6.5630 Advanced Topics in Cryptography

Problem Set 2

Due: November 25, 2024

The focus of this problem set is a notion called *differing inputs obfuscation* (*diO*). Informally, it strengthens indistinguishability obfuscation (*iO*) by guaranteeing that obfuscations of two circuits are computationally indistinguishable as long as it is hard to find an input on which the two circuits output different values. In other words, if there is a distinguisher that distinguishes between $diO(C)$ and $diO(C')$, then there is an algorithm that finds an input x such that $C(x) \neq C'(x)$.

Definition 1 (Distributions on Pseudo-equivalent Circuits)**.** *Let $\mathcal{D}_n$ be a sequence of distributions (indexed by n) on pairs of n-input circuits (C, C') of the same size $s = s(n)$. We say $\mathcal{D}_n$ is a* PEC-distribution *if for every probabilistic polynomial-time (p.p.t.) algorithm B, we have*

$$\Pr[C(x) \neq C'(x) \mid (C, C') \leftarrow \mathcal{D}_n;\ x \leftarrow B(C, C')] \leq |C|^{-\omega(1)}.$$

That is, the probability that a p.p.t. algorithm B finds a differing input in a pair of randomly chosen circuits (C, C') from the distribution $\mathcal{D}_n$ is small.

Definition 2 (Differing Inputs Obfuscation)**.** *An indistinguishability obfuscator $\mathcal{O}$ is a differing inputs obfuscator for a PEC-distribution $\mathcal{D}_n$ if $\mathcal{O}(C_0)$ and $\mathcal{O}(C_1)$ are computationally indistinguishable when $(C_0, C_1) \leftarrow \mathcal{D}_n$. We say $\mathcal{O}$ is a* differing inputs obfuscator *(without specifying a distribution) if it is a differing inputs obfuscator for every PEC-distribution.*

Notes on the security parameter:

- When we say two distributions D_1 and D_2 are computationally indistinguishable (denoted $D_1 \approx_c D_2$), we mean that any adversary that runs in time polynomial in λ has at most a negligible in λ advantage in distinguishing between D_1 and D_2.
- In this problem set, we omit writing the security parameter for obfuscation algorithms $\mathcal{O}$, letting $\mathcal{O}(C)$ denote $\mathcal{O}(C, 1^{\lambda=|C|})$. (If one wishes to have a larger security parameter, one can always pad C with extra dummy gates, which do not change the functionality.)

1 Baby *diO*

It is an open question whether a differing inputs obfuscator exists. Here, we ask you to prove a special case.

(a) Assuming indistinguishability obfuscation exists, show that there exists an $\mathcal{O}$ such that for all PEC-distributions $\mathcal{D}_n$ satisfying

$$\max_n \max_{(C,C') \leftarrow \mathcal{D}_n} \left| \{x \in \{0,1\}^n \,:\, C(x) \neq C'(x)\} \right| \leq 1,$$

$\mathcal{O}$ is a differing inputs obfuscator for $\mathcal{D}_n$.

2 iO is Best Possible Obfuscation

There is a certain sense in which an indistinguishability obfuscator $\mathcal{O}$ is a "best possible" obfuscator, meaning roughly that if any obfuscator with a security property Π exists, then $\mathcal{O}$ (with a sufficient amount of padding) also has security property Π. Fully formalizing this is beyond the scope of this problem set, but we ask you to prove a special case.

For a circuit C and a natural number $q \geq |C|$, let $Pad(C, q)$ denote the circuit obtained by padding C with dummy gates to a obtain an equivalent circuit of size q.

(a) Assume there exists a differing inputs obfuscator. Show that for every indistinguishability obfuscator $\mathcal{O}$ there exists a polynomial p such that the algorithm $\mathcal{O}_p$ given by $\mathcal{O}_p(C) = \mathcal{O}(Pad(C, p(|C|)))$ is also a differing inputs obfuscator.

3 Put Your Money Where Your Mouth Is (Part 1)

Sometimes when it looks like we need a differing inputs obfuscator, we can get by using indistinguishability obfuscator and other cryptographic objects. The combination of the next three problems will give an example of this.

A confident cryptographer wants to offer a prize (perhaps some cryptocurrency) to anyone who can break certain cryptographic assumptions. Specifically, anyone who finds a non-zero x that hashes to the same value that the all-zeroes string hashes to (hence, breaking the security of the hash function) should be able to see a secret message. On the other hand, it should be such that, if certain cryptographic assumptions are true, it is infeasible for anyone to claim the prize.

You will show that some version of this is possible, if a differing inputs obfuscator exists. (Note: think about why witness encryption alone does not suffice.)

(a) Assume a differing inputs obfuscator exists. Assume $H_k : \{0,1\}^{2\lambda} \to \{0,1\}^\lambda$ is a collision resistant hash family indexed by $k \in \{0,1\}^\lambda$. Show there are polynomial time functions $\mathsf{Enc}(k, b)$ and $\mathsf{Dec}(x, C)$ such that

- **Functionality:** For all $k \in \{0,1\}^\lambda$ and $b \in \{0,1\}$ and for all non-zero $x \in \{0,1\}^{2\lambda}$ satisfying $H_k(x) = H_k(0^{2\lambda})$

$$\Pr[\mathsf{Dec}(x, \mathsf{Enc}(k, b)) = b] = 1.$$

- **Security:** When $k \leftarrow \{0,1\}^\lambda$ we have

$$(k, \mathsf{Enc}(k, 0))) \approx_c (k, \mathsf{Enc}(k, 1))).$$

4 Somewhere Statistically Binding Hash Functions

We would like to to replace diO in the last problem with iO. To do this, here (and in many applications) cryptographers use an "iO friendly" version of collision resistant hash functions called *somewhere statistically binding hash functions.*

Definition 3 (Somewhere Statistically Binding (SSB) Hash Functions)**.** *An SSB Hash Family consists of two polynomial time algorithms* Hash *and* Gen *with the following properties:*

- ***Key Generation:*** Gen *takes as input a security parameter λ and an index $i \in [2\lambda]$ and outputs a key k.*
- ***Shrinking:*** Hash *takes as input a key k and a string $x \in \{0,1\}^{2\lambda}$ and outputs a hash v of length at most λ. For succinctness, we write $H_k(x) = \mathsf{Hash}(k, x)$.*
- ***Binding on i:*** *If $k \leftarrow Gen(1^\lambda, i)$, then (with probability one) $H_k(x) \neq H_k(y)$ for all $x, y \in \{0,1\}^{2\lambda}$ that differ on their i'th bit. (Equivalently, there exists a (possibly inefficient) function that given $H_k(x)$, outputs the i'th bit of x.)*
- ***Binding Indistinguishable:*** *For all $i, j \in [2\lambda]$, we have that the outputs of $Gen(1^\lambda, i)$ and $Gen(1^\lambda, j)$ are computationally indistinguishable.*

We ask you to construct this object and show it is at least a collision resistant hash function.

(a) Show that if a fully homomorphic encryption scheme (you may use properties of the GSW scheme we saw in class if you wish) exists, then an SSB hash family exists.

(b) Show that every SSB hash family is also collision resistant hash family.

5 Put Your Money Where Your Mouth Is (Part 2)

Now we will accomplish problem 3, just using iO (assuming H_k is an SSB Hash function).

(a) Let H_k be an SSB Hash family. Assume indistinguishability obfuscation exists. Show there are polynomial time functions $\mathsf{Enc}(k, b)$ and $\mathsf{Dec}(x, C)$ such that

- **Functionality:** For all $k \in \{0,1\}^\lambda$ and $b \in \{0,1\}$ and for all non-zero $x \in \{0,1\}^{2\lambda}$ satisfying $H_k(x) = H_k(0^{2\lambda})$
$$\Pr[\mathsf{Dec}(x, \mathsf{Enc}(k, b)) = b] = 1.$$
- **Security:** When $k \leftarrow \mathsf{Gen}(1^\lambda, i = 1)$, we have
$$(k, \mathsf{Enc}(k, 0))) \approx_c (k, \mathsf{Enc}(k, 1))).$$

Hint: Try using hybrids that alternate between using iO security and binding indistinguishability until you can remove b from the circuit entirely.