AFRICAN INSTITUTE FOR MATHEMATICAL SCIENCES

(AIMS RWANDA, KIGALI)

Name: Sittana Osman Afifi Mohamedelmubarak ch 1
Zero-Knowledge proofs: Implementation of the Graph Isomorphism Protocol Date: May 19, 2020

1 Introduction

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 cant leave these companies to keep their privacy; for example, you can't take a loan from the bank without revealing your financial history, also you can't buy a product online without revealing your visa card password, moreover, you can't prove you are in a specific country without leak your physical address.
Here is where zero-knowledge proof comes in, for example, the Dutch bank ING innovative Zero-Knowledge Range concept used to prove their secret within some range, such that they can prove they are over 18 without reviling their actual age or they can prove that they have sufficient funds to get loans by proving it's within some range without telling them exactly how much they have[1].

1.1 Research Objectives

1. To explore Zero-Knowledge proof protocol and mathematics behind it.
2. To demonstrate Zero-Knowledge proof protocol for graph isomorphism using python.

1.2 Thesis structure

In this chapter, we introduce the definition of Zero-Knowledge proof and some mathematical background, in addition, applications of Zero-Knowledge proof, the objective of the thesis. In chapter two we explain graph isomorphism and graph isomorphism based Zero-Knowledge Proof with mathematical proof.
In chapter three we show an implementation of graph isomorphism and graph isomorphism based Zero-Knowledge Proof using python. Finally, in chapter four, we present the conclusions and recommendations of this research.

1.3 A very simple definition

Zero-knowlege proof (ZKP) is a way of handling data verification without revealing that piece of data.
Basically, we have two parties:

1. **Prover**: the party that knows the secret and wants to prove to the second party that he knows the secret.
2. **Verifier**: the party that wants to learn (verify) if the prover know the secret or not.

A Zero-Knowledge proof must satisfy three properties:[2]

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 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 ensured by necessitating that the verifier can do the protocol alone without knowing any additional information other than from its interaction with the prover.

Three interpretations, yielding different notions of zero-knowledge:[3]

1. **Perfect Zero-Knowledge** requires that the two probability distributions be identical.
2. **Statistical Zero-Knowledge** requires that these probability distributions be statistically close (i.e., the variation distancebetween them is negligible).
3. **Computational (or rather general) Zero-Knowledge** requires that these probability distributions be computationally indistinguishable.

Note that Graph isomorphism based on perfect Zero-Knowledge protocol.

Definition 1. *(Interactive proof)[4]*
An interactive proof $< P, V >$ for language L is a two-party protocol in which a computationally unrestricted prover, P, interacts with a probabilistic polynomial-time verifier V, by exchanging messages. Both parties share a common input x. At the end, V either accepts or rejects and both completeness and soundness properties hold.

zero-knowledge is an additional property of an interactive proof.
There are main two types of zero-knowledge proof:[5]

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.

In this thesis we focuse on interactive zero-knowledge proofs.

1.4 Brief History of ZKP

Zero-knowledge was introduced in 1989 by Shafi Goldwasser, Silvio Micali, and Charles Rackoff in their paper "The Knowledge Complexity of Interactive Proof-System"[6] 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.
Goldreich, Micali, and Wigderson showed how to construct zero-knowledge systems from any NP-set.[7]

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

[8]
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 probabilty will reduce to an acceptable level if we repeat this process as often as necessary.
If you are verifier, it is easy to simulate (when you ask the question you already know the answer so in the simulation you can just give yourself back the correct answer).
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.
We can see a difficulty in transforming this example into a mathematical idea: You can't force yourself to be color-blind.

1.6 Application of ZKP [9]

1. **Authentication systems**: ZKP help the user who wants to verify its identity (password for example) via some secret to a second party, but the second party should not learn anything about this secret.

2. **Ethical behavior**: ZKP uses 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 swction, 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.

Definition 2. *(Transcript)*
A transcript is a sequence of messages take an output from P and V and put them in a list.

Definition 3. *(Language)[10]*
A language L is a set of strings over an alphabet,

Definition 4. *(polynomial time algorithm)[11]*
Algorithm A runs in polynomial time if for every string s, $A(s)$ terminates in less than or equal $p(\mid s \mid)$ "steps", where $p(.)$ is some polynomial function.
where $\mid s \mid$ the length of s.

Definition 5. *(Decision problem)[11]*
a decision problem consist of: Problem X is a set of strings, Instance s is one string, Algorithm A solves problem X with:

$$\textit{Yes if } s \in X$$

$$\textit{No if } s \notin X$$

Definition 6. *(NP)[12]*
*A decision problem Q is in **NP** if there is a polynomial time algorithm $V(I, X)$ with the following properties:*

1. *If I is a YES-instance of Q, then there is some X with size polynomial in $\mid I \mid$ so that $V(I, X)$ =YES*
2. *If I is a NO-instance of Q, then for all X, $V(I, X)$ = NO.*

X is usually called a **witness**.

Definition 7. *(Negligible function)[4]*
The function $f : N \longrightarrow R$ is called negligible if for all $c{>}0$ and sufficiently large n, $f(n){<}n^{-c}$. f is called nonnegligible if there exists $a, c{>}0$ such that for all sufficiently large n, $f(n){>}n^{-c}$.

Definition 8. *(Computational Indistinguishability $\approx_c$) [13]*
Let $\{X_n\}_{n\in\mathbb{N}}$ and $\{Y_n\}_{n\in\mathbb{N}}$ be two ensembles (sequences of probability distributions). We say $\{X_n\}_{n\in\mathbb{N}}$ and $\{Y_n\}_{n\in\mathbb{N}}$ are computationally indistinguishable if for any non-uniform probabilistic polynomial time algorithm D, there exists a negligible function ϵ such that for every $n \in \mathbb{N}$,

$$\mid Pr_{t\longleftarrow X_n}[D(t) = 1] - Pr_{t\longleftarrow Y_n}[D(t) = 1] \mid \leq \epsilon(n)$$

2.1 Zero-Knowledge proof

Definition 9. *(zero-knowlege proof):[2] Suppose L is a language. A zero-knowledge protocol is a method by which two probabilistic polynomial-time algorithms P and V communicate with P seeking to convince V that $x \in L$ and satisfying properties (1), (2), and (3) described below:-*

1. ***Completeness***: *If $x \in L$ (the statement is true), w is the correct witness, and both the verifier and the prover follow the prtocol properly,then the verifier would be convinced (V outputs $ACCEPT$).*

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

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

3. ***Zero-knowledge***: *The prover doesn't reveal any information more than he knows the secret, by necessitating that the verifier can execute the protocol alone without knowing additional information than what should know from his interaction with the prover. So we can perform a protocol using a standalone simulator S which produces a transcript that is indistinguishable from the one produces from the interaction between P and V.*
Formally, if for all $x \in L$ there is a probabilistic polynomial time algorithm S which can output a transcript $\acute{\tau}$ such that $\tau \approx_c \acute{\tau}$, where τ is the distribution of the original interaction(between P and V) transcript. The algorithm S is often called a simulator.

References

[1] E. Morais, T. Koens, C. Wijk, and A. Koren, "A survey on zero knowledge range proofs and applications," 07 2019.

[2] V. Goyal and 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.

[3] O. Goldreich, *Foundations of cryptography: a primer*, vol. 1. Now Publishers Inc, 2005.

[4] S. Almuhammadi and C. Neuman, "Security and privacy using one-round zero-knowledge proofs," in *Seventh IEEE International Conference on E-Commerce Technology (CEC'05)*, pp. 435–438, IEEE, 2005.

[5] G. Couteau, *Zero-knowledge proofs for secure computation.* PhD thesis, 2017.

[6] 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.

[7] K. M. . H. Sun, "Great Ideas in Theoretical Computer Science, Lecture 9: Zero-Knowledge Proofs," 2013. URL: `http://resources.mpi-inf.mpg.de/departments/d1/teaching/ss13/gitcs/lecture9.pdf`. Last visited on 2020/05/17.

[8] K. Chalkias and M. Hearn, "Demonstrate how zero-knowledge proofs work without using maths," 02 2019.

[9] J. Hasan, "Overview and applications of zero knowledge proof (zkp)," *International Journal of Computer Science and Network*, vol. 8, pp. 436–440, 2019.

[10] I. Dinur and S. Shalev-Shwartz, “ Advanced Topics in Complexity: PCP Theory, Lecture 1,” 2004. URL: https://www.cse.huji.ac.il/~pcp/lecture-notes/lect01/lect01.pdf. Last visited on 2020/05/17.

[11] K. Wayne, “Lecture Slides for Algorithm Design, INTRACTABILITY II,” 2020. URL: https://www.cs.princeton.edu/~wayne/kleinberg-tardos/pdf/08IntractabilityII.pdf. Last visited on 2020/05/17.

[12] M. Dinitz, “ Introduction to Algorithms / 600.463 Algorithms I,” 2014. URL: https://www.cs.jhu.edu/~mdinitz/classes/IntroAlgorithms/Fall2014/Lectures/lecture21.pdf. Last visited on 2020/05/17.

[13] N. R. Gowravaram, *Zero Knowledge Proofs and Applications to Financial Regulation.* PhD thesis, 2018.