CS6115: Structure versus Hardness in Cryptography	September 29, 2021
Lecture 16 : Other Applications iO	
Lecturer: *Shweta Agrawal*	Scribe: *Saswata Mukherjee*

1 Intoduction

Till now, we have seen deniable encryption, using indistinguishability obfuscation (Sahai-Waters paper). After that, public key encryption has been constructed, from secret key encryption, using $i\mathcal{O}$. In this lecture, we are going to see (from [20]), $i\mathcal{O}$ construction for P, from $i\mathcal{O}$ for NC^1 and fully homomorphic encryption scheme, with decryption in NC^1.

1.1 Review of $i\mathcal{O}$ and FHE

Definition 1.1 (Homomorphic Encryption)**.**
A homomorphic encryption scheme is defined as four-tuple of ppt algorithms $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$.

1. $\mathsf{KeyGen}(1^\lambda)$: It gives output $(\mathsf{pk}, \mathsf{sk}, \mathsf{evk})$ the public key, secret key and evaluation key.

2. $\mathsf{Enc}(\mathsf{pk}, m)$: If m is a messege then it outputs some ciphertext c.

3. $\mathsf{Dec}(\mathsf{sk}, c)$: It outputs some messege bit m_1 from ciphertext c.

4. $\mathsf{Eval}(\mathsf{evk}, f, c_1, \ldots, c_l)$: For some function f and ciphertexts $c_1, \ldots, c_l$ it outputs $f(c_1, \ldots, c_l) = c_f$.

Correctness: The above scheme is correct if

$$\mathsf{Pr}[\mathsf{Dec}_{sk}(\mathsf{Eval}_{evk}(f, c_1, \ldots, c_l)) \neq f(m_1 \ldots, m_l)] = negl(\lambda)$$

where $c_i \leftarrow \mathsf{Enc}(pk, m_i) \;\; \forall i$.
Compactness: The scheme is compact if the size of $\mathsf{Eval}_{evk}(f, c_1, \ldots, c_l)$ is bounded by $\mathsf{poly}(\lambda)$ bits and it is independent of the function and number of inputs to it.

Definition 1.2 (Indistinguishability Obfuscator)**.**
Indistinguishability Obfuscator $i\mathcal{O}$ is a ppt algorithm so that:
Given input some program P_0, $i\mathcal{O}(P_0)$ satisfies:

1. $i\mathcal{O}(P_0)$ can be computed in time polynomial over the description of P_0.

2. $i\mathcal{O}(P_0)$ preserves functionality.

3. For any ppt adversary A and programs P_1, P_2, with equal complexity and functionality,

$$|\mathsf{Pr}[A(i\mathcal{O}(P_1)) = 1] - \mathsf{Pr}[A(i\mathcal{O}(P_2)) = 1]|$$

is negligible.

2 Construction

Say, $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ is our given FHE scheme, so that, it's decryption circuit is in NC^1. In the following procedure, we construct $i\mathcal{O}$ for a circuit C with polynomial depth.

1. Say, λ is our security parameter and $(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{KeyGen}(1^\lambda)$, $(\mathsf{pk}_2, \mathsf{sk}_2) \leftarrow \mathsf{KeyGen}(1^\lambda)$.
2. Encrypt C using two public keys. Say, $e_1 \leftarrow \mathsf{Enc}_{\mathsf{pk}_1}(C)$, $e_2 \leftarrow \mathsf{Enc}_{\mathsf{pk}_2}(C)$.
3. Now, we obfuscate of a program $P \in \mathrm{NC}^1$ and description of P is given below.

For $i\mathcal{O}$ construction of C, we have to assure two main points. The first one is $i\mathcal{O}(C)$ should hide C and from the encryption of the circuit, we have the security guarantee. Now, we need to make sure that it preserves functionality.
Although Dec circuit is in NC^1, we cannot simply give the obfuscated Dec circuit publicly. As, in that case, by computing $\mathsf{Dec}(\widehat{C})$, the circuit C can be easily recovered (Here, $\widehat{C}$ is the encryption of circuit C).

So, we describe the program $P \in \mathbf{NC}^1$ by following procedure.

Here, we define **Universal Circuit** U. Given a circuit C and some input x, $U(C, x)$ computes $C(x)$. Let, U_x be the circuit $U(., x)$ where x is hard-wired.
From definition of $\mathsf{FHE.Eval}$, $\mathsf{Eval}_{\mathsf{evk}}(U_x, \widehat{C}) = \widehat{C(x)}$. So, we want to assure that the input to the obfuscated Dec circuit was computed as $\mathsf{Eval}_{\mathsf{evk}}(U_x, \widehat{C})$, for some x.
Say, R_1, R_2 are the circuits, which compute $y_1 \leftarrow \mathsf{Eval}_{\mathsf{evk}_1}(U_x, e_1)$ and $y_2 \leftarrow \mathsf{Eval}_{\mathsf{evk}_2}(U_x, e_2)$ respectively. Let π_1 and π_2 are the values of internal wires of circuits R_1 and R_2, on input x.

Define $P = P_{\mathsf{pk}_1,\mathsf{pk}_2,\mathsf{sk}_1,\mathsf{e}_1,\mathsf{e}_2}$, on input $(x, y_1, y_2, \pi_1, \pi_2)$ in the following way:

1. Check whether $R_1(x) = y_1$ and $R_2(x) = y_2$ by using π_1, π_2.
2. if condition (1) is satisfied then output $\mathsf{Dec}(\mathsf{sk}_1, y_1)$.

$\mathsf{FHE.Dec}$ circuit is in NC^1 and as we have π_1, π_2, we can check if $R_1(x) = y_1$ and $R_2(x) = y_2$ by using log depth circuit. Hence, $P_{\mathsf{pk}_1,\mathsf{pk}_2,\mathsf{sk}_1,\mathsf{e}_1,\mathsf{e}_2} \in \mathrm{NC}^1$. We can obfuscate $P_{\mathsf{pk}_1,\mathsf{pk}_2,\mathsf{sk}_1,\mathsf{e}_1,\mathsf{e}_2}$ by our known $i\mathcal{O}$ for NC^1 circuit. In this way we assure functionality.

Proof of Security:

We prove indistinguishability property for this construction, by hybrid argument. Here the $i\mathcal{O}$ for P adversary challenger gives two circuits $C_0, C_1 \in$ P and it receives $i\mathcal{O}(C_b)$ for some random $b \in \{0, 1\}$. It has to guess b.
And for the reduction from one hybrid to another, $i\mathcal{O}$ for NC^1 challenger or the $\mathcal{FHE}$ challenger is invoked. Through these hybrids, we transform obfuscation of C_0 to obfuscation of C_1 and H_{i+1} is indistinguishable from H_i for all i.

1. H_0 : The real world with $e_1 = \mathsf{Enc}_{\mathsf{pk}_1}(C_0)$, $e_2 = \mathsf{Enc}_{\mathsf{pk}_2}(C_0)$ with obfuscated $P_{\mathsf{pk}_1,\mathsf{pk}_2,\mathsf{sk}_1,\mathsf{e}_1,\mathsf{e}_2}$.
2. H_1 : Here we make $e_2 = \mathsf{Enc}_{\mathsf{pk}_2}(C_1)$, using FHE security.

3. H_2 : $P_{\mathsf{pk}_1,\mathsf{pk}_2,\mathsf{sk}_1,\mathsf{e}_1,\mathsf{e}_2}$ is changed to $P_{\mathsf{pk}_1,\mathsf{pk}_2,\mathsf{sk}_2,\mathsf{e}_1,\mathsf{e}_2}$ (it uses sk_2 instead of sk_1), using $i\mathcal{O}$ security.
4. H_3 : Make $e_1 = \mathsf{Enc}_{\mathsf{pk}_1}(C_1)$ using FHE security.
5. H_4 : Change P to $P_{\mathsf{pk}_1,\mathsf{pk}_2,\mathsf{sk}_1,\mathsf{e}_1,\mathsf{e}_2}$ again, using $i\mathcal{O}$ security.
   And here we arrive at real world with C_1.

References

[20] Lecture 20. Using indistinguishability obfuscation. `https://people.eecs.berkeley.edu/~sanjamg/classes/cs276-fall14/scribe/lec20.pdf`.