Discovering Algorand: The First Truly Decentralised Permissionless Blockchain (Oral Communication)

Stefano De Angelis[1]

Algorand Inc., Boston, Massachusetts, U.S.A
stefano@algorand.com

Blockchain is a disrupting technology that stands to revolutionise traditional computer infrastructures and the Internet itself. It consists of a distributed ledger of immutable data, freely accessible and editable by everyone. Blockchains change the way transactional systems work, they secure peer-to-peer transactions –such as payments, data, and asset transfers– without relying on any trusted third party; and usher in new pioneering methods to represent and exchange value (digital and physical) –such as cryptocurrencies, fungible and non-fungible tokens, and smart contracts. However, like any historical revolution, the blockchain comes with its challenges: in their first decade, blockchains have suffered technical limitations, mostly related to *decentralisation*, *scalability*, and *security* guarantees. The famous blockchain trilemma states that any blockchain can only guarantee only two out of those three properties simultaneously.

In this speech, I would like to introduce Algorand, an open, permissionless, truly decentralised, and efficient blockchain that has solved the trilemma through cryptography. In the first part of my speech, I will address the importance of the trilemma and how Algorand has solved it; then I will present Algorand's main features and capabilities. Finally, I will conclude my speech with a reflection on the '*code is law*' concept, outlining the limitations caused by smart contracts' weaknesses and the best practices to mitigate them in Algorand.

Decentralisation Needs Randomness. Blockchains entrust the consensus protocol to maintain a consistent state of the ledger across the network. Consensus replaces centralised trusted regulators of traditional systems. To this extent, decentralisation is paramount: the more authorities agree on the ledger state, the harder is for an attacker to compromise it. However, by increasing the number of authorities, consensus becomes more and more difficult to achieve. Consequently, to preserve security through decentralisation, scalability is inevitably affected. This is mostly related to the *leader election*, i.e. how parties agree on the next authority in charge of updating the ledger. For instance, Bitcoin and Ethereum, the two world's most famous blockchains, adopt Proof-of-Work (PoW) which introduces *randomness* to leader election with mining. However, mining requires hard computation activities which makes the protocol very slow. A valid alternative to PoW is the Proof-of-Stake (PoS) where authorities, called *validators*, are elected according to a monetary commitment. Most PoS implementations replace randomness with voting-based protocols to elect leaders; although this choice still impacts scalability when it comes to large networks. Recent blockchains like Solana mitigate this problem by relaxing decentralisation.

Algorand introduces the Pure Proof of Stake (PPoS) protocol which solves the trilemma by applying randomness to the PoS leader election. PPoS replaces mining with the execution of Verifiable Random Functions (VRF), which are faster, and do not require wasting of resources. PPoS ensures scalability, having a small set of validators randomly selected, security, because of a crittografic sortition, and decentralisation allowing anyone to participate in the protocol.

Algorand: Everything You Need to Know. Alongside a fast, secure, and decentralised infrastructure, ALGORAND provides an array of capabilities at the same level of the consensus protocol –such as the *Algorand Virtual Machine (AVM)*, *Algorand Standard Asset (ASA)*, and *Atomic Transfer (AT)*. They are combined to build powerful decentralised applications (dApps) solving real-world use-cases.

- The AVM is a computation engine running on every node of the network. It supports the execution of smart contracts written in a language called *Transaction Execution Approval Language (TEAL)*. Algorand Smart Contracts are trustless, error-free, programs enforcing custom transaction approval logic.
- The ASA provides a standardised framework to create any type of asset on the ALGORAND blockchain without the need for complex smart contract logic. ASAs can be fungible (such as stablecoins, utility tokens, etc), non-fungible (tickets, digital art, etc.), restricted fungible assets (such as securities), and restricted non-fungible assets (such as real estate and certifications). Transfer rules can be also enforced on ASAs by creators/delegates to freeze specific accounts and/or clawback their assets back.
- The AT offers a secure way to group many transactions and execute them at once, sequentially. The execution is truly atomic, which means that the transactions are either all executed or none of them are. This feature is great for use cases like debt settlement, decentralised exchanges, delivery-versus-payment settlements.

This part of my speech will be a technical session. I will give an overview of the above ALGORAND functionalities and how they can be used. Then, I will also give a practical demonstration of how to interact with them by illustrating simple use cases.

Is the Code Law? Algorand Standpoint. In blockchains, *code is law* is a statement enforced by smart contracts representing ultimate authorities for certain operation approval. Once implemented, smart contracts are immutable logics approved by the network, hence considered as '*law*'. However, the code is not perfect: bugs are always under the corner, and attackers could hack them to break the rules of that '*law*'. For instance, in recent years hundreds of flaws have been exploited in smart contracts implementing decentralised finance (DeFi) applications. Such hacks have had dramatic consequences causing billions of dollars stolen in 2021.

Although ALGORAND provides an efficient and reliable infrastructure relying on strong technology, buggy smart contracts approving unwanted transactions are still possible. For instance, in January 2022, the Tinyman dApp –an automated market maker on Algorand– was hacked, causing the loss of approximately $3M in tokens from the contract pool. To prevent such kinds of hacks, it is crucial to define secure execution patterns and smart contract models enforcing established best practices.

In the final part of my speech, I will give an overview of the vulnerabilities that might affect the ALGORAND smart contracts. Therefore, as a case study, I will describe the Tinyman hack, i.e. how it was exploited by the hackers and how could have been avoided. Finally, I will introduce the '*Algorand Secure Coding*' project, which aims at defining structured best practices for building secure TEAL logic –such as how to handle asset transfers, group transactions, contract-to-contract calls, and more.