Securing the Zero-Knowledge Proof of Quadratic Residuosity agains rainbow attacks in a HTTP authentication system

Jakob Povsic

2020

Contents

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 [**10.1145/22145.22178**] 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

These ideas were originally defined in a paper published in 1985 by Goldwasser, Micali and Rackoff [2].

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 $11 \cdot 13 = 221$

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.
Considering the verifier is an ”interactive Turing machine”, a view are 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 subsequently labeled as a ”simulator” whose output is computationally from the random variable U that the verifier ”sees” in an interaction with an honest prover.

Indistinguishability of random variables. Random variables U and V are considered ”replaceable” when our predictions of the origin of a random sample x is uncorrelated to the distribution from which the sample came.
By focusing on the sample size and the computation time we can by bounding the parameters focus on 3 notions of indistinguishability: equality, statistical indistinguishability and computational indistinguishability. Two distributions can be considered equal if we cannot correlate the distributions given an infinite sample size and infinite computing power.
Two random variables are statistically indistinguishable if the remain uncorrelated give an polynomial sample size and infinite computing power, while they are computationally indistinguishable given polynomial time and polynomial computing power.

4 Languages with Zero-Knowledge interactive proof systems

Bounded-error probabilistic polynomial time languages. Goldwasser, Micali and Rackoff [2] have identified all BBP languages have zero-knowledge proof systems.

> Trivially, all languages in BPP have perfect zero-knowledge proof systems. (A language is in BPP if there is a probabilistic, polynomial time machine which on each input computes membership in the language with small probability of error.)

Languages not known to be in BPP. First languages to be recognised to have a zero-knowledge proof system was quadratic residuosity and quadratic non-residuosity languages.
Graph isomorphism and non-isomorphism have been shown to have zero-knowledge proof system as well.

4.1 Examples of languages

Discrete logarithm In [2] a zero-knowledge proof system has been proposed for quadratic residuosity, a specific variant of the discrete logarithm problem.

Protocol. Quadratic residue
Public input. x, n
Provers private input. w, such that $x \equiv w^2 \pmod{n}$
P $\rightarrow$ V: Prover chooses random $u \leftarrow \mathbb{Z}_n^*$ and sends $y = u^2$ to the verifier.
P $\leftarrow$ V: Verifier chooses $b \leftarrow_R \{0, 1\}$
P $\rightarrow$ V: If $b = 0$ Peggy sends u to the Victor, if $b = 1$ Peggy sends $w \cdot u \pmod{n}$.

Verification Let z be the value sent by Peggy. Victor accepts the proof in case $b = 0, z^2 \equiv y \pmod{n}$ or $b = 1, z^2 \equiv xy \pmod{n}$

4.2 NP languages

In their paper [1] Goldreich, Micali and Wigderson proposed a zero-knowledge proof system for Graph 3-Colorability problem. By reducing Graph 3-Colorability to any $L \in NP$, they have proved that any language $L \in NP$ has a zero-knowledge proof system.

4.2.1 Zero-Knowledge proofs for Graph 3-Colorability

The protocol assumes an arbitrarily secure encryption scheme.

At the beginning of each iteration the prover sends $R_1, R_2, ..., R_n$, an encrypted permutation of the 3-colouring of the common input graph $G(V, E)$ to the verifier.

The verifier sends a random edge $e \in_R E$ to the prover.

The prover sends a permuted colouring of v and u, ($e = (v, u) \in E$) and the encryption secret for R_v, R_u.

The verifier checks the proof by encrypting the permutations and comparing them with the encrypted permutations submitted at the beginning, by checking that the permuted colourings for v and u are not equal, and by checking that permuted colourings are in the $\{1, 2, 3\}$ set.

After successfully repeating for m^2 iterations, $m = |E|$ for a graph $G(V, E)$, the verifier accepts.

4.2.2 Zero-Knowledge proofs for Languages in NP

Cook-Levin theorem states that any problem in $L \in NP$ can be reduced to the Boolean-Satisfiability problem in polynomial time by a deterministic Turing machine. There also exists a reduction from Graph 3-Colorability problem to SAT (Boolean-Satisfiability problem), meaning that there exists an efficient fixed reduction of every language $L \in NP$ to Graph 3-Colorability, parties can compute a 3-col instance from a common input and follow the zero-knowledge protocol.

4.3 Graph isomorphism

5 Non-Interactive Zero knowledge proofs

Non-interactive zero-knowledge proof are zero-knowledge proofs that require no interaction between the prover and the verifier

5.1 Shamir-Fiat heuristic

6 Application of Zero-Knowledge protocol as a password based authenticaton method over HTTP

References

[1] Oded Goldreich, Silvio Micali, and Avi Wigderson. “How to Prove all NP-Statements in Zero-Knowledge, and a Methodology of Cryptographic Protocol Design”. In: vol. 263. Aug. 1986, pp. 171–185. DOI: 10.1007/3-540-47721-7_11.

[2] 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.