Unmasking the Illusion: The Shortcomings of "Zero-Knowledge" Rollups in Achieving Privacy

Adrià TORRALBA-AGELL
Universitat Oberta de Catalunya
atorralbaag@uoc.edu

Ghazaleh KESHAVARZKALHORI
Universitat Autònoma de Barcelona
ghazaleh.keshavarzkalhori@uab.cat

Cristina PÉREZ-SOLÀ
Universitat Autònoma de Barcelona
cristina.perez@uab.cat

David MEGÍAS
Universitat Oberta de Catalunya
dmegias@uoc.edu

Jordi HERRERA-JOANCOMARTÍ
Universitat Autònoma de Barcelona
jordi.herrera@uab.cat

***Abstract*—The rise of Layer 2 (L2) solutions, including Payment Channel Networks, sidechains, and rollups, has aimed to tackle the scalability challenges of Layer 1 (L1) blockchains like Bitcoin and Ethereum. Among these, Zero-Knowledge Rollups (ZK-Rollups) have emerged as a compelling solution by utilizing ZK-SNARKs to bundle multiple transactions, thereby enhancing throughput and reducing costs. However, despite their technical sophistication, ZK-Rollups do not inherently provide transaction privacy, a common misconception given the "Zero-Knowledge" nomenclature. This paper explores the privacy limitations of ZK-Rollups, emphasizing the need for privacy-preserving features that align with the expectations set by their name. We also review the strategies being developed by various projects to address these limitations. Furthermore, we propose the community begin adapting other names for the technology, such as "Verifiable Rollup" (verRollup), "Incrementally Verifiable Computation Rollup" (ivcRollup), or "Succinct Rollup" (sucRollup) that better represent the current capabilities of rollups. This work contributes to the ongoing discussion on achieving a balance between efficiency, scalability, and privacy in blockchain technologies.**

***Index Terms*—privacy, zero-knowledge rollups, blockchain, scalability, sustainability**

I. Introduction

Permissionless blockchains (a.k.a. Layer 1, L1), such as Bitcoin and Ethereum, have revolutionized the digital landscape by enabling decentralized and trustless payments and applications. However, these systems face significant scalability challenges, which manifest as limitations in the number of transactions per second they can handle, the processing speed of these transactions, and the fees that need to be paid to include them in the blockchain. As blockchain adoption grows, these scalability issues become increasingly problematic, hindering the broader application of these technologies.

To address these scalability problems, various Layer 2 (L2) solutions have been designed and deployed. Among these, Zero-Knowledge Rollups (ZK-Rollups) have emerged as one of the most prominent solutions for scaling the Ethereum blockchain. ZK-Rollups leverage Zero-Knowledge Proofs to bundle multiple transactions together, thereby significantly increasing throughput and reducing costs on the main blockchain. Moreover, because transactions are done off-chain, the speed at which they are processed is also increased.

Despite their technical sophistication and benefits, ZK-Rollups are often misunderstood. The term "ZK" in their name refers to Zero-Knowledge, which in cryptographic contexts often implies privacy. However, ZK-Rollups do not inherently provide privacy for the transactions they process. Instead, their primary advantage lies in enabling computational compression, enhancing scalability and efficiency.

This paper aims to raise awareness about the privacy limitations of ZK-Rollups, a fact well-known within a small circle of experts but often overlooked by the broader L2 user base (and the general Ethereum community). Additionally, we explore potential methods to introduce privacy features to ZK-Rollups, thereby aligning their capabilities more closely with the privacy expectations implied by their name. Consequently, the scope of this work is restricted to Bitcoin-like and EVM-like permissionles blockchains, with a particular focus on ZK-Rollups to address both existing and potential advancements in their functionality.

The remainder of this paper is organized as follows: later in this same Section, (Section I-A), we present why this matter is important in terms of sustainability; on Section II we explain the inner workings of a ZK-Rollups, as well as, an end-to-end example of the workflow a transaction submitted to L2 takes in order to be considered valid on L1. Section III presents what SNARKs and STARKs are; on Section IV we perform an analysis about which are the problems and solutions around privacy on blockchains (both on L1 and L2). Finally, on Section V we draw the conclusions of this article.

A. Sustainability

The concept of sustainability, typically associated with environmental and economic contexts, is increasingly relevant to the digital and cybersecurity realms. As the digital transformation of society accelerates, the need for sustainable technological infrastructures has become increasingly important. This includes not only the physical aspects of the Internet and Information and Communications Technologies (ICT) ecosystems but also the underlying protocols and systems that ensure their security, efficiency, and resilience. In this respect, sustainable cybersecurity is essential for maintaining the integrity and functionality of our increasingly digitized world. Sustainable cybersecurity, as defined by Stifel [1], emphasizes the deliberate and responsible interactions of all stakeholders within the ICT ecosystem. This approach is crucial for preserving the long-term usability and security of

digital infrastructures. In the context of blockchain technology, ZK-Rollups exemplify a move towards sustainability by addressing the scalability challenges that threaten the viability of permissionless blockchains like Ethereum. By enabling higher transaction throughput and reducing costs, ZK-Rollups contribute to a more sustainable blockchain ecosystem. However, the sustainability of ZK-Rollups also encompasses the security and privacy aspects essential for maintaining trust and functionality in blockchain systems. While ZK-Rollups significantly enhance the efficiency of blockchain transactions, they do not inherently provide privacy, a critical component of sustainable cybersecurity. Ensuring that these systems are both scalable and secure requires a wider approach that includes the development of privacy-preserving protocols. Integrating privacy features into ZK-Rollups aligns with the principles of sustainable cybersecurity, promoting a balanced and resilient digital ecosystem. This can be achieved through security-by-design and privacy-by-design approaches, which prioritize the inclusion of robust security and privacy measures from the outset.

The environmental impact of blockchain technologies, particularly the energy consumption associated with proof-of-work consensus mechanisms, has been a subject of significant debate. The ability of ZK-Rollups to handle a larger number of transactions at a lower cost can foster wider adoption and economic viability, particularly for applications with high transaction volumes. By enhancing the economic sustainability of blockchain technology, ZK-Rollups play a crucial role in ensuring its long-term viability and responsible growth. However, the direct impact of ZK-Rollups and other Layer 2 solutions on the overall energy consumption of Proof-of-Work blockchains might be less significant than initially assumed, as the energy expenditure related to mining, which is the primary energy consumer, remains largely unaffected by the number of transactions processed. The true impact of ZK-Rollups on energy consumption lies in their ability to enable greater scalability without a proportional increase in Layer 1 computation, which could indirectly lead to energy savings by potentially delaying the need for more energy-intensive Layer 1 scaling solutions or by accommodating increased transaction volumes with existing infrastructure. Compared to other L2 solutions like Optimistic Rollups, which may require more on-chain computation in case of disputes, ZK-Rollups offer a potentially greener alternative.

II. A BRIEF INTRODUCTION TO ZK-ROLLUPS

Rollups are a Layer 2 technique designed to aggregate multiple transactions into a single batch to be posted on the underlying blockchain along with a proof of their correctness. Rollups can be divided into two main types based on the method of proof generation and validation: Optimistic Rollups [2], which utilize *fraud proofs*, and Zero-Knowledge Rollups, which rely on *validity proofs*.

Zero-Knowledge Rollups [3] (or ZK-Rollups) are backed by *validity proofs*, which typically employ a Succinct Non-interactive ARgument of Knowledge (SNARK) or a Scalable Transparent ARgument of Knowledge (STARK).[1] These proofs allow for the compression of substantial amounts of computation into a small, succinct proof that is efficiently verifiable on Layer 1, offering a faster alternative when compared to re-executing all transactions.

Ultimately, a ZK-Rollup is an on-chain smart contract (or a set of smart contracts) which has two principal functions: (1) to process deposits and withdrawals, and (2) to verify (Zero-Knowledge) proofs that ensure all off-chain processes comply with the established rules. Additionally, and in order to ensure the correct behavior of ZK-Rollups, they need to post the actual transaction data computed off-chain onto Layer 1, providing with what is known as *data availability* [4].[2]

ZK-Rollups rely on validity proofs for its correctness. The Zero-Knowledge framework used when proving, needs from the *plain* transactions involved on the proof to be used as the *witness* when validating the proof.

a) Exploring an end-to-end transaction process for ZK-Rollups: The example presented in this section corresponds to the methodology used by Polygon's zkEVM [5]. Nonetheless, this process is easily generalizable to other ZK-Rollups technologies currently deployed such as zkSync Era [6] or Scroll [7].

Figure 1 shows the process a transaction submitted to L2 has to take in order to be accepted and verified in L1. The process starts when an L2 User sends a transaction to the Sequencer[3] of L2. This –plain– transaction is seen by anybody acting as a Sequencer on L2.[4] The Sequencer executes it and produces the execution trace to be proved by the Executor. This transaction is now ready to be included on an *L2 block* (please, note that this block is *different* from the L1 block where the L2 transaction will be validated).

Thus, following the example from Figure 1, once a transaction is executed and sequenced, it is included in an L2 block (on the Figure, block number `11.895.904`). Blocks may include many transactions and in turn, are bundled in batches (on the Figure, batch number `2.012.146`). Finally, batches are aggregated in Sequences and those are sent to L1 as *data availability*. Please, note that this process happens *before* a proof of correctness is provided for the correct state transition on L2. Finally, each batch is proved and, ultimately, an aggregated proof (a recursive proof [15]) of those batches are sent to Ethereum to be verified.

In particular, the red rectangle on Figure 1 (L1 > Sequence > Input Data, **`batches.transactions`** and its **`data`**)

[1] Both presented in Section III.

[2] Data availability refers to the assurance that the necessary data for verifying a Layer 1 block is truly accessible to all participants in the network.

[3] The Sequencer is the entity that executes and orders the transactions submitted by L2 users to the ZK-Rollup.

[4] Although, currently, there is only one –centralized– Sequencer that is enabled with the ability to order and execute transactions, Polygon's zkEVM team have a road-map to decentralize the sequencing process, by implementing a network with a custom consensus mechanism [8]. Thus, the Sequencers (or, currently, nodes just synchronizing and validating the network) are able to access transactions that they did not receive directly from the submitting user. In particular, any node just synchronizing and validating the network can retrieve batch data (data corresponding to L2 blocks, batches, and sequences) from other nodes in three different ways [9]:

1) From an L2 trusted sequencer before the (unproven) batches are sent to L1.
2) Directly from L1 after the batches are being sequenced (but not proved).
3) After correctness of execution has been proved by the Executor, and verified on-chain though the execution of an smart contract call.

Please note that, in all cases, the transaction data retrieved is simply encoded, not encrypted.

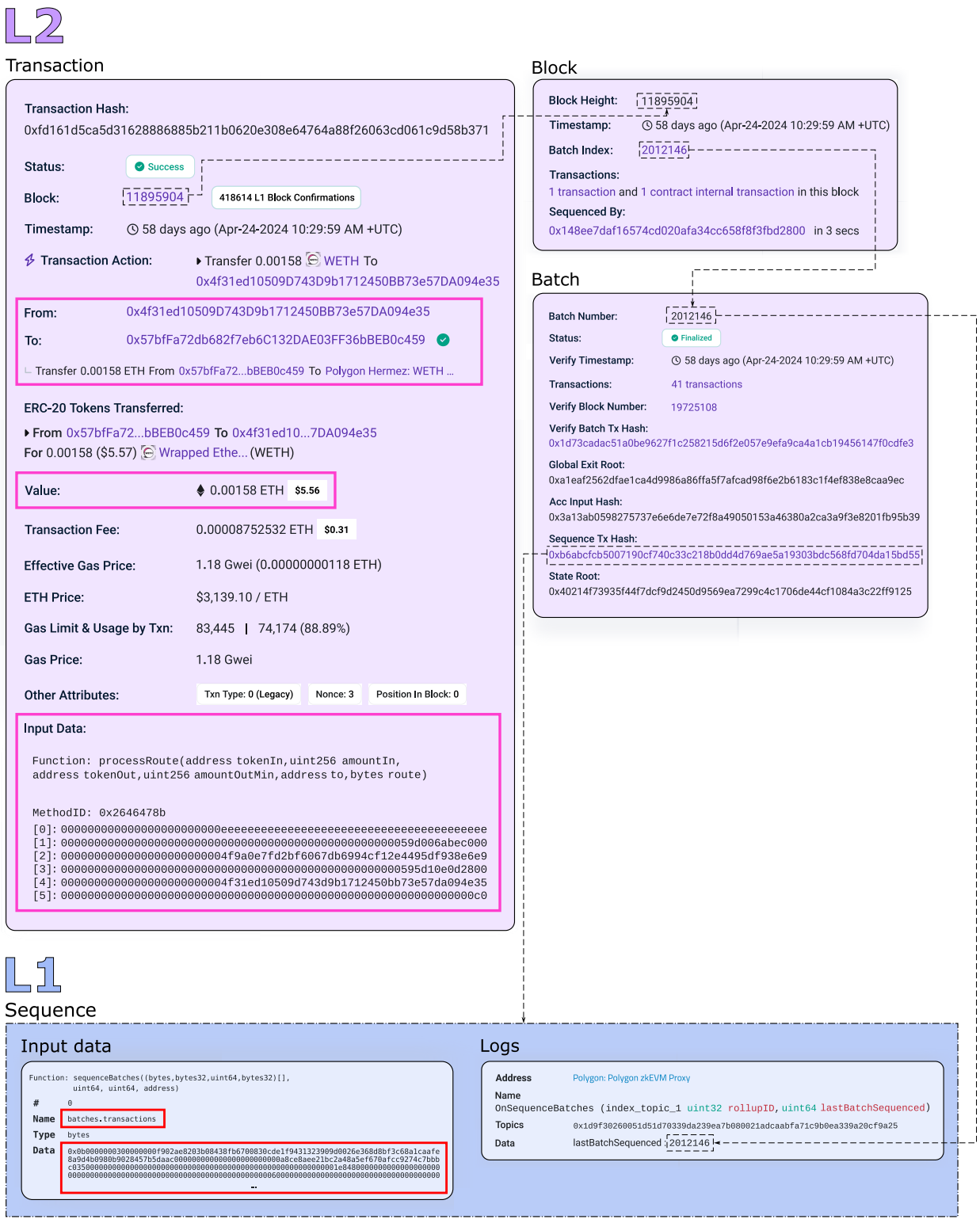

Figure 1. Example of an end-to-end transaction process submitted to Polygon's zkEVM. Adapted from Etherscan's zkEVM Explorer and Etherscan's Ethereum Explorer [10–13]. Inspired on the visualization from Jarrod Watts on X [14]. The `from`, `to`, `value` and `input data` for the L2 transaction are highlighted with a purple rectangle. Highlighted in red, we find **`data availability`** sent to L1 smart contract. The dashed lines correspond to the flow a submitted L2 transaction takes until its data availability is included onto L1.

are the encoded –plain– transactions sent to L1 as *data availability* for that Sequence.

In conclusion, plain access to transactions bundled in a ZK-Rollups is crucial and intrinsic to its correct behavior.

III. A BRIEF INTRODUCTION TO SNARKS AND STARKS

Zero-Knowledge Proofs were introduced by Goldwasser et al. [16] as a method of proving a certain logical statement to another party without giving out any information apart from the known facts of the statement to be proven. An example of such a setting is when a so-called Prover proves knowing

the pre-image of a hash to another entity, named a Verifier, by just providing the proof and the known hash. A widely used category of Zero-Knowledge Proofs is Zero-Knowledge Succinct Non-interactive ARguments of Knowledge (ZK-SNARK) demonstrating features such as:

Succinct refers to the fact that the size of the proof is very small compared to the size the transaction takes. Since SNARK proofs are succinct, they are very fast to verify within a matter of milliseconds. Usually, proof lengths range up to some hundred bytes, even for huge sized statements.

Non-interactive, meaning that only one message is exchanged between the Verifier and the Prover. Using the Fiat-Schamir heuristic [17], an interactive proof system can be transformed into a non-interactive one, speeding up the verification and reducing the rounds of communication.

ARguments imply protection for the Verifier against the Prover with computational limitations[5] in the sense that no Prover can create false proofs that are verified by the scheme. Thus, the use of arguments help to ensure the trust on the proving scheme providing with two key properties:

Soundness, which stands for the fact all false proofs are discarded by the Verifier.

Correctness, which ensures that all correct proofs must pass the verification process completely.

Knowledge is the final element in SNARK and its role can increase the efficiency of the protocol. This states that no Prover can create a proof or argument without the knowledge of a particular witness.

Apart from the described features of SNARKs, for SNARKs to be **Zero-Knowledge** hence categorized as ZK-SNARKs, a very strong assumption is needed securing that no extra knowledge (Zero-Knowledge) is disclosed while proving the statement in question.

As a trusted setup can be a problem hard to handle in many scenarios using ZK-SNARKs, using Zero-Knowledge Scalable Transparent ARguments of Knowledge (ZK-STARK) can also be implemented. STARKs were introduced as scalable Zero-Knowledge Proofs with no trusted setup [18], being a subclass of SNARKs, standing for:

Scalable The proof generation process of a STARK tend to be faster than of a SNARK with roughly the same verification time, resulting in faster proof generation for a program in comparison to SNARKs, yielding scalability.

Transparent STARKs rely on a publicly verifiable randomness using collision resistant hash functions, hence are transparent.

ARgument Same as explained in SNARK.

Knowledge Same as explained in SNARK.

In comparison to SNARKs, STARKs use verifiable public randomness for the generation of the proof, hence being transparent with no toxic waste. Being based on collision-resistant hash functions, STARKs are needless of a trusted setup, but the proof size for a STARK is extensively larger than of a SNARK.

Although SNARKs and STARKs seem to be used just to provide credibility and trust for computations, most of the use cases have traditionally been to provide privacy. We discuss the uses of ZK-SNARKs and ZK-STARKs applied to privacy on Section IV-A.

[5]In particular, we consider polynomial-time Provers.

IV. Privacy on the blockchain

While one of the most well-known features of blockchains is traceability and transparency, there have been efforts to enable privacy when transacting on blockchain. There are a number of techniques suitable for enabling privacy on blockchains [19], such as Zero-Knowledge Proofs [16], Homomorphic Encryption [20][6], Commitment Schemes [21][7] or Ring Signatures [22][8]. It is important to note that, while those are techniques that can be applied to enhance privacy when transacting, not all of them work towards the same goal. For instance, Zero-Knowledge Proofs could be used with the purpose to mask the amount transacted, while Ring Signatures can only help enabling privacy from the sender.

Before digging in-depth on this problem, we consider two types of privacy problems for blockchain transactions:

Privacy on identity. Refering to the fact that it is impossible to guess the identity of any given address (either the sender or the receiver) for any given transaction on a blockchain.

Privacy on the transaction. Referring to the fact that the "content"[9] of a transaction is hidden, and does not provide any other information, besides the fact that a transaction was performed.

A. State-of-the-art about privacy on L1 and L2

Nowadays, privacy for *simple*[10] transactions on L1 is being pursued through various designs and proposals, aiming to guarantee a certain amount of privacy, though many challenges remain to be addressed (e.g. Monero [23] and Zcash [24]). In both cases, those blockchains are taking advantage of Zero-Knowledge frameworks in order to enable with privacy-preserving features such as hiding the amount transacted. For instance, Monero uses Bulletproofs [25]; a short non-interactive Zero-Knowledge Proof that does not require a trusted setup, in order to mask the amount transacted between the sender and the receiver; while Zcash makes use of Halo2 [26], a Zero-Knowledge framework that overcomes the shortcomings regarding the need for a trusted setup from Halo [27], in order to achieve the same goal.

For simple transactions, there are also projects that enable with privacy and scalability in the form of a Layer 2, such as Payy [28], which builds an UTXO-based architecture to

[6]Homomorphic Encryption is a cryptographic method that enables computations to be performed on encrypted data without decrypting it, thereby, preserving data privacy throughout the computational process.

[7]Commitment schemes are cryptographic protocols that allow one party to commit to a chosen value while keeping it hidden, with the ability to reveal the value later, ensuring both binding (the value cannot be changed) and hiding (the value remains secret until revealed).

[8]Ring signatures are a type of digital signature that allows a member of a group to sign a message anonymously on behalf of the group, making it computationally infeasible to determine which group member's key was used to produce the signature.

[9]By "content" we mean here both the "value" and the "input data" in the case of Ethereum.

[10]*Simple* referring to transactions transferring the native currency of the blockchain (e.g. BTC on the Bitcoin network, or Ether on the Ethereum network).

provide with privacy[11] when dealing with native currency transactions, as well as, ERC20 transfers.

While privacy, even in combination with scalability, has seen progress for simple transactions, providing privacy for more general computations (e.g., a Turing-complete calculation) remains an open problem. Albeit there are some solutions available (both for L1 and L2), they are far from definitive. For this reason, this is an area of extensive active research on both Layer 1 and Layer 2.

B. Problem statement: lack of privacy on ZK-Rollups

Whereas there are means for enabling privacy while interacting with the blockchain, our research shows that those techniques are not being used over general-purpose ZK-Rollups, despite this technology including "ZK" (as on Zero-Knowledge) on its name. In the particular case of ZK-Rollups, we have found that, since access to plain L2 transactions is needed and must be granted to any independent party, it is currently impossible to guarantee neither Privacy on Identity, nor Privacy on the Transaction (as aforementioned in this section). In order to overcome this issue, we believe that two different problems must be solved:

a) Problem 1: The first, and main, problem when dealing with privacy on a ZK-Rollup comes from the need to provide with data availability onto Layer 1 in order for it to be secure, reliable, and censorship-resistant. In particular, Figure 1 shows this problem. At the end of the sequencing process, the ZK-Rollup needs to provide with **`data availability`** (the red rectangle at the bottom of the Figure) onto Layer 1. The reason behind this need is to provide an –untrusted– way to reconstruct the State Tree of the computation happening off-chain. With this, an L2 user is able to prove ownership of assets on L2, even in the case where a malfunctionig (or malicious) Sequencer withholds the data. Thus, in order to be able to perform this action, the transactions are encoded, but not encrypted. So, as we have seen in Section II-0a, anybody monitoring the zkEVM smart contract, is able to retrieve this data, decode the information, and have access to the plain transactions executed on L2, defeating both Privacy on Identity and Transaction.

b) Problem 2: The second problem we have found regarding privacy on ZK-Rollups arises from the way information on Layer 2 is shared among their peers. In particular, the first screenshot of Figure 1 (L2 > Transaction) shows the contents of an off-chain transaction placed on the Polygon's zkEVM. More specifically, it displays an ERC20 transfer where information like `from`, `to`, `value` or `input data` is clearly available in plain text. Recall from Section II-0a that the peers on L2 share the executed –plain encoded, but not encrypted– transactions among them.

Hence, we believe that "ZK-Rollup" is somehow a bad name for this technology, since they are not exploiting the "Zero-Knowledge" capabilities of Zero-Knowledge framework used. In fact, ZK-Rollups are exploiting the properties from SNARKs (and/or STARKs), not the properties from a ZK-SNARKs/ZK-STARKs. In particular and, among other properties, they make extensive use of the 'S' from both Zero-Knowledge frameworks: Succinct and Scalable, respectively (as explained on Section III). Therefore, we propose the community to begin adapting other names for the technology, such as: "Verifiable Rollup" (verRollup), "Incrementally Verifiable Computation Rollup" (ivcRollup), or "succinct Rollup" (sucRollup) that better represent the capabilities currently Rollups have.

[11]Currently, it implements a system that works toward achieving privacy on identity, and privacy on the transaction.

C. Enhancing privacy in ZK-Rollups

To overcome this lack of privacy, some projects aim to surmount the lack of "ZK" from ZK-Rollups. For instance, AZTEC (Anonymous Zero-Knowledge Transactions with Efficient Communication) [29] propose a framework that implements a UTXO-based model, which makes use of Non-Interactive Zero-Knowledge (NIZK) [30] (in particular, a Range Proof) in order to prove that a number (in this case, the addition of two elliptic curve points) is within a certain –valid– range. After that, it makes use of homomorphic encryption to perform logical checks over encrypted values.

AZTEC aims to be a Layer 2 (a ZK-Rollup-like scalability solution), however, as of June 2024 and to the best of our knowledge, this technology does not have an mainnet product, not even a testnet launch. Additionally, AZTEC is not EVM-compatible, making it unsuitable for implementing privacy on a transaction level over Ethereum. Another example, aligned with AZTEC's goals is Polygon Miden [31]. Similarly, it is not EVM-compatible since it implements a more general zkVM.

Besides these projects, there are other proposals aiming to solve the two problems proposed earlier in this section. Approaches like exploring the uses of recursion when aggregating proofs with the goal of, not only compress computation, but hide sensitive information in the process, while ensuring correctness.

V. Conclusions

The speed at which blockchain technology is developed has left behind a significant number of new terms and concepts that, without time to be properly defined, are *marketingly* adopted by the community, much to the dismay of those working in this field. For instance, it is well-known today that smart contracts are neither contracts nor smart but simple programs that are executed redundantly to ensure their execution integrity. Similarly, we highlight that, unfortunately, ZK-Rollups do not provide Zero-Knowledge properties. Although its denomination comes from ZK-SNARKs techniques, only the SNARK mechanism is used. We show how current ZK-Rollups implementations store information in plain data to allow for later verification, and Zero-Knowledge promises are vanished in the marketing air. Therefore, while L2 solutions do provide a sustainability path for the blockchain environment in terms of scalability, privacy issues still have a long way to go before being properly addressed.

Acknowledgements

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). We also extend our gratitude to Ignasi Ramos for his assistance with the end-to-end transaction flow for Polygon, and to Muriel Rovira-Esteva for her invaluable insights and help in improving the visualization of Figure 1.

References

[1] M. Stifel, "Securing the modern economy: Transforming cybersecurity through sustainability," 2018. [Online]. Available: https://publicknowledge.org/policy/securing-the-modern-economy-transforming-cybersecurity-through-sustainability/

[2] "Optimistic Rollups." [Online]. Available: https://ethereum.org

[3] "Zero-Knowledge rollups." [Online]. Available: https://ethereum.org

[4] "Data availability." [Online]. Available: https://ethereum.org/en/developers/docs/data-availability/

[5] "Polygon zkEVM | Scaling for the Ethereum Virtual Machine." [Online]. Available: https://polygon.technology/polygon-zkevm

[6] "ZKsync." [Online]. Available: https://zksync.io

[7] "Scroll - Native zkEVM Layer 2 for Ethereum." [Online]. Available: https://scroll.io/

[8] "Proof of Efficiency: A new consensus mechanism for zk-rollups - Layer 2," Feb. 2022, section: Layer 2. [Online]. Available: https://ethresear.ch/t/proof-of-efficiency-a-new-consensus-mechanism-for-zk-rollups/11988

[9] "State management - Polygon Knowledge Layer." [Online]. Available: https://docs.polygon.technology/zkEVM/architecture/protocol/state-management/#trustless-l2-state-management

[10] zkevm.polygonscan.com, "Polygon zkEVM Transaction Hash (Txhash) Details | Polygon zkEVM." [Online]. Available: https://zkevm.polygonscan.com/tx/0xfd161d5ca5d31628886885b211b0620e308e64764a88f26063cd061c9d58b371

[11] "Polygon zkEVM Blocks #11895904 | Polygon zkEVM." [Online]. Available: https://zkevm.polygonscan.com/block/11895904

[12] "Batch #2012146 | Polygon zkEVM." [Online]. Available: https://zkevm.polygonscan.com/batch/2012146

[13] etherscan.io, "Ethereum Transaction Hash (Txhash) Details | Etherscan." [Online]. Available: https://etherscan.io/tx/0xb6abcfcb5007190cf740c33c218b0dd4d769ae5a19303bdc568fd704da15bd55

[14] Jarrod Watts [@jarrodWattsDev], "Following a transaction from L2 -> L1," May 2024. [Online]. Available: https://x.com/jarrodWattsDev/status/1796456174780600635

[15] "Proof generation phase - Polygon Knowledge Layer." [Online]. Available: https://docs.polygon.technology/zkEVM/architecture/zkprover/stark-recursion/proof-generation-phase/

[16] S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof-systems," in *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*, Weizmann Institute of Science and O. Goldreich, Eds. Association for Computing Machinery, Oct. 2019.

[17] 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*. Baltimore, Maryland, United States: ACM Press, 1990, pp. 416–426.

[18] E. Ben-Sasson, I. Bentov, Y. Horesh, and M. Riabzev, "Scalable Zero Knowledge with No Trusted Setup," in *Advances in Cryptology – CRYPTO 2019*, A. Boldyreva and D. Micciancio, Eds. Cham: Springer International Publishing, 2019, pp. 701–732.

[19] A. Satybaldy and M. Nowostawski, "Review of Techniques for Privacy-Preserving Blockchain Systems," in *Proceedings of the 2nd ACM International Symposium on Blockchain and Secure Critical Infrastructure*. Taipei Taiwan: ACM, Oct. 2020, pp. 1–9.

[20] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "ON DATA BANKS AND PRIVACY HOMOMORPHISMS."

[21] G. Brassard, D. Chaum, and C. Crépeau, "Minimum disclosure proofs of knowledge," *Journal of Computer and System Sciences*, vol. 37, no. 2, pp. 156–189, Oct. 1988.

[22] R. L. Rivest, A. Shamir, and Y. Tauman, "How to Leak a Secret," in *Advances in Cryptology — ASIACRYPT 2001*, C. Boyd, Ed. Berlin, Heidelberg: Springer, 2001, pp. 552–565.

[23] "The Monero Project." [Online]. Available: https://www.getmonero.org/index.html

[24] "Zcash: Privacy-protecting digital currency." [Online]. Available: https://z.cash/

[25] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell, "Bulletproofs: Short Proofs for Confidential Transactions and More," in *2018 IEEE Symposium on Security and Privacy (SP)*, May 2018, pp. 315–334.

[26] "halo2.club." [Online]. Available: https://halo2.club/

[27] S. Bowe, J. Grigg, and D. Hopwood, "Recursive Proof Composition without a Trusted Setup," 2019, publication info: Preprint. MINOR revision.

[28] C. Moore and S. Gandhi, "L2 Ethereum ZK Rollup for Private and Compliant Transactions."

[29] "The Privacy-first Layer 2 On Ethereum." [Online]. Available: https://aztec.network/

[30] M. Blum, P. Feldman, and S. Micali, "Non-interactive zero-knowledge and its applications," in *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*, Weizmann Institute of Science and O. Goldreich, Eds. Association for Computing Machinery, Oct. 2019.

[31] "Polygon Miden | A ZK-optimized rollup with client-side proving." [Online]. Available: https://polygon.technology/polygon-miden