Zero Knowledge Proofs and Applications

Report

by

Abhinav Vannala
Kriti Bapnad
Shubham Gupta
Simsarul Vengasseri

Infinity Labs, UST Global
30/08/2019

Abstract

Consider a Prover want to prove a statement to a Verifier. How much extra information is the Verifier going to learn during the course of this proof, beyond the mere fact that the statement is true? This is the essence of zero knowledge proofs. A zero knowledge proof reveals no information beyond the bit of information that the statement is true.

Beyond theoretical interest, this has emerged to have practical importance. How do we design and implement zero knowledge proofs based systems to meet the practical needs? This report presents a study on zero knowledge proofs, elements of design for building a zero knowledge proof based system and its applications.

Contents

1 Introduction

The concept of zero knowledge proofs are first proposed in 1984 by Shafi Goldwasser, Silvio Micali and Charles Rackoff [3]. A zero knowledge proof is a cryptographic method by which a prover can prove to a verifier that there exists some information with the prover without revealing the information. A zero knowledge proof may consist of a trusted setup followed proof generation and verification. If the trusted setup is compromised, the zero knowledge proofs are not valid.

A typical zero knowledge proof consist of a prover key and a verifier key. The prover can generate a proof with the help of prover key. The verifier verifies the proof with the help of verifier key. This entire zero knowledge protocol can be either interactive or non-interactive. An interactive zero knowledge proof will consist of more interactions between prover and verifier inorder for the verifier to get convinced with high probability.

Any zero knowledge proof must satisfy the soundness, completeness and zero knowledgeness criteria. The completeness of an interactive zero knowledge proof increases with the increase in the number of interactions. As the number of interactions increases, the probability of completeness increases exponentially. Interactive zero knowledge proofs are typically less computationally expensive compared to non-interactive zero knowledge proofs.

2 Zero Knowledge Proofs

This section will consist of a study on various interactive and non-interactive zero knowledge proofs.

2.1 Fiege-Fiat-Shamir Method

This is an interactive zero knowledge proof using modular arithmetic. The probability of verification of this zero knowledge proof to fail decreases exponentially with the increase in the number of interactions between the prover and the verifier.

2.2 Zk-SNARKs

Zk-SNARKs refers to zero knowledge succinct non-interactive arguments of knowledge. This works using the principles of elliptical curve pairing. This is a non-interactive zero knowledge proof with an intial trusted setup. [5]

2.3 Bullet Proofs

Bullet proofs are zero knowledge proofs that does not require intial trusted setup. They have short proof size and take less memory space. However, the verification time grows linearly with the size of knowlege. [1]

2.4 STARKS

This comes with an added improvement in terms of scalabilty to bullet proofs. They have larger proof size. This works on the basis of non-collision based hash functions. The verification time grows on logarithmically with the knowledge size and hence scalable. [2]

3 Building applications using Zk-SNARKs

This section consist of ways to build an applications using Zk-SNARKs. We consider a game called MasterMind for applying this concept. In this board game, there are two players: a codebreaker and a codemaster. The codemaster sets a secret combination of coloured pegs (e.g. red, red, green, blue), and the codebreaker has to guess what the combination is. The players follow this sequence,

1. Codebreaker makes a guess (e.g. YRGB)
2. Codemaster gives a clue (3 black, 0 white)
3. Repeat from (1) until the game ends, or the codebreaker runs out of turns.

The clue is made of zero to four black or white pieces; each black piece means that there exists a peg in the guess which exactly matches a peg in the secret of the same colour and in the same position, while each white peg means that there exists a peg in the secret of the same colour but in a different location.

Applied to the Mastermind board game, snarks could thereby prove that a clue about a secret combination of colours is correct, without revealing the secret itself. A commitment at the start of the game about the colours chosen will prevent code master to create a fraudulent proof about a different set of colours at a later stage in the game. We will establish a constraint that the clue of black and white pegs are correct as obtained. This is proven with the help of snarks.

In real life, the physical setup of the gameboard prevents cheating, but this is not easy online, as a remote server or client could fraudulently manipulate the game state. Additionally, to simply hash the solution and reveal it at the end of the game is insufficient, since there is no way for the codebreaker to be sure, mid-game, that the codemaster is not cheating. In fact, mid-game clue verification is a step up above real-world gameplay, where the solution is always hidden from view. [4]

3.1 How does it work?

Codemaster can commit to the colours chosen with the help of Pederson Commitments. Due to the homomorphic properties of this commitment, the codebreaker can generate clues by performing computation on the top of these commitments. These computations does not reveal any information about the secret colours that the codemaster is holding. Codebreaker can use ZK-SNARKs to verify the proof from the codemaster about the commitment is true.

3.1.1 Pederson Commitment

Consider a large prime number p and a group of mod-p integers Z_p. Let g and h be the elements of this group such that the discrete log $log_g(h)$ is unknown. A Pederson commitment to x with randomness r is the group element $C_r(x) = g^x h^r$, and can be de-committed by revealing the values of x and r. Following are the homomorphic properties Perderson commitments enjoy,

$$C_r(x)^a = C_{ar}(ax)$$

$$C_{r1}(x1)C_{r2}(x2) = C_{r1+r2}(x1 + x2)$$

Thus, without knowing the values $x1$ and $x2$ that two commitments hide, any party can compute a commitment to any fixed linear combination of $x1$ and $x2$. Commitment allows one to commit to a choosen value while keeping it secret from others. Commitment schemes are designed so that a party cannot change the value or statement after they have committed to it.

3.1.2 Proof Generation

Proof generation consist of generating a proof string with the help of prover key and the secret. Proof does not contain any information about the secret but conveys that the proover knows a secret.

3.1.3 Proof Verification

Proof verification is done by the verifier with the help of verifier key and proof. This makes the verifier convinced that the prover indeed holds the secret.

4 Applications

In this section, various applications of zero knowledge proofs are studied as follows:

1. Logging into website
2. Core identity
3. Risk Assurance

4.1 Logging into website

Traditional methods use either of the following was to authenticate,

1. Store username and password which is not good.
2. Basic password encryption which is also not good because it can be decrypted vary easily.
3. Hashing the credentials will be safer but still using rainbow table one can decode a hash back to original password.

With the help of zero knowlege proofs we can create secure remote authentication mechanism by providing proof of password. In this case, the password is not shared with the website. Thus the password is never revealed and not stored on the server reducing security concerns. This technology can be applied to enhance security in cryptocurrency wallets. Current authentication systems where you have to login by entering your password, hash your password, and using a server store it in a database. This makes it prone to hacking and privacy breaches. Another method to login without password is to store hash of your password on cloud, this is also prone to security breaches and your data does not truely belong to you.

This method is different from few popular login mechanisms like Microsoft Passport and FIDO. Microsoft passport works on asymmetrical key pair system but one of the keys is inbuilt into the hardware. FIDO is Canadian cellular telephone service which works on passwordless login. When a user registers an account using fingerprint or a secret PIN or some other security device, a public-private key pair is generated, which is unique for that

service and user account. Only the public key is sent to the server, which will be associated with the user's account for future user verification. The private-secret key and other information related to it rests with the user on their local device. In both of these services, the server carry the burden of protecting a hash of your secret key which can be cracked with various attack tools like rainbow tables. Zero Knowledge Proof based login mechanism reduce the burden of security for the server.

4.2 Core identity

Zero knowledge proofs can be used to create scalable identities. Users can create a core identity from which derived identities are created for various services and activities. Derived identities can be verified to have derived from core identity with the help of zero knowledge proofs. This makes the core identity protected even if a derived identity is compromised. [6]

This will be an added layer of security layer where biometric is used directly as identity. Any attacker can target the usage pattern of the users by tracking the biometric. With the help of derived identities, it narrow down the options of an attacker to track the privacy of the user.

4.3 Risk Assurance

Risk Assurance for hedge funds with the help of zero knowledge proofs. This work introduces a new tool for a fund manager to verifiably communicate portfolio risk characteristics to an investor. Fund manager describes the portfolio contents indirectly by specifying the asset quantities with cryptographic commitments. Then, without de-committing the portfolio composition, the manager can use zero knowledge proofs to reveal chosen features to the investor. Hedge funds seeks for high returns using secret strategies. With Zero knowledge proofs the Investor can now verify that the investments are in the range of the risk preference specified. [7]

We consider a universe of asset types A_i and b_i denoting the amount of asset type A_i. Let C_i be the commitment to each of these asset amounts. Once we publish these commitments made of Pederson commitment scheme, the investor can verify that these asset amounts belong the range specified in the contract. The truthfulness of commitment values can be verifed us-

ing snarks. This is practically feasible when the hedging is performed with crypto-currencies or cryto-tokens.

5 Conclusion

Through this report, we have discussed about various zero knowledge proofs, elements of design for a zero knowledge proofs based system, core mathematical properties enabling the working of zero knowledge proofs as well as various applications. Zero knowledge proofs have emerged from a theoretical study to practical implementations.

References

[1] Benedikt et.al. Bulletproofs: Short proofs for confidential transactions and more. https://eprint.iacr.org/2017/1066.pdf.

[2] Eli Ben-Sasson et.al. Scalable, transparent, and post-quantum secure computational integrity. https://eprint.iacr.org/2018/046.pdf.

[3] Shafi Goldwasser et.al. The knowledge complexity of interactive proof-systems. http://groups.csail.mit.edu/cis/pubs/shafi/1985-stoc.pdf, 1985.

[4] Koh Wei Jie. Zero-knowledge proofs, a board game, and leaky abstractions: how i learned zk-snarks from scratch. https://medium.com/@weijiek/how-i-learned-zk-snarks-from-scratch-177a01c5514e.

[5] Christian Reitwiebner. zksnarks in a nutshell. https://chriseth.github.io/notes/articles/zksnarks/zksnarks.pdf.

[6] Thomas Hardjono. Alex "Sandy" Pentland David Shrier. Core identities for future transaction systems, 2016.

[7] Michael Szydlo. Risk assurance for hedge funds using zero knowledge proofs. http://www.eecs.harvard.edu/~cat/papers/szydlo05risk.pdf.