Cryptography 101: From Theory to Practice

Chapter 15 – Zero-Knowledge Proofs of Knowledge

Rolf Oppliger

March 30, 2022

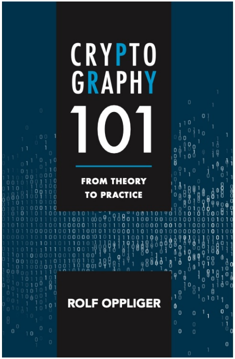
Terms of Use

whoami

rolf-oppliger.ch
rolf-oppliger.com

- Swiss National Cyber Security Centre NCSC (scientific employee)
- eSECURITY Technologies Rolf Oppliger (founder and owner)
- University of Zurich (adjunct professor)
- Artech House (author and series editor for information security and privacy)

Reference Book



ISBN 978-1-63081-846-3

https://books.esecurity.ch/crypto101.html

Challenge Me

Outline

15. Zero-Knowledge Proofs of Knowledge

15. Zero-Knowledge Proofs of Knowledge

15. Zero-Knowledge Proofs of Knowledge

15.1 Introduction

- On a high level of abstraction, a **proof** is just a method to establish truth
- To prove a claim, one has to convince somebody (or everybody) that the claim is true
- The details of a proof depend on the situation (and whether one has a philosophical, legal, scientific, or mathematical stance)
- Cryptography is about applied mathematics, so the stance is purely mathematical

15. Zero-Knowledge Proofs of Knowledge

15.1 Introduction

- The goal of a (mathematical) proof is to derive a claim from a set of axioms, using some well-defined (syntactical and semantical) derivation rules
- No matter who provides the proof and how it is generated, it must be complete and sound, and each derivation step must be comprehensible and logical
- If a single step is missing, then the entire proof is invalid and must be rejected

15. Zero-Knowledge Proofs of Knowledge

15.1 Introduction

- In this sense, a (mathematical) proof is verifier-centric, meaning that only the verifier is needed
- Whoever generates the proof and with what computational power is pointless and doesn't matter
- The proof is independent from the prover and is transferable by default
- This means that the proof can be shown to anybody, and that any person can — at least in principle — verify the proof

15. Zero-Knowledge Proofs of Knowledge

15.1 Introduction

- Every decision problem can be expressed as a **language membership problem**
- For a given input $x \in \{0,1\}^*$, it must be decided whether it is a member of language $L \subseteq \{0,1\}^*$ (i.e., YES or NO)

$$L = \{x \mid \exists \pi : V(x, \pi) = \text{YES}\}$$

- The language L thus consists of all $x \in \{0,1\}^*$, for which there is a proof π that can be verified by V

15. Zero-Knowledge Proofs of Knowledge

15.1 Introduction

Definition 15.1 (Proof system)

A proof system for membership in L is an algorithm V, such that for all $x \in \{0,1\}^*$ the following two requirements are fulfilled:

- **Completeness:** If $x \in L$, then there exists a proof π with $V(x,\pi) = \text{YES}$
- **Soundness:** If $x \notin L$, then for all proofs π it must be the case that $V(x,\pi) = \text{NO}$

A proof system is complete if all $x \in L$ can be proven to be in L, and it is sound if no $x \notin L$ can be proven to be in L

15. Zero-Knowledge Proofs of Knowledge

15.1 Introduction

- Such a proof system is **efficient** (or an **NP proof system**), if V is also efficient
- This means that $V(x, \pi)$ halts after at most a polynomial number of steps for every x and π (where the polynomial is taken over the length of x)
- A proof system allows one to prove language membership, but it does not automatically allow one to prove nonmembership, i.e., $x \notin L$

15. Zero-Knowledge Proofs of Knowledge

15.1 Introduction

- This is where the work of Shafi Goldwasser, Silvio Micali, and Charles Rackoff comes into play

15. Zero-Knowledge Proofs of Knowledge

15.1 Introduction

- After the discovery of public key cryptography in the 1970s, this work was the next major breakthrough in modern cryptography (in the 1980s)
- They modified the notion of a proof system by introducing two ingredients
  - Randomness and the possibility of make errors
  - Interaction
- The resulting proof systems are called **interactive**
- An interactive proof is modeled after a factual proof in the real world (e.g., Pepsi Challenge)

15. Zero-Knowledge Proofs of Knowledge

15.1 Introduction

Definition 15.2 (Interactive proof system)

An interactive proof system for membership in L is a pair (P, V) that consists of a function P (prover) and a PPT algorithm V (verifier), such that for all $x \in \{0,1\}^*$ the following two requirements are fulfilled:

- **Completeness:** If $x \in L$, then $\Pr[(P,V)(x) = \text{YES}] \geq 2/3$
- **Soundness:** If $x \notin L$, then for all P' it must hold that $\Pr[(P',V)(x) = \text{YES}] \leq 1/3$

The values $2/3$ and $1/3$ are arbitrary and can be replaced with $1/2 + 1/p(|x|)$ (instead of $2/3$) and $1/2 - 1/p(|x|)$ (instead of $1/3$) for some polynomial $p(\cdot)$

15. Zero-Knowledge Proofs of Knowledge

15.1 Introduction

- An interactive proof system has the **zero-knowledge** property, if whatever V can compute when interacting with P it can also compute without interacting with P
- If $V(view)$ refers to V's view of a protocol execution with P (that includes x, all random values chosen by V, and all messages exchanged between P and V), then a protocol leaks no information, if $V(view)$ can be efficiently simulated without interacting with P
- This means that there is an efficient algorithm S (simulator) that can generate $S(x)$ that is indistinguishable from $V(view)$, i.e., $S(x) \cong V(view)$

15. Zero-Knowledge Proofs of Knowledge

15.1 Introduction

Definition 15.3 (Zero-knowledge)

An interactive proof system (P, V) for L is (computationally) *zero-knowledge* if there exists a PPT algorithm S, such that for all $x \in L$ the relation $S(x) \cong (P, V)(x)$ holds

- The simulation property or paradigm is key to zero-knowledge
- Note that it allows one to define zero-knowledge without having to define what knowledge is

15. Zero-Knowledge Proofs of Knowledge

15.2 Zero-Knowledge Authentication Protocols

- An interactive proof system can be used for entity authentication (e.g., challenge-response-based authentication protocols)
- The zero-knowledge property is useful, because it ensures that such a protocol leaks no information about the (secret) authentication information
- All protocols require a mechanism that allows the verifier to learn the prover's public key in some certified form (not further addressed here)

15. Zero-Knowledge Proofs of Knowledge

15.2 Zero-Knowledge Authentication Protocols — Fiat-Shamir

- In 1985, Goldwasser, Micali, and Rackoff published their seminal work
- In 1986, Amos Fiat and Shamir proposed the first authentication protocol that has the zero-knowledge property
- Similar to the Rabin public key cryptosystem, the Fiat-Shamir protocol is based on the modular square function $f(x) = x^2 \bmod n$ for $n = pq$
- It takes its security from the fact that computing square roots modulo n and factoring n are computationally equivalent

15. Zero-Knowledge Proofs of Knowledge

15.2 Zero-Knowledge Authentication Protocols — Fiat-Shamir

- The Fiat-Shamir protocol is a challenge-response protocol with an additional commitment step
- The prover P commits to a certain value before the challenge-response part takes place
- For $n = pq$, P has a private key x that is randomly chosen from $\mathbb{Z}_n^*$
- The respective public key $y = x^2 \bmod n$ is provided to the verifier V
- The protocol must be executed in multiple rounds (to make the success probability for cheating sufficiently small)

15. Zero-Knowledge Proofs of Knowledge

15.2 Zero-Knowledge Authentication Protocols — Fiat-Shamir

Table 15.1
A Round in the Fiat-Shamir Authentication Protocol

P		V
(n, x)		(n, y)
$r \xleftarrow{r} \mathbb{Z}_n^*$		
$t = r^2 \bmod n$	$\xrightarrow{t}$	
	$\xleftarrow{c}$	$c \xleftarrow{r} \{0, 1\}$
$s = (rx^c) \bmod n$	$\xrightarrow{s}$	
		$s^2 \stackrel{?}{\equiv} ty^c \pmod{n}$
		(*accept* or *reject*)

15. Zero-Knowledge Proofs of Knowledge

15.2 Zero-Knowledge Authentication Protocols — Fiat-Shamir

- The protocol is complete, because

$$s^2 \equiv r^2(x^c)^2 \equiv t(x^2)^c \equiv ty^c \pmod{n}$$

- To show that the system is sound, one must look at the adversary and ask what he or she can do in each round
- The adversary can randomly select a $t \in_R \mathbb{Z}_n^*$, wait for V to provide a challenge $c \in_R \{0, 1\}$, and then simply guess s
- The success probability is negligible
- There are more subtle attacks to consider

15. Zero-Knowledge Proofs of Knowledge

15.2 Zero-Knowledge Authentication Protocols — Fiat-Shamir

- If the adversary is able to predict c, then he or she can prepare himself or herself to provide the correct response
  - If $c = 0$, then the protocol can be executed as normal, i.e., the adversary can randomly select r and send $t = r^2 \bmod n$ and $s = r$ to V
  - If $c = 1$, then the adversary can randomly select $s \in_R \mathbb{Z}_n^*$, compute $t = (s^2/y) \bmod n$, and send these values to V
- It is not possible for the adversary to prepare himself or herself for both cases (otherwise, he or she could also extract the private key x)

15. Zero-Knowledge Proofs of Knowledge

15.2 Zero-Knowledge Authentication Protocols — Fiat-Shamir

- Because the adversary can predict the challenge c with a probability of $1/2$, the cheating probability is $1/2$ in each round
- This suggests that the protocol must be executed in multiple rounds
- If the protocol is repeated k times, then the cheating probability is $1/2^k$
- This value decreases exponentially and can be made arbitrarily small

15. Zero-Knowledge Proofs of Knowledge

15.2 Zero-Knowledge Authentication Protocols — Fiat-Shamir

- The Fiat-Shamir protocol has the zero-knowledge property, because a dishonest verifier V' can use an efficient program S to simulate the protocol and compute transcripts and triples (t, c, s) that are indistinguishable from real triples
- If $p = 3$, $q = 5$, $n = 15$, $x = 7$, $y = 7^2 \bmod 15 = 4$ and $y^{-1} \bmod 15 = 4$ $[4 \cdot 4 = 16 \equiv 1 \pmod{15}]$, then S can
  - Assume $c = 0$
  - Randomly select $r = 2$
  - Compute $t = 2^2 \bmod 15 = 4$ and $s = 2$
- The triple $(4, 0, 2)$ is computationally indistinguishable from a real protocol transcript

15. Zero-Knowledge Proofs of Knowledge

15.2 Zero-Knowledge Authentication Protocols — Fiat-Shamir

- Similarly, S can
  - Assume $c = 1$
  - Randomly select $s = 3$
  - Compute $t = 3^2 \cdot 4 \bmod 15 = 6$
- Again, the triple $(6, 1, 3)$ is computationally indistinguishable from a real protocol transcript
- The same is true for $(1, 1, 7)$, $(4, 0, 8)$, and so on and so forth
- The Fiat-Shamir protocol is conceptually simple, but it is not very efficient
- Consequently, there are several variants that speed things up using some form of parallelization

15. Zero-Knowledge Proofs of Knowledge

15.2 Zero-Knowledge Authentication Protocols — Guillou-Quisquater

- In 1988, Louis C. Guillou and Jean-Jacques Quisquater proposed a more efficient variant of the Fiat-Shamir protocol
- Instead of working with squares and binary challenges, the Guillou-Quisquater protocol works with e-th powers (where e is prime) and challenges between 0 and $e - 1$ (instead of 0 or 1)
- The security of the resulting protocol is based on the RSA problem, i.e., computing e-th roots modulo n without knowing the prime factorization of n or $\phi(n)$

15. Zero-Knowledge Proofs of Knowledge

15.2 Zero-Knowledge Authentication Protocols — Guillou-Quisquater

Table 15.2

A Round in the Guillou-Quisquater Authentication Protocol

P		V
(n, x)		(n, y)
$r \xleftarrow{r} \mathbb{Z}_n^*$		
$t = r^e \bmod n$	$\xrightarrow{t}$	
	$\xleftarrow{c}$	$c \xleftarrow{r} \{0, \ldots, e-1\}$
$s = (rx^c) \bmod n$	$\xrightarrow{s}$	
		$s^e \overset{?}{\equiv} ty^c \pmod{n}$
		$(accept$ or $reject)$

15. Zero-Knowledge Proofs of Knowledge

15.2 Zero-Knowledge Authentication Protocols — Schnorr

- In 1989, Schnorr proposed a zero-knowledge authentication protocol that is based on the DLP
- It is assumed that a large prime p and a generator g of $\mathbb{Z}_p^*$ are known (they can be either system parameters or part of the public key pairs), and that P has a private key x and a respective public key $y \equiv g^x \pmod{p}$
- K is a security parameter

15. Zero-Knowledge Proofs of Knowledge

15.2 Zero-Knowledge Authentication Protocols — Schnorr

Table 15.3

A Round in the Schnorr Authentication Protocol

P		**V**
(p, g, x)		(p, g, y)
$r \xleftarrow{r} \mathbb{Z}_p^*$		
$t = g^r \bmod p$	$\xrightarrow{t}$	
	$\xleftarrow{c}$	$c \xleftarrow{r} \{0, \ldots, 2^k - 1\}$
$s = r + cx \pmod{p-1}$	$\xrightarrow{s}$	
		$g^s \stackrel{?}{\equiv} ty^c \pmod{p}$
		$(\textit{accept}$ or $\textit{reject})$

15. Zero-Knowledge Proofs of Knowledge

15.3 Noninteractive Zero-Knowledge

- A zero-knowledge proof or protocol is interactive by default
- This means that there are messages sent back and forth (between P and V)
- There are application settings in which this level of interaction is neither possible nor welcome
- Consequently, people have been looking for possibilities to prove statements in zero-knowledge without requiring any form of interaction
- This leads to the notion of a noninteractive zero-knowledge proof, in which a single message is sent from P to V

15. Zero-Knowledge Proofs of Knowledge

15.3 Noninteractive Zero-Knowledge

- After having introduced the notion of zero-knowledge and the Fiat-Shamir protocol it became clear that the latter can be turned into a DSS (that is noninteractive)
- The idea is to replace c with a hash value $c = h(m, t)$ that takes into account the message m and the commitment t
- This idea has become known as the **Fiat-Shamir heuristic**
- The pair (t, s) then yields a digital signature for m

15. Zero-Knowledge Proofs of Knowledge

15.3 Noninteractive Zero-Knowledge

Table 15.4
Fiat-Shamir DSS

System parameters: —

Generate

(1^l)

$$p, q \xleftarrow{r} \mathbb{P}_{l/2}$$
$$n = p \cdot q$$
$$x \xleftarrow{r} \mathbb{Z}_n^*$$
$$y = x^2 \bmod n$$

$((n, x), (n, y))$

Sign

$((n, x), m)$

$$r \xleftarrow{r} \mathbb{Z}_n^*$$
$$t = r^2 \bmod n$$
$$c = h(m, t)$$
$$s = (rx^c) \bmod n$$

(t, s)

Verify

$((n, y), m, (t, s))$

$$b = (s^2 \equiv ty^c \pmod{n})$$

(b)

15. Zero-Knowledge Proofs of Knowledge

15.3 Noninteractive Zero-Knowledge

- More generally, noninteractive zero-knowledge proofs require no interaction between P and V
- Instead, a single message is sent from the prover to the verifier
- In 1988, Blum, Paul Feldman, and Micali showed that a common reference string generated by a trusted party and accessible to P and V is sufficient to achieve zero-knowledge without interaction
- Their model is called the **common reference string** model
- It was later shown that noninteractive zero-knowledge is impossible to achieve in the standard model

15. Zero-Knowledge Proofs of Knowledge

15.4 Final Remarks

- The notions of interactive proof systems and zero-knowledge were introduced in the mid-1980s
- For the first three decades, they were theoretically stimulating research topics but not really used in the field
- This has changed tremendously
- Zero-knowledge has experienced a strong revival, especially in its noninteractive form to provide computational integrity

15. Zero-Knowledge Proofs of Knowledge

15.4 Final Remarks

- Techniques
  - BulletProof
  - SNARK (Succinct Noninteractive ARgument of Knowledge)
  - zk-SNARK (zero-knowledge SNARK)
  - STARK (Scalable Transparent ARgument of Knowledge)
  - zk-STARK (zero-knowledge STARK)
  - ...
- They are heavily used in blockchains and cryptocurrencies (e.g., Zcash)

Questions and Answers

Thank you for your attention