**CS 276 – Cryptography** Oct 6, 2014

# Lecture 10: Non-Interactive Zero-Knowledge (NIZK) and the Hidden-Bit Model

*Instructor: Sanjam Garg* *Scribe: Preetum Nakkiran*

## 1 NIZK Proof Systems

We now consider a different class of Zero-Knowledge proof systems, where no interaction is required: The Prover simply sends one message to the Verifier, and the Verifier either accepts or rejects. Clearly for this class to be interesting (not collapse to P), we must have some additional structure: both the Prover and Verifier additionally have access to a random public string $\sigma$ (trusted to be random by both). For example, they could derive $\sigma$ by looking at sunspot patterns. Formally:

**Definition 1** *A* NIZK proof system *for input $x$ in language $L$, with witness $\omega$, is a set of efficient (PPT) algorithms $(K, P, V)$ such that:*

1. *Key Generation: $\sigma \leftarrow K(1^k)$ generates the random public string.*
2. *Prover: $\pi \leftarrow P(\sigma, x, \omega)$ produces the proof.*
3. *Verifier: $V(\sigma, x, \pi)$ outputs $\{0, 1\}$ to accept/reject the proof.*

*Which satisfy the completeness, soundness, and zero-knowledge properties below.*

Note: We will assume throughout that $x$ is of polynomially-bounded length, ie we are considering the language $L \cap \{0,1\}^{P(k)}$.

**Definition 2** Completeness. *$\forall x \in L, \omega \in R_L(x)$:*

$$\Pr[\sigma \leftarrow K(1^k); \pi \leftarrow P(\sigma, x, \omega) : V(\sigma, x, \pi) = 1] = 1$$

**Definition 3** Non-Adaptive Soundness. *$\forall x \notin L$:*

$$\Pr_{\sigma}[\sigma \leftarrow K(1^k); \exists\ \pi\ s.t.\ V(\sigma, x, \pi) = 1] = neg(k)$$

The above definition is "non-adaptive", because it does not allow a cheating prover to decide which statement to prove after seeing the randomness $\sigma$. We may also consider the stronger notion of "adaptive soundness", where the prover is allowed to decide $x$ after seeing $\sigma$:

**Definition 4** Adaptive Soundness.

$$\Pr_{\sigma}[\sigma \leftarrow K(1^k); \exists\ (x, \pi)\ s.t.\ V(\sigma, x, \pi) = 1] = neg(k)$$

**Definition 5** (Non-Adaptive) Zero-Knowledge. *The exists a PPT simulator $S$ such that $\forall x \in L, \omega \in R_L(x)$, the two distributions are computationally indistinguishable:*

1. *$\sigma \leftarrow K$*

2. $\pi \leftarrow P(\sigma, x, \omega)$

3. *Output* $(\sigma, \pi)$

*And the simulator output:*

1. $(\sigma, \pi) \leftarrow S(1^k, x)$

2. *Output* $(\sigma, \pi)$

That is, the simulator is allowed to generate the distribution of randomness $\sigma$ together with $\pi$. Note that if we did not allow $S$ to produce $\sigma$, this definition would be trivial (a verifier could convince himself by running the simulator, instead of interacting with $P$). Allowing $S$ to generate $\sigma$ still keeps the definition zero-knowledge (since a verifier sees both $(\sigma, \pi)$ together), but puts $P$ and $S$ on unequal footing.
We could also consider the adaptive counterpart, where a cheating verifier can choose $x$ after seeing $\sigma$:

**Definition 6** (Adaptive) Zero-Knowledge. *The exists a PPT simulator split into two stages $S_1, S_2$ such that for all PPT attackers $\mathcal{A}$, the two distributions are computationally indistinguishable:*

1. $\sigma \leftarrow K$

2. $(x, \omega) \leftarrow \mathcal{A}(\sigma)$, *s.t.* $(x, \omega) \in R_L$

3. $\pi \leftarrow P(\sigma, x, \omega)$

4. *Output* $(\sigma, x, \pi)$

*And the simulator output:*

1. $(\sigma, \tau) \leftarrow S_1(1^k)$

2. $(x, \omega) \leftarrow \mathcal{A}(\sigma)$

3. $\pi \leftarrow S_2(\sigma, x, \tau)$

4. *Output* $(\sigma, x, \pi)$

*Where $\tau$ should be thought of as local state stored by the simulator (passed between stages).*

Now we show that adaptive soundness is not much harder to achieve.

**Theorem 1** *Given a NIZK $(K, P, V)$ that is* non-adaptively sound*, we can construct a NIZK that is* adaptively sound*.*

**Proof.** Let us call a particular $\sigma$ "bad for $x_0$" if (for $x_0 \notin L$) there exists a false proof for $x_0$ using randomness $\sigma$: $\exists\ \pi$ s.t. $V(\sigma, x, \pi) = 1$.
By non-adaptive soundness of $(K, P, V)$, we have $\Pr_\sigma[\sigma \text{ bad for } x_0] = neg(k)$. By repeating this NIZK polynomially-many times (using fresh randomness, and requiring a correct proof every iteration), we can ensure $neg(k) \leq 2^{-2P(k)}$.
Now by the union bound:

$$\Pr_\sigma[\exists\ (x, \pi) \text{ s.t. } V(\sigma, x, \pi) = 1] = \Pr_\sigma[\sigma \text{ bad for some } x] \leq 2^{P(k)} \Pr_\sigma[\sigma \text{ bad for } x_0] \leq 2^{-P(k)}$$

So this repeated scheme is adaptively-sound. ■

## 2 NIZK in the Hidden-Bit Model

The hidden-bit model is a variant on the common-reference-string NIZK, where the prover can selectively reveal only parts of the random string to the verifier. (Imagine clouds obscuring the random string in the sky from the verifier, and the prover can choose which clouds to clear).

**Definition 7** *A NIZK in the hidden-bit model for statement $x$ (with witness $\omega$) is efficient algorithms $(K_H, P_H, V_H)$ such that:*

1. *$r \leftarrow K_H(1^k)$ generates the hidden random string ($\ell$-bits).*
2. *$(I, \phi) \leftarrow P_H(r, x, \omega)$ generates the indices $I \subseteq [\ell]$ to reveal, and the proof $\phi$.*
3. *$V_H(I, \{r_i\}_{i \in I}, x, \phi)$ accepts or rejects, given the indices $I$, the random string $r$ at indices $I$, statement $x$, and proof $\phi$.*

*And which satisfy the usual completeness, soundness, and zero-knowledge properties as previously defined.*

**Theorem 2** *Given a NIZK $(P_H, V_H)$ in the hidden-bit model, we can construct a NIZK $(P, V)$ in the normal model. (Using trapdoor one-way permutations).*

**Proof.** Let the common-reference-string $\sigma$ in the normal model be partitioned into $\ell$ blocks of $k$-bits each: $\sigma = \sigma_1 \dots \sigma_\ell$. Let $\mathcal{F}$ be a family of $2^k$ trapdoor OWPs, and let $B(\cdot)$ be the corresponding hard-core bit. We may assume the soundness error of $(P_H, V_H)$ (that is, the probability of $r$ allowing a fake proof) is at most $2^{-2k}$, by the same parallel-repetition argument as above.
The protocol for the normal $(P, V)$ is:
**Prover** $P(\sigma, x, \omega)$**:**

1. Sample trapdoor OWP: $(f, f^{-1}) \leftarrow \mathcal{F}(1^k)$.
2. Let $\alpha_i = f^{-1}(\sigma_i)$ for $i \in [\ell]$
3. Compute hidden-bit $r_i = B(\alpha_i)$ for $i \in [\ell]$
4. Run the HBM prover: $(I, \phi) \leftarrow P_H(r, x, \omega)$
5. Send $(f, I, \{\alpha_i\}_{i \in I}, \phi)$ to verifier.

**Verifier** $V(\sigma, x, f, I, \{\alpha_i\}_{i \in I}, \phi)$**:**

1. Confirm $f \in \mathcal{F}$, and $f(\alpha_i) = \sigma_i \ \forall i \in I$
2. Compute the revealed bits $r_i = B(\alpha_i) \ \forall i \in I$
3. Output $V_H(I, \{r_i\}_{i \in I}, x, \phi)$

Intuitively, $\sigma_i$ hides $r_i$ because $\sigma_i \overset{f}{\leftarrow} \alpha_i \overset{B}{\rightarrow} r_i$, so by security of the hard-core bit, the verifier cannot find $r_i = B(\alpha_i)$ from $\sigma_i = f(\alpha_i)$.
Notice that if the prover is honest, then $\alpha_i$ will be distributed uniformly random as well (since $f^{-1}$ is a permutation), and $r_i$ will be unbiased as well (since $B(\cdot)$ is a hard-core bit). So this reduces

exactly to the HBM distributions, and completeness of this protocol is clear (from completeness of $(P_H, V_H)$). For soundness: for a fixed $f = f_0$, the distribution of $r_i$ is uniformly random, so by the soundness of $(P_H, V_H)$ we have

$$\Pr_{\sigma}[P^* \text{ can cheat using } f_0] \leq 2^{-2k}$$

However, a cheating $P^*$ may be able to cleverly pick $f$ to influence $r_i$, allowing him to cheat. Since we know there are only $2^k$ possible choices of $f$ (the verifier confirms $f$ is properly sampled), we can use the union bound to prove soundness:

$$\Pr_{\sigma}[\ \exists \text{ some } f \in \mathcal{F} \text{ s.t. } P^* \text{ can cheat}] \leq 2^{-k}$$

Note that more serious problems can occur if $V$ does not confirm $f$ is properly sampled. For example, if $f$ is not a permutation, then $f^{-1}(\sigma_i)$ can be multi-valued, and the prover can choose to "explain" $\sigma_i$ using either $\alpha_i$ or $\alpha_i'$ – which is a problem if $B(\alpha_i) \neq B(\alpha_i')$.
Now to prove zero-knowledge, we construct a sequence of prover-hybrids. Differences from the previous hybrid are in red:

$H_0$ (normal model)

1. $\sigma_1 \ldots \sigma_\ell = \sigma \leftarrow \{0,1\}^{kl}$
2. $(f, f^{-1}) \leftarrow \mathcal{F}$
3. $\alpha_i = f^{-1}(\sigma_i)\ \forall i \in [\ell]$
4. $r_i = B(\alpha_i)\ \forall i \in [\ell]$
5. $(I, \phi) \leftarrow P_H(r, x, \omega)$
6. Output $(\sigma, f, I, \{\alpha_i\}_{i \in I}, \phi)$

$H_1$

1. $(f, f^{-1}) \leftarrow \mathcal{F}$
2. $\alpha_i \overset{\$}{\leftarrow} \{0,1\}^k\ \forall i \in [\ell]$
3. $r_i = B(\alpha_i)\ \forall i \in [\ell]$
4. $(I, \phi) \leftarrow P_H(r, x, \omega)$
5. $\sigma_i = f(\alpha_i)\ \forall i \in [\ell]$
6. Output $(\sigma, f, I, \{\alpha_i\}_{i \in I}, \phi)$

In $H_1$, we sample $\alpha_i$ uniformly then generate $\sigma_i$ (instead of sampling $\sigma_i$, and generating $\alpha_i$). This induces an exactly identical distribution, since $f$ is a permutation.

$H_2$

1. $(f, f^{-1}) \leftarrow \mathcal{F}$
2. $r_i \overset{\$}{\leftarrow} \{0,1\}\ \forall i \in [\ell]$
3. $(I, \phi) \leftarrow P_H(r, x, \omega)$
4. $\alpha_i \overset{\$}{\leftarrow} B^{-1}(r_i)\ \forall i \in [\ell]$
5. $\sigma_i = f(\alpha_i)\ \forall i \in [\ell]$
6. Output $(\sigma, f, I, \{\alpha_i\}_{i \in I}, \phi)$

In $H_2$, we again switch the sampling order: first sample the (unbiased) bit $r_i$, then sample $\alpha_i$ from the pre-image of $r_i$ (which can be done efficiently by simply trying random $\alpha_i$s until $B(\alpha_i) = r_i$). This distribution is exactly identical to $H_1$. (The sampling order can be thought of as factoring the joint distribution, as: $\Pr(\alpha_i, r_i) = \Pr(r_i)\Pr(\alpha_i | r_i)$)

$H_3$

1. $(f, f^{-1}) \leftarrow \mathcal{F}$
2. $r_i \overset{\$}{\leftarrow} \{0,1\}\ \forall i \in [\ell]$
3. $(I, \phi) \leftarrow P_H(r, x, \omega)$
4. $\alpha_i \overset{\$}{\leftarrow} B^{-1}(r_i)\ \forall i \in [\ell]$
5. $\sigma_i = f(\alpha_i)\ \forall i \in I$
6. $\sigma_i \overset{\$}{\leftarrow} \{0,1\}^k\ \forall i \notin I$
7. Output $(\sigma, f, I, \{\alpha_i\}_{i \in I}, \phi)$

In $H_3$, we only generate $\sigma_i$ honestly for $i \in I$, and output random $\sigma_i$ for $i \notin I$.
To argue that this is computational indistinguishable, first notice that for a fixed (known) bit $r$, the distributions

$$\{f(B^{-1}(r)\} \overset{c}{\approx} \{f(B^{-1}(\overline{r})\} \tag{1}$$

Where the randomness is over sampling the pre-image $B^{-1}$. Distinguishing the above distributions is by definition equivalent to guessing the hard-core bit, so they are indistinguishable. Given the above, we can further argue that

$$\{f(B^{-1}(r)\} \overset{c}{\approx} \mathcal{U}_k \tag{2}$$

Where $\mathcal{U}_k$ is uniform over $\{0,1\}^k$. To see this, notice that $\mathcal{U}_k$ can be equivalently generated by first sampling a random bit $b$, then outputting $f(B^{-1}(b))$, since $f$ is a permutation. Therefore, any distinguisher for (2) can also be used to distinguish (1) with at least as much distinguishing-advantage (in fact, twice as much). Finally, (2) justifies swapping $\sigma_i = f(\alpha_i) = f(B^{-1}(r_i))$ with random for $i \notin I$ in hybrid $H_3$.

$H_4$

1. $(f, f^{-1}) \leftarrow \mathcal{F}$

2. $(I, \{r_i\}_{i \in I}, \phi) \leftarrow S_H(1^k, x)$

3. $\alpha_i \overset{\$}{\leftarrow} B^{-1}(r_i)\ \forall i \in I$

4. $\sigma_i = f(\alpha_i)\ \forall i \in I$

5. $\sigma_i \overset{\$}{\leftarrow} \{0,1\}^k\ \forall i \notin I$

6. Output $(\sigma, f, I, \{\alpha_i\}_{i \in I}, \phi)$

Finally, $H_4$ simply swaps the hidden-bit prover $P_H$ for the hidden-bit simulator $S_H$, which is indistinguishable by the zero-knowledge property of $(P_H, V_H)$. So $(P, V)$ is a NIZK system in the normal model. ■