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

Jakob Povsic

2020

Contents

1 Abstract

In this thesis I will introduce the notion of zero-knowledge proofs, their variants and formal definitions. In the second part of the thesis I will present a protocol for password based authentication based on the zero-knowledge proof of quadratic residuosity. In the last part of the thesis I will document the implementation of the protocol as an EAP authentication method.

2 Introduction

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

2.1 Example

A famous example of a zero-knowledge proof protocol made by [QQQ+89] is The Strange Cave of Ali Baba.

Ali Baba's cave is a cave with a single entrance, that splits into two tunnels that meet in the middle. Where the tunnels meet is a door that can only be opened with a secret passphrase.

Peggy* wants to prove to Victor † that she knows the secret passphrase, but she doesn't want to revel the secret nor does she want to reveal her knowledge of the secret to anyone else besides

Victor turns away from the entrance of the cave, so he cannot Peggy. Peggy enters the cave and goes into one of the tunnels at random. Victor turns around and tells Peggy which tunnel to come out of. Peggy knowing the secret can pass through the door in the middle and emerge from the tunnel requested by Victor.

Given that Peggy didn't know the secret she would still be able to emerge from the tunnel that she initially entered if Victor requested it. Since Victor is choosing the tunnel at random, Peggy has 50% chance of entering the correct tunnel, and by repeating this process her chances of cheating become vanishingly small ($\frac{1}{2^n}$).

Inversely if Peggy emerges from tunnels Victor requests, he can be convinced that Peggy knows the secret passphrase with a very high probability ($1 - \frac{1}{2^n}$).

Further more any third party observing the interaction cannot be convinced of the validity of the proof because it cannot be assured that the interaction was truly random. For example, Victor could have told Peggy his questions in advance, so Peggy would produce a convincing looking proof.

*Peggy is acronym for **Prover**

†Victor is an acronym for **Verifier**

3 Zero-Knowledge proofs

Traditional theorem proofs are logical arguments that establish truth through inference rules of a deductive system based on axioms and other proven theorems.

Zero-knowledge proofs in contrast are probabilistic in nature and can *"convince"* the verifier of the truth of a theorem with an arbitrarily small probability of error.

First defined by Goldwasser, Micali and Rackoff in [GMR85] as an interactive two-party protocol between a prover and a verifier.

The parties are technically defined as interactive Truing machines in a *pair of interactive Turing machines setup*.

The protocol uses the the quadratic residuosity problem to embed its proof.

There are three main ingredients that make interactive zero-knowledge proofs work.

1. Interaction - The prover and the verifier exchange messages back and forth.

2. Hidden Randomization - The verifier relies on randomness that is hidden from the prover, and thus unpredictable from him.

3. Computational Difficulty - The prover embeds his proof in computational difficulty of some other problem.

3.1 Interactive Zero-Knowledge Proofs

[‡]*Interactive zero-knowledge proof systems* are formally defined as interactive proof systems that conveys in zero-knowledge the proof of x membership in language L.

Interactive zero-knowledge proof systems are defined by three properties.

Completeness

Any honest prover can convince the verifier with overwhelming probability.

For each $k \in \mathbb{N}$ and sufficiently large n;

$Pr(x \in L; P(x) = y; V(y) = 1) \geq 1 - \frac{1}{n^k}$

Soundness

Any verifier following the protocol will reject a cheating prover with overwhelming probability.

For each $k \in \mathbb{N}$ and sufficiently large n;

$Pr(x \notin L; P(x) = y; V(y) = 0) \geq 1 - \frac{1}{n^k}$

[‡]Initially ZKPS were defined as *interactive* two-party protocols, later on *non-interactive* [BFM88] were described that slightly differ in some nuances, which we will explore further on in this thesis.

Zero-Knowledge

In a zero-knowledge proof system for L, a verifier can in polynomial time extract only the proof of membership in L when interacting with a prover. The essence of such a system is the idea that the verifiers "view" of an interaction with a prover, can be "simulated" in polynomial time.

Any interactive protocol is zero-knowledge if the probability distribution of observed messages is indistinguishable from a distribution that can be simulated on public inputs.

3.1.1 Indistinguishability of Random Variables

Let $U = \{U(x)\}$ and $V = \{V(x)\}$ be two families of random variables, where x is from a language L, a particular subset of $\{0,1\}^*$.

In the framework for distinguishing between random variables, a "judge" is given a sample selected randomly from either $V(x)$ or $U(x)$. A judge studies the sample and outputs either a 0 or a 1, depending on which distribution he thinks the sample came from.

$U(x)$ essentially becomes "replaceable" by $V(x)$, when x increases and any judges prediction becomes uncorrelated with the origin distribution. By bounding the *size* of the sample and the *time* given to the judge we can obtain different notions of indistinguishability.

Equality Given that $U(x)$ and $V(x)$ are equal, they will remain indistinguishable, even if the samples are of arbitrary size and can be studied for an arbitrary amount of time.

Statistical Indistinguishability Two random variables are statistically indistinguishable, when given a polynomial sized sample and an arbitrary amount of time, the judges verdict remains meaningless.

Let $L \subset \{0,1\}^*$ be a language, $U(x)$ and $V(x)$ are statistically indistinguishable on L if,

$$\sum_{\alpha \in \{0,1\}^*} |prob(U(x) = \alpha) - prob(V(x) = \alpha)| < |x|^{-c}$$

for $\forall c > 0$, and sufficiently long $x \in L$.

Computational Indistinguishability Two random variables are computationally indistinguishable, if judges verdict remains meaningless given a polynomial sized sample and polynomial amount of time.

Let $L \subset \{0,1\}^*$ be a language, poly-bounded families of random variables $U(x)$ and $V(x)$ are computationally indistinguishable on L if for all poly-sized family of circuits C, $\forall c > 0$, and a sufficiently long $x \in L$

$$|P(U, C, x) - P(V, C, x)| < |x|^{-c}$$

Any two families that are *computationally indistinguishable* are considered *indistinguishable* in general.

3.1.2 Approximability of Random Variables

The notion of approximability described the degree to which a random variable $U(x)$ can be "generated" by a probabilistic Turing machine.

A random variable $U(x)$ is *perfectly approximable* if there exists a probabilistic Turing machine M, such that for $x \in L$ $M(x)$ is *equal* to $U(x)$.
$U(x)$ is statistically or computationally approximable if $M(x)$ is statistically or computationally indistinguishable from $U(x)$.

Generally speaking when saying a family of random variables $U(x)$ is *approximable* we mean that it is *computationally* approximable.

3.1.3 Definition of Zero-Knowledge

What the zero-knowledge property is addressing is the absence of meaningful information that can be extracted from the protocol by an honest or an cheating verifier.
The verifiers *view* is all data he sees in the interaction with the prover as well data already possessed by the verifier, for example previous interactions with the prover.

An interactive protocol is *perfectly* zero-knowledge if the verifiers view is *perfectly approximable* for all verifiers. Generally we say an interactive protocol is zero-knowledge when its *computationally* zero-knowledge.

4 Languages with Zero-Knowledge Interactive Proof Systems

One of the main components that make Zero-Knowledge proofs work is the encoding of the proof in the *solution* of another "problem". The choice of the "problem" heavily relies on the specific application of the ZKP protocol.

A technical term from the computational complexity theory for a "problem" is *language*. And the "problem" is the task of proving the membership of x in language L

Alongside specific ZKPs described for specific languages, ZKPs have been also studies for classes of languages defined by their computational complexity

4.1 Quadratic residuosity problem

The first language with a ZKP protocol described in [GMR85], was for quadratic residuosity with a *perfect* zero-knowledge protocol and for quadratic non-residuosity with a *statistically* zero-knowledge protocol.

The problem of quadratic residuosity is much older however and was first described by Gauss in 1801. Quadratic residues come from modular arithmetic a branch of number theory.

Quadratic Residues For $a, n \in \mathbb{Z}$, $n > 0$, a and n are co-prime. a is a *quadratic residue* if $\exists x : x^2 \equiv a \pmod{n}$, otherwise a is a *quadratic non-residue*

Problem Given numbers a and $n = pq$, where p and q are unknown different primes, and [§]$(\frac{a}{n}) = 1$, determine wether a is a quadratic residue modulo n or not.

This problem is difficult because no efficient algorithm exists for factorisation of semi-primes.

[§]Jacobi symbol

Protocol

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 $w \cdot u \pmod{n}$.

Bounded-error probabilistic polynomial time languages. Goldwasser, Micali and Rackoff [GMR85] 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.

4.2 Concrete examples of languages with zero-knowledge proof systems

4.2.1 Discrete logarithm

In [GMR85] 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 $b = 1, z^2 \equiv xy \pmod{n}$

4.3 NP languages

In their paper [GMW86] Goldreich, Micali and Wigderson proposed a zero-knowledge proof system for Graph 3-Colorability problem. By reducing any $L \in NP$ to Graph 3-Colorability, they have proved that any language $L \in NP$ has a zero-knowledge proof system.

4.3.1 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.

4.3.2 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.

4.4 Non-Interactive Zero-Knowledge proofs

4.4.1 Introduction

In the execution of any *interactive* zero-knowledge proof protocol the most expensive resource is the *"interaction"* itself. Any cryptographic computations can be done relatively quickly compared to how long it takes to exchange a message between the prover and the verifier a few hundred times.

Blum, Feldman and Micali [BFM88] showed that a non-interactive zero-knowledge protocol is achievable by sharing a common reference string between the prover and the verifier. The original protocol was single-theorem and required a fresh reference string for each proof.

Properties of zero-knowledge proofs

1. Interaction: The prover and the verifier talk back and forth.
2. Hidden Randomisation: The verifier tosses coins that are hidden from the prover and thus unpredictable to him.
3. Computational Difficulty: The prover embeds in his proofs the computational difficulty of some other problem.

Non-interactive zero-knowledge proofs are different as interaction between the prover and the verifier is one directional, and the zero-knowledge no longer relies on the secrecy of the randomisation.

The only property lacking in NIZKPs is deniability.

The paper proved the existence of a NIZK for all languages $L \in NP$, by providing providing a protocol for proving the 3COL problem.

The paper [BFM88] describes a protocol for proving the Graph 3-Colorability problem, to which any lanugange $L \in NP$ can be reduced to according to the Cook-Levin theorem.

4.5 Applications

Modern NIZKPs zk-SNARK (zero-knowledge succinct non-interactive argument of knowledge) and Bulletproofs are used in privacy preserving cryptocurrencies Zcash and Monero.

4.6 Defintion

NIZK are defined between a Probabilistic Turing machine A (Prover) and a polynomial deterministic algorithm (Verifier). The definition of NIZK still contains completeness, soundness and zero-knowledge. With a difference in that the definition of zero-knowledge is simpler because the of no interaction between the verifier and the prover, meaning we do not have to worry about a possibility of a cheating verifier.

4.7 Fiat-Shamir Heuristic

Fiat and Shamir [FS87], described a signature scheme

References

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and applications. *Proceedings of the Annual ACM Symposium on Theory of Computing*, pages 103–112, 01 1988.

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology — CRYPTO' 86*, pages 186–194, Berlin, Heidelberg, 1987. Springer Berlin Heidelberg.

[GMR85] 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, page 291âĂŞ304, New York, NY, USA, 1985. Association for Computing Machinery.

[GMW86] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to prove all np-statements in zero-knowledge, and a methodology of cryptographic protocol design. volume 263, pages 171–185, 08 1986.

[QQQ+89] Jean-Jacques Quisquater, Myriam Quisquater, Muriel Quisquater, MichaÃńl Quisquater, Louis Guillou, Marie Guillou, GaÃŕd Guillou, Anna Guillou, GwenolÃľ Guillou, Soazig Guillou, and Thomas Berson. How to explain zero-knowledge protocols to your children. pages 628–631, 08 1989.