**Zero-Knowledge Protocols**

A proof of Knowledge is a protocol between two parties: the Prover and the Verifier. Both parties are aware of predicate $Q$ for which the prover possesses some witness $w$, and attempts to convince the Verifier that it has knowledge of $w$ by correctly responding to queries which require knowledge of $w$ to answer. These two parties operates in following rounds:

The general setup is as follows:

- the predicate $Q$. The predicate $Q$ is assumed to be polynomial-time computable: i.e., given $w'$ one can efficiently test $Q(w')$.
- The Prover has input $Q$, $w$ so that $Q(w) = true$. She wishes to prove knowledge of such $w$. Note that there can be many such $w$’s that satisfy $Q$.
- Verifier has input $Q$.

The informal requirements for the above setup to make sense are as follows:

- Given $Q$, it is “hard” to find a $w$ such that $Q(w) = true$.
- The prover is reluctant to reveal $w$ (otherwise the trivial solution of the prover transmitting $w$ to the verifier is acceptable).
- $Q$ is efficiently checkable.

Let us see two simple toy examples of such procedures to get a feeling of how they work.

The first one relates to the game “Where’s Waldo.” In this game there is a large board with different characters on it, one of them being “Waldo.” Alice claims she knows the exact location of the board where Waldo is located. In this setting, $Q$ is comprised of the board and the witness $w$ is the $x, y$ coordinate of the Waldo character. Specifically $Q$ corresponds to the procedure that receives two coordinates $x, y$ and returns true if Waldo is present in the cell of the board that is marked by the $x, y$ coordinates. Alice wants to prove to Bob that she knows Waldo’s location without revealing too much information about the location of Waldo in the board? A possible protocol solution is as follows: Alice covers the board with another white board whose length and width are at least twice as large compared to the original board. Moreover there is a small hole on the white board. In order for Alice to show that she knows the location of Waldo, she moves the white board so that the hole reveals “Waldo.”

Observe that due to the fact that the white cover-board is twice as large as the board that contains the characters and the fact that the hole in the white board is very small (merely large enough for Waldo’s head to appear through it) the above protocol will not leak much information about Alice’s secret witness (the coordinates of Waldo).
You can play the “Where’s Waldo” game in http://www.ebaumsworld.com/waldo.shtml

Another toy example of a proof of knowledge is the magic word door problem.

![Figure 1: Magic Words Door](image)

The point here is that Alice knows the magic words (secret) to open the secret door between 3 and 4, but she doesn’t want to reveal her secret. So to prove Alice knows the magic word that opens the door, Alice and Bob play the following game:

1. Bob stands at point 1
2. Alice walks all the way into cave, either 3 or 4.
3. After Alice disappeared into the cave, Bob walks to point 2.
4. Bob shouts to Alice asking either to come out of the left passage or come out of the right passage.
5. Alice complies, using the magic words to open secret door if she has to
6. Alice and Bob repeat step 1-5 $t$ times

The above game nicely illustrates that a probabilistic procedure can be used to design such protocols. In particular after $t$ repetitions of the above game, Bob can be convinced that with probability $1 - \frac{1}{2^t}$ Alice knows the magic words.

**Three-Move zero-knowledge protocols**

In this lecture we will concentrate on “3-move” protocols. The communication flows between the two players are restricted to be as follows:

- $P \rightarrow V$: commitment
- $P \leftarrow V$: challenge
- $P \rightarrow V$: response

Moreover, we will be interested in protocols that the challenge submitted by the verifier in the second step of the above protocol is a random string of a certain size (cf. the choice of Bob in the magic word protocol that challenged Alice to either appear from the left or from the
Formal Requirements

Let us now come to the task of formalizing the required properties for a proof of knowledge:

Suppose that we have a three-move protocol between the Prover and Verifier. We want to satisfy the following three properties:

1. Completeness: assuming that both prover and verifier are honest (i.e., they are following the protocol specifications faithfully) then the protocol succeeds (i.e., the verifier accepts the proof) with sufficiently large probability. The definition of “sufficiently large” can vary. Preferably we would like that the probability of success is very close to 1.

Completeness is a sort of correctness condition: it asserts that the protocol that we have at hand indeed allows a honest prover that possesses a witness to convince the honest verifier.

2. Soundness: An interactive proof is sound if there is an algorithm \( M \) with the following properties:
   - \( M \) is polynomial time.
   - Given any (possibly dishonest) prover that with non-negligible probability can successfully execute the protocol with the honest verifier, then \( M \) can use the prover in a black-box fashion to extract the witness.

Soundness is intuitively a property that guarantees that any convincing prover that interacts with the honest verifier must know a witness.

3. Zero-Knowledge (ZK) Property: There exists a simulator (an algorithm) that can simulate (upon input of the assertion to be proven, but without interacting with the real prover) an execution of the protocol that for an outside observer cannot be distinguished from an execution of the protocol with the real prover.

The Zero-knowledge is intuitively a property that guarantees that the verifier cannot extract any knowledge from its interaction with the honest prover.

We will concentrate on the case of “honest-verifier zero-knowledge”. In this case the verifier is assumed to be faithfully executing the protocol specifications (thus, it is honest) but it may execute additional computation on the side to extract knowledge about the witness. This scenario sometimes is called “semi-honest.” Even though this is a relaxation from a full-fledged formalization of the zero-knowledge requirement it is still very useful (and much easier to be attained). Moreover in the case of 3-move protocols that we concentrate here, we can force a verifier to behave in a (semi-)honest way.

Schnorr Identification Protocol

Next we connect the protocol to the identification protocol that we talked in previous lecture. Suppose \( g \) is a generator of \( \langle g \rangle \) in \( \mathbb{Z}_p^* \), say \( \langle g \rangle = QR(p) \) and let \( h = g^x \) be a public value.

Completeness:

Observe that if both the prover and the verifier are honest, it holds that \( g^x = g^{w+cx} = g^w (g^x)^c = y h^c \) with probability 1. Thus, the honest prover will convince the honest verifier always.

Soundness: Let \( P \) be a prover that convinces the honest verifier with probability \( \alpha \). Recall that \( P \) is a probabilistic program that runs in two stages:

- \( P(\text{first}, g, h, p, q) \rightarrow \langle y, \text{aux(\text{internal})} \rangle \)
Suppose we are capable of generating two accepting conversations from P that have the same first move but with different challenges (same first move, different second move).

\[
\langle y, c, S \rangle \quad \langle y', c', S' \rangle
\]

Given that both are accepting it holds that,

\[
y^S = yhc^c \quad y'^{S'} = yhc'^{c'} \quad c \neq c'
\]

and as a result

\[
g^{S - S'} = \frac{h^c}{h^{c'}} \iff g^{S - S'} = h^{c - c'} \iff h = g^{\frac{S - S'}{c - c'}}
\]

Note that \((c - c')^{-1}(\text{mod}q)\) exists and can be found.

As a result we can reconstruct the witness \(x\) as \(\frac{S - S'}{c - c'}(\text{mod}q)\).

We will return to the coming lecture to the discussion on how we can use P to obtain two accepting conversations with the same first move.

H.V.(Honest Verifier)Z.K. Simulator:

We present below a simulator that given the public-predicate information \((g, p, q, h)\) is capable of simulating accepting conversations between a honest prover and a (semi-)honest verier.

The simulator: choose \(c \leftarrow_R Z_q\) and \(S \leftarrow_R Z_q\) and output \(\langle g^S h^{-c}, c, S \rangle\).

Claim: this is indistinguishable from accepting conversation between honest prover/honest Verifier.

Verify this fact by yourself by considering the probability distribution \(D_{\text{real}}\) that corresponds to the following “box”

\[
\langle g^w, c, w + rc(\text{mod}q) \rangle \quad w \leftarrow_R Z_q, c \leftarrow_R Z_q
\]

and the distribution \(D_{\text{simul}}\) that correspond to the following “box”

\[
\langle g^s h^{-c}, c, s \rangle \quad s \leftarrow_R Z_q, c \leftarrow_R Z_q
\]