Decentralization

- Foundation Crypto
  - Public Key Systems
    - RSA
    - ECDSA
  - Hashing
    - Characteristics
    - Algorithms (MD5, SHA-X)
    - Use as PoW
- P2P systems – Napster vs Gnutella
- Blockchain – is this before or after IPFS
  - Bitcoin intro – transaction, blocks, mining, distributed consensus
  - Ethereum – smart contracts
  - ERC-20 Tokens
- IPFS
  - Self-Certified Identities
  - Content-Addressable block store
- ERC-20 Tokens
- Filecoin

Decentralized Computing

- Involves a system where the hardware and software resources are generally *Distributed* across a network
- Location of Control is a key part of Decentralized Computing
  - There should be no central point of control
  - Often you apply a test: If a very powerful entity wanted to shut down the system – could they do it? How easily?
    - Impossible : Highly decentralized
    - Difficult: partially decentralized
- Often involves groups of users contributing resources to be shared by all
  - Content
  - Software
  - Compute capability
  - Storage

Napster

- Classic example in Computing:
- released in 1999 by Shawn Fanning/Sean Parker
- Enabled global (illegal) distribution of copyrighted material (music/video)
- Operated over P2P – so partially decentralized
- Needed Napster.com to initially link consumer with suppliers
- Court Order shutdown the service in 2001
- Followed by systems like Gnutella
- Limewire was a Gnutella client – came under court scrutiny – tried to make their client legal – inserted code into version 5.5.10 and later that allowed them to cut users off from the network -

Decentralized Systems: Pros/Cons/Ethics

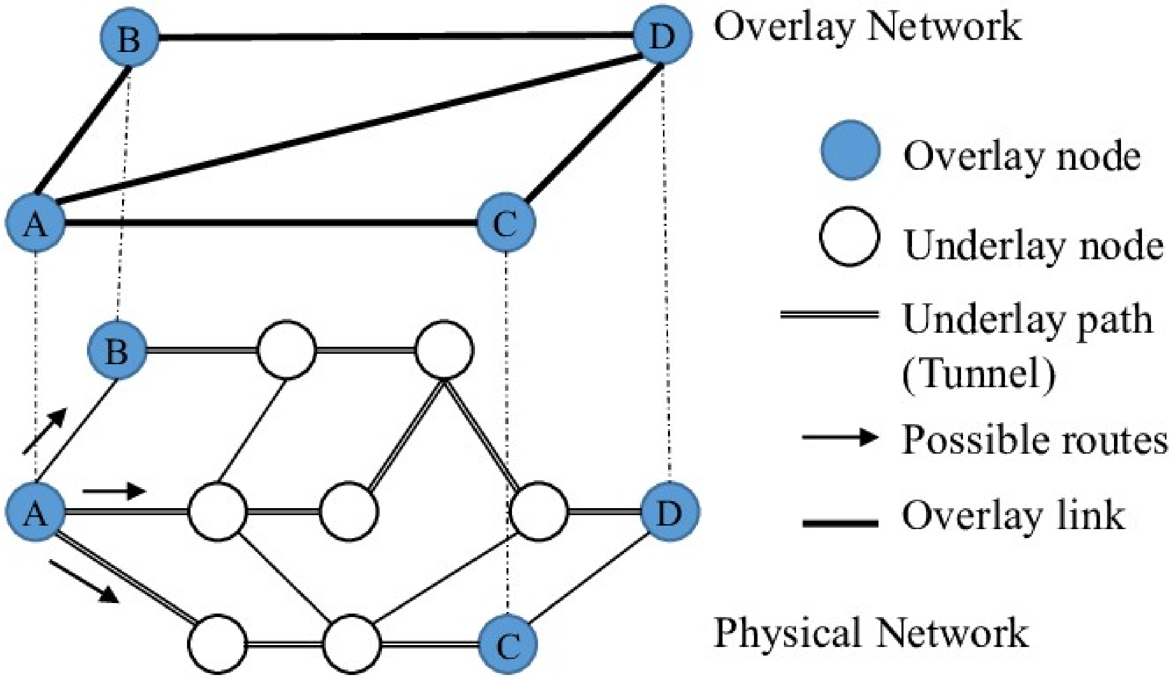
- Some Pros
  - Makes for highly resilient systems than can survive outages
  - Often makes very efficient use of resources
  - Denies special position for the controlling party
  - Censorship Resistance : (Good for freedom fighters etc)
- Some Cons
  - Sometimes very inefficient or resource intensive (at one point P2P traffic with illegal content occupied > 50% of bandwidth on Internet Trunks)
  - Subject to the ‘Tragedy of the Commons’ – no incentives to manage common resource pool wisely
  - Difficult to Incentivize people to work towards the common good
  - Censorship Resistance : (Good for Child Pornographers, Drug dealers, Revenge Porn) – Individuals may not like Government control but societies usually dox

Key Technologies

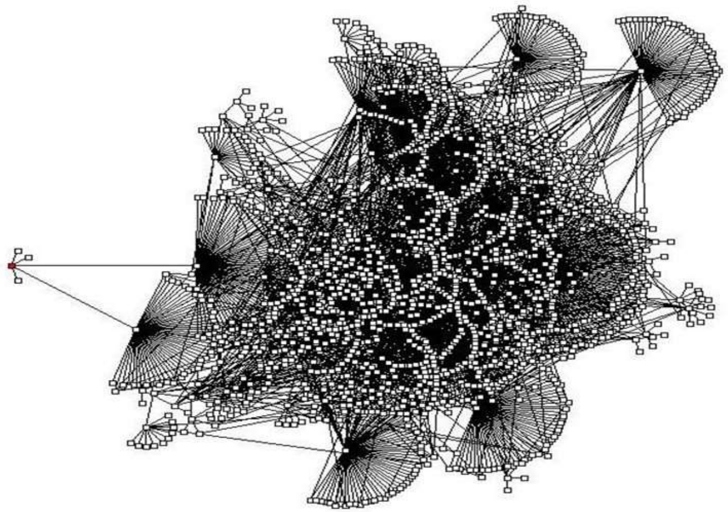
- Peer-to-Peer (P2P) networking to provide transport & discovery
- Some key crypto technologies
  - Hashing
  - Distributed Hash Tables (DHTs)
  - Public/Private Keys
  - Signatures
  - Merkle Trees
- Higher Level abstractions built on these
  - Blockchains
  - IPFS

Peer to Peer Networking - Gnutella

- A node wishing to join the Gnutella Network (somehow) finds some node that is already on it
  - Some peers may be hard-coded into the client; nodes can refuse connection and re-direct
- Leads to the establishment of an Overlay Network – a set of nodes communicating over a mesh of single point-to-point IP connections



Gnutella Primitives

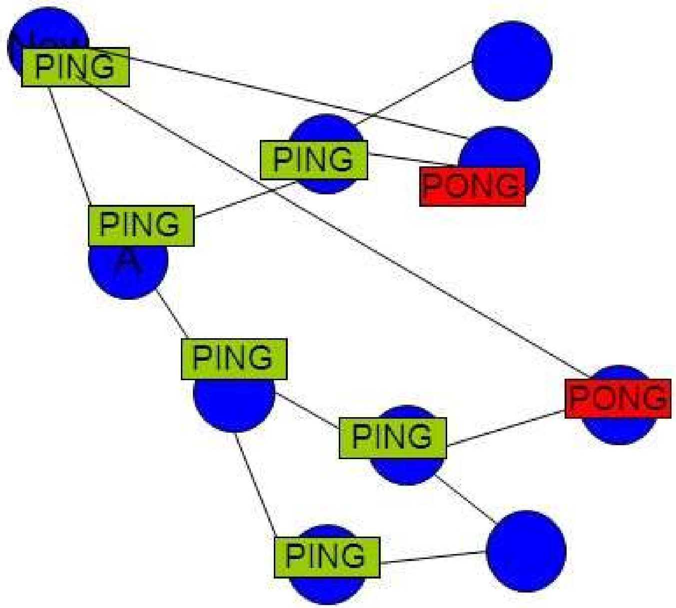


- Used as a distributed search protocol
- Messages
  - Ping – used to discover hosts
  - Pong – response to ping with client address
- Network construction showed on next slide
- Very Resilient
  - Node failure causes neighbours to seek new peers
  - Difficult for authorities to shut down
  - Difficult for network operators to try to manage – even when they are trying to be helpful to it!

Joining Gnutella Network

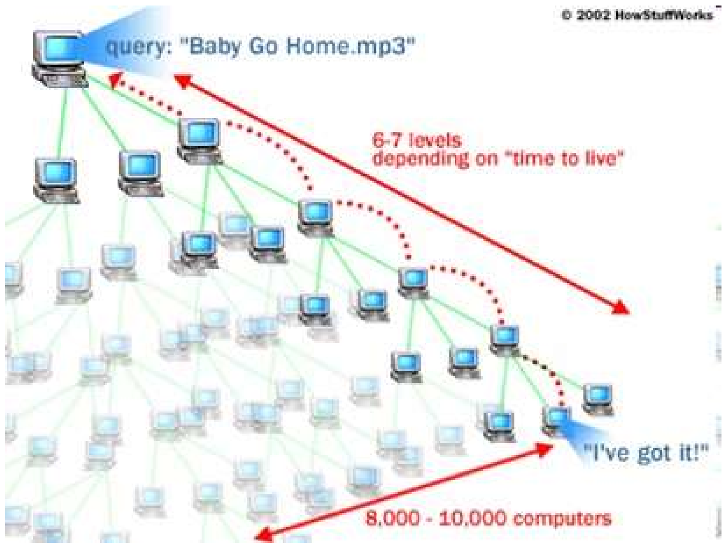
- The new node connects to a well known 'Anchor' node.
- Then sends a PING message to discover other nodes.
- PONG messages are sent in reply from hosts offering new connections with the new node.
- Direct connections are then made to the newly discovered nodes.

Gnutella Network



Searching the P2P network

- Query – search mechanism
- QueryHit – response to query, include info necessary to get data
- Finding content
  - Ping/Query sent to all connected clients
  - Pong/QueryHit sent back along return path
  - TTL mechanism to limit distance
  - File downloads by Http on direct connection



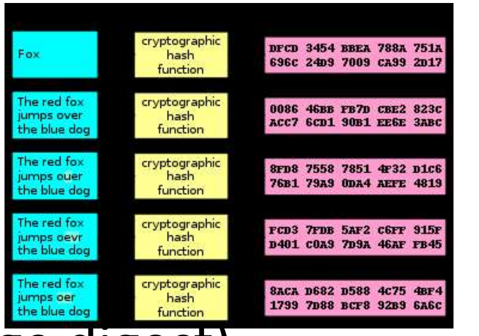
Uses of P2P Networks

- Can be constructed for many purposes
  - Content distribution (Bittorrent)
  - Distribution of transactions (Bitcoin, Ethereum)
  - Storage of information (DHT’s mentioned later)
  - File Sharing (InterPlanetary File System(IPFS), Swarm)
  - Person-to-Person Messaging (Whisper)

Fundamental Technologies – Cryptographic Hash Functions

- Checksums have been used for years to detect errors in blobs of data
- Iterate over the bytes in a blob => fixed size checksum

- Hash functions were invented (sometime in the 1950/60s) along with the concept of a hash table. A hash function
  - Operates on the "Key" field of a hash tables to produce a fixed size result
  - People invent their own hash functions – it is desirable that the hash function 'spreads' well over the range of possible outputs



Cryptographic Hash Functions

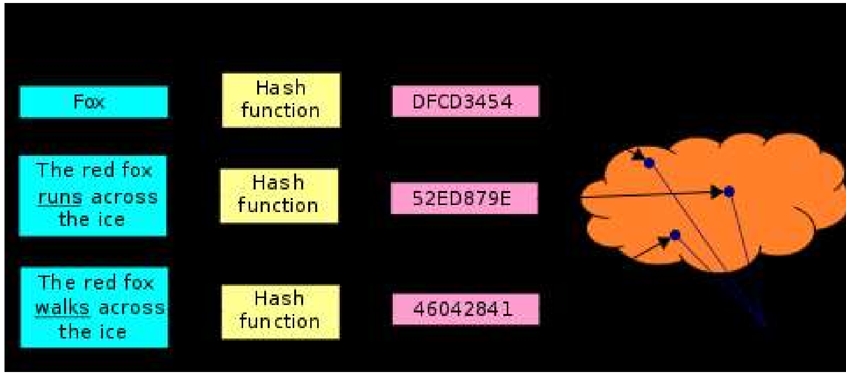
- Iterate over a blob of data to produce a fixed size result (message digest)
  - Deterministic – always produces the same output for the same message
  - Quick to compute
  - Infeasible to generate a message that yields a given hash value
  - Infeasible to find two messages with the same hash value
  - A small change in the message should change the hash value dramatically (avalanche effect)
- Note that a hash of a document (file,block,movie) etc uniquely identifies that document – can be used in a document database to retrieve content.
- Used to prove document integrity – If you can compare to a trusted hash – you know the document has not been altered
- Can be used to commit to things that have not yet been revealed (later revealing a ‘pre-image’ of the hash

Evolution of Hash Functions

- Cryptographic primitives get attacked! – If any flaws are found, replacements are designed
  - Message Digest 5 (MD5) designed by Ron Rivest in 1991 – produces a 128-bit digest – considered broken
  - SHA-1 – US government NIST in 1995 - 160 bit digest – considered broken
  - Sha-2- consists of Sha-256 & SHA-512 – worries about SHA-256, but in widespread usage
  - SHA-3 – US Government NIST in 2015 – based on Keccak primitives

Distributed Hash Tables

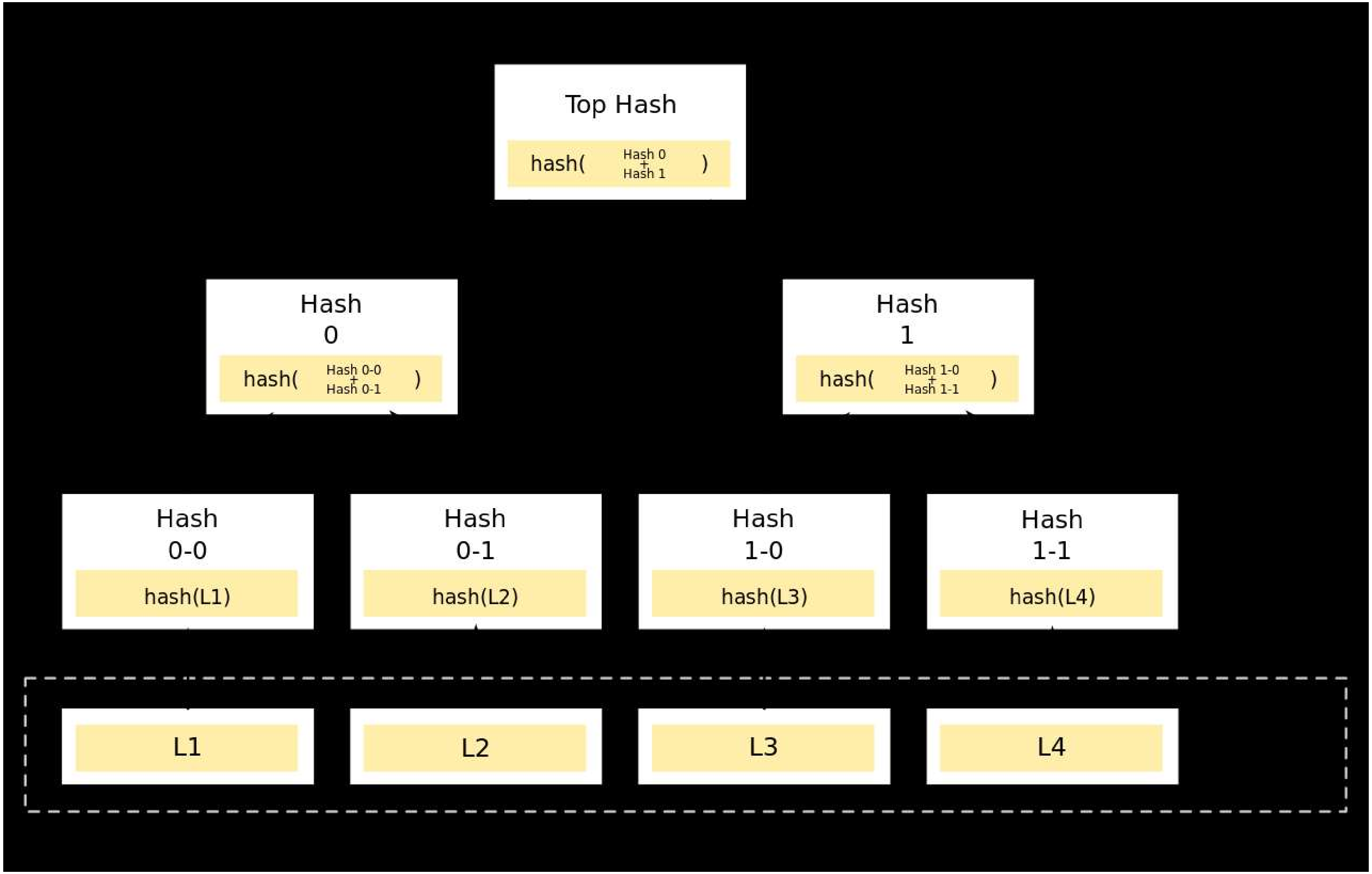
- The nodes on a P2P network can act like the buckets in a Hash Table and store records, documents
- Each P2P node generates a node-id
- Arrange the links in the P2P network so that it is easy to locate a node
- Insert content into the network by storing in the node that has a node id that is 'nearest' to the hash of the document



Merkle Tree

- A tree in which every leaf node is labelled with the hash of a data block
- Every non-leaf is labelled with the hash of the labels of its child nodes
- Diagram on next slide shows a binary hash tree – but can be of arbitrary degree

- Enables entire tree to be verified easily
- Can easily prove that a block belongs to a tree
- If you want to prove that a block is included in a tree – can provide its hash and the Merkle-Path (or hashes) that takes you to the root



By Azaghal - Own work, CC0, https://commons.wikimedia.org/w/index.php?curid=18157888

Proof-of-Work

- Initially proposed in 1997 by Adam Back in his HashCash system
- Trying to replicate the concept of a 'stamp' in Email
- Construct the info contents of stamp: recipient, date, version etc
- Append a random string – Hash the resulting string
- Compute a SHA-1 hash – see if the first 20 bits(5 hex digits) are 0
- If not, increment the string and try again – on avg 2^{20} tries:
- Quick to verify

X-Hashcash: 1:20:1303030600:adam@cypherspace.org::McMybZIhxKXu57jd:ckvi

Recipient, Date, etc | Nonce → Hash: 0x000000000….0f56d3

Difficulty

PoW is the basis for Bitcoin Mining

- Now a huge industry in China, Russia, Iceland and elsewhere
- GPUs and ASICs calculate PoW on each Bitcoin block in order to gain the ‘Block Reward’
- ‘Difficulty’ – number of zeros on hash – is dynamically adjusted

Public/Private Key Systems

- First disclosed (invented?) in 1976 by Diffie & Hellman
- Involves creating two linked keys – one that is made public (PK) and another that is kept secret.
- Either Key works such that
  - Key (Plaintext Message) = Ciphertext Message
    [you can encrypt the message with either key]
- Applying one key reverses the effect of the other
  - PK (Plaintext) = Ciphertext; SK(Ciphertext) = Plaintext
  - SK(Plaintext) = Ciphertext; PK(Ciphertext) = Plaintext
- If PK_A is public, Anyone can read SK_A(Plaintext) but only A could have produced it (signature)
- Anyone can produce PK_A(Plaintext) but only A can read it (Encryption/Enveloping)

Two Commonly Used Systems

- Rivest, Shamir & Adleman (RSA)
  - A key pair generation process yields the 2 related keys
  - Security level determined by key length: 512, 1024, 2048… currently RSA-2048 is recommended for good security in a 30-year time-frame
- Elliptic Curve Cryptography (ECC)
  - First proposed in 1985 but did not enter wide use until 2004/5
  - Generate a private key at random – can derive the public key

Digital Signatures & Identities

- Signing a message is done by applying a HASH function to the message – generating the message digest
- Apply a private key to the Digest to produce a signature

- Users can generate identities just by inventing key-pairs

- Sometimes it is useful to associate a “Name” with a public key
- Can be done on a pair-wise basis – Key Signing party!
- Can also be done with a certificate - Public Key Infrastructure (PKI)

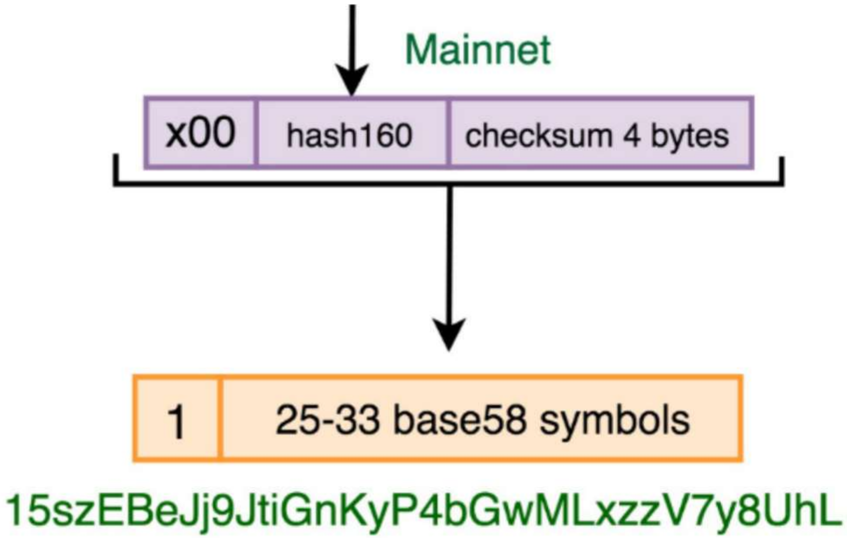
Name: Donal O’Mahony PubKey: 0x567fda3c6 DateofIssue ValidUntil:	Signature of Trusted Entity

Cryptographic Payment - Bitcoin

- Most electronic payments require a Trusted Intermediary (e.g. Bank, CC company, Revolut, Paypay)
- Bitcoin was described in 2008 by Satoshi Nakamoto in his paper "Bitcoin: A Peer-to-Peer Electronic Cash System"
- Set out to create an Internet-wide, decentralized, payment system that did not require any intermediaries

Identities & Transactions

- Bitcoin users come into being when a user generates a key pair
  - The public key is hashed and the lower 160 bits is pre-pended with x00 and a 4-byte checksum added – this is the ‘Bitcoin Address’
  - There is no registration
  - Initially, the user has no Bitcoin, but anyone can send him some if provided with his public key
  - Once the user has acquired some (from a friend or an exchange), he can send it to other bitcoin addresses



Transactions

- Bitcoin transactions involve ‘coins’ that are referred to as Unspent Transaction Outputs (UTXOs)

- Each transaction has references a list of inputs(UTXOs) and a list of outputs

- In the simplest case, Alice has a single UTXO worth 10BTC and wants to send it to Bob

Alice’s 10BTX UTXO Alice’s Signature	Bob’s new UTXO

- When this is written to the Ledger, Bob has a new 10BTC UTXO and Alice’s UTXO is ‘spent’

Transactions….

- Bob can use his new 10BTC UTXO in future transactions by referencing this newly created output
- Under the hood, the signature verification is implemented with Bitcoin Script – this is a set of simple stack-based primitives that are executed by anyone who needs to test the validity of the transaction
- If Alice wants to pay Bob 7 BTC, but she only has a 10BTC UTXO, she can create a transaction with 2 outputs - 7BTC for Bob and another output that sends 3BTC back to herself – she might well create a new address for this!
- The sum of the inputs should (almost) match the sum of the outputs – a small surplus of input over output can be pocketed by the entity that verifies the transaction as a transaction fee

Putting Transactions into Blocks - Mining

- Alice launches her transaction by sending it over the P2P network – where it will be delivered to every Bitcoin node in the network - any of the nodes can engage in mining
- Miners build a pool into which each new transaction is added
- Miners select transactions from this pool and pack them into a potential new block
- They include the block hash of the last block in this new one – so that the blocks link into a block-chain or ledger
- They add a special ‘Coinbase’ transaction which pays them the ‘Block Reward’ of (currently) 12.5BTC
- Then they engage in a Proof-of-Work exercise to find a nonce that generates a block hash with the required difficulty.
- If they find a valid block – they broadcast it to all other nodes
- All miners try to build on top of the longest chain
- The block reward halves every 210,000 blocks (4 years @ 1block/10mins) – started @50 BTC – now 6.25 - goes to zero after 21 million coins produced (in 2140)

Observing the Bitcoin Blockchain

- Blockchain explorers like blockchain.com allow visibility into what is happening on the blockchain
- https://www.blockchain.com/explorer

Bitcoin Limitations

- Bitcoin involves every miner in the world doing (almost) exactly the same computations to achieve a result (the ledger) which represents a “Distributed Consensus” on what the ledger should be – It does this with no controlling middle-man
- As a ledger (database) it is extremely slow and inefficient
- One block is produced every 10 minutes
- With a median to average transaction size – this allows 3.3 to 7 Transactions/second
- Bitcoin script is extremely limited in implementing logic beyond a simple address-to-address transfer