A Survey on Zero-Knowledge Proof in Blockchain

Nargess Seifi
nargessseifi@outlook.com

September 12, 2024

Abstract

This Survey explores the role of blockchain technology in the corporate sector, focusing on the challenges of ensuring privacy in transactions. While implementation issues are not major barriers, privacy concerns persist, leading to the investigation of methods such as mix networks, ring signatures, and off-chain protocols. The lecture specifically examines the efficacy of zero-knowledge proof (ZKP) methods in addressing these privacy challenges. It provides a comprehensive survey of blockchain technology, covering structural aspects, consensus algorithms, and future trends, with a spotlight on the significance of data security and privacy. A special focus is placed on the efficacy of non-interactive zero-knowledge proofs (NIZKPs) in addressing privacy challenges, providing a powerful tool for secure and private transactions. By demystifying NIZKPs, the lecture underscores their role in enhancing privacy and efficiency across various applications, particularly within corporate blockchain networks.

Keywords: zero-knowledge, blockchain, cryptocurrency

Contents

1 Introduction

As blockchain technology reshapes corporate landscapes, its promise of efficiency comes with the challenge of safeguarding transactional privacy. Amid security concerns, the intricate issues of identity and transaction privacy emerge as critical aspects demanding attention. This survey navigates the complex terrain of blockchain privacy challenges, spotlighting the potential of zero-knowledge proofs (ZKPs) as a cryptographic tool to address these concerns. Specifically, we explore the role of non-interactive ZKPs in fortifying the privacy foundations of corporate blockchain networks.

2 Security and Privacy of Blockchain

The blockchain is constructed to handle inherent security violation such as the tampering, alterations, unpredictability, distributed denial of service attacks, reduplication and attacks related to double spending. Blockchain requires more supplementary traits for securing data and maintaining its privacy when concerned with the distributed storage, data sharing etc. Security infringement rise due to inability in providing relation between two bodies that are observed from the system which requires confidentiality. Bit coin though ensures anonymity it does not ensure the inability in describing relations and this is termed as the unlinking ability which makes transactions difficult by creating troubles in floating measures against the depersonalization attacks and additionally provoke anyone to access the files freely and access any one transaction and relate it with their transaction and making the user to lose their privacy on transactions.

2.1 Security of blockchain

Security in blockchain can be defined as the protection of transaction information and data in a block (whatever form of data) against internal and peripheral, malevolent and unintentional threats. Typically, this protection involves detection of threat, prevention of threat, appropriate response to threat using security policies, tools and IT services. Some important ideas and principles in security are listed below:

a). Defense in penetration. This is a strategy which uses numerous corrective measures to protect the data. It follows the principle that protecting data in multiple layers is more efficiently as opposed to single security layer.

b). Minimum privilege. In this strategy the access to data is reduced to the lowest level possible to reinforce elevated level of security.

c). Manage vulnerabilities. In this strategy we check for vulnerabilities and manage them by identifying, authenticating, modifying and patching.

d). Manage risks. In this strategy we process the risks in an environment by identifying, assessing and controlling risks.

e). Manage patches. In this strategy we patch the flawed part like code, application, operating system, firmware etc. by acquiring, testing and installing patches.

Blockchain technology uses many techniques to achieve the security of transaction data or block data, irrespective of the usage or data in the block. Many applications such as bitcoin use the encryption technique for data safety. using a combination of public and private key is a way to securely encrypt and decrypt data. The other most secure concept of blockchain is that the longest chain is the authentic one. This eliminates the security risks due to 51and fork problems. As the longest chain is the ultimately authentic, the other attacks become null and void as they end up being orphaned forks.

2.2 Privacy of blockchain

Privacy is the capability of a single person or a group to seclude themselves or data therefore expressing themselves discerningly. Privacy in blockchain means being able to perform transactions without leaking identification information. At the same time, privacy allows a user to remain compliant by discerningly divulging themselves without showcasing their activity to the entire network. The goal of enhancing privacy in blockchains is to make it extremely difficult for other users to copy or use other users' crypto profile.

A typical transaction data structure in blockchain provides fields with addresses of a sender and a receiver, and some data (or payload) that is recorded on blockchain nodes. Most privacy issues are not applicable for public networks, because public blockchain networks such as Bitcoin and Ethereum have anonymous accounts without any need for additional KYC-procedures ("know your customer"). To illustrate privacy ensuring methods for corporate blockchain networks, two types of issues are discussed: identity privacy and transaction privacy.

2.2.1 Identity privacy issues

Identity privacy implies the issues of disclosing addresses of a sender and a receiver. Public blockchain networks with anonymous accounts are free to use diverse pseudo-random addresses for one authentic user to obfuscate the actual sender and receiver [6]. In corporate blockchain networks there are two most common methods to obfuscate sender and receiver credentials in a blockchain: mix networks and ring signatures. Mix networks algorithm includes intermediaries or trusted third parties (TTP) that assembles a group of transactions, obfuscate or mix the addresses of transactions parties and after that sends to target addresses. As a result, the other blockchain participants do not have details to interpret relationships between senders and target receivers (Fig. 1):

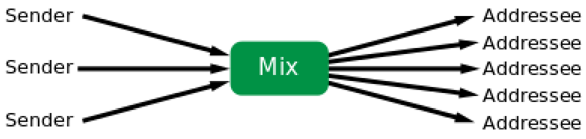


Figure 1: Mix service

Such algorithms are easy to implement and compatible with various blockchain protocols by adjusting TTP or applying cloud storage. However, TTP becomes the most crucial point of vulnerability for all the blockchain nodes. Therefore, mix networks are applicable only in corporate solutions with intermediaries as a central point that satisfies complex requirements for its infrastructure and governance model. To overcome these limitations ring signatures methods are applied. These methods provide the functionality to frame a group of addresses as a ring. Using this ring a sender generates an electronic signature by utilizing addresses of ring participants in sequential order as follows.

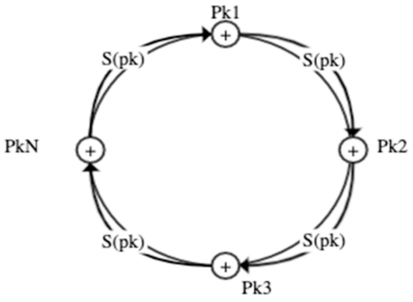


Figure 2: Ring signature

Fig. 2 depicts the public keys of ring participants and the process of computing a group's electronic signature. Although ring signature methods provide a strong mechanism to solve identity privacy issues, it leads to performance limitations due to the complex process of computing electronic signatures. What is more, an attacker might use simple search methods to discover the exact sender if the number of ring participants is relatively small.

2.2.2 Transaction privacy issues

Transaction privacy issues refer to the requirements for confidentiality of data or some content recorded in a transaction. Most corporate blockchain networks set "off-chain" protocols with additional services for private data that is not replicated among blockchain nodes. Thus, an additional database for private data aside from a blockchain node has to be installed (Fig. 3). While blockchain protocol coexists with additional protocol to synchronize private data, one loses the traceability of transactions and data in a blockchain. Also, whereas the programming logic of additional protocol is isolated from algorithms of smart contracts in blockchain, some issues of binding might arise and lead to a complicated debugging process.

While blockchain protocol coexists with additional protocol to synchronize private data, one loses the traceability of transactions and data in a blockchain. Also, whereas the programming logic of additional protocol is isolated from algorithms of smart contracts in blockchain, some issues of binding might arise and lead to a complicated debugging process.

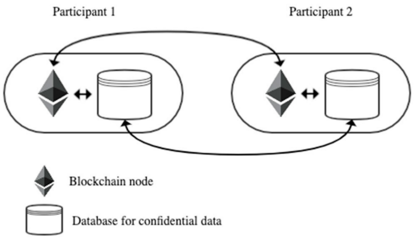


Figure 3: Off-chain protocols

3 Zero-knowledge proofs

The theoretical concept of ZKPs was introduced in the late 80s by Goldwasser, Micali, and Rackoff, 1989. Conceptually, there are many different use cases, but none of them have become economically important. This has changed with the advent of blockchain technology, where ZKP technology has been integrated in some applications.

By using a zero-knowledge proof (ZKP), a party can prove to other parties that a computation was executed correctly. There is no need to replicate the computation—only the proof needs to be verified. Ideally, verifying a ZKP needs significantly less resources than reexecuting the computation. note that the efficiency gains of ZKPs increase linearly in the number of validators. The second and more obvious benefit of ZKP technology is privacy. By using a ZKP, one can prove the correctness of a computation without revealing any additional information except for whether it is indeed correct or not. For example, a blockchain user can prove that he is indeed allowed to make a payment without revealing his identity to the network. Existing applications are the privacy-protecting cryptocurrency Zcash and the Tornado cash protocol on Ethereum. The privacy and confidentiality of data is also important outside of blockchains. Two examples are a person who wants to prove that she voted without revealing her vote, or a company that wants to prove its solvency without revealing its balance sheet.

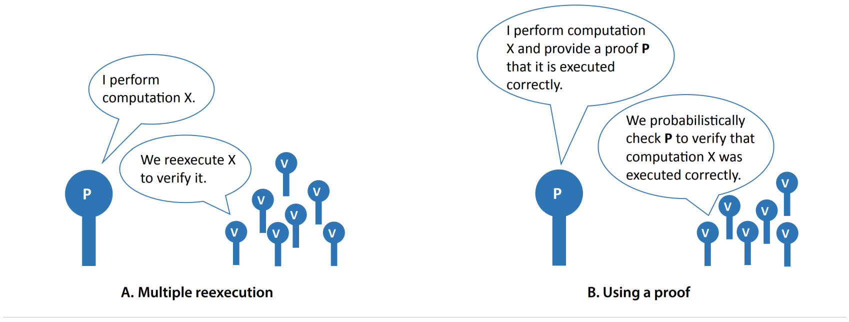

I perform computation X.
We reexecute X to verify it.
P
V
A. Multiple reexecution
I perform computation X and provide a proof P that it is executed correctly.
We probabilistically check P to verify that computation X was executed correctly.
P
V
B. Using a proof

Figure 4: Reexecution vs. Using a Proof

3.1 ZKPs properties

ZKP are characterized by the following properties:

- **Completeness** an efficient method to verify if the statement is correct.
- **Soundness** negligible probability to confirm if the statement is incorrect.
- **Zero-Knowledge** blockchain participants except for the data owner do not have any additional information about private data.

3.2 ZKP principles for blockchain

Zero-knowledge proofs are divided into interactive and non-interactive protocols. The first group implies interactive process when both prover and verifier are online during several rounds of messaging, while non-interactive ones allow a prover to demonstrate knowledge of a certain piece of information without any interaction with a verifier

3.2.1 Interactive ZKPs

Interactive zero-knowledge proofs apply the following algorithm:

1. Prover sends to a verifier send certain message known as a commitment.
2. Verifier responds with the string known as a challenge.
3. Prover sends a message with ZK-prove computed using the commitment (step 1) and the challenge (step 2).
4. Steps 1-3 continue before verifier accepts the commitment.

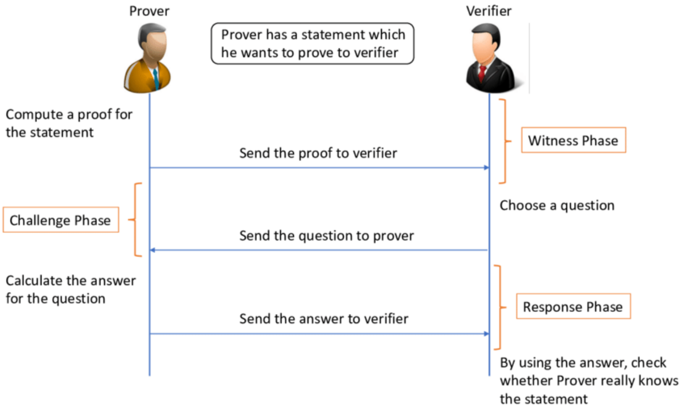

Prover
Verifier
Prover has a statement which he wants to prove to verifier
Compute a proof for the statement
Send the proof to verifier
Witness Phase
Choose a question
Challenge Phase
Send the question to prover
Calculate the answer for the question
Response Phase
Send the answer to verifier
By using the answer, check whether Prover really knows the statement

Figure 5: Zero-knowledge proofs interactive protocols

3.2.2 Non-interactive ZKPs

While the interactive process involves several rounds of communication, it reduces blockchain network performance and also imposes supplementary requirements to be online for both a prover and a verifier during the process. Hence non-interactive zero-knowledge proofs (NIZK) protocols were adjusted for blockchain. The scheme for NIZK protocols is as follows:

1. Prover generates a random number and calculates a number of challenges emulating several rounds of communication.
2. Prover sends generated proof based on a number of challenges emulated on step 1.
3. Verifier check the proof and responds if proof is correct.

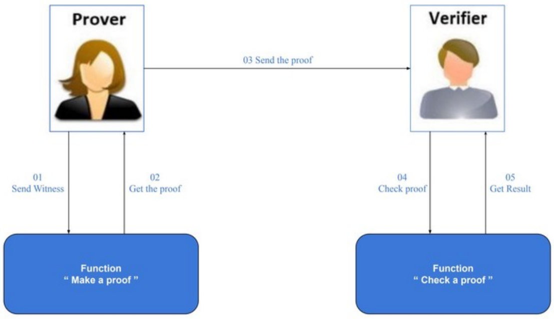


Figure 6: Zero-knowledge proofs non-interactive protocols

NIZK protocols are more capable of applying in blockchain due to the absence of rounds of messaging. NIZK scheme is defined by the algorithms: Setup, Prove, and Verify.

1. Setup (λ) is an initiation algorithm to set parameters that is responsible for security of NIZK scheme C, where input parameter λ defines the security level for blockchain.
2. Prove (x, w) is a function to generate proof applying a number of challenges, input parameter w represents secret information concerning this statement (witness), and parameter x serves to generate a proof π.
3. Verify (x, π) is a function that receives an input parameter π and outputs Boolean value b, which is equal to 1 if verifier accepts the proof and 0 otherwise.

Components of the NIZK scheme are depicted in the Figure 7.

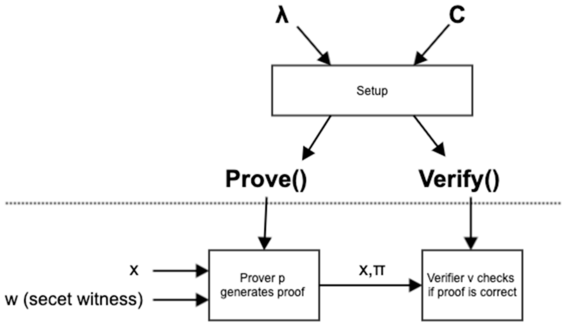


Figure 7: NIZK scheme

The following major criteria to measure the performance of NIZK protocols might be applied:

- Algorithmic complexity to generate proof (or time to generate proof).
- Algorithmic complexity to check if a proof is correct (or time to check proof).
- Communication complexity (proof size in bits).
- Size of transactions (average per transaction).
- Trusted setup (yes/no).

The last criteria define if generated proofs rely on initial parameters (i.e., if initial parameters are compromised, the whole NIZK protocols is also compromised)

3.3 INTUITIVE EXAMPLE: Two Balls and the Colorblind Friend

To get a first intuition of a ZKP, we will discuss a trivial example by Chalkias and Hearn, 2019, which is called "two balls and the colorblind friend." Imagine two friends, Peggy and Victor. Peggy is an expert in ZKPs and wants to teach the concept to Victor. She knows that Victor is colorblind and hands him two identical balls except one is green and the other is red. They agree that the balls are exactly identical apart from their color, which Victor cannot evaluate because of his colorblindness. Peggy (the prover) claims that she can prove to Victor (the verifier) that the balls have indeed different colors without revealing which is which (i.e., the zero-knowledge part). This example is illustrated in Figure 8A.

The proof in this example works like a small game. Victor takes the two balls, puts them behind his back, shuffles them (Figure 8B), and then again shows them to Peggy (Figure 8C). He knows whether he has switched the balls or not since he shuffled them consciously. Furthermore, Peggy knows as well because she can differentiate between the colors. She then tells him that he switched, which is an indication that the balls have to be differently colored.

The proof, however, does not end here. Victor might think that Peggy was just lucky and correctly guessed by chance. The probability of doing so is indeed 50 percent, and therefore he repeats the experiment. An excellent mathematician, he knows that the probability of having two balls with different colors is P(balls have two colors) = $1 - 0.5n$, where n is the number of times they repeat the game.

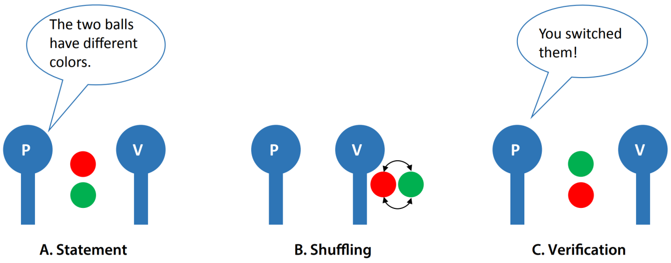


Figure 8: The Two Balls and the Colorblind Friend

Already after n = 10 iterations, the probability that the balls are indeed differently colored is P = 99.9 percent. Thus, Peggy can prove her claim that the balls have different colors without revealing which ball has which color. If she intends to convince other people of the proof, she would need to repeat the same steps for each verifier.

4 Application of ZKPs in blockchain

When it comes to blockchains, there are two main applications of ZKPs: data confidentiality and efficiency. An example for the former use case is privacy-protecting payments, whereas for the latter, it is increased scalability by implementing rollups.

4.1 Data Confidentiality

An example where data confidentiality is applied are privacy-protecting transfers. In standard blockchains, each transaction is registered on the blockchain and is hence visible to everybody. Even though a user remains pseudonymous, it can be possible to derive a real identity using transaction data from and to an address. By using ZKPs, it is possible to prove that someone is allowed to spend a certain amount without revealing the sender and receiver address and amount to the whole network. Zcash, introduced by Ben Sasson et al., 2014, is probably the most famous cryptocurrency that leverages ZKPs for private transactions. To understand how Zcash works, we first recap Bitcoin's (BTC) transaction model. The Bitcoin network uses unspent transaction outputs (UTXOs) to determine which address is allowed to spend a specific amount of the currency, illustrated in Figure 10. Consider an example where Alice receives 5 Bitcoin (BTC) in a transaction from another network participant. The network stores the information that a UTXO worth 5 BTC exists. If Alice wants to send 5 BTC to her friend Bob, she creates a transaction in which she references the UTXO. She also needs to sign the transaction with her private key to prove that she can rightfully spend the UTXO. Next, Alice propagates the transaction to the network, which then includes it in a block. By sending the 5 BTC to Bob, a new UTXO is created that he can use again. In Zcash a similar construct is used called commitment. Say that Alice has received funds via the Zcash blockchain, which comes in the form of a commitment that, contrary to a UTXO, does not reveal the address and the available amount to spend to the network. Conceptually, it is a hash of her address, the spendable amount, and a serial number denoted as s; i.e., Commitment = H(address, amount, s). If Alice creates a transaction to send the funds to Bob, she does the following (Figure 9).

1. She prepares a ZKP to prove that she is allowed to spend the input commitment that corresponds to the commitment's hash.

2. She creates a new output commitment, which is a hash of Bob's address, the amount, and the serial number s.

3. She propagates the commitment together with her ZKP to the network, which then includes it in a block.

4. Simultaneously, she creates a nullifier that is derived from the serial number s. She propagates it to the network, which stores all nullifiers. The nullifier set is a list of hashes that show which input commitments have been spent. If she wanted to double spend the same input commitment again, the same nullifier would result and the transaction would be considered invalid by the network.

5. She privately sends Bob the information that the output commitment relates to his address, the amount included, and the serial number s. Bob needs this information to be able to spend the funds himself.

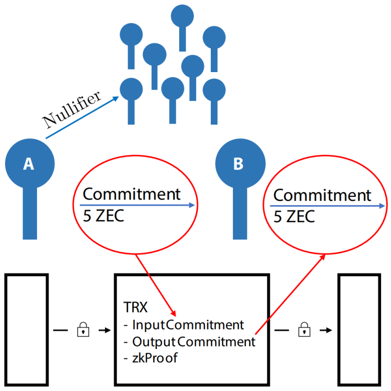


Figure 9: Zcash model

The ZKP proves she is in possession of the private key corresponding to the address and that the amount of the input commitment equals the amount of the output commitment without revealing her address nor the amount.

4.2 Efficiency

A trade-off that many blockchain networks face is the trilemma of security, decentralization, and scalability. Many protocols that emphasize their role of a decentralized network, such as, for example, Bitcoin and Ethereum, face at some point the problem that the demand for block space is much higher than the supply. In other words, the demand for sending transactions exceeds what the network can handle, which in turn makes sending a transaction in these networks pretty expensive. It would be relatively easy to increase scalability by raising the number of transactions that can be included in a block. However, a highly decentralized network relies on the assumption that any regular person who is interested in participating in the network and verifying that the state of the network is correct can do so with a reasonable hardware (e.g., with a regular desktop computer). But if more transactions are sent over the network, the network needs to store more data and the verification of all the transactions becomes more expensive, which in turn requires better hardware and excludes some participants from verifying the state of the network. Thus, blockchain developers are interested in finding solutions that increase scalability without harming decentralization or security. The use of ZKPs promises to remedy exactly that through the use of so called rollups, which are a layer 2 scaling solution.17 In a regular blockchain all transactions are included within blocks of the base chain as illustrated in Figure 9. As discussed earlier, verifying all transactions can be very expensive—a big computational burden is to check all the signatures to ensure that someone is allowed to execute a certain transaction. The idea of a rollup is that a network participant called a sequencer18 batches (or rolls up) transactions off-chain (on layer 2) and creates a proof that proves the validity of these transactions on layer 1. This process is depicted in Figure 8.

Consequently, network participants do not have to verify each individual transaction and signature but only the compressed proof, which is much less expensive. The proofs are based on ZKPs but currently do not incorporate privacy and instead focus on proving the validity of the transactions in a succinct way. Thus in this context, they are often called validity proofs and not ZKPs.19 Basically, what happens is that on layer 1 (on the main chain), a smart contract is created that acts as the proof's on-chain verifier. The sequencer is the prover who creates proofs off-chain on layer 2 and sends a transaction to the on-chain smart contract, which verifies the proof.

So far, we only described the validity of the transactions. However, we also need to record in the blockchain who sent a transfer to whom. This recording is done by publishing the state changes in the call data of the transaction sent to the verifier smart contract. In that way, the current account balances are always updated on-chain. Hence, executing (i.e., checking the validity of the transactions and signatures) is done off-chain, but settling the transactions (i.e., state changes) is on-chain. By using a rollup, much

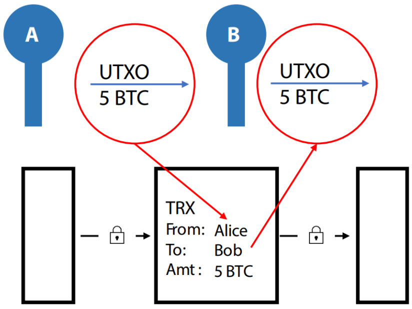


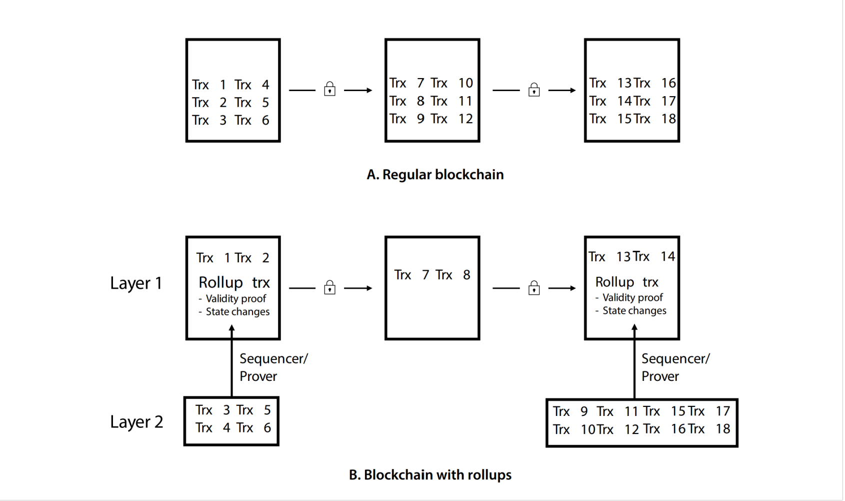
Figure 10: Bitcoin UTXO model

less block space is used on the base chain. Additionally, since the computational effort used to verify the proof scales logarithmically with the size of the input data (i.e., the number of transactions), the fee per transaction when using a rollup decreases in the number of transactions in the rollup and is generally much lower compared to sending a transaction directly on layer 1.

Note that it is still possible to make regular transactions on layer 1, as illustrated in Figure 11B, where some transactions are batched on layer 2 but some are still on the main chain. Moreover, a rollup does not need to send a transaction batch each block. The sequencer could wait until enough transactions are accumulated and only publish a batch every few blocks and in doing so reduce the fee per transaction. This process, however, has the disadvantage of a delayed finality time, and if the sequencer is not decentralized, it comes with some counterparty risk.

On layer 2, developers can build any type of smart contract or application. It is even possible to think of a layer 3 or layer 4 to further increase scalability. In each layer transactions would be batched and sent to the lower layer. However, the more layers that are introduced, the longer the finality time for the upper layers.

There are also other blockchain applications that use ZKPs for efficiency reasons. For example, Filecoin uses them for a decentralized storage application. Instead of storing data with a large company providing a cloud service, Filecoin enables storing data in a decentralized way. To do that in a confidential way, the data to be stored are split up into small chunks and are stored by several nodes across the network. The storage providers earn a reward by storing the data, and they use a ZKP to efficiently prove that they do store them correctly.



A. Regular blockchain

B. Blockchain with rollups

Figure 11: Rollups Illustrated Conceptually

5 Conclusion

In summary, this survey has explored the challenges of security and privacy in blockchain, focusing on the effectiveness of zero-knowledge proofs (ZKPs). Our analysis highlighted the significance of non-interactive ZKPs, offering practical solutions for identity and transaction privacy in corporate blockchain networks.

The survey underscores the value of integrating non-interactive ZKPs to ensure secure and efficient transactions without compromising confidentiality. The presented example demonstrated the real-world impact of implementing ZKPs in blockchain networks.

As blockchain technology evolves, the continued exploration and adoption of cryptographic techniques like non-interactive ZKPs are crucial for addressing dynamic security and privacy concerns. This survey contributes valuable insights to the ongoing conversation about fortifying the foundations of secure transactions within corporate blockchain networks.

Refrences

1. Privacy methods and zero-knowledge poof for corporate blockchain
2. A SURVEY ON SECURITY AND PRIVACY ISSUES OF BLOCKCHAIN TECHNOLOGY
3. An Introduction to Zero-Knowledge Proofs in Blockchains and Economics
4. SECURITY AND PRIVACY ISSUES OF BLOCKCHAIN TECHNOLOGIES