Horizons of Zero-Knowledge

Jakob Povsic

2020

Contents

1 Introduction **2**

2 Interactive proof systems **2**

3 Zero-Knowledge proofs **3**

4 Zero-Knowledge **3**

4.1 Proof system 3

4.2 Definition 4

4.3 Definiton of Zero-Knowledge 4

5 Interactive Zero-Knowlege proofs **4**

5.1 Interactive proof systems 4

5.2 Common NP-Complete problems 4

5.2.1 Quadratic residue 4

5.2.2 Graph isomorphism 4

5.2.3 Discrete logarithm 4

6 Non-Interactive Zero knowledge proofs **4**

6.1 zk-SNARK 4

6.1.1 Shamir-Fiat heuristic 4

6.2 Bulletproofs 4

6.3 zk-STARK 4

Abstract

This paper will focus on the field of Zero-Knowledge proof (ZKP) protocols, their applications and their role in a digitalised society.

The mathematical background of ZKPs by looking at general theorem proofs, properties of interactive proof systems, the definition of Zero-Knowledge and non-interactive ZKPs.

We will look at how specific protocols are implemented using quadratic residues, discrete logarithms and graph isomorphisms as well as how Fiat-Shamir heuristics can be used to make ZKPs non-interactive in protocols like zk-SNARK.

Finally we will provide a prototype implementation of an authentication scheme using a ZKP protocol (tbd which one, hopefully zk-STARK).

1 Introduction

Something about how privacy and security are becoming ever more important as our world is becoming ever more digitalised and connected.

2 Interactive proof systems

An interactive proof system is an efficient theorem proving system, where the verifier is a probabilistic polynomial time machine that exchanges the messages with the prover. The notion of probability makes the proof more relaxed allowing a verifier to be incorrectly convinced of a proof with a very small probability of error n^{-k} for any positive constant k and sufficiently large inputs of size n.

GMR [1] formalised interactive proof systems as an interactive protocol between two interactive Turing machines where the prover with unbounded processing power produces an efficient proof which the computationally bound verifier computes on in polynomial time.

Conditions for an interactive proof system:

- Completeness - For each k and a sufficiently large x in L, a prover accepts with probability at least $1 - |x|^{-k}$
- Soundness - For each k and a sufficiently large x not in L, a prover accepts with probability at most $|x|^{-k}$

Generally we can say that for any language L in NP there exists a proof system in which a verifier can in polynomial time verify whether a word x is in a language L.

3 Zero-Knowledge proofs

These ideas were originally defined in a paper published in 1985 by Goldwasser, Micali and Rackoff [1].

In an interactive proof system proper formalisations exist to express a notion of ”knowledge” transmitted from the prover to the verifier. A ”Zero-Knowledge proof” is a system where no additional knowledge is transmitted besides the proof.

For example in a Zero-Knowledge proof system a prover can prove to a verifier that the number 221 is not a prime without revealing the factorisation $11 \cdot 13 = 221$

Notion of a simulator The main idea behind the concept of Zero-Knowledge is that whatever the verifier could learn from data transmitted by the prover he could have leaned by himself.

This idea is technically formalised by defining the ”view” of a verifier as a random variable U. A view is bits the verifier sees on tapes during the execution of the interactive protocol.

We can say that an interactive protocol is ”Zero-Knowledge” if there exists a probabilistic Turing machine M or later labeled as a ”simulator” whose output is statistically indistinguishable from the random variable U that the verifier ”sees” in an interaction with an honest prover.

4 Zero-Knowledge

Interactive proof system formalisation defined by Goldwasser, Micali and Rackoff [1].

The definition of Zero-Knowledge applies to any interactive protocol (A, B) not necessarily just interactive proof systems.

We can say that an interaction is ZK when the random variable of a view the B has is computationally indistinguishable from a distribution that can be polynomially computed from the parameter x when interacting with A.

4.1 Proof system

In their paper [1] Goldwasser, Micali and Rackoff defined an ”interactive proof system” as a system in which a prover can on input x create a string α, with which the ”verifier” can compute on x and α to check whether x is in L.

The main purpose of the paper was how much information in transmitted to the verifier in the interactive proof system for L. An example provided by the authors:

> Consider SAT, the NP-complete language. In the obvious proof system, to prove $F \in SAT$, the prover gives a satisfying assignment I for the formula F, which the verifier then checks in polynomial time. This assignment give the verifier much more knowledge than merely the fact that $F \in SAT$; it also gives a satisfying assignment.

4.2 Definition

4.3 Definiton of Zero-Knowledge

5 Interactive Zero-Knowlege proofs

5.1 Interactive proof systems

5.2 Common NP-Complete problems

5.2.1 Quadratic residue

5.2.2 Graph isomorphism

5.2.3 Discrete logarithm

6 Non-Interactive Zero knowledge proofs

6.1 zk-SNARK

6.1.1 Shamir-Fiat heuristic

6.2 Bulletproofs

6.3 zk-STARK

References

[1] S Goldwasser, S Micali, and C Rackoff. “The Knowledge Complexity of Interactive Proof-Systems”. In: *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*. STOC âĂŹ85.

Providence, Rhode Island, USA: Association for Computing Machinery, 1985, 291âĂŞ304. ISBN: 0897911512. DOI: 10.1145/22145.22178. URL: https://doi.org/10.1145/22145.22178.