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

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

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 ${C_\Gamma}^S$ 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 ${C_\Gamma}^S$:

Algorithm 1 The circuit for the neighbor function, ${C_\Gamma}^S$ padded out to size S.

```
function C_Γ^S(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)
```

We are going to show that eq. (1) for $S = O(D \cdot \text{overhead})$ where "overhead" is the additive overhead of $i\mathcal{O}$ obfuscation. We will do this by first showing that the non-existence of an extractor to find a path from v_0 to v implies that $\mathcal{A}$ necessarily does not know $\phi(K, c)$ for a $c \in C_V \subset V$ where the vertices in C_V border a graph cut which separates v_0 and v. Note that the base case holds for all $S \geq \text{poly}(\lambda)$.

Then, we inductively build up a series of games to show that $\mathcal{A}$ cannot learn *any* $\phi(K, v)$ for $v \in V_1$ where V_1 are the vertices on the side of the cut containing v. At each inductive step, we restrict the security game to hold for $S \geq O(\mathcal{I} \cdot \text{overhead})$ where $\mathcal{I}$ is the number of calls to induction.

Lemma 2.1 (Base Case Game)**.** *Assuming that there is no extractor E such that* $\mathbf{Pr}[E(\Gamma, v_0, v) = P] \geq \frac{1}{p(\lambda)}$ *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

$$\mathbf{Pr}[\mathcal{A}(\mathcal{O}({C_\Gamma}^S), v_0, v, \phi(K, v_0)) \in \phi(K, C_V)] < \epsilon. \tag{2}$$

We define $C_V \subset V$ to be

$$\{\, u \mid (w,u) \in C_E \text{ and } u \text{ on the side of } v \,\} \bigcup \{\, w \mid (w,u) \in C_E \text{ and } w \text{ on the side of } v \,\}.$$

In words, C_V 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. (2), 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 ${C_\Gamma}^S$ and create circuit $\mathcal{O}({C_\Gamma}^S)$ as well as $\phi(K', v_0)$
2. Run $\mathcal{A}(\mathcal{O}({C_\Gamma}^S), v_0, v, \phi(K', v_0))$ to get all labels $\phi(K', v_0), \ldots \phi(K', v)$ 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 ${C_\Gamma}^S(\phi(K', v_\ell), v_\ell)_j \in \{\, \phi(K', v_0), \ldots, \phi(K', v) \,\}$ and then setting $v_{\ell+1} = \Gamma(v_\ell)_j$.

□

We can look at lemma 2.1 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 2.1) which contain v.

Lemma 2.2 (Inductive Game Hypothesis)**.** *Let $H \subset V$ be a "hard" set of vertices such that $\mathcal{A}$ cannot, with non-negligible probability, produce $\phi(K, h)$ where $h \in H$. Note that the base case has $H = C_V$. Assuming adaptive security of constrained PRFs, one way functions, and the existence of indistinguishable obfuscation, we then have for any $w \in \Gamma(h)$ for all $h \in H$,*

$$\mathbf{Pr}[\mathcal{A}(\mathcal{O}({C_\Gamma}^S), v_0, w, \phi(K, v_0)) = \phi(K, w)] < \epsilon.$$

Proof. We are going to use a series of indistinguishable hybrids along with the circuit defined in 2 to show the above

- $\mathtt{Hyb}_0$: In the first hybrid, the following game is played
  1. The challenger gives the adversary w^* in plaintext.
  2. $K \leftarrow \{0,1\}^\lambda$ and $\phi(K, v_0) = (F(K, v_0), v_0)$ where K is some fixed secret drawn from a uniform distribution

3. The challenger generates $\mathcal{O}({C_\Gamma}^S)$ and gives the program to $\mathcal{A}$
4. $\mathcal{A}$ outputs guess g and wins if $g = \phi(K, w^*)$

- $\texttt{Hyb}_1$: We replace ${C_\Gamma}^S$ with ${C_\Gamma}^S$ as defined in circuit 2. Fix the constant $z^* = f(F(K, w^*))$
- $\texttt{Hyb}_{2,1}$ We replace circuit 2 with circuit 3 where we set $Y^* = (1, y)$ such that $\Gamma(y)_1 = w^*$. 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 $j \in \{ 2, \ldots, \deg^{-1}(w^*) \}$ We replace Y^* with $Y^* \cup (j, y)$ such that $\Gamma(y)_j = w^*$. Note after the last of these hybrids, we have that $F(K, w^*)$ is always set to $\perp$.
- $\texttt{Hyb}_3$: We puncture the PRF at w^* and set $S = \{ w^* \}$.
- $\texttt{Hyb}_4$: Set $z^* = f(t)$ where t is chosen at random

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

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

where z^* in ${C_\Gamma}^{S^*}$ is the image on a OWF of a randomly chosen point. As we will show in lemma 2.3, lemma 2.4, and lemma 2.6, 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 2.3. *$\texttt{Hyb}_0$ 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 ${C_\Gamma}^S$ as defined in circuit 1 and circuit 2 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 2.4. *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 obfuscating a program which calls ${C_\Gamma}^S$ internally and returns $\perp$ for the j-th input.
2. We can go from $\texttt{Hyb}_{2,j-1}$ (or $\texttt{Hyb}_1$) to a padded out version of itself.
3. If an adversary can produce $\texttt{Hyb}_{2,j-1}$ (or $\texttt{Hyb}_1$) and $\texttt{Hyb}_{2,j}$ which are of the same size 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 $\texttt{Hyb}_0/\texttt{Hyb}_1$.

First, assume towards contradiction that there exists an adversary $\mathcal{A}$ that can distinguish two consecutive hybrids in $O(T')$ time with polynomial advantage $\epsilon' > \epsilon/10d$. For simplicity, say that the input size to all of our circuits is n. Also, let C_0 be the circuit from the first hybrid and C_1 the one from the second. Let C_i^{Mid} be a circuit such that $C_i^{\text{Mid}}(X) = C_0(X)$ if $X_i = 0$ and $C_i^{\text{Mid}}(X) = C_1(X)$ if $X_i = 1$. Note that by our construction of the hybrids, C_0 and C_1 differ on at most 1 input (which is the appended vertex y to Y^*); call this input α. Then, $C_i^{\text{Mid}} = C_0$ if $\alpha_i = 0$ and $C_i^{\text{Mid}} = C_1$ if $\alpha_i = 1$. So, if we build an adversary $\mathcal{B}$ to tell if $C_i^{\text{Mid}} = C_0$ or $C_i^{\text{Mid}} = C_1$ with probability γ, we have that $\mathcal{B}(C_0, C_1)$ can tell if α_i is 0 or 1 with probability γ. Thus, $\mathcal{B}$ can reconstruct α with probability at least γ^n. Note that this implies that $\mathcal{B}$ can learn $\phi(K, y)$ where $y \in H$ (the set of hard-to-guess vertices) by construction. But here we run into a problem, the game the distinguishing adversary plays between C_0 and C_1 (and so forth) does not give the circuits from *both* games to $\mathcal{A}$. Rather, $\mathcal{A}$ gets either the circuit C_0 or C_1. Note though that given C_0, $\mathcal{A}$ can construct a larger version of C_1 by obfuscating a program which calls C_0 internally and returns $\perp$ for the j-th input. Call this larger circuit C_1' with circuit size λ'. Moreover, $\mathcal{A}$ can construct a larger version of C_0, which we will call C_0', by simply padding it out to the size of C_1'. Now, if there exists a distinguishing adversary for this larger game with circuit size λ', we can break the inductive hypothesis of the scheme with circuit size λ.

Note that we are showing that if an adversary can distinguish between hybrids with a larger parameter size, then it can break the security of lower parameter hybrids. So, we then need to apply the same argument to the smaller parameter sized versions to update each recursive step. This basically means that we either only support poly-log depth (if we have multiplicative overhead iO) or we assume existance of additive overhead iO Now, if $\mathcal{B}$ can distinguish C_0, C_1, we can create and adversary $\mathcal{D}$ which can output a label of a vertex in H with probability γ^n. $\mathcal{D}(C_0)$ proceeds as follows:

1. Create a circuit $i\mathcal{O}(C_1')$ which calls C_0 internally but returns $\perp$ instead of the j-th output if $v = y$. Also, create $i\mathcal{O}(C_0')$ to be the padded version of C_0 such that $i\mathcal{O}(C_0')$ and $i\mathcal{O}(C_1')$ are the same size.

2. Run $\mathcal{B}(i\mathcal{O}(C_0'), i\mathcal{O}(C_1'))$ n times to get the differing input string $m \in \{0,1\}^n$. Return m.

So now, we just need to build $\mathcal{B}$ to tell if $C_i^{\text{Mid}} = C_0$ or C_1 with probability $\gamma^n \geq \epsilon$. To do so, we 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 a random obfuscation of C_b via re-obfuscating the existing C_b
   (b) Sample a random obfuscation of C_i^{Mid} via re-obfuscating C_i^{Mid}
   (c) Have $\mathcal{A}$ distinguish between C_b and C^{Mid}
   (d) Output 1 if successful.

   Note that we can estimate ϵ_b' as the number of successful runs, which we will denote $\sum_{j \in [I]} S_{i,j}$, 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 parameters such that $\gamma^n \geq \frac{\epsilon}{10d}$ to appendix A. □

Lemma 2.5. *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 2.3, the indistinguishably follows directly from the definition of indistinguishable obfuscation. □

Lemma 2.6. *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 2 Circuit for the neighbor function, ${C_\Gamma}^S$ 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 3 Circuit for the neighbor function, ${C_\Gamma}^S$ 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 $F(K, u_1), F(K, u_2), \ldots, F(K, u_d)$

Abstract

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

[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. 2.4

[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 2.4

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 2.4.

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

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

We then have that

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

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

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

So we finally have that

$$\mathbf{Pr}[\epsilon_1' > \epsilon_0'] \geq 1 - \exp\left(-\frac{I\epsilon'}{12}\right) - \exp\left(-\frac{I\epsilon'}{12}\right) \geq 1 - 2\exp\left(-\frac{I\epsilon'}{12}\right). \tag{3}$$

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.