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 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 Proof Systems

Interactive proof systems are proof systems between a prover and a verifier. The prover is a computationally unbounded polynomial time Turing machine and the verifier is a probabilistic polynomial time Turing machine. *Completeness* and *soundness* are enough to define an interactive proof system. The **IP** complexity class

3.1.1 Completeness

Any honest prover can convince the verifier with overwhelming probability.

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

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

3.1.2 Soundness

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

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

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

3.1.3 Interactive Polynomial Time Complexity

Any problem solvable by an interactive proof systems is in the class of **IP**.

3.1.4 Other Variants of Interactive Proof Sytems

Arthur-Merlin protocol Problems in the class AM, the Arthur-Merlin protocol is similar to IP, with the difference in that its a *public-coin protocol*. Meaning that verifiers internal state is visible to the prover, while in IP the state is hidden. I has been proven that AM is equally powerful as IP and that AM's public internal state gives the prover no advantage.

Multi Prover Interactive Proofs

3.2 Interactive Zero-Knowledge Proofs

Interactive zero-knowledge proof systems are *interactive proof systems* that conveys in zero-knowledge the proof of x membership in language L. All together an interactive zero-knowledge proof system is define b by three properties.

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.

3.2.1 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.

3.2.2 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.

3.2.3 Definition of Zero-Knowledge

The zero-knowledge property is addressing 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 theoretical 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 languages with ZKPs, they have been also studies related to classes of languages defined by their computational complexity.

In this thesis we are focusing on the zero-knowledge proof of quadratic residuosity, but generally ZKP protocols exists for any language in NP [GMW86], assuming one way-functions exist in IP[‡]

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.

[‡] Class of problems solved by an *interactive proof system*

[§] Jacobi symbol

The problem of quadratic residuosity is considered difficult, because prime factorisation is hard.

Protocol

Public inputs $n, x : (\frac{x}{n}) = 1$ and
Provers private input $w : x \equiv w^2 \pmod{n}$

- 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 $b = 0$ Prover sends u to the Verifier, if $b = 1$ Prover sends $z = w \cdot u \pmod{n}$.
- Verifier accepts if, $[b = 0], z^2 \equiv y \pmod{n}$ or $[b = 1], z^2 \equiv xy \pmod{n}$ or rejects and halts otherwise.

This protocol is repeated m times.

4.2 Computational Complexity Classes

4.2.1 Bounded-Error Probabilistic Polynomial Time Languages

Or **BBP** in short, is in computational complexity theory a class of problems solvable by a probabilistic Turing machine in polynomial time with a bounded error to at most $1/3$ or $2^{-ck}; c > 0$ for k iterations.

4.2.2 Non-deterministic Polynomial Time

Or **NP** is a class of problems solvable by a non-deterministic Turing machine in polynomial time. Or rather proof of any language in NP can be verified by a deterministic polynomial time Turing machine.

In [GMW86] proved that every problem in NP has a zero-knowledge proof system, by describing a ZKP protocol for the Graph 3-Colouring problem (3-COL)

Minimum colouring problem is problem in graph theory, of what is the minimal k *proper* colouring of a graph, so that no adjacent vertices are the same colour.

An instance of 3-COL is proven to be *NP-Hard* because a polynomial reduction exists from *Boolean-Satisfiability problem* (3-SAT) to 3-COL [Mou].

According to Cook's theorem [Coo71] 3-SAT is NP-Complete, and any language in $L \in NP$ can be reduced by a polynomial deterministic Turing machine to 3-SAT. Furthermore because polynomial reductions are *transient*, any language $L \in NP$ can be reduced to an instance of 3-COL.

5 Application as a password based authentication system

5.1 Modern password based authentication systems

Modern password based authentication systems usually have two functions.

Setup The user first creates a unique identifier in the system and then picks a password. Both values are submitted to the protected system. The identifier is stored in plain text, while the password is ”extended” by a key derivation function and stored.

Key Derivation Key derivation is a process of ”extending” a weak key, by using an additional high entropy value called ”salt”.

Authentication When the user wants to authenticate with the system, he submits the identifier and the password in plain text. The system uses the identifier to locate

References

[Coo71] Stephen A Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158, 1971.

[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.

[Mou] Lalla Mouatadid. Introduction to complexity theory: 3-colouring is np-complete.

[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.