Zero-Knowledge Proofs

Honesty in the Digital World

In the analogue world, there exists devices called lie detectors. Their efficiency is based on a premise that when a person is lying, they exhibit signs such as sweating and an increased heart rate, which can be detected.

The field of zero-knowledge proofs is an attempt to build a protocol that behaves like a lie detector, but in the digital realm. Suppose Alice wishes to prove to Bob that she is rich. She could send a bank statement to Bob to prove this; however, by doing so, Bob learns not only the fact that Alice is rich, but other potentially sensitive data such as her income sources and spending habits. A zero-knowledge proof would allow Bob to learn only of the facts that he needs to learn, and nothing else.

Definitions

The founding paper of this field was TBD. They gave three criteria a proof system should satisfy to be a zero-knowledge proof system. We will define a "proof system" more carefully later, but for now we can think of it as a system for a *prover* to convince a *verifier* of the truth of some *statement*.

1. Completeness. If the statement is true, then the prover can convince the verifier.
2. Soundness. If the statement is false (i.e. the prover is "cheating"), the prover cannot convince the verifier.
3. Zero-Knowledge. The verifier, even if cheating, does not learn any information except for the validity of the statement.

Applications

TBD. digital signature, election fraud, mix-nets, verifiable outsourced computation, ring/group signatures, zerocoin, malicious to honest-but-curious, lie prevention, proof-carrying data

Zero-knowledge Proof Systems

First, we revisit and try to make more precise our definition of a zero-knowledge proof system from section TBD. To do this, we first review one definition of the complexity class NP (nondeterministic polynomial-time); it turns out that the definition of an interactive proof system will be a modification to the proof systems used in NP.

TBD.

A Zero-Knowledge Proof System for Graph Isomorphism

Let us define a proof system for graph isomorphism. For concreteness, we encode our graphs by labelling the vertices with consecutive natural numbers, and giving the graph as an edgelist. Hence, the following two graphs (which happen to be isomorphic) are encoded as TBD.

The decisional graph isomorphism problem is: given two graphs G_0 and G_1, determine if G_0 and G_1 are isomorphic. It is easy to see that this problem is in NP: the witness is an isomorphism between the graphs, or more concretely a permutation of the vertices, encoded as a list of $|V|$ integers.

Now let us describe an interactive proof system for this problem. The prover is trying to prove that G_0 and G_1 are isomorphic, and furthermore, that he possesses the isomorphism $G_0 \leftrightarrow G_1$ denoted by w. The interactive proof consists of r *rounds*, and each round consists of three *steps*. In the first step, the prover randomly permutes the vertices of G_0 to obtain a graph H, and sends H over. In the second step, the verifier sends a single bit $b \in \{0, 1\}$. In the third step, the prover sends an isomorphism between G_b and H.

The verifier rejects if, at any round, he fails to verify the isomorphism, and accepts if after r rounds he has not rejected.

Exercise: show that this proof system is complete, sound and zero-knowledge.

Completeness. We will show that in the case that the statement is true (the prover really possesses a witness w), he will convince an honest verifier. We can show this for every round. In the first step, let the isomorphism between G_0 and H be h. Suppose the verifier sends $b = 0$; then the prover can simply send h. Otherwise, if the verifier sends $b = 1$, then the prover can simply send $h \circ w$, which is an isomorphism between H and G_1.

Soundness: Suppose the graphs are not isomorphic, and the cheating verifier is trying to convince an honest prover. The verifier sends a graph H which can be isomorphic to G_0 or G_1, but not both. Let H be not isomorphic to G_c for $c \in \{0, 1\}$. With probability at least $\frac{1}{2}$ the verifier will choose $b = c$ and the prover cannot possibly produce an isomorphism between G_b and H in the last step.

With r rounds, the chance that a cheating prover is not caught in 2^{-r}.

Zero-Knowledge: We assume an arbitrary verifier, who might be crafting his queries to try to learn something about w, interacts with an honest prover. It is clear that in the case that the verifier is honest, he learns nothing about w, since at each round the verifier learns either

1. A random isomorphism $G_0 \sim H$, or
2. w composed with a random isomorphism $G_0 \sim H$.

This is equivalent to sampling from the space of random isomorphisms of G_0 r times, where each sample is independent, and this distribution does not depend on w, the protocol is zero-knowledge.

However, this proof is not sufficient; there are protocols where an honest verifier learns nothing, but a dishonest one does! TBD. Reference: Theorem 2 of GMW.

Removing Interactivity

The proof system presented above is interactive, that is, the messages that a verifier and prover send at various steps depend on previous messages. Example of a non-public-coin interactive proof system? It is also **public-coin**, that is, an honest verifier sends uniformly random challenges. Example of an interactive, non-public-coin proof system?

In a public-coin protocol, if there is a randomness source (e.g. the NIST random beacon) trusted by both parties, then the verifier can use that instead of generating his own randomness. If the challenges are big enough, the Fiat-Shamir heuristic allows one to derive a non-interactive proof system from a public-coin interactive proof system under some conditions. The heuristic is to use a collision-resistant hash function to deterministically generate the random challenges. The prover computes a transcript where the verifier's challenges are replaced by the output of the hash function.

Statements

The protocol above allows a prover to convince a verifier about statements of the form "this graph is isomorphic to that one" in zero-knowledge. Now, we try to formalize the notion of a "statement" so we can generalize it.

TBD.

Arguments vs Proofs

AC-SAT

Efficient ZKPs based on the Discrete Log Problem

Linear-Time ZKPs for AC-SAT

Pairing-Based SNARKs