0368.4162 Foundations of Cryptography May 29, 2019

Non-Interactive Zero-Knowledge

Lecturer: Nir Bitansky

1 Previously

We've defined zero knowledge proofs (ZKPs) and explained how to use them in order to create anonymous coins. We then saw a ZKP system for all of NP (specifically, we've seen a ZKP for an NP-complete problem). This makes ZKPs very expressive, you can practically prove anything you can think of! In particular, they are the ultimate tool to achieving privacy on Blockchains. In principle, you can do everything encrypted (or under commitments) and prove that it is "valid" without revealing anything about it.

Today we will continue to talk about ZKPs and focus on certain properties that are essential for using them on the blockchain.

2 Recalling the Definition

Before that let's start by recalling their definition.

Definition 2.1 (Zero Knowledge Interactive Proofs (ZKPs) for NP)**.** *(P, V), where P, V are efficient prover and verifier is a ZKP for a relation $R \in NP$ if it satisfies:*

- **Completeness:** *for any $(x, w) \in R$,*

$$\Pr\left[\langle P(w), V\rangle(x) = 1\right] = 1 \ ,$$

  where $\langle P(w), V\rangle(x)$ denote the output of V after a joint interaction with $P(w)$ on common input x.

- **Soundness:** *for any $x \notin L(R)$ and any malicious prover strategy P^*,*

$$\Pr\left[\langle P, V\rangle(x) = 1\right] \leq negl(n) \ .$$

  If soundness only holds against bounded provers it's called an argument *rather than a proof.*

- **Zero Knowledge:** *for any efficient malicious verifier strategy V^* there exists an efficient simulator S such that for any $(x, w) \in R$*

$$View_{V^*}(\langle P(w), V^*\rangle(x)) \approx S(x) \ ,$$

  where the random variable $\langle P(w), V^*\rangle(x)$ denotes V^*'s output after an interaction with $P(w)$ on common input x. Here $\approx$ denotes computational indistinguishability of distributions.

3 Constructions

We have seen a three-message ZKP (Hamiltonicity) with soundness half. We proved it is *honest-verifier zero knowledge* (HVZK), but it is not hard to show that it is in fact also ZK against malicious verifiers.

How would you reduce the soundness error to 2^{-n}?

The natural answer is parallel repetition. That is, we will run n independent copies of the protocol in parallel.

Claim 3.1. *The n-fold parallel repetition of the Hamiltonicity protocol has soundness error* 2^{-n}.

Proof. Let (α, β, γ) denote the messages of a single instance of the protocol. Fix the (worst possible) first prover message $\alpha_1, \ldots, \alpha_n$ in the repeated protocol. Then for any α_i when choosing β_i at random, there is chance at most $1/2$ that this challenge can be answered in a way that satisfies the verifier. Thus the probability that all messages β_i allow satisfying the verifier is at most 2^{-n}. □

Q: Is the protocol still ZK? Why wouldn't it be?

A1: you can definitely prove that it is still HVZK.

A2: But it is unlikely to be ZK, we'll give soon a good reason for why that is the case. For now, you can ask your self what is the problem in simulating a malicious verifier (you'll have to think first about why simulating a malicious verifier in one instance is possible).

A3: We'll see soon that it does satisfy certain privacy properties also against malicious verifiers.

4 Non-Interactive ZK

Interactive proofs are somewhat challenging to use on the blockchain, e.g. for Zerocoin. Who would the prover interact with?

Can there be non-interactive zero knowledge?

Suggestion: apply Fiat-Shamir. One natural thing we can do is apply the Fiat-Shamir paradigm, which we've previously encountered. That is rather than having an actual verifier, the prover will derive the verifier's randomness using a hash function H, modeled as a random oracle.

Claim 4.1. *Let* $\alpha, \beta = H(\alpha), \gamma$ *be the Fiat-Shamir version of three-message proof with soundness error* s. *Then when* H *is modeled as a random oracle, any prover making at most* Q *queries to* H *can make the verifier accept with probability at most* $(Q+1)s$.

Proof sketch. Assume w.l.o.g that the prover queries α for the proof α, β, γ that it outputs. For any query α made to the oracle, the probability that $\beta = H(\alpha)$ has ana answer γ that would satisfy the verifier with probability at most s. The proof follows by a union bound over all $Q+1$ queries. □

So, in the random oracle model this will be sound, and you can hope that it remains sound when applying an appropriate hash function.

Can this be ZK?

Claim 4.2. *There cannot be a (completely) non-interactive ZKP for language* L, *unless it is easy to decide.*

Proof. If there is then S can efficiently decide the language. For any $x \in L$ it must create accepting proofs, because the real prover does. For any $x \notin L$ $S(x)$ will not output such proofs or it would break soundness. (Think: where did we use completeness... hint: we did.) □

Corollary 4.3. *If there exists a hash function* H *such that the Fiat-Shamir transform of, say, the Hamiltonicity protocol sound, the Hamiltonicity protocol cannot be ZK against malicious verifiers.*

Proof sketch. The claim is that the collapsed protocol is also ZK. This is because you could always consider a malicious verifier in the original that chooses its challenges based on the hash function H. □

Fiat-Shamir hash functions are believed to exist, at least if we allow to sample the hash function at random. In fact, you can even prove that they exist under standard assumptions for protocols like Hamiltonicity assuming that the commitments are implemented for example using encryption.

4.1 What We Actually Mean by NIZK

We won't give up on non-interactive ZK. We will settle for a relaxed notion of NIZK in *the common random string model*. In this model a trusted party samples a random string R that both the verifier and prover can use in their proofs.

Definition 4.4 (Non-Interactive ZK (NIZK) for NP)**.** *A NIZK for an NP relation R consists of efficient algorithms (P, V, S_1, S_2) with the following syntax:*

- $\pi \leftarrow P(x, w, crs)$ *gets a statement x, witness w, and string crs and outputs a proof π.*
- $b \leftarrow V(x, \pi, crs$ *gets a statement x, proof π, and string crs and outputs $b \in \{Acc, Rej\}$.*
- $(\widetilde{crs}, td) \leftarrow S_1$ *outputs a simulated $\widetilde{crs}$ and corresponding trapdoor td.*
- $\widetilde{\pi} \leftarrow S_2(x, td)$ *gets a statement x the trapdoor td and outputs a simulated proof $\widetilde{\pi}$.*

The algorithms satisfy the following properties:

- **Completeness:** *for any $(x, w) \in R$ and crs, the verifier $V(x, \pi, crs)$ will accept π generated by $P(x, w, crs)$.*
- **Soundness:** *for any malicious prover P^* and any $x \notin L$*

$$\Pr_{crs \leftarrow \{0,1\}^n} [V(x, \pi, crs) = Acc \mid \pi \leftarrow P^*(x, crs)] \leq negl(n) \ .$$

  Again, if soundness only holds against bounded provers it's called an argument *rather than a proof.*

- **Zero Knowledge:** *for any $(x, w) \in R$*

$$crs, \pi \approx \widetilde{crs}, \widetilde{\pi} \ ,$$

  where $crs \leftarrow \{0,1\}^n, \pi \leftarrow P(x, w, crs)$ and $\widetilde{crs}, td \leftarrow S_1, \widetilde{\pi} \leftarrow S_2(x, td)$.

It is instructive to think why the impossibility of one-message ZK does not hold here.

Adaptive Security. Both of the soundness and ZK requirements can be naturally strengthened to be *adaptive*. Namely, soundness holds even if the malicious prover chooses $x \notin L$ depending on the crs string. ZK holds even if x, w are chosen depending on the crs string.

Discussion. Having a common random string doesn't seem like an unreasonable assumption. Common random bits do seem we can use the stock market, sunspots, or SHA(0).

The delicate point about NIZKs is the ZK guarantee. Indeed, in this definition, we allow the simulator to generate a simulated $\widetilde{crs}$, whereas in the real world there is a different crs. Indeed, the ZK guarantee of NIZKs is different. One example of this is that interactive ZK proofs are non-transferable — having interacted with prover in a ZK proof does not give the verifier the ability to prove that statement to another verifier (that will use its own fresh coins).

Still, NIZKs do capture quite a lot, perhaps most of what we expect them to. Intuitively, they hide any information about the witness that could not be learned in the idea world where proofs are simulated.

Example: value of anonymous coins. Imagine that we have two anonymous coins represented as two commitments

$$C_0, C_1 \text{ where } C_b = Com(v_b) \text{ is a commitment to their value.}$$

and you would like to prove that C_0 and C_1 have the same value, i.e. $v_0 = v_1$. Assume you do so using a NIZK proof.

Claim 4.5. *In the real world no attacker can tell anything about the value of the coins except that they are equal. That is, for any v and v'*

$$C_0, C_1, \pi, crs \approx C'_0, C'_1, \pi', crs$$

Where $crs \leftarrow \{0,1\}^n$ (in both), $C_b \leftarrow Com(v), C'_b \leftarrow Com(v')$, π, π' are NIZKs that the two commitments are two the same value.

Proof. Let S_1, S_2 be the simulators. Then:

$$C_0, C_1, \pi, crs \overset{zk}{\approx} C_0, C_1, S_2(C_0, C_1, td), S_1 \overset{hiding}{\approx} C'_0, C'_1, S_2(C'_0, C'_1, td), S_1 \overset{zk}{\approx} C'_0, C'_1, \pi', crs$$

where now td is the trapdoor that S_1 generates.

The above holds because the efficient simulator cannot break the hiding of the commitments. □

4.2 How to Construct NIZK

We will describe one simple approach of constructing NIZKs through what is known as witness indistinguishability (WI).

Let's go back to the Fiat-Shamir version of the (repeated) Hamiltonicity protocol. We already said that it cannot be ZK, but does it still has meaningful privacy properties. Specifically we can show that it is witness indistinguishable. WI is a relaxation of ZK, which only say that if a statement has many witnesses than the verifier does not learn which witness the prover used.

Definition 4.6 (Witness Indistinguishable Interactive Proofs (WIPs) for NP)**.** *(P, V), where P, V are efficient prover and verifier is a WIP for a relation $R \in NP$ if it satisfies:*

- **Completeness**
- **Soundness**
- **Witness Indistinguishability:** *for any efficient malicious verifier V^* and any $(x, w_0), (x, w_1) \in R$*

$$View_{V^*}(\langle P(w_0), V^* \rangle (x)) \approx View_{V^*}(\langle P(w_1), V^* \rangle (x)) \ .$$

  (Note that the only difference is that we use two different witnesses.)

WI may seem significantly weaker than ZK, for example it is vacuous if we think of statements that only have a single witness. As we shall see, however, it could be very meaningful.

Claim 4.7. *The (repeated) Hamiltonicity protocol is WI.*

Proof sketch. First observe that the non-repeated protocol is WI. Why? because it is ZK against malicious verifiers, and ZK implies WI (make sure you see why).

This can be shown to imply that the repeated protocol is WI, but what is known as a hybrid argument roughly speaking, you can imagine gradually going from a proof that uses the first witness w_0 in all instances to a proof that uses w_1, in the middle the prover uses w_0 for the first $i \in \{0, \ldots, n\}$ instances and w_1 for the last $n - i$ instances. If the verifier can distinguish a proof with all w_0 from one with all w_1, then it can also distinguish some hybrid that is different only on a single instance and thus break WI for a single one. □

Claim 4.8. *The Fiat-Shamir collapse of the protocol is also WI.*

You should already known what is the proof by now...

The NIZK. So we have a truly non-interactive WI proof. How can we use it to construct NIZKs in the common random string model?

References