Dissertation

Eduardo Oliveira

March 15, 2025

1 Blockchain

1.1 Foundational Aspects

Blockchain is a decentralized, immutable ledger technology designed to facilitate secure and transparent transactions within distributed networks. Initially conceptualized for the Bitcoin blockchain **?**, this technology has since evolved into a multi-purpose infrastructure underpinning various domains, including finance, supply chain management, and digital identity verification.

The development of blockchain, however, did not occur in isolation. The concept of a cryptographically secured chain of blocks predates Bitcoin and draws from earlier research on distributed consensus and cryptographic techniques. A key component in blockchain structures is the Merkle tree, introduced by Ralph Merkle in the 1980s **?**. These trees enable efficient data integrity verification by organizing hashes in a hierarchical structure, which is crucial for maintaining the integrity of blockchain data.

Building on these foundational cryptographic concepts, Stuart Haber and W. Scott Stornetta proposed a method for securely time-stamping digital documents in 1991 **?**. This innovation was significant because it prevented backdating and tampering, laying the groundwork for immutable records. In 1992, Haber, Stornetta, and Bayer further refined this approach by incorporating Merkle trees into their time-stamping system, thereby improving efficiency and strengthening security **?**. These advancements not only contributed to the development of blockchain but also highlighted the potential of decentralized, immutable ledgers for maintaining verifiable records.

The inherent properties of blockchain—decentralization, immutability, transparency, and security—make it particularly well-suited for addressing challenges in scientific reproducibility. By maintaining an auditable and tamper-proof history of research data and workflows, blockchain ensures long-term verifiability and integrity. This application leverages the foundational principles established by early cryptographic and distributed systems research, demonstrating how blockchain can extend beyond financial transactions to support rigorous scientific reproducibility.

Contextualizing Decentralization and Distributed Systems

Decentralization and distributed systems are foundational principles that underpin blockchain technology and its applications. Unlike traditional systems that rely on a central authority to manage operations, decentralized systems distribute control and decision-making across a network of participants. This architecture eliminates single points of failure, enhances security, and fosters transparency.

Decentralization: A Core Principle

Decentralization ensures that no single entity has overarching control over the system. In blockchain networks, this is achieved through consensus protocols that govern how nodes validate transactions and maintain the ledger. Decentralized systems can be categorized into levels such as fully centralized, semi-decentralized, or fully decentralized, depending on the degree of control dispersed among participants **?**. Fully decentralized systems, such as Bitcoin or Ethereum, operate without middlemen or central authorities, empowering users with full control over their assets and data.

This distribution of control enhances fault tolerance by ensuring that if one node or server fails, others can maintain the network's functionality. For example, decentralized storage systems replicate data across multiple nodes, ensuring redundancy and reliability even in the face of node failures **?**. Additionally, decentralization improves transparency by making all transactions publicly verifiable on an immutable ledger.

Role of Distributed Systems

Distributed systems are integral to achieving decentralization in blockchain. These systems rely on a peer-to-peer network where each node maintains a copy of the blockchain ledger. When a transaction occurs, it is broadcast to all nodes, validated through consensus mechanisms, and added to the ledger **?**. This architecture ensures that all nodes agree on the current state of the blockchain without requiring a central authority.

Distributed computing also plays a critical role in maintaining real-time synchronization across nodes. Consensus protocols like Proof of Work (PoW) or Proof of Stake (PoS) ensure that only valid transactions are added to the blockchain while resolving conflicts between competing blocks **?**. This decentralized approach enhances security by making it computationally infeasible for malicious actors to alter historical records.

Absence of Central Authority

The absence of a central authority in blockchain networks is a key differentiator from traditional systems. In conventional architectures, a single entity governs

operations, creating potential vulnerabilities such as censorship, inefficiency, and single points of failure. Blockchain eliminates these risks by distributing control among participants who collectively validate transactions.

For example:

- In decentralized finance (DeFi), users interact directly with smart contracts rather than intermediaries like banks or payment processors **?**.
- Decentralized applications (dApps) operate on blockchains without relying on centralized servers or databases. Instead, they use smart contracts to execute transactions securely and transparently **?**.

This non-centralized structure not only enhances security but also promotes user sovereignty by giving individuals control over their data and assets without needing to trust third parties.

1.2 Consensus Mechanisms in Blockchain

Consensus mechanisms are fundamental to blockchain networks, ensuring agreement among distributed nodes without requiring centralized authority. These mechanisms validate transactions and maintain the integrity of the ledger, preventing issues such as double-spending and malicious attacks.

1.2.1 Proof of Work (PoW)

Proof of Work (PoW) was first implemented in Bitcoin **?** and remains one of the most well-known consensus mechanisms. PoW requires network participants, known as miners, to solve complex cryptographic puzzles using computational resources. The first miner to find a valid solution can append a new block to the blockchain and receive a block reward. This process ensures security but comes at the cost of significant energy consumption **?**. Additionally, the difficulty adjustment mechanism ensures that blocks are produced at a steady rate by modifying the complexity of the puzzle based on the total computational power of the network.

1.2.2 Proof of Stake (PoS)

Proof of Stake (PoS) is a consensus mechanism designed to address the scalability and energy inefficiencies of Proof of Work (PoW). Its development reflects a paradigm shift in blockchain security and governance, with roots in early cryptocurrency discourse and iterative improvements over time. The concept emerged in 2011 through discussions on the Bitcointalk forum, predating its formal implementation. The first practical application appeared in 2012 with PPCoin (later Peercoin), introduced by Sunny King and Scott Nadal **?**. While Peercoin pioneered PoS, it adopted a hybrid PoW/PoS model to bootstrap initial security, where PoW mined the first blocks, and PoS secured subsequent ones. This hybrid approach aimed to mitigate Bitcoin's soaring energy costs,

which exceeded $150,000 daily in 2012. The first pure PoS implementation arrived in 2013 with NXT Blackcoin, which eliminated mining entirely and relied solely on staking. These projects demonstrated that decentralized consensus could be achieved without energy-intensive computations, setting the stage for modern PoS systems like Ethereum 2.0 **?**and Cardano **?**.

1.2.3 Mining and Block Validation

Mining is the process by which transactions are validated and added to a blockchain. In PoW-based systems, miners compete to solve cryptographic puzzles, while in PoS-based systems, validators are selected to propose and confirm blocks based on their stakes. Mining serves two key purposes: securing the network by making attacks computationally expensive and issuing new tokens as rewards. This incentive structure aligns participant behavior with the network's security goals **?**. For example, Bitcoin employs PoW mining, while Ethereum 2.0 uses PoS validation.

1.2.4 Byzantine Fault Tolerance (BFT)

Byzantine Fault Tolerance (BFT) is a property of distributed systems that allows them to function correctly even if some nodes act maliciously or fail **?**. Traditional consensus mechanisms, such as Practical Byzantine Fault Tolerance (PBFT) **?**, require $2f + 1$ honest nodes out of $3f + 1$ total nodes to tolerate f Byzantine faults. PBFT-based systems provide high efficiency and finality but require a known set of validators, making them more suitable for permissioned blockchains like Hyperledger Iroha.

Hyperledger Iroha incorporates a specialized BFT consensus mechanism called YAC (Yet Another Consensus), which is optimized for voting-based block validation and low-latency operations. This integration ensures that Iroha can achieve consensus efficiently while maintaining the robustness expected of BFT systems.

1.2.5 YAC Consensus and Byzantine Fault Tolerance

The YAC (Yet Another Consensus) algorithm ensures Byzantine Fault Tolerance (BFT) **?** by employing a voting-based mechanism to achieve consensus in permissioned blockchain networks. Here's how YAC achieves BFT:

1. Voting for Block Hash

Validators in the network vote on the hash of the proposed block rather than its entire content. This reduces communication overhead while ensuring consistency among honest nodes.

2. Fault Tolerance

YAC tolerates Byzantine faults by requiring a supermajority (e.g., 2/3 of validators) to agree on the block hash. This guarantees that even if some nodes act maliciously or fail, the system can still reach consensus.

3. Finality

Once a supermajority is reached, the block is considered finalized, and all honest nodes accept it as part of the blockchain. This prevents forks and ensures the integrity of the ledger.

4. Permissioned Design

YAC is specifically designed for permissioned blockchains, where validators are pre-approved entities. This controlled environment enhances security and reduces the likelihood of large-scale malicious attacks.

YAC's lightweight design and focus on efficient communication make it suitable for enterprise-grade applications, such as Hyperledger Iroha, where high performance and reliability are critical **?**.

Consensus Formulae for YAC

The YAC (Yet Another Consensus) algorithm ensures Byzantine Fault Tolerance (BFT) by implementing a voting-based process. The key formulae are as follows:

1. Fault Tolerance

The network must satisfy:

$$n \geq 3f + 1$$

Where:

- n: Total number of nodes in the network.
- f: Maximum number of Byzantine (malicious or faulty) nodes tolerated.

This ensures that the network can tolerate up to f Byzantine nodes while still achieving consensus.

2. Supermajority Agreement

For a block to be finalized, a supermajority of nodes must agree on its hash:

$$v > \frac{2n}{3}$$

Where:

- v: Number of votes for the block hash.
- n: Total number of nodes.

This condition guarantees both safety and liveness in consensus.

3. Voting Process

The voting process involves two phases:

1. **Proposal Phase**: A leader node proposes a block hash.
2. **Voting Phase**: Nodes vote on the proposed hash, and votes are collected until the supermajority condition is met.

4. Finality

Once a block achieves supermajority agreement, it is considered finalized and added to the blockchain. This ensures that all honest nodes accept the same block, preventing forks.

1.3 Public and Private Blockchains

Blockchains can be classified based on their accessibility and governance models, primarily into public and private blockchains.

1.3.1 Public Blockchains

Public blockchains, such as Bitcoin and Ethereum, are open to anyone, allowing unrestricted participation in the network. These blockchains prioritize decentralization and security at the expense of scalability. Consensus mechanisms in public blockchains typically rely on PoW or PoS, where participants must follow established protocols to validate transactions **?**. Public blockchains offer transparency, as all transactions are recorded on a publicly accessible ledger, making them suitable for applications requiring trustless environments.

1.3.2 Private and Permissioned Blockchains

Private blockchains restrict participation to authorized entities, making them suitable for enterprise applications. Hyperledger Fabric and Hyperledger Iroha are examples of permissioned blockchain frameworks designed for regulated environments where identity verification and compliance are critical **?**. Private blockchains provide improved scalability and efficiency since they do not require energy-intensive consensus mechanisms like PoW. However, they trade off decentralization, as a governing authority typically oversees the network.

1.4 Hybrid Blockchain Models

Hybrid blockchain models combine aspects of both public and private blockchains. These architectures allow organizations to maintain a private ledger while interacting with public networks for verification and transparency purposes. Such approaches are particularly useful in industries where both privacy and auditability are required, such as supply chain management and financial services **?**.

Public Key Cryptography in Blockchain

Public key cryptography plays a fundamental role in securing blockchain networks by enabling secure transactions, identity verification, and data integrity without requiring a centralized authority. It forms the foundation for digital signatures, key management, and encryption mechanisms that ensure trust and security in decentralized environments.

Role of Public Key Cryptography in Blockchain

Blockchain networks rely on asymmetric cryptography, also known as public key cryptography, to authenticate and authorize transactions. Each participant in the network possesses a pair of cryptographic keys: a **public key**, which serves as an address that others can use to send transactions, and a **private key**, which is used to sign transactions and prove ownership. When a user initiates a transaction, they generate a **digital signature** using their private key, allowing other participants to verify the authenticity of the transaction without revealing the private key itself.

This mechanism ensures that only the rightful owner of an asset can authorize its transfer, preventing fraud and unauthorized access. Additionally, cryptographic hashing techniques complement public key cryptography by ensuring data integrity and linking transactions in an immutable ledger.

Commonly Used Cryptographic Ciphers and Standards

Several cryptographic ciphers and standards are widely used in blockchain implementations to provide strong security guarantees:

- **RSA (Rivest-Shamir-Adleman):** A traditional public key cryptosystem based on the difficulty of factoring large prime numbers. While RSA is widely used in general cryptographic applications, its key sizes are relatively large compared to modern alternatives, making it less practical for blockchain applications.
- **Elliptic Curve Cryptography (ECC):** A more efficient asymmetric cryptography scheme that provides the same level of security as RSA but with significantly smaller key sizes. This efficiency makes ECC the preferred choice for blockchain applications.

- **ECDSA (Elliptic Curve Digital Signature Algorithm):** A widely adopted digital signature scheme based on ECC, used in Bitcoin and Ethereum to secure transactions.
- **EdDSA (Edwards-curve Digital Signature Algorithm):** A modern alternative to ECDSA, known for its faster signature verification and improved security properties. It is used in newer blockchain protocols like Monero.
- **X25519:** A secure key exchange protocol based on Curve25519, commonly used in cryptographic operations for secure communication in blockchain applications.

Elliptic Curve Cryptography (ECC) in Blockchain

Elliptic Curve Cryptography (ECC) is a type of public key cryptography that leverages the mathematical properties of elliptic curves over finite fields to provide strong security with smaller key sizes. The security of ECC is based on the **Elliptic Curve Discrete Logarithm Problem (ECDLP)**, which is computationally hard to solve.

In blockchain systems, ECC is primarily used for:

1. **Digital Signatures:** ECC enables the creation of compact and secure digital signatures, such as those used in Bitcoin (ECDSA) and newer blockchain protocols (EdDSA).
2. **Key Pair Generation:** Blockchain wallets generate private-public key pairs using elliptic curves, ensuring that users can securely sign and verify transactions.
3. **Scalability and Efficiency:** Due to its small key size and lower computational requirements, ECC allows blockchain networks to process transactions more efficiently while maintaining security.

Bitcoin, for instance, uses the **secp256k1** elliptic curve for key generation and signing, which provides a 256-bit key length offering high security with lower processing overhead compared to traditional cryptographic methods.

1.5 Conclusion

Consensus mechanisms, mining, and fault tolerance strategies are critical in ensuring blockchain security and functionality. The choice between PoW, PoS, and BFT-based approaches impacts the efficiency, decentralization, and security of blockchain networks. Furthermore, the distinction between public, private, and hybrid blockchains influences their applications, with public blockchains prioritizing trustlessness and private blockchains emphasizing control and efficiency. Understanding these foundational aspects enables the development of blockchain-based solutions tailored to the needs of Open Science and research reproducibility.

Conclusion

Public key cryptography is a cornerstone of blockchain security, enabling authentication, digital signatures, and secure communication. The adoption of ECC and its derivatives, such as ECDSA and EdDSA, has significantly improved the efficiency and scalability of blockchain networks, making them resilient to attacks while minimizing computational and storage costs. Future blockchain advancements may incorporate more advanced cryptographic techniques, including post-quantum cryptography, to further enhance security in decentralized systems.

References

Blockchain-based decentralized storage systems for sustainable data self-sovereignty: A comparative study. *Sustainability*, 16(17):7671, 2024.

Dave Bayer, Stuart Haber, and W. Scott Stornetta. Improving the Efficiency and Reliability of Digital Time-Stamping. In Renato Capocelli, Alfredo De Santis, and Ugo Vaccaro, editors, *Sequences II*, pages 329–334, New York, NY, 1993. Springer New York.

Joseph Bonneau, Andrew Miller, Jeremy Clark, Arvind Narayanan, Joshua A. Kroll, and Edward W. Felten. Sok: Research perspectives and challenges for bitcoin and cryptocurrencies. *Proceedings - IEEE Symposium on Security and Privacy*, pages 104–121, 2015.

Miguel Castro and Barbara Liskov. Practical byzantine fault tolerance. In *Proceedings of the Third Symposium on Operating Systems Design and Implementation (OSDI)*, pages 173–186, 1999.

Vitalik Buterin et al. Combining ghost and casper. *arXiv preprint arXiv:2003.03052*, 2020.

Stuart Haber and W. Scott Stornetta. How to time-stamp a digital document. *J. Cryptology*, 3(2):99–111, January 1991.

Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Proceedings of the 37th Annual International Cryptology Conference (CRYPTO)*, pages 357–388. Springer, 2017.

Sunny King and Scott Nadal. Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. 2012.

Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, 1982.

Ralph C. Merkle. A Digital Signature Based on a Conventional Encryption Function. In G. Goos, J. Hartmanis, D. Barstow, W. Brauer, P. Brinch Hansen, D. Gries, D. Luckham, C. Moler, A. Pnueli, G. Seegmüller, J. Stoer, N. Wirth, and Carl Pomerance, editors, *Advances in Cryptology — CRYPTO '87*, volume 293, pages 369–378. Springer Berlin Heidelberg, Berlin, Heidelberg, 1988. Series Title: Lecture Notes in Computer Science.

Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2009.

Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press, 2016.

Investopedia Staff. What is decentralized finance (defi) and how does it work?, 2024.

Investopedia Staff. Decentralized applications (dapps): Definition, uses, pros and cons, 2025.

Makoto Takemiya and Kenta Tanaka. Yac: Bft consensus algorithm for blockchain. *arXiv preprint arXiv:1809.00554*, 2018.