1 \(iO\) for Polynomial-sized Circuits

**Definition 1 (Indistinguishability Obfuscator)** A uniform PPT machine \(iO\) is an *indistinguishability obfuscator* for a collection of circuits \(C_\kappa\) if the following conditions hold:

- **Correctness.** For every circuit \(C \in C_\kappa\) and for all inputs \(x\), \(C(x) = iO(C(x))\).
- **Polynomial slowdown.** For every circuit \(C \in C_\kappa\), \(|iO(C)| \leq p(|C|)\) for some polynomial \(p\).
- **Indistinguishability.** For all pairs of circuits \(C_1, C_2 \in C_\kappa\), if \(|C_1| = |C_2|\) and \(C_1(x) = C_2(x)\) for all inputs \(x\), then \(iO(C_1) \simeq iO(C_2)\). More precisely, there is a negligible function \(\nu(k)\) such that for any (possibly non-uniform) PPT \(A\),
  \[
  \left| \Pr[A(iO(C_1)) = 1] - \Pr[A(iO(C_2)) = 1] \right| \leq \nu(k)
  \]

**Definition 2 (Indistinguishability Obfuscator for NC\(^1\))** Let \(C_\kappa\) 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 \(iO_{NC^1}\) is an indistinguishability obfuscator for circuit class \(NC^1\) if it is an indistinguishability obfuscator for \(C_\kappa\).

Given an indistinguishability obfuscator \(iO_{NC^1}\) for circuit class \(NC^1\), we shall demonstrate how to achieve an indistinguishability obfuscator \(iO\) 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 \((\text{Gen}, \text{Enc}, \text{Dec}, \text{Eval})\) as follows:

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

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 \(sk = a\) and \(pk = g^a\). For ciphertexts \(\text{Enc}(pk, m_1) = (g^{r_1}, g^{ar_1} \cdot m_1)\) and \(\text{Enc}(pk, m_2) = (g^{r_2}, g^{ar_2} \cdot m_2)\), observe that
\[
\text{Enc}(pk, m_1) \cdot \text{Enc}(pk, m_2) = (g^{r_1+r_2}, g^{a(r_1+r_2)} \cdot m_1 \cdot m_2) = \text{Enc}(pk, m_1 \cdot m_2)
\]
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 \(\text{Eval}(\cdot, \cdot, \cdot)\) is at most polynomial in the security parameter \(\kappa\).
1.1 Construction

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

1. Generate \((pk_1, sk_1) \leftarrow Gen(1^\kappa)\) and \((pk_2, sk_2) \leftarrow Gen(1^\kappa)\).

2. Encrypt \(C\), encoded in canonical form, as \(e_1 \leftarrow Enc(pk_1, C)\) and \(e_2 \leftarrow Enc(pk_2, C)\).

3. Output an obfuscation \(\sigma = (iO_{\text{NC}^1}(P), pk_1, pk_2, e_1, e_2)\) of program \(P_{pk_1, pk_2, sk_1, e_1, e_2}\) as described below.

The evaluation procedure accepts the obfuscation \(\sigma\) 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 \(f_1 \leftarrow Eval(U_x, e_1)\) and \(f_2 \leftarrow Eval(U_x, e_2)\), respectively.

2. Denote by \(\omega_1\) and \(\omega_2\) the set of all wires in \(R_1\) and \(R_2\), respectively. Compute \(\pi_1 : \omega_1 \rightarrow \{0, 1\}\) and \(\pi_2 : \omega_2 \rightarrow \{0, 1\}\), 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 \(P_{pk_1, pk_2, sk_1, e_1, e_2}(x, f_1, \pi_1, f_2, \pi_2)\).

Program \(P_{pk_1, pk_2, sk_1, e_1, e_2}\) has \(pk_1, pk_2, sk_1, e_1, e_2\) embedded.

1. Check whether \(R_1(x) = f_1 \land R_2(x) = f_2\). \(\pi_1\) and \(\pi_2\) enable this check in logarithmic depth.

2. If the check succeeds, output \(Dec(sk_1, f_1)\); otherwise output \(\bot\).

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 \(iO_{\text{NC}^1}(P_{pk_1, pk_2, sk_1, e_1, e_2})\).

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 \(iO(C_1)\). Recall that \(e_1 = Enc(pk_1, C_1)\), \(e_2 = Enc(pk_2, C_1)\), and \(\sigma = (iO_{\text{NC}^1}(P_{pk_1, pk_2, sk_1, e_1, e_2}), \ldots)\).

\(H_1\) : We instead generate \(e_2 = Enc(pk_2, C_2)\), relying on the semantic security of the underlying fully homomorphic encryption scheme.

\(H_2\) : We alter program \(P_{pk_1, pk_2, sk_1, e_1, e_2}\) such that it instead embeds \(sk_2\) and outputs \(Dec(sk_2, f_2)\). The output of the obfuscation procedure becomes \(\sigma = (iO_{\text{NC}^1}(P_{pk_1, pk_2, sk_2, e_1, e_2}), \ldots)\); we rely on the properties of functional equivalence and indistinguishability of \(iO_{\text{NC}^1}\).

\(H_3\) : We generate \(e_1 = Enc(pk_1, C_1)\) 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 \(P_{pk_1, pk_2, sk_1, e_1, e_2}\) and arrive at an honest execution of \(iO(C_1)\).
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 (Setup, KeyGen, Enc, Dec) as follows:

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

We combine an indistinguishability obfuscator iO with a digital signature scheme (Gen, Sign, Verify).

- Let Setup ≡ Gen and KeyGen ≡ Sign.
- Enc outputs iO(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 ⊥ otherwise.
- Dec outputs the result of c(sk_{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 Com(·; r) the commitment algorithm of a non-interactive commitment scheme with explicit random coin r.

- Let σ be a common random string. Setup(1^n) outputs (mpk = (σ, c_1, c_2), msk = r_1), where c_1 = Com(0; r_1) and c_2 = Com(0[id]; r_2).
- KeyGen(msk, id) produces a proof π = P(σ, x_{id}, s) for the following language L: x ∈ L if there exists s such that

\[
\begin{align*}
&c_1 = \text{Com}(0; s) \lor (c_2 = \text{Com}(\text{id}^*; s) \land \text{id}^* \neq \text{id}) \\
\text{Type I witness} \quad \text{Type II witness}
\end{align*}
\]

- Let P_{id,m} be a program which outputs m if V(σ, x_{id}, π_{id}) = 1 or outputs ⊥ otherwise. Enc(mpk, id, m) outputs iO(P_{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 = Com(id^*; r_2), relying on the hiding property of the commitment scheme.

\(H_2\) : We switch to the Type II witness using π_{id} \forall i ∈ [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 = Com(1; r_1).

20-3