**Day 1: Progress Report – Blockchain Learning**

Completed 4 lectures covering the following core topics:

* Centralized, decentralized, and distributed systems
* Definition and structure of blockchain
* Hash functions, digests, and Merkle trees
* Miners and smart contracts
* Block structure and transaction data
* Block hashes and their significance
* Permissioned vs. permissionless blockchains
* Challenge–response in consensus mechanisms
* Provenance tracking in data
* IPFS (Inter Planetary File System) and the distributed web

**Day 2: Progress Report – Blockchain Learning**

**Lectures Covered:** Lecture 5 to Lecture 9 (5 lectures)

**Focus:** Cryptographic foundations and Bitcoin fundamentals

**Core Topics Learned:**

**🔐 Cryptographic Foundations**

* Cryptographic Hash Functions  
  • Properties: Collision Resistance, Pre-image Resistance, Avalanche Effect, Puzzle-Friendliness  
  • Introduction to the SHA-256 algorithm\*
* Hash Pointers and Hashchains  
  • Concept of hash pointers as secure references to previous data  
  • Use of hashchains in ensuring data integrity

**🌲 Merkle Trees**

* Structure and purpose in efficient data verification
* Use in blockchain for verifying large transaction sets efficiently

**✍️ Digital Signatures & Public Key Cryptography**

* Purpose of digital signatures in verifying authenticity
* Public and Private Key Mechanism  
  • Public Key (used for encryption)  
  • Private Key (used for decryption)
* RSA Algorithm: Basics of public key encryption\*
* Digital Signing Workflow  
  • Signing with private key, verification with public key  
  • Efficiency via signing the hash (digest) instead of full message

**💰 Cryptocurrency Concepts**

* Introduction to Cryptocurrency secured by hashchains and digital signatures
* **Bitcoin Basics**  
  • Coin creation and supply  
  • Sending payments  
  • Double spending problem and blockchain’s solution  
  • Anonymity in Bitcoin

**📜 Bitcoin Script**

* Introduction to Bitcoin’s scripting language (similar to Forth)\*  
  • Stack-based, uses postfix notation  
  • Common instructions and examples of real-world Bitcoin scripts

**🌐 Bitcoin P2P Network**

* Joining the Bitcoin peer-to-peer network
* Transaction broadcast (flooding mechanism)
* Mining and block generation  
  • Mining puzzle and Proof-of-Work  
  • Block flooding and propagation  
  • Consensus via longest valid chain  
  • Handling latency in block propagation

**Day 3 Progress Report – Blockchain Learning**

Lecture covered: 10 to 13 (4 lectures)

1. Consensus
2. why Consensus
3. example
4. why it can be difficult in certain scenarios
5. Distributed consensus
   1. Faults in distributed systems
      1. Crash fault
      2. Network/portioned fault
      3. Byzantine fault
   2. Properties
      1. Termination
      2. Validity
      3. Integrity
      4. Agreement
6. Synchronous vs Asynchronous system
   1. Asynchronous system
      1. FLP85
   2. Synchronous system
      1. Paxos
      2. Raft
      3. Byzantine fault tolerance (BFT)
7. Correctness of distributed consensus protocol
   1. Safety
   2. Liveliness
8. Consensus in an open system
   1. Traditional distributed consensus protocol
      1. Message passing
      2. Shared memory
9. Consensus in a bitcoin network
10. Consensus algorithm in bitcoin
    1. PoW
       1. Cryptographic hash as PoW
       2. Hashcash PoW
11. Bitcoin Proof of Work
    1. Solving the double spending problem
12. Sybil attack
13. DOS (denial of services) attack
14. Breaking bitcoin PoW
    1. The monopoly problem
    2. PoW power consumption
15. Handling monopoly and power consumption – proof of stake(PoS)
16. PoS (Proof of stake)
    1. Variant of stake
17. PoB (proof of burn)
18. Difference between PoW, PoS and PoB
19. PoET (Proof of elapsed time) {intel}
    1. PoET over trusted environments
20. Mining bitcoins
    1. The life of a miner
    2. Mining difficulty
    3. Setting the difficulty
    4. Hash rate vs difficulty
    5. Mining hardware
    6. Mining pool methods
       1. Pay per share (PPS)
       2. Proportional
       3. Pay per last N share (PPLNS)
       4. Pros and cons

**Day 4 Progress Report – Blockchain Learning**

Lectures: 14 to 17

1. Permissioned model
   1. Use cases
      1. Provenance tracking of assets
2. Smart contracts
   1. Design limitations
      1. Sequential execution
      2. Non-deterministic execution
      3. Execution on all nodes
3. Do we really need to execute contracts at each node?
   1. What if the node that executes the contracts is faulty?
   2. Use state machine replication
4. State machine replication
5. State machine
6. Smart contract state machine – crowd-funding
7. Distributed state machine replication
   1. Place copies of the state machine on multiple independent server
   2. Receive client requests, as an input to the state machine
   3. Propagate the inputs to all the servers
   4. Execute the inputs based on the order decided, individually at each server
   5. Sync the state machines across the servers, to avoid any failure
   6. If output state is produced, inform the clients about the output
8. Permissioned blockchain and state machine replication
9. Why distributed consensus
10. Faults in distributed consensus
    1. Crash fault
    2. Network or partitioned faults
    3. Byzantine faults
       1. Malicious behaviour in nodes
       2. Hardware fault
       3. Software error
11. Consensus for three processes
12. Requirement of a consensus algorithm
    1. Termination
    2. Agreement
    3. Integrity
13. Different consensus algorithms
    1. Crash or network faults
       1. PAXOS
       2. RAFT
    2. Byzantine faults (including crash or network failures)
       1. Byzantine fault tolerance (BFT)
       2. Practical byzantine fault tolerance (PBFT)
14. PAXOS
    1. Types of nodes
       1. Proposer
       2. Acceptor
       3. Learner
    2. Making a proposal:
       1. proposer process
       2. Acceptor’s decision making
       3. Acceptor’s message
    3. Accepting a value:
       1. Proposer’s decision making
       2. Accept message
       3. Notifying learner
    4. Single proposer: No rejection
15. Handling failure:
    1. Proposer failure
    2. Dueling proposers
16. RAFT consensus
17. RAFT
18. Electing the leader:
    1. voting request
    2. Follow node’s decision making
    3. Majority voting
19. Multiple leader candidate:
    1. Current leader failure
    2. Simultaneous request vote
20. Committing entry log
21. Handling failure
22. Byzantine general problem
23. Practical byzantine fault tolerant
    1. Practical byzantine fault tolerant model
       1. Assumptions (asynchronous network, faulty nodes)
       2. 3f+1 nodes req. to tolerate f byzantine faults
    2. Practical byzantine fault tolerant algorithm
       1. Three phase protocol
          * Pre-prepare phase
          * Prepare phase (2f) (excluding its own)
          * Commit phase (2f+1) (including its own)
       2. View change protocol
          * Handle primary failure
          * Ensures progress despite faulty/malicious leader/primary
    3. Correctness
       1. Safety
       2. Liveness
24. Consensus in permissioned model

**Day 5 Progress Report – Blockchain Learning**

Lecture covered (18 -24)

1. Blockchain defined
   1. Problem; difficult to track asset transfers in a business network
   2. Solution; shared, replicated, permissioned ledger
      1. Consensus, provenance, immutability and finality
2. Key concepts and benefits of blockchain for business
   1. Shared ledger
   2. Smart contracts
   3. Security
   4. Consensus
   5. Reduced time
   6. Removes cost
   7. Reduces risk
   8. Enables new business model (IoT integration into supply chain)
3. Degree of centralization
   1. From 100% centralized to 100% decentralized in a gradient manner
4. Permissionless vs permissioned blockchain
   1. Access
   2. Scale
   3. Consensus
   4. Identity
   5. Asset
5. The Linux foundation: Hyperledger project
6. Hyperledger fabric: distributed ledger platform
7. Hyperledger composer: accelerating time to value
8. Actors in a blockchain solution
   1. Regulator
   2. B2B transaction
   3. Membership services
   4. Traditional data resources
   5. Traditional processing platforms
   6. Blockchain network operator
   7. Blockchain developer
   8. Blockchain architect
9. Component in a blockchain solution
   1. Ledger
   2. Smart contract
   3. Peer network
   4. Membership
   5. Events
   6. System management
   7. Wallet
   8. System integration
10. Ledger component
    1. Blockchain
       1. A linked list of blocks (hashchain)
       2. Each block describes a set of transactions
       3. Immutable – blocks can’t be tempered
    2. World state
       1. Stores the most recent state of smart contracts/output of transations
       2. Stored in a traditional database (e.g. key value store)
       3. Data elements can be added, modified, deleted, all recorded as transactions on blockchain
11. Block detail (simplified)
    1. The first block known as genesis block
    2. Block contents (explained)
12. Ledger example: A change of ownership transaction
13. How applications interact with ledger
14. Blockchain events
15. Integrating with existing systems – possibilities
16. Byzantine general problem (was supposed to be cover above but dues to series in playlist, got misplaced)
    1. Three generals problem
       1. Lieutenant faulty
       2. Commander faulty
    2. Four generals problem
       1. Lieutenant faulty
       2. Commander faulty
    3. Byzantine generals model
    4. Lamport-shostak-pease algorithm
17. Hyperledger fabric V1 architecture
    1. External CA, membership services, fabric CA
    2. Client application / SDK (HFC)
    3. Ordering service
    4. Peer
       1. Endorser
       2. Committer
       3. Ledger
       4. Chaincode
       5. Events
    5. Admin etc
    6. Nodes and roles
       1. Committing peer
       2. Endorsing peer
       3. Ordering node
    7. Transaction Flow
       1. Endorse > order > validate
    8. Transaction flow in 7 steps
       1. Propose transaction
       2. Execute proposed transaction
       3. Propose response
       4. Order transaction
       5. Deliver transaction
       6. Validate transaction
       7. Notify transaction
    9. Key benefit of transaction flow
18. Ordering services
    1. Configuration options
       1. SOLO
          1. Single node for development
       2. Kafka: crash fault tolerant consensus (Deprecated)
          1. 3 nodes minimum
          2. Odd numbers of nodes recommended
19. Channels
    1. Single channel network
    2. Multi-channel network
20. Fabric peer
21. Client application
22. Fabric certificate authority
23. Organisations
    1. Each organisation defines:
       1. Membership services provider (MSP) for identities
       2. Admin (s)
       3. Users
       4. Peers
       5. Orderers (optional)
    2. A network can include many organisations representing a consortium
    3. Each organisation has an ID
24. Consortium network
    1. An example consortium network of 3 organisations
       1. Org 1 and 3 run peers
       2. Org 2 provides the ordering services only
25. MSP overview
26. Transport Layer Security (TLS)
27. User Identities
28. Admin identities
29. Peer and orderer identities
30. Channel MSP information
31. New user registration and enrolment
32. Transaction signing
33. Hyperledger fabric network setup
    1. Configure and start ordering service
    2. Configure and start peer node
    3. Install chaincode
    4. Create channels
    5. Join channels
    6. Instantiate chaincode in channel
34. Endorsement policies
    1. Endorsement system chaincode
    2. Validation system chaincode
    3. Endorsement policy syntax
    4. Endorsement policy example

**Day 6 Progress Report – Blockchain Learning**

Lecture (25 to 31)

1. Fabric demo on IBM blockchain cloud

* Covering lecture 25, 27 and 28
* (tried to open the site and perform the demo by myself but I guess the site has removed the functionality as per today’s time)
* I am not sure but I guess there are some changes occurred from the time when lecture took place and today. Which was causing some disturbances while checking out with the things.

1. Hyperledger fabric explainer

* Covering lecture 26
* An informative video of around 2 and half minute for understanding of Hyperledger fabric.

**Hyperledger composer (lecture 29, 31) with one optional video (lect. 30)**

1. Hyperledger composer: accelerating time to value
   1. A suite of high-level application abstractions for business network
   2. Emphasis on business centric vocabulary
   3. Reduce risk, and increase understanding and flexibility
2. Features
   1. Model your business networks, test and expose via APIs
   2. Applications invoke transactions to interact with business network
   3. Integrate existing systems of record
3. FULLY OPEN AND PART OF LINUX FOUNDATION HYPERLEDGER {not as per today}
4. Goal for Hyperledger composer
5. Extensive, familiar, open development toolset
   1. Data modelling
   2. Javascript business logic
   3. Web playground
   4. Client libraries
   5. Editor support
      1. Atom
      2. Vscode
   6. CLI utilities
   7. Code generation
   8. Existing systems and data
6. User role in blockchain solution {a single organization may play multi roles
   1. Network service provider
   2. Network service consumer
   3. Business service provider
   4. Business service consumer
   5. End user
7. Key concept for the business service provider
8. Example: vehicle auction developer
9. Business service provider develops three components
   1. Smart contracts
   2. Business logic
   3. Presentation logic
10. Key development concepts
    1. Model files
    2. Access control list
    3. Transaction processor
    4. Business network definition
11. Assets, participants and transactions
12. Access control
13. Event and queries
14. Smart contract development: composer playground
15. General purpose development: visual studio tool
16. Creating the business and end user applications
17. Debugging
18. Network administration
19. Two roles with “admin” responsibilities
    1. Network service consumer
    2. Business service consumer
20. Key concept for administrators
21. Network service consumer packages resources in a BNA file
22. Connection profiles to Hyperledger fabric
23. Participant identity
24. Business network cards
25. System of record integration
26. Exploiting loopback: examples
27. How composer maps to fabric chaincode
28. Hyperledger composer outlook
29. Get started with hyperleger composer

**Day 7 Progress Report – Blockchain Learning**

Lecture: 32 to 37

**Blockchain use cases**

1. Sample use cases by industry
   1. Financial services
   2. Public sector
   3. Retail
   4. Insurance
   5. Supply chain & logistic
2. What makes a good blockchain use case?
   1. A business problem to be solved with efficiency than it’s alternatives
   2. An identifiable business network
   3. A need of trust
3. Understand the business problem
   1. Specific business problem/ challenges
      1. Scope the business challenge up front
   2. It’s current solution
   3. What specific aspects of this business problem will be addressed (assuming the business problem is large)
4. Understanding the participants
   1. Who are the business network participants/ their roles
   2. Who are the specific people within the organization and what are their job roles?
5. Understanding the participants in details (refer to notes)
6. Identities
7. Understanding the assets and transactions
   1. What assets are involved and its associated key info
   2. What transactions are involved and its associated assets
8. Defining transactions
9. Additional point of understanding (refer to notes)
10. Addressing business value
11. Building communities in blockchain networks
    1. Consortium based network
    2. Founder directed network
    3. Community based network

**Payments and secure trading**

1. Cross border payments
2. Stellar protocol and network
3. Ripple protocol and network
4. Permissioned network for payments and settlement
5. Project UBIN: SGD on distributed ledger
   1. Project Ubin phase 1: technical architecture
   2. Project ubin phase 2
   3. Project ubin phase 2 decentralized netting
   4. Project ubin: phase 2 (refer to notes to differentiate between b and d)
6. Blockchain for commercial paper
7. Components, processes in securities trading
8. Securities T+3 trade life cycle
9. Security settlement in low liquidity markets
10. Private equity administration

**Blockchain in finance services: compliance and mortgage**

1. Compliance (KYC, AML)
2. Know your customer (KYC)
3. Shared KYC solution
4. Information sharing
5. Privacy and consent
6. Mortgage processes
7. Example mortgage origination process
8. Syndicated loans

**Blockchain in financial services: financial trade**

1. Overview of international trade
2. Blockchain enabled future state
3. Trade finance elaborated
4. Trade finance: advantages using blockchain
5. We.trade: trade finance network
6. Supply chain financing – as – is
7. Blockchain solution roles/responsibilities

**Day 8 Progress Report – Blockchain Learning**

Lecture covered: 38 to 44

**Revolutionizing global trade**

1. IBM blockchain for trade logistics
   1. How products wait at the terminal even after reaching, due to documentation
   2. Import/export cost spent on air courier for customs paperwork
   3. Potential savings using electronic documents
2. Members involved
   1. Exporter’s bank
   2. Exporter
   3. Export authority
   4. Port of loading
   5. Port of entry
   6. Import customs
   7. Importer
   8. Importer’s bank
3. Cargo path
4. Document path
5. Event recording and history
6. Blockchain node
   1. Secure data exchanges
   2. Tamper-proof documents and e-signs
   3. Digital and automated workflows
   4. Real time visibility & analytics
7. Logistic data challenge
   1. Error-prone info
   2. Incomplete/inconsistent info
   3. Border delays
   4. Lack of shipment visibility
8. Logistic data challenge - Root causes
   1. Multiple data formats
   2. Too many p2p interactions
   3. Absence of messaging standard
9. Key industry challenges
   1. Banks
   2. Importers & exporters
   3. Carriers
   4. Forwarders
   5. Ports
   6. Authorities
10. Global trade digitization (GTD)
    1. What
    2. How
    3. benefits
11. Global trade digitization (GTD)
    1. Paperless trade
    2. Shared visibility
12. GTD documents and events
13. Other issues:
    1. empty container repositioning
       1. Blockchain for container management
    2. Port operations
       1. Blockchain for port operations

**Blockchain in supply chain-1**

1. Food safety
   1. Consumers scan or input the tracing code printed on the product package
2. food traceability and safety enabled by blockchain
   1. blockchain capabilities
      1. asset identification ownership and transfer
      2. asset aggregation and disaggregation
      3. geo specific compliance
      4. traceability, reduce food recalls
   2. cognitive and IoT capabilities
      1. location, temperature, small computer
      2. anomaly detection
      3. real-time analytics and optimization
3. FDA food safety modernization Act
   1. foreign supplier verification program (FSVP)
      1. formulation
      2. transport
      3. packaging
      4. storage
      5. came into effect in the us in may 2017
   2. supply chain visibility
      1. what
      2. how
4. noteworthy startups
   1. provenance:
   2. skuchain
5. supply chain orchestration

**blockchain in supply chain -2**

1. diamond provenance
2. everledger
3. The diamond lifecycle
4. Supply chain compliance
5. Addressing supply chain fraud

**Blockchain in other industries**

1. Blockchain for healthcare: use cases
   1. Patient consent and health data exchange
   2. Payment and claims
   3. Pharma supply chain provenance and traceability
   4. Rethinking clinical trial management
2. Patient mediated health data exchange
3. Notable startup: guardTime
4. Loyalty points exchange
5. Notable startup: https://loyyal.com
6. Blockchain in energy market: gridChain
7. Renewable energy flexibility
   1. What
   2. How
   3. Benefit
8. Media
   1. Jo music (Ethereum based)
   2. Licensing + payments = Ethereum

**Day 9 Progress Report – Blockchain Learning**

Blockchain in government:

1. Blockchain and government
   1. Government need to maintain
      1. Daily operations/activities
      2. Gov. assets
      3. Details of people, organizations and institutions
      4. Records of people
      5. Business transactions
2. Multi institutional or multi-organization
   1. Different level of governance
      1. Village, panchayat, citites
      2. District
      3. State
      4. Country
   2. Every level builds its own ledger of data
      1. Different access management policies
      2. Role based access control or access management
   3. Different priority of data
      1. High priority or highly secured data – restricted access – need prevent from unauthorized access( e.g. aadhaar data)
3. Blockchain can help in management of gov data at different levels
   1. The block can contain huge amount of data
   2. The data can not be altered without colluding majority of the blocks
   3. Data access as transactions – can check or verify who has accessed what
4. Government and cyber crime
   1. Gov database is a major target for hackers
   2. Cyber war: actions by a nation-state to penetrate another nation’s computers or networks
5. Processing of gov data
   1. Data is shared among multiple organizations at different level of gov structure
   2. The problem of data breaches increases at every level
      1. Data duplication
      2. Data multiplicity
   3. Protection of data gets diluted if multiple copies of same data exist
6. Use cases: sharing of passport data with multiple parties in different scenarios
7. Government information sharing system
8. How blockchain helps
   1. Access and verification of a central data
      1. Data is in a central database
      2. Access to the database are the transaction
      3. Every such transactions (access to the data) is logged in a blockchain
      4. Data can be accessed only through the blockchain
      5. Anyone can verify who has accessed data and for what purpose
   2. Sharing of data
      1. Data is in the blockchain
      2. Everyone can verify which data has been shared
      3. Data cannot be altered
   3. Sharing of data and access control
      1. Keep both the data and the access at the blockchain
      2. Anyone can verify the data and the access
      3. Neither data nor access can be altered
      4. Access cannot be denied
9. Government use cases: worldwide
   1. Russia
   2. South korea
   3. Singapore
   4. India
      1. indiaChain ( a trial solution/ pillot project)
   5. USA
      1. Financial management
      2. Procurement
      3. It asset and supply chain management
      4. Patents, copyright management
      5. Federal personnel workforce data and so on
10. Case study – audit and compliance
    1. Financial data of an organization
    2. Auditing requires information about all key transactions over the reporting period
11. Auditing and compliance
    1. What if data is stored in a central server?
       1. The problem of a central server
       2. Who will manage the server?
    2. What is the validity of data provided by different divisions?
       1. What if the voice from two divisions do not match?
    3. Put data in blockchain
       1. Collects transaction records from diverse set of divisions
       2. No one can temper the data, but everyone can verify
    4. Blockchain is append-only
       1. Once a transaction has been recorded, it cannot be removed without changing the view of others
    5. Blockchain has multiple advantages
       1. Reduces the cost of auditing
       2. Auditors have global view of the data
       3. Compliance becomes passive to active
12. Citizen identity
    1. Control the access through blockchain
13. Blockchain for defense
    1. Multi – organizational information flow
14. Defense secure messaging and transaction platform
    1. Defense cyber security relies on secrecy of information and trust among individuals
    2. No need to insure that
       1. Only the privileged information has been accessed
       2. Information logs has not been tampered
       3. Provenance tracking of information origin and flow\
15. A success story: Estonia
    1. Digital id and decentralized distributed system
    2. Multiple benefits
16. Digital identity
    1. People known by their identity
    2. Identity as a collection of attributes
    3. Individuals do not have any control over the information that comprises their identities
    4. Identity fraud
17. Digital identity data is typically decentralized
    1. Passport
    2. DL
    3. Voter id
    4. Aadhaar
    5. Banking passbook
18. Digital identity – single sign on (SSO)
    1. Single identity for various purposes
    2. Widely conceptualized in software industry
19. SSO and decentralization
20. Fundamental principles of digital identity management
    1. Self – sovereign identity (privacy control)
    2. Distributed trust model
21. Why blockchain for identity management
    1. User centric design
    2. Automated and real time verification through smart contracts
    3. No one can tamper with the identity information of individuals;
22. Hyperledger Indy
    1. An example of a student who’s applying for a job and want degree transcript from college.
    2. Distributed identifier (DID)
    3. Pairwise relationship
    4. Trust anchors
23. Hyperledger indy – plenum consensus
    1. Similar to smart contract but tuned for verifying digital identity
    2. Uses redundant BFT (RBFT) algorithm for consensus
24. Startup for digital identity
    1. secureKey (sovrin) {recheck this Anurag}
25. open standard for digital identity
    1. IBM and Hyperledger (DIF) (2017)

**Day 10 Progress Report – Blockchain Learning**

Lecture covered: upto 52

Topic covered: blockchain security

1. Open network: security properties
   * 1. Identity
     2. Transactions
     3. transaction validation
     4. Transaction ordering
   1. Security
      1. Correct transaction validation
      2. Ledger immutability
   2. Privacy
      1. Pseudonymity, in some cases anonymity
   3. Assumtions:
      1. >50% computing power complies with protocol
      2. User wallet is safely maintained
      3. All contracts are deterministic
   4. Attack the asuumptions and human error
   5. Hard forks are sometime inevitable
2. Blockchain for enterprise world
3. Enterprise blockchain applications: security considerations
4. Security and privacy: key differentiation of fabric
5. Security in cloud/ hardware
6. Intel software guard extensions (SGX)
7. Coco framwork

Membership and access control in fabric

1. Identities and policies required at every stage
2. Membership and access control architecture
3. MSP details
4. A standard PKI based MSP for fabric
5. MSPs: building blocks for access polices
6. Blockchain crypto service provider (BCCSP)
   1. Abstraction
   2. Pluggability
   3. Multiple BCCSP
   4. International standard support
7. Integration with hardware security modules HSM
8. Tools to bootstrap a network
   1. Cryptogen
   2. Configtxgen

Privacy in fabric

1. Privacy in a blockchain system
   1. Transaction data privacy
   2. State data privacy
   3. Smart contract privacy
   4. User privacy
2. Privacy using channel in Hyperledger fabric
   1. Channel
   2. Channel creation
3. Data privacy using encryption
4. Data privacy using encryption within chaincode
   1. Options for encryption
      1. Transaction data
      2. Chaincode
      3. Chaincode state
5. Chaincode transient data
   1. Transient data:
   2. How to leverage it?
6. Smart contract confidentiality
7. Anonymous and unlinkable transactions (identity mixer)
   1. Ecerts
   2. TCerts
8. X509 vs identity mixer
9. Anynoymous and unlinkable transactions: auditability
10. Anynoymous and unlinkable transactions: revocation
11. Privacy with zero knowledge proof cryptography
12. zeroCash. UTXO ownership model with privacy
13. UTXO model with privacy
14. Ledger in Hyperledger fabric
15. State database options
16. sideDB motivation
17. sideDB multiple collections
18. define collections for each chaincode and channel
    1. define collections during chaincode deployment
    2. using channel configuration

Day 11 & 12 progress:

Course finalized as per the playlist.

Research Aspects – 1 (consensus scalability)

1. blockchain consensus protocols
   1. permissionless blockchain
      1. proof of work
      2. proof of stake
      3. proof of burn
      4. proof of elapsed time
   2. permissioned blockchain
      1. BFT
      2. PBFT
      3. RBFT
   3. POW vs PBFT
   4. PoW scalability
      1. Two magic numbers in PoW
         1. Block frequency
         2. Block size
         3. Transaction throughout – 7 transactions per second
   5. Performance vs scalability for Pow and BFT
   6. PoW vs PBFT – consensus finality
   7. PoW consensus vs BFT consensus
2. Bitcoin NG

Research aspect 2 (bitcoin NG)

1. Bitcoin NG
   1. Issue with Nakamoto consensus (PoW)
      1. Transaction scalability
      2. Issues with forks
   2. Bitcoin NG: a scalable PoW protocol
   3. Bitcoin vs bitcoin-NG
   4. Key blocks
   5. Microblocks
   6. Confirmation time

Research aspect 3: (collective signing)

1. Byzcoin
   1. Requirement for blockchain consensus
   2. Some background
      1. Collective signing (CoSi)
   3. Collective signing (CoSi)
   4. CoSi architecture
   5. CoSi based on Schnorr multisignature
      1. Key generation
      2. Signing
      3. Verification
      4. Proof
   6. CoSi protocol
   7. Scaling CoSi further
   8. BLS signature
      1. Key signature
      2. Signing
      3. Verification
   9. Advantages of BLS

Research aspect 4: (Byzcoin)

1. Problem with bitcoin
   1. No verifiable commitment
2. Problem with bitcoin-NG
   1. A faulty key block is verified only after end of the round
3. PBFTcoin – a strawman design
   1. Assumption : 3f+1 trustees with f failures
4. Problem of PBFT
5. Open the consensus group
6. Replace MAC with CoSi
7. Improve efficiency
8. CoSi as BFT protocol
9. Further improvement
10. Byzcoin performance

Secure Multiparty computation (MPC) over blockchain

1. MPC
2. Formal definition
3. Decentralized solution
4. Yao’s millionaire problem
   1. Precondition
   2. Protocol step 1 and 2 & 3
   3. Other protocols
      1. Oblivious transfer
      2. EGL protocol
      3. Yao’s garbled circuit
5. Problem with MPC
6. Fair MPC
7. Solve fair MPC – use a public bulletin board
8. Witness encryption

Algorand -1

1. Cryptocurrencies
   1. Bitcoin
   2. Ethereum
   3. Ripple
   4. Zcash
   5. Algorand
2. Bitcoin overview
   1. Key idea
   2. Communication
   3. Key assumption
3. Bitcoin technical limitation
   1. Resource wastage
   2. Concentration of power
   3. Vulnerable
   4. Scalability
   5. Ambiguity
4. Algorand: overview
   1. Key idea
   2. Communication
   3. Key assumption
5. Algorand: technical advancement
   1. Trivial computation
   2. True decentralization
   3. Finality of payment
   4. Scalability
   5. Security
6. Algorand: security perspective
   1. Avoid sybil attack
   2. Resilient denial of service attacks
7. Algorand: architecture
8. Cryptographic sortition
   1. Selection procedure
   2. Proof verification
   3. Seed selection
9. Block proposal
10. BA\*
    1. Two phase
    2. Strong synchrony versus weak synchrony
    3. Final consensus
    4. Tentative consensus
    5. Come out of tentative consensus
11. BA\* overall procedure
12. Algorand: summary
    1. No forks
    2. No miners
    3. No proof of work
    4. No wait for confirmation
    5. Trivial computation
    6. Perfect scalability
    7. Great security

Blockchain for data analytics: (blockchain for big data)

1. Big data
   1. Volume
   2. Velocity
   3. Variety
2. Big data processing
   1. Early and mid 2000s
      1. Zookeeper at yahoo
      2. Bigtable and mapreduce at google
      3. Cassandra at facebook
   2. Open source projects
      1. Hadoop distributed file system (HDFS)
      2. Hadoop mapreduce
3. Big data challenges
   1. Who will control the infrastructure when there are multiple actors involved?
   2. How well can trust the data?
   3. How do you monetize the data?
4. Blockchain use case: shared control of big data infrastructure
   1. Blockchain database
      1. BigchainDB
   2. A use case
5. Blockchain use case: audit trails on data
   1. //
   2. How will you proof that you are the originator of the data
   3. What about the crashes and malicious behaviors?
6. bigchainDB
   1. //
   2. Features
      1. Decentralization
      2. Query
      3. Immutability
      4. Byzantine fault tolerant
      5. Low latency
      6. Customizable
      7. Rich permissioning
   3. Decentralized ecosystem
7. Blockchain vs Distributed database vs bigchainDB

Blockchain for data analytics: (blockchain and AI)

1. Two disruptive technologies
   1. Blockchain
   2. AI
2. Blockchain can provide an excellent backbone for the development of AI algorithms
   1. Secure the data input
   2. Make every step visible to everyone
   3. Create data marketplace (bigchainDB)
3. singularityNET
4. singularityNET features:
   1. interoperability
   2. data sovereignty and privacy
   3. modularity
   4. scalability
5. singularityNET: An example
6. singularityNET: architecture
7. artificial neural network
8. DBC and ANN
9. DBC
   1. Objectives
   2. Process flow
10. numerAI
11. hedge fund
12. matrix ai network
13. objectives of matrix
14. matrix PoW + PoS
15. matrix: automated smart contract generation
16. matrix: security validation

comparing ecosystems: Ethereum

1. introduction to Ethereum
2. a brief history
3. Ethereum architecture
4. Account types, gas and transactions
5. Ethereum smart contracts
6. Ethereum virtual machine
7. Example solidity code
8. Smart contract patterns
9. Additional capabilities
10. Inter-contract calls

Ethereum development and quorum

1. Smart contract development and deployment
2. Developer framework: application layer
3. Enterprise Ethereum alliance
4. Quorum
   1. //
   2. Architecture
   3. Transaction processing
   4. Consensus

Corda

1. Corda
   1. Brief overview
   2. Legal agreement as a foundational concept
   3. Key design principle
   4. Peer to peer network
   5. Global corda network for interoperability
   6. UTXO state machine model
   7. Vaults and states
   8. The corda ledger
   9. Contracts
   10. Transactions can split and merge states
   11. Transaction and contract execution
   12. Transaction proposal and finality
   13. Transaction attachments
   14. Transaction flow
   15. Flow example
   16. Consensus
   17. Validating transaction histories
   18. Validity consensus
   19. Notaries: uniqueness consensus
   20. Example flow involving notary
   21. Notaries and transaction time-windows
   22. Oracle
   23. Corda node architecture

Concluding the course:

1. What we’ve covered
   1. The bitcoin blockchain
   2. The blockchain data structure
   3. The consensus
   4. Blockchain security
   5. Permissioned model: blockchain 2.0
   6. Hyperledger fabric: use cases
   7. Applications – government, finance, supply chain, data science
   8. Research aspects – scalability, security, privacy …
2. What we’ve not covered
   1. There are numerous blockchain platforms
      1. Ethereum
      2. Quorum
      3. Stellar
      4. Multichain
      5. Openchain
   2. How to write programs in individuals – explore it yourselves, the fundamental concept is the same
3. Guardtime
4. Remme
5. Simply vital health
6. Provenance – blockchain based supply chain
7. BlockVerify – anti counterfeit measure
8. Followmyvote
9. OpenBazzar
10. Arcade city

The end.