

AFRICAN INSTITUTE FOR MATHEMATICAL SCIENCES
(AIMS RWANDA, KIGALI)

Name: Sittana Osman Afifi Mohamedelmubarak
Zero Knowledge proof

ch 1
Date: May 16, 2020

1 Introduction

[Jan: Title suggestion: Zero-Knowledge Proofs: Implementation of the Graph Isomorphism Protocol]

Recently many companies and banks ask their clients about personal data like their contact information, bank receipt, their tastes, or even their behavior patterns. This information is very valuable; A lot of companies try to collect as much information about their clients as they can, so they'll be able to sell more ads, give better recommendations or keep users longer on their platforms. This is a problem since users can't leave these companies to keep their privacy. Here is where zero-knowledge proof comes in, for example, users can prove to the bank that they have sufficient funds to get loans without telling them exactly how much they have.

[Jan: please pay attention: they'll, can't]

[Jan: "since users can't leave these companies", I don't understand the sentence, try rephrasing]

[Jan: In general: please expand a rephrase. Explain more what the problem with privacy is. Consider expanding the loan example you give and/or more examples.]

1.1 Research Objectives

1. To explore Zero-Knowledge proof protocol and mathematics behind it. [Jan: Be consistent: Is "proof" in ZK proof capitalized or not? Here it isn't, later it is.]
2. To demonstrate Zero-Knowledge proof protocol for graph isomorphism using python.

1.2 Brief History of ZKP

[Jan: This section should come after you explain what ZKPs are.]

Zero-knowledge was introduced in 1989 by Shafi Goldwasser, Silvio Micali, and Charles Rackoff in their paper "The Knowledge Complexity of Interactive Proof-System"[1] The first version of their paper has existed since 1982 and was rejected three times from major conferences (FOCS 83, STOC 84, FOCS 84) before appearing in STOC 85. [Jan: Cite a source for this.] Goldreich, Micali, and Wigderson showed how to construct zero-knowledge systems from any NP-set. [Jan: Cite the paper and expand. Explain briefly what NP is.]

1.3 A very simple definition

Zero-knowledge proof (ZKP) is a way of handling data verification without revealing that piece of data.

Basically, we have two parts: [Jan: Not “parts”, but “parties”, here and below.]

1. **Prover:** the part that knows the secret and wants to prove to the second party that he knows the secret.
2. **Verifier:** the part that wants to be convinced that the prover has the secret.

[Jan: Maybe: wants to learn (verify) if the prover knows the secret or not.]

A Zero-Knowledge Proof must satisfy three properties: [2] [Jan: If this is from a website, put a link in the reference. Also you can cite this, but please also cite a textbook.]

1. **Completeness:** if the statement is true and prover and verifier are both honest (applying the protocol probably) then the verifier will be convinced by the fact.
[Jan: probably? Also watch space before parenthesis.] [Jan: I would say verifier will be convinced by the prover.]
2. **Soundness:** if the statement is false no verifier will be convinced by a cheating prover this statement is true, except with a small probability.
3. **Zero-Knowledge:** the prover does not leak any information more than that he knows the secret. this [Jan: Please watch the details] ensured by necessitating that the verifier can do the protocol alone without knowing any additional information other than from its interaction with the prover.
[Jan: First sentence is unclear, please rephrase.]
[Jan: Clarify: Assuming the statement is true, the verifier can simulate the protocol with honest prover by itself.]

[Jan: Indicate somewhere that interactive proof is an important concept on its own, and that “zero-knowledge” is an additional property of an interactive proof.]

There are main two types of zero-knowledge proof: [3]

1. **Interactive zero-knowledge proof:** This type includes interchanging several messages between the prover and the verifier.
2. **Non-interactive zero-knowledge proof:** In this type, there is no interaction, it contains only a single flow from the prover to the verifier.

[Jan: Indicate that your thesis focuses on interactive ZKPs.]


1.4 Simple example (Two balls and the color-blind friend)

Imagine you have a friend (the verifier) who is color-blind and can't see the difference between a green and a red ball, to him, the balls have the same color and you (the prover) want to prove to him that they are in fact different. He doesn't need to know which is red and which is green, just whether or not they are different. So you give him the two balls and take note of which ball is in which hand. Then, he puts the balls behind his back and chooses to either switch them around or not. After that, he shows them to you again. Now you have to tell him whether or not the balls have switched hands. If they were the same color you would not

be able to differentiate between the two options. So, there would be no way you could guess correctly with a probability of more than 0.5 since the probability that you can guess randomly the right answer is 50%, but this probability will reduce to an acceptable level if we repeat this process as often as necessary.

This proof is zero-knowledge protocol because your friend just learns there are two different balls but he doesn't know which one is red and which one is green.

[Jan: This is a nice example! If it's taken from somewhere, please cite.] 

[Jan: You can expand on this example: 1) If you are verifier, it is easy to simulate (when you ask the question you already know the answer so in simulation you can just give yourself back the correct answer). 2) We can see a difficulty in transforming this example into a mathematical idea: You can't force yourself to be color-blind.] 

1.5 Application of ZKP

[Jan: Citations would be nice. Also please take another pass over the language.] 

1. **Authentication systems:** ZKP helps the user who wants to demonstrate its identity (password for example) via some secret to a second party without learning anything about this secret.
2. **Ethical behavior:** ZKP is used to oblige a user to prove that his behavior is correct according to the protocol, because ZKP saves the privacy of the user's secret during the process of providing the proof.
3. **Nuclear disarmament:** ZKP assists inspectors to confirm whether or not an object is indeed a nuclear weapon without recording, sharing, or revealing the internal workings which might be secret.
4. **Blockchain:** by using ZKP we can perform a valid transaction with keeping the sender, the recipient, and all other transaction details remain hidden.

2 Mathematical Background:

In the previous section, we gave a brief introduction about ZKP, in this chapter, we introduce some concepts in graph theory and mathematical aspect of ZKP, in addition Graph Isomorphism based Zero-Knowledge Proofs.

2.1 Zero-Knowledge Proof

[Jan: In the citation, please credit the lecturer (Goyal) as the first author.] 

Definition 1. (*zero-knowledge proof*):[4] Suppose L is a language. A zero-knowledge protocol is an interaction between two probabilistic polynomial-time algorithms P and V with P trying to convince V that $x \in L$ and satisfying properties (1), (2), and (3) described below:-

1. **Completeness:** If $x \in L$, w is the correct witness, and the protocol is honestly executed, then V outputs *ACCEPT*.

2. **Soundness:** if $x \notin L$, for every probabilistic polynomial time algorithm \hat{P} , there is a negligible function $\text{negl}(\cdot)$ such that:

$$\Pr[\hat{P} \text{ convinces } V \text{ that } x \in L] \leq \text{negl}(\cdot).$$

3. **Zero-knowledge:** This definition represents the idea that P does not leak any information by necessitating that V can perform the protocol alone, and if V can perform the protocol alone, then V does not learn any additional information from its interaction with P . This intuition is formalized by necessitating that there is a standalone simulator S which can produce a transcript which is indistinguishable from that of P and V .

Formally, if for all $x \in L$ there is a probabilistic polynomial time algorithm S which can output a transcript $\hat{\tau}$ such that $\tau \approx_c \hat{\tau}$, where τ is the distribution of the original interaction (between P and V) transcript. The algorithm S is often called a simulator.

[Jan: I'm afraid you have to write this definition in your own words!]



[Jan: Also you need to explain more. What is witness? What is negligible? Transcript? \approx_c ?



I would even suggest explaining what is language, probabilistic polynomial-time, etc.]

References

- [1] S. Goldwasser, S. Micali, and C. Rackoff, “The knowledge complexity of interactive proof systems,” *SIAM Journal on computing*, vol. 18, no. 1, pp. 186–208, 1989.
- [2] H. ANWAR, “What is zkp? a complete guide to zero knowledge proof,” Nov. 2018.
- [3] G. Couteau, *Zero-knowledge proofs for secure computation*. PhD thesis, 2017.
- [4] J. Ackerman, “Introduction to Cryptography, Lecture 19: Zero-Knowledge Proofs I,” 2018. URL: https://www.cs.cmu.edu/~goyal/s18/15503/scribe_notes/lecture21.pdf. Last visited on 2020/05/13.