Algorand's Core Technology

(in a nutshell)

by Silvio Micali

A Massive Opportunity. Today, there is a massive opportunity to reboot the financial systems of the world.

Data networks are faster than ever before: a message can travel around the globe in a fraction of a second for a negligible cost.

However, money does not flow fast at all. A simple financial transaction can take days to clear and finalize. The process is expensive too: 5 trillion dollars are wasted every year to transaction fees of all types. And 2.2 billion people around the world lack access to modern financial services altogether: their transactions are simply too small to be profitable to banks.

With the right technology, we can do better.

The Blockchain Promise. For a few years, developers and innovators have had a hunch that the right technology is the blockchain; that blockchains hold the key to a more efficient and more inclusive financial system.

In simple terms, a blockchain is a public ledger. It is a sequence of transactions organized in blocks, guaranteeing three fundamental properties:

1. Everybody can read every block, so the blocks become common knowledge.
2. Everybody can write a transaction in a future block.
3. No one can alter the transactions in a block or the order of the blocks.

So rather than having a hidden database kept by a central authority--as banks have--and rather than having transactions pass through layers of secret databases in order to clear, we could have a single, public ledger that can be read by everybody, in which everybody can write, but nobody can alter what has been written.

With these properties, a blockchain's use is essentially unlimited. Indeed, blockchain technology could bring about a faster, cheaper, more secure, *borderless economy.*

An Apparent Trilemma. So far, blockchains have largely remained aspirational. Perhaps the best evidence of the aspirational nature of blockchains today is the famous *blockchain trilemma.* Sustained by the evidence of 2000 and counting blockchain projects so far, the trilemma essentially states that existing blockchains can offer at most two of the following three properties:

- Security
- Scalability
- Decentralization

Algorand™

Indeed, there are no good options in the trilemma. Without decentralization, we remain in the financial system that already exists today: exclusive and secretive. Without security, blocks of transactions can vanish: a person who has delivered goods or services may discover that the payment received for those goods and services has vanished; debts may be erased; and the public ledger may be tampered with by adversaries to their own benefit. Without scalability, you would only be able to transact with a small network; you could not participate in the global financial system. No user should have to contemplate compromising on any of these fundamental properties.

Cost, Speed & Security

To be sure, security, scalability, and decentralization are not independent variables, but quite interrelated ones. Together, they affect measures important for both individual and institutional users: namely, speed, security, and cost.

A blockchain that is not fast cannot possibly scale. But guaranteeing speed by increasing cost is not a solution: a blockchain that is too costly to operate cannot scale. Indeed, if the cost is borne by everyone, then few people will join the blockchain. And if the cost is confined to a few entities, then the system will be centralized. And any centralized system is inherently insecure because fewer targets are easier to attack than millions of targets.

The Good News. Fortunately, the trilemma is only a description of the past and a suggestion on the difficulty of achieving all three properties together. Algorand is excited to have built a blockchain that delivers all three properties.

The Blockchain Challenge. There are two different needs in a blockchain. The first is making the blockchain tamper-proof. This need has been solved via one of the simplest and oldest primitives in cryptography: the one-way hash function. Essentially, the hash of the latest block is included as part of the next block. All blockchains share this approach, so they are all equal in this respect.

The second need is generating new blocks: *how you select a new block to be appended to the chain.* This is the real challenge, and different blockchains have different approaches to it.

A new block should contain a set of valid transactions that do not appear in the blockchain so far. The problem is that at any point in time two users may have seen the same sequence of blocks but may see different new transactions. This is the case because, in a distributed ledger, each transaction is not instantaneously propagated through the network. Typically, it is transmitted from each user to a few other users, who then transmit it to still other users, until the transaction reaches all users. Accordingly, at every point in time, the sets of new valid transactions seen by different users may be different even when they have a significant overlap.

In sum, a user U may think the new block ought to be $Block_U$ and a user V thinks it ought to be $Block_V$. So: whose envisaged block should be appended to the chain?

Popular Prior Approaches & Their Fatal Flaws. Various approaches have been used for choosing the next block: in particular, *proof-of-work, delegated proof of stake*, and *bonded proof of stake*. All these approaches, however, suffer from the following fatal flaw:

> *the whole economy is at the mercy of a small part of the economy.*

This flaw is fatal as it involves both security and decentralization.

Having the destiny of the entire economy be at the mercy of a small fraction of that economy is a ticking bomb. To be sure, if such members misbehave, they devalue all assets in the economy, including theirs. But if their own assets form only a small part of the total economy, it may be easy to compensate them for their losses, adding some profit for good measure, and generate a huge damage to everybody else. No small subset of an economy should be able to control the whole economy.

Let me explain how this flaw arises in prior approaches.

Proof-of-Work

The first approach is proof-of-work, famously used by Nakamoto for Bitcoin and inherited by many other blockchains. In this approach, at a very high level, users race to solve a very complex cryptographic puzzle. The first one to solve the puzzle has the right to append the next block to the chain. Proof-of-Work suffers from several flaws.

First Flaw: Proof-of-Work Does Not Scale. Proof-of-work is very slow. Bitcoin's crypto puzzles are so hard in order to guarantee that one solution is found only every 10 minutes, no matter how many miners try to solve the crypto puzzle. We can understand expensive and fast. But expensive and slow is hard to understand. The world is a large place and one block of transactions every 10 minutes is hardly enough.

Second Flaw: Proof-of-Work Results in De-Facto Centralization. Proof-of-work has caused a tremendous concentration of power. This centralization is a consequence of the fact that Proof-of-Work is both expensive and wasteful. The amount of computation performed by the *miners*—that is, the users trying to solve the crypto puzzles— is stunning. Mining today utilizes racks and racks of specialized hardware and consumes an enormous amount of electricity. One miner wins the race and generates the new block, and the efforts of all the others are wasted. Without the subsidies that Bitcoin currently offers, the cost of posting a single transaction on Bitcoin's blockchain is around $20. Not the way to go if you want to use the blockchain for everyday transactions like buying a slice of pizza or if you want to use it offer financial services to those 2.2 billion who are currently not served by the financial system.

The common user would lose money if she tried to solve the crypto puzzle with her laptop. Win or lose, she must pay for the electricity necessary to power the computations of her laptop. This amount of electricity may not be big, but her probability of winning is so small that, in expectation, she would lose money.

Only professional miners, who have made the capital expenditure necessary to buy racks and racks of hyper specialized mining equipment, can expect to make a small profit. Accordingly, only they participate in block generation. Furthermore, miners consociate in *mining* pools.

Today, Bitcoin's blockchain is controlled by just three mining pools and Ethereum's by just two mining pools. If they so decide, or if they are bribed to do so, these mining pools can rewrite the database: they can erase blocks or change the order of blocks. Proof-of-work has turned what was intended to be a decentralized system into an extremely centralized one.

Third Flaw: Proof-of-Work Is Not Secure. As we said, any blockchain that is centralized, whether by design or *de facto*, is insecure. But proof-of-work has additional vulnerabilities, and it is especially vulnerable to network attacks. A blockchain ultimately is a communication protocol, and any such protocol is executed over an underlying communication network. An adversary may thus attack either the protocol—e.g., by sending messages that are different from the prescribed ones—or the communication network itself—e.g., by interfering with routers, cables, etc.

Just how insecure proof-of-work is may be underestimated because the current way of analyzing a blockchain's security is flawed. This analysis typically focuses only on protocol attacks and neglects network attacks that, especially in the context of proof-of-work, can be deadly. For instance, in a proof-of-work blockchain, an adversary capable of partitioning the communication network for an hour or two could double-spend with impunity. In a successful partitioning attack, an adversary prevents the messages sent by the users belonging to a set of users A from reaching the users in a separate set B, and vice versa. Network partitioning has not attracted much attention, because it is considered too expensive to be practical.

But the cost of a network attack may be justified, once the gains are high enough. A truly borderless economy may be valued in trillions of dollars. And an adversary may be willing to 'invest' millions of dollars, if he stands to illicitly gain billions of dollars.

Fourth Flaw: Forks. Another disadvantage of proof-of-work is the unavoidable existence of forks. Whenever two or more users solve the crypto puzzles within a few seconds of each other, the chain branches because users may now see multiple candidates for the next block. A fork may continue to exist for a while, and all its branches may even be elongated by the addition of new blocks. But eventually, all branches but one will die, and all the blocks in the dead branches will disappear.

Forks are an unwelcome source of uncertainty and delay. If a payment made to you appears in the latest block added to the chain, you cannot consider yourself paid and ship the goods. This is so because some branch may overcome the current chain and your block may end up in a dead branch and vanish. Before considering yourself paid, you would need to wait for a sequence of blocks to be added to yours, so as to minimize the chance that a soft fork will arise and the block containing your payment will end up on a dead branch.

How long should you wait for? Some people recommend six blocks to be added after yours to be confident that your block will remain on the chain. Others recommend an even longer wait, if the payment made to you is sizable. Thus, rather than waiting ten minutes, to have reasonable confidence in the finality of a transaction, in reality you have to wait hours.

Some people have suggested making the crypto puzzles easier in order to speed up the process, for instance by making it possible to find a solution every minute, rather than every 10 minutes. However, by doing so, the probability of getting two solutions within a few seconds of each other increases significantly. The system can cope with an occasional soft fork, but not with very frequent forks.

Expenses, slowness, and uncertainty are indeed major flaws in the proof-of-work approach, but they pale in comparison with its fatal flaw.

The Fatal Flaw in Proof-of-Work. Recall the already discussed fatal flaw: *the whole economy is at the mercy of a small part of the economy.*

In proof-of-work, this small part of the economy is that owned by the miners. Since the miners own only a small fraction of the money in a proof-of-work blockchain, the chain is not secure.

Delegated Proof-of-Stake

A different approach is delegated proof-of-stake (PoS). This is a very simple idea. The community empowers a few special users, the delegates, to choose the next block, at least for a while. (For example, in EOS, the number of the delegates is 21.)

Delegated PoS is, therefore, centralized from the get-go. Hopefully, the chosen delegates are honest to begin with. However, relying on delegates remaining honest for a long time is risky.

Once again, we have that *the whole economy is at the mercy of a small part of the economy*. Indeed, in a delegated-PoS blockchain the delegates may own a tiny fraction of the total money in the system, yet the whole blockchain is secure if and only if the majority of delegates are honest.

Additional Security Problems. Even assuming that there is an ironclad guarantee that all the delegates will remain honest forever, they can easily be attacked. In particular, they can be brought down by a denial of service (DoS) attack. In such an attack, an adversary bombards any user of his choice with zillions of junk messages, causing the buffer of that unfortunate user to overflow. If a delegate were so bombarded, he would be unable to perform his job, namely collating new and valid transactions into the next block. The blockchain would grind to a halt.

DoS attacks are quite cheap and can be mounted instantly against not only 21 people but even 1000 people. Since delegates are known, even if they were kept in power for just a day or an hour or a minute, a determined adversary could bring down all the delegates by a fast DoS attack.

Bonded Proof-of-Stake

Bonded PoS allows 20 users, 200 users, as many as are willing, to put some money on the table—a bond—where they can no longer touch it. These are the users who select the next block on behalf of all of us. If they misbehave, their money is confiscated.

Does this approach work?

Let me ask a simpler question: how much of your disposable income can you afford to put hostage on the table? The answer is a very small amount. Bonded PoS therefore, not only makes it possible, but actually makes it easy for big thieves with deep pockets to put a disproportionate amount of money on the table for the sole purpose of controlling the blockchain.

But so what? If they misbehave, they lose their hostage money. However, a truly decentralized, scalable and secure blockchain should secure trillions of dollars in assets. And by misbehaving, a malicious user stands to make a few billion dollars. This being the case, do you think that he may fear having a few million dollars confiscated? This is just the price of doing business. And it is a small price at that.

Once more, in bonded PoS, we have the same fatal flaw: *the whole economy is at the mercy of a small part of the economy.* Indeed, in a bonded PoS this small part of the economy consists of (the owners of) "the money laying hostage on the table."

In sum, prior approaches suffer from several drawbacks. We need a better design.

Algorand's Logic and Pure Proof-of-Stake. Algorand's logic is simple: it ties the security of the whole economy to the honesty of the *majority* of the economy, and makes it impossible for a small subset of the economy to control the fate of the whole economy.

Algorand is based on a new Proof-of-Stake: Pure PoS. Essentially, a Pure PoS does not try to keep users honest by the fear of imposing fines. Rather, it makes cheating by a minority of the money impossible and cheating by a majority of the money stupid.

In Algorand no money is ever hostage. All money is always where it should be: in your wallet, at your fingertips ready to be spent, or in the various financial instruments that the Algorand blockchain offers you. When you take in consideration all the money in the system, wherever it may be, the system is secure when most of the money is in honest hands.

As we said, it is impossible for the owners of a small fraction of the money to harm the whole system, and it would be silly for the owners of the majority of the money to misbehave so as to devalue their own holdings.

For example, in proof-of-work or bonded PoS, a few users can prevent other users from transacting. In Algorand, only the owners of the majority of the money could prevent other users from transacting. But if they did so, the reputation of the currency would be greatly harmed, the currency would no longer universally accepted, and its purchasing power would be greatly diminished. Not a good outcome for the owners of the majority of the money.

Implementing Pure PoS. Let us now see how Algorand uses pure PoS for choosing the next block. Remember the trilemma? We want block generation to be simultaneously scalable, secure, and decentralized.

At a very high level, in Algorand, a new block is constructed in two phases.

- In the first phase, a single token is randomly selected, and its owner is the user who proposes the next block.

- In the second phase, 1000 tokens are selected among all tokens currently in the system. The owners of these 1000 tokens are selected to be part of a phase-2 'committee,' which approves the block proposed by the first user.

  Accordingly, some member of the committee may be chosen twice or more generally some *k* times, in which case, that member will have *k* votes in the committee to approve the next block.

Why is the second phase necessary?

In any society, and blockchains are no exception, there is always a small percentage of bad actors to be found; let us say 1 percent. Maybe 2 percent. If one is unfortunate enough to live in a very dangerous society, 10 percent may be bad actors. Perhaps even 20 percent! But in no society will bad actors be in a majority; otherwise, there would not be a society. A society exists insofar as the majority of its members follow prescribed rules.

Assume that 10 percent of the tokens in Algorand belong to dishonest people. Then, one in ten times, the user selected in phase 1 to propose a block may be a bad actor. Accordingly, he may tell some users that the block is *X* and other users that the block is *Y* and so on and so forth, thereby creating disagreement about what the blockchain is.

Phase 2 eliminates this problem. Indeed, if you select at random 1000 tokens when at most 10 percent of the tokens are in dishonest hands, the probability that the majority of the selected coins belong to bad actors, that is, the probability that the majority of the committee's votes are cast by bad actors is so low as to be negligible.

Assume that, this time around, you have not been selected to propose a block. Nor have you been selected to be part of a committee to approve a proposed block. But you see that a given block *B* has been approved by, say, 700 votes of the committee. Then, you know that *B* is indeed the next block.

A Key Question. At this high-level description, several questions naturally arise. Let us start with the most obvious one: *Who randomly chooses the committee?*

Suppose I tell you that I do. Then you might say "That is the most centralized system ever, and you are at the center of it!" Suppose I tell you that all users discuss until they agree on a thousand committee members who then agree on the block. Then you might tell me that, humanity being what it is, an entire lifetime is not going to be enough to select the thousand committee members that we require.

Algorand takes an unorthodox approach: *the committee members select themselves.* You may think "What? That is a terrible idea! Because if I am a bad actor, I am going to select myself to be a member of this committee. And the next. And the one after that..." But not so fast.

To belong to the committee, one of your coins must win an individual, cryptographically fair lottery that you run in isolation—that is, without talking to anyone else—in the privacy of your own computer. And since the lottery is cryptographically fair, you cannot alter the chances of being selected in the slightest. (Not even a nation state with huge computational resources would be able to increase the probability of being selected.)

To select 1,000 random tokens among, say, 10,000,000,000 tokens, each token is selected with probability 1,000/10,000,000,000—i.e., with probability 1 in 10 million.

So, as soon as a user sees a block being proposed, she asks herself: can I be a member of the committee selected to approve a block? And how many votes will I have?

To answer these questions, she runs the cryptographic lottery on her laptop for each one of the tokens that she owns. (If a user has *n* tokens, additional technology essentially allows her to run a single lottery instead of *n* separate lotteries!)

Once a user runs her lottery, one of two cases occurs. Either none of her tokens wins the lottery, in which case whatever opinion she expresses about the block will be ignored. Or some $k>1$ of her tokens win the lottery, in which case, she obtains a winning ticket, namely a short proof that everybody can easily verify that she has *k* votes in the committee. In this latter case, she propagates through the network (i) a winning ticket proving that she has *k* votes and (ii) her opinion about the block.

Solving the Trilemma. Let me argue that this sketched approach is—finally!— simultaneously scalable, secure, and decentralized.

Scalability

How long does it take for a user to run her own lottery? Roughly a microsecond, no matter how many tokens she has. That is indeed superfast. (Additionally, all the lotteries are run independently of each other, so that no user has to wait for other users to finish running their lotteries.)

Once selected, each member propagates to the network a single, short and immediately computed message. So no matter how many users are in the system, the maximum number of messages that need to be propagated is 1000 short messages. Is this scalable? Yes!

Security

Now we come to security. Assume that I am a very powerful adversary, capable of corrupting users extremely quickly whenever I want. Clearly, I would love to corrupt the members of the committee, but I have a problem: I do not know who they are.

This is so because committee members are selected by a secretly run, cryptographically fair, individual lottery. Thus, only they know who they are, up to the moment in which they propagate through the network both their winning tickets and their opinions about the block. Only at that time I might learn who the committee members are and, given my super powers, I can immediately corrupt the entire committee. But so what? Corrupting them at this time is too late. Whatever the committee members had to say, they have already said, and their winning tickets and up-or-down opinions about the block are virally propagating throughout the network. I have no more power to put their messages back in the bottle than a government has the power to put back in the bottle messages virally propagated by WikiLeaks.

In other words, the Algorand approach is secure because beforehand, an adversary does not know whom to corrupt, and by the time he does, corruption is useless.

Contrast this with having a fixed 1000-strong committee. As discussed before, even if the committee stayed in power for a minute, it would be vulnerable to a DoS attack. If the committee stayed in power for much longer, say a week, then the members could even be corrupted in the physical world via traditional means such as bribes. In Algorand's case however, one would not know against whom to mount a DoS attack and once the committee has spoken, the DoS attack is useless.

Decentralization

Finally, we come to decentralization. Are there a few users in charge of choosing the next block? No, there are not. Nor is there a fixed, 1000-strong, committee in charge of approving the block. This time, a committee has been randomly (and secretly) selected. Next time, a different committee will be randomly (and secretly) selected. Everybody has the chance of participating in the generation of a new block.

Algorand's Non-Forkable Chain

An additional advantage of Algorand's technology is that its chain never forks. This is so because only one block can have the required threshold of committee votes. Accordingly, in Algorand all transactions are final. Once a block appears, you can count on it to be forever part of the chain. And if the new block contains a payment made to you, you may consider yourself paid and send the goods immediately.

The financial world has already its own risks and there is no need to burden it with the additional uncertainty of `block disappearance.' By the way, when I said that Algorand's chain never forks, I somewhat lied. Indeed, forks may occur in Algorand but they are very rare. The probability of forking in Algorand is, by design, 10^{-18}. This probability may appear a strange choice, but it actually has a natural explanation. Physicists tell us that 10^{18} happens to be the number of seconds from the Big Bang until now. In other words, if you produce a block a second, a very good clip by the way, you may see a soft fork, but you would have to wait for the lifetime of the universe to see it.

Algorand™

Silvio Micali

Founder

Silvio Micali has been on the faculty at MIT, Electrical Engineering and Computer Science Department, since 1983. Silvio's research interests are cryptography, zero knowledge, pseudorandom generation, secure protocols, and mechanism design.

In 2017, Silvio founded Algorand, a fully decentralized, secure, and scalable blockchain which provides a common platform for building products and services for a decentralized economy. At Algorand, Silvio oversees all research, including theory, security and crypto finance.

Silvio is the recipient of the Turing Award (in computer science), of the Goedel Prize (in theoretical computer science) and the RSA prize (in cryptography). He is a member of the National Academy of Sciences, the National Academy of Engineering, and the American Academy of Arts and Sciences.

Silvio has received his Laurea in Mathematics from the University of Rome, and his PhD in Computer Science from the University of California at Berkeley.