Cryptographic Methods in Blockchains

**Input**

- Fox
- The red fox jumps over the blue dog

**Digest**

- DFC3 3454 B8EA 788A 751A 696C 24D9 7009 CA99 2D17
- 0086 46BB FB7D CBE2 823C ACC7 6CD1 90B1 EE6E 3ABC
- 8FD8 7558 7851 4F32 D1C6 76B1 79A9 0DA4 AEFE 4819
- FCD3 7FD8 5AF2 C6FF 915F D401 C0A9 7D9A 46AF FB45
- 8ACA D682 D588 4C75 4BF4 1799 7D88 BCF8 92B9 6A6C
Learning Goals

- Explain the foundational theories, concepts, and technologies used in the cryptographic methods in Blockchains
  - Describe transaction protection and validation principles
  - Explain identity management principles using the Blockchain solutions
  - Explain access control models
  - Describe privacy management principles using the Blockchain solutions
Key Concepts

• **Transaction**
  - An agreement, communication, or movement carried out between a buyer and a seller to exchange an asset for payment

• **Cryptography**
  - Practice and study of techniques for secure communication in the presence of third parties called adversaries

• **Identity Management**
  - Ensuring that the proper people have the appropriate access to technology resources

• **Access Control**
  - Selective restriction of access to places or resources

• **Privacy**
  - Ability of an individual or group to seclude themselves or information about themselves, and thereby express themselves selectively
Cryptography, identity management, access control, privacy

- Bitcoin
- Ethereum
- Hyperledger Fabric
Cryptography, identity management, access control, privacy

- **Bitcoin**
- Ethereum
- Hyperledger Fabric
Bitcoin Transaction Lifecycle

Image: https://dev.to/gmfcastro/the-bitcoins-lifecycle-overview-1fld
Bitcoin Transaction and Cryptography

- Bitcoin follows the **unspent transaction outputs** (UTXO) model
  - **Security**
    - Maintains a Merkle proof of ownership
  - **Privacy**
    - Defines a data structure that each account holder can hold multiple instances of BTCs
      - *For example, a user may use different addresses for different transactions*
  - **Scalability**
    - No constraints of account-based model
      - *A user can send multiple transactions without worrying about the transaction sequence*

- **Synchronous communication**
  - Hash-based PoW to ensure the security of transactions
Bitcoin Transaction and Cryptography

Bitcoin consists the concepts of **digital keys**, addresses, digital signatures, and hashing that secure the ecosystem of Bitcoin network.

- **Digital keys (public and private)**
  - Stored in a user wallet
  - Follows elliptic curve cryptography
  - Public key is calculated from the private key
  - Private key sign transaction to spend funds
  - Private key is 256-bit long random string
Bitcoin Transaction and Cryptography

Bitcoin consists the concepts of digital keys, addresses, digital signatures, and hashing that secure the ecosystem of Bitcoin network

- **Bitcoin addresses**
  - Bitcoin generate addresses from the public key
    - Begin with the digit 1 (e.g., 1J7mdg5rbQyUHENYdx39WVK7fsLpEoXZy)
  - Receive funds
Bitcoin consists the concepts of digital keys, addresses, **digital signatures**, and hashing that secure the ecosystem of Bitcoin network.

- **Digital signatures**
  - Digital signatures utilise for signing and verifying
  - Private key create signatures that is required to spend bitcoin
  - Provide a proof that you own the private key without revealing it
  - Makes sure that a transaction cannot be modified by anyone after signed
  - Digital signature is created by the elliptic curve digital signature algorithm (**ECDSA**)
Bitcoin Transaction and Cryptography

Bitcoin consists the concepts of digital keys, addresses, digital signatures, and hashing that secure the ecosystem of Bitcoin network.

- **Hashing**
  - SHA256, RIPEMD160, and Base58Check to make a bitcoin address from a public key
  - *Bitcoin address = Base58Check(RIPEMD160(SHA256(public-key)))*
  - Connect the block to previous block
  - Manage transactions within Merkle tree
  - Hashing is used for crypto puzzles
    - *In the mining of Proof of Work algorithm*
**Bitcoin Identify Management**

- **No association** between the Bitcoin address and any user’s identity

- **Ownership** within Bitcoin platform is established through
  - Digital keys, Bitcoin addresses, and digital signatures

- Users are **identified** by their **Bitcoin addresses**
  - A Bitcoin address represents the ownership of a private/public key pair

- **Peer-to-Peer authentication**
  - Allows nodes to authenticate each other’s identity using ECDSA
Access Control

- Bitcoin network is based on a Proof of Work, not access control
  - Network can be open and no encryption is required for Bitcoin traffic

- No permission is required to become part of the Bitcoin network
  - Anyone can join the network and start participating in the Bitcoin network
  - Transactions are publicly available and anyone can see the transactions
Privacy

- **Bitcoin is pseudonymous**
  - It means funds are not tied to real-world entities but rather Bitcoin addresses
  - Owners of Bitcoin addresses are not explicitly identified, but all transactions on the Blockchain are public
  - Pseudonymity of a system as a privacy property to protect user’s identity

- **Users can generate as many key pairs**
  - Multiple Bitcoin addresses as users want
  - It keep user from being linked and trace back to a common owner

- **Tor Transport**
  - Offers anonymity, untraceability and privacy
Bitcoin - What have we learned?

- Unspent transaction output model
- Digital keys, addresses, digital signatures, and hashing
- Ownership – through digital keys, bitcoin addresses, and digital signatures
- Proof-of-Work, not access control
- Bitcoin is pseudonymous
Contents

- Cryptography, identity management, access control, privacy
  - Bitcoin
  - Ethereum
  - Hyperledger Fabric
Ethereum Transaction Lifecycle

65 byte ECDSA signature

Ethereum Transaction and Cryptography

- Ethereum follows the account-based online transaction model
  - The token is signed by the message writer (sender)
  - The writer’s ownership of token value can be attested, and
  - The writer’s spending account has sufficient balance for the transaction
Ethereum Transaction and Cryptography

- Two types of account
  - Externally-owned account (EOA)
    - A user on a Ethereum network that has 20-byte address and hold valid private key
  - Contract account (CA)
    - Creates on Ethereum network when deploy a contract

- Use account nonce to prevent double-spend

- Proof of Work nonce to satisfy Ethereum Proof of Work (crypto puzzle) through mining process
Ethereum Transaction and Cryptography

Ethereum performs three different types of transactions:

- **Transfer funds between EOA**
  - *For example, One EOA transfer funds (e.g., 10 Ethers) to another EOA*

- **Deploy a contract on a Ethereum network**
  - *Deploy a compiled contract (e.g., the bytecode) on an Ethereum network*

- **When execute a deployed contract**
  - *EOA executes a deployed contract that considers as a transaction on a Ethereum network*
Ethereum performs three different types of transactions:

- **Transfer funds between EOA**
  - For example, one EOA transfers (e.g., 10 Ethers) to another EOA
- **Deploy a contract on a Ethereum network**
  - Deploy a compiled contract (i.e., the bytecode) on an Ethereum network
- **When execute a deployed contract**
  - EOA executes a deployed contract that considers as a transaction on a Ethereum network

All transaction types follow the same structure:

- From *(sender)*, To *(recipient)*, Value *(amount of fund)*, Data *(encoded arguments)*, Signature, Gas limit *(maximum gas spent for a transaction)*, Gas price *(sender willing to pay for a transaction)*
Ethereum consists of **digital keys**, addresses, digital signatures, and hashing that secure the ecosystem of Ethereum network

- **Digital keys (public and private)**
  - Stored in a user wallet
  - Follows elliptic curve cryptography
  - Public key is calculated from the private key
  - The private key is used to create signatures
  - Private key sign transaction to spend funds
    - *Ownership of the account and associated contracts*
  - Private key is 256-bit long random string
    - *Follows Keccak-256 hash algorithm*
Ethereum Transaction and Cryptography

Ethereum consists of digital keys, addresses, digital signatures, and hashing that secure the ecosystem of Ethereum network.

- **Ethereum addresses**
  - Unique identifiers that derived from the public key
  - Calculate the hash of public key using Keccak-256 and keep last 20 bytes
    - `0x001d3f1ef827552ae1114027bd3ecf1f086ba0f9`
    - `0x` represent hexadecimal-encoded Ethereum address
  - Receive funds
  - Also represent contracts (such address are not backed by private key)
Ethereum Transaction and Cryptography

Ethereum consists of digital keys, addresses, digital signatures, and hashing that secure the ecosystem of Ethereum network.

- **Digital signatures**
  - Private key create digital signature that is required to spend ether
  - Digital signatures authenticate owners of accounts or users of contracts
  - Digital signature can sign any message
  - Makes sure that a transaction cannot be modified by anyone after signed
  - Digital signature is created by the elliptic curve digital signature algorithm (ECDSA)
Ethereum Transaction and Cryptography

Ethereum consists of digital keys, addresses, digital signatures, and **hashing** that secure the ecosystem of Ethereum network.

- **Hashing**
  - Keccak-256 to make an Ethereum address from a public key
  - Ethereum address = \textit{Keccak256}(public-key)
    - Keep only the last 20 bytes
  - Connect the block to previous block
  - Manage transactions within Merkle tree
  - Hashing is used for crypto puzzles
    - In the mining of Proof of Work algorithm
  - Transaction integrity
Ethereum Identify Management

- Similar to Bitcoin
- **No association** between the Ethereum address and any user’s identity
- **Ownership** within Ethereum platform is established through
  - Digital keys, Ethereum addresses, and digital signatures
- Users are **identified** by their **Ethereum addresses**
  - An Ethereum address represents the ownership of a private/public key pair
Access Control

- Ethereum network is based on a *Proof of Stake*, not access control
  - Network can be open and no encryption is required for Ethereum traffic

- **No permission** is required to become part of the Ethereum network
  - Anyone can join the network and start participating in the Ethereum network
  - Transactions are publicly available and anyone can see the transactions

- Applications can implement *role based access control* in Solidity
● Similar to Bitcoin, **Ethereum is also pseudonymous**
  ○ It means **funds are not tied to real-world entities** but rather Ethereum addresses
  ○ **Owners** of Ethereum addresses are **not explicitly identified**, but all transactions on the Blockchain are public
  ○ **Pseudonymity** of a system as a privacy property to protect user’s identity
Privacy

- Users can generate as many key pairs
  - Multiple Ethereum addresses as users want
  - It keep user from being linked and trace back to a common owner

- Zero-Knowledge proof
  - Enterprise version

- Mixers for Privacy
  - Swapping Ether for Ether to further anonymise the transacting ether
  - Protect against de-anonymization attack
Ethereum - What have we learned?

- Account based transaction model
- Digital keys, addresses, digital signatures, and hashing
- Ownership – through digital keys, addresses, and digital signatures
- Proof of Work, not access control
  - **Solidity** can be used to define Access Control policies
- Ethereum is pseudonymous
Cryptography, identity management, access control, privacy

- Bitcoin
- Ethereum
- Hyperledger Fabric
HLF Transaction Lifecycle

1. Transaction Proposal
2. Transaction Simulation
3. Endorsed Proposal Response
4. Transaction Submission
5. Transaction Ordering and New Block Creation
6. New Block Distribution
7. Event Delivery

Image: https://www.bitdeal.net/hyperledger-fabric
HLF Transaction and Cryptography

HLF is a permissioned blockchain, so the transaction process is different as compared to Bitcoin and Ethereum

- HLF follows an execute-order-validate architecture for transactions
  - Execute a transaction and check its correctness
    - Endorsing process
  - Order transactions
    - Order by pluggable consensus protocol
  - Validate transactions
    - Validate against an application-specific endorsement policy before committing them to the ledger
HLF Transaction and Cryptography

HLF is a permissioned blockchain, so the transaction process is different as compared to Bitcoin and Ethereum

- HLF follows a execute-order-validate new architecture
  - Execute a transaction and check its correctness
    - Endorsing process
  - Order transactions
    - Order by pluggable consensus protocol
  - Validate transactions
    - Validate against an application-specific endorsement policy before committing them to the ledger

- HLF executes transactions before reaching final agreement on their order
Hyperledger Fabric consists the concepts of **digital keys**, digital signatures, certificates, and hashing that secure the HLF ecosystem.

- **Digital keys (public and private)**
  - Private key sign the transaction and produce digital signature
  - Public key included in the transaction payload sent to peers / orderers
  - Peers and orderers verify the signatures using the public key in the transaction
  - MSP is formed by an instance of a PKI infrastructure
Hyperledger Fabric consists the concepts of digital keys, digital signatures, certificates, and hashing that secure the HLF ecosystem.

- Digital keys (private and public)
  - Private key is used to sign the transaction and produce digital signature.
  - Public key is included in the transaction payload sent to peers/orderers.
  - Peers and orderers verify signatures using the public key in the transaction.
  - MSP is formed by an instance of a PKI infrastructure.

- Certificate Authority (certificate issuer) sign the certificate of party by their private key
  - To prove identity without revealing information in it
  - To make sure integrity of the certificate
    - Tampering will invalidate the certificate
HLF Transaction and Cryptography

Hyperledger Fabric consists the concepts of digital keys, digital signatures, certificates, and hashing that secure the HLF ecosystem.

- **Digital signatures**
  - Authentication mechanisms rely on digital signatures.
  - Transaction digital signature sent to peers / orderers.
  - Digital signatures also provide guarantees on the integrity of the signed message.
  - Digital signature mechanisms require each party to hold two cryptographically public and private keys.
  - Digital signature corresponding to the public key that could be verified by recipients.
  - Digital signature is created by the elliptic curve digital signature algorithm (ECDSA).
Hyperledger Fabric consists the concepts of digital keys, digital signatures, certificates, and hashing that secure the HLF ecosystem.

- **Hashing**
  - Hyperledger Fabric uses SHA256 for hashing
  - Connect the block to previous block
  - Manage transactions within the block (hash of the block contents)
HLF Transaction and Cryptography

Hyperledger Fabric consists the concepts of digital keys, digital signatures, certificates, and hashing that secure the HLF ecosystem.

- Digital certificates
  - Certification authority issues digital certificates
    - Responsible for registration of identities, issuance of Enrollment Certificates, certificate renewal, and revocation
  - Compliant with the X.509 standard
    - Standard defining the format of public key certificates
  - Holds a set of attributes relating to the holder of the certificate
Hyperledger Fabric consists the concepts of digital keys, digital signatures, certificates, and hashing that secure the HLF ecosystem.

- **Membership service provider (MSP)**
  - Offers an abstraction of membership operations
  - Manages identities
    - *Authenticate clients who want to join the blockchain network*
  - Certificates authority is used in MSP to provide identity verification
  - On the network, it is a bunch of directories that contains digital certificates
HLF Identify Management

- Hyperledger Fabric provides a membership identity service that manages user ID
  - Service authenticates participants on the network
    - Hyperledger Fabric uses X.509 certificates to authenticate every entity and member in the network

- Access control as an additional layers of permission
  - Authorise to use specific network operations
    - For example, a specific user could be permitted to invoke a chaincode application, but be blocked from deploying new Chaincode

- Hyperledger working on the concept of TrustID
  - TrustID would decentralise the identity within HLF
Access Control

- Hyperledger Fabric uses **access control lists** (ACLs) to manage access to resources
  - Based on the rule that evaluates to true or false
  - User chaincode, events stream source, or system chaincode are considered resources
    - *For example, If a participant node make a request to call multiple system chaincodes, then all of the ACLs for those system chaincodes must be satisfied*
  - The list of resources are found in `configtx.yaml`

- Hyperledger fabric supports an **attribute-based access control**

- Default there are 2 types of policies that are associated with resources
  - Signature Policies and Implicit Meta Policies
Privacy

- **Hyperledger Fabric equipped with a membership infrastructure**
  - Enables participants of the network to authenticate themselves in transactions
  - Prove authorisation to perform a variety of system operations

- **Channels**
  - Segregate network into channels
    - Channels represent a subset of participants that are authorised to see the data for the chaincode that are deployed on that channel
Privacy

- **Private transactions**
  - Private-data to keep ledger data private from other organizations on the channel

- **Chaincode-based access control logic**
  - Restrict data access to certain roles by building chaincode-based access control logic

- **Zero-knowledge proof (ZKP)**
  - Anonymous client authentication with Identity Mixer
  - Privacy-preserving exchange of assets with Zero-Knowledge asset transfer
Hyperledger Fabric - What have we learn?

• Execute-order-validate architecture

• Digital keys, digital signatures, certificates and hashing

• Membership identity management – user IDs

• Access control lists

• Segregate network into channels
Summary

- Cryptography, identity management, access control, privacy
  - Bitcoin
  - Ethereum
  - Hyperledger Fabric
### Summary

<table>
<thead>
<tr>
<th></th>
<th>Permissionless</th>
<th>Permissionless</th>
<th>Permissioned</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Transaction</strong></td>
<td>Unspent transaction output model</td>
<td>Account based transaction model</td>
<td>Execute-order-validate architecture</td>
</tr>
<tr>
<td><strong>Ecosystem</strong></td>
<td>Digital keys, addresses, digital signatures, hashing</td>
<td>Digital keys, addresses, digital signatures, hashing</td>
<td>Digital keys, digital signatures, certificates and hashing</td>
</tr>
<tr>
<td><strong>Identity management</strong></td>
<td>Ownership</td>
<td>Ownership</td>
<td>Membership identify management</td>
</tr>
<tr>
<td><strong>Access control</strong></td>
<td>Proof-of-Work</td>
<td>Proof-of-Work, Solidity to define Access Control policies</td>
<td>Access control lists</td>
</tr>
<tr>
<td><strong>Privacy</strong></td>
<td>Pseudonymous</td>
<td>Pseudonymous</td>
<td>Channels</td>
</tr>
</tbody>
</table>
Further Reading


**Blog posts**

- [https://medium.com/@kctheservant/transactions-in-ethereum-e85a73068f74](https://medium.com/@kctheservant/transactions-in-ethereum-e85a73068f74)
- [https://hackernoon.com/role-based-access-control-for-the-ethereum-blockchain-bcc9dfbcfe5c](https://hackernoon.com/role-based-access-control-for-the-ethereum-blockchain-bcc9dfbcfe5c)
Short Videos

- Ethereum vs Hyperledger
  https://youtu.be/tjrvWvX4diA
- Ethereum Security Overview
  https://youtu.be/gGYki6KYTs8
- Privacy On Ethereum
  https://youtu.be/Dv9jiOc8kOY
- What is Hyperledger
  https://youtu.be/Y177TCUc4g0
- Byzantine Fault Tolerance
  https://youtu.be/3wUp5V-4s8Y
- Hyperledger fabric security
  https://youtu.be/Y177TCUc4g0
- Cryptography, Security Modelling, Privacy, and Confidentiality in Hyperledger
  https://youtu.be/RT33CuFqgdA
- Security Vulnerabilities in Chaincode
  https://youtu.be/aFRfojctDY

- How Blockchain Transactions Work (Adding Data to Blockchains)
  https://youtu.be/l_VBuzSV9vA
- Bitcoin - Proof of work
  https://youtu.be/9V1bipPkCTU
- Bitcoin - The security of transaction block chains
  https://youtu.be/8zgvzmKZ5vo
- Proof-of-Stake (vs proof-of-work)
  https://youtu.be/M3EFi_POhps

- How Blockchain Transactions Work (Adding Data to Blockchains)
  https://youtu.be/l_VBuzSV9vA