Security of Ethereum Layer 2s

Ionuț Roșca[1], Alexandra-Ina Butnaru[2], and Emil Simion[3]

[1] Faculty of Computer Science, Alexandru Ioan Cuza University of Iaşi, România
[2] Faculty of Computer Science, Alexandru Ioan Cuza University of Iaşi, România
[3] Politehnica University of Bucharest, România

Abstract. Since the proposal of Bitcoin in 2008, the world has seen accelerated growth in the field of blockchain and discovered its potential to immensely transform most industries, one of the first and most important being finance. The blockchain trilemma states that blockchains can have security, scalability, and decentralization, but never all three at the same time, in the same amount. At the moment, the most successful blockchains have a lack of scalability that researchers and developers try to alleviate by solutions like layer 2s. Most of these solutions rely on cryptographic primitives and technologies, like collision-free hash function or zero-knowledge proofs. In this paper we explore a few of the most popular solutions available now, their improvements to scalability, their drawbacks and security risks.

Key words: blockchain, ethereum, layer 2s, security

1 Introduction

Blockchains are digital ledgers with no central authority (i.e. bank, company, government) implemented in a tamper-resistant, distributed way (without a central repository). It allows the users to anonimously record transaction on a ledger shared by the community with no practical chance of changing its history. In 2008 the concept together with other cryptographic and technological ideas were used in order to create modern cryptocurrencies: electronic cash protected by mathematical mechanisms instead of a central authority.

Although the first such blockchain based cryptocurrency was Bitcoin, its lack of scalability, support for smart contracts and exaggerated consumption of resources has made other cryptocurrencies gain popularity and support, such as Ethereum.

Ethereum[1] is a public blockchain and the first one to introduce what are known as *smart contracts*. Smart contracts are programs which run on the Ethereum blockchain. The idea is the same as running a program on your own computer, but instead of your own computer, a decentralized network of computers is used and the program is run by validator nodes.

Before we can talk about the security of Ethereum Layer 2s, we must first understand what we mean by "Layer 2". According to Vitalik Buterin[2], there are two ways to scale a blockchain ecosystem:

1. The first solution is to scale the blockchain itself in order to have a higher transaction capacity. The blockchain itself is called the "Layer 1 (L1)". In our case, the Layer 1 is Ethereum.
2. The second solution implies a change in the way users use the blockchain. Instead of performing all the activity directly on the blockchain, users perform the majority of their activity off-chain i.e. external to the blockchain, in a Layer 2 protocol. On the Layer 1 there is a smart contract that processes deposits and withdrawals, but also verifies that the activity happening off-chain (on the Layer 2) is following the rules; this is achieved by using proofs. There are multiple ways to do the proofs, but they all share the property that verifying the proofs on-chain is much cheaper than doing the original computation off-chain.

2 The Blockchain Trillema

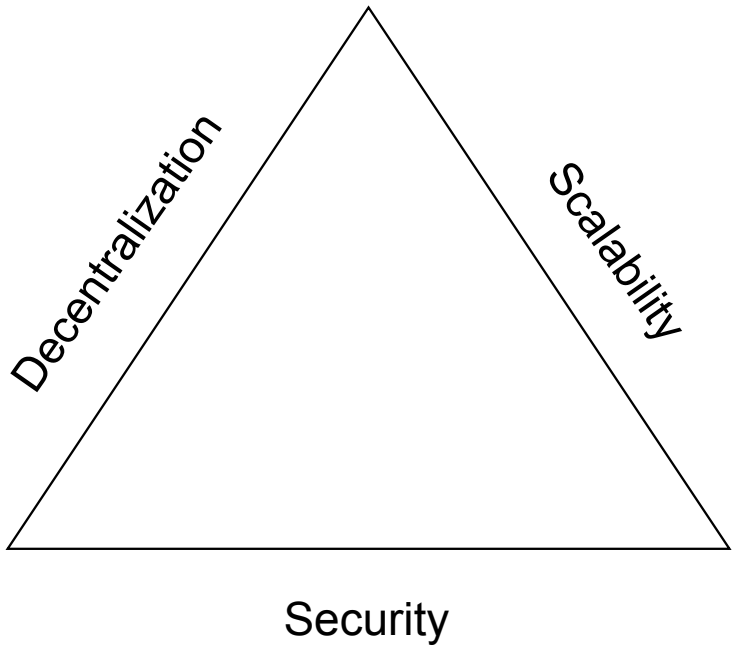

Fig. 1. The blockchain trillema illustrated.

The blockchain trilemma refers to the idea that a public blockchain cannot reach the desired level of decentralisation, security, and scalability all at once. *Scalability* is the ability of the blockchain to accommodate a higher volume of transactions, *security* is the ability to protect the ledger from different attacks

or blockchain's defence against double-spending, *decentralisation* is the redundancy in the network that makes sure fewer entities do not control the network. The fundamental design of many decentralized networks means that increasing scalability tends to weaken decentralization or security.

Blockchains can perform a limited amount of transactions per second, also referred as TPS, which at the present time is too low to allow large scale adoption as an alternative to already-existing and centralized technologies. For instance, VISA is capable of processing more than 24,000 TPS[8]. This is in contrast with the actual performance of the public blockchains such as Bitcoin and Ethereum, which score an average of 7 TPS and 20 TPS, respectively. The TPS rating is highly dependent on the level of congestion of the blockchain. In times of high congestion, the TPS rates can decrease.

Layer 1 blockchain solutions help to improve the base protocols by changing how they operate as regards processing data. For example, the Ethereum network has moved in 2022 from a Proof-of-Work (PoW) to a proof-of-stake (PoS) consensus algorithm. This new method of mining supports faster transaction speeds and more efficient energy use in the mining process.

Another layer 1 scaling solution is *sharding*, which breaks down the job of authenticating and validating transactions into smaller pieces. It spreads the workload better across the peer-to-peer (P2P) network to bring in more computing power from more nodes, allowing for blocks to be completed faster.

However, layer 1 solutions are often not efficient enough to meet the expectations for blockchains' scale. Layer 2 scaling solutions increase throughput without tampering with any of the original decentralization or security characteristics that are integral to the original blockchain.

3 Layer 2 Technologies

Some layer 2 scaling solutions derive their security directly from the base layer, Ethereum, such as *optimistic rollups*, *zero-knowledge rollups* or *state channels*. Other solutions involve the creation of new chains in various forms that derive their security separately from Mainnet, such as *sidechains*, *validiums*, or *plasma chains*. *Mainnet* refers to the public and available for users instance of a blockchain or protocol on the blockchain. The ETH currency on mainnet has real-world value. Ethereum also has *testnets* which are instances of Ethereum, meant to be used publicly by users for testing purposes. The ETH currency on testnets does not have any real-world value.

Plasma and *sidechains* move both data and computation off-chain. Game theory data availability concerns tells us that this is not a feasible solution for all applications. *Rollups*, on the other side, use a hybrid approach, moving computation and state storage off-chain, but keeping some data per transaction on-chain and compressing as much as possible by replacing it with computations. A key point here is data being on-chain, therefore anyone can locally process the operations in the rollup, malicious nodes can no longer do harm by delay and there is no need to map assets to owners, which makes rollups general-purpose.

Rollup layer 2 solutions make use of a contract responsible with maintaining the state root (the Merkle root of the state of the rollup). Whenever someone wants to publish a collection of compressed transactions, called a *batch*, it will contain the state root present in the contract and the Merkle tree root after processing the transactions. The rollup contract will check its current state root with the previous one contained in the batch, and in case of a match it will accept it and update the state root.

In case of deposits and withdrawals with inputs or outputs outside the rollup, the contract needs to take responsibility. If the batch has inputs outside, the transactions that wants to publish the batch will have to transfer the assets to the contract. If the batch has outputs going outside the rollup, the program will initiate the withdrawals upon processing the batch.

Due to the fact that anyone can publish a batch, the correctness of post-batch state root needs to be ensured, otherwise the node submitting the batch could very well transfer all the assets inside the rollup to themselves. There are two main approaches to this, which have given the two main types of rollups: optimistic rollups and ZK rollups. These scalability solutions can realistically lead to about 500 TPS in the case of optimistic rollups, and more than 2000 TPS for the ZK counterpart.

3.1 Sidechains

Sidechains cannot be technically called Layer 2s. However, they are used as a scaling solution and many sidechains for Ethereum do exist. A sidechain has *it's own security properties* and *it's own consensus mechanism*, thus it does not rely on security of the base layer. Practically, sidechains are blockchains that run in parallel with the main chain.

In order to move assets between a sidechain and the main chain, bridges are used. The type of bridged used, however, is a matter of choice of the developers. According to the Ethereum documentation[3], while there are many types of bridge designs, three ways to facilitate the cross-chain transfer of assets stand out:

- **Lock and mint** - Lock assets on the source chain and mint assets on the destination chain.
- **Burn and mint** - Burn assets on the source chain and mint assets on the destination chain.
- **Atomic swaps** - Swap assets on the source chain for assets on the destination chain with another party.

Given the fact that sidechains are not Layer 2s, we will not focus on them in this paper.

3.2 State Channels

State channels provide a way to conduct activity that would normally be conducted on the main chain, on a separate channel. When the users stop conducting further activities, they will settle only the results of all the activities

performed on the state channel up to that point, to the main chain. For example, imagine that two players are playing a chess game. In order to win the game, both of the players need to make move. In this scenario, the players will make all the moves inside a state channel and they will post only the result of the game on the Layer 1. In this way, the users only need to pay the depositing transaction (in this scenario, the users wanting to play a chess game) and the withdrawal transaction (in this scenario, a user won the game).

In practice, a user A locks an amount of ETH into a smart contract. The user A then signs an off-chain message that transfers x amount of ETH to user B. Then, user B does the same for user A, and so on. This exchange can continue as long as both parties want that. When a user wants to end the exchange, they will publish a signed exit message to the Layer 1. The smart contract will verify both of their signatures and if the verification is successful, the final balances for both parties will be settled on Ethereum.

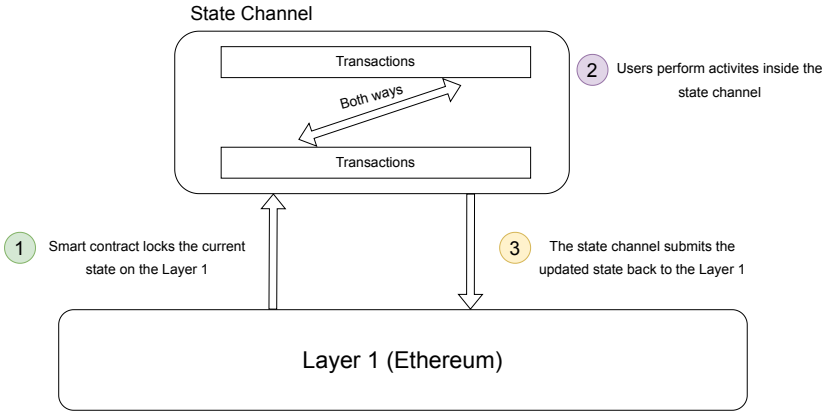

Fig. 2. A simple illustration of a state channel.

State channels are not limited only to Ethereum. Bitcoin has its own Layer 2 that uses state channels: Lighthing Network[4].

3.3 Plasma

A Plasma chain is a separate blockchain anchored to Ethereum Mainnet but executing transactions off-chain with its own mechanism for block validation. Plasma chains are sometimes referred to as "child" chains, essentially smaller copies of the Ethereum Mainnet. Practically, a Plasma chain is made of Merkle trees and smart contracts. This allows Plasma to create unlimited "child" chains. Developers can build multiple chains on top of each child chain to create a tree-like structure.

Merkle trees are named after Ralph Merkle. Bitcoin and other cryptocurrencies use Merkle trees, also called hash trees, to encode and encrypt blockchain

data efficiently and securely. Merkle trees are created by repeatedly calculating hashing pairs of nodes until there is only one hash left: the root hash or Merkle root, which is a summary value. They are constructed using a bottom-up approach in which each transaction is hashed, then each pair of transactions is concatenated and hashed together, and so on until there is one hash for the entire block[6].

Plasma chains use fraud proofs to arbitrate disputes[5]. Fraud Proofs present evidence that a state transition was incorrect. They reflect an *optimistic* view of the world: the assumption is that blocks represent only correct states of L2 data, until proven otherwise. In reality, a committed block could well include an incorrect state transition[9].

Plasma Withdrawals and Deposits

To make a deposit, a user sends the asset to the smart contract that manages the Plasma Chain. The Plasma Chain will assign a unique ID to the asset. An operator then generates a batch of Plasma transactions they have received off-chain at intervals.

To withdraw an asset, the contract starts a "challenge period." During this time, anyone can use Merkle branches to invalidate the withdrawal if they can prove the exit is fraudulent. After the challenge period is up, the user can withdraw the asset.

One advantage Plasma has over State Channels is that the capital requirements are a lot lower. Also, a user can send assets to participants who are not part of the system. The "mass exit" problem is one of the biggest concerns with Plasma. If many users exit their Plasma chain simultaneously, they could flood the root chain and congest the network. Things like fraudulent activity or network attacks could cause such a mass exodus.

Another "con" of plasma chains is a lack of complexity. Users can't perform the same kinds of complex actions as they can on Sidechains. That's because of the necessary precautions in place to keep user's funds secure.[10]

3.4 Optimistic Rollups

Optimistic Rollups use *fraud proofs* in order to guarantee the validity of the chained rollups. They are called optimistic because there is no proof published on the blockchain in order to ensure the validity of the off-chain transactions, they are assumed to be. The contract responsible with keeping the history of state roots and the hash of each batch will revert an incorrectly computed one (and all subsequent batches) if anyone publishes to the chain a proof that the post-state root is wrong in a limited window of time called the *challenge period*.

Some of the challenges we see with optimistic rollups are moving assets and data from one rollup into another without paying the price of going through the base layer and motivating nodes to verify rollups in full, as we cannot rely on altruism when the system starts to scale.

Optimism

Optimism is a rollup layer-2 scaling solution for Ethereum which helps reduce the transaction fees and time on the Ethereum chain without sacrificing decentralization and security. It uses single-round fraud proofs executed on layer-1 in order to ensure the validity of a transaction batch.

Because fraud proofs are still developing, the system currently permits invalid state roots, which is a critical vulnerability, as funds can be stolen in this scenario. Also, if the centralized validator goes down, users cannot produce blocks themselves and exiting the system requires new block production, leading to possible freezes of funds.

Arbitrum

Arbitrum [12, 13] is an optimistic rollup running on Arbitrum Virtual Machine (AVM), that uses multi-round fraud proofs executed off-chain. It allows for high network performance due to its single point of disagreement and offers better security than other solutions like Optimism because of its more complex multi-round fraud proofs.

A centralized sequencer receives users' transactions and regularly sends the batch to the mainnet. Independent Validators (currently whitelisted, a trade-off of descentralization for scalability) read transaction batches from L1, execute them and submit a resulting layer 2 state root to the layer 1 chain.

The biggest security risk with Arbitrum is that none of the whitelisted verifiers checks the published state because at present time, fraud proofs assume at least one honest and able validator.

3.5 ZK Rollups

Unlike optimistic rollups which rely on fraud proofs, ZK-rollups rely on zero-knowledge proofs [11] to demonstrate with cryptographic certainty that the proposed changes to Ethereum's state are truly the end-result of executing all the transactions in the batch.

ZK rollups rely on the mainnet for the following:

- *Data availability*: for every transaction processed off-chain, some of the data is published on Ethereum, allowing anyone to reproduce the rollup's state and validate the chain; this is important because it allows permissionless, independent verification of the L2 chain's state
- *Transaction finality*: once the L1 accepts the validity proof, the transaction is finalized, and that once approved, a transaction cannot be reversed, therefore Ethereum functions as a settlement layer for ZK rollups
- *Censorship resistance*: for efficiency, most ZK rollups make use of an operator node ("supernode") that executes transactions, produces batches, and submits blocks to L1, but as it chooses which transactions are and are not included in a batch, censorship questions arise; for safety reasons, nodes are free to submit

transactions directly to the rollup contract on Mainnet if they think they are being censored by the operator

In some ZK rollup solutions, the operator is a centralized entity, called a sequencer, who executes transactions, aggregates them into batches, and submits to Ethereum. In this scenario, the sequencer is the only entity allowed to produce L2 blocks and add rollup transactions to the ZK-rollup contract.

In solutions where the operator role rotates between nodes, a set of validators is used in a Proof-of-Stake fashion. Potential operators deposit funds in the rollup contract, the size of their stake influencing their chances of being selected. In case of malicious behaviour, the operator's stake will be slashed. This mechanism is used as incentive for the nodes to post valid blocks.

Other ways of selecting the operator include: sequencer auction, where an auction is held (e.g. every day) to determine who will be the sequencer for the next day, DPoS voting (there is a single sequencer selected with an auction but if they perform poorly token holders can vote to kick them out and hold a new auction to replace them), or total anarchy.

Probably the most popular ZK proof protocols at the moment are ZK-SNARK, which stands for Zero-Knowledge Succinct Non-interactive Argument of Knowledge, which relies on elliptic curves and probabilities for security, PLONK [14] (Permutations over Lagrange-bases for Oecumenical Noninteractive arguments of Knowledge) which use Kate commitments and ZK-STARK (acronym for Zero-Knowledge Scalable Transparent Argument of Knowledge) which are similar to ZK-SNARKs, except that they rely on hash functions, similar to proof-of-work blockchains, offering more significant quantum resistance. At the moment, STARKs have larger proof sizes, an increased electrical demand to power the network and longer timeframes in terms of verifying the proofs and are considered more scalable and transparent than SNARKs.

Polygon Hermez

Hermez is an open-source ZK-Rollup that aims to be optimized for security and cost. There is no central operator, the system runs an auction in which anyone can bid to become the operator for a set number of blocks. Users can be censored if the operator refuses to include their transactions and users lack resources to propose blocks themselves, so a main security aspect is the need to trust the operators because when the user does a regular or forced withdraw and their funds exceed a certain threshold a timer activates, the operators can trigger emergency mode and transfer the user's funds to the governance.

Although the system implements a 7 day delay in the enforcement of code upgrades, it is still vulnerable to funds being stolen as a consequence of an uncaught malicious code upgrade.

ZKSync

ZKSync is the first EVM-compatible ZK-rollup and it makes use of SNARK cryptographic validity proofs to provide scalable and low-cost transactions on Ethereum.

In order to set up a ZK-SNARK there must be a trusted group of developers in order for code not to be manipulated and vulnerability information divulged, and although it is done only once, the setup phase undermines the protocol's decentralization.

Other security issues involve the central operator (a live and trustworthy operator is vital to the health of the system because it is the only entity that can propose blocks), the vulnerability in case of a quantum computer attack and the posibility of force exit: if a user experiences censorship from the operator with regular exit they can submit their withdrawal requests directly on L1 and the system is then obliged to service this request.

StarkNet

StarkNet is a general purpose ZK rollup built using STARK cryptographic proof system, currently using a single Sequencer. ZK-STARKs remove the need for a trusted setup by using publicly verifiable randomness to create trustless verifiable systems and are also quantum-resistant and more scalable in computational speed and size than ZK-SNARKs.

The main security vulnerabilities arise from the immediate code upgrades (which can be malicious and lead to stolen funds), incorrect implementation of the proof system, and the unique operator (which can censor users by not including their transactions or freeze funds by censoring withdrawals).

3.6 Validium

Validium is a scaling solution that enforces integrity of transactions using validity proofs like ZK-rollups, but doesn't store transaction data on the Ethereum Mainnet. These "validity proofs" can come in the form of ZK-SNARKs (Zero-Knowledge Succinct Non-Interactive Argument of Knowledge) or ZK-STARKs (Zero-Knowledge Scalable Transparent ARgument of Knowledge). While off-chain data availability introduces trade-offs, it can lead to massive improvements in scalability (validiums can process $\sim 9,000$ transactions, or more, per second)[15].

Funds belonging to validium users are controlled by a smart contract on Ethereum. Validiums offer near-instant withdrawals, much like ZK-rollups do; once the validity proof for a withdrawal request has been verified on Mainnet, users can withdraw funds by providing Merkle proofs. The Merkle proof validates the inclusion of the user's withdrawal transaction in a verified transaction batch, allowing the on-chain contract to process the withdrawal.

However, validium users can have their *funds frozen and withdrawals restricted*. This can happen if data availability managers on the validium chain

withhold off-chain state data from users. Without access to transaction data, users cannot compute the Merkle proof required to prove ownership of funds and execute withdrawals. This is the major difference between validiums and ZK-rollups: their positions on the data availability spectrum.

References

1. Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform. Vitalik Buterin (2014). https://ethereum.org/669c9e2e2027310b6b3cdce6e1c52962/Ethereum_Whitepaper_-_Buterin_2014.pdf
2. An Incomplete Guide to Rollups. Vitalik Buterin. https://vitalik.ca/general/2021/01/05/rollup.html
3. Ethereum. Bridges — Bridge Types. https://ethereum.org/en/developers/docs/bridges/#bridge-types
4. Joseph Poon, Thaddeus Dryja. The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments. Version 0.5.9.2, January 14, 2016. https://lightning.network/lightning-network-paper.pdf
5. Ethereum. Plasma chains. https://ethereum.org/en/developers/docs/scaling/plasma/
6. Sáez de Ocáriz Borde, Haitz. (2022). An Overview of Trees in Blockchain Technology: Merkle Trees and Merkle Patricia Tries.
7. Plasma: Scalable Autonomous Smart Contracts https://plasma.io/plasma-deprecated.pdf
8. VISA. Small Business Retail. https://usa.visa.com/run-your-business/small-business-tools/retail.html
9. Avihu Levy, Uri Kolodny. Starkware. https://medium.com/starkware/validity-proofs-vs-fraud-proofs-4ef8b4d3d87a
10. Moralis Academy. Comparing Layer-2 Ethereum Scaling Solutions. https://academy.moralis.io/blog/comparing-layer-2-ethereum-scaling-solutions
11. Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. SIAM J. Comput., 18(1):186–208, 1989.
12. Harry Kalodner, Steven Goldfeder, Xiaoqi Chen, S. Matthew Weinberg, and Edward W. Felten, Princeton University. Arbitrum: Scalable, private smart contracts. https://www.usenix.org/system/files/conference/usenixsecurity18/sec18-kalodner.pdf
13. Offchain Labs, Inc. Arbitrum Nitro: A Second-Generation Optimistic Rollup. https://github.com/OffchainLabs/nitro/blob/master/docs/Nitro-whitepaper.pdf
14. Gabizon, A., Williamson, Z. & Ciobotaru, O. PLONK: Permutations over Lagrange-bases for Oecumenical Noninteractive arguments of Knowledge. (Cryptology ePrint Archive, Paper 2019/953,2019), https://eprint.iacr.org/2019/953,https://eprint.iacr.org/2019/953
15. Ethereum. Validium. https://ethereum.org/en/developers/docs/scaling/validium/