Perfect and Computational Zero-Knowledge Proof

January 26, 2022

1 Perfect Zero-Knowledge Proof

The definition of perfct zero-knowledge is below.

Definition:

Let (P, V) be an interactive proof system for some language L. We say that (P,V) is perfect zero-knowledge if for every probabilistic polynomial-time interactive machine V there exists a probabilistic polynomial-time algorithm M* such that for every $x \in L$ the following two conditions holds:*

1. *With probability at most 1/2, on input x, machine M* outputs a special symbol denoted $\perp$ (i.e., Pr[M*(x)=$\perp$] $\leq$ 1/2).*
2. *Let m*(x) be a random variable describing the distribution of M*(x) conditioned on M*(x) $\neq \perp$ (i.e., Pr[m*(x)=a] = Pr[M*(x)=a|M*(x) $\neq \perp$]). Then the following random variables are identically distributed:*
   - $< P, V^* > (x)$. i.e., the output of interactive machine V^* after interacting with the interactive machine P on common input x.
   - $m^*(x)$. i.e.,the output of machine M^* on input x.

Machine M is called a perfect simulator for the interaction of V* with P.*

What is simulator and why we need it? In the zero-knowledge proof system, we know that the Verifier could not get any "new knowledge" from the Prover, this is the definition of the ZKP, and which means that, for a language L is zero-knowledge, whatever can be computed after interacting with Prover on input $x \in L$ can also be computed from x by Verifier. Considering that, we could use a simulator to simulate the interactive process. Although the simulator does not having access to the Prover, it is still able to simulate the interaction of Verifier with Prover, since the Verifier does not gain any knowledge from Prover.

It is known that $\mathcal{BPP} \subseteq \mathcal{IP}$ since that languages in $\mathcal{BPP}$ could be viewed as each having a Verifier that decides on membership[1] without any interaction. Note that every language in $\mathcal{BPP}$ has a perfect zero-knowledge proof system in which the prover does nothing and the verifier checks by itself whether to accept or reject the common input. To demonstrate the zero-knowledge of this "dummy Prover", one can present for every Verifier a simulator that is essentially identical to Verifier.

[1]The membership here is related to the definition of Interactive Proof System, which is, given a string n, the prover represents the proof that n is a **member** of some language and the verifier checks whether that proof is correct.

2 Computational Zero-Knowledge Proof

The word "perfect", in some kind, means "idealistic" or "unrealistic". For the practical purpose, there is no need to be "perfect simulate" the output of Verifier after interacting with Prover. Thus, it suffices to generate a probability distribution that is computationally indistinguishable from the output of Verifier after interacting with Prover. This kind of relaxation is consistent with our principle: *for a language L is zero-knowledge, whatever can be computed after interacting with Prover on input $x \in L$ can also be computed from x by Verifier.*

Before giving the definition of Computational Zero-Knowledge Proof, we give a brief about the computationally indistinguishable. Here we consider ensembles indexed by strings from a language L. We say that the ensembles $\{R_x\}_{x\in L}$, $\{S_x\}_{x\in L}$ are computationally indistinguishable if for **every** probabilistic polynomial-time algorithm D, for **every** polynomial $p()$ and for **all** sufficiently long $x \in L$, it holds that:

$$|Pr[D(x, R_x) = 1] - Pr[D(x, S_x) = 1]| \leq \frac{1}{p(|x|)}$$

Definition:

Let (P, V) be an interactive proof system for some language L. We say that (P,V) is computational zero-knowledge (or zero-knowledge) if for every probabilistic polynomial-time interactive machine V there exists a probabilistic polynomial-time algorithm M* such that the following two ensembles are computationally indistinguishable:*

- $\{< P, V^* > (x)\}_{x\in L}$. i.e., the output of interactive machine V^* after interacting with the interactive machine P on common input x.
- $\{m^*(x)\}_{x\in L}$. i.e.,the output of machine M^* on input x.

Machine M is called a perfect simulator for the interaction of V* with P.*

3 An Alternative Formulation of Zero-Knowledge

The alternative formulation of zero-knowledge considers the view of the Verifier, rather than only the output of the Verifier after interacting with the Prover. The view of the Verifier is the entire sequence of the local configurations of the Verifier during an interaction. Thus, it suffices to only consider the random tape of the Verifier and the message sequence during the interaction since the local configuration could be deduced by these two elements.

Definition:

Let (P, V) be an interactive proof system for some language L. We denote by $view^P_{V^*}(x)$ a random variable describing the content of random tape of V^* and messages V^* receives from P during a joint computation on common input x. We say that (P,V) is zero-knowledge if for every probabilistic polynomial-time interactive machine V there exists a probabilistic polynomial-time algorithm M* such that the following two ensembles, $\{view^P_{V^*(x)}\}_{x\in L}$ and $\{M^*(x)\}_{x\in L}$ are computationally indistinguishable.*

4 *Honest-Verifier Zero-Knowledge

The reason that we discuss the honest-verifier zero-knowledge here is related to the **simulator**. As we all know, the idea behind zero-knowledge is that the communication between the Prover and Verifier does not reveal any information about the secret. So there must exist a simulator, now knowing the secret, could simulate the communication process. Also, if the simulator wants, the simulator could falsify the communication, and a falsified communication and real communication must not be distinguishable.

So, the honest Verifier zero-knowledge means that we can expect the Verifier behaves honestly. In normal cases, the Verifier could be malicious.

Definition:

Let (P, V) be an interactive proof system for some language L. We denote by $view_V^P(x)$ a random variable describing the content of random tape of V and messages V receives from P during a joint computation on common input x. We say that (P, V) is honest-verifier zero-knowledge if there exists a probabilistic polynomial-time algorithm M such that the following two ensembles, $\{view_V^P(x)\}_{x\in L}$ and $\{M(x)\}_{x\in L}$ are computationally indistinguishable.