0368.4162 Foundations of Cryptography May 22, 2019

Zero-Knowledge Proofs

Lecturer: Nir Bitansky

1 Must a Proof Convey Knowledge?

Typically, when we think about the concept of *a proof* our most essential requirement is that *they're convincing* — if a statement has a proof then it must be true. Different disciplines of course have different takes on what convincing means (e.g., it could be merely evidence in criminology, or perhaps a fully detailed mathematical proof in a calculus). Something that seems inherent to almost any kind of proof is that it doesn't only tell us that a given assertion x is true, but it also gives us some information on *how it is true* (e.g., a weapon with fingerprints suggests how a murder was performed). In some scenarios, we may wish to avoid conveying such knowledge (for instance, we may like to prove that there's an eye witness to a murder, without revealing their identity and thereby risking them).

Zero-knowledge proofs, introduced by Goldwasser, Micali and Rackoff [GMR85], suggest that proofs could be both convincing and yet, somewhat magically, not convey any knowledge, except of course for the fact that the assertion is true.

An Example: Sudoku Puzzles [GNPR07]. A Sudoku board is a 9×9 board divided into 3×3 subsquares where the goal is to fill the board with integers in $\{1, \ldots, 9\}$ so that every integer appears exactly once in every row, column, or subsquare. A corresponding puzzle comes with some initial hints (partial assignment of integers) and has to be completed to a full solution.

Can you convince someone that you know a solution without revealing anything about it?

	2		5		1		9	
8			2		3			6
	3			6			7	
		1				6		
5	4						1	9
		2				7		
	9			3			8	
2			8		4			7
	1		9		7		6	

Unsolved Sudoku

4	2	*6*	5	*7*	1	*3*	9	*8*
8	*5*	*7*	2	*9*	3	*1*	*4*	6
1	3	*9*	*4*	6	*8*	*2*	7	*5*
9	*7*	1	*3*	*8*	*5*	6	*2*	*4*
5	4	*3*	*7*	*2*	*6*	*8*	1	9
6	*8*	2	*1*	*4*	*9*	7	*5*	*3*
7	9	*4*	*6*	3	*2*	*5*	8	*1*
2	*6*	*5*	8	*1*	4	*9*	*3*	7
3	1	*8*	9	*5*	7	*4*	6	*2*

Solved Sudoku

Consider the following proof strategy using cards as props:

1. The prover, who knows a solution, takes 81 cards with nine sequences of the numbers $\{1, \ldots, 9\}$. It then places them on the board according to the solution, so that their backs (which all look the same) are facing the verifier. It flips over only the cards participating in the hint.

2. The verifier now chooses a random one of the three constraints (rows, columns, or subsquares) and asks the prover to prove that the constraint is satisfied.

3. The prover does this as follows. Say for instance that the verifier asked to prove consistency of the rows. The prover stacks each of the rows, and shuffles its cards at random behind her back (turning around is also fine), and hands the nine corresponding shuffled stacks to the verifier.

4. The verifier checks that each such stack contains all numbers $\{1, \ldots, 9\}$.

Notice that if the solution is valid, the prover will always manage to pass the test. However, if the prover placed an invalid solution on the board, there will be at least one type of constraint that is violated, and will catch her w.p. at least $1/3 \times 1/9$. By repeating this proof n times independently, the probability that the prover manages to cheat can be decreased exponentially in n.

What Does a Verifier Learn? If the solution was valid, in each such proof, the verifier simply obtains 9 stacks of $\{1, \ldots, 9\}$ at a random order. This doesn't convey any information — the verifier doesn't need a solution to generate 9 shuffled stacks, she can do it herself. In particular, the verifier doesn't gain the ability to prove to a third party that she knows a solution.

The above proof seems nothing like what we're used to. It's *interactive* and it's *randomized* (two aspects that turn out to be essential for zero knowledge). Is it just a card trick? or can we actually prove meaningful statements this way? does it have a digital analog where parties simply exchange messages? Actually, the last proof system is already very expressive (when generalized to $n^2 \times n^2$ Sudoku boards) and has a natural digital analog. First, let's figure out how to define these proof systems.

2 Defining Zero-Knowledge Proofs

We'll start by defining the concept of interactive proofs (IPs), without addressing the zero-knowledge requirement. Indeed, while born with zero-knowledge, this concept turned out to be fascinating on its own, and has completely revolutionized complexity theory (in particular, it can be seen as the seed for major results such as the characterization of PSPACE and the PCP theorem).

Definition 2.1 (Interactive Proofs (IPs))**.** *An interactive proof system (P, V) for a language $L \subseteq \{0,1\}^*$ consists of an efficient verifier V and a prover P, which are both interactive. We require:*

- **Completeness:** *for any $x \in L$,*
$$\Pr\left[\langle P, V\rangle(x) = 1\right] \geq 0.9 \ ,$$
where $\langle P, V\rangle(x)$ denote the output of V after a joint interaction with P on common input x.
- **Soundness:** *for any $x \notin L$ and any malicious prover strategy P^*,*
$$\Pr\left[\langle P, V\rangle(x) = 1\right] \leq 0.1 \ .$$

A common requirement is that the honest prover is also efficient, which we will call an *efficient prover IP*. The ability to obtain an efficient prover depends on the complexity of the language and may require that the honest prover will get an extra auxiliary input.[1] For instance, to prove NP statements efficiently we mush provide the prover with a witness.

Defining Zero Knowledge. We now move to formally define what it means for a protocol to be zero knowledge. The intuition that we'd like to capture is that

Whatever the verifier learns from a proof of a true statement, she could have learned on her own.

This is formally captured by requiring that there's an efficient simulation algorithm $S(x)$ that can *simulate* the *view* of the verifier.

Definition 2.2 (Zero Knowledge (ZK))**.** *A proof system (P, V) for L is zero knowledge if for any n.u. PPT V^* (with arbitrarily long output), there exists a n.u. PPT S such that for every $x \in L$,*
$$\langle P, V^*\rangle(x) \approx S(x) \ ,$$
where the random variable $\langle P, V^*\rangle(x)$ denotes V^*'s output after an interaction with P on common input x. Here $\approx$ denotes computational indistinguishability of distributions.

[1] Note that an efficient prover IP where the prover doesn't get any auxiliary input implies that $L \in BPP$ (think why).

Unlike the honest verifier that only outputs its decision bit, the malicious V^* can produce an output that arbitrarily (but efficiently) depends on its view in the protocol's execution. We can assume w.l.o.g that the verifier outputs the view itself, including all messages and her randomness.

The Simulation Paradigm More Broadly. The zero knowledge definition is a special case of a more general *simulation paradigm*, whose main idea is to capture at once *all the possible attacks*. Indeed, the verifier may try to learn different types of information. For instance, for an NP language L, it may try to learn a witness, or to just learn the first bit of the witness. Simulation guarantees that all of these potential attacks would fail, unless these are pieces of information that can anyhow be learned from the instance.

More generally, the simulation paradigm aims to capture the fact that whatever effect the adversary can have on our cryptographic scheme or protocol is restricted to what the adversary can do in *an ideal world*. (In ZK ideal world, the only information given to the adversary is the fact that the statement is true.)

3 Constructing Zero Knowledge Protocols

Theorem 3.1 ([GMW86])**.** *Assuming OWFs, any NP language has a zero-knowledge proof.*

We will prove this theorem, but first we need to define commitments.

Commitments. To be precise, GMW did not assume OWFs, but a more expressive primitive called *a commitment scheme*, which were later constructed from OWFs [Nao89]. We will now define such commitments and use them to prove the GMW theorem. Intuitively speaking, such commitment schemes are digital analogs of "locked boxes". The sender puts a message in a box, locks it, and sends it to the receiver. The receiver cannot see what's in the box, until the opening phase, when the sender unlocks the box. The sender on the other hand cannot change the message, after she had sent the box.

Today, we'll define and use a simple case of commitments schemes that are non-interactive. The more general definition addresses protocols.

Definition 3.2 (Non-Interactive Commitment Scheme)**.** *A commitment scheme is given by a PPT commitment algorithm Com satisfying:*

- **Computational Hiding:**

$$\{Com(m_0)\}_{\substack{n\in\mathbb{N}\\ m_0,m_1\in\{0,1\}^n}} \approx_c \{Com(m_1)\}_{\substack{n\in\mathbb{N}\\ m_0,m_1\in\{0,1\}^n}} \ .$$

- **Perfect Binding:** *For any messages m_0, m_1 and randomness r_0, r_1 if $Com(m_0; r_0) = Com(m_1; r_1)$, then $m_0 = m_1$. (This means that any commitment can be "opened", by exhibiting a randomness, to a unique message.)*

The GMW Protocol. To construct a protocol for any NP language it suffices to construct a protocol for one NP-complete language The GMW protocol is for the language of 3-colorable graphs

$$3COL = \{(U, E) \mid \exists \sigma : U \to [3] : \forall (u, v) \in E, \sigma(u) \neq \sigma(v)\} \ .$$

We'll design a zero knowledge proof system (P, V) for the language $3COL$ that has an efficient prover that gets as auxiliary input a legal coloring σ of G. The system will have perfect completeness (i.e. the completeness error is $\sigma = 0$) and soundness error $s = 1 - \frac{1}{|E|}$.

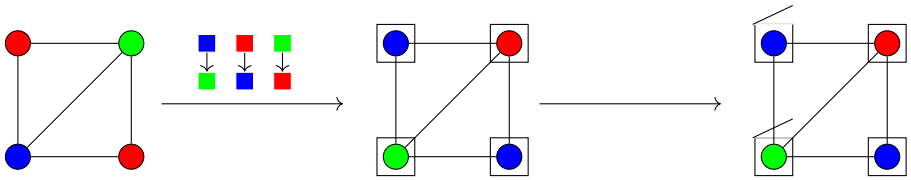

Figure: Illustration of the GMW protocol.
The sealed black boxes represent the commitments. The open boxes are the opened commitments.

Protocol $\langle P(\sigma), V\rangle\,(U, E)$**:**

1. P: chooses a random permutation $\varphi : [3] \to [3]$, and sends V commitments to all colors after applying the permutation:
$$(Com(\varphi(\sigma(v))) \mid v \in U) \quad .$$
2. V: samples $e \leftarrow E$ at random and sends $e = (u, v)$ to P.
3. P opens the commitments corresponding to u and v.
4. V verifies that the two opened colors are distinct.

Claim 3.3. *The above protocol is a zero knowledge proof with soundness error* $1 - \frac{1}{|E|}$.

Proof Sketch. Once again completeness of the protocol is straightforward and we'll focus on soundness and ZK.

Soundness. To see that the protocol has soundness error $s = 1 - \frac{1}{|E|}$, note that if (U, E) is not three colorable, then in any coloring, there must exist an edge e that is not properly colored. Since the commitment is perfectly binding, it fixes a coloring of the corresponding vertices (some of the commitments may be invalid altogether, so this coloring may be partial). The probability that the verifier asks to open an improper edge (where improper may mean either that the coloring is improper or that the commitments are invalid) is thus at least $1/|E|$.

Zero-Knowledge Simulation. To get some high-level intuition re zero knowledge, let us indeed think of the commitments as ideal locked boxes. Then, all that the verifier sees is a bunch of opaque boxes, it then opens two boxes of his choice and simply sees two distinct random colors, this is something that it could simulate on its own. We will only formally show a simulator for honest verifiers (who output their view: their randomness and received prover messages).

The simulator will act as follows:

1. S samples an a random edge $e \leftarrow E$ as the verifier's randomness.
2. It then computes a set of commitments $\{C_u \mid u \in U\}$ as the first message:
   - For every $w \notin e$, $C_w = Com(0)$ is a commitment to junk (say zero).
   - For $e = (u, v)$, it picks two random distinct colors $\sigma(u), \sigma(v) \in \{1, 2, 3\}$, and sets $C_u = Com(\sigma(u))$ and similarly $C_v = Com(\sigma(v))$.
3. The third message is simply the opening of C_u, C_v.

The only difference between the simulated view and the real view of the (honest verifier) is that in the real view the verifier obtains commitments to an actual coloring, rather than junk. However, due to the hiding of the commitment, the two are indistinguishable. □

References

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing, May 6-8, 1985, Providence, Rhode Island, USA*, pages 291–304, 1985.

[GMW86] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to prove all np-statements in zero-knowledge, and a methodology of cryptographic protocol design. In *Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings*, pages 171–185, 1986.

[GNPR07] Ronen Gradwohl, Moni Naor, Benny Pinkas, and Guy N. Rothblum. Cryptographic and physical zero-knowledge proof systems for solutions of sudoku puzzles. In *Fun with Algorithms, 4th International Conference, FUN 2007, Castiglioncello, Italy, June 3-5, 2007, Proceedings*, pages 166–182, 2007.

[Nao89] Moni Naor. Bit commitment using pseudo-randomness. In *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, pages 128–136, 1989.