Abstract

1 Preliminaries

1.1 Punctured PRF

A punctured PRF is a simple type of constrained PRF ([BW13, BGI14, KPTZ13]) where a PRF is well defined on all inputs except for a specified, polynomial-sized set. We will adopt the notion specified in [SW14].

Definition 1.1 (Punctured PRF)**.** A puncturable family of PRF s F mapping is given by a tuple of algorithms $(\text{Key}_F, \text{Puncture}_F, \text{Eval}_F)$. satisfying the following conditions:

- Functionality preserved under puncturing: For every PPT adversary $\mathcal{A}$, $S \subseteq \{0,1\}^n$ and every $x \in \{0,1\}^n$ where $x \notin S$, we have that

$$\mathbf{Pr}\left[\texttt{Eval}_F(K,x) = \texttt{Eval}_F(K_S,x) \mid K \leftarrow \texttt{Key}_F(1^\lambda), \texttt{K}_S = \texttt{Puncture}_F(K,S)\right] = 1.$$

- Pseudorandom at punctured points: For every PPT adversary $\mathcal{A}, \mathcal{B}$ such that $\mathcal{A}(1^\lambda)$ outputs a set S and state $\mathbf{st}$, consider an experiment where $K \leftarrow \texttt{Key}_F(1^\lambda)$ and $K_S = \texttt{Puncture}_F(K,S)$. Then, we have that

$$\left|\mathbf{Pr}\left[\mathcal{B}(\texttt{st}, K_S, S, \texttt{Eval}_F(K,S)) = 1\right] - \mathbf{Pr}\left[\mathcal{B}(\texttt{st}, K_S, S, U_{m\cdot|S|})\right]\right| \leq \texttt{negl}(\lambda).$$

1.2 Indistinguishable Obfuscation

We will use the definition of indistinguishable obfuscation as presented in [GGH+16].

Definition 1.2 (Indistinguishable obfuscation)**.** A uniform PPT machine $\mathcal{O}$ is an indistinguishable obfuscator for a class of circuits $\mathcal{C}$ if for every circuit $C \in \mathcal{C}$ we have that

$$\mathbf{Pr}[C'(x) = C(x) \mid C' \leftarrow \mathcal{O}(C)] \leq \texttt{negl}(\lambda)$$

and for any PPT distinguisher $\mathcal{D}$ and two pairs of circuits C_0, C_1 such that $C_0(x) = C_1(x)$ for all x, then

$$\left|\mathbf{Pr}\left[\mathcal{D}(\mathcal{O}(\lambda, C_0)) = 1\right] - \mathbf{Pr}\left[\mathcal{D}(\mathcal{O}(\lambda, C_1)) = 1\right]\right|.$$

Definition 1.3 (Homomorphic Indistinguishable Obfuscationf ([BKP23]))**.** We will use the definition of homomorphic indistinguishable obfuscation as presented in [BKP23]. Homomorphic indistinguishable obfuscation (H$i\mathcal{O}$) is a variation on indistinguishable obfuscation where an obfuscated circuit, C, can be composed with another circuit C' to produce an obfuscated circuit $C \circ C'$ that computes $C(x) \circ C'(x)$ for all x. As outlined in [BKP23], the size of the circuit remains polynomial after a polynomial number of compositions. Formally, an H$i\mathcal{O}$ scheme consists of the following three algorithms

- Obfuscate$(1^\lambda, C)$: Takes as input a circuit C and outputs an obfuscated circuit $\hat{C}$.
- Eval$(\hat{C}, x)$: Takes as input an obfuscated circuit $\hat{C}$ and an input x and outputs a string $y = C(x)$.
- Compose$(\hat{C}, C')$: Takes as input an obfuscated circuit $\hat{C}$ and a circuit C' and outputs an obfuscated circuit $\hat{C}'$ such that $\hat{C}'(x) = (C' \circ C)(x)$ for all x.

The scheme must satisfy standard notions of correctness and indistinguishably, though adopted to the homomorphic setting. Specifically, we require

- **Homomorphic Indistinguishablity**: For any $\lambda, k \geq 0$, and circuits $C_0^0, \ldots, C_k^0$ and $C_0^1, \ldots, C_k^1$, of size at most k where
$$C_k^0 \circ \cdots \circ C_0^0 = C_k^1 \circ \cdots \circ C_0^1,$$
then it holds that
$$\begin{aligned} & \text{Compose}(\cdots \text{Compose}(\text{Obfuscate}(1^\lambda, C_0^0), C_1^0), \cdots, C_k^0) \\ \overset{c}{\approx} \; & \text{Compose}(\cdots \text{Compose}(\text{Obfuscate}(1^\lambda, C_0^1), C_1^1), \cdots, C_k^1). \end{aligned}$$

2 DAG Label Obfuscation from Additive Overhead iO

2.1 DAG Randomized Traversal

Say that we have a sparse, potentially exponentially sized, graph $\mathcal{G} = (V, E)$ with polynomial depth D, and forall $v \in V, \deg(v) \leq d$. Moreover, for simplicity, assume that for all v,

$$\deg^{-1}(v) = \left| \{ u \in V \mid \exists j \in [d], \Gamma(u)_j = v \} \right| \leq d.$$

In words, there are at most d edges into a vertex. As a note, our construction just requires that $\deg^{-1}(\cdot) = O(1)$ but for the sake of simplicity we fix $\deg^{-1}(\cdot) \leq d$.

We also require that $\mathcal{G}$ is equipped with a neighbor function, Γ, which can be computed in polynomial time. We define a randomized and keyed labelling function $\phi : \{0,1\}^\lambda \times V \to \{0,1\}^{\text{poly}(\lambda)}$ such that given, $\phi(K, v_0)$ for root v_0, a PPT adversary which runs in time at most $T(\lambda)$, $\mathcal{A}$, which does not know a path from v_0 to v,

$$\mathbf{Pr}[\mathcal{A}(\mathcal{O}(C_\Gamma^S), v_0, v, \phi(K, v_0)) = \phi(K, v)] \leq \epsilon \tag{1}$$

for function C_Γ^S where $C_\Gamma^S(\phi(K, u)) = \phi(K, \Gamma(u)_1), \ldots, \phi(K, \Gamma(u)_d)$ and the circuit is padded to size S. if $\Gamma(u) \neq \emptyset$ and otherwise $\Gamma(u)$ returns a $\perp$ string. We fix the adversary's advantage to $\epsilon < \text{poly}(\lambda)$ and runtime to $T(\lambda) \leq \text{poly}(\lambda, \frac{1}{\epsilon})$ as we will need to show that a set of a potentially exponential number of games *does not have exponential security loss* nor or *reduce down to security against an exponentially strong adversary.*

2.2 Instantiation

We define $\phi(K, v) = F(K, v)$ for $K \xleftarrow{\$} \{0,1\}^\lambda$, and we can now define our neighbor function C_Γ^S.

$$C_\Gamma^S(\phi(K, v), v) = C_1 \circ \cdots \circ C_S \circ \ C_\Gamma(\phi(K, v)) \tag{2}$$

where C_Γ is defined in Algorithm 1 and $C_1, \ldots C_S$ is C_P as defined in Algorithm 2 or a simple circuit which will be defined within context. We will use the shorthand $\text{H}i\mathcal{O}(C_\Gamma^S)$ to denote $C_P \circ \cdots \circ C_P \circ \ \text{H}i\mathcal{O}(C_\Gamma)$.

Algorithm 1 The circuit for the neighbor function, C_Γ.

```
function C_Γ(X, v)
    if f(X) ≠ f(F(K, v)) then
        return ⊥
    if Γ(v) = ∅ then
        return ⊥
    u_1, ... u_d = Γ(v)
    return F(K, u_1), F(K, u_2), ..., F(K, u_d)
```

Algorithm 2 The circuit for the padding function, C_P where the circuit size is q.

```
function C_P(x)
    return x
```

Theorem 2.1 (Label Extractibility)**.** *Given an HiO scheme, then for all $v \in V$, and uniform fixed polynomial sized extractor E circuit, we have that if there exists a PPT adversary $\mathcal{A}$ such that*

$$\mathbf{Pr}[\mathcal{A}(HiO(C_\Gamma^S), v_0, v, \phi(K, v_0)) = \phi(K, v)] > \epsilon \tag{3}$$

then

$$\mathbf{Pr}[E(\mathcal{A}, HiO(C_\Gamma^S), v_0, v, \phi(K, v_0)) = P] > \mathtt{negl}(\lambda) \tag{4}$$

where ϵ is a fixed advantage such that $\epsilon < \text{poly}(1/\lambda)$, P is a path from v_0 to v in $\mathcal{G}$, and $S = O(\mathtt{dep} \cdot d)$.

2.3 Proof of theorem 2.1

Intuitively, showing how to construct an extractor given an adversary with advantage seems to be an intractable problem. Rather than approach the proof in this way, we will show the contrapositive. I.e. we will assume that there does not exist a uniform extractor for a fixed $\mathcal{A}$ and graph instance and then show that such an adversary has negligible advantage. We will do this by making an inductive argument on the graph and parameter S, finally giving us theorem 2.1 for $S \geq O(\mathtt{dep} \cdot d)$.

We will first define some useful notation. Let $H_\mathcal{I} \subset \mathbb{Z}^+ \times V$ be a "hard" set of composition depth and vertices for the $\mathcal{I}$-th step of induction such that $\mathcal{A}$ cannot, with non-negligible probability, produce $\phi(K, h)$ where $(s, h) \in H_\mathcal{I}$. More formally, for any fixed uniform $\mathcal{V}$,

$$\mathbf{Pr}\left[\mathcal{V}\left(\mathcal{A}, \mathrm{H}i\mathcal{O}(C_\Gamma^s), v_0, h, \phi(K, v_0)\right) = \phi(K, h)\right] \leq \mathtt{negl}(\lambda).$$

Further, let $\mathcal{P}$ be the set of all paths from v_0 to v in $\mathcal{G}$.

Inductive Proof

Now, we will start with the base case. The base case is in essence non-cryptographic and holds for any $S \in \mathbb{Z}$.

Lemma 2.2 (Base Case Game)**.** *Assuming that there is no extractor E such that*

$$\mathbf{Pr}[E(\mathcal{A}, HiO(C_\Gamma^S), v_0, v) = P] \geq \frac{1}{p(\lambda)}$$

where $P \in \mathcal{P}$, then for any PPT $\mathcal{A}$, there exists some graph cut $C_E \subset E$ which separates v_0 and v and a set C_V such that

$$\mathbf{Pr}[\mathcal{A}(\mathcal{O}(C_\Gamma^S), v_0, v, \phi(K, v_0)) \in \phi(K, C_V)] < \epsilon \tag{5}$$

for any $S \geq \text{poly}(\lambda)$. We have

$$H_0 = \mathbb{Z}^+ \times \{\, u \mid (w, u) \in C_E \text{ and } u \in T \,\} \bigcup \{\, w \mid (w, u) \in C_E \text{ and } w \in T \,\}.$$

where $V = S \bigcup T$ is a partition of V such that $v_0 \in S$ and $v \in T$. In words, for any composition depth, H_0 are the vertices just adjacent to the cut and on the same side as v.

Proof. We will show that if $\mathcal{A}$ can break eq. (5), then we can construct an extractor, E, which finds a path from v_0 to v with non-negligible probability.

Assume that for every possible cut, $\mathcal{A}$ is able to produce a single label in this cut for a vertex w. Then, we note that there must be at least 1 path from v_0 to w and from w to v as otherwise, w would not be in the cut. Moreover, we note that $\mathcal{A}$ must be able to produce a label for all vertices on at least one path from v_0 to w as otherwise, we can change the cut to include the edges between where $\mathcal{A}$ is able to produce a label and not able to produce a label. Using the same argument, we can show that $\mathcal{A}$ must be able to produce all labels on a path from w to v.

Note that $\mathcal{A}$ is not given the specific cut C_E but rather C_E is chosen based off of the adversary. So, we can build an extractor to do the following:

1. Create an iO obfuscated circuit with a random key, K', for C^S_Γ and create circuit $\mathcal{O}(C^S_\Gamma)$ as well as $\phi(K', v_0)$
2. Run $\mathcal{A}(\mathcal{O}(C^S_\Gamma), v_0, v, \phi(K', v_0))$ to get all labels $\phi(K', v_0), \dots \phi(K', v)$ for some path from v_0 to v.
3. Recreate the path from v_0 to v via checking which vertex matches to adjacent labels in the path: I.e. starting with $\ell = 0$, we can learn the $\ell + 1$ vertex via finding $j \in [d]$ such that $C^S_\Gamma(\phi(K', v_\ell), v_\ell)_j \in \{\, \phi(K', v_0), \dots, \phi(K', v) \,\}$ and then setting $v_{\ell+1} = \Gamma(v_\ell)_j$.

□

We can look at lemma 2.2 as a "base case" of sorts. We now inductively build up a series of games such that $\mathcal{A}$ cannot find any label in V_1 where V_1 are the vertices on side of the cut (as defined in lemma 2.2) which contain v.

Lemma 2.3 (Inductive Game Hypothesis)**.** *Assuming adaptive security of constrained PRFs, one way functions, the existence of homomorphic indistinguishable obfuscation, and the hardness of producing a label for $h \in H_{\mathcal{I}}$, then we have that*

$$H_{\mathcal{I}+1} = \{H_{\mathcal{I}} \cup \{\, (s, \Gamma(v)) \mid (s, v) \in H_{\mathcal{I}} \,\}\} \bigcap \{\, [2(\mathcal{I}+1) \cdot d, \infty) \times V \,\}.$$

Proof. We are going to use a series of indistinguishable hybrids along with the circuit defined in 3 to show the above

- $\texttt{Hyb}_0$: In the first hybrid, the following game is played
  1. $K \leftarrow \{0,1\}^{\lambda'}$ and $\phi(K, v_0) = (F(K, v_0), v_0)$ where K is some fixed secret drawn from a uniform distribution
  2. The challenger generates $\text{Hi}\mathcal{O}(C^S_\Gamma)$ and gives the program to $\mathcal{A}$
  3. The challenger gives the adversary h^* in plaintext.
  4. $\mathcal{A}$ outputs guess g and wins if $g = \phi(K, h^*)$
- $\texttt{Hyb}_1$: We replace C^S_Γ with the punctured circuit defined in algorithm 3. Fix the constant $z^* = f(F(K, w^*))$
- $\texttt{Hyb}_{2,1}$ We replace circuit 3 with circuit 4 where we set $Y^* = (1, y)$ such that $\Gamma(y)_1 = h^*$. So then, we have that have $F(K, \Gamma(y)_1) = \perp$. Moreover, we set the punctured set, S to $\emptyset$ (i.e. we do not puncture the PRF).

- $\texttt{Hyb}_{2,j}$ for $j \in \{2, \ldots, \deg^{-1}(h^*)\}$ We replace Y^* with $Y^* \cup (j, y)$ such that $\Gamma(y)_j = h^*$. Note after the last of these hybrids, we have that $F(K, h^*)$ is always set to $\perp$ on line 10 of circuit 4.
- $\texttt{Hyb}_3$: We puncture the PRF at h^* and set $S = \{h^*\}$.
- $\texttt{Hyb}_4$: Set $z^* = f(t)$ where t is chosen at random

Finally, we can note that if $\texttt{Hyb}_0 \overset{c}{\approx} \texttt{Hyb}_4$, then

$$\mathbf{Pr}[\mathcal{A}(C_\Gamma^S, v_0, h^*, \phi(K, v_0)) = \phi(K, h^*)] \overset{c}{\approx} \mathbf{Pr}[\mathcal{A}(C_\Gamma^{S^*}, v_0, h^*, \phi(K, v_0)) = \phi(K, h^*)]$$

where $C_\Gamma^{S^*}$ is the circuit which sets z^* to be the image on a one way function of a randomly chosen point. As we will show in lemma 2.4, lemma 2.5, and lemma 2.7, an adversaries advantage between games in $\texttt{Hyb}_0$ and $\texttt{Hyb}_3$ is at most $\epsilon/2$. Thus, if $\mathcal{A}$ can produce $\phi(K, v) = (\sigma_v, v)$ with advantage $\epsilon/2$ in $\texttt{Hyb}_3$, then $\mathcal{A}$ can find a pre-image for z^* under f with non-negligible probability and thus break the security of a one way function. We then have that the advantage of the adversary in $\texttt{Hyb}_0$ cannot be more than ϵ. □

Lemma 2.4. *$\texttt{Hyb}_0$ and $\texttt{Hyb}_1$ are distinguishable with advantage at most $\epsilon/10$.*

Proof. Assume towards contradiction that $\epsilon \in \text{poly}(1/\lambda)$. Note that for all inputs (z, v) to C_Γ^S as defined in circuit 1 and circuit 3 are equivalent and thus indistinguishable by the definition of indistinguishable obfuscation. So, if $\epsilon \in \text{poly}(\lambda)$, then an adversary cannot distinguish the hybrids with probability more than $\epsilon/10$. □

Lemma 2.5. *Each hybrid from $\texttt{Hyb}_1$ to $\texttt{Hyb}_{2,1}$ and $\texttt{Hyb}_{2,j-1}$ to $\texttt{Hyb}_{2,j}$ for $j \in 2, \ldots, \deg^{-1}(w^*)$ is distinguishable with advantage at most $\epsilon/(10d)$. Thus, $\texttt{Hyb}_1$ and $\texttt{Hyb}_{2,\deg^{-1}(w^*)}$ are distinguishable with advantage at most $\epsilon/10$.*

Proof. This proof will be a modification of the proof in [IPS15] for the simple case of weak extractible obfuscation. The key idea lies on two observations:

1. We can go from $\texttt{Hyb}_{2,j-1}$ (or $\texttt{Hyb}_1$) to a "padded out" version of $\texttt{Hyb}_{2,j}$ by composing C_Γ^S with a circuit which returns $\perp$ for the j-th input.
2. We can go from $\texttt{Hyb}_{2,j-1}$ (or $\texttt{Hyb}_1$) to a re-randomized, slightly larger, version of itself by composing $\text{Hi}\mathcal{O}(C_\Gamma^S)$ with C_{iden} (the identity circuit).
3. If an adversary can produce $\texttt{Hyb}_{2,j}$ solely from $\texttt{Hyb}_{2,j-1}$ (or $\texttt{Hyb}_1$) and can distinguish them with advantage at least $\epsilon/10d$, then we can build an adversary, $\mathcal{B}$, which can produce a label $\phi(K, h)$ for $h \in H$ in $\texttt{Hyb}_0/\texttt{Hyb}_1$.

For simplicity, say that the input size to all of our circuits is n. Let $s \in \mathbb{Z}$ such that $s \geq 2 \cdot d \cdot \mathcal{I}$. Also, let C_0^s be the circuit from the first hybrid and C_1^s the one from the second.

At a high level, we will show that an adversary can create a *larger version* of C_1 given only C_0. Then, if an adversary can distinguish between C_0^{s+2} and C_1^{s+2}, the adversary can recover a label in $H_\mathcal{I}$ for instances of C_0^s.

First, given C_0^s, $\mathcal{A}$ can construct a larger version of C_1^{s+1} by composing C_0^s with a program that returns $\perp$ for the j-th input of h^0 on line 10. $\mathcal{A}$ can also produces C_0^{s+1} from C_0^s via composing C_0^s with the identity circuit.

Now, assume towards contradiction that there exists an adversary $\mathcal{A}$ that can distinguish C_0^{s+2} and C_1^{s+2} with advantage $\epsilon' > \epsilon/10d$ in $O(T')$ time. Let C_i^{Mid} be a circuit such that

$C_i^{\text{Mid}}(X) = C_0^{s+2}(X)$ if $X_i = 0$ and $C_i^{\text{Mid}}(X) = C_1^{s+2}(X)$ if $X_i = 1$.
We can see that C_0^{s+1} and C_1^{s+1} differ on at most 1 input which we will call α. Then, $C_i^{\text{Mid}} = C_0^{s+2}$ if $\alpha_i = 0$ and $C_i^{\text{Mid}} = C_1^{s+2}$ if $\alpha_i = 1$. So, if we build an adversary $\mathcal{B}$ which can tell if $C_i^{\text{Mid}} = C_0^{s+2}$ or $C_i^{\text{Mid}} = C_1^{s+2}$ with probability γ, we have that $\mathcal{B}(C_0^{s+2}, C_1^{s+2})$ can be used to check if α_i is 0 or 1 with probability γ. So then, $\mathcal{B}$ can be used to learn $\phi(K, \alpha)$ with probability at least γ^n. Because $(s, \alpha) \in H$, we get that $\mathcal{B}$ can break the inductive hypothesis!

To build $\mathcal{B}$ to tell if $C_i^{\text{Mid}} = C_0$ or C_1 with probability $\gamma^n \geq \epsilon$, we will make oracle calls to $\mathcal{A}$:

1. Run $I = \left\lceil \frac{12(\ln 2 + \ln n - \ln(1-\epsilon) + \ln 2)}{\epsilon'} \right\rceil$ iterations of the following experiment to estimate advantage ϵ'_b for $b \in \{0, 1\}$
   (a) Sample an obfuscation of C_b^{s+2} by composing C_b^{s+1} with the identity circuit.
   (b) Sample an obfuscation of C_i^{Mid} by composing a function which will return $\perp$ on line 10 if $\alpha_i = 0$ and otherwise return the output of C_0^{s+1}.
   (c) Have $\mathcal{A}$ distinguish between C_b^{s+2} and C_i^{Mid}
   (d) Output 1 if successful.

   Note that we can estimate ϵ'_b as the number of successful runs, which we will denote $\sum_{j \in [I]} \text{succ}_{i,j}$, divided by I.
2. If $\epsilon'_1 > \epsilon'_0$, then $C^{\text{Mid}} = C_0$, otherwise, $C^{\text{Mid}} = C_1$.

Note that $\mathcal{B}$ runs in time $O(T'I)$. So, if we set the upper-bound on the runtime of the adversary in eq. (1) to $O(T'I)$), then $\mathcal{B}$ can learn $\phi(K, y)$ with probability $\gamma^n \geq \frac{\epsilon}{10d}$.
We differ the proof that I is the correct choice of parameter to appendix A. □

Lemma 2.6. *The game in* $\texttt{Hyb}_{2,\deg^{-1}(w^*)}$ *is indistinguishable from* $\texttt{Hyb}_3$ *with probability at most* $\epsilon/10$.

Proof. As with lemma 2.4, the indistinguishably follows directly from the definition of indistinguishable obfuscation. □

Lemma 2.7. *The game in* $\texttt{Hyb}_3$ *is indistinguishable from* $\texttt{Hyb}_4$.

Proof. Assume towards contradiction that $\epsilon \in \text{poly}(1/\lambda)$. We now show that if the advantage of $\mathcal{A}$ is greater than $\epsilon/10$, then we can create a reduction, $\mathcal{B}$, which can break the security of the PRF at the punctured point. $\mathcal{B}$ first chooses a message w^* and submits this to the constrained PRF challenger and gets back the punctured PRF key $K(\{w^*\})$ and challenge a. $\mathcal{B}$ then runs the experiment in $\texttt{Hyb}_{2,\deg^{-1}(w^*)}$ except that $z^* = f(a)$. If a is the output of the PRF, then we are in $\texttt{Hyb}_{2,\deg^{-1}(w^*)}$, if a is the output of a random function, then we are in $\texttt{Hyb}_3$. □

Algorithm 3 Circuit for the neighbor function, C_Γ^S with PRF key K and constant w^*, z^*

function $C_\Gamma^S(X, v)$
  if $v \neq w$ and $f(X) \neq f(F(K, v))$ **then**
    return $\perp$
  if $v = w$ and $f(X) \neq z^*$ **then**
    return $\perp$
  if $\Gamma(v) = \emptyset$ **then**
    return $\perp$
  $u_1, \ldots u_d = \Gamma(v)$
  return $F(K, u_1), F(K, u_2), \ldots, F(K, u_d)$

Algorithm 4 Circuit for the neighbor function, C_Γ^S with punctured PRF key $K(S)$ and constant w^*, Y^*, J^*, z^*

function $C_\Gamma^S(X, v)$
  if $v \neq w$ and $f(X) \neq f(F(K, v))$ **then**
    return $\perp$
  if $v = w$ and $f(X) \neq z^*$ **then**
    return $\perp$
  if $\Gamma(v) = \emptyset$ **then**
    return $\perp$
  $u_1, \ldots u_d = \Gamma(v)$
  while $\exists j^* \in [d], (j^*, u_j) \in Y^*$ **do**
    Set $F(K, u_{j^*}) = \perp$
  return $F(K, u_1), F(K, u_2), \ldots, F(K, u_d)$

References

[BGI14] Elette Boyle, Shafi Goldwasser, and Ioana Ivan. Functional signatures and pseudorandom functions. In *International workshop on public key cryptography*, pages 501–519. Springer, 2014. 1.1

[BKP23] Kaartik Bhushan, Venkata Koppula, and Manoj Prabhakaran. Homomorphic indistinguishability obfuscation and its applications. *Cryptology ePrint Archive*, 2023. 1.3

[BW13] Dan Boneh and Brent Waters. Constrained pseudorandom functions and their applications. In *Advances in Cryptology-ASIACRYPT 2013: 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part II 19*, pages 280–300. Springer, 2013. 1.1

[GGH+16] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. *SIAM Journal on Computing*, 45(3):882–929, 2016. 1.2

[IPS15] Yuval Ishai, Omkant Pandey, and Amit Sahai. Public-coin differing-inputs obfuscation and its applications. In *Theory of Cryptography: 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II 12*, pages 668–697. Springer, 2015. 2.5

[KPTZ13] Aggelos Kiayias, Stavros Papadopoulos, Nikos Triandopoulos, and Thomas Zacharias. Delegatable pseudorandom functions and applications. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 669–684, 2013. 1.1

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 475–484, 2014. 1.1

A Proof of Parameters in Lemma 2.5

As a reminder, we set $I = \left\lceil \frac{12(\ln 2 + \ln n - \ln(1-\epsilon) + \ln 2)}{\epsilon'} \right\rceil$ where I is the number of iterations of the experiment define in lemma 2.5.

WLOG, say that $C^{\text{Mid}} = C_0$, then

$$\begin{aligned}\gamma = Pr[\epsilon_1' > \epsilon_0'] &= \mathbf{Pr}\left[\sum_{j\in[I]} \text{succ}_{1,j} > \sum_j \text{succ}_{0,j}\right] \\ &\geq \mathbf{Pr}\left[\sum_{j\in[I]} \text{succ}_{1,j} > \frac{I\epsilon'}{2}\right] \cdot \mathbf{Pr}\left[\sum_{j\in[I]} \text{succ}_{0,j} < \frac{I\epsilon'}{2}\right].\end{aligned}$$

We then have that

$$\mathbf{Pr}\left[\sum_j \text{succ}_{1,j} > I\epsilon' \cdot \tfrac{1}{2}\right] \geq 1 - \exp\left(-\frac{I\epsilon'}{2^2 \cdot 3}\right) = 1 - \exp\left(-\frac{I\epsilon'}{12}\right). \quad \text{(by the Chernoff bound)}$$

And, if iO distinguishing advantage is at most α and $\delta = \frac{\epsilon'}{2\alpha} - 1$

$$\begin{aligned}\mathbf{Pr}\left[\sum_j \text{succ}_{0,j} < \frac{I\epsilon'}{2}\right] = 1 - \mathbf{Pr}\left[\sum_j \text{succ}_{0,j} \geq (1+\delta)I\alpha\right] &\geq 1 - \exp\left(-I\alpha\left(\frac{\epsilon'}{2\alpha} - 1\right)^2 \cdot \frac{1}{3}\right) \\ &\qquad \text{(by the Chernoff bound)} \\ \geq 1 - \exp\left(-\frac{I\epsilon'^2}{12\alpha}\right) \geq 1 - \exp\left(-\frac{I\epsilon'}{12}\right).. &\qquad \text{(as } \epsilon' > \alpha\text{)}\end{aligned}$$

So we finally have that

$$\mathbf{Pr}[\epsilon_1' > \epsilon_0'] \geq 1 - \exp\left(-\frac{I\epsilon'}{12}\right) - \exp\left(-\frac{I\epsilon'}{12}\right) \geq 1 - 2\exp\left(-\frac{I\epsilon'}{12}\right). \tag{6}$$

Setting $I \geq \frac{12(\ln 2 + \ln n - \ln(1-\epsilon) + \ln 2)}{\epsilon'} \in \text{poly}(n, 1/\epsilon, 1/\epsilon')$, we have that

$$\begin{aligned}\gamma^n &\leq \left(1 - 2\exp\left(-\frac{I\epsilon'}{12}\right)\right)^n \\ &\leq 1 - 2n \cdot \exp\left(-\frac{I\epsilon'}{12}\right) = 1 - 2n \cdot \exp\left(-\ln n + \ln(1-\epsilon) - \ln 2\right) \\ &= 1 - (1-\epsilon) = \epsilon\end{aligned}$$

as desired.