1=>Two techniques are predominaly using for securing the chain and for efficient validation and verification:

1=> hashing

2=> asymmetric key encryption

2=> simple symmetric key encryption=> caeser method(the same key used for encryption and decryption)

3=>For the simple symmetric key it is easy to derive the “secret” key from the encrypted data.

symmetric keys have other issues such

(i) key distribution -- how do you send the key to the parties involved

(ii) you need to create different secret key for different receivers, you cannot share the same key with different participants.

4=>On the contrary, in a public-key encryption, you can publish the public key for any participant to use and not reveal the private key.

5=>public-key-cryptography=> private keys are locked up and public keys are published.

6=> encrypting function holds two properties:

(public key , private key) pair has the unique quality that data is encrypted with the private key can be decrypted only with corresponding public key and vice versa.

for ex=> sender's private key- 'b'

sender's public key- 'B'

receiver private key - 'k'

receiver public key- 'K'

1=>firstly sender encrypte the data with its private key

2=>transact to the receiver,then receiver encrypt the data with its public key 3=receiver decrypt this data with its private key

4=>then use sender's public key to decrypt the signed transaction data.

7=>RSA(Rivest Shamir Adelman)algorithm=> password-less user authentication

for ex=> for accessing a virtual matchine in amazon claw

8=>ECC(elliptic curve cryptography)=> used in bitcoin and ethereum : generating the key pair

9=>256 bit ECC key-pair is equivalent in strength to approximately 3072-bit RSA key-pair. Thus ECC is much stronger encryption than RSA method.

10=>hashing=>transforms and maps of arbitrary length of input data value to a unique fixed length value.

input data can be document , tree data or a block data

even a slight difference in the input data will produce the totally different hash output value

11=>Hashing function requirements

1=>it should be a one way function(means no one can derive the original items hash from hash value)

2=>it should be collision free(means hash value uniquely represent the original items hash)

12=>two type of hashing=>

1=>simple hashing

2=>merkle tree structured hashing

13=>Simple hash(all the data items are linearly arranged and hashed) is used when

1=>there is a fixed number of items to be hashed,

such as the items in a block header

2=>we are verifying the composite block integrity

14=> merkle tree structured hash(the data is at the leaf nodes of the tree and this structure helps the

efficency repeated operations)is used, when there is variable number of items to be hashed or when number of items differ from block to block,

such as

a=>transaction hash

b=>state hash

c=>receipt hash

tree root value in a bitcoin blockis calculated using hash of transactions.

15=> In ethereum , hashing is used to generate :

1=>account addresses

2=>digital signatures

3=>transaction hash

4=>state hash

5=>receipt hash

6=>block header hash

16=> Keccak 256 is a commonly used algorithm for hash generation in Ethereum blockchain.

17=>most common hash size is 256-bit hash

1=>sha(secure hash algorithm) 3

2=>sha(secure hash algorithm) 256

3=>keccak 256

18=>256 bit hash value space is too long that is approx 10^77 or 2^256 so it is very hard to generate two of the same hash values of 256 bits

19=> Digital signing of a transaction/document involves

1=> hashing the content of the document

2=> then encrypting it with private key

3=>But receiver can recompute(decrypt by public key of the sender's) the hash of the original data received and compared it with received hash to verify the integrity of the document.

20=> to manage the integrity of transaction

1=>secure and unique account address

2=>autorization of the transaction by the sender through digital signing

3=>verification of the content of the transation is not modified

21=> Account are generated with the help of public and private keys pairs:

1=>256-bit random number is private key

2=>elliptic-curve cryptography algorithm applied to private key to generate public key

3=>hashing applied to public key to get account address(20 bytes too small)

22=> transaction for transferring assets should be:

1=>authorized

2=>non-repudiable

3=>unmodifiable

23=> complete transaction verification nedd to verify:

1=>time-stamp

2=>nonce

3=>account balances

4=>sufficiency of fees

24=> In Ethereum, the block hash is the hash of all the elements in the block header.

25=>purpose of Block hash

1=>it allows for the formation of the chain link by embedding previous block hash in the current block header.

2=> verification of the integrity of the block and the transations.

26=> If a participant node tampers with a block, it results in

1=> hash changing

2=> mismatch of hash values

3=>the local chain of node rendered in an invalid state

27=> main components of ethereum block:

1=> block header

2=> transaction hash

3=> transaction root

4=> state hash

5=> state root

28=> integrity of a block

1=>block header contents not tempered with

2=>transation not tampered with

3=>state transitions are computed, hashed and verified

29=> every state change requires state root(hash) re-computation. instead of recomputing hash for the entire set of states only the effected path in the merkle tree needs to be recomputed.