CS 600.442 – Modern Cryptography October 31, 2016

Lecture 14: NIZK (I)

Instructor: Abhishek Jain *Scribe: Arka Rai Choudhuri*

1 Non-Interactive Zero Knowledge

So far we have discussed the case of interactive zero-knowledge proofs. But what if Alice has the resource to send only a single message to Bob? This proof will now become "non-interactive". But 1-message zero-knowledge is only possible for languages in **BPP**. This is because any simulator that can simulate the "single" message can use this as a witness for x. But this is pretty useless, at the very least we want to be able to prove statements for languages in **NP**.

Fortunately, our savior is a "random string in the sky". This means that both Alice and Bob have access to a *common random string* that was honestly generated by someone they both trust. This string is something beyond the influence of either participant. While this is a departure from the model we have been considering, how can we hope to prove statements non-interactively using the common random string?

Let us start by formally defining non-interactive proofs,

Definition 1 *A non-interactive proof system for a language L with witness relation R is a tuple of algorithms $(\mathsf{K}, \mathsf{P}, \mathsf{V})$ such that:*

1. **Setup:** $\sigma \leftarrow \mathsf{K}(1^n)$ *outputs a common random string.*
2. **Prove:** $\pi \leftarrow \mathsf{P}(\sigma, x, w)$ *takes as input a common random string σ and a statement $x \in L$ and a witness w and outputs a proof π.*
3. **Verify:** $\mathsf{V}(\sigma, x, \pi)$ *outputs 1 if it accepts the proof and 0 otherwise.*

A non-interactive proof system must satisfy completeness and soundness properties given below.

Completeness: $\forall x \in L, \forall w \in R(x)$:

$$\Pr[\sigma \leftarrow \mathsf{K}(1^n);\ \pi \leftarrow \mathsf{P}(\sigma, x, w) : \mathsf{V}(\sigma, x, \pi) = 1] = 1$$

Non-Adaptive Soundness: There exists a negligible function $\nu(\cdot)$ such that $\forall x \notin L$

$$\Pr[\sigma \leftarrow \mathsf{K}(1^n);\ \exists\ \pi \text{ s.t. } \mathsf{V}(\sigma, x, \pi) = 1] \leq \nu(n)$$

Adaptive Soundness: There exists a negligible function $\nu(\cdot)$ such that

$$\Pr[\sigma \leftarrow \mathsf{K}(1^n);\ \exists\ (x, \pi) \text{ s.t. } x \notin L \wedge \mathsf{V}(\sigma, x, \pi) = 1] \leq \nu(n)$$

The reader should note, in non-adaptive soundness, the adversary chooses x before seeing the common random string whereas in adaptive soundness, it can choose x depending upon the common random string. Adaptive soundness is a stronger notion of soundness.

Similar to soundness, we will define two variants of Non-interactive Zero Knowledge (NIZK). Before we proceed, we note that we can transform any NIZK proof system with non-adaptive soundness into one that achieves adaptive soundness in a manner similar to the hardness amplification done earlier in the course. The details can be found in [1].

Since each statement can have multiple witnesses, we define the following set for each $x \in L$:

$$R(x) = \{w \mid R(x, w) = 1\}$$

where R is the relation for the language L.

Definition 2 (Non-Adaptive NIZK) *A non-interactive proof system* $(\mathsf{K}, \mathsf{P}, \mathsf{V})$ *for a language* L *with witness relation* R *is non-adaptive zero-knowledge if there exists a PPT simulator* $\mathcal{S}$ *s.t. for every* $x \in L, w \in R(x)$*, the output distribution of the following two experiments are computationally indistinguishable:*

$\mathsf{REAL}(1^n, x, w)$	$\mathsf{IDEAL}(1^n, x)$
$\sigma \leftarrow \mathsf{K}(1^n)$	$(\sigma, \pi) \leftarrow \mathcal{S}(1^n, x)$
$\pi \leftarrow \mathsf{P}(\sigma, x, w)$	
Output (σ, π)	*Output* (σ, π)

Here the simulator generates both the common random string and the simulated proof given the statement x as input. The simulated common random string can depend on x and thus can be used only for a single proof.

We proceed to define the adaptive variant below.

Definition 3 (Adaptive NIZK) *A non-interactive proof system* $(\mathsf{K}, \mathsf{P}, \mathsf{V})$ *for a language* L *with witness relation* R *is adaptive zero-knowledge if there exists a PPT simulator* $\mathcal{S} = (\mathcal{S}_0, \mathcal{S}_1)$ *s.t. for every* $x \in L, w \in R(x)$*, the output distribution of the following two experiments are computationally indistinguishable:*

$\mathsf{REAL}(1^n, x, w)$	$\mathsf{IDEAL}(1^n, x)$
$\sigma \leftarrow \mathsf{K}(1^n)$	$(\sigma, \tau) \leftarrow \mathcal{S}_0(1^n)$
$\pi \leftarrow \mathsf{P}(\sigma, x, w)$	$\pi \leftarrow \mathcal{S}_1(\sigma, \tau, x)$
Output (σ, π)	*Output* (σ, π)

Here τ is the "trapdoor" for the simulated common random string σ that is used by $\mathcal{S}_1$ to generate an accepting proof for x without knowing the witness. This definition also captures the definition of *reusable* common random strings.

Recall, unlike the definition of (interactive) zero-knowledge, we don't define the zero-knowledge property over "all non-uniform PPT adversary V^*". We leave it to the reader to see why it is the case.

We make a few remarks about the NIZK definition before we proceed further,

- In NIZK, the simulator is given the seemingly extra power to choose the common random string along with a possible trapdoor that allows for simulation without a witness.
- In the interactive case, we gave the simulator the extra power to "reset" the verifier. Is this extra power inherent?
- It turns out that a simulator must always have some extra power over the normal prover, otherwise, the definition would be impossible to realize for languages outside **BPP**.

- We justify the extra power since we require indistinguishability of the joint-distribution over the common random string and the proof.

Lemma 1 *There exists an efficient transformation from any non-interactive proof system* $\mathsf{K}, \mathsf{P}, \mathsf{V}$ *with non-adaptive soundness into a non-interactive proof system* $\mathsf{K}', \mathsf{P}', \mathsf{V}'$ *with adaptive soundness.*

2 NIZKs for NP

I. Non-adaptive Zero Knowledge: We first construct NIZKs for **NP** with non-adaptive zero-knowledge property using the following two steps:

1. Construct a NIZK proof system for **NP** in the **hidden-bit model**. This step is unconditional.

2. Using trapdoor permutation, transform any NIZK proof system for language in the hidden-bit model to a non-adaptive NIZK proof system in the common random string model.

In today's class we shall define NIZKs in the hidden-bit model, and show a transformation from NIZKs in hidden-bit model to NIZKs in the common-random string model. In the next class we shall build NIZKs for **NP** in the hidden-bit model.

II. Adaptive Zero Knowledge: Next, we transform non-adaptive NIZKs for **NP** into adaptive NIZKs for **NP**. This step only requires one-way functions, which are implied by trapdoor permutations. This will be a part of the homework.

Putting all the steps together, we obtain adaptive NIZKs for **NP** based on trapdoor permutations.

3 The Hidden-Bit Model

In this section we shall describe the hidden-bit model and define NIZK in the hidden-bit model. It is important to note that this model provides a step towards our ultimate goal of building NIZKs for **NP**, and is not meant to be realistic.

In this model, the prover is given some sequence of bits that are hidden from the verifier. To prove some statement $x \in L$, the prover may choose to reveal some of some of these bits to the verifier. The remaining bits remain hidden from the verifier. Also, the prover cannot tamper these bits before revealing them to the verifier.

Definition 4 *A non-interactive proof system for a language* L *with witness relation* R *in the hidden-bit model is a tuple of algorithms* $(\mathsf{K}_{\mathsf{HB}}, \mathsf{P}_{\mathsf{HB}}, \mathsf{V}_{\mathsf{HB}})$ *such that:*

1. **Setup:** $r \leftarrow \mathsf{K}(1^n)$ *outputs a common random string.*

2. **Prove:** $\pi \leftarrow \mathsf{P}_{\mathsf{HB}}(r, x, w)$ *generates the indices* $I \subseteq [|r|]$ *of* r *to reveal along with a proof* π*.*

3. **Verify:** $\mathsf{V}_{\mathsf{HB}}(I, \{r_i\}_{i \in I}, x, \pi)$ *outputs 1 if it accepts the proof and 0 otherwise.*

We define zero-knowledge below. A non-interactive proof system must satisfy completeness and soundness properties defined earlier.

Definition 5 (NIZK in Hidden-Bit Model) *A non-interactive proof system* $(\mathsf{K}_{\mathsf{HB}}, \mathsf{P}_{\mathsf{HB}}, \mathsf{V}_{\mathsf{HB}})$ *for a language L with witness relation R in the hidden-bit model is (non-adaptive)zero-knowledge if there exists a PPT simulator $\mathcal{S}_{\mathsf{HB}}$ s.t. for every $x \in L, w \in R(x)$, the output distribution of the following two experiments are computationally indistinguishable:*

$\mathsf{REAL}(1^n, x, w)$	$\mathsf{IDEAL}(1^n, x)$
$r \leftarrow \mathsf{K}_{\mathsf{HB}}(1^n)$	$(I, \{r_i\}_{i \in I}, \pi) \leftarrow \mathcal{S}_{\mathsf{HB}}(1^n, x)$
$(I, \pi) \leftarrow \mathsf{P}_{\mathsf{HB}}(r, x, w)$	
Output $(I, \{r_i\}_{i \in I}, \pi)$	*Output* $(I, \{r_i\}_{i \in I}, \pi)$

4 From NIZK in HB model to NIZK in CRS model

We sketch our intuition for the construction. We need to transform a "public" random string into a "hidden" random string. If the prover samples a trapdoor permutation (f, f^{-1}) with hardcore predicate h. Given a common random string $\sigma = \sigma_1, \cdots, \sigma_n$, the prover can compute $r = r_1 \cdots, r_n$ where:

$$r_i = h(f^{-1}(\sigma_i)).$$

Since f is a permutation and h is a hard-core predicate, r is guaranteed to be random. Then we can treat r as the hidden random string, revealing only parts of it to V. We now proceed to the construction.

Construction. Let $\mathcal{F} = \{f, f^{-1}\}$ be a family of 2^n trapdoor permutations with hardcore predicate h. We assume that it is easy to test membership of $\mathcal{F}$. Let $(\mathsf{K}_{\mathsf{HB}}, \mathsf{P}_{\mathsf{HB}}, \mathsf{V}_{\mathsf{HB}})$ be a NIZK proof system for L in the hidden-bit model with soundness error $\frac{1}{2^{2n}}$. The procedures for K, P and V are given below.

```
K(1^n)
for every $i \in [n]$
    $\sigma_i \xleftarrow{\$} \{0,1\}^n$
Output $\sigma = \sigma_1 \sigma_2 \cdots \sigma_n$
```

```
P(σ, x, w)
$(f, f^{-1}) \xleftarrow{\$} \mathcal{F}(1^n)$
for every $i \in [n]$
    $\alpha_i = f^{-1}(\sigma_i)$
for every $i \in [n]$
    $r_i = h(\alpha_i)$
$(I, \Phi) \leftarrow \mathsf{P}_{\mathsf{HB}}(r, x, w)$
Output $\pi = (\sigma, f, I, \{\alpha_i\}_{i \in I}, \Phi)$
```

```
$\mathsf{V}(\sigma, x, \pi)$
$(\sigma, f, I, \{\alpha_i\}_{i \in I}, \Phi) \leftarrow \pi$
Check $f \in \mathcal{F}$
Check for every $i \in I$
    $f(\alpha_i) = \sigma_i$
for every $i \in I$
    $r_i = h(\alpha_i)$
Output $\mathsf{V}_{\mathsf{HB}}(I, \{\alpha_i\}_{i \in I}, x, \Phi)$
```

Theorem 2 *Given that* $(\mathsf{K}_{\mathsf{HB}}, \mathsf{P}_{\mathsf{HB}}, \mathsf{V}_{\mathsf{HB}})$ *is a NIZK proof system for* L *in the hidden-bit model with soundness error* $\frac{1}{2^{2n}}$*, then our construction* $(\mathsf{K}, \mathsf{P}, \mathsf{V})$ *above is a NIZK proof system for* L *in the CRS model.*

Proof. We need to argue that each property of the NIZK proof system in the CRS model is satisfied.

Completeness

We sample each σ_i uniformly at random, and since f^{-1} is a permutation, α_i will be uniform random. For completeness, by definition, we assume that the prover is honest and thus picks f and f^{-1} correctly. Since h is a hardcore predicate, each r_i is also random. Since the sampling and computations are done independently, we get r to be uniformly distributed. Now this reduces to the *hidden-bit model*. Completeness follows from the completeness of (K_{HB}, P_{HB}, V_{HB})

Soundness

When we fix f to be f_0, r is uniformly distributed. Thus, from the (non-adaptive) soundness of $(\mathsf{K}_{\mathsf{HB}}, \mathsf{P}_{\mathsf{HB}}, \mathsf{V}_{\mathsf{HB}})$, we have

$$\mathsf{Pr}[\sigma \leftarrow K(1^n) : \mathsf{P}^* \text{ can cheat using } f_0] \leq \frac{1}{2^{2n}}$$

There are only 2^n possible choices for f, and the verifier checks if f is indeed from $\mathcal{F}$. By the union bound, we have

$$\mathsf{Pr}[\sigma \leftarrow K(1^n) : \mathsf{P}^* \text{ can cheat}] = \mathsf{Pr}[\sigma \leftarrow K(1^n) : \mathsf{P}^* \text{ can cheat for some } f] \leq \frac{1}{2^n}$$

Zero-Knowledge

Let $\mathcal{S}_{\mathsf{HB}}$ be the simulator for $(\mathsf{K}_{\mathsf{HB}}, \mathsf{P}_{\mathsf{HB}}, \mathsf{V}_{\mathsf{HB}})$. The simulator is,

```
$\mathcal{S}(1^n, x)$
$(I, \{r_i\}_{i \in I}, \Phi) \xleftarrow{\$} \mathcal{S}_{\mathsf{HB}}(1^n, x)$
$(f, f^{-1}) \xleftarrow{\$} \mathcal{F}$
for every $i \in I$
    $\alpha_i = h^{-1}(r_i)$
for every $i \in I$
    $\sigma_i = f(\alpha_i)$
for every $i \notin I$
    $\sigma_i \xleftarrow{\$} \{0,1\}^n$
Output $(\sigma, f, I, \{\alpha_i\}_{i \in I}, \Phi)$
```

Here $h^{-1}(r_i)$ denotes sampling from the pre-image of r_i, which can be done efficiently by simply trying random α_i's until $h(\alpha_i) = r_i$. This method has low expected number of attempts as we are trying to match just one bit r_i. To prove zero-knowledge, we build a sequence of hybrids below. We argue the computational indistinguishability of the hybrids at the end. Changes from the previous hybrid are marked with a box.

```
$H_0(1^n, x, w) = \mathsf{REAL}(1^n, x, w)$
$\sigma \xleftarrow{\$} K(1^n)$ where $\sigma = \sigma_1, \cdots, \sigma_n$
$(f, f^{-1}) \xleftarrow{\$} \mathcal{F}$
for every $i \in [n]$
    $\alpha_i = f^{-1}(\sigma_i)$
for every $i \in [n]$
    $r_i = h(\alpha_i)$
$(I, \Phi) \leftarrow \mathsf{P_{HB}}(r, x, w)$
Output $(\sigma, f, I, \{\alpha_i\}_{i \in I}, \Phi)$
```

```
$H_1(1^n, x, w)$
$\boxed{\alpha_i \xleftarrow{\$} \{0,1\}^n \text{ for every } i \in [n]}$
$(f, f^{-1}) \xleftarrow{\$} \mathcal{F}$
for every $i \in [n]$
    $\boxed{\sigma_i = f(\alpha_i)}$
for every $i \in [n]$
    $r_i = h(\alpha_i)$
$(I, \Phi) \leftarrow \mathsf{P_{HB}}(r, x, w)$
Output $(\sigma, f, I, \{\alpha_i\}_{i \in I}, \Phi)$
```

```
$H_2(1^n, x, w)$
$\boxed{r_i \xleftarrow{\$} \{0,1\}^n \text{ for every } i \in [n]}$
$(f, f^{-1}) \xleftarrow{\$} \mathcal{F}$
for every $i \in [n]$
    $\boxed{\alpha_i = h^{-1}(r_i)}$
for every $i \in [n]$
    $\sigma_i = f(\alpha_i)$
$(I, \Phi) \leftarrow \mathsf{P_{HB}}(r, x, w)$
Output $(\sigma, f, I, \{\alpha_i\}_{i \in I}, \Phi)$
```

```
$H_3(1^n, x, w)$
$r_i \xleftarrow{\$} \{0,1\}^n$ for every $i \in [n]$
$(f, f^{-1}) \xleftarrow{\$} \mathcal{F}$
for every $i \in [n]$
    $\alpha_i = h^{-1}(r_i)$
$(I, \Phi) \leftarrow \mathsf{P_{HB}}(r, x, w)$
$\boxed{\text{for every } i \in I}$
    $\boxed{\sigma_i = f(\alpha_i)}$
$\boxed{\text{for every } i \notin I}$
    $\boxed{\sigma_i \xleftarrow{\$} \{0,1\}^n}$
Output $(\sigma, f, I, \{\alpha_i\}_{i \in I}, \Phi)$
```

```
$H_4(1^n, x) = \mathsf{IDEAL}(1^n, x)$
$\boxed{(I, \{r_i\}_{i \in I}, \Phi) \xleftarrow{\$} \mathcal{S}_{\mathsf{HB}}(1^n, x)}$
$(f, f^{-1}) \xleftarrow{\$} \mathcal{F}$
$\boxed{\text{for every } i \in I}$
    $\alpha_i = h^{-1}(r_i)$
for every $i \in I$
    $\sigma_i = f(\alpha_i)$
for every $i \notin I$
    $\sigma_i \xleftarrow{\$} \{0,1\}^n$
Output $(\sigma, f, I, \{\alpha_i\}_{i \in I}, \Phi)$
```

$\mathbf{H_0} \approx \mathbf{H_1}$: In H_1, we sample α_i at random and then compute σ_i. This is in contrast to H_0 where we sample σ_i before computing α_i. Since f is a permutation, H_1 induces the same distribution as H_0.

$\mathbf{H_1} \approx \mathbf{H_2}$: In H_2, we first sample r_i before sampling α_i from the pre-image of r_i. This distribution is identical to H_1.

$\mathbf{H_2} \approx \mathbf{H_3}$: In H_3, we output a random σ_i for $i \notin I$. From the security of the hard-core predicate h, it follows that

$$\{f(h^{-1}(r_i))\} \approx_c U_n$$

Indistinguishability of H_2 and H_3 follows using the above equation.

$\mathbf{H_3} \approx \mathbf{H_4}$: In H_4, we swap $\mathbf{P_{HB}}$ with $\boldsymbol{\mathcal{S}}_{\mathbf{HB}}$. Indistinguishability follows from the zero-knowledge property of $(\mathsf{K}_{\mathsf{HB}}, \mathsf{P}_{\mathsf{HB}}, \mathsf{V}_{\mathsf{HB}})$.

Thus $H_0 \approx H_4$. This gives us the zero-knowledge proof. ■

In the next class we shall construct NIZKs for all languages in **NP** in the hidden-bit model.

References

[1] Jonathan Katz. Advanced Topics in Cryptography - Lecture 11, 2004. http://www.cs.umd.edu/~jkatz/gradcrypto2/NOTES/lecture11.pdf.