Zero-Knowledge Proof (ZKP): An Analysis

Authors: Alexandre Serras(97505), Diogo Cruz(98595), Gonçalo Leal(98008), Nuno Fahla (97631), Vasco Regal(97636)

Date: 09/05/2022

Index

1 Framework

As the Internet evolved, the number of websites, personal computers and mobile devices increased, and more and more organizations and services began to use the Internet as a base for the future. As a result, much sensitive personal and financial data started to be stored online and sent over the internet. This raised the question of how to protect personal data from unwanted people.
When we have sensitive or personal information or data, we would like for that information to be protected and accessed by only us or entities we trust. This is one of the most important assets organizations and web services must offer their users, protection of their data and to ensure that the person, program, or entity trying to gain access to a piece of information is not an intruder but one of the intended recipients of that information.
In a constant effort to stay ahead of hackers and fraudsters, authentication technology is constantly (and quickly) evolving. The latest becomes outdated extremely quickly. The rapid change of authentication has seen an industry evolve from the creation of basic passwords to upgraded methods like one-time passwords (OTPs) or time-based one-time passwords (TOTPs) and multi-factor authentication.

2 Purpose

The aim of this assignment is to stimulate a reflection and detailed analysis of the Zero-Knowledge Proof Authentication Protocol (ZKP) by thoroughly and clearly explaining the concept and its mathematical models, how it works and how it is used, and the differences between ZK-Snarks and ZK-Starks.
We will also explore the evolution of authentication and authentication protocols and lastly, we will discuss the economic and social impact of the ZKP protocol.

3 Introduction

3.1 Authentication

Authentication is a method of preventing unauthorized users from accessing sensitive information. If not secure or implemented, cybercriminals can gain access to a system and steal it. This identification process verifies a user's identity, usually through something they know (such as a password), something they have (such as an ID card or USB token), or something they are (via biometrics).
By verifying a user's identity, authentication verifies that they have the authorization to access resources and services.

A common example is entering a username and password when logging into a website. By entering the correct credentials, the website knows who you are and that it is actually you accessing the website. This is called password-based authentication.

Although this is just one example, there are many other types of authentications.

- **Multi-factor authentication**, method that requires two (2FA) or more (MFA) independent ways to identify a user. Examples include codes generated from the user's smartphone, captcha tests, fingerprints, voice biometrics or facial recognition.

Figure 1: Authentication

- **Certificate-based authentication**, Method used to identify users, machines, or devices by using digital certificates. A digital certificate is an electronic document based on the idea of a driver's license or a passport.
- **Biometric authentication**, A method that relies on the unique biological characteristics of an individual: facial recognition, fingerprint scanners, speaker recognition, eye scanners, etc.
- **Token-based authentication**, Enable users to enter their credentials once and receive a unique encrypted string in exchange. The token can be used to access protected systems instead of entering their credentials all over again.
- **Single Sign-On (SSO) Authentication**, Allows users to authenticate and access multiple applications without re-entering credentials.

3.2 Authentication Protocol

An authentication protocol is defined as a computer system communication protocol that may be encrypted and is specifically designed for the secure transmission of authenticated data between two parties (e.g., client and server). In simple terms, it is the set interaction and verification rules that endpoints (laptops, desktops, phones, servers, etc.) or systems use to communicate.

For as many different applications that users need access to, there are just as many standards and protocols and selecting the right authentication protocol for an organization is essential for ensuring secure operations and use compatibility.

Here are some examples of the most common authentication protocols.

- **Password Authentication Protocol (PAP)**,Requires a username and password combination to access a given system, which validates the provided credentials
- **Challenge Handshake Authentication Protocol (CHAP)**, Verifies a user to a given network with a higher standard of encryption using a three-way exchange of a "secret."

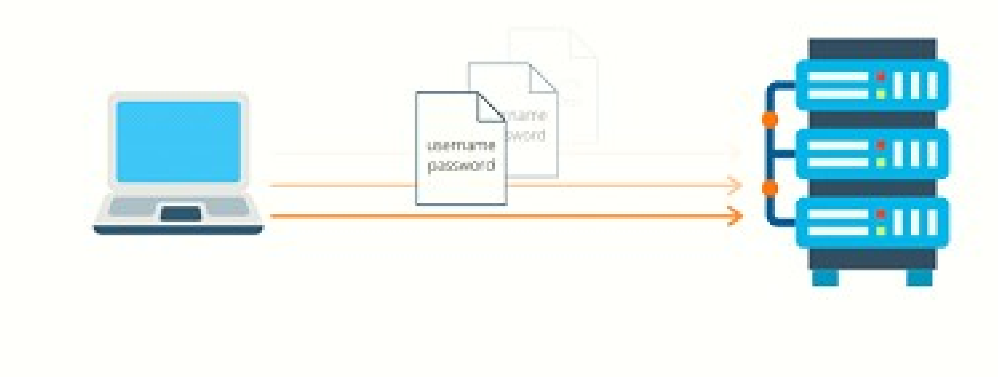


Figure 2: Authentication

- **Extensible Authentication Protocol (EAP)**,Supports many types of authentications, from one-time passwords to smart cards. When used for wireless communications, EAP is the highest level of security as it allows a given access point and remote device to perform mutual authentication with built-in encryption.

We will not go too much into how each of these protocols works, as this is not the main topic of this paper. However, there is one authentication protocol that is attracting a lot of interest and growing enthusiasm in the community, called the Zero-Knowledge Proof Authentication Protocol, which addresses the most common security problems that currently affect digital services.

3.3 Zero-Knowledge Proof (ZPK) Protocol

The changing use of digital services by users in recent years has led to significant vulnerabilities in the security and management of users' private data. Traditional methods of user authentication have revealed many vulnerabilities through which a user's private information can be stolen and exploited, with serious financial and reputational consequences for users and the holders of that data.

Much of the recent data compromise is due to vulnerabilities in integrated third-party systems where the password or validated process was stored or transmitted, highlighting the need for new authentication method.

ZKP delivers the following benefits:

- Zero-Knowledge: if the statement is true, the verifier will not know anything other than that the statement is true. Information about the details of the statement will not be revealed.
- Completeness: If the statement is true, the honest verifier (that is one following the protocol properly) will be able to prove that the statement is true every time.
- Soundness: if the statement is false, it is almost impossible to an astronomically small chance that someone could fake the result to the verifier that the statement is true.

4 CyberSecurity

We chose to include this section in the work before moving on to the issue of zpk since the protocol that we will be addressing primarily at work follows many regulations and security standards, therefore it is vital to offer an introduction to this topic.
With the evolution of technology and the importance they have gained in people's everyday lives, it has become vital to have everything as private and safe as possible, including the issue of data protection and, in recent years, on the internet. various standards and guidelines were formed concerning its operation and that were essential to adhere with.
Therefore, the definition of CyberSecutiry is :"Cyber security is the application of technologies, processes and controls to protect systems, networks, programs, devices and data from cyber attacks" by ITGovernance of UK.

Figure 3: CyberSecurity

With all of this growth, many jobs are dependent on the Internet, and if they are vulnerable and easily attacked services, it can jeopardize the entire company's working day and thus the profits of the same, which is why the CyberSecurity area has been gaining more and more prominence and growth, as well as being necessary ways to have sensitive data stored in the most secret places possible and that only those who are supposed to have access can access that.
It has previously been researched and established that the main source that contributes to security failures is the end user, that is, the human being and at the level of security decisions the more the human person has to take the worse it will be in terms of security, because Because human beings tend to use patterns when choosing a password, for example, associating the password with something that characterizes them or also due to the fact that in order not to have to memorize many passwords, they end up writing them in places on the computer or physical which in itself is even more serious than being something predictable.
Finally, if the protocol that we will describe enters this sphere of security, it will have to follow several pre-established standards and norms, ensuring that the programs and applications that utilize it do not have faults and vulnerabilities resulting from the usage of the zpk and basically, the user must place as little weight as possible on the protocol and how it will unfold, utilizing items that the user has but are encrypted and stored in such a way that decryption is difficult, so that the protocol operates correctly and without problems. security flaws that expose the full finished product

5 What is zpk?

In 1985 MIT researchers Shafi Goldwasser, Silvio Mical and Charles Rackoff first introduced the topic of Zero Knowledge in a paper.

Zero Knowledge Proof (ZKP) is a cryptographic protocol that ensures authentication in a way that no passwords need to be exchanged. This method guarantees that no passwords are stolen, since no password is revealed. Moreover, this protocol is so safe and secure that no one else can find out what the communication is about or what is being exchanged.

This protocol allows someone to prove that he knows a secret (or many secrets) to the other person at the other ”end” of the communication without having to share the secret. Zero Knowledge stems from the fact that no information is revealed during the process, but the other ”end” recognizes that the prover knows the secret.

Zero Knowledge is used when we do not trust the other ”end”, but need to convince them that we know something in order to maintain the communication channel.

This protocol can be illustrated with a simple example:

5.1 Penguin example

This example is based on the same logic as the old game Where's Waldo, where a photo full of penguins contains a puffin. The prover wants to show the Verifier that he knows where the puffin is, but he does not want to reveal the puffin's location. The proof is given with the help of a board with a hole. The prover places the image behind the board with the puffin aligned with the hole, so when the verifier looks through it he sees the puffin, but does not know where it is in the photo.

This example was used to explain Zero knowledge proof to a 10 years girl who understood perfectly the idea behind ZKP.

5.2 Sudoku example

This example is quite more complex. It used the well known puzzle Sudoku and has the objective of proving to someone that a certain algorithm is capable of solving any sudoku level. To do this, the verifier chooses a level and the prover solves it. However, the solution is not shared with the verifier, instead he chooses a random square, row or column and the prover shows him the number in that square. If the numbers do not repeat themselves and are 1, 2, 3, 4, 5, 6, 7, 8 and 9 then we can assume that the solution is corret.

Nonetheless, the algorithm could have had luck and got one level right. In order to create confidence between the two entities, the test is repeated as many times as the verifier thinks necessary. If in all of them the prover shows the correct set of numbers, we can assume that he has the right algorithm.

5.3 ZKP Evolution

As we said before, Zero Knowledge Proof dates back to 1985. However, this protocol evolved through the years. In 2003 appeared the Non-interactive ZKP, which was an evolution of ZKP that required no interaction between the prover and the verifier. In 2012, Nir Bitansky, Ran Canetti, Alessandro Chiesa and Erin Tromer published an evolution of Non-interactive ZKP called Succint Non-Interactive Arguments of Knowledge or by other works zk-SNARKs. This discovery allowed the creation of a cryptocurrency called ZCash in 2014. Another crytocurrency followed the same line and in 2015 Ethereum launched. This cryptocurrency made use of Non-Interactive Zero Knowledge Proofs. In 2017, a more efficient ZKP construction was publish under the name of Bulletproofs. Lastly, in the year of 2018 the Scalable Transparent Arguments of Knowledge (zk-STARKS) was launched.

5.4 ZKP uses

With the appearance of cryptocurrency and the blockchain, this protocol started to be more used. Although it can be used in several different contexts like Authentication Systems, Oversight, Nuclear Disarmament or Blockchains, ZKP grew a lot under the influence of cryptocurrency, specially with Ethereum and his creator.

5.5 Introduction to ZK-Snarks

ZK-SNARK is a type of zero-knowledge technology. ZK-SNARK is an acronym meaning Zero Knowledge Succinct Non-interactive Argument of Knowledge. Succinct because proofs can be quickly verified (small, representative proofs). Non-interactive since there is no communication back and forth between the prover and the verifier (hence, no interactions). Finally, it's an argument of knowledge because the chance of generating a proof without possessing the actual knowledge to support that proof are very slim.

Although ZK-SNARK has real world implementations, this protocol exposes a few flaws that can in theory be exploited. One of them is the need for a set of public parameters, defined in a consensus between prover and verifier. Since these parameters will be used in the protocol to complete verifications, if a third party entity has access to the randomness that generated the parameters, false proofs could be generated and incorrectly verified. In the aforementioned implementations, this exploit is minimized by focusing on a trusted, cryptographically robust, parameter definition.

Another flaw is the fact that this protocol assumes that malicious participants have limited computational power, making it nearly impossible to forge fake proofs. This assumption is not completely accurate, if we consider for example the uprising of quantum computation, which will eventually be a threat to integrity of the ZK-SNARK.

5.6 Introduction to ZK-Starks

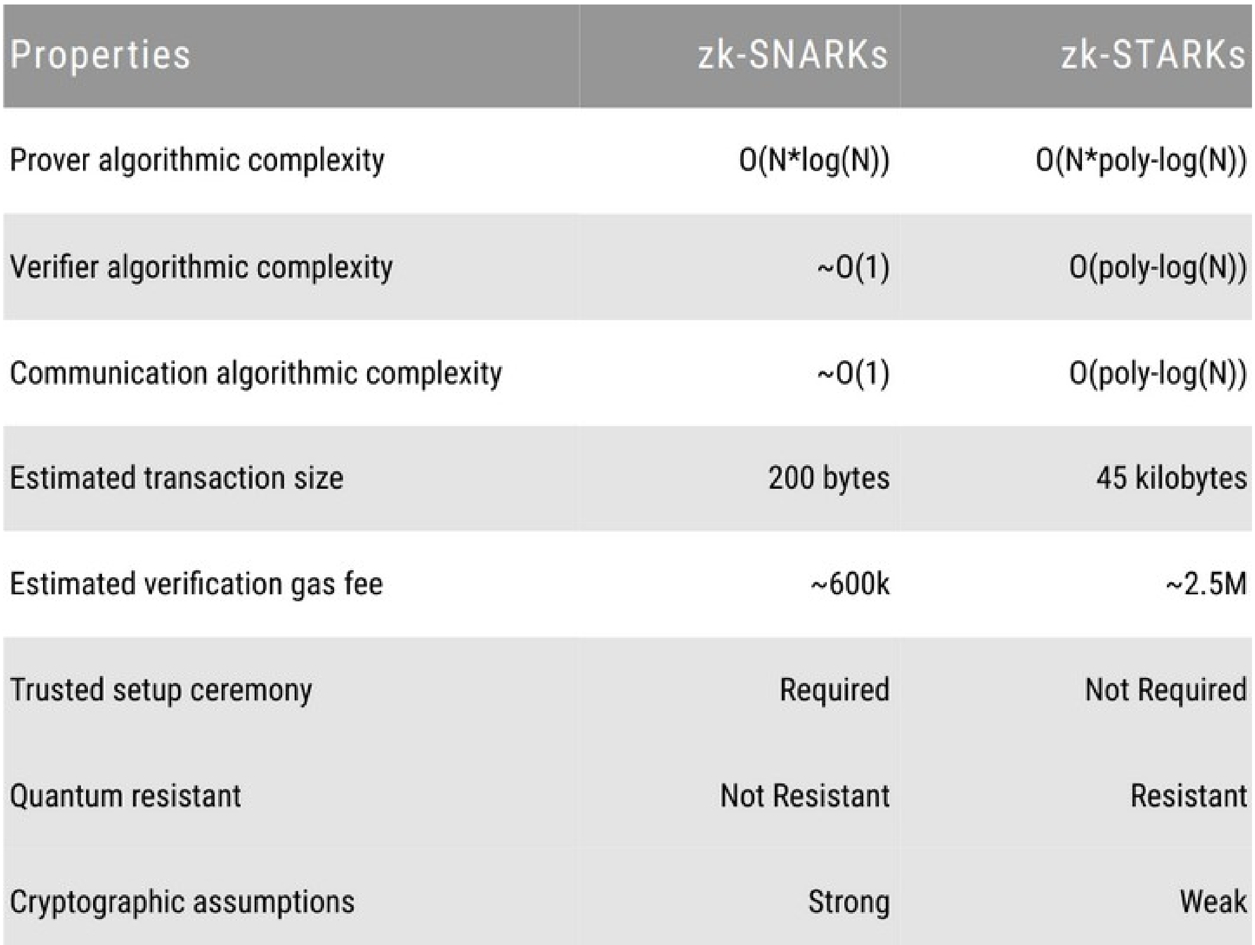
ZK-STARK is another variant of the technology, focused on transparency. By using collision-resistant hash functions, the protocol doesn't require pre-agreed values, presenting an isolated cryptographic layer, which makes proof forging a way harder feat.

ZK-STARK are probably the most state of the art in terms of zero knowledge derived technologies, due to its minimal encryption assumptions. The only observed disadvantages of using this version would be the size of each transaction, which could be a problem on certain contexts.

5.7 ZK-Snarks vs ZK-Starks

From the previous summaries of both technologies, we can quickly infer ZK-STARK is a zero knowledge technology way more robust and cost-efficient than the ZK-SNARK model. The main advantages rely on the cryptography layer: Avoiding a pre-protocol consensus on encryption parameters really boosts the whole privacy concept, also avoiding brute-force computational forgeries.

ZK-SNARK's only advantage over ZK-STARK is the size of the transaction. Since ZK-STARK's transaction's average size is significantly larger, it requires more computational power to verify proofs. In this table we can compare some indicators of both protocols.



Properties	zk-SNARKs	zk-STARKs
Prover algorithmic complexity	O(N*log(N))	O(N*poly-log(N))
Verifier algorithmic complexity	~O(1)	O(poly-log(N))
Communication algorithmic complexity	~O(1)	O(poly-log(N))
Estimated transaction size	200 bytes	45 kilobytes
Estimated verification gas fee	~600k	~2.5M
Trusted setup ceremony	Required	Not Required
Quantum resistant	Not Resistant	Resistant
Cryptographic assumptions	Strong	Weak

Figure 4: Table :ZK-Snarks vs ZK-Starks

5.8 ZK-Snarks Mathematical Models:

ZK-Snarks is based on very well-founded and efficient but also complex mathematical methods to be such an efficient method, which means it was already quite safe and previously scalable.
Assume you choose a word and perform a SHA256, a hash function widely used in cryptography, of the word 100000000 times; this operation will be computationally demanding due to the large number of iterations. So, one of the first requirements is that the computation time be much greater than the verification time; that is, it should not be necessary to take each part of the computation to prove that the computation is correct.
ZK-Snarks is based on very well-founded and efficient but also complex mathematical methods to be such an efficient method, which means it was already quite safe and previously scalable.
For this, we could use random sampling, which means checking only x iterations and assuming that if all are correct, the rest is correct. For example, if we test a chain of 5000 from the 100000000 and they are all correct, we will assume that the rest is correct.
Using the Fiat–Shamir heuristic concept. The Fiat–Shamir heuristic is a technique for generating a digital

signature from an interactive proof of knowledge. In this manner, a fact (for example, knowledge of a specific secret number) can be publicly demonstrated without revealing underlying information, where we essentially compute the Merkel root, Merkle trees are a type of data structure that contains a tree of summarized information about a larger piece of data, such as a file, and is used to verify its contents, and you never know which node to calculate next, that is, only when we have the proof where we are currently going right that we are given the next value to indicate which proof, and this is quite good for preventing attacks, and a note that when there is an error calculating the merkle root, the proof does not stop, it will calculate the branches all at the same time, so an attacker never knows where it is.

However, because it is quite fragile, the computation on the verification side is a little fragile; an attacker only needs to flip a bit for the rest of the chain to be calculated incorrectly and thus break security.

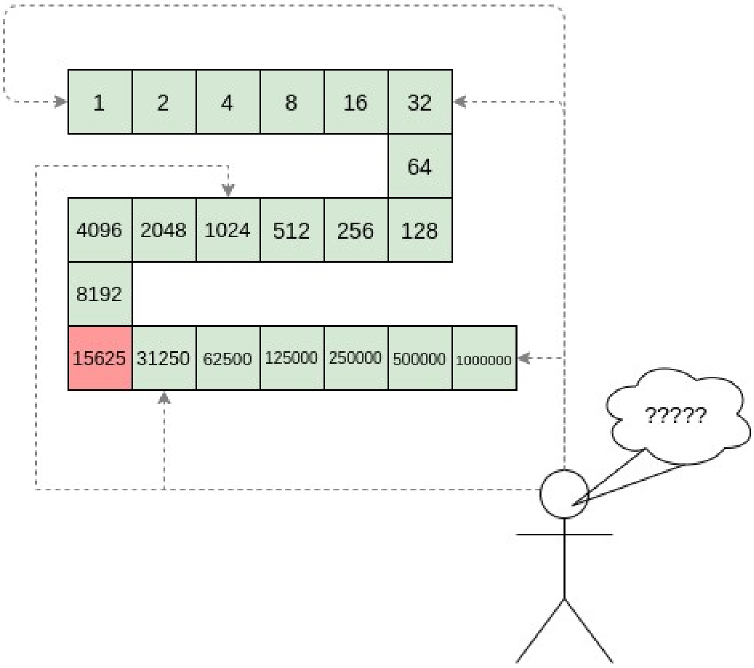


Figure 5: Example of a security fail

As can be seen in the figure above, the string that was being calculated was always multiplied by 2. In practice, this is not done; it only serves to demonstrate the procedure. The attacker was able to change a bit in the red cell, causing the entire rest of the string to be incorrectly recalculated, and so when the user tried to prove who he was, even putting everything exactly right, he was unable to authenticate himself. How can a verifier check each proof part if each part of the computation is not examined individually?

With the goal of achieving A(x) + B(x) = C(x), where each of the expression's components are polynomials of the type A(x) = 2x+1, we want that for any value of x, both sides of the expression are equal. Knowing that P(x)=Z(x)*H(x), where Z(x)=(x-x1)*(x-x2)*...*(x-xn) and H(x) is a polynomial.

Any polynomial equal to zero in some set is a multiple (polynomial) of the simplest (lowest degree) polynomial equal to zero in that same set, as a corollary.

Knowing the corollary from above and that P(x) = Z(x)*Q(x) + R(x) because we know that P(x) is always 0 for the entire set R(x) is also 0 so Q (x) =H(x), and according to the corollary above P(x) = F(x+2)-F(x+1)-F(x) because we know that this is 0, we are according to the expression of the Fibonacci sequence and by tinkering

with the expressions, we find that H(x)=(F(x+2)-F(x+1)-F(x)/Z(x)
So we got it with just one count, which could be equivalent to many iterations, assuming that x =0,1...98. With one count, it was the equivalent of 100 iterations, reducing computational complexity. What was required now was to verify with polynomials rather than going through the process of verifying polynomial coefficient by polynomial coefficient, and thus it is solved with polynomial commitments. If you can verify the equation with polynomials, you are implicitly verifying all of the number equations, which shortens the process.
Understanding that P(x+2)- P(x+1)- P(x) = Z(x) * H (x) The image below explains the theory of how it can be applied.

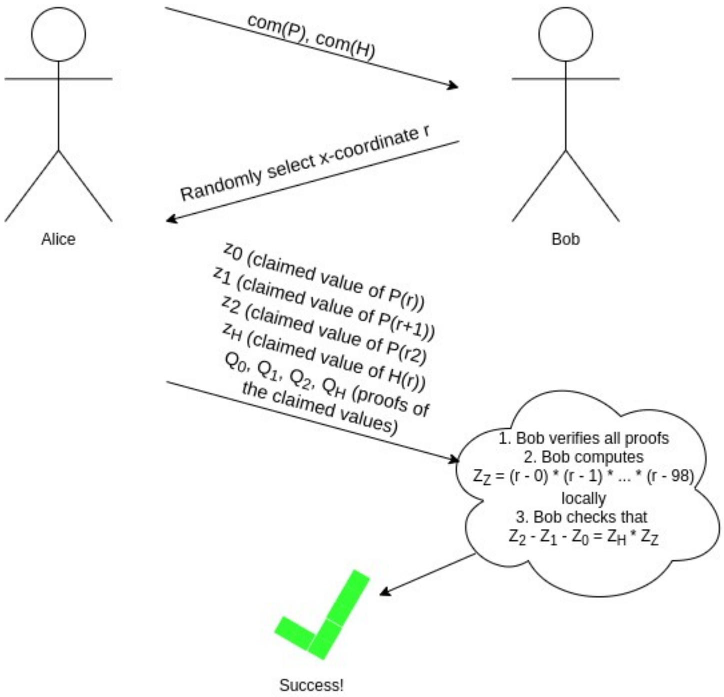

com(P), com(H)
Randomly select x-coordinate r
Alice
Bob
$z0$ (claimed value of P(r))
$z1$ (claimed value of P(r+1))
$z2$ (claimed value of P(r2)
z_H (claimed value of H(r))
Q_0, Q_1, Q_2, Q_H (proofs of the claimed values)
1. Bob verifies all proofs
2. Bob computes
Z_Z = (r - 0) * (r - 1) * ... * (r - 98)
locally
3. Bob checks that
$Z_2 - Z_1 - Z_0 = Z_H * Z_Z$
Success!

Figure 6: Illustrative example of the proposed solution

Analyzing the image above, the ”r” that Bob sends to Alice is basically a cryptographic function between (comp(P),com(H)) so there is no way for Alice to cheat, because they are the ones who you have to say the com(P),com(H) first before knowing the ”r”, then the proofs are generated and Bob checks if everything is going well and if Alice is who she says she is, without ever asking her. Alice tells Bob exactly what she knows.
On the basis of this mathematical theory, the three most common approaches to adapting the zk proof, from what are really polynomial commitments, have emerged: KRI, KATE, and bulletproofs.
Whereas KRI is the most commonly used, as illustrated in the diagram below, in this model we will combine the solution arranged above with the Merkle root mentioned in the first part of this module.
Essentially, this model will also apply to the s polynomials. Assuming that com(P) is a merkle root of a set of P iterations, what we will have to do is a set of iterations to prove that for degree n it also holds the expression P(x)= Q(x2)*x*R(x2), where Q(x) means the even coefficients and R(x) means the odd coefficients. Alice wishing to prove something must send the Merkle root of P, Q, and R; Bob will generate a random number ”r” according to a cryptographic function, and with this ask to calculate the Merkle root of S, where S (x) =Q (x)*r*R and with that of the four Merkle roots we have will select 100 random values of x and see if the two

expressions match for all 100 random values selected. Having said that, we'll do this P (x) = S (x) where we'll only have half a degree until we can directly check the S without a P.

Bob can prove that Alice appears to be who she says she is without providing your password because the accounts will be hitting right and the "r" as generated by the verifier will never be able to be cheated on this method.

Alice essentially does everything. Bob is simply there to force her to demonstrate that she has/knows what she claims to have.

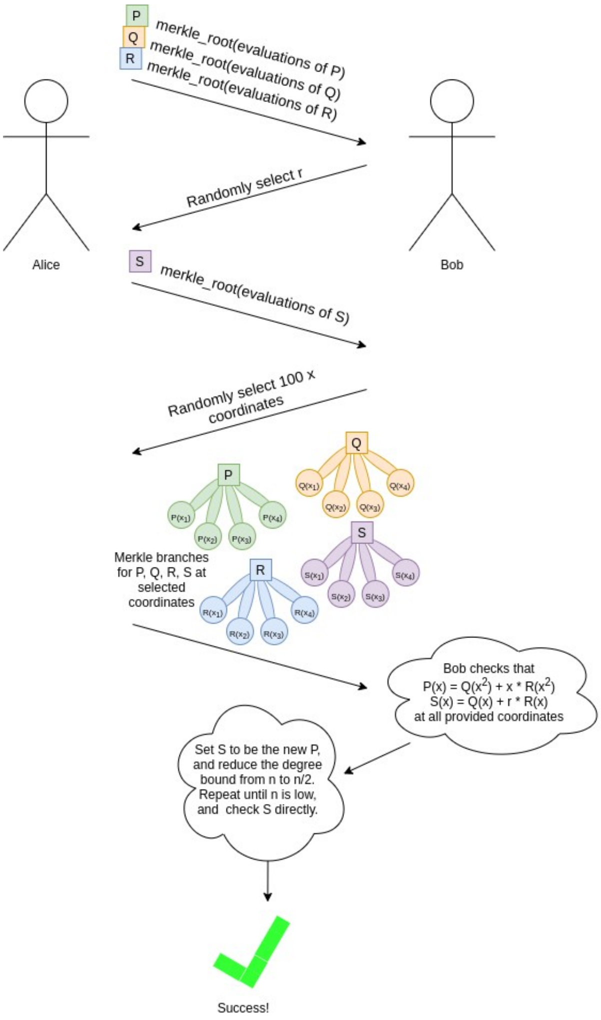

P
Q
R
merkle_root(evaluations of P)
merkle_root(evaluations of Q)
merkle_root(evaluations of R)
Alice
Bob
Randomly select r
S
merkle_root(evaluations of S)
Randomly select 100 x coordinates
P
P(x1)
P(x2)
P(x3)
P(x4)
Q
Q(x1)
Q(x2)
Q(x3)
Q(x4)
S
S(x1)
S(x2)
S(x3)
S(x4)
R
R(x1)
R(x2)
R(x3)
R(x4)
Merkle branches for P, Q, R, S at selected coordinates
Bob checks that P(x) = Q(x²) + x * R(x²) S(x) = Q(x) + r * R(x) at all provided coordinates
Set S to be the new P, and reduce the degree bound from n to n/2. Repeat until n is low, and check S directly.
Success!

Figure 7: Illustrative example of KRI

6 Social Impact

6.1 Electronic vote

When it comes to elections, there are a slew of requirements that must be followed, owing to the fact that the vote is both significant and private. We don't know who someone voted for in an election because we don't

know who they voted for.
That said, electronic voting is something that is not widely accepted socially because it appears to violate the mandatory requirement of a vote, which is that the vote is secret, and with the constant computer attacks that have occurred, the fear that one of these attacks will affect an election and influence the results, or even that the entire process of online voting will go well, it is to be expected.
A ”safe electronic voting system verifiable from end to end using blockchain-based on zero-knowledge” was established to try to overcome this problem, and it will be briefly discussed how it aims to accomplish it.
The zk solution was configured by the privacy issue to define, in addition to necessary data security, and manages to illustrate what is a necessity for individual use. There is no way to sway the outcome of the election. A voter number and a fingerprint that shows the voter is who he says he is are required to register to vote. The data will all be encrypted with this fingerprint, and the system will check the level to see if it has proper data, and if it is the first and only vote that does the operations.
The vote will be encrypted and saved on the blockchain.
According to Wikipedia Cramer-Shoup is , 5 keys are generated through the Cramer-Shoup cryptosystem model, ”The Cramer–Shoup system is an asymmetric key encryption algorithm and was the first efficient scheme proven to be secure against adaptive chosen ciphertext attack using standard cryptographic assumptions.” This is how the user will be able to put their data in the bulletin to prove who they say they are if any of these tests fail.
Because everything is based on zk proof that is validated by the blockchain, it takes a long time for an attacker to modify a blockchain cell, on average 3 years when elections last 10 hours.
Furthermore, because the votes use the Cramer-Shoup Cryptosystem, the votes are guaranteed to be secure in terms of personal privacy.
It should also be highlighted that only administrators will have access to the blockchains where the data is stored; they are private blockchains. but the information stored there will be mostly encrypted.
Even after presenting a proposal for a solution that zero-knowledge presents for this problem, we returned to what was already covered in the introduction to this topic, even having a system that shows very high signs of success when put into practice, most likely was to criticize the credibility of this type of voting because of the population’s mentality and after the elections, people, in general, would question whether those values correspond to real values and if not, whether it was a ”computer error” that manipulated the results because, as a rule, everything in today’s society ends up being a ”computer problem,” and in this case, the same criticism was more likely to occur, even if it was unfounded.

6.2 Nuclear Warhead Disarmament Verification

Since the end of The Cold War there have been political talks and treaties in order to abolish the use of nuclear weapons, thousands of warheads have been dismantled but thousands still remain, in storage or operational. One surprising use for zero-knowledge proof is in the effort of disarming nuclear weapons, since countries can say they have disarmed a number of warheads but in reality are keeping them as a deterrent. To make sure this isn’t the case, zero-knowledge proof comes into play making sure a country proves a warhead is disarmed without revealing any more information of that specific warhead. Traditionally, what is used is an ”unbiased and independent” third party, like a team of engineers, that go into the field and make sure a certain nuclear weapon in truly dismantled, this raises a big problem: is the team truly genuine, independent and trustworthy?

Alexander Glaser, Boaz Barak and Robert Goldston invented a zero-knowdledge proof way to solve this problem, back in 2013. The protocol begins with placing the warhead in a container, for privacy reasons, installing detectors around it to measure neutrons and compare the readings to a template warhead. After that, the prover prepares pairs of detectors, each with a random offset, then, the verifier choses a random pair and takes note of it's offset. The next step is to check all the remaining pairs to see if they contain an identical offset and the verifier selects a random angle. With the preparations complete, the prover places a detector of the first pair at the chosen angle and the other detector of the first pair at the same angle but in the template warhead, not the one being evaluated. The emission of neutrons then begins, if the measurements are close enough between warheads, the nuclear weapon is deemed safe. This process if run as many times as the verifier considers necessary to reach the desired trust level.

6.3 Bank loans

ING, a dutch corporation, has created a system called Zero-Knowledge Range Proof with their blockchain team. It differs from what we have previously explained since it works to prove that a number is within a range of numbers, without every revealing what the number is. This can be applied to bank loans very easily, say someone wants to know if they meet the requirements for a given amount of lent money but, due to personal reasons, don't want anyone to know how much income or possessions they have. This is where ZKRF comes in, given a wide enough range so a number can't be easily guessed, the algorithm would allow/refuse the bank loan to proceed depending on that person's data.

This use for ZKP would make the financial world process client's requests much faster, since many of the regulations would be bypassed, like data-gathering forms and concerns of personal data leaks, making everything faster and cheaper in some cases. This is only one of many cases in which Zero-Knowledge Range Proof would be useful for banking systems, and as one can expect, one of many cases where it can be a problem solver in other systems as well.

6.4 Reduction in the need for computation in blockchain

One big limitation is the need for a lot computational power, so much so that it makes it impossible for the algorithms to be run in slow and mobile devices, due to their complexity. With Z-SNARK we have zkps short and non-interactive proofs that are verified within milliseconds, due to only one message being sent. With this ”succint” version we reach the most cost-efficient way of publishing a zero-knowledge proof to a block chain: between the prover and verifier, a string is generated to be used as a common reference, commonly known as the system's the public parameters, ocurring during the initial setup fase.

This work is being done by Zcash developers, making the zk-SNARKs algorithm more efficient and optimized, with the objective to generate proofs for complex functions, since these are too computationally intensive for a lot of applications.

6.5 Loss of information

Zero-knowledge proof is amazing for making sure another party has the correct information, without ever sharing it. But with that, there is a major drawback: the loss of information. Imagine the following case: three people share a house, each one has copy of a key to open the house but sometimes they leave the key at work. When this happens, since they know and trust each other, Person A opens the door for Person B, although Person A has zero-knowledge of Person B's key's location, they have proof of Person B being a tenant because they have seen Person B open the door before. One day, Person A and Person B leave their keys at work, and when Person C arrives, they reveal their key is also at work, rendering all the contents inside the house lost and impossible to access. Taking this as a metaphor for three servers, all with valuable information, if all of them go down at once, that information is lost, leaving only the proof that they all knew the valuable information. This is a major drawback for zero-knowledge proof, especially for zk-SNARK algorithms.

7 Conclusion

We live in the era of the data, information has an impressive value. We are living a revolution where we will learn new ways of working, managing and working the data. There are a lot of companies specialized in selling and interpreting data, earning tons of money with it.

Inevitably, as more and more companies try to sell our personal information and companies related information, we have understood and accepted that we cannot trust who and what is in the internet and that we must protect ourselves against every possible threat out there. This lead to more investments in security mechanisms to avoid disclosing important and vital information on the web.

Zero Knowledge Proof was created in order to solve these problems. As we discovered throughout this document, ZKP protects our information, by creating trust mechanisms that do not require information exchange.

Also, this document refers some of the possible applications of this protocol. From games to finances, we should prepare ourselves for the future and accept that Zero Knowledge Proofs are going to be one of the most used forms of authentication protocols used in the world.

We may expect major changes in services like banks or sales due to this technology. As more and more of these services are turning their efforts to the blockchain, we can expect that they will secure their transactions with ZKP based protocols. This will also allow them to save computing space, saving money by declaring less transactions in the blockchain.

As Vitalik beleaves and wrote on twitter: ”I expect ZK-SNARKs to be a significant revolution as they permeate the mainstream world over the next 10-20 years.”. So, we surely should keep an eye on this.

8 References

[1] *zk-stark* [Online] https://docs.ethhub.io/ethereum-roadmap/layer-2-scaling/zk-starks/

[2] *what are zk-stark* [Online] https://minaprotocol.com/blog/what-are-zk-snarks

[3] *Video explaining zk-snark* [Online] https://www.youtube.com/watch?v=suF5MtUv3hE&abchannel

[4] *Papper about text* [Online] https://eprint.iacr.org/2018/466.pdf

[5] *E-vote* [Online] https://www.dn.pt/politica/23-anos-de-testes-e-nenhum-resultado-porque-nao-avanca-o-voto-eletronico-13262590.html

[6] *Krammer* [Online] https://en.wikipedia.org/wiki/Cramer%E2%80%93Shoup_cryptosystem

[7] *Snarks* [Online] https://vitalik.ca/general/2021/01/26/snarks.html

[8] *Tau* [Online] https://math.fandom.com/wiki/Tau_(constant)

[9] *SHA2* [Online] https://pt.wikipedia.org/wiki/SHA-2

[10] *Passwords evolution* [Online] https://www.transmitsecurity.com/blog/passwords-and-the-evolution-of-imperfect-authentication

[11] *Authentication* [Online] https://auth0.com/intro-to-iam/what-is-authentication/

[12] *authentication* [Online] https://www.techtarget.com/searchsecurity/definition/authentication/

[13] *authentication* [Online] [10/04/2022]https://hackernoon.com/eli5-zero-knowledge-proof-78a276db9eff

[14] *authentication* [Online] [10/04/2022]https://en.wikipedia.org/wiki/Zero-knowledge_proof/

[15] *authentication* [Online] [10/04/2022]https://www.techtarget.com/searchsecurity/definition/authentication/