Zero-Knowledge

Boran Erol

March 2024

1 Zero-Knowledge Proofs

The interaction reveals nothing beyond the validity of the proposition.

The definition avoids "what is knowledge" altogether!

Here's an example of ZK using color blindness:

ZK is useful for identification, upgrading semi-honest security to malicious security,

ZK is at the right level of abstraction. It's simple enough to be studied and realized. The feasibility/limitations delinate what is attainable.

ZK is just a means to an end. There are weaker definitions that are also useful.

If you want efficient protocols that use ZK, you need to understand the inner workings of ZK. If you use ZK in a modular fashion, you'll get cool protocols, but they won't have optimal efficiency.

1.1 Definitions and Motivation

ZK proofs for all of NP is Avi Wigderson's favorite result. Here's how he explains it. Let's say you are arguing with your friend. You'd like to convince him that you are correct. However, your friend doesn't trust you. Therefore, in order to convince your friend, you have to provide information. ZK proofs are convincing without providing any knowledge!

There's a problem with ZK protocol, even 3 decades after its founding: we still don't have a good, concise language to express our proofs. We have machines interacting, noticeable, negligible probabilities, it's just all a mess. One big open question in protocol design is to come up with a easy to use language.

Definition 1 (Zero Knowledge)**.**

Definition 2 (Statistical ZK)**.** $\forall PPTV^* : \exists PPTS : \forall x \in L : \forall z$

$$S(x, z) \approx_s View_{V^*}(P(x, w), V^*(x, z))$$

Lemma 1.1. If NP $\subseteq$ SZK, the polynomial hierarchy collapses to the second level.

Proof. □

A natural follow-up question to ask is for which relaxations of SZK can we still have proofs for all languages in NP?

There are two possible paths to take:

1. Make the ZK condition computational – **CZK Proofs**. In this situation, you have undisputable soundness but the secrecy is computational.
2. Make the soundness condition computational using WI – **SZK arguments**. In this situation, you always have some lingering doubts about the soundness but the secrecy is unconditional.

1.2 SZK argument for Graph Hamiltonicity

Theorem 1.2. If statistically-hiding commitments exist, then there's a SZK argument for Graph Hamiltonicity.

Proof. We'll use the exact same protocol we used to construct a PZK protocol for HAM. □

However, the soundness of this protocol is not negligible. Whether this protocol is still SZK under parallel repetition is an open problem. To construct a SZK argument for every language in NP, we can instead use WI.

Whenever you have a relation for an NP language, this relation is fully determined by the verifier from the NP definition. However, notice that $L \in$ NP can have infinitely many different NP-Relations R_L.

1.3 ZK for QR

Let x be a QR mod N. Assume P has some w such that $x = w^2 \pmod{N}$.

1. P samples r uniformly at random from $Z_{\mathbb{N}}^*$ and sends $y = r^2$.
2. V samples b uniformly at random and sends it to P.
3. If $b = 0$, the prover sends $z = r$. V checks if $z^2 = y$.
4. If $b = 1$, the prover sends $z = wr$. V checks if $z^2 = xy$.

soundness. Assume x is not a quadratic residue. Then, if you send y that is a QR, there's no way to satisfy the condition at $b = 1$. If you send y that is not a QR, you can't satisfy $b = 0$. □

1.4 Questions and Answers

Is there a public-coin ZK im(possibility) result?

2 Proof of Knowledge

2.1 Definition and Motivation

Proving that something exists versus proving that you know something is not the same thing. Proof of knowledge tries to ensure that the prover actually knows the witness. Throughout these definitions, $n = |x|$.

Definition 3. A proof system has **knowledge soundness if error** κ if there exists a PPT κ such that for every prover P^* that convinces V of x with probability $\epsilon > \kappa$, $K^{P^*(\cdot)}$ outputs w such that $(x, w) \in R_L$ with probability at least $\epsilon(n) - \kappa(n)$.

Here's an equivalent definition:

Definition 4. A proof system has **knowledge soundness if error** κ if there exists a κ such that for every prover P^* that convinces V of x with probability $\epsilon > \kappa$, $K^{P^*(\cdot)}$ outputs w such that $(x, w) \in R_L$ and runs in time $\frac{n}{\epsilon(n)-\kappa(n)}$

Here's a stronger definition:

Definition 5 (strong proof of knowledge)**.** A proof system has **strong knowledge soundness** if there's a negligible function μ and a PPT K such that for every prover P^* that convinces V of x with probability $\epsilon > \mu$, $K^{P^*(\cdot)}$ outputs w such that $(x, w) \in R_L$ with probability at least $1 - \mu(n)$.

Lemma 2.1. Knowledge soundness implies soundness.

Notice that knowledge soundness is a property of the verifier.

Lemma 2.2. Sequential composition reduces knowledge error exponentially.

In fact, in PoK, we can replace exponentially small error with zero error.

Lemma 2.3. Let (P, V) be an interactive protocol with exponentially small knowledge soundness error. Then, there's an interactive protocol (P', V') with zero knowledge soundness error.

Proof. The idea is to run the knowledge extractor in parallel with a brute-force search on the witness.

□

2.2 Constructing ZKPOKs

Consider the regular ZK protocol for QR.

Just like simulations, we're going to rewind the prover using knowledge extractors. This makes sense: if we could extract the knowledge without rewinding, the proof would no longer be ZK!

proof of knowledge soundness. We assume WLOG that P^* has its randomness hardwired and is deterministic. Let $\kappa = \frac{1}{2}$.

Here's the knowledge extractor:

1. K invokes P^* and receives some y.
2. K sends P^* the query $b = 0$ and receives z_0.
3. K rewinds and sends P^* the query $b = 1$ and receives z_1.
4. K outputs $w = z_1/z_0 \pmod N$.

If P^* convinces V with probability more than κ, since it is deterministic, it answers both queries truthfully. We thus conclude the proof. □

Let's now consider the regular ZK protocol for HAM.

proof of knowledge soundness. We assume WLOG that P^* has its randomness hardwired and is deterministic. Let $\kappa = \frac{1}{2}$.

Here's the knowledge extractor:

1. K invokes P^* and receives a commitment c.
2. K sends P^* the query $b = 0$ and receives a cycle with w.
3. K rewinds and sends P^* the query $b = 1$ and receives $\pi, \tilde{G}$.
4. K outputs the cycle in the graph using **obvious, but finish still.**

If P^* convinces V with probability more than κ, since it is deterministic, it answers both queries truthfully. We thus conclude the proof. □

These examples might lead you to think that proving ZKPOK is easy, but these examples are only because the probability analysis is very clean. In general, with arbitrary probabilities, proving ZKPOK is extremely annoying.

Sigma protocols try to solve this issue.

The protocols we have constructed so far have high error. Let's now reduce the error to negligible using sequential composition.

Let $n = |x|$.

Theorem 2.4. Running HAM n times sequentially is strong ZKPOK.

Proof. Consider the following extractor strategy. □

In fact, it can be proven that it's impossible to get a strong ZKPOK with negligible error without sequential composition.

2.3 Applications of ZKPOK

In practice, the alternative defintion of ZKPOK has an issue. How do we know when K is expected polynomial time? Depending on $\epsilon(n)$, K might run in expected exponential time.

A classic use of ZKPOK is the following:

Within a protocol, the prover proves the proof. To prove security, the simulator needs the witness, so it needs to run the knowledge extractor.

Usually, when using ZKPOKs in proofs of security, the simulator plays the role of the verifier and interacts with the prover. If the verifier rejects, the simulator just halts. If the verifier accepts, then the simulator has to extract the witness.

What is the expected running time of the this simulator?

The probability that P convinces V is $\epsilon(n)$. Notice that we have no clue what $\epsilon(n)$ is. For simplicity, let's assume $\kappa(n) = 0$. Then, the expected running time of the simulator is

$$(1 - \epsilon(n)) \cdot poly(n) + \epsilon(n) \cdot \frac{poly(n)}{\epsilon(n)} = poly(n)$$

Let's now assume that $\kappa(n)$ is negligible. Then,

$$(1 - \epsilon(n)) \cdot poly(n) + \epsilon(n) \cdot \frac{poly(n)}{\epsilon(n) - \kappa(n)} = poly(n) + \frac{\epsilon(n)}{\epsilon(n) - \kappa(n)}$$

Notice that this is not necessarily negligible!

Witness-extended emulation

Lemma 2.5. If (P, v) is a ZKPOK, then there exists a witness extended emulator for (P, V).

In MPC, we can also formalize an ideal ZK functionality:

$$\mathcal{F}_{zk}((x, w), x) = (\lambda, R(x, w))$$

Lemma 2.6. If (P, V) is a ZKPOK, then it securely computes the ideal ZK functionality.

2.3.1 Applications of ZKPOK

1. A ZK proof for QNR
2. Non-Oblivious Encryption
3. Prove a property of statistically commmited value.
4. Identification Schemes

2.3.2 A ZK proof for QNR

1. V samples $b \overset{\$}{\leftarrow} \{0, 1\}$, $y \overset{\$}{\leftarrow} \mathbb{Z}_N$.
2.

POK allows you to get implicit statements and turn it into explicit knowledge.

2.3.3 Non-Oblivious Encryption

Provide an encryption and prove that you know what's encrypted.

Motivation:

1. Prevent copying (e.g., in auction).
2. Guarantee non-malleability (didn't take a previous ciphertext and maul).

2.3.4 Prove Property of Statistical Commited Value

Consider a statistically-hiding commitment scheme. Because hiding is statistical, a commitment value c can be a commitment to any message.

Suppose the you're committing to your identity. In this scenario, it's often useful to prove a property of your identity (the range of your age). Here, since existence is guaranteed, ZK proofs are useless. Instead, the committer can prove that they know a decommitment to a message with a certain property.

3 Constant-Round CZK Proofs for NP

Goldreich and Krawczyk proved that languages with constant-round, public-coin BBZK protocols are in BPP. In other words, constant-round, public-coin BBZK protocols are useless.

We'd like to construct a CZK proof with negligible soundness and constant round complexity. Recall that parallel repetition achieves this, but constructing a simulator for parallel repetition is an open problem.

We'll need to address:

1. Malleability
2. Aborts in simulation

We'll require statistically hiding commitments for the verifier and statistically binding commitments for the prover. Essentially, because the prover is unbounded, we're ensuring security against the prover in both cases.

There's a technical issue:

V^* might decide to decommit depending on the commitment it receives.

Here's a naive attempt to construct a simulator:

1. Commit to garbage values.
2. If $V^*(c) = ABORT$, halt.
3. If $V^*(c) \neq ABORT$, rewind and adjust the garbage based on the decommitment received. Repeat this process until V^* doesn't abort.

Let

$$s(n) = \Pr[V^* \text{ doesn't abort given garbage commitments}]$$

$$t(n) = \Pr[V^* \text{doesn't abort given valid commitments}]$$

Notice that the expected number of repetition for the simulator in the third step is $s(n)/t(n)$. By the computational hiding of the commitment scheme, the difference between $s(n)$ and $t(n)$ is negligible. However, this isn't enough! If $s(n) = 2^{-n}$ and $t(n) = 2^{-2n}$, we still get exponentially many repetitions.

In 1996, Goldreich and Kahan show how to get around this issue. At a high level, they fix this problem by estimating the probability that V^* aborts. The analysis is complicated. Alon Rosen thinks its overkill for such a simple protocol.

In 2004, Rosen provided an alternative solution with a much simpler analysis at the cost of adding two rounds. The idea of the protocol is to make V^* commit in a way that allows the simulator to extract the value of b before c is even sent.

In order to achieve this, the verifier creates a secret share of b and sends the secret shares to V^* before V^* commits. The simulator can rewind the verifier to learn b before it commits to c by learning both shares. In the real protocol, V^* learns nothing by the security of the secret-sharing scheme.

Notice that this doesn't seem to be a POK. The extractor can't rewind and change its b, since it has committed to it. Lindell, in 2013, solves this issue in "A note on the constant-round ZKPOKs". The analysis of this protocol is even more complicated than GK96.

4 Witness Indistinguishability

WI is a relaxation of ZK for languages in NP. It was introduced by Feige and Shamir in 1990.

Definition 6. We say that a protocol π is **witness indistinguishable** if $\forall V^*(x,z) : \forall (x, z, w_1, w_2)$:

$$View_{V^*}(P(x,w_1), V^*(x,z)) \approx_c View_{V^*}(P(x,w_2), V^*(x,z))$$

Definition 7. We say that a protocol π is **witness independent** if $\forall V^*(x,z) : \forall (x, z, w_1, w_2)$:

$$View_{V^*}(P(x,w_1), V^*(x,z)) \approx_s View_{V^*}(P(x,w_2), V^*(x,z))$$

Lemma 4.1. ZK implies WI.

Proof.

$$View_{V^*}(P(x,w_1), V^*(x,z)) \approx_c S^*(x,z) \approx_c View_{V^*}(P(x,w_2), V^*(x,z))$$

□

Corollary 4.1.1. If statistically-binding commitments exist, then every language in NP has a witness indistinguishable proof.

Corollary 4.1.2. If statistically-hiding commitments exist, then every language in NP has a witness independent argument.

Definition 8. We say that an interactive protocol (P, V) is **unbounded simulation** if $\forall PPTV^*$: $\exists S : \forall x \in L : \forall w \in R_L(x) : \forall z$

$$View_{V^*}(P(x,w), V^*(x,z)) \approx_c S(x)$$

Lemma 4.2. (P, V) has an unbounded simulation if and only if it's witness indistinguishable.

Proof. Assume (P, V) has an unbounded simulation. Then,

$$View_{V^*}(P(x,w_1), V^*(x,z)) \approx_c S(x) \approx_c View_{V^*}(P(x,w_2), V^*(x,z))$$

Conversely, ... □

Notice that WI is useless for unique witness NP languages. You can reveal the witness and still satisfy the definition. Like ZK, WI is a property of the prover.

Notice that the protocol can leak information about the set of all witnesses.

If P disregards the auxiliary input, WI is trivial. Therefore, for exponential-time provers, WI is irrelevant since P can just ignore the auxiliary input and brute-force search the witness. Since we require the prover to be polynomial time, we now have **computational soundness** instead of unconditional soundness. Because of this, these are also called arguments instead of proofs.

Lemma 4.3. WI is closed under parallel repetition.

Proof. Without loss of generality, we prove the theorem for executing two protocols in parallel.

□

4.1 Open Problems

1. Can Strong WI be achieved non-interactively?
2.

5 Fiat-Shamir Heuristic (Transform)

6 NIZKs

6.1 Definitions and Motivation

ZK proofs seem to depend on 3 crucial ingredients:

1. Interaction
2. Secret Random Coins of the Verifier
3. Computational Hardness Assumptions

NIZKs try to remove interaction from the picture. They were first introduced by Blum, Feldman, Micali in 1988 in the CRS model.

Definition 9. A pair of PPT algorithms (P, V) is a **NIZK proof system** if

Notice that the simulator doesn't get access to the witness whereas the prover does.

NIZKs can also be used to produce CCA2 secure PKE systems.

NIZKs seem related to constant-round ZK protocols. However, NIZKs require a shared reference string. If you can construct a secure-coin flipping protocol when executed in parallel, this gives you a constant-round ZK proof from a NIZK. The Goldreich Kahan 1996 paper can be seen as constructing a secure coin-flipping protocol under parallel repetition.

Blum, Santis, Micali, Persiano improved on this later.

We're going to assume a common **random** string, which we're going to denote with crs.

Rafi assumes that the prover is polynomial time. There are constructions where the prover is unbounded, but Rafi doesn't care.

The protocol is going to be zero-knowledge if there's a simulator S that can produce crs' and Π' that is indistinguishable from crs and Π.

Intuitively, the simulator is going to encode a trapdoor into crs' which it will use to convince the verifier, but the prover won't be able do such a thing since crs in the actual interaction is fully random.

This is similar to the idea behind witness indistinguishability.

Rafi has a variation on the definition where the simulators crs with the trapdoor can be used in the original protocol without the prover being able to crack the trapdoor.

Feige, Lapidot, Shamir (FLS)

One-way trapdoor permutations.

Let $\mathcal{F}$ be a family of OWPs.

The prover can sample f, f^{-1} using its own private randomness. Using this OWP, the prover can slowly reveal the hardcore bits of this OWP to the verifier.

Intuitively, the prover has some wiggle room at committing to certain strings, but it can't control all the randomness it's committing to.

First of all, through magic, P commits to a cycle graph G'.

Then, P sends V a permutation Π such that the cycle in G' and the cycle in $\Pi(G)$ overlap. P then opens all non-edges of $\Pi(G)$ such that it overlaps with the non-edges of the original graph G'.

We're showing that G is isomorphic to G' without telling V what G' is apart from the fact that

In general, (in 2004) it's unknown how to get adaptive NIZK from non-adaptive NIZK. However, it is possible to get from non-adaptive soundness to adaptive soundness.

6.2 Adaptive Soundness from Non-Adaptive Soundness

Let $Bad_k = \bar{L} \cap \{0,1\}^k$.

The intuition is as follows: for length k strings, repeat the proof $2k$ times to achieve adaptive soundness.

Using this, for any given $x \in Bad_k$, we'll prove that only a fraction of 2^{-2k} random strings allow the prover to cheat. Then, summing over all possible bad strings and using a union bound, only a fraction of 2^{-k+1} strings allow the prover to cheat.

6.3 NIZK in CRS from NIZK in Hidden-Bits using Trapdoor OWPs

In the Hidden-Bits model, the prover is given some random string r with $|r| = n$. The prover, along with its proof π, sends $I \subseteq [n]$ to the verifier. Then, the verifier gets all r_i's with $i \in I$ and learns nothing about the other random bits.

Theorem 6.1. Assuming the existence of trapdoor OWPs, given a NIZK proof system in the hidden-bits model (P', V'), we can construct a NIZK proof system (P, V) in the CRS model.

Proof. Let (P', V') a be NIZK proof system in the hidden-bits model. Let k be the security parameter for (P', V').

Let $\mathcal{F}$ be a trapdoor OWP family.

Here are some assumptions we make:

1. The set of "legal" (f, f^{-1}) is efficiently decidable. In other words, given (f, f^{-1}), you can efficiently decide whether it was output by **Gen**.
2. For randomly chosen x, $h(x)$ is a uniformly random bit.

Here are some simplifying assumptions without loss of generality:

1. The random string generated by $Gen(1^k)$ has length k. Thus, there are at most 2^k distinct f's.
2. The soundness error of (P', V') is at most 2^{-2k}. We can ensure that this condition holds by running $2k$ copies of (P', V') in parallel. For a more substantive discussion, review the argument for upgrading non-adaptive soundness to adaptive soundness.
3.

Let n be the length of the random string r' given to P' in the original NIZK system.

Let r be a random string of length nk.

Intuitively, P will just use the hardcore bit of the trapdoor OWP f to get n hidden random bits.

1.

The completeness of the new protocol is immediate.

Let's now prove the soundness of the new protocol.

Fix (f, f^{-1}). Then, for any uniformly random r, r' will also be uniformly distributed.

Here, it is crucial that we're fixing (f, f^{-1}) **before** considering a random r. If (f, f^{-1}) is dependent on r, our claim definitely doesn't hold

Then, by the soundness of (P', V'), we have that

$$\Pr[P^* \text{ can cheat using } f] \leq 2^{-2k}$$

Recall that there are at most 2^k choices of f. Therefore,

$$\Pr[P^* \text{ can cheat using ANY } f] \leq 2^k \cdot 2^{-2k} = 2^{-k}$$

Let's now prove that the new protocol is ZK. Let Sim' be the simulator for (P', V').

Here's how we construct $Sim(1^k, x)$:

1. Run $Sim'(1^k, x)$ to obtain (r'_I, π, I).
2. Run $Gen(1^k)$ to obtain (f, f^{-1}).
3. If $i \in I$, set $r_i = f(z_i)$ where $h(z_i) = r'_i$ and $z_i \xleftarrow{\$} \{0,1\}^k$.

4. If $i \notin I$, set $r_i \overset{\$}{\leftarrow} \{0,1\}^k$.

$\square$

6.4 NIZK for Graph Hamiltonicity from Trapdoor OWPs in the CRS Model (Feige, Lapidot, Shamir)

7 NIWIs