CS 276 – Cryptography November 10, 2014

Lecture 20: Using Indistinguishability Obfuscation

Instructor: Sanjam Garg *Scribe: Albert Ou*

1 $i\mathcal{O}$ for Polynomial-sized Circuits

Definition 1 (Indistinguishability Obfuscator) *A uniform PPT machine $i\mathcal{O}$ is an* indistinguishability obfuscator *for a collection of circuits $\mathcal{C}_\kappa$ if the following conditions hold:*

- Correctness. *For every circuit $C \in \mathcal{C}_\kappa$ and for all inputs x, $C(x) = i\mathcal{O}(C(x))$.*
- Polynomial slowdown. *For every circuit $C \in \mathcal{C}_\kappa$, $|i\mathcal{O}(C)| \leq p(|C|)$ for some polynomial p.*
- Indistinguishability. *For all pairs of circuits $C_1, C_2 \in \mathcal{C}_\kappa$, if $|C_1| = |C_2|$ and $C_1(x) = C_2(x)$ for all inputs x, then $i\mathcal{O}(C_1) \stackrel{c}{\simeq} i\mathcal{O}(C_2)$. More precisely, there is a negligible function $\nu(k)$ such that for any (possibly non-uniform) PPT A,*

$$\big|\Pr[A(i\mathcal{O}(C_1)) = 1] - \Pr[A(i\mathcal{O}(C_2)) = 1]\big| \leq \nu(k)$$

Definition 2 (Indistinguishability Obfuscator for $\mathbf{NC}^1$) *Let $\mathcal{C}_\kappa$ be the collection of circuits of size $O(\kappa)$ and depth $O(\log \kappa)$ with respect to gates of bounded fan-in. Then a uniform PPT machine $i\mathcal{O}_{\mathbf{NC}^1}$ is an* indistinguishability obfuscator *for circuit class $\mathbf{NC}^1$ if it is an indistinguishability obfuscator for $\mathcal{C}_\kappa$.*

Given an indistinguishability obfuscator $i\mathcal{O}_{\mathbf{NC}^1}$ for circuit class $\mathbf{NC}^1$, we shall demonstrate how to achieve an indistinguishability obfuscator $i\mathcal{O}$ for all polynomial-sized circuits. The amplification relies on fully homomorphic encryption (FHE).

Definition 3 (Homomorphic Encryption) *A* homomorphic encryption scheme *is a tuple of PPT algorithms* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *as follows:*

- $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is a semantically-secure public-key encryption scheme.*
- $\mathsf{Eval}(\mathsf{pk}, C, e)$ *takes public key* pk*, an arithmetic circuit C, and ciphertext $e = \mathsf{Enc}(\mathsf{pk}, x)$ of some circuit input x, and outputs* $\mathsf{Enc}(\mathsf{pk}, C(x))$.

As an example, the ElGamal encryption scheme introduced in a preceding lecture is homomorphic over the multiplication function. Consider a cyclic group G of order q and generator g, and let $\mathsf{sk} = a$ and $\mathsf{pk} = g^a$. For ciphertexts $\mathsf{Enc}(\mathsf{pk}, m_1) = (g^{r_1}, g^{ar_1} \cdot m_1)$ and $\mathsf{Enc}(\mathsf{pk}, m_2) = (g^{r_2}, g^{ar_2} \cdot m_2)$, observe that

$$\mathsf{Enc}(\mathsf{pk}, m_1) \cdot \mathsf{Enc}(\mathsf{pk}, m_2) = (g^{r_1+r_2}, g^{a(r_1+r_2)} \cdot m_1 \cdot m_2) = \mathsf{Enc}(\mathsf{pk}, m_1 \cdot m_2)$$

Note that this scheme becomes additively homomorphic by encrypting g^m instead of m.

Definition 4 (Fully Homomorphic Encryption) *An encryption scheme is* fully homomorphic *if it is both compact and homomorphic for the class of all arithmetic circuits. Compactness requires that the size of the output of* $\mathsf{Eval}(\cdot, \cdot, \cdot)$ *is at most polynomial in the security parameter κ.*

1.1 Construction

Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ be a fully homomorphic encryption scheme. We require that Dec be realizable by a circuit in $\mathbf{NC}^1$. The obfuscation procedure accepts a security parameter κ and a circuit C whose size is at most polynomial in κ.

1. Generate $(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{Gen}(1^\kappa)$ and $(\mathsf{pk}_2, \mathsf{sk}_2) \leftarrow \mathsf{Gen}(1^\kappa)$.

2. Encrypt C, encoded in canonical form, as $e_1 \leftarrow \mathsf{Enc}(\mathsf{pk}_1, C)$ and $e_2 \leftarrow \mathsf{Enc}(\mathsf{pk}_2, C)$.

3. Output an obfuscation $\sigma = (i\mathcal{O}_{\mathbf{NC}^1}(P), \mathsf{pk}_1, \mathsf{pk}_2, e_1, e_2)$ of program $P_{\mathsf{pk}_1,\mathsf{pk}_2,\mathsf{sk}_1,e_1,e_2}$ as described below.

The evaluation procedure accepts the obfuscation σ and program input x.

1. Let U be a universal circuit that computes $C(x)$ given a circuit description C and input x, and denote by U_x the circuit $U(\cdot, x)$ where x is hard-wired. Let R_1 and R_2 be the circuits which compute $f_1 \leftarrow \mathsf{Eval}(U_x, e_1)$ and $f_2 \leftarrow \mathsf{Eval}(U_x, e_2)$, respectively.

2. Denote by ω_1 and ω_2 the set of all wires in R_1 and R_2, respectively. Compute $\pi_1 : \omega_1 \to \{0, 1\}$ and $\pi_2 : \omega_2 \to \{0, 1\}$, which yield the value of internal wire $w \in \omega_1, \omega_2$ when applying x as the input to R_1 and R_2.

3. Output the result of running $P_{\mathsf{pk}_1,\mathsf{pk}_2,\mathsf{sk}_1,e_1,e_2}(x, f_1, \pi_1, f_2, \pi_2)$.

Program $P_{\mathsf{pk}_1,\mathsf{pk}_2,\mathsf{sk}_1,e_1,e_2}$ has pk_1, pk_2, sk_1, e_1, and e_2 embedded.

1. Check whether $R_1(x) = f_1 \wedge R_2(x) = f_2$. π_1 and π_2 enable this check in logarithmic depth.

2. If the check succeeds, output $\mathsf{Dec}(\mathsf{sk}_1, f_1)$; otherwise output $\perp$.

The use of two key pairs and two encryptions of C, similar to CCA1-secure schemes seen previously, eliminates the virtual black-box requirement for concealing sk_1 within $i\mathcal{O}_{\mathbf{NC}^1}(P_{\mathsf{pk}_1,\mathsf{pk}_2,\mathsf{sk}_1,e_1,e_2})$.

1.2 Proof of Security

We prove the indistinguishability property for this construction through a hybrid argument.
Proof. Through the sequence of hybrids, we gradually transform an obfuscation of circuit C_1 into an obfuscation of circuit C_2, with each successor being indistinguishable from its antecedent.

H_0 : This corresponds to an honest execution of $i\mathcal{O}(C_1)$. Recall that $e_1 = \mathsf{Enc}(\mathsf{pk}_1, C_1)$, $e_2 = \mathsf{Enc}(\mathsf{pk}_2, C_1)$, and $\sigma = (i\mathcal{O}_{\mathbf{NC}^1}(P_{\mathsf{pk}_1,\mathsf{pk}_2,\mathsf{sk}_1,e_1,e_2}), \ldots)$.

H_1 : We instead generate $e_2 = \mathsf{Enc}(\mathsf{pk}_2, C_2)$, relying on the semantic security of the underlying fully homomorphic encryption scheme.

H_2 : We alter program $P_{\mathsf{pk}_1,\mathsf{pk}_2,\mathsf{sk}_2,e_1,e_2}$ such that it instead embeds sk_2 and outputs $\mathsf{Dec}(\mathsf{sk}_2, f_2)$. The output of the obfuscation procedure becomes $\sigma = (i\mathcal{O}_{\mathbf{NC}^1}(P_{\mathsf{pk}_1,\mathsf{pk}_2,\mathsf{sk}_2,e_1,e_2}, \ldots)$; we rely on the properties of functional equivalence and indistinguishability of $i\mathcal{O}_{\mathbf{NC}^1}$.

H_3 : We generate $e_1 = \mathsf{Enc}(\mathsf{pk}_1, C_1)$ since sk_1 is now unused, relying again on the semantic security of the fully homomorphic encryption scheme.

H_4 : We revert to the original program $P_{\mathsf{pk}_1,\mathsf{pk}_2,\mathsf{sk}_1,e_1,e_2}$ and arrive at an honest execution of $i\mathcal{O}(C_1)$.

2 Identity-Based Encryption

Another use of indistinguishability obfuscation is to realize identity-based encryption (IBE).

Definition 5 (Identity-Based Encryption) *An* identity-based encryption scheme *is a tuple of PPT algorithms* $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *as follows:*

- $\mathsf{Setup}(1^\kappa)$ *generates and outputs a master public/private key pair* $(\mathsf{mpk}, \mathsf{msk})$.
- $\mathsf{KeyGen}(\mathsf{msk}, \mathsf{id})$ *derives and outputs a secret key* $\mathsf{sk}_{\mathsf{id}}$ *for identity* id.
- $\mathsf{Enc}(\mathsf{mpk}, \mathsf{id}, m)$ *encrypts message* m *under identity* id *and outputs the ciphertext.*
- $\mathsf{Dec}(\mathsf{sk}_{\mathsf{id}}, c)$ *decrypts ciphertext* c *and outputs the corresponding message if* c *is a valid encryption under identity* id*, or* $\perp$ *otherwise.*

We combine an indistinguishability obfuscator $i\mathcal{O}$ with a digital signature scheme $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$.

- Let $\mathsf{Setup} \equiv \mathsf{Gen}$ and $\mathsf{KeyGen} \equiv \mathsf{Sign}$.
- Enc outputs $i\mathcal{O}(P_m)$, where P_m is a program that outputs (embedded) message m if input sk is a secret key for the given id, or $\perp$ otherwise.
- Dec outputs the result of $c(\mathsf{sk}_{\mathsf{id}})$.

However, this requires that we have encryption scheme where the "signatures" do not exist. We therefore investigate an alternative scheme. Let (K, P, V) be a non-interactive zero-knowledge (NIZK) proof system. Denote by $\mathsf{Com}(\cdot; r)$ the commitment algorithm of a non-interactive commitment scheme with explicit random coin r.

- Let σ be a common random string. $\mathsf{Setup}(1^\kappa)$ outputs $(\mathsf{mpk} = (\sigma, c_1, c_2), \mathsf{msk} = r_1)$, where $c_1 = \mathsf{Com}(0; r_1)$ and $c_2 = \mathsf{Com}(0^{|\mathsf{id}|}; r_2)$.
- $\mathsf{KeyGen}(\mathsf{msk}, \mathsf{id})$ produces a proof $\pi = P(\sigma, x_{\mathsf{id}}, s)$ for the following language L: $x \in L$ if there exists s such that
$$\underbrace{c_1 = \mathsf{Com}(0; s)}_{\text{Type I witness}} \vee \underbrace{(c_2 = \mathsf{Com}(\mathsf{id}^*; s) \wedge \mathsf{id}^* \neq \mathsf{id})}_{\text{Type II witness}}$$
- Let $P_{\mathsf{id},m}$ be a program which outputs m if $V(\sigma, x_{\mathsf{id}}, \pi_{\mathsf{id}}) = 1$ or outputs $\perp$ otherwise.
  $\mathsf{Enc}(\mathsf{mpk}, \mathsf{id}, m)$ outputs $i\mathcal{O}(P_{\mathsf{id},m})$.

We briefly sketch the hybrid argument:

H_0 : This corresponds to an honest execution as described above.

H_1 : We let $c_2 = \mathsf{Com}(\mathsf{id}^*; r_2)$, relying on the hiding property of the commitment scheme.

H_2 : We switch to the Type II witness using $\pi_{\mathsf{id}_i} \forall i \in [q]$, corresponding to the queries issued by the adversary during the first phase of the selective-identity security game.

H_3 : We let $c_1 = \mathsf{Com}(1; r_1)$.