Security Assumptions in Permissionless Blockchains and Layer 2 Scalability Solutions: A Comprehensive Taxonomy

Adrià TORRALBA-AGELL[a,c,1,*], Cristina PÉREZ-SOLÀ[b,c,2]

[a]*Internet Interdisciplinary Institute (IN3), Universitat Oberta de Catalunya*
Rambla del Poblenou, 154, 08018 Barcelona (Spain)
[b]*Universitat Autònoma de Barcelona,*
[c]*CYBERCAT - Center for Cybersecurity Research of Catalonia,*

Abstract

Blockchains face significant scalability limitations that currently prevent their mass adoption, despite their contribution to the transformation across various sectors. Layer 2 (L2) solutions emerge as a promising avenue to alleviate these limitations, offering scalability enhancements while trying to retain the core security guarantees of underlying blockchains. However, despite their increasing prominence, the security assumptions underpinning Layer 2 solutions have not been sufficiently scrutinized. Addressing this gap, this paper contributes a rigorous examination of the security assumptions inherent in the most used blockchain Layer 2 scalability solutions (Payment Channel Networks, Optimistic Rollups, Zero-Knowledge Rollups, Validium, and Volition), by compiling and specifying all security assumptions that permissionless blockchains rely on, as well as, the additional security assumptions that Layer 2 solutions currently need on top of the inherited ones. Our analysis reveals that Layer 2 solutions need additional security assumptions beyond those already established by their underlying blockchains (Layer 1, or L1). By delineating these assumptions and their implications, our research facilitates a deeper understanding of the security challenges in blockchain scalability solutions.

Keywords: Blockchain, Scalability, Security, Payment Channels, Zero-Knowledge Rollups, Optimistic Rollups, Validium, Volition

1. Introduction

The emergence of the Bitcoin white paper in 2009 [7] enabled the creation of blockchain technology. This technology's extensive adoption and implementation have brought about numerous transformations in our interactions with the world. These range from the creation of secure and decentralized digital currencies [7] to the establishment of equitable and secure voting systems [8], and even encompass the facilitation of trustworthy digital identification across the Internet [9].

This surge in popularity gave rise to a multitude of applications, including decentralized applications [10], decentralized finance [11], non-fungible tokens [12], blockchain-based games [13], and AI-enhanced blockchain systems [14], all leveraging this innovative technology. However, this growth also exposed many blockchains to severe congestion, resulting in suboptimal performance and high transaction fees. Notably, the Ethereum network encountered average transaction fees of almost $200 during peak periods in May 2022 [15].

In response to these challenges, a variety of scalability solutions have been proposed to enhance blockchain performance. While some have proven successful, others have failed—often due to security issues—sparking heated debates within the community that ultimately led to hard forks and the creation of new cryptocurrencies. Concrete examples of these security lapses can be found at rekt.news or hacked.slowmist.io, which catalogs hacks and exploits typically used to drain funds from blockchain systems, particularly through Layer 2 extensions. According to a leaderboard of the most lucrative hacks involving blockchain systems [16], 31 of the top 50 occurred between 2022 and April 2024[3], and many involved bridge smart contracts[4] connecting Layer 1 and Layer 2. Given these ongoing incidents, it is crucial to understand the security assumptions that underpin both Layer 1 and Layer 2 blockchain systems.

A motivating example of the risks associated with Layer 2 solutions is the 2022 hack of the Ronin network [17, 18], the Ethereum side chain developed to support the play-to-earn game *Axie Infinity*. As the game grew in popularity, the Ethereum main chain became too slow and expensive to handle the high volume of in-game transactions, prompting developers to deploy a Layer 2 solution to reduce fees and improve scalability. The Ronin bridge allowed users to move assets between

*Corresponding author
Email addresses: atorralbaag@uoc.edu (Adrià TORRALBA-AGELL), cristina.perez@uab.cat (Cristina PÉREZ-SOLÀ)
[1]Website: https://0xAdriaTorralba.github.io
[2]Website: https://cpsola.com

[3]16 in 2022, 13 in 2023, and 2 in 2024.

[4]A bridge smart contract is a program that connects two or more blockchains and manages the transfer of assets or data between them. It typically locks or burns tokens on one chain, while minting or releasing corresponding tokens on another chain. This process ensures a secure and reliable way for blockchains to communicate and share resources without trusting external intermediaries.

Table 1: Comparison of our work against the current literature.

Work	L1 Security Analysis	Discuss L1 Security assumptions	L2 Security Analysis	Discuss L2 Security assumptions	Empirical Results
Wang et al. (2019) [1]	✓	✗	✗	✗	*Only for L1*
Soni et al. (2019) [2]	✓	✗	✗	✗	✗
Yu et al. (2022) [3]	✓	✗	✗	✗	✗
Gangwal et al. (2022) [4]	✗	✗	*Only for PCNs*	✗	*Only for L2s*
Koegl et al. (2023) [5]	✗	✗	*Only for rollups*	✗	✗
Liu et al. (2025) [6]	✓	✓	✗	✗	*Only for L1s*
Our work	✓	✓	✓	✓	✓

Ethereum and the side chain, but its security relied on a limited set of trusted validators. In March 2022, attackers exploited this trust model by compromising a majority of the validators and forged transactions that drained over $600 million from the bridge. A more rigorous security analysis of the bridge's underlying assumptions, particularly its validator threshold and centralization risks, might have exposed this vulnerability before it was exploited. This incident illustrates the urgent need to systematically examine the security assumptions that underpin Layer 2 scalability solutions.

Our contributions. Over the last few years, a number of surveys on Layer 2 techniques for scaling blockchains have been made [19], providing comprehensive analysis of various scaling solutions including payment channels, rollups, and sidechains [20]. Additional research has examined Layer 2 protocols specifically within the context of decentralized applications and their implementation challenges [21], and even some of them are rollup-specific [22], offering detailed security frameworks for blockchain Layer 2 protocols. While they provide useful insights, they do not provide an exhaustive security analysis for those solutions. This paper contributes to the existing body of knowledge by providing an in-depth survey of the security assumptions underlying permissionless blockchains, and their Layer 2 scalability solutions. To that end, the paper first contextualizes the work by presenting the blockchain scalability problem and surveying current solutions. Then, the paper delves into an examination of the security assumptions fundamental to blockchain systems, and proceeds to scrutinize how these are extended to cover Layer 2 solutions. Additionally, the paper proposes a categorization of both the existing Layer 2 solutions and their security assumptions. To the best of our knowledge, this is the first work to explicitly analyze the security assumptions upon which blockchain Layer 2 solutions rely. Table 1 summarizes the previous work done on the topic of security analysis on Layer 1 and Layer 2 against our manuscript, highlighting that we provide with an analysis of security on Layer 1 and Layer 2, as well as, discussing the security assumptions on both layers, while providing with empirical results and/or case studies in some cases. In particular, and to the best of our knowledge, there is no other study that explicitly lists and specifies the security assumptions that a permissionless blockchain should have, as well as, how Layer 2 solutions are inheriting those assumptions, and which are the additional security assumptions that each Layer 2 make on top of the inherited ones.

Paper organization. The rest of the paper is organized as follows: Section 2 introduces the blockchain scalability challenge, from the problems to the approach proposed to solve this problem, going through the goals it aims to achieve. Section 3 presents the solutions that are currently used to scale blockchains. Section 4 performs an analysis of the security assumptions that Layer 1 and Layer 2 are currently relying on. Section 5 presents the discussion regarding practical implications of our study, as well as, particular instances of attacks occurred on blockchain that happened due to a fault on one or more security assumptions. Moreover, this section also presents the trade-off we found between security, scalability and decentralization on the analyzed Layer 2 solutions. Finally, Section 6 presents the conclusions and future work for this article.

2. Blockchain scalability: problems, goals and approach

In this Section, we discuss the main scalability issues blockchains face, along with their associated goals and the approaches proposed to overcome them. We focus on three practical constraints—throughput, hardware requirements, and networking—and then address the broader theoretical perspective of the Blockchain Trilemma. Finally, at the end of this Section, we briefly comment how Layer 2 trade-off the corners of the Trilemma.

2.1. Problems: throughput, storage and networking

From a practical point of view, permissionless blockchains currently encounter major challenges in three areas:

2.1.1. Throughput

The most common metric for measuring blockchain scalability is transaction throughput, typically expressed in transactions per second (TPS). However, this metric alone does not always capture scalability accurately. Appendix A presents a discussion of alternative metrics.

Figures 1 and 2 illustrate the range of TPS for Bitcoin and Ethereum from January to September 2023. Bitcoin processes approximately 3 to 8 TPS, with occasional spikes above 8 TPS, while Ethereum ranges between 10 and 14 TPS, peaking once above 18 TPS. These values fall significantly short of traditional

non-blockchain payment systems, like VISA [23], which consistently average around 1700 TPS.

The throughput of a blockchain is closely tied to block size. While increasing block size can theoretically raise throughput, it also raises block propagation delay and increases node resource demands, potentially centralizing the network. Although studies exist indicating that moderate block size increases do not profoundly impact security (e.g. increasing Bitcoin's limit to four megabytes [24]), this remains a sensitive parameter.

2.1.2. Hardware requirements

Two critical hardware considerations are computation and storage resources. Keeping node costs low encourages broader participation and supports decentralization. In permissionless blockchains, nodes must download and verify the network's entire history. For instance, Ethereum's full history surpasses one terabyte of raw data [25], making it challenging for less powerful devices (e.g., IoT) to join.

2.1.3. Networking

Finally, data in most traditional blockchains is broadcast and replicated to all nodes. This method can limit scalability as the network expands, particularly when bandwidth is constrained. Table 2 compares the hardware and bandwidth requirements for Bitcoin [26], Ethereum [27], and Solana [28].

2.2. Goals: Fulfill the Blockchain Trilemma

Beyond these practical issues, the core challenge of scaling blockchains lies in preserving security and decentralization. This challenge is captured by the *Blockchain Trilemma* [30], introduced by Vitalik Buterin, which argues that scalability, decentralization, and security cannot all be maximized simultaneously (Figure 3).

Security ensures that only valid, immutable transactions persist.

Decentralization distributes control across many nodes.

Scalability supports high throughput and future growth without extensively altering node infrastructure.

Networks that emphasize one or two properties inevitably diminish the others to some degree (e.g. Nano [31], IOTA [32], XRP [33], eosio [34], Bitcoin [26] or Ethereum [27]). Although some researchers propose adding a fourth property (the so-called "quadrilemma") [35, 36], no consensus exists on its definition. Thus, our focus remains on scalability, security, and decentralization [37, 38].

2.3. Approach: From monolithic to modular blockchains

The ultimate goal is to address the Blockchain Trilemma without sacrificing any of its core components. However, the constraints closely parallel the CAP theorem [39], where consistency, availability, and partition-tolerance cannot be fully achieved at once. As a result, blockchain consensus protocols typically balance these factors—for example, Proof-of-Work emphasizes availability, while Proof-of-Stake focuses on consistency.

Initial efforts attempted to build a *monolithic* blockchain, incorporating all tasks in one integrated system. This approach frequently struggles as the network expands. A more recent strategy divides the functionality into modular *layers* or *components*, each responsible for a core blockchain function:

Consensus for block agreement and transaction ordering.

Execution for processing and updating state.

Data availability to ensure that the data remain accessible.

Settlement for finalizing transactions.

Interoperability for trustless cross-chain communication.

Each layer aligns more strongly with one corner of the Trilemma; for instance, data availability addresses scalability, while consensus and execution protect security, and interoperability promotes decentralization. The resulting *modular* approach seeks to flexibly swap, improve, or combine components to maintain a balanced emphasis on scalability, security, and decentralization even as the system evolves.

2.4. Trade-offs in Layer 2 Solutions

Layer 2 solutions offer a range of approaches to enhance scalability, but each must balance security and decentralization alongside throughput. For instance, Validium relies on off-chain data availability, which can offer higher transaction throughput. However, this design introduces additional trust assumptions: if the off-chain entity providing data availability fails or acts maliciously, the security of the system can be compromised. Consequently, Validium trades away on-chain security guarantees for increased scalability.

Similar trade-offs exist across various Layer 2 designs:

Optimistic Rollups rely on fraud proofs and a challenge period. This maintains a high level of compatibility with the underlying network, but prolongs finality.

ZK-Rollups use Zero-Knowledge Proofs for state validity. They enhance security and reduce trust assumptions but can be resource-intensive to implement, adding complexity to the overall system, and affecting scalability and decentralization.

Validium offers fast and low-cost transactions by moving data off-chain, but reduces security guarantees.

Hybrid Approaches. In pursuit of better alignment with the Blockchain Trilemma, recent proposals explore combining multiple data availability mechanisms or consensus models to maximize security without unduly sacrificing decentralization. Approaches sometimes referred to as "hybrid rollups" split transaction and data publishing between off-chain and on-chain resources. Alternative data availability models, such as *data shards* or *Data Availability Committees (DACs) [40]*, introduce

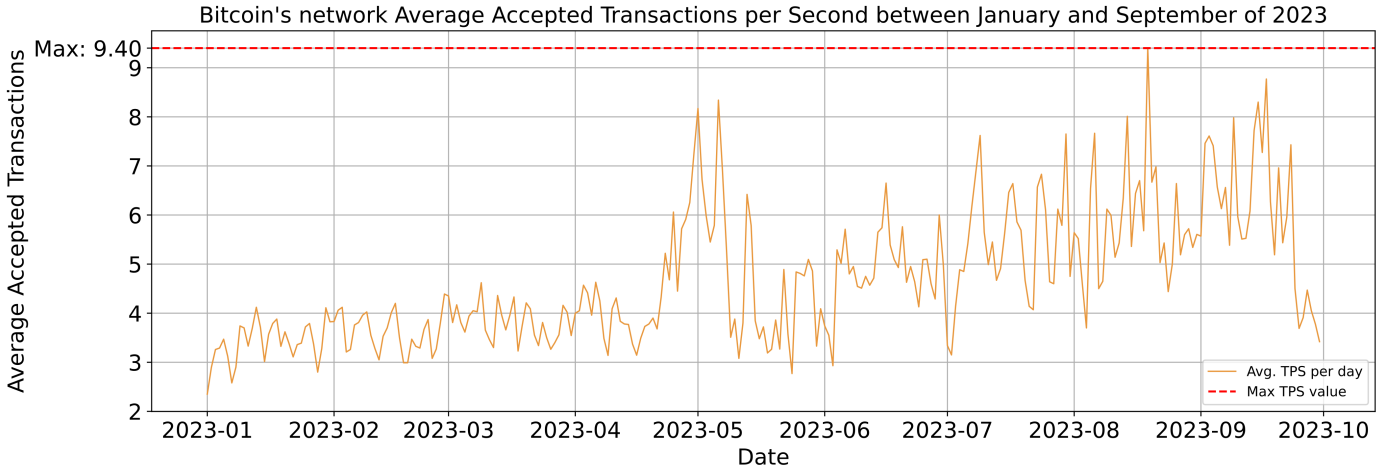

Figure 1: Historic data of Transactions per Second (TPS) of the Bitcoin network. Data obtained from statoshi.info.

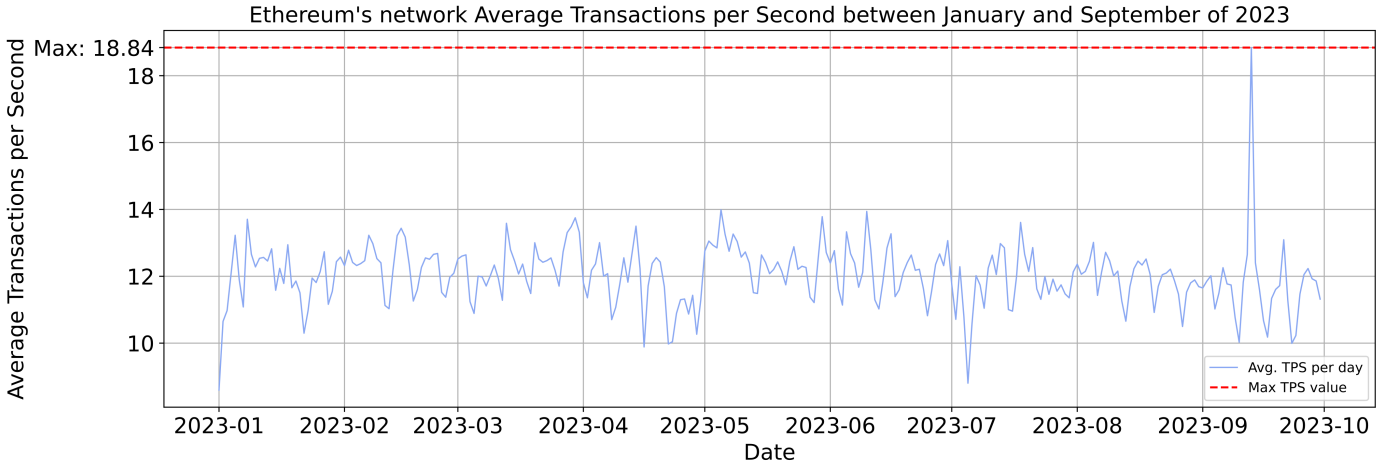

Figure 2: Historic data of Transactions per Second (TPS) of the Ethereum network. Data obtained from migalabs.io.

Table 2: Comparison of Bitcoin, Ethereum and Solana networks in terms of hard drive, CPU cores, RAM memory and bandwidth requirements.

Network	Hard Drive Space	Number of CPU Cores	Amount of RAM	Internet Bandwidth	Number of Nodes[4]
Bitcoin[1]	>450 GB HDD	1	1 GB	5 Mbps	≈ 10.000
Ethereum[2]	>900 GB SSD	2-4	4-8 GB	25 Mbps	≈ 6.000
Solana[3]	>2 TB SSD	>12	128 GB	300 Mbps	≈ 1.200

[1] Data from Bitcoin documentation.
[2] Data from Ethereum documentation.
[3] Data from Solana documentation.
[4] Number of nodes obtained from [29].

redundancy and partial trust assumptions that can improve security over fully off-chain methods.

The core challenge for Layer 2 designs is to find a configuration that upholds security and decentralization while delivering the scaling benefits required for real-world use. Hybrid approaches hold promise in mitigating the typical trade-offs by selectively blending off-chain scalability techniques with strong on-chain assurances. By distributing trust and validation across multiple parties or networks, it may be possible to maintain robust security properties, achieve higher throughput, and preserve a significant degree of decentralization. Future research will clarify which hybrid models offer the best balance across the Trilemma's competing dimensions.

Once we have introduced the security assumptions for Layer 1 and Layer 2, we perform an in-depth comparison of the trade-offs that Layer 2 present in Section 5.3.

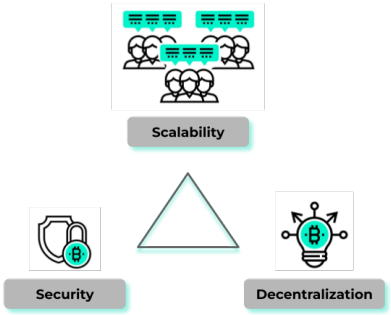

Figure 3: Blockchain Trilemma. Icons from Vecteezy.com.

3. Blockchain scalability solutions: Layer 0, Layer 1 and Layer 2

This Section first outlines the boundaries of this study and then summarizes the existing approaches to scale blockchains, categorized by the layer of their deployment.

3.1. Scope of this study: Bitcoin or Ethereum Virtual Machine-like blockchains

In this work, we focus on the security properties of Layer 2 solutions. However, to contextualize these approaches, we briefly review how each Layer 0, 1, and 2 addresses scalability.

While the previous Section discussed scalability in general, here we focus on solutions and techniques primarily designed to scale Bitcoin and Ethereum.

Moreover, through this study, we consider only *public* (or *permissionless*) blockchains. These are open networks in which anyone can interact and participate on the consensus validation.

Furthermore, we consider Layer 0 (L0) to be the underlying infrastructure on which the blockchain is built. This layer includes both virtual and physical infrastructures such as servers, network protocols, and communication channels. Layer 1 (L1) refers to the blockchain itself, the actual protocol that implements the rules and allowed operations on the blockchain protocol. Finally, we consider as Layer 2 (L2) the additional protocols built on top of L1 protocols, that provide additional functionalities and/or scalability capabilities.

3.2. Layer 0 (L0)

Before even tackling improvements over the blockchain itself, there exist solutions that focus their efforts on the propagation protocol (i.e., trying to solve the issue presented in Section 2.1.3), where enhancements made directly over the way the information is transmitted by the nodes of the blockchain could improve its overall throughput. For more information about Layer 0 improvements, we refer to [41].

Let us present concrete examples of projects that implement enhancements over Layer 0.

bloXroute [42] is known as the first Blockchain Distribution Network (BDN) [43]. This technology enables faster block propagation and, therefore, transactions. In summary, it allows to increase the block size, reduce the block interval and minimize the risk of forks.

Velocity [44] is an improved propagation protocol based on erasure codes [45]. Roughly speaking, it aims to increase the throughput by mining larger blocks.

Kadcast [46] is a protocol specifically designed for efficient block propagation, offering speed, security, and efficiency. It uses the widely recognized Kademlia architecture [47], which is a structured overlay topology known for its effectiveness in facilitating efficient broadcast operations with adjustable redundancy and overhead.

Erlay [48] is a transaction dissemination protocol that significantly reduces the bandwidth consumption by a node. It not only enhances the network's security by enabling more connections to be established at a lower cost but also strengthens privacy by fortifying the network against attacks. Essentially, Erlay achieves increased network connectivity while minimizing the impact on bandwidth and latency, making it a cost-effective solution.

3.3. Layer 1 (L1)

Layer 1 (L1) solutions modify the blockchain's core consensus or data structures (i.e., on-chain improvements). For a broader survey of these solutions, see [19, 36].

The following are the most prominent examples in this category.

3.3.1. Block size

In this Section, we consider solutions that modify (directly or indirectly) the block size. As the ultimate goal, these solutions aim to increase the *effective* block size of the blockchain.

The first –and obvious– approach we can consider on this category is to increase directly the block size since, increasing the block size allows to fit more transactions into blocks and, therefore, increases throughput. However, as introduced above in Section 2.1.1, this measure also increases the block propagation delay and forces the nodes to use more computational resources to validate and finalize transactions, increasing the barrier of entry for new nodes joining the network. Bitcoin Cash (BCH) [49] arose from a hard-fork from the Bitcoin (BTC) network in 2017, that decided to increase the Bitcoin block size from one megabyte to four megabytes. Currently BCH is using 32 megabytes as block size [50].

Another approach over the Bitcoin network related to the block size was the soft-fork activated (through the BIP-141[5], SegWit [51]) in August 2017. This BIP consisted on the reorganization of the digital signatures of transactions on the block, separated from the rest of transaction data, and a modification on the way of measuring transaction size (now called transaction weight). This restructuring resulted on a reduced *transaction size* which, in turn, allowed a larger effective block size.

[5] BIP: Bitcoin Improvement Proposal.

3.3.2. Sharding

This approach presents the ability to split the database into many *shards* in a way that the nodes validating the blockchain do not need to keep track of the whole blockchain. Instead, nodes are grouped in *communities* and each community keeps track of a shard of the whole database. In this way, the storage requirements (recall Section 2.1.2) are decreased, thus lowering the barrier to entry into the network.

This approach was once considered as part of the Ethereum roadmap; however, the rapid development of Layer 2 rollups [52] (more in Section 3.4.2) and the appearance of Danksharding [53] (and Proto-Danksharding [54] as an intermediate step) to improve further the rollups has lead to a rollup-centric scaling approach instead of a sharding-centric one [55].

3.3.3. Probabilistic consensus

In this category, improvements made over the consensus mechanism are explored.

Proof-of-Work (PoW). It was the first blockchain protocol that brought consensus to the Bitcoin network. Its strengths are security and decentralization; however, it offers low scalability and consumes large amounts of energy. Several proposals were made in order to improve the scalability capabilities that PoW has.

Improving Proof-of-Work. Let us briefly present three of them.

Bitcoin-NG, proposed in 2016 by Eyal et al. [56], aims to enhance blockchain scalability using Proof-of-Work (PoW). It introduces a segregation of the Bitcoin consensus process into leader election and transaction serialization. Time is divided into epochs, and within each epoch, a leader is elected through PoW to create blocks continuously without the need of performing PoW. This approach results in two types of blocks: key blocks; generated by miners for leader selection, and microblocks; created by the elected leader within their epoch and containing actual transactions. Bitcoin-NG achieved a significant increase in transactions per second (around 10 times), but confirmation times are longer, requiring waiting for approximately 100 key blocks to prevent double-spending.

Bicomp, introduced by Jiao et al. [57] in 2018, it builds upon the Bitcoin-NG approach and aims to reduce the power of the elected leader. It follows a similar structure with the creation of two types of blocks: macroblocks and microblocks. The process occurs in rounds, each with an elected leader. Contesting miners use PoW for leader election through macroblocks, while transactions are packed into microblocks by miners also utilizing PoW. In each round, the elected leader receives multiple simultaneously mined microblocks and serializes them into a single macroblock, which is then broadcasted to the entire network.

Greedy Heaviest Observed Sub-Tree (GHOST) was proposed by Sompolinsky and Zohar [58] as a solution to enhance fairness, mining power utilization, and prevent double-spending in PoW blockchains. The GHOST rule was developed to address the challenges faced at high transaction throughputs in Bitcoin, where even unsophisticated attackers could exploit the system. By changing the main chain selection from the longest chain to the chain with the heaviest sub-tree, the GHOST rule considers PoW blocks that do not make it into the main chain. This approach improves security and enables a high throughput Bitcoin blockchain, capable of handling up to 200 TPS. However, the process of finding the main chain in GHOST poses challenges and potential vulnerabilities to denial of service attacks. Ethereum has incorporated a simplified version of GHOST in some of its versions.

Proof-of-Stake. Proof-of-Stake (PoS) [59] consensus was introduced as an alternative to PoW mainly due to its high energy usage. PoS selects the block creators (i.e., the validators) depending on the coins owned by the validator. Unfortunately, PoS did not solve the scalability problem and, in terms of security, it is vulnerable to new attacks, such as the Long Range Attack [60] or the Nothing at Stake Attack [61]. Nonetheless, relevant networks such as Ethereum are using this protocol as the basis for its consensus mechanism, while constantly improving it through EIPs[6]. Although PoS is a weaker consensus mechanism in terms of security, it effectively reduces the energetic cost of running the network.

Improving Proof-of-Stake. Let us present two improvements made over PoS.

Delegated Proof-of-Stake (DPoS) [62] was introduced with the aim of improving the scalability of PoS. In this variant of the protocol, the participants, based on their stakes, periodically choose delegates from their ranks. These delegates are responsible for creating blocks on behalf of the other participants.

Ouroboros [63] is an improved version of a PoS consensus that is currently used on the Cardano network [64]. The protocol ensures security and enhances the scalability of PoS by appointing delegates within an epoch through a coin-flipping algorithm. Nodes generate a verifiable random number to demonstrate their suitability for delegate status. Each epoch is composed of multiple slots. Delegates employ a multi-party computation method based on their stakes to randomly select block creators for the slots in an epoch. These same delegates also participate in electing delegates for the subsequent epoch. Ouroboros Praos, introduced by David et al. [65] in 2018, is another PoS protocol akin to Ouroboros. It delivers security in a semi-synchronous blockchain environment, safeguarding against compromise of blockchain stakeholders.

[6] EIP: Ethereum Improvement Proposal.

Proof-of-X. Besides PoW and PoS, there are other Proof-of-X (probabilistic) consensuses presented as alternatives to PoW (or Nakamoto-like consensuses) and PoS (and its improved versions). Let us briefly present one of them.

Proof-of-Space [66] is a consensus mechanism in which validators commit disk storage space to the blockchain network. Validators, often called "farmers", are selected to create new blocks based on the amount of space they have allocated. The more space provided, the greater the probability of being chosen, promoting fairness and energy efficiency.

3.3.4. Hybrid consensus

Hybrid consensus protocols combine two or more existing approaches to balance security and scalability. Examples include system designs that merge PoW with BFT, or PoS with GHOST, yielding improved throughput at the cost of added complexity.

Ethereum Gasper, originally on the Ethereum roadmap, there was a proposal called Ethereum Casper [67] that combined PoS and Byzantine fault tolerance (BFT) [68]. Casper was intended to be deployed on the Ethereum network on 2020; however, further research shifted this proposal to implement the Ethereum Gasper, which is the merge between Ethereum Casper and GHOST (explained in Section 3.3.3).

ByzCoin Protocol [69] combines the Bitcoin-NG (explained in Section 3.3.3) with PBFT [68] to achieve fast finality on the Bitcoin network. The consensus protocol was built on top of a Collective Signing (CoSi) method [70] to further improve PBFT by using a tree-structured communication.

Byzantine Agreement (BA) Protocol [71] is the consensus protocol used on Algorand. It achieves finality in less than a minute. Moreover, Algorand uses another additional random verifiable function to scale the BA algorithm.

3.4. Layer 2 (L2)

The scaling solutions described in this section enable the extraction of computational processes from the primary network (Layer 1) and execute them *off-chain*. This means that, instead of performing the computing intensive part of the activity over the blockchain directly, the bulk and computational-consuming part of the job can be outsourced on a Layer 2 and, finally, post an update on Layer 1 in order to reflect the changes happened off-chain.

A crucial characteristic of these solutions is the fact that they are implemented completely separated from Layer 1 in such a way that, in theory, they do not require changes over the existing Layer 1 protocol for them to work.

Let us present the different Layer 2 technologies considered in this article.

3.4.1. Payment Channel Networks (PCNs)

Payment Channel Networks (PCNs) [72] facilitate the creation of a peer-to-peer network layered on the primary blockchain network, granting participants the freedom to carry out unlimited transactions without being constrained by the inherent limitations of the underlying blockchain. However, Payment Channels within this framework must address various concerns related to security and reliability. The most prominent examples of this approach are the Lightning Network [72] for the Bitcoin network, and the Raiden Network [73] for the Ethereum network.

For an in-depth explanation about how Payment Channel Networks work, please refer to Appendix C.

3.4.2. Rollups

Rollups are a Layer 2 technique that allow to bundle transactions together and publish them in batch on the blockchain, along with a proof for its correctness.

Depending on the manner this proof is generated and validated, there exists two different flavors for Rollups: Optimistic Rollups –backed by *fraud proofs*–, and Zero-Knowledge Rollups –backed by *validity proofs*–.

Let us explain what the Optimistic Rollups and Zero-Knowledge Rollups are.

Optimistic Rollups. Optimistic rollups [74], as stated before, are a kind of rollup that is based on *fraud proofs*. In short, a fraud proof presents an evidence that a state transition was *incorrect*.

Hence, Optimistic Rollups are considered to be "optimistic" because they assume that the off-chain state transition computed is valid. Optimistic Rollups rely on the fraud proof to detect the cases where an invalid transaction is computed. The way it works is the following: after a rollup batch is computed and posted onto Ethereum, the system sets a time window (called the *challenge period*) during which everyone can challenge the results of the posted computation and present a fraud proof, invalidating the posted result. On the one hand, in case the fraud proof succeeds, the rollup re-executes the transactions posted on that batch and updates the rollup state (and underlying blockchain) accordingly. Additionally, when a fraud proof succeeds, the Sequencer responsible for including the invalid state transition is punished and receives a penalty. On the other hand, if no successful fraud proof was submitted during the challenge period time-window, then the published state transition is considered valid and it is accepted on Ethereum.

For an in-depth explanation about how Optimistic Rollups work, please refer to Appendix D.

Zero-Knowledge Rollups. Zero-Knowledge Rollups [75] (or ZK-Rollups) are backed by *validity proofs*. Validity proofs represent off-chain computations sent to Layer 1. They typically use a Succinct Non-interactive ARgument of Knowledge (SNARK) or a Scalable Transparent ARgument of Knowledge (STARK) as the Zero-Knowledge Proof. With this proof, the rollup is able to compress a huge amount of computation into a small –succinct– proof that is fast and easy (at least faster and

easier than re-executing all the transactions bundled together) to verify on the Layer 1.

Before deep diving into SNARKs and STARKs, let us present a couple of examples that will help us better understand how Zero-Knowledge Proofs work.

Two balls and the color-blind friend. Consider two spheres that are identical in every feature except for their color: one is red, and the other is green. Suppose a person, Victor (the Verifier), that is unable to tell these colors apart due to a red–green color blindness. He doubts whether these spheres are truly different in color. There is another person, Peggy (the Prover), that wants to prove Victor that the spheres are of different colored without disclosing which sphere is red and which sphere is green.

The protocol proceeds as follows:

1. Both spheres are given to Victor, who conceals them behind his back.
2. Victor selects one sphere at random, shows it to Peggy (who can distinguish red and green), and then returns it behind his back.
3. Victor may or may not switch the spheres behind his back.
4. Victor picks one of the two spheres at random and presents it, asking, "Did I switch the sphere?"
5. Peggy, by observing the color, answers whether a switch occurred.

If the spheres were truly the same color, any correct response to the question would be a mere guess with probability 0.5. Repeated ensembles of rounds would provide a statistical measure. Consistent correct answers become extremely unlikely under random guessing. Thus, when the prover consistently answers correctly across multiple trials, Victor must conclude that the spheres differ in color.

Throughout this process, Victor never learns which sphere is red and which is green. The only information he gains is that the spheres are distinguishable by its color. In other words, no additional knowledge is revealed other than the fact that the two spheres are not the same color.

Where's Wally?. One well-known illustration of a Zero-Knowledge Proof is the "*Where's Wally?*" example. In this scenario, the Prover wants to demonstrate to the Verifier that they know Wally's position on a particular page of a *Where's Wally?* book, but without revealing the exact location.

This example has two different parts: a Setup and a Protocol. Regarding the setup,

- The Prover prepares a large black board, twice the size of the book in both dimensions.
- The board has a small cut-out hole, exactly the size of Wally.

The protocol goes as follows:

1. The Prover places the black board over the page so that Wally is visible through the hole.
2. Because the board is larger than the book, the Verifier cannot determine the precise position of the hole on the page.
3. The Verifier can see Wally through the hole but cannot see any other part of the page.
4. Hence, the Verifier is convinced that the Prover knows where Wally is but learns nothing else about his location.

Please, note that in both examples explained here, the Prover is able to proof to the Verifier of this statement because they *knows* the knowledge they want to prove.

For an in-depth explanation about how Zero-Knowledge Rollups work, please refer to Appendix D.

SNARKs and STARKs. Both SNARK and STARK, in the context of blockchains, allow you to prove that the post-state of a group of transactions is correct without having to re-execute them.

SNARK stands for *Succinct Non-interactive ARguments of Knowledge*:

Succinct: The proof is significantly smaller than the transaction data it covers. Verifying these small proofs usually takes only a few milliseconds, and the proof itself often spans only a few hundred bytes.

Non-interactive: There is no ongoing communication between the Prover and Verifier. By applying the Fiat–Shamir heuristic [76], an interactive system can be transformed into a non-interactive one, which speeds up verification.

ARguments: This mechanism protects the Verifier from being tricked by a Prover with limited computational power. In theory, a Prover with infinite power could generate false proofs, but these arguments ensure soundness.

Knowledge: A Prover cannot generate a valid proof if they do not actually possess the specific underlying data (the "witness"). Without this knowledge, creating a legitimate proof is impossible.

STARK stands for *Scalable Transparent ARgument of Knowledge*. In essence, STARKs rely on weaker cryptographic assumptions (for example, they do not require a Trusted Setup) but face drawbacks like larger verification times and bigger proof sizes.

Scalable: Moving the bulk of the computation off-chain greatly reduces on-chain execution costs. Proof size and verification time usually grow only logarithmically with respect to the input size.

Transparent: STARKs do not need a *(toxic) Trusted Setup*, where a special reference string is built and shared. Instead, STARKs rely on publicly verifiable randomness to create any necessary setup data, avoiding the risk of "toxic waste." See [77].

ARgument: Identical to what is explained in Section 3.4.2.

Knowledge: Identical to what is explained in Section 3.4.2.

Security differences between SNARKs and STARKs. While SNARK and STARK are used to create Zero-Knowledge Proofs, they differ in their cryptographic assumptions. SNARKs typically rely on elliptic curve pairings, which require a Trusted Setup. If this setup is compromised, it can undermine the entire system. In contrast, STARKs remove the need for a Trusted Setup by relying on publicly verifiable randomness and collision-resistant hash functions. This approach is generally considered more robust against quantum attacks. However, STARK proofs take more space and time to verify, so they trade off performance benefits for stronger security guarantees.

Additionally, it is important to state that both rollups post the plain transaction data computed off-chain into the Layer 1 as a way to achieve data availability[7].

Validium. Validium [78] is a technique very similar to a ZK-Rollup. Its main difference relies on the fact that, while a ZK-Rollup uses off-chain computation and on-chain data availability, validiums use an off-chain approach for both characteristics: computation and data availability. The main benefit of using Validiums over a ZK-Rollup is thus in terms of cost: the major part of the costs when using a ZK-Rollup comes from providing on-chain data availability [79].

Nonetheless, in order to withdraw funds from Layer 2 back to Layer 1, the user needs to prove to the bridge smart contract that they own the requested amount on Layer 2. This is done using a Merkle proof of inclusion [80]. The only guaranteed way to provide with such proof is by having access to transaction history and building the proof by yourself.

In this sense, ZK-Rollups always guarantee the ability to build such proofs (since they always post the plain text transaction onto Layer 1; i.e., they provide with on-chain data availability), whereas Validium does not do that. They typically rely on an off-chain data availability platform where these transactions are stored.

Volition. Volition [81] offers a dynamic way to adjust the data availability when publishing transactions onto Ethereum. They offer flexibility when sending transactions: allowing to send them *à la* ZK-Rollups (i.e., posting the plain transactions along with the proof for its correctness) or send them *à la* Validium (i.e., keeping the data availability off-chain).

[7] Note that, with "data availability", we refer to the fact that data produced off-chain (i.e., on Layer 2) is available on Layer 1 in a way that no network partition has the ability to withhold data.

4. Security analysis

4.1. Blockchains: Layer 1

The aim of this section is to describe the underlying security assumptions that Layer 1 blockchain schemes inherently possess.

4.1.1. Reliable consensus

Consensus is how blockchain participants agree on the network's state. In a distributed system such as a blockchain, all nodes must reach agreement on transactions and blocks.[8]

The problem of reaching consensus in a distributed system has been studied since 1977, when Lamport [82] presented a framework for proving the correctness of a multi-process program. Specifically, the author demonstrated that two distinct criteria have to be met for ensuring correctness: safety and liveness[9].

In 1980, Pease et al. [83], formally defined an algorithm to reach consensus in the context of faulty (or malign) parties in the system.

Later, in 1985, Fisher et al. [84] presented the Fisher, Lynch and Paterson (FLP) impossibility. This theorem states that achieving agreement[10], validity[11] and termination[12] simultaneously is impossible in an asynchronous system. For this reason, various –relaxed– versions of consensus have emerged in an aim to circumvent the FLP impossibility.

While Fisher et al. do not make explicit references to safety and liveness, they underscore the challenges of ensuring both agreement (safety) and termination (liveness) in distributed systems.

Finally, in 1999, Brewer presented the CAP conjecture (which later, in 2002, became a theorem thanks to the proof by Gilbert and Lynch [85]) that, briefly, states that any distributed data storage can achieve, at most, only two of the following three properties:

Consistency Ensures that every read receives the most recent write or results in an error.

Availability Guarantees that every request receives a non-error response, without the assurance that it contains the most recent write.

Partition Tolerance Allows the system to continue operating even when an arbitrary number of messages are dropped or delayed by the network between nodes.

[8] Technically speaking, there are two ways to participate in this process: by exchanging messages over the network, or by using a shared memory. Nonetheless, due to the nature of blockchain systems, the lack of trust, and the physical distance of the nodes conforming the network, the second approach is unfeasible.

[9] Defined in Section 4.1.4.

[10] Agreement implies that two fault-free nodes can not output two different values.

[11] Validity implies that if all the fault-free nodes propose the same value y, then every fault-free node has to output y as well.

[12] Termination implies that every fault-free node will eventually output a value.

This theorem, however, announced that there are trade-offs between consistency and availability in the presence of network partitions. Thus, in this setting, the CAP theorem is simply an instance of the fundamental fact that you cannot achieve both safety and liveness in an unreliable distributed system [86].

All of this previous research resulted on the creation and appearance of fault-tolerant, *relaxed* versions of consensus protocols that are applicable to blockchain systems, while ensuring the two main properties Lamport stated. For instance, Bitcoin only offers *weak* consistency, i.e., different nodes on the system might end up having different views of the blockchain as a result of forks.

Considering this, the consensus protocols that blockchains are currently using can be categorized in one of these three main categories:

Nakamoto consensus. This category comprises from the original Proof-of-Work of Bitcoin network to any other tweaking made to this kind of consensus. For instance, Ethereum's Proof-of-Work (before The Merge [87]), Zcash, or Monero's Proof-of-Work would fit here.

Proof-of-X. Since one of the biggest criticisms of Bitcoin is its intensive energy usage, Proof-of-X consensus mechanisms aim to replace the power-hungry computation with other more efficient computation (or remove the computation altogether). Protocols like Proof-of-Stake, Proof-of-Space, among others would fit here.

Hybrid proofs (also known as *ebb-and-flow* consensus). In order to circumvent the poor performance as well as safety limitations (e.g., weak consistency and low fault-tolerance), the appearance of consensus protocols where a *committee* (rather than a single node) that collectively drives the consensus have emerged. Hybrid proofs that contemplate multiple committees also would fit here. Examples in this category can be the Algorand or ByzCoin consensus protocols.

Thus, Layer 1 blockchains assume they run a reliable consensus protocol that either favours consistency or availability under partition tolerance.

4.1.2. Secure cryptographic primitives

Hash functions are the fundamental cryptographic building block used in blockchains. The National Institute of Standards and Technology (NIST) provides the standard criteria for evaluating hash function security [88].

For more information about this criterion, we refer to Appendix B.

Let us exemplify which is the role of hash functions for the particular case of Bitcoin and Ethereum.

Bitcoin On the one hand, Bitcoin primarily makes intensive use of the Secure Hash Algorithm 256 (SHA-256) [89] for mining, transaction verification, and the creation of the digital signatures.

Regarding mining, SHA-256 hash function is used to compute the hash of a *block candidate* which contains the previous block hash, the Merkle Root [80] (which is the hash root of all the transactions in the Merkle Tree, the hash function to compute this root is, again, SHA-256), the timestamp, the difficulty target (a parameter that is adjusted every 2016 blocks in a way that every block takes roughly 10 minutes to be mined), the nonce (a 32-bit field that miners can change to try to get a hash result lower than the target), among others.

Bitcoin uses the SHA-256 hash function in combination with the Elliptic Curve Digital Signature Algorithm (ECDSA) (more on that below in this Section) to create and verify digital signatures for transactions. Finally, RIPE Message Digest-160 (RIPEMD-160) [90] is used within the address generation.

Although RIPEMD-160 is used on top of the SHA-256 digest, this hash function does not provide more security on top of the SHA-256, since its solely purpose is to provide a shorter Bitcoin address.

Ethereum On the other hand, Ethereum [91] makes use of a variation of Secure Hash Algorithm-3 (SHA-3) [92] known as Keccak-256 [93, 94]. In particular, this hash function is used in a range of different aspects of Ethereum.

Starting with transaction message digests, when a new transaction is created in Ethereum, the transaction's content, including sender address, recipient address, value, and other parameters, is hashed using the Keccak-256 hash function. The resulting hash, known as the transaction message digest, uniquely identifies the transaction and is used for transaction verification and inclusion in blocks.

Another use of this hash function can be found in the block header and block hashes. Each block in the Ethereum blockchain contains a block header, consisting of various fields like previous block hash, transaction Merkle root, and timestamp. The block header is hashed using Keccak-256 to produce the block hash, which uniquely identifies the block. The block hash is crucial for block verification and linking blocks together.

Additionally, when Ethereum was using Proof-of-Work as its consensus mechanism, it used Ethash [95] which, in turn, made use of Keccak-256 as the hash function to mine blocks.

The security of hash functions is measured in security bits. Table 3 summarizes the security strengths presented by the NIST regarding the security bits the different hash functions Bitcoin and Ethereum networks rely on.

Thus, in summary, Layer 1 blockchain systems assume their hash functions are secure, that is, there do not exist generic attacks that are able to find preimages, second-preimages and collisions with costs lower than the security bits described in Table 3.

Table 3: Security bits of hash functions used on Bitcoin and Ethereum network. Extracted from [96].

Hash function	Collision Resistance Strength in bits	Preimage Resistance Strength in bits	Second Preimage Resistance Strength in bits
SHA-256	128	256	$256 - L(M)^*$
SHA-512	256	512	$512 - L(M)^*$
SHA3-256	128	256	256
SHA3-512	256	512	512
RIPEMD-160**	< 80	160	$160 - L(M)^*$

* $L(M)$ is defined as $L(M) = \log_2 \frac{\text{len}(M)}{B}$, where $\text{len}(M)$ is the length of the message M in bits and B is the block length of the function in bits. On the hash functions listed, $B = 512$ for SHA-256, SHA3-256 and $B = 1024$ for SHA-512 and SHA3-512.
** Data deduced from the similarity of RIPEMD-160 to SHA-1.

The other cryptographic primitive blockchains heavily rely on is the digital signature used to sign transactions in a pseudo-anonymous manner.

Both Bitcoin and the Execution Layer of the Ethereum network make use of Elliptic Curve Digital Signature Algorithm (ECDSA) [97], which is considered secure by the NIST[13]. Additionally, since November 2021[14], Bitcoin also supports Schnorr signatures [101]. The Consensus Layer of Ethereum makes use of BLS (Boneh-Lynn-Schacham) signatures [102].

It is important to note that the different signature schemes considered in this article make use of algebraic curves [103]. In particular, ECDSA and Schnorr signatures make use of elliptic curves [104], a particular case of algebraic curves. The actual curve used on Bitcoin (for both ECDSA and Schnorr signatures) and on the Execution Layer of Ethereum is secp256k1. The specification of this elliptic curve can be found in [105] defined by the Standards for Efficient Cryptography Group (SECG) [106]. Ethereum BLS signatures make use of BLS12-381 [107] as the specific algebraic curve, defined by Electric Coin Co. [108].

Finally, with respect to signature schemes that require a *nonce*[15] in order to generate the signature, it is extremely important that this nonce comes from a reliable source of randomness, or at least, derive it deterministically [109, 110, 111, 112, 113].

Thus, in summary, blockchain systems assume that Layer 1 makes use of secure signature schemes, implemented on a secure algebraic curve.

4.1.3. Decentralization

Decentralization is crucial for blockchain systems. It enables both consensus and censorship resistance. A peer-to-peer network where all nodes have equal roles creates this decentralization. That is, there are no nodes that are "more important" or have "more responsibility" than others. Moreover, in the case of blockchain, there are additional properties required to obtain a decentralized network. Namely, physical geographical distribution, mining/staking distribution, among others.

In order to achieve that, decentralization is encouraged through a number of different techniques:

- Lowering the barrier of entry (in terms of hardware cost) to use and participate in the network.
- Providing with different software clients.
- Providing with different hardware architecture support.
- Encouraging the spread and installation of nodes in different geographical locations. Additionally, when setting up a node on an IaaS platform (e.g., Amazon Web Service), avoiding choosing a platform that already has a relevant number of nodes running.
- Encouraging ownership diversity by, for instance, the implementation of a variety of exchanges.
- Encouraging developers to join by facilitating development and maintenance.

In the case of Ethereum, the Ethereum Community has devoted much effort in implementing and maintaining a number of different clients (both on Execution and Consensus Layer) along with the support of running a full node on different hardware architectures (x86 and ARM [114]). Geth [115], Nethermind [116] or Erigon [117] are examples of Ethereum Execution clients, while Lighthouse [118], Lodestar [119] or Nimbus [120] are examples of Ethereum Consensus clients. The Diversify Now website [121] displays a graph showing the distribution of the different clients for both the Consensus and Execution Layer of Ethereum. As of April 2024, it shows that the Consensus Layer of Ethereum is doing better towards decentralization on client implementations than compared to Execution Layer. On the former, we can see that Prysm [122] runs on around 40% of the network, followed by Lighthouse with around 35%. However, Teku [123] and Nimbus have also relevance in the space with around 10% each. Nonetheless, regarding Execution, we observe a centralization of clients towards

[13] The digital signature scheme used by both Bitcoin and Ethereum is approved by NIST, whereas the particular algebraic curve is not approved by NIST.

[14] Due to the activation of BIPs 340, 341 and 342 [98, 99, 100], also known as "Taproot".

[15] A cryptographic nonce is a random (or pseudo-random) number used only once in a cryptographic communication.

Geth, with more than 60% of the nodes in the network. The Ethereum Network Monitoring project [124] shows different dashboards with decentralization metrics for both Consensus and Execution Layer of Ethereum.

Additionally, some Ethereum Execution clients offer the possibility to trade-off trust and sync facilitation by providing *checkpoints* [125] that ease the initial synchronization process, reducing the amount of data the new hardware joining the network has to download and process (and, hence, shrinking the time needed to bootstrap), but at the expense of lowering the trust of the node's setup. In this same direction, Bitcoin also offers this trade-off by offering configuration flags that disable signature verification for historic –trusted– blocks [126].

Regarding Bitcoin, Bitcoin Core client is compatible with x86 and ARM hardware architectures. Moreover, besides Bitcoin Core, there exist other clients such as Libbitcoin [127], Bitcoin Knots [128], or Btcd [129]. Bitnodes [130] show a ranking with the client distribution for the Bitcoin Network. Unlike Ethereum, we can see that (as of February 2024) more than 95% of the nodes are running a version of the Bitcoin Core client. Bitnodes [131] also show a geographic distribution of these nodes.

In an aim to measure the level of centralization a blockchain has, the Minimum Nakamoto Coefficient can be used, which is a generalization of the Gini Coefficient and the Lorenz Curve [132] that are typically used to measure inequality.

The Nakamoto Coefficient counts how many different parties verifying the system must collude in order to gain (at least) $33.\hat{3}\%$ of the total stake of the network. In particular, given a subsystem s with K entities and let $p_1 > \cdots > p_K$ be the proportions of the subsystem controlled by each of the K participants s.t. $\sum_{i=1}^{K} p_i = 1$, the Nakamoto coefficient of a subsystem N_s is the minimum number of entities whose proportions added get, at least $33.\hat{3}\%$ control.[16]

$$N_s := \min\{k \in [1, \ldots, K] : \sum_{i=1}^{k} p_i \geq 0.\hat{3}\}.$$

Moreover, if a decentralized system is composed of S subsystems, where N_s denotes the Nakamoto coefficient of the subsystem s then, the Minimum Nakamoto Coefficient $N_{\min}$ is defined as:

$$N_{\min} := \min\{N_1, \ldots, N_s\}.$$

A table with the Minimum Nakamoto Coefficients for networks like Polygon [133], Ethereum [27], Solana [28] or MINA [134] can be found in [135].

For a more in-depth study on the decentralization of Bitcoin and Ethereum, we refer to [136] and [137].

4.1.4. Specific security assumptions

In this Section, the specific security assumptions are outlined from the explanations provided in the previous Section.

[16] Extracted from [132].

From secure cryptographic primitives. In the context of blockchain, the specific security assumptions that we can extract regarding cryptographic primitives are the following.

Secure hash functions. As discussed above, hash functions are the main cryptographic primitives used in blockchains, both in Layer 1 and Layer 2 alike. In particular, from the properties that a secure hash function must have (explained in Appendix B), the two most important properties in this context are the **infeasibility of collisions**[17] and **second preimage resistance**[18]. Those two properties are vital to guarantee the correct behavior of blockchains. On the one hand, regarding mining, it is crucial to guarantee that it is unfeasible to find a new block that hashes to an existing –confirmed– block, since this would allow to rewrite the blockchain. On the other hand, regarding signatures, if you are able to find second preimages of an address, you could generate valid signatures using a different private key and, in turn, you could make arbitrary changes to the transaction without altering the signature.

Secure digital signatures. Secure digital signatures provide three properties:

- **Authentication.** In the context of digital signatures, refers to the process of verifying the identity of the signer.
- **Non-repudiation (Undeniability).** In the context of digital signatures, refers to the fact that a digital signature, once signed, cannot be revoked or disowned by the signer on a future moment.
- **Immutability.** In the context of digital signatures, refers to the inability to present an altered content with a valid tampered digital signature.

Figure 4 summarizes the aforementioned specific security assumptions. In green, we can see the umbrella security assumption we are working on, while in blue we can see the direct two specific security assumptions extracted from the security assumption. Finally, in orange, we can see the two most important properties extracted from secure hash functions, along with the three particular properties a secure digital signature must provide.

From reliable consensus. In this study, we consider equivalent "reliable consensus" and "consensus is safe and live". Let us introduce safety and liveness for consensus protocols.

Safety. This property implies that any transaction (or state change) considered correct and final by one properly-working node of the system will be eventually considered correct and final by every node following the rules

[17] It is computationally unfeasible to find two different messages with the same hash value within a reasonable amount of time

[18] It is computationally unfeasible to find a second input message that generates the same hash value within a reasonable amount of time.

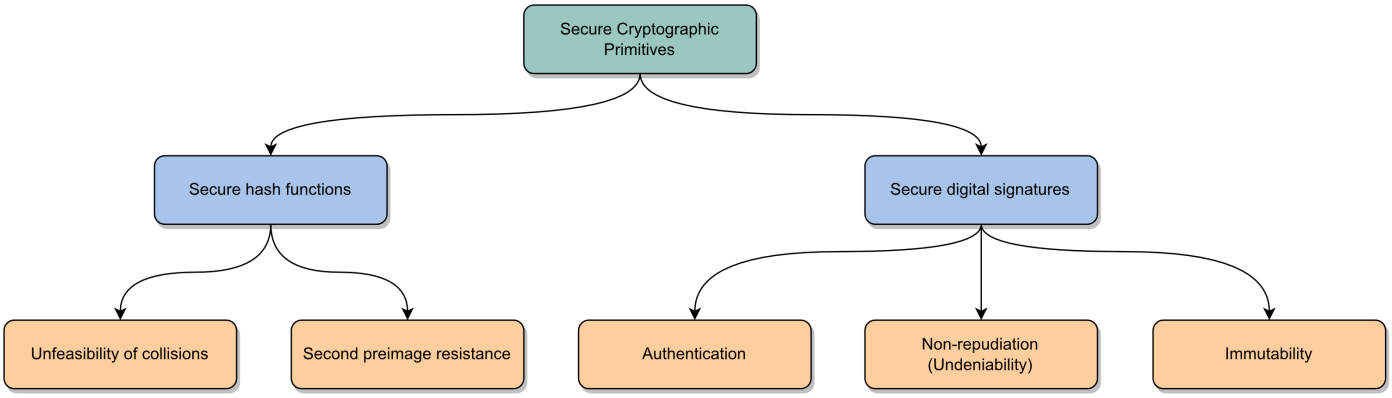

Figure 4: Specific security assumptions from secure cryptographic primitives

of the consensus. This property also ensures that no two transactions (or state change) considered final by two different properly-working nodes will ever conflict.

Liveness. This property states that as long as there is, at least, one transaction that is suitable to be included on the blockchain (that is not considered final by any properly-working operating node) then, the set of finalized transactions by (at least) one properly-operating node will include the aforementioned transaction.

From these two properties, a number of specific security assumptions can be drawn.

Trusted genesis block. Derived from safety, the trusted genesis block is yield as a specific security assumption. An attack regarding the genesis block of a blockchain is discussed by Garay et al. [138] where they present a security model where an adversary pre-computes a genesis block, leading to plausible –but flawed– blockchains.[19]

For this reason, we place the ability of "bootstrap using the same genesis block" as a security assumption of Layer 1 systems, ensuring **secure bootstrapping**.

Additionally, it is necessary to assume that the ability to access the genesis block at any given time is granted, so anyone can bootstrap the blockchain.

Actors are rational. There are incentives to follow the consensus. Derived from both safety and liveness, Layer 1 blockchains make a specific security assumption about the rational[20] behavior of actors and the incentives involved in the consensus process. The design of incentives is aimed at encouraging rational actors to adhere to the consensus rules. This assumption typically encompasses token creation, token management, maximum token supply (if applicable), and token removal from the network. Furthermore, it is assumed that this security assumption provides the framework for defining the rules and the value of the token.

This specific security assumption must cover, at least, the following aspects:

Handle token creation. It should handle how the tokens are created and distributed. This also includes how the tokens should be initially distributed (e.g., through an Initial Coin Offering). Usually, the creation and distribution of new minted tokens is done as a reward for validating the information on the network (e.g., as a mining reward in Proof-of-Work or a stake reward in Proof-of-Stake).

Handle maximum token supply. It should determine the total amount (if any) of tokens that can be in circulation at any given time.

Handle token removal. In case of need, it should also define the burning mechanisms that allow to remove tokens permanently from circulation.

Layer 1 blockchain systems assume sound and well implemented incentive mechanisms that allow the creation, distribution and removal of tokens, as well as that the actors involved on the system are rational.

Eventual Consensus. Derived from liveness, eventual consensus can be extracted. Since every blockchain needs a consensus protocol to agree on the built blocks, they also need a way to share this information. When synchronizing the network, *weak synchronization* might happen. Weak synchronization refers to the event where a relevant fraction of nodes do not have an up-to-date blockchain state even after an extended period of time. Baek et al. [140] and Decker et al. [141] present the problems that carry the fact that the Bitcoin network has delayed block propagation.

[19] In the article, they present an attack where an adversary is allowed with polynomial-time pre-computation so he can compute a very long –private– chain and then reveal blocks at the rate honest players compute new blocks. Doing that, the adversary can break many properties of a sound permissionless blockchain.

[20] Rational as defined in [139].

It is worth noting that block size has a direct impact over the synchronization capabilities of the network. In particular, the propagation time is directly proportional to the block size.

Thus, it is supposed that the system has *eventual* consensus and that the information is synchronized among the nodes fast enough to avoid the problems weak synchronization has.

From reliable consensus and decentralization. By considering both reliable consensus and decentralization, we derive two robust and crucial security assumptions with significant implications.

Provide with (on-chain) data availability. Derived from liveness and decentralization, on-chain data availability can be drawn. On-chain data availability ensures that any user is able to download all the data required to verify any given transaction at any given point on time.

Thus, Layer 1 blockchain systems assume that they have the means to provide with on-chain data availability.

(Eventual) Censorship-resistance. Derived from liveness and decentralization, (eventual) censorship-resistance can be drawn. Eventual censorship-resistance refers to the fact that nobody can prevent anyone to "participate" (understanding "participate" as "use") in the blockchain network. In the case of Bitcoin, this means that valid transactions will eventually be included in the blockchain, although the time they take to be included in a future block may be affected by parameters such as the fee or the standardness of its scripts. This provides **guaranteed finality** of transactions.

Figure 5 summarizes the aforementioned specific security assumptions, obtained from a reliable consensus and decentralization. In green, we can see the umbrella security assumptions we are working on. In purple, we have safety and liveness from a reliable consensus, while in blue we can see the five specific security assumptions extracted from it. Finally, in orange, we can extract the property of secure bootstrapping guaranty (obtained from a trusted genesis block), the guaranteed finality property provided by the (eventual) censorship-resistance, and the three different techniques regarding the economic incentives around the tokens on the network.

4.2. Blockchains: Layer 2

In this Section, we explore, for every Layer 2 solution considered, which are the security assumptions that they need to rely on: both properties and specific security assumptions they are inheriting from Layer 1, and which additional security assumptions they also require. The term inherit in this context denotes the Layer 2 solution's requirement for a given property or security assumption which is already required by the underlying Layer 1 solution (thus no additional guarantees are added for the Layer 2 solution to be secure). We also consider security assumptions that are inherited from Layer 1, but modified if they rely and implement the "same" security assumption as in Layer 1, but modified to make it compatible with the Layer 2 solution.

4.2.1. Payment Channel Networks

As we have stated in Section 3.4.1, PCNs allow to create a peer-to-peer network on top of an existing blockchain that makes it possible to detach from the limitations imposed by the Layer 1, and perform as many transactions as desired (limited, usually, by the underlying hardware running these off-chain transactions).

Let us review the inherited, inherited but modified, and additional security assumptions PCNs need to work properly.

Inherited. PCNs rely on the following specific security assumptions.

Cryptographically secure hash function. PCNs make use of hash functions in order to execute multi-hop payments [142]. In particular, the infeasibility of finding preimages on the hash function is vital since multi-hop payments make use of Hash Time-Locked Contracts (HTCL) [143] and those, in turn, are secure as long as it is infeasible to find preimages on the hash functions on a reasonable amount of time. Since Layer 2 transactions are also valid transactions on Layer 1, they use the same hash functions.

Secure digital signatures. Since transactions made using Payment Channel Networks are actual –valid– "regular" transactions, PCN rely on secure digital signatures to sign those transactions and provide with security when sending the transactions over the channel. PCNs make use of digital signatures in order to sign and verify off-chain transactions (among other functions). For this reason, PCNs assume that the digital signature used on Layer 2 is also secure (recall Section 4.1.4).

Consensus is safe and live. PCNs must assume that the underlying consensus is safe and live because if any party in the channel wishes to cheat and broadcast an old state of the channel, the other party must be capable of responding by sending a transaction to the underlying Layer 1 and obtaining confirmation, even in scenarios where the other party stops answering or disappears. Moreover, transactions sent onto Layer 1 still need to be validated for correctness, including verifications to ensure that only the owner of the funds can redeem the transaction.

(Eventual) Censorship-resistance. Payment Channel Networks rely on timelocks [144] to handle the non-cooperative closing case. This is the case where one end of the channel closes it unilaterally. In order to ensure soundness and prevent someone from redeeming a favorable –non current– transaction to the underlying network, PCN use timelocks. Additionally, timelocks along with (eventual) censorship-resistance are also used on multi-hop payments. Those are the payments that are

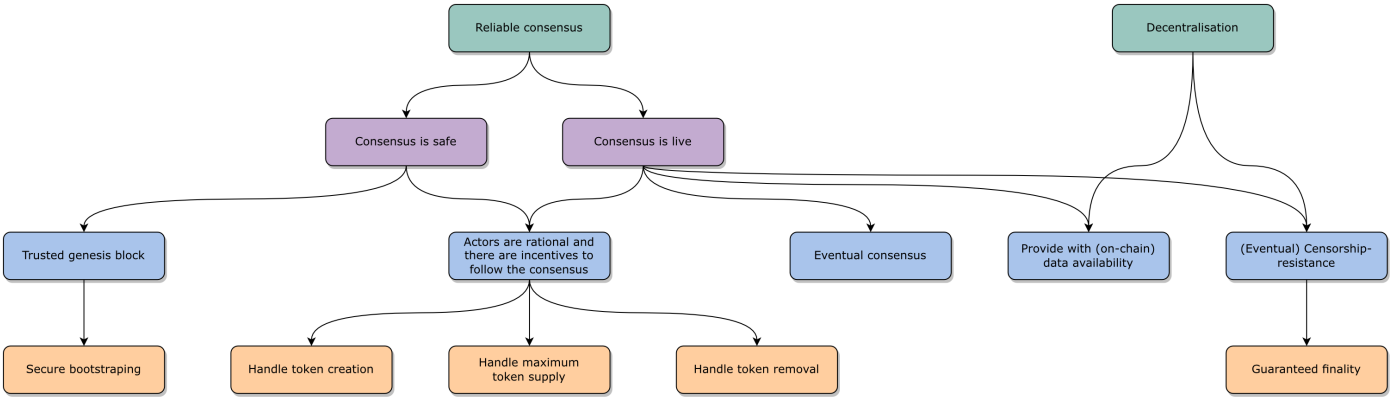

Figure 5: Specific security assumptions from reliable consensus and decentralization.

routed through a set of intermediaries in order to reach the final recipient. For the aforementioned reasons, PCN needs to rely on (eventual) censorship-resistance over the Layer 1, thus, it is impossible to censor a certain transaction and/or party from publishing transactions into Layer 1 for an arbitrary long amount of time.

Inherited, but modified.

Actors act rationally. Besides inheriting and assuming rational behavior from Layer 1, on PCNs, there are a number of of situations where actors involving the L2 system are assumed to act rationally. However, in this study, we highlight one of them: unilateral channel closing. It is expected that the party wanting to close the channel broadcasts the last commitment transaction, since broadcasting an older commitment transaction would result in a potential loss of funds.

Additional.

Bridge smart contract is secure. The way PCNs open and close channels is directly using regular Bitcoin transactions (although their scripts are more complex than a simple payment to an address). For this reason, it must be assumed that the "bridge smart contract" (that is, the scripts included in the transactions opening the channel) used to open the channel is secure.

Node is always online. Since PCNs rely on timelocks in order to ensure soundness, they need to assume additionally that both ends of the channel are continuously online, monitoring the blockchain for possible revoked commitment transactions that could close the channel in no-longer valid states. Nodes that want to use PCNs without being always online may delegate this responsibility to Watchtowers [145, 146, 147], services that monitor the underlying blockchain and automatically respond to fraudulent unilateral channel closings by broadcasting penalty transactions on behalf of the legitimate owner.

Although PCNs make use of additional cryptographic primitives at the networking level [148], we skip them because they are out of the scope of this study.

4.2.2. *Optimistic Rollups*

Recall from Section 3.4.2 that Optimistic Rollups serve the purpose to gather transactions together and present only state updates to Layer 1, securing the system using fraud proofs. This section reviews the inherited, inherited but modified, and additional security assumptions for Optimistic Rollups.

Inherited.

Cryptographically secure hash functions. Optimistic Rollups make use of hash functions in its inner workings (in particular, they are used to compute the Merkle root [80] when encoding the pre- and post-state of the Layer 2). Typically, since rollups are dealing with plain Layer 1 transactions, they make use of the same implementation of the underlying hash function, inheriting it.

Consensus is safe and live. Unlike PCNs, Optimistic Rollups send batched transactions to Layer 1 and depend on a safe and live underlying consensus to validate that the state transition is properly done. For this reason, rollups need to rely on a safe and live consensus on the underlying blockchain.

(Eventual) Censorship-resistance. Optimistic Rollups rely on Layer 1 for censorship-resistance. In an Optimistic Rollup, a centralized entity (usually called *the Operator*) is responsible for processing and executing transactions, and submitting rollup blocks to Layer 1. This approach has censoring-related consequences. On the one hand, the rollup Operator can directly censor users by going completely offline, or by refusing to produce blocks that include certain transactions in them. On the other hand, rollup Operators can also prevent users from withdrawing funds deposited in the rollup contract by withholding state data necessary to build the Merkle proof of ownership (in the case of Ethereum). In order to avoid this problem, Optimistic Rollups usually enable the users to send a *forced Layer 1 transaction*, where the users have a way to directly send the transaction to Layer 1 in the case of a non-cooperative Operator. Additionally, since fraud

proofs are a time-sensitive matter, Optimistic Rollups need from (eventual) censorship-resistance in order to ensure that a challenge to a state transition in the system will always be taken into account on time.

Provide with (on-chain) data availability. Optimistic Rollups send the actual data of the transactions processed off-chain into Layer 1 to provide with data availability. That is, they send the plain text transaction data to be written on the blockchain directly. For this reason, they need to assume that the consensus is safe and live so they can access the data in order to reconstruct and recompute the state of the blockchain in case of need.

Inherited, but modified.

Secure digital signatures. Since Optimistic Rollups are dealing with actual Layer 1 transactions, they need to ensure that those are properly signed and managed. For this reason, Optimistic Rollups need to assume that the digital signature used to sign the transactions is secure. It is usually the case that the signature used is the same that the one used on Layer 1, in order to improve compatibility. However, since Optimistic Rollups "compress" transactions, they can be implementing other types of digital signatures[21] that enable *aggregating* digital signatures into a single "master" digital signature (e.g., BLS [149] signatures).

Actors act rationally. Optimistic Rollups rely on fraud proofs in order to ensure a correct state transition. In particular, they rely on the fact that if anyone is caught trying to submit an incorrect state transition, this entity is punished by an slashing mechanism of the "bond" when proposing a new batch. For this reason, in order to provide with soundness to the system, Optimistic Rollups need to assume that the actors involved on the system are rational and follow the proper rules when proposing new batches to avoid getting punished.

Additional.

Bridge smart contract is secure. Considering that Optimistic Rollups run a whole new network in parallel of the existing Layer 1 network, they usually rely on a smart contract that "bridges" the assets from one network to another [150]. As we have seen in Section 1, bridges are a clear target of attacks, so they need to be secure and trustworthy. In order to achieve that, it is encouraged the audit of the smart contracts involved on the bridge, for example, through Quantstamp [151], OpenZeppelin [152] or Veridise [153]. For more information regarding implementing secure smart contracts bridges, we refer to McCorry et al. [154].

[21] While this may be possible to be implemented, to the best of our knowledge, it is not currently implemented on any of Ethereum's mainnet Optimistic Rollups.

Fraud proofs have a wide enough time window. Optimistic Rollups rely on *fraud proofs* in order to validate the transition state from one block to another. In an Optimistic Rollup, it is assumed that anyone submitting a rollup block is well behaved and a fraud proof is presented by any other party on the system proving that the state transition is invalid. Due to the nature of this proof, a wide enough time window is required to give enough time to other participants to evaluate and challenge the state transition. State-of-the-art implementations of Optimistic Rollups currently allow for a time window of seven days [155].

4.2.3. *Zero-Knowledge Rollups*

Recall, from Section 3.4.2, that Zero-Knowledge Rollups, likewise Optimistic Rollups, serve the purpose of compressing computation (in this case, transactions in a blockchain) and presenting only state updates to Layer 1 but, whereas Optimistic Rollups make use of fraud proofs to secure the solution, Zero-Knowledge Rollups make use of Zero-Knowledge Proofs to securise the state transitions. This Section presents the inherited, inherited but modified, and additional security assumptions for Zero-Knowledge Rollups.

Inherited.

Consensus is safe and live. Likewise Optimistic Rollups, Zero-Knowledge Rollups also send the plain –batched– transactions to Layer 1, while relying on the safety and liveness of the underlying consensus to verify the Zero-Knowledge Proof, ensuring a correct state transition from batch to batch. Thus, for this reason, they rely on a safe and live consensus mechanism on Layer 1.

(Eventual) Censorship-resistance. Similarly to Optimistic Rollups, Zero-Knowledge Rollups use a *Sequencer* that is responsible for executing transactions, producing batches and submitting blocks to Layer 1. This entity is also responsible for transaction ordering. A possible censorship problem is avoided in Zero-Knowledge Rollups by providing data availability (by relying on a safe and live consensus) and by offering a way to submit a censored transaction directly to Layer 1. Just like Optimistic Rollups, Zero-Knowledge Rollups typically implement *forced Layer 1 transactions* as well, enabling the users to directly interact with Layer 1 when facing a non-cooperative Operator. For this reason, Zero-Knowledge Rollups need to assume that the underlying Layer 1 is (eventually) censorship-resistant.

Provide with (on-chain) data availability. See Provide with on-chain data availability in Section 4.2.2.

Inherited, but modified.

Cryptographically secure hash functions. Zero-Knowledge Rollups make use of hash functions in order to properly work (just like Optimistic Rollups, the state of the

Layer 1 and Layer 2 is encoded through a Merkle tree). For this reason, the hash functions used must be secure. However, in this case, Zero-Knowledge Rollups usually prefer to implement a "Zero-Knowledge Friendly" hash function, that enables an easier implementation of the proof. For this reason, other hash functions (e.g., Poseidon [156]) are used in this Layer 2 solution.

Secure digital signatures. See Secure Digital Signatures in Section 4.2.2.

Actors act rationally. ZK-Rollups typically involve many actors in the system. Those actors can be categorized as *Sequencers*, whose main functionality is to order and evaluate transactions; and *Aggregators* (or *Executors*), whose main functionality is to take the traces produced by Sequencers when evaluating the transactions and to produce a proof of its state transition. Thus, in order to encourage good behavior of those actors, economic incentives must be provided. This is accomplished using a number of different techniques. In particular, we can distinguish two different incentives:

Regarding transaction ordering. The Sequencer earns transaction fees paid directly by L2 users when submitting transactions. This is usually paid in the native L2 token (e.g., Bridged Ether in case of Ethereum). The amount paid depends on the gas price, which is set by the users based on how much they are willing to pay for the execution of their transaction.

Regarding proving. To incentivize the Aggregator for each batch sequenced, this flow is typically used: the Sequencer locks a number of tokens on the L1 contract proportional to the number of batches that are sequenced and are to be proved. Once a proof of the batches is provided, the Aggregator can claim this reward.

Thus, to maximize its income, the Sequencer would prioritize transactions with higher gas prices.

Additional.

Bridge smart contract is secure. See Bridge smart contract is secure in Section 4.2.2.

Conversion from zkEVM to EVM is correctly done. Zero-Knowledge Rollups feature an additional framework that translates the EVM instructions into a zkEVM (zero-knowledge Ethereum Virtual Machine) that includes the "Zero-Knowledge" capabilities on each "regular" instruction. The way this conversion is handled depends on each particular implementation. Zero-Knowledge Rollups assume that this conversion is properly done. Typically, this is carried out by "translating" each operational code adding the Zero-Knowledge capabilities (e.g., Polygon zkEVM [157] implements a ROM that translates each OPCODE from the Ethereum Virtual Machine to their Zero-Knowledge Ethereum Virtual Machine [158]), or by implementing a custom compiler that takes a "regular" program and compiles that source code to a Zero-Knowledge capable executable (e.g., zkSync Era [159] use their compiler to translate Solidity or Vyper code into their zkEVM bytecode [160]).

Zero-Knowledge framework is secure and sound. The way a Zero-Knowledge Proof is generated and verified varies from framework to framework. Each Zero-Knowledge framework has a trade-off between a set of different characteristics, including proof generation time, verification time, and length of the proof. Modern Zero-Knowledge frameworks are built from the conjunction of a functional commitment scheme (e.g., a polynomial commitment scheme [161]) and an Interactive Oracle Proof (IOP) [162]. Both cryptographic primitives need to be secure and trustworthy. For an study regarding the threads around building SNARK-based Zero-Knowledge Frameworks, we refer to Chaliasos et al. [163].

Trusted Setup is correctly generated.[22] Zero-Knowledge Rollups making use of a Zero-Knowledge Framework that relies on a Trusted Setup or a Common Reference String (CRS) also assume that the "toxic waste" used to generate this setup is safely and properly disposed. Otherwise, a leaked toxic waste can be used to generate false proofs that the Verifier (in this case, the Layer 1) will accept as valid (an example, by hand, about how this can be achieved can be found in [164].).

4.2.4. Validium

Recall from Section 3.4.2 that Validiums are a kind of Zero-Knowledge Rollup that rely on *off-chain* data availability when publishing the plain transactions of the batches. This section presents the inherited, inherited but modified, and additional security assumptions for Validiums.

Inherited.

Consensus is safe and live. See Consensus is safe and live in Section 4.2.3.

(Eventual) Censorship-resistance. See (Eventual) Censorship-resistance in Section 4.2.3.

Inherited, but modified.

Secure digital signatures. See Secure digital signatures in Section 4.2.2.

Cryptographically secure hash functions. See Cryptographically secure hash functions in Section 4.2.3.

Actors act rationally. See Incentives to follow the rules for the Aggregator and the Sequencer in Section 4.2.3.

[22] Only for Zero-Knowledge Frameworks that require a Trusted Setup (usually SNARK-based).

Additional.

Provide with (off-chain) data availability. Since Validiums do not provide with on-chain data availability, they need to provide with other means to ensure data availability. Without data availability, an offline (or malicious) Operator can lock the funds indefinitely on the Validium without possibility of withdrawal. For this reason, it is crucial for the proper working of Validium that they have means to ensure access to the necessary data to rebuild the state of the network at any given time.

Conversion from zkEVM to EVM is correctly done. See Conversion from zkEVM to EVM is correctly done in Section 4.2.3.

Zero-Knowledge framework used is secure and sound. See Zero-Knowledge framework used is secure and sound in Section 4.2.3.

Trusted Setup is correctly generated.[23] See Trusted Setup is correctly generated in Section 4.2.3.

Bridge smart contract is secure. See Bridge smart contract is secure in Section 3.4.2.

4.2.5. Volition

Volition behaves like a Zero-Knowledge Rollup if it provides with data availability to Layer 1, and behaves like a Validium otherwise. Thus, it inherits and adds the security assumptions of each technology depending on the "mode" the Volition is running.

4.2.6. Summary

Table 4 summarizes the inherited, inherited but modified, and additional security assumptions for each of the studied Layer 2 solutions.

Regarding the **security assumptions directly inherited** from Layer 1, we can see that all of them assume that the underlying network features a safe and live consensus, and that it provides with (eventual) censorship-resistance. Moreover, they all rely on cryptographically secure hash functions and digital signatures (either directly inheriting those properties from Layer 1, or by slightly modifying either of those security assumptions); and assume that all actors involved the systems (Layer 1 and Layer 2) act rationally.

In addition to that, both kinds of rollups (Optimistic Rollups and Zero-Knowledge Rollups) rely on Layer 1 to provide with on-chain data availability. Validum rely on off-chain data availability instead, and Volition has this dynamic behavior that can act as either Zero-Knowledge Rollup or Validium. It is important to note that Payment Channel Networks do not need to rely on data availability for a proper functioning.

Regarding the **additional security assumptions** required by Layer 2 solutions, we observe that all of them rely on a secure bridge smart contract.

[23] Only for Zero-Knowledge Frameworks that require a Trusted Setup (usually SNARK-based).

Payment Channel Networks, on top of that, only need to assume that the node is always online; and Optimistic Rollups require to assume that there is a wide enough time window set up so any party on the system has sufficient time to submit a fraud proof.

Finally, Zero-Knowledge Rollups (and its two derivatives Validium and Volition) need to assume three additional security assumptions, all of them regarding Zero-Knowledge Proofs. Namely, they are required to assume that the zkEVM to EVM conversion is done correctly, that the Zero-Knowledge framework used is secure and sound, and, when a SNARK-based Zero-Knowledge Schema is used, they also need to assume that the Trusted Setup is correctly generated (disposing the toxic waste safely, or by generating it through Multi-Party Computation).

It is important to note that all the Layer 2 solutions need to rely on additional security assumptions in order to have a sound Layer 2 scalability system. Nonetheless, the different types of solutions offer a trade-off between additional security assumptions and benefits in terms of usability and cost. For an study about this trade-off, we refer to [165].

5. Discussion

In this Section, we first expose the practical implications we envision for our study. Then, we review a number of attacks that have happened on the Layer 1 and/or Layer 2 ecosystem in which the violation of key security assumption (or many of them) led to real-world vulnerabilities. Finally, we present the trade-offs between Security, Scalability, and Decentralization that the analyzed Layer 2 solution has.

5.1. Practical Implications

The findings of our study have direct practical applications for developers, auditors, and protocol designers working on Layer 2 blockchain solutions. By systematically identifying and analyzing the security assumptions underlying various Layer 2 mechanisms, our work provides a foundation for evaluating the robustness of these systems before deployment. Developers can use our categorization to better understand the trade-offs involved in different scalability approaches, while auditors and security researchers can rely on it as a framework for threat modeling and risk assessment. This contributes to building more secure and transparent Layer 2 infrastructures.

Furthermore, our analysis supports informed decision-making for ecosystem stakeholders, such as governance bodies, foundations, and investors, who must evaluate the long-term sustainability of Layer 2 technologies. By revealing how security assumptions vary across implementations and how they extend (or diverge from) those of the underlying Layer 1, our work highlights areas that require more scrutiny and potential standardization. In this sense, our study contributes not only to academic understanding but also to the responsible and secure scaling of blockchain networks in practice.

Table 4: Summary of security assumptions of Layer 2.

	Payment Channel Networks	Optimistic Rollups	Zero-Knowledge Rollups	Validium	Volition
Cryptographically secure hash functions (on L2 as well)	○	○	◐[1]	◐[1]	◐[1]
Secure digital signatures (on L2 as well)	○	◐[2]	◐[2]	◐[2]	◐[2]
Consensus is safe and live	○	○	○	○	○
(Eventual) Censorship-resistance	○	○	○	○	○
Provide with data availability		○	○	●[3]	◐[4]
Actors act rationally (on L2 as well)	◐[5]	◐[6]	◐[6]	◐[6]	◐[6]
Bridge smart contract is secure	●	●	●	●	●
Node is always online	●				
Fraud proofs have a wide enough time window		●			
Conversion from zkEVM to EVM is correctly done			●	●	●
Zero-Knowledge framework used is secure and sound			●	●	●
Trusted Setup is correctly generated (toxic waste safely disposed or generated using MPCs)			●[7]	●[7]	●[7]

○ Inherited from Layer 1.
◐ Inherited from Layer 1, but modified.
● New additional security assumption.
[1] Zero-Knowledge Rollups may be using other "Zero-Knowledge Friendly" hash functions (e.g., Poseidon [156]).
[2] Rollups may be using other signatures that enable them to aggregate several digital signatures together (e.g., BLS Signatures).
[3] Validium relies on off-chain infrastructure to provide with data availability.
[4] Volition relies on on-chain data availability when running in "Zero-Knowledge Rollup" mode, and provides with off-chain data availability otherwise.
[5] Actors on PCN need to act rationally when routing multi-hop payments, and when handling the closing of the channel.
[6] Actors in rollups need to act rationally and to have incentives to follow the rules.
[7] Only for Zero-Knowledge frameworks that require a Trusted Setup (usually SNARK-based).

5.2. *Empirical Validation*

To bridge the gap between theory and practice, this Section examines real-world incidents where violated security assumptions led to severe compromises in blockchain systems. By revisiting these cases, we aim to demonstrate how abstract security models translate into concrete vulnerabilities, and how a deeper understanding of foundational assumptions can help prevent similar failures in future designs.

For a more in-depth study of known attacks to Layer 2 solutions, we refer to [4, 166].

1. **Ethereum Classic 51% Attacks (2020–2021).** Attackers monopolized hash power to reverse transactions, violating the honest-majority assumption in PoW systems [167]. **Security assumption violated:** Reliable consensus (safety and liveness).

2. **Ronin Bridge Hack (2024).** A contract upgrade introduced dead code, leaving `minimumVoteWeight` uninitialized, which is a key security check on the contract [168]. **Security assumption violated:** Bridge smart contract is secure.

3. **zkSync Era ZK-Rollup Vulnerability (2023).** Flaws in zkEVM proof generation allowed balance spoofing [169]. **Security assumption violated:** Zero-Knowledge framework used is secure and sound.

4. **Theoretical StarkEx Data Withholding (2022).** This is a theoretical attack where a Validium operator temporarily locked funds by withholding data [170]. **Security assumption violated:** Provide with data availability.

5. **Possible ZCash Toxic Waste Leak (2018).** Flawed setup

ceremonies potentially exposed toxic waste providing with the ability to forge proofs [171]. **Security assumption violated:** Trusted Setup is correctly generated (toxic waste safely disposed or generated using MPCs).

6. **Lightning Network Balance LockDown (2020).** Attackers temporarily lock a victim's funds in a payment channel by initiating HTLCs that never complete, preventing the victim from using their balance until the timelocks expire [172]. **Security assumption violated:** Actors act rationally (on L2 as well).

7. **DoubleUp Roll: Double-spending in Arbitrum.** The attack exploits state rollback mechanisms in Optimistic Rollups by strategically inducing delays in transaction finality, enabling double-spending through cross-chain application vulnerabilities that trust soft-finalized L2 transactions before hard-finalization on Ethereum [173]. **Security assumptions violated:** (Eventual) Censorship-Resistance, and Actors on L2 have incentives to follow the rules.

For an in-depth analysis of attack vectors found on public blockchains, we refer to Section 3 on [174].

5.3. Trade-offs of Layer 2 solutions

Layer 2 solutions present distinct approaches to balance the Blockchain Trilemma. Our analysis reveals critical security-performance (scalability)-decentralization trade-offs across implementations, which will be examined in detail in the following Sections.

5.3.1. Security-Performance Trade-offs in Layer 2 solutions

The current landscape of Layer 2 solutions present, inherently, security and performance (scalability) trade-offs. In particular, they present these trade-offs when handling data availability, and when considering state validation.

Data Availability Trade-offs.

- **Rollups**: Maintain full L1 data availability for robust security at the cost of limit throughput to around $\approx \times 15$ L1 TPS due to on-chain storage costs [175].

- **Validium**: Achieves $\approx \times 30$ TPS [176] through off-chain data storage but introduces trusted custody models, since a majority of data committee members can temporarily freeze withdrawals [170].

- **Volition**: Hybrid model allowing per-transaction DA selection allows for cost reduction on selected non-critical transactions, while maintaining the ability to transact like in a regular rollup when desired.

State Validation Trade-offs.

- **Fraud Proofs (Optimistic)**: 7-day withdrawal delays prevent real-time DeFi use, but enable full EVM compatibility with great gas cost reduction compared to L1.

- **Validity Proofs (Zero-Knowledge)**: Instant finality enables sub-2min withdrawals but requires specialized zkEVMs introducing new attack surfaces.

Please note that PCNs do not need to provide with DA nor they rely on state validation to be secure, so these trade-offs are not applicable to this technology.

5.3.2. Security-Decentralization Trade-offs in Layer 2 solutions

The security-decentralization trade-off manifests acutely in Sequencer and Executor (Prover) architectures, and data availability models. In particular, Sequencer and Executor centralization plays a fundamental role about the decentralization of the scaling technology given its important role on the network. Regarding the Data Availability Committees, they are a key component of the security of technologies like Validium, thus, the more decentralized they are, the less trust needs to be assessed, increasing security.

Sequencer and Executor Centralization Trade-offs.

- To the best of our knowledge, currently all production-ready rollups run over a centralized single Sequencer, creating single points of failure.

- Emerging decentralized Sequencer pools (e.g., Espresso Systems [177]) show latency increase versus centralized models but prevent transaction censorship, and increase decentralization.

- While the crafting of Zero-Knowledge Proofs could be easily decentralized, recent research points out that specialized hardware is needed in order to create the proofs about the state transitions of the blockchain [178]. The high Prover requirements are a steep wall to overcome when considering joining the Executor network of Provers.

Data Availability Committees.

- Validium's Data Availability Committees typically are formed by 8 different parties. A compromise of 5 out of 8 nodes could freeze >$1B worth in assets [179].

5.3.3. Hybrid Approaches to workaround the Blockchain Trilema

In an attempt to workaround the Blockchain Trilema, and its trade-offs, hybrid approaches like Volition have emerged.

Volition's Adaptive Security.

- On Volition, users can select per-transaction data availability: high-value transfers can make use of L1 data availability (more expensive), while microtransactions can go through off-chain data availability (much cheaper).
- Dynamic proofs allow for the mixing of validity and fraud proofs for increased convenience, while maintaining low fees and costs [180].

This analysis demonstrates that while no Layer 2 solution fully resolves the Trilemma, hybrid architectures significantly advance the security-scalability frontier. We advise users to evaluate these trade-offs against specific use case requirements: high-value DeFi may prioritize ZK-Rollup for its security, while gaming applications could opt for Validium's throughput despite reduced decentralization guarantees.

For an in-depth comparison about Layer 2 solutions, please refer [165, 181].

6. Conclusions and Future Work

The landscape of blockchain technology continues to evolve, highlighting scalability as a primary challenge for which Layer 2 solutions have become the leading approach. However, these rapidly emerging solutions often lack thorough documentation and examination of their underlying security assumptions.

Layer 2 solutions rely on Layer 1 fundamentals that ensure security and transaction inclusion. These fundamentals include correct block finalization, censorship resistance, and the use of robust cryptographic primitives, as well as the collaboration of actors with well-aligned economic incentives. Without these principles, they could not guarantee their functionalities or offer meaningful improvements over the base layer.

In addition, Layer 2 solutions introduce further assumptions—most notably, the security of bridging smart contracts that link Layer 2 solutions back to the main chain. These bridge contracts must remain free of exploitable vulnerabilities to protect transacted and hold assets. Other supplementary assumptions vary by approach, with some solutions requiring additional trust in on- or off-chain components.

Our findings underscore the importance of explicitly defining and rigorously evaluating these assumptions to ensure robust and trustworthy scalability solutions.

Several avenues remain open for future research to deepen the understanding of Layer 2 security. First, the development of formal models to rigorously capture the security assumptions underlying Layer 2 protocols would enable more precise threat analysis and verification. Second, as governance mechanisms (particularly those involving decentralized autonomous organizations , DAOs) play a central role in protocol upgrades and emergency responses, studying their security implications and failure modes is essential. Third, the economic incentives that drive honest participation in Layer 2 systems remain underexplored; future work could analyze how deviations from assumed economic behaviors might expose systems to manipulation or DoS attacks. Fourth, many security guarantees rely on idealized user behavior; understanding how real users interact with Layer 2 systems could reveal vulnerabilities that arise from usability gaps or non-rational user behavior. Finally, a deeper analysis of the consequences of each assumption failing would provide valuable guidance for both developers and researchers, helping to prioritize mitigations and improve system resilience in practice.

Acknowledgements

We would like to thanks Prof. David Megías for proofreading and correcting this writing.

This work is linked to the projects PID2021-125962OB-C33 SECURING/NET and PID2021-125962OB-C31 SECURING/CYBER, funded by the Ministerio de Ciencia e Innovación, la Agencia Estatal de Investigación and the European Regional Development Fund (ERDF), as well as the ARTEMISA International Chair of Cybersecurity and the DANGER Strategic Project of Cybersecurity C062/23, both funded by the Spanish National Institute of Cybersecurity through the European Union - NextGenerationEU and the Recovery, Transformation and Resilience Plan; and the Catalan Agència de Gestió d'Ajuts Universitaris i de Recerca (AGAUR) grants SGR2021-00643 and SGR2021-01508. Moreover, A. Torralba-Agell is funded by grant 2023 FI-1 00241 from the Catalan Agència de Gestió d'Ajuts Universitaris i de Recerca (AGAUR), and with the support of a doctoral grant from the UOC.

References

[1] Hai Wang, Yong Wang, Zigang Cao, Zhen Li, and Gang Xiong. An overview of blockchain security analysis. In *Cyber security: 15th international annual conference, CNCERT 2018, beijing, china, august 14–16, 2018, revised selected papers 15*, pages 55–72. Springer Singapore, 2019.

[2] Sumit Soni and Bharat Bhushan. A Comprehensive survey on Blockchain: Working, security analysis, privacy threats and potential applications. In *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, volume 1, pages 922–926, July 2019. doi: 10.1109/ICICICT46008.2019.8993210. URL https://ieeexplore.ieee.org/document/8993210/.

[3] Chao Yu, Wenke Yang, Feiyu Xie, and Jianmin He. Technology and Security Analysis of Cryptocurrency Based on Blockchain. *Complexity*, 2022(1):5835457, 2022. ISSN 1099-0526. doi: 10.1155/2022/5835457. URL https://onlinelibrary.wiley.com/doi/abs/10.1155/2022/5835457. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1155/2022/5835457.

[4] Ankit Gangwal, Haripriya Ravali Gangavalli, and Apoorva Thirupathi. A survey of Layer-two blockchain protocols. *Journal of Network and Computer Applications*, 209:103539, January 2023. ISSN 1084-8045. doi: 10.1016/j.jnca.2022.103539. URL https://www.sciencedirect.com/science/article/pii/S1084804522001801.

[5] Adrian Koegl, Zeeshan Meghji, Donato Pellegrino, Jan Gorzny, and Martin Derka. Attacks on Rollups. In *Proceedings of the 4th International Workshop on Distributed Infrastructure for the Common Good*, pages 25–30, Bologna Italy, December 2023. ACM. ISBN 979-8-4007-0458-1. doi: 10.1145/3631310.3633493. URL https://dl.acm.org/doi/10.1145/3631310.3633493.

[6] Zekai Liu and Xiaoqi Li. SoK: Security Analysis of Blockchain-based Cryptocurrency, March 2025. URL http://arxiv.org/abs/2503.22156. arXiv:2503.22156 [cs].

[7] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. 2009.

[8] Syada Tasmia Alvi, Mohammed Nasir Uddin, Linta Islam, and Sajib Ahamed. DVTChain: A blockchain-based decentralized mechanism to ensure the security of digital voting system voting system. *Journal of King Saud University - Computer and Information Sciences*, 34(9):6855–6871, October 2022. ISSN 1319-1578. doi: 10.1016/j.jksuci.2022.06.014. URL https://www.sciencedirect.com/science/article/pii/S1319157822002221.

[9] Clare Sullivan and Eric Burger. Blockchain, Digital Identity, E-government. In Horst Treiblmaier and Roman Beck, editors, *Business Transformation through Blockchain: Volume II*, pages 233–258. Springer International Publishing, Cham, 2019. ISBN 978-3-319-99058-3. doi: 10.1007/978-3-319-99058-3_9. URL https://doi.org/10.1007/978-3-319-99058-3_9.

[10] Decentralized applications (dapps), . URL https://ethereum.org.

[11] Decentralized finance (DeFi), . URL https://ethereum.org.

[12] Non-fungible tokens (NFT). URL https://ethereum.org.

[13] Top Blockchain Games. URL https://dappradar.com/rankings/category/games.

[14] Dalila Ressi, Riccardo Romanello, Carla Piazza, and Sabina Rossi. AI-enhanced blockchain technology: A review of advancements and opportunities. *Journal of Network and Computer Applications*, 225: 103858, May 2024. ISSN 1084-8045. doi: 10.1016/j.jnca.2024.103858. URL https://www.sciencedirect.com/science/article/pii/S1084804524000353.

[15] Ethereum average transaction fee chart (in USD), . URL https://blockchair.com/ethereum/charts/average-transaction-fee-usd.

[16] Rekt - Leaderboard. URL https://www.rekt.news/.

[17] yyctrader. Axie Infinity's Ronin Bridge Exploited For More Than $600M - "The Defiant", March 2022. URL https://thedefiant.io/news/nfts-and-web3/axie-infinity-hack-600m.

[18] Hackers steal more than $600 million from maker of Axie Infinity, March 2022. URL https://www.nbcnews.com/tech/tech-news/hackers-steal-600-million-maker-axie-infinity-rcna22031.

[19] Junfeng Xie, F. Richard Yu, Tao Huang, Renchao Xie, Jiang Liu, and Yunjie Liu. A Survey on the Scalability of Blockchain Systems. *IEEE Network*, 33(5):166–173, September 2019. ISSN 1558-156X. doi: 10.1109/MNET.001.1800290. Conference Name: IEEE Network.

[20] Abdelatif Hafid, Abdelhakim Senhaji Hafid, and Mustapha Samih. Scaling Blockchains: A Comprehensive Survey. *IEEE Access*, 8: 125244–125262, 2020. ISSN 2169-3536. doi: 10.1109/ACCESS.2020.3007251. URL https://ieeexplore.ieee.org/document/9133427/. 69 citations (Crossref) [2023-02-14].

[21] Ankit Gangwal, Haripriya Ravali Gangavalli, and Apoorva Thirupathi. A survey of Layer-two blockchain protocols. *Journal of Network and Computer Applications*, 209:103539, January 2023. ISSN 10848045. doi: 10.1016/j.jnca.2022.103539. URL https://linkinghub.elsevier.com/retrieve/pii/S1084804522001801. 2 citations (Crossref) [2023-02-14].

[22] Louis Tremblay Thibault, Tom Sarry, and Abdelhakim Senhaji Hafid. Blockchain Scaling Using Rollups: A Comprehensive Survey. *IEEE Access*, 10:93039–93054, 2022. ISSN 2169-3536. doi: 10.1109/ACCESS.2022.3200051. URL https://ieeexplore.ieee.org/document/9862815/. 2 citations (Crossref) [2023-02-14].

[23] Visa Transactions per Second. URL https://usa.visa.com/run-your-business/small-business-tools/retail.html.

[24] Arthur Gervais, Ghassan O. Karame, Karl Wüst, Vasileios Glykantzis, Hubert Ritzdorf, and Srdjan Capkun. On the Security and Performance of Proof of Work Blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 3–16, Vienna Austria, October 2016. ACM. ISBN 978-1-4503-4139-4. doi: 10.1145/2976749.2978341. URL https://dl.acm.org/doi/10.1145/2976749.2978341.

[25] Ethereum blockchain size chart, . URL https://blockchair.com/ethereum/charts/blockchain-size.

[26] Bitcoin - Open source P2P money, . URL https://bitcoin.org/en/.

[27] Ethereum Homepage, . URL https://ethereum.org.

[28] Solana | Web3 Infrastructure for Everyone. URL https://solana.com/es.

[29] StarkWare. Redefining Scalability, December 2021. URL https://medium.com/starkware/redefining-scalability-5aa11ffc5880.

[30] What is the Blockchain Trilemma? URL https://www.ledger.com/academy/what-is-the-blockchain-trilemma.

[31] Nano Foundation | Goals, alliances, team and advisors. URL https://nano.org/en/nano-foundation.

[32] IOTA. URL https://www.iota.org.

[33] Home | XRPL.org, . URL https://xrpl.org/.

[34] Home – EOSIO Blockchain Software & Services, . URL https://eos.io/.

[35] Francesco Mogavero, Ivan Visconti, Andrea Vitaletti, and Marco Zecchini. The Blockchain Quadrilemma: When Also Computational Effectiveness Matters. In *2021 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6, September 2021. doi: 10.1109/ISCC53001.2021.9631511. ISSN: 2642-7389.

[36] Abdurrashid Ibrahim Sanka and Ray C.C. Cheung. A systematic review of blockchain scalability: Issues, solutions, analysis and future research. *Journal of Network and Computer Applications*, 195: 103232, December 2021. ISSN 10848045. doi: 10.1016/j.jnca.2021.103232. URL https://linkinghub.elsevier.com/retrieve/pii/S1084804521002307.

[37] The Limits to Blockchain Scalability, . URL https://vitalik.ca/general/2021/05/23/scaling.html.

[38] Why sharding is great: demystifying the technical properties, . URL https://vitalik.ca/general/2021/04/07/sharding.html.

[39] NULS. Why it is Impossible to Solve Blockchain Trilemma?, March 2019. URL https://nuls.medium.com/why-it-is-impossible-to-solve-blockchain-trilemma-c03745debb6.

[40] Introducing Data Availability Committees. URL https://research-development.nomadic-labs.com/introducing-data-availability-committees.html.

[41] Jyoti Yadav and Ranjana Shevkar. Performance-Based Analysis of Blockchain Scalability Metric. *Tehnički glasnik*, 15(1):133–142, March 2021. ISSN 1846-6168, 1848-5588. doi: 10.31803/tg-20210205103310. URL https://hrcak.srce.hr/clanak/367642%3F. Publisher: Sveučilište Sjever.

[42] Uri Klarman, Soumya Basu, Aleksandar Kuzmanovic, and Emin Gun Sirer. bloXroute: A Scalable Trustless Blockchain Distribution Network. 2019.

[43] bloXroute - DeFi trading tools, Mempool Services, Defi performance. URL https://bloxroute.com/.

[44] Nakul Chawla, Hans Walter Behrens, Darren Tapp, Dragan Boscovic, and K. Selçuk Candan. Velocity: Scalability Improvements in Block Propagation Through Rateless Erasure Coding. In *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 447–454, May 2019. doi: 10.1109/BLOC.2019.8751427.

[45] M. Luby. LT codes. In *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, pages 271–280, Vancouver, BC, Canada, 2002. IEEE Comput. Soc. ISBN 978-0-7695-1822-0. doi: 10.1109/SFCS.2002.1181950. URL http://ieeexplore.ieee.org/document/1181950/.

[46] Elias Rohrer and Florian Tschorsch. Kadcast: A Structured Approach to Broadcast in Blockchain Networks. In *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, AFT '19, pages 199–213, New York, NY, USA, October 2019. Association for Computing Machinery. ISBN 978-1-4503-6732-5. doi: 10.1145/3318041.3355469. URL https://dl.acm.org/doi/10.1145/3318041.3355469.

[47] Petar Maymounkov and David Mazières. Kademlia: A Peer-to-Peer Information System Based on the XOR Metric. In Peter Druschel, Frans Kaashoek, and Antony Rowstron, editors, *Peer-to-Peer Systems*, Lec-

ture Notes in Computer Science, pages 53–65, Berlin, Heidelberg, 2002. Springer. ISBN 978-3-540-45748-0. doi: 10.1007/3-540-45748-8_5.

[48] Gleb Naumenko, Gregory Maxwell, Pieter Wuille, Alexandra Fedorova, and Ivan Beschastnikh. Erlay: Efficient Transaction Relay for Bitcoin. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, pages 817–831, New York, NY, USA, November 2019. Association for Computing Machinery. ISBN 978-1-4503-6747-9. doi: 10.1145/3319535.3354237. URL https://dl.acm.org/doi/10.1145/3319535.3354237.

[49] Bitcoin Cash - Peer-to-Peer Electronic Cash, . URL https://bitcoincash.org/en/.

[50] Bitcoin Cash Protocol, . URL https://www.reference.cash/protocol/forks/hf-20180515.

[51] BIP 0141 - Bitcoin Wiki. URL https://en.bitcoin.it/wiki/BIP_0141.

[52] Scaling. URL https://ethereum.org.

[53] Danksharding. URL https://ethereum.org.

[54] EIP-4844: Shard Blob Transactions. URL https://eips.ethereum.org/EIPS/eip-4844.

[55] A rollup-centric ethereum roadmap, October 2020. URL https://ethereum-magicians.org/t/a-rollup-centric-ethereum-roadmap/4698/44.

[56] Ittay Eyal, Adem Efe Gencer, Emin Gün Sirer, and Robbert van Renesse. {Bitcoin-NG}: A Scalable Blockchain Protocol. pages 45–59, 2016. ISBN 978-1-931971-29-4. URL https://www.usenix.org/conference/nsdi16/technical-sessions/presentation/eyal.

[57] Zhenzhen Jiao, Rui Tian, Dezhong Shang, and Hui Ding. Bicomp: A Bilayer Scalable Nakamoto Consensus Protocol, September 2018. URL http://arxiv.org/abs/1809.01593. arXiv:1809.01593 [cs].

[58] Yonatan Sompolinsky and Aviv Zohar. Secure High-Rate Transaction Processing in Bitcoin. In Rainer Böhme and Tatsuaki Okamoto, editors, *Financial Cryptography and Data Security*, Lecture Notes in Computer Science, pages 507–527, Berlin, Heidelberg, 2015. Springer. ISBN 978-3-662-47854-7. doi: 10.1007/978-3-662-47854-7_32.

[59] Proof-of-stake (PoS). URL https://ethereum.org.

[60] Evangelos Deirmentzoglou, Georgios Papakyriakopoulos, and Constantinos Patsakis. A Survey on Long-Range Attacks for Proof of Stake Protocols. *IEEE Access*, 7:28712–28725, 2019. ISSN 2169-3536. doi: 10.1109/ACCESS.2019.2901858. Conference Name: IEEE Access.

[61] Gabriel Antonio F. Rebello, Gustavo F. Camilo, Lucas C. B. Guimarães, Lucas Airam C. de Souza, Guilherme A. Thomaz, and Otto Carlos M. B. Duarte. A security and performance analysis of proof-based consensus protocols. *Annals of Telecommunications*, 77(7):517–537, August 2022. ISSN 1958-9395. doi: 10.1007/s12243-021-00896-2. URL https://doi.org/10.1007/s12243-021-00896-2.

[62] Delegated proof of stake - Bitcoin Wiki. URL https://en.bitcoin.it/wiki/Delegated_proof_of_stake.

[63] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017*, Lecture Notes in Computer Science, pages 357–388, Cham, 2017. Springer International Publishing. ISBN 978-3-319-63688-7. doi: 10.1007/978-3-319-63688-7_12.

[64] Discover Cardano. URL https://cardano.org/discover-cardano/.

[65] Bernardo David, Peter Gaži, Aggelos Kiayias, and Alexander Russell. Ouroboros Praos: An Adaptively-Secure, Semi-synchronous Proof-of-Stake Blockchain. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018*, Lecture Notes in Computer Science, pages 66–98, Cham, 2018. Springer International Publishing. ISBN 978-3-319-78375-8. doi: 10.1007/978-3-319-78375-8_3.

[66] Giuseppe Ateniese, Ilario Bonacina, Antonio Faonio, and Nicola Galesi. Proofs of Space: When Space Is of the Essence. In Michel Abdalla and Roberto De Prisco, editors, *Security and Cryptography for Networks*, Lecture Notes in Computer Science, pages 538–557, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10879-7. doi: 10.1007/978-3-319-10879-7_31.

[67] Vitalik Buterin and Virgil Griffith. Casper the Friendly Finality Gadget, January 2019. URL http://arxiv.org/abs/1710.09437. arXiv:1710.09437 [cs].

[68] Miguel Castro and Barbara Liskov. Practical Byzantine Fault Tolerance.

[69] Eleftherios Kokoris-Kogias, Philipp Jovanovic, Nicolas Gailly, Ismail Khoffi, Linus Gasser, and Bryan Ford. Enhancing Bitcoin Security and Performance with Strong Consistency via Collective Signing, August 2016. URL http://arxiv.org/abs/1602.06997. arXiv:1602.06997 [cs].

[70] Ewa Syta, Iulia Tamas, Dylan Visher, David Isaac Wolinsky, Philipp Jovanovic, Linus Gasser, Nicolas Gailly, Ismail Khoffi, and Bryan Ford. Keeping Authorities "Honest or Bust" with Decentralized Witness Cosigning. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 526–545, May 2016. doi: 10.1109/SP.2016.38. ISSN: 2375-1207.

[71] Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. Algorand: Scaling Byzantine Agreements for Cryptocurrencies. In *Proceedings of the 26th Symposium on Operating Systems Principles*, pages 51–68, Shanghai China, October 2017. ACM. ISBN 978-1-4503-5085-3. doi: 10.1145/3132747.3132757. URL https://dl.acm.org/doi/10.1145/3132747.3132757.

[72] Joseph Poon and Thaddeus Dryja. The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments. 2016.

[73] Raiden Network. URL https://raiden.network/.

[74] Optimistic Rollups. URL https://ethereum.org.

[75] Zero-Knowledge rollups. URL https://ethereum.org.

[76] U. Feige and A. Shamir. Witness indistinguishable and witness hiding protocols. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing - STOC '90*, pages 416–426, Baltimore, Maryland, United States, 1990. ACM Press. ISBN 978-0-89791-361-4. doi: 10.1145/100216.100272. URL http://portal.acm.org/citation.cfm?doid=100216.100272. 273 citations (Crossref) [2023-02-14].

[77] ZKProof Community Reference. URL https://docs.zkproof.org/pages/reference/reference.pdf.

[78] Validium. URL https://ethereum.org.

[79] [EthCC] Livestream 3. Carlos Matallana - Introducing zkBlob and data-compression to the zkEVM, July 2023. URL https://www.youtube.com/watch?v=E4HxKHRbFZ4.

[80] Ralph C. Merkle. A Digital Signature Based on a Conventional Encryption Function. In Carl Pomerance, editor, *Advances in Cryptology — CRYPTO '87*, Lecture Notes in Computer Science, pages 369–378, Berlin, Heidelberg, 1988. Springer. ISBN 978-3-540-48184-3. doi: 10.1007/3-540-48184-2_32.

[81] Volition. URL https://ethereum.org.

[82] L. Lamport. Proving the Correctness of Multiprocess Programs. *IEEE Transactions on Software Engineering*, SE-3(2):125–143, March 1977. ISSN 1939-3520. doi: 10.1109/TSE.1977.229904. Conference Name: IEEE Transactions on Software Engineering.

[83] PeaseM, ShostakR, and LamportL. Reaching Agreement in the Presence of Faults. *Journal of the ACM (JACM)*, April 1980. doi: 10.1145/322186.322188. URL https://dl.acm.org/doi/10.1145/322186.322188. Publisher: ACM PUB27 New York, NY, USA.

[84] Michael J. Fischer, Nancy A. Lynch, and Michael S. Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM*, 32(2):374–382, April 1985. ISSN 0004-5411, 1557-735X. doi: 10.1145/3149.214121. URL https://dl.acm.org/doi/10.1145/3149.214121.

[85] Seth Gilbert and Nancy Lynch. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *ACM SIGACT News*, 33(2):51–59, June 2002. ISSN 0163-5700. doi: 10.1145/564585.564601. URL https://dl.acm.org/doi/10.1145/564585.564601.

[86] Seth Gilbert and Nancy A. Lynch. Perspectives on the CAP Theorem. *Computer*, 45(02):30–36, February 2012. ISSN 0018-9162. doi: 10.1109/MC.2011.389. URL https://www.computer.org/csdl/magazine/co/2012/02/mco2012020030/13rRUxC0SHz. Publisher: IEEE Computer Society.

[87] The Merge. URL https://ethereum.org.

[88] National Institute of Standards and Technology, June 2023. URL https://www.nist.gov/. Last Modified: 2023-06-01T11:28-04:00.

[89] Secure Hash Algorithms, June 2022. URL https://en.

wikipedia.org/w/index.php?title=Secure_Hash_Algorithms&oldid=1094940176. Page Version ID: 1094940176.
[90] RIPEMD, May 2023. URL https://en.wikipedia.org/w/index.php?title=RIPEMD&oldid=1155037448. Page Version ID: 1155037448.
[91] Vitalik Buterin. Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform.
[92] SHA-3, May 2023. URL https://en.wikipedia.org/w/index.php?title=SHA-3&oldid=1155901410. Page Version ID: 1155901410.
[93] Keccak Team. URL https://keccak.team/keccak.html.
[94] Hudson Jameson. Answer to "Which cryptographic hash function does Ethereum use?", January 2016. URL https://ethereum.stackexchange.com/a/554.
[95] Ethash, . URL https://ethereum.org.
[96] Information Technology Laboratory Computer Security Division. Hash Functions | CSRC | CSRC, January 2017. URL https://csrc.nist.gov/projects/hash-functions.
[97] Information Technology Laboratory Computer Security Division. Digital Signatures - Cryptographic Algorithm Validation Program | CSRC | CSRC, October 2016. URL https://csrc.nist.gov/Projects/cryptographic-algorithm-validation-program/digital-signatures.
[98] BIP-340, June 2023. URL https://github.com/bitcoin/bips/blob/master/bip-0340.mediawiki. original-date: 2013-11-19T17:18:41Z.
[99] BIP-341, June 2023. URL https://github.com/bitcoin/bips/blob/master/bip-0341.mediawiki. original-date: 2013-11-19T17:18:41Z.
[100] BIP-342, June 2023. URL https://github.com/bitcoin/bips/blob/master/bip-0342.mediawiki. original-date: 2013-11-19T17:18:41Z.
[101] Nils Fleischhacker, Tibor Jager, and Dominique Schröder. On Tight Security Proofs for Schnorr Signatures. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology – ASIACRYPT 2014*, Lecture Notes in Computer Science, pages 512–531, Berlin, Heidelberg, 2014. Springer. ISBN 978-3-662-45611-8. doi: 10.1007/978-3-662-45611-8_27.
[102] Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. Aggregate and Verifiably Encrypted Signatures from Bilinear Maps. In Eli Biham, editor, *Advances in Cryptology — EUROCRYPT 2003*, Lecture Notes in Computer Science, pages 416–432, Berlin, Heidelberg, 2003. Springer. ISBN 978-3-540-39200-2. doi: 10.1007/3-540-39200-9_26.
[103] *Algebraic Curves*. URL https://link.springer.com/book/9780387903613.
[104] Anthony W. Knapp. *Elliptic Curves*. Princeton University Press, 1992. ISBN 978-0-691-08559-3.
[105] Daniel R L Brown. SEC 2: Recommended Elliptic Curve Domain Parameters.
[106] Standards for Efficient Cryptography Group, . URL https://www.secg.org/.
[107] Sean Bowe. BLS12-381: New zk-SNARK Elliptic Curve Construction, March 2017. URL https://electriccoin.co/blog/new-snark-curve/.
[108] Empower people with economic freedom. URL https://electriccoin.co/.
[109] Thomas Pornin. Deterministic Usage of the Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA). Request for Comments RFC 6979, Internet Engineering Task Force, August 2013. URL https://datatracker.ietf.org/doc/rfc6979. Num Pages: 79.
[110] Nicolas T. Courtois, Pinar Emirdag, and Filippo Valsorda. Private Key Recovery Combination Attacks: On Extreme Fragility of Popular Bitcoin Key Management, Wallet and Cold Storage Solutions in Presence of Poor RNG Events, 2014. URL https://eprint.iacr.org/2014/848. Report Number: 848.
[111] Michael Brengel and Christian Rossow. Identifying Key Leakage of Bitcoin Users. In Michael Bailey, Thorsten Holz, Manolis Stamatogiannakis, and Sotiris Ioannidis, editors, *Research in Attacks, Intrusions, and Defenses*, Lecture Notes in Computer Science, pages 623–643, Cham, 2018. Springer International Publishing. ISBN 978-3-030-00470-5. doi: 10.1007/978-3-030-00470-5_29.
[112] Joachim Breitner and Nadia Heninger. Biased Nonce Sense: Lattice Attacks Against Weak ECDSA Signatures in Cryptocurrencies. In Ian Goldberg and Tyler Moore, editors, *Financial Cryptography and Data Security*, Lecture Notes in Computer Science, pages 3–20, Cham, 2019. Springer International Publishing. ISBN 978-3-030-32101-7. doi: 10.1007/978-3-030-32101-7_1.
[113] Joppe W. Bos, J. Alex Halderman, Nadia Heninger, Jonathan Moore, Michael Naehrig, and Eric Wustrow. Elliptic Curve Cryptography in Practice. In Nicolas Christin and Reihaneh Safavi-Naini, editors, *Financial Cryptography and Data Security*, Lecture Notes in Computer Science, pages 157–175, Berlin, Heidelberg, 2014. Springer. ISBN 978-3-662-45472-5. doi: 10.1007/978-3-662-45472-5_11.
[114] Ethereum on ARM documentation — Ethereum on ARM documentation 0.0.1 documentation, . URL https://ethereum-on-arm-documentation.readthedocs.io/en/latest/index.html.
[115] Geth Ethereum Client. URL https://geth.ethereum.org/.
[116] Nethermind Client | An Ethereum execution layer client. URL https://www.nethermind.io/nethermind-client.
[117] Erigon | Efficient Ethereum Client, June 2020. URL https://erigon.ch/.
[118] Lighthouse: Ethereum consensus client, October 2023. URL https://github.com/sigp/lighthouse. original-date: 2018-07-06T07:56:39Z.
[119] Lodestar Consensus Client. URL https://lodestar.chainsafe.io/.
[120] Nimbus, a Lighter Ethereum Client. URL https://nimbus.team/index.html.
[121] Client Diversity | Ethereum. URL https://clientdiversity.org.
[122] Prysmatic Labs. URL https://prysmaticlabs.com.
[123] Teku | Ethereum 2.0 Client for Institutional Staking. URL https://consensys.io/teku.
[124] MonitorEth. URL https://monitoreth.io/.
[125] Sync modes (Checkpoints). URL https://geth.ethereum.org/docs/fundamentals/sync-modes.
[126] Andrew Chow. Answer to "Does Bitcoin Core validate signatures by default?", August 2020. URL https://bitcoin.stackexchange.com/a/98718.
[127] Libbitcoin. URL https://libbitcoin.info/.
[128] Bitcoin Knots, . URL https://bitcoinknots.org/.
[129] btcd, November 2023. URL https://github.com/btcsuite/btcd. original-date: 2013-08-06T18:10:52Z.
[130] Bitcoin Network Snapshot - Bitnodes, . URL https://bitnodes.io/nodes/.
[131] Bitnodes, . URL https://bitnodes.io/.
[132] Balaji S. Srinivasan. Quantifying Decentralization, October 2017. URL https://news.earn.com/quantifying-decentralization-e39db233c28e.
[133] Polygon - The Value Layer of the Internet, . URL https://polygon.technology/.
[134] Mina Protocol - Home. URL https://minaprotocol.com/.
[135] Chainflow.io. URL https://nakaflow.io/.
[136] Adem Efe Gencer, Soumya Basu, Ittay Eyal, Robbert van Renesse, and Emin Gün Sirer. Decentralization in Bitcoin and Ethereum Networks, March 2018. URL http://arxiv.org/abs/1801.03998. arXiv:1801.03998 [cs].
[137] Qinwei Lin, Chao Li, Xifeng Zhao, and Xianhai Chen. Measuring Decentralization in Bitcoin and Ethereum using Multiple Metrics and Granularities, February 2021. URL http://arxiv.org/abs/2101.10699. arXiv:2101.10699 [cs].
[138] Juan A. Garay, Aggelos Kiayias, Nikos Leonardos, and Giorgos Panagiotakos. Bootstrapping the Blockchain, with Applications to Consensus and Fast PKI Setup. In Michel Abdalla and Ricardo Dahab, editors, *Public-Key Cryptography – PKC 2018*, Lecture Notes in Computer Science, pages 465–495, Cham, 2018. Springer International Publishing. ISBN 978-3-319-76581-5. doi: 10.1007/978-3-319-76581-5_16.
[139] Ozan Isler. Rational Actors. In Todd K. Shackelford and Viviana A. Weekes-Shackelford, editors, *Encyclopedia of Evolutionary Psychological Science*, pages 1–2. Springer International Publishing, Cham, 2020.

ISBN 978-3-319-16999-6. doi: 10.1007/978-3-319-16999-6_2155-1. URL https://doi.org/10.1007/978-3-319-16999-6_2155-1.

[140] Seungjin Baek, Hocheol Nam, Yongwoo Oh, Muoi Tran, and Min Suk Kang. On the Claims of Weak Block Synchronization in Bitcoin. URL https://eprint.iacr.org/undefined/undefined.

[141] Christian Decker and Roger Wattenhofer. Information Propagation in the Bitcoin Network.

[142] Lukas Aumayr, Pedro Moreno-Sanchez, Aniket Kate, and Matteo Maffei. Blitz: Secure {Multi-Hop} Payments Without {Two-Phase} Commits. pages 4043–4060, 2021. ISBN 978-1-939133-24-3. URL https://www.usenix.org/conference/usenixsecurity21/presentation/aumayr.

[143] Bitcoin Optech. Hash Time Locked Contract (HTLC). URL https://bitcoinops.org/en/topics/htlc/.

[144] Timelock - Bitcoin Wiki. URL https://en.bitcoin.it/wiki/Timelock.

[145] Thaddeus Dryja. Unlinkable Outsourced Channel Monitoring.

[146] Arash Mirzaei, Amin Sakzad, Jiangshan Yu, and Ron Steinfeld. FPPW: A Fair and Privacy Preserving Watchtower For Bitcoin, 2021. URL https://eprint.iacr.org/2021/117. Publication info: Published elsewhere. Major revision. Financial Cryptography and Data Security 2021.

[147] Sergi Delgado. sr-gi/bolt13, October 2023. URL https://github.com/sr-gi/bolt13. original-date: 2020-03-04T15:00:09Z.

[148] bolts/08-transport.md at master · lightning/bolts. URL https://github.com/lightning/bolts/blob/master/08-transport.md.

[149] Dan Boneh, Ben Lynn, and Hovav Shacham. Short Signatures from the Weil Pairing. In Colin Boyd, editor, *Advances in Cryptology — ASIACRYPT 2001*, Lecture Notes in Computer Science, pages 514–532, Berlin, Heidelberg, 2001. Springer. ISBN 978-3-540-45682-7. doi: 10.1007/3-540-45682-1_30.

[150] Adam Back, Matt Corallo, Luke Dashjr, Mark Friedenbach, Gregory Maxwell, Andrew Miller, Andrew Poelstra, Jorge Timón, and Pieter Wuille. Enabling Blockchain Innovations with Pegged Sidechains. 2014.

[151] Quantstamp: The Leader in Web3 Security. URL https://quantstamp.com.

[152] OpenZeppelin. URL https://www.openzeppelin.com.

[153] Veridise Home. URL https://veridise.com.

[154] Patrick McCorry, Chris Buckland, Bennet Yee, and Dawn Song. SoK: Validating Bridges as a Scaling Solution for Blockchains, 2021. URL https://eprint.iacr.org/2021/1589. Publication info: Preprint. MINOR revision.

[155] Why is the Optimistic Rollup challenge period 7 days? URL https://kelvinfichter.com/pages/thoughts/challenge-periods/.

[156] Lorenzo Grassi, Dmitry Khovratovich, Christian Rechberger, Arnab Roy, and Markus Schofnegger. Poseidon: A New Hash Function for Zero-Knowledge Proof Systems, 2019. URL https://eprint.iacr.org/2019/458. Publication info: Published elsewhere. USENIX Security '21.

[157] Polygon zkEVM | Scaling for the Ethereum Virtual Machine, . URL https://polygon.technology/polygon-zkevm.

[158] zkevm-rom, September 2023. URL https://github.com/0xPolygonHermez/zkevm-rom. original-date: 2021-09-03T15:29:35Z.

[159] https://matter labs.io. zkSync — Accelerating the mass adoption of crypto for personal sovereignty., . URL https://zksync.io.

[160] https://matter labs.io. Compiler zkSync Era, . URL https://era.zksync.io/docs/tools/compiler-toolchain/overview.html.

[161] Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. Constant-Size Commitments to Polynomials and Their Applications. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, and Masayuki Abe, editors, *Advances in Cryptology - ASIACRYPT 2010*, volume 6477, pages 177–194. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. ISBN 978-3-642-17372-1 978-3-642-17373-8. doi: 10.1007/978-3-642-17373-8_11. URL http://link.springer.com/10.1007/978-3-642-17373-8_11. Series Title: Lecture Notes in Computer Science.

[162] Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. Interactive Oracle Proofs, 2016. URL https://eprint.iacr.org/2016/116. Publication info: Preprint. MINOR revision.

[163] Stefanos Chaliasos, Jens Ernstberger, David Theodore, David Wong, Mohammad Jahanara, and Benjamin Livshits. SoK: What don't we know? Understanding Security Vulnerabilities in SNARKs.

[164] slides/2023_ethbarcelona_barcelona/kzg_example_by_hand.pdf at main · 0xAdriaTorralba/slides. URL https://github.com/0xAdriaTorralba/slides/blob/main/2023_ETHBarcelona_Barcelona/kzg_example_by_hand.pdf.

[165] Adrià Torralba-Agell and Cristina Pérez-Solà. A Comparison of Layer 2 Techniques for Scaling Blockchains. volume 265, Santander, España, 2022. Ed. Universidad de Cantábria. ISBN 978-84-19024-14-5. doi: https://doi.org/10.22429/Euc2022.028.

[166] Christof Ferreira Torres, Albin Mamuti, Ben Weintraub, Cristina Nita-Rotaru, and Shweta Shinde. Rolling in the Shadows: Analyzing the Extraction of MEV Across Layer-2 Rollups. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 2591–2605, Salt Lake City UT USA, December 2024. ACM. ISBN 979-8-4007-0636-3. doi: 10.1145/3658644.3690259. URL https://dl.acm.org/doi/10.1145/3658644.3690259.

[167] Ethereum Classic Hit by Third 51% Attack in a Month, . URL https://www.coindesk.com/markets/2020/08/29/ethereum-classic-hit-by-third-51-attack-in-a-month.

[168] Shashank. Ronin Bridge Hack Analysis, August 2024. URL https://blog.solidityscan.com/ronin-bridge-hack-analysis-d8f64b8fe683.

[169] ChainLight. Uncovering a ZK-EVM Soundness Bug in zkSync Era, November 2024. URL https://blog.chainlight.io/uncovering-a-zk-evm-soundness-bug-in-zksync-era-f3bc1b2a66d8.

[170] StarkEx Validium ransom attack - HackMD, . URL https://notes.ethereum.org/DD7GyItYQ02d0ax_X-UbWg?view.

[171] Zcash Counterfeiting Vulnerability Successfully Remediated, February 2019. URL https://electriccoin.co/blog/zcash-counterfeiting-vulnerability-successfully-remediated/.

[172] Cristina Pérez-Solà, Alejandro Ranchal-Pedrosa, Jordi Herrera-Joancomartí, Guillermo Navarro-Arribas, and Joaquin Garcia-Alfaro. LockDown: Balance Availability Attack Against Lightning Network Channels. In Joseph Bonneau and Nadia Heninger, editors, *Financial Cryptography and Data Security*, pages 245–263, Cham, 2020. Springer International Publishing. ISBN 978-3-030-51280-4. doi: 10.1007/978-3-030-51280-4_14.

[173] Zhiyuan Sun, Zihao Li, Xinghao Peng, Xiapu Luo, Muhui Jiang, Hao Zhou, and Yinqian Zhang. DoubleUp Roll: Double-spending in Arbitrum by Rolling It Back. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, CCS '24, pages 2577–2590, New York, NY, USA, December 2024. Association for Computing Machinery. ISBN 979-8-4007-0636-3. doi: 10.1145/3658644.3690256. URL https://dl.acm.org/doi/10.1145/3658644.3690256.

[174] Konstantinos Papageorgiou, Alexandros Fakis, Georgios Spathoulas, and Athanasios Kakarountas. An Overview of Security Issues for Blockchain Based Distributed Applications. In Nikolaos Pitropakis and Sokratis Katsikas, editors, *Security and Privacy in Smart Environments*, pages 187–203. Springer Nature Switzerland, Cham, 2025. ISBN 978-3-031-66708-4. doi: 10.1007/978-3-031-66708-4_9. URL https://doi.org/10.1007/978-3-031-66708-4_9.

[175] L2BEAT - Activity Summary, . URL https://l2beat.com/scaling/activity?tab=rollups.

[176] L2BEAT - Activity Validium, . URL https://l2beat.com/scaling/activity?tab=validiumsAndOptimiums.

[177] Jeb Bearer, Benedikt Bünz, Philippe Camacho, Binyi Chen, Ellie Davidson, Ben Fisch, Brendon Fish, Gus Gutoski, Fernando Krell, Chengyu Lin, Sishan Long, Dahlia Malkhi, Kartik Nayak, Keyao Shen, Alex Xiong, and Nathan Yospe. The Espresso Sequencing Network: HotShot Consensus, Tiramisu Data-Availability, and Builder-Exchange.

[178] Stefanos Chaliasos, Itamar Reif, Adrià Torralba-Agell, Jens Ernstberger, Assimakis Kattis, and Benjamin Livshits. Analyzing and Benchmarking ZK-Rollups. In Rainer Böhme and Lucianna Kiffer, editors, *6th Conference on Advances in Financial Technologies (AFT 2024)*, volume 316 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 6:1–6:24, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-345-4. doi: 10.4230/LIPIcs.AFT.2024.6. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.AFT.2024.6. ISSN: 1868-8969.

[179] L2BEAT - The state of the layer two ecosystem, . URL https://l2beat.com/scaling/summary.

[180] Gabriele Picco and Andrea Fortugno. Dynamic Fraud Proof, February 2025. URL http://arxiv.org/abs/2502.10321. arXiv:2502.10321 [cs].

[181] Soonduck Yoo. Comparative analysis of blockchain trilemma. *International journal of advanced smart convergence*, 12(1):41–52, March 2023. doi: 10.7236/IJASC.2023.12.1.41. URL https://doi.org/10.7236/IJASC.2023.12.1.41.

[182] jakub. Lightning Network Explained – Finematics, April 2019. URL https://finematics.com/lightning-network-explained/.

[183] State management - Polygon Knowledge Layer, . URL https://docs.polygon.technology/zkEVM/architecture/protocol/state-management/#trustless-l2-state-management.

Appendix A. Performance metrics used in blockchain

Appendix A.1. Transaction Throughput

The most well-known and common metric for measuring blockchain scalability is transaction throughput (i.e., *Transactions per Second* or *TPS*). All blockchains have a block size (which, in some cases, can be variable) that determines the amount of transactions that can be fitted in a block. Additionally, blockchains also have a *block time* that determines the elapsed time between blocks and, in consequence, how fast new blocks are created and added to the chain. These two characteristics determine the *transaction throughput* as the amount of transactions per second the blockchain is able to confirm.

While this metric is easy to compute, its usefulness is rather low when used to compare the overall performance of a network (even when compared against traditional non-blockchain based systems). The reason behind that downfall is that this metric falls short in capturing the diversity of operations a single transaction may perform.

Appendix A.2. Other performance metrics

Since transaction throughput is not enough to measure the overall performance of a network, other metrics are used in combination to the aforementioned one in order to enrich and complement the study. Let us present some other metrics:

Latency is the time it takes for a transaction to be confirmed. In particular, it measures the time between submitting a transaction to the network and the first confirmation acceptance by the network (assuming that enough fees are paid so the transaction is included on the next created block). For example, on Bitcoin, the latency is 600 seconds, while on Ethereum the latency is between 12 and 15 seconds.

Bootstrap time is the time it takes for a new node to be fully synchronized[24] with the network.

Cost per confirmed transaction in terms of computation, network and storage resources.

Cost to maintain a full node in terms of computation, network and storage resources.

A study of scalability of blockchains using other performance metrics besides TPS, usually yields a better and richer outcome and gives a better approach about the overall infrastructure performance.

Appendix B. Properties of hash functions

Recall that, in Section 4.1.2, secure hash functions are discussed. The NIST defined that the main problems a secure hash function have to overcome are three [96]:

Collision resistance This property asserts that it is computationally unfeasible to discover two distinct inputs to a hash function that yield the same hash value. In other words, finding two different messages, denoted as m_1 and m_2, where $\text{hash}(m_1) = \text{hash}(m_2)$ while $m_1 \neq m_2$, is extremely difficult.

Preimage resistance The preimage resistance property states that given a hash value chosen at random, it is computationally impractical to find an input message that produces that specific hash value. In simpler terms, it is challenging to determine the original message from its hash value.

Second preimage resistance This property implies that it is computationally challenging to find a second input message that generates the same hash value as a specific, known input message. In other words, given a message m_1, it is highly improbable to find another message m_2 such that $\text{hash}(m_1) = \text{hash}(m_2)$.

Appendix C. How Payment Channel Networks work

Figure C.6 illustrates how the bridging and off-chain transactions work for Payment Channel Networks.

1. **Funding Transaction:** each involved party deposit an amount into a 2-of-2 multisig transaction. In the example on the Figure, each party deposit 1 BTC into the Funding Transaction.

2. **Off-chain transactions:** once the Funding Transaction is confirmed on-chain, the parties can transact as many times as desired with the hard constraint that the maximum amount transacted at any time given by any party shall not exceed the sum of the amount deposited on the Funding Transaction (in the example above, 2 BTC).

[24] A new node joining the network is *fully synchronized* with the network if it has downloaded and stored all the blockchain data history prior to the current moment.

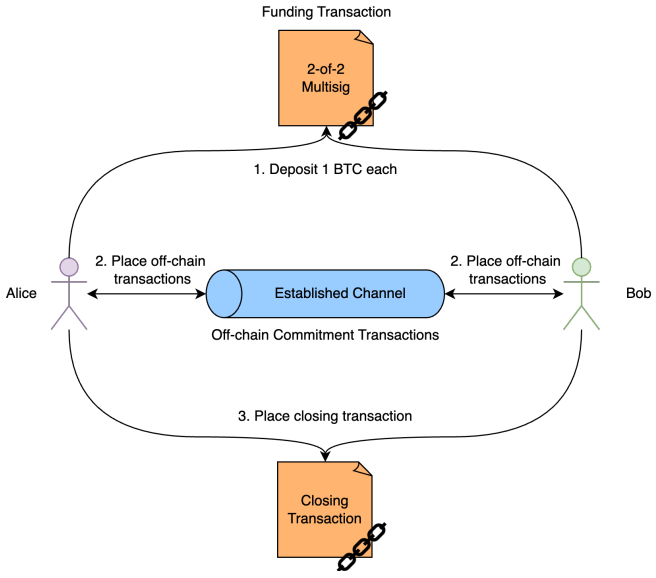

Figure C.6: Step-by-step of the Funding Transaction, Off-chain Transactions, and Closing Transaction of a Payment Channel Network. Diagram inspired on [182].

3. **Closing Transaction:** once the parties are satisfied with the off-chain trades, both parties submit the Closing Transaction onto the L1 with the updated balances. In the non-cooperative case, one party can redeem a transaction to the blockchain that is valid right now, invalidating a possible malicious attack from a devious adversary.

Appendix D. How Zero-Knowledge Rollups and Optimistic Rollups work

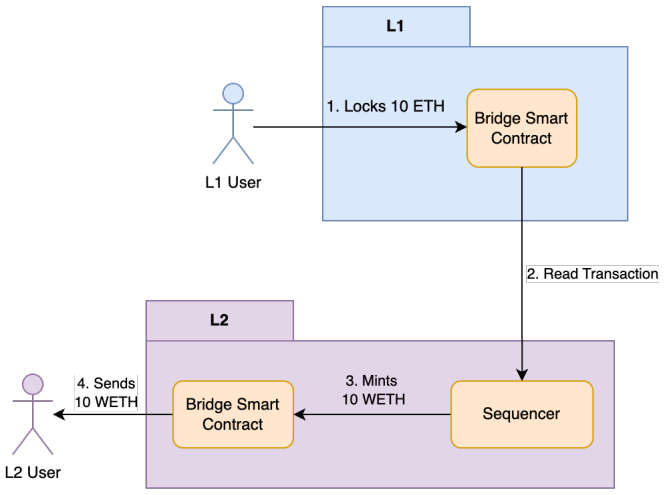

Figure D.7: Step-by-step of an L1 to L2 bridge.

Figure D.7 illustrates the step-by-step that an L1 to L2 bridging transaction follows.

1. **Locks 10 ETH:** the L1 user interacts with the L1 bridge smart contract depositing and locking the desired amount to be bridged.
2. **Sequencer reads transaction:** the Sequencer on L2 reads the pending bridge transactions from L1 and processes them.
3. **Sequencer mints 10 WETH:** the Sequencer mints the corresponding locked amount by invoking the proper function that handles this logic on the bridge smart contract on the L2 side.
4. **Bridge smart contract sends the funds:** once the funds are created, the bridge smart contract sends the funds to the specified L2 User wallet address.

Please note that the L2 to L1 bridging process is equivalent to the explained above, simply replacing "mint" by "burn" and "lock" by "release", as well as, inverting all the arrows on the diagram.

Figure D.8 illustrates the step-by-step that a transaction submitted to a ZK-Rollup follows from the moment that an L2 User submits it, until it is verified on L1 and it is properly updated on L2.

1. **Sends L2 tx:** the L2 User sends the desired transaction to an L2 Sequencer.
2. **Batches txs (and Sends DA to L2 nodes):** the Sequencer orders the transactions from the mempool and bundles them in batches to be sent to both other L2 nodes, and to the L1 bridge smart contract in order to provide with data availability.
3. **Aggregator pulls data:** once data is provided as DA on L1, the Aggregator pulls the data that need to be processed and proved to L1.
4. **The Aggregator provides with the proof:** once the Aggregator has the proof done, it is sent to the prover smart contract to be verified.
5. **New L2 State Root:** if the proof is correctly validated, the prover smart contract yields a new validated L2 State Root.
6. **New L2 State Root Consolidation:** this new validated L2 State Root is shared among all the peers on the L2 network, so the nodes can continue the process from this new state root.

Please note that the workflow for an Optimistic Rollups is slightly different compared to ZK-Rollups, since they rely on fraud proofs. Thus, instead of relying on an Aggregator to provide with validity proofs, the proof and verification are done by providing with a challenge window period (usually 7 days) in which any L2 party watching and processing transactions can submit a fraud proof to invalidate a state transition. In particular, on Figure D.8 we can replace the "Aggregator" entity with a "Watching commitee" and we have the workflow for an Optimistic Rollup.

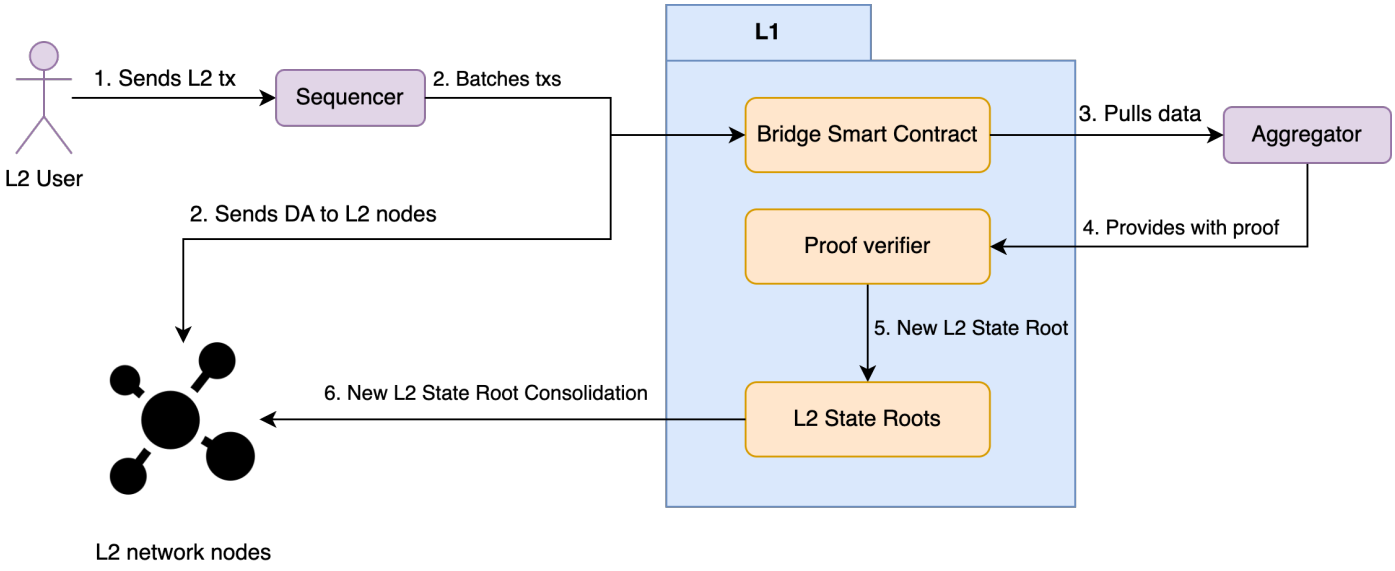

Figure D.8: Step-by-step of a ZK-Rollup transaction workflow. Diagram inspired on [183].