitary num] which will be used only once

Scanned by CamScanner

1 comp... 100 ... x ...

attempt

$n = 121$

$\bar{x} = 97$ hrs

$H_0 : \mu \geq 100$

$H_1 : \mu < 100$

$\bar{x} = 97$ hrs $\quad n = 121$

$\bar{x} - \mu = \frac{97 - 100}{3} = \frac{-x}{\frac{x}{11}}$

Hash function:

A function is any function which can be used to map data of arbitary size onto data of fixed size.

Ex MD5, SHA, SHA256, SHA512 $\qquad$ MD5 – messagedigest 5

double SHA256 is mostly used Hash function.

Properties:–

(i) One-way

(ii) Collision free

(iii) Avalanche Effect

Ex: $x \rightarrow H(x)$ ✓

$H(x) \longrightarrow x$ ✗ not possible

Ex $f(x) = x^2$

$x = 1 \quad f(x) = 1$

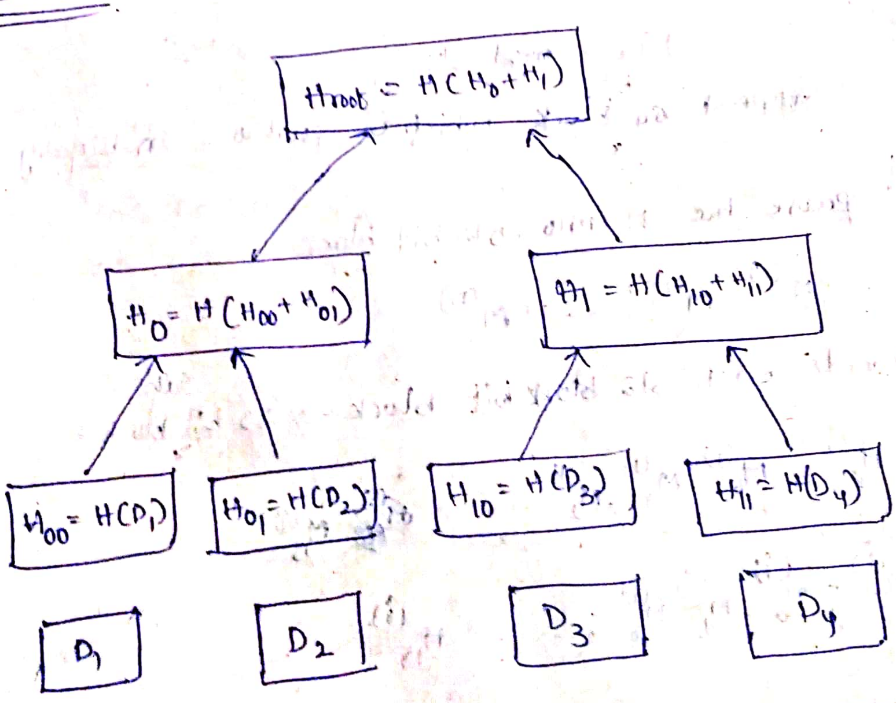
$x = -1, \quad f(x) = 1$

→ "If a minor change in data it will give a great change in hashfunction."

for 2 values are mapping to same value, this collision occurs.

In cryptographically secured function this collision won't occur.

Merkle Tree

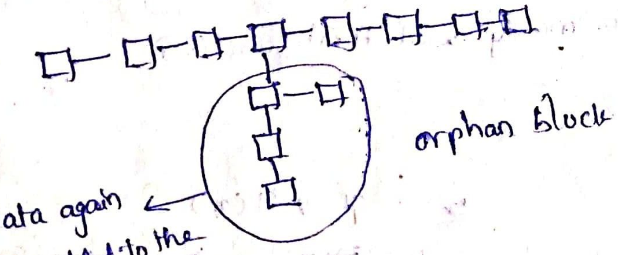


5/8/19

Double-SHA 256 H_k = Hash (H_{k-1} | Transactions | Nonce)

orphan block

Uncl processing



This data again will be added to the block chain.

6/8/19

SHA - Secure Hash algorithm - 256

1. preprocessing.   multiple of 512 bits

Message → M   size = l bits

- Append the bit '1' at the end of the 'M'

- Append k zero (0) bits such that

$k + l + 1 \mod 512$

- Append 64 bit block which is equal to 'l' in binary.

2. parse the M into 512 bit block

$M^{(1)}, M^{(2)} \ldots M^{(n)}$

3. Divide each 512 ~~block~~ bit block → 32 bit sub blocks

$M^{(1)} = M_0^{(1)}, M_1^{(1)}, M_2^{(1)} \ldots M_{15}^{(1)}$

$M^{(i)} = m_0^{(i)}, M_1^{(i)} M_2^{(i)} \ldots M_{15}^{(i)}$

4. Initial Hash $H^{(0)}$ → Which is of 256 bits

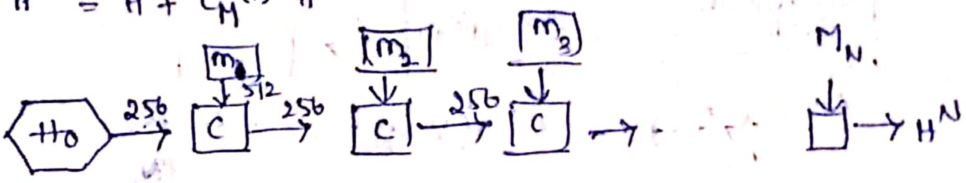
5. $H^{(i)} = H^{(i-1)} + C_{M(i)} H^{(i-1)}$ where i = 1 to N.

&

C_M → compression function.

(i/p.) 512 bits ⟶ 256 bits. (o/p)

$H^1 = H^0 + C_M^{(1)} \cdot H^0$



Bit Coin:

Bitcoin is a decentralized, Permission less, Peer-to-Peer crypto-currency put forth in 2009.

$\frac{\cong 4 \text{ years}}{2{,}10{,}000 \text{ blocks}}$

Operations :-

— Transaction Management

— Money Issuance

Jan 2009	50 BTC
Nov 2012	25 BTC
2016	12.5 BTC

7/8/19

Bit coin: Uses of public key Cryptography

Keys

→ private key

→ public key.

- If we use private key for encryption/decryption then we use public key for decryption/encryption.

Scanned by CamScanner

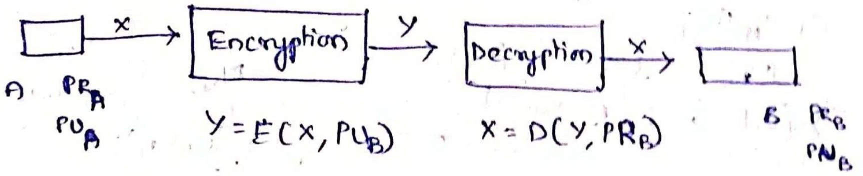
$Z = \frac{\bar{x} - \mu}{\sigma} = \frac{97 - 10}{3} = \frac{3}{11}$

Bit

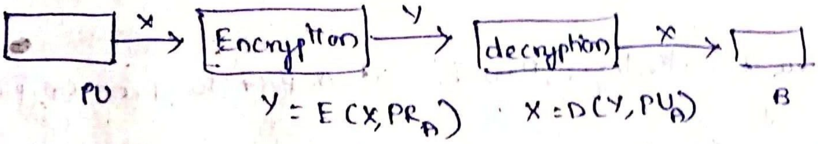
Private key → private kept secret

Advantages

- Confidentiality (Encryption/decryption)
- Authentication



2).



→ Digital Signatures.

ECDSA - Electric Curve Digital Signature Algorithm.

14/8/19

Bit Coin Anonymity:

Address ⇒ Public key + hash 160

These addresses will not provide any info. regarding the people who are present in the transaction.

- In future may be possible, still researches are going on.

Bitcoin Script:

FORTH language — Bitcoin script is almost similar to the 'Forth' language

→ FORTH language.

→ stack

— Reverse polish notation (RPN)

(Postfix notation)

2 * 5 + 10 → Postfix

2 5 * 10 +

20
~~10~~
~~10~~
~~5~~
~~2~~

20 (o/p).

Ex in FORTH language

function (n -- n') DUP 6 < If DROP 5 ELSE 1 – Then.

'n' as i/p.

c-equivalent code

```
function (n) {
    return ((n<6) ? 5 : (n-1));
}
```

DUP ⇒ Duplicate the top of stack

DUP 6 < →postfix

⇒ DUP < 6 →

→ if (top of stack < 6)

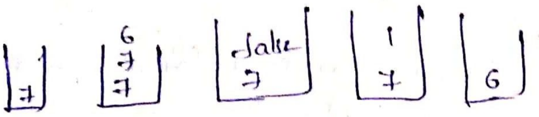
DROP 5 ⇒ return 5

else DUP 1 —

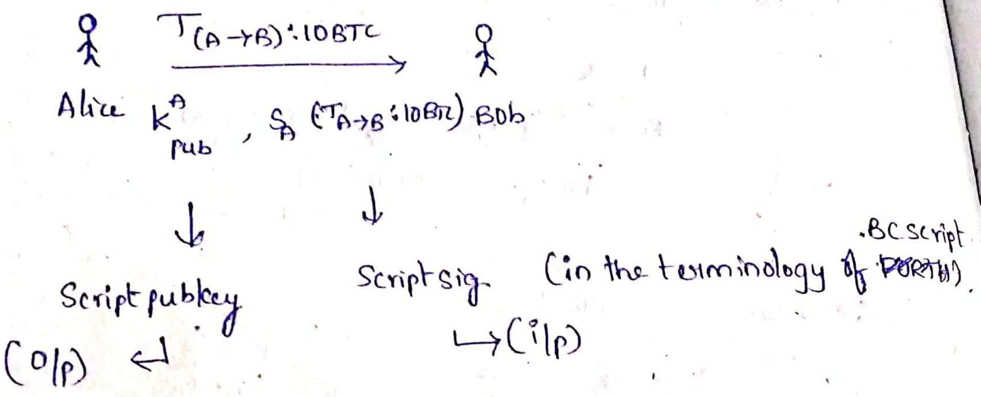
⇒ return (DUP −1)

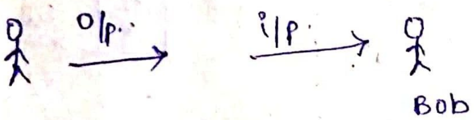
$\bar{x} = 97$ hrs

$$z = \frac{\bar{x}-\mu}{\frac{\sigma}{\sqrt{n}}} = \frac{97-100}{\frac{3}{\sqrt{121}}} = \frac{-\cancel{3}}{\frac{\cancel{3}}{11}} = -11$$



return +6//





Script sig

operands

Transaction (i/p)

18E14A7 B6A - - - - -

D6196.7K6 - - - -

signature of Alice.

Public key ~~of~~ of Alice

operators

Transaction (o/p)

OP_DUP

OP-HASH 160

16UWLLR - - - - - -

OP-EQUALVERIFY

OP-CHECKSIG

← Address of Alice

o/p

Script Pu

i/p

S

o/p

Script Pub key:

OP-DUP OP-HASH160 <pubkeyHash> OP-EQUALVERIFY OP-CHECKSIG

i/p

ScriptSig: <sig> < pubkey>.

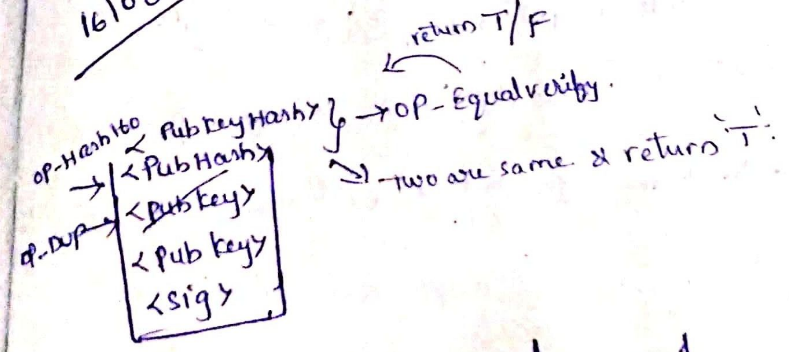
step1: Write i/p 1st & then o/p.

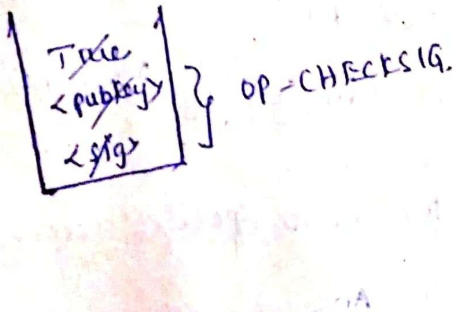
<sig> <pub key> OP-DUP OP-HASH160 <pubkeyHash> OP-EQUALVERIFY OP-CHECKSIG

Actual bitcoin script.

— this is present in each & every transaction of Bitcoin.

16/08/19





Scanned by CamScanner

$\frac{\sigma}{\sqrt{n}}$ $\sqrt{121}$ = -11

$z = -11$

(ii)

Turing Complete:

→ A language which has a capability to solve all types of problems (computation) is called Turing complete.

→ As we don't have loops in the BitCoin script, it is not Turing complete.

- BitCoin script has
  - 256 op codes — operations
  - Arithmetic operations
  - conditional operations (if, then)
  - logical operations
  - Cryptographic operations.
    - Hashfunction.
    - signatures
  - No loops → inorder to get rid of the hours together computation, caused because of ∞ looping, so, to make it finish in a particular time, loops were avoided

Scriptsig: OP_TRUE
scriptpubkey: {empty} } Alicedone transaction.

ie, as there is no specific receiver, so, any one can use the Bitcoins, (or) Any one can uspend the o/p.

(ii) Unspendable Bitcoins:

~~op~~- script Pubkey: OP_RETURN

no signature ie., (no Scriptsig)

=> here ~~and~~ no one can access & spend & use the Bitcoins spent by the sender here.

3). Freezing~~a~~ Bitcoins until a time in future.

Script Pubkey :<expiry time>.

until this expiry time expires, then only one can use the BTC in future. Until the time BTC were freezed.

Script Pubkey : < expiry time> OP_CHECKLOCKTIMEVERIFY OP_DROP OP_DUP OP_HASH 160 <pubkeyhash> OP_EQUALVERIFY OP_CHECKSIG

scriptsig: <sig> <pubkey>

if current time < expiry time OP_DROP the script

if current time > expiry time then OP_DUP, OP_HASH160 ....

script will be executed.

Scanned by CamScanner