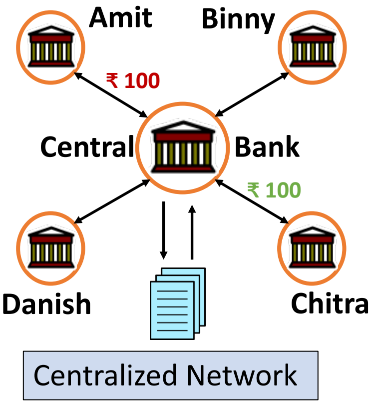
Introduction to the Fundamentals of Blockchain

Subodh Sharma

Subhashis Banerjee

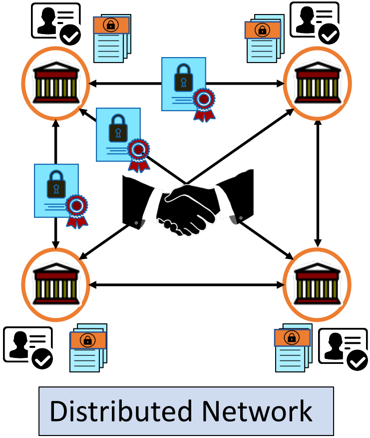
IIT Delhi, Computer Science Department

What is Blockchain?



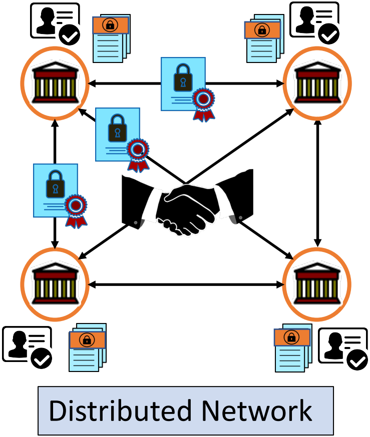
- Central authority oversee the transaction; implicit trust placed on the overseer.
- Maintain a ledger of records, balance information per account etc.

What is Blockchain?



- No central authority; register of transactions and other meta-information is *replicated* and *distributed*.
- Every transaction and account information is visible to everyone
- The records are *untamperable*
- How do we agree on the transaction or the order on them?
- How do we stop double-spending?
- What problems in digital systems can blockchain solve?

What is Blockchain?



- Blockchain is a data-structure that contains an ordered sequence of transaction records and other meta-information
- Each participant in the network can have a fully copy of the blockchain
- The records are chained via *hash pointers*
- All updates to Blockchain via *distributed consensus*

Properties satisfied by Blockchain?

- Replicated storage of the chain makes data *available* even when n/w faults take place

Properties satisfied by Blockchain?

- Replicated storage of the chain makes data *available* even when n/w faults take place
- Every participant having visibility of the entire global state of the chain lends *transparency* and *local verifiability*

Properties satisfied by Blockchain?

- Replicated storage of the chain makes data *available* even when n/w faults take place
- Every participant having visibility of the entire global state of the chain lends *transparency* and *local verifiability*
- Hash chains (through hash pointers) provides *data integrity*

Properties satisfied by Blockchain?

- Replicated storage of the chain makes data *available* even when n/w faults take place
- Every participant having visibility of the entire global state of the chain lends *transparency* and *local verifiability*
- Hash chains (through hash pointers) provides *data integrity*
- Hash chains also provide chronology of data's existence. This gives *traceability*

Properties satisfied by Blockchain?

- Replicated storage of the chain makes data *available* even when n/w faults take place
- Every participant having visibility of the entire global state of the chain lends *transparency* and *local verifiability*
- Hash chains (through hash pointers) provides *data integrity*
- Hash chains also provide chronology of data's existence. This gives *traceability*
- Digitally signed transactions provide *accountability*

Properties satisfied by Blockchain?

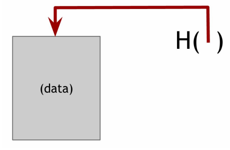
- Replicated storage of the chain makes data *available* even when n/w faults take place
- Every participant having visibility of the entire global state of the chain lends *transparency* and *local verifiability*
- Hash chains (through hash pointers) provides *data integrity*
- Hash chains also provide chronology of data's existence. This gives *traceability*
- Digitally signed transactions provide *accountability*
- *Distributed consensus* provides trust and, consequently, reliability

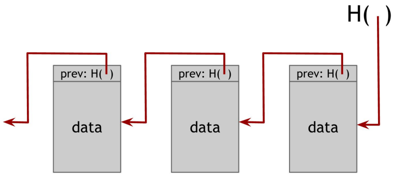
Components in Blockchain: CHF

Cryptographic Hash Functions (CHF) have three properties:

- Collision-resistance: Infeasible to find two values x and y s.t. $x \neq y$, yet $H(x) = H(y)$.
- Hiding: Given $y = H(x)$, there is no feasible way to figure out the value of x. If x is not drawn from a domain that is *spread out*, then choose a secret value r from a probability distribution that has *high min-entropy* s.t. the hiding property holds for $H(r||x)$.
  - Application in creating **commitment**.
- Puzzle-friendliness : If for every possible $n-$bit output value y, if k is chosen from a distribution with high min-entropy, then it is infeasible to find x such that $H(k||x) = y$ in time significantly less than 2^n.
  - Application in **search puzzles**. Given a nonce n, the hash function H, and the output target set Y find x s.t. $H(n||x) \in Y$

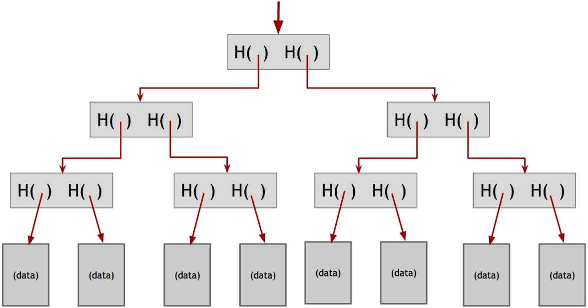
Components in Blockchain: Hash Chains





- HashPtr:Simply a pointer to where the information is stored together with the hash of the information.
- HashChain: List of HashPtrs. As long as the head ptr is stored securely (i.e. an adversary can't access it), we will have a **tamper-evident** log.

Components in Blockchain: Merkle Trees



- Hashptrs organised in a binary tree
- Property: provides concise proof of *membership*

Components in Blockchain: Digitial Signatures

- Properties: Valid signatures must verify and infeasible to forge signatures
- Public keys as digital identities: decentralized identity management
- Consequence: one can make many identities

Components in Blockchain: Distributed Consensus

The consensus Problem

- **Agreement:** All honest processes must agree on the *same* value
- **Validity:** If all the honest processes have the same initial value, then the agreed upon value must all be that same value
- **Termination:** Every honest process must *eventually* decide on a value.

Results from Distributed Computing

Failure Mode	Synchronous System	Asynchronous system
No failure	Agreement	Agreement
Crash failure	Agreement $f < n$	No agreement
Byzantine failure	Agreement $f \leq n/3$	No agreement

Table: Results on Agreement. n is the total number of processes and f is the number of failure-prone processes.

- Impossibility of distributed consensus with one faulty processor. M. Fischer, N. Lynch, M. Paterson. *Journal of ACM*, 1985.
- Consensus in the presence of partial synchrony. C. Dwork and N. Lynch. *Journal of ACM*, 1988.

Proof of Work

- Rely on **hash puzzle**.
- The node proposing a block is required to find a number, or nonce, s.t. $H(nonce||prev_hash||tx_1 \cdots tx_n) < tgt$
- Target space is quite small in comparison to the output space of the hash function H.
- Fixed protocol to assign the target space

Many other consensus protocols: PBFT, Proof-of-stake, Algorand (cryptographic sortition), Hashgraph, etc.

Putting it all together

Simplified protocol

- New transaction are bcast to all the nodes; each node selects and collects transactions into a block.
- In each round a random node (vsi proof-of-XYZ) is chosen who gets to bcast its block.
- Other nodes decide on the block (accept: if all transactions in it are valid)
- Implicit acceptance: node express acceptance by attaching the block in their local copies of the chain and including the hash of the accepted block in the next block they propose.

Addressing Denial of Service Attack

- Say A dislikes B and decides to not include any transaction from B in any block that she proposes.
- B's transactions may genuinely not get included in a block in a round where A is proposing.
- However due to random node selection, B's transactions will get eventually added to a block.

Can Blockchain solve the privacy problem?

- While blockchains can support data minimisation, can they support purpose limitation?
- How is distributed yet regulated access control will be implemented in Blockchain?
- While distributed consensus may ensure safety, can it guarantee no private information leaks through insider attacks?
- How are private keys secured from privileged software?

References

- Bitcoin and Cryptocurrency Technologies. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder.