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 $(\mathrm{Key}_F, \mathrm{Puncture}_F, \mathrm{Eval}_F)$. 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

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

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

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

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

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

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 $v \in V, \deg(v) \le d$. Moreover, for simplicity, assume that for all v,

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

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 $\deg^{-1}(\cdot) \le d$.

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,

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

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 $\Gamma(u) \ne \emptyset$ and otherwise $\Gamma(u)$ returns a $\perp$ string. We fix the adversary's advantage to $\epsilon < \text{poly}(\lambda)$ and runtime to $T(\lambda) \le \text{poly}(\lambda, \frac{1}{\epsilon})$ 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 Using Constant Overhead iO

We let q blah blah blah

2.3 Instantiation

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

Algorithm 1 The circuit for the neighbor function, C_Γ^S padded out to size S.

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

We are going to show that eq. (1) for $S = O(D \cdot \mathcal{O})$ where $\mathcal{O}$ is the additive overhead of indistinguishable 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 $S \ge \text{poly}(\lambda)$.

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 $S \ge O(\mathcal{I} \cdot \text{overhead})$ 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*

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

for any $S \geq \mathrm{poly}(\lambda)$. 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_Γ^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 $H_\mathcal{I} \subset V$ be a "hard" set of vertices for the $\mathcal{I}$-th step of induction 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 additive overhead indistinguishable obfuscation, we then have for any $w \in \Gamma(h)$ and $S \geq q(\mathcal{I}, \lambda)$ 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 $g = \phi(K, w^*)$

- $\mathtt{Hyb}_1$: We replace C_Γ^S with C_Γ^S as defined in circuit 2. Fix the constant $z^* = f(F(K, w^*))$
- $\mathtt{Hyb}_{2,1}$ 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).
- $\mathtt{Hyb}_{2,j}$ 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 $\perp$.
- $\mathtt{Hyb}_3$: We puncture the PRF at w^* and set $S = \{\, w^* \,\}$.
- $\mathtt{Hyb}_4$: Set $z^* = f(t)$ where t is chosen at random

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

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

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 $\mathtt{Hyb}_0$ and $\mathtt{Hyb}_3$ is at most $\epsilon/2$. Thus, if $\mathcal{A}$ can produce $\phi(K, v) = (\sigma_v, v)$ with advantage $\epsilon/2$ in $\mathtt{Hyb}_3$, 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 $\mathtt{Hyb}_0$ cannot be more than ϵ. □

Lemma 2.3. *$\mathtt{Hyb}_0$ and $\mathtt{Hyb}_1$ 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 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 $\mathtt{Hyb}_1$ to $\mathtt{Hyb}_{2,1}$ and $\mathtt{Hyb}_{2,j-1}$ to $\mathtt{Hyb}_{2,j}$ for $j \in 2, \ldots, \deg^{-1}(w^*)$ is distinguishable with advantage at most $\epsilon/(10d)$. Thus, $\mathtt{Hyb}_1$ and $\mathtt{Hyb}_{2,\deg^{-1}(w^*)}$ 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 $\mathtt{Hyb}_{2,j-1}$ (or $\mathtt{Hyb}_1$) to a "padded out" version of $\mathtt{Hyb}_{2,j}$ by obfuscating a program which calls C_Γ^S internally and returns $\perp$ for the j-th input.
2. We can go from $\mathtt{Hyb}_{2,j-1}$ (or $\mathtt{Hyb}_1$) to a padded out version of itself.
3. If an adversary can produce $\mathtt{Hyb}_{2,j-1}$ (or $\mathtt{Hyb}_1$) and $\mathtt{Hyb}_{2,j}$ 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 $\mathtt{Hyb}_0/\mathtt{Hyb}_1$.

For simplicity, say that the input size to all of our circuits is n. Let $s \in \mathbb{Z}$ such that $s \geq q(\mathcal{I}, \lambda)$. 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 "padded out" C_1 given only C_0. Then, if an adversary can distinguish between $C_0^{s'}$ and $C_1^{s'}$ for $s' \geq q(\mathcal{I}+1, \lambda)$, we can break the inductive hypothesis.

First, given C_0^s, $\mathcal{A}$ can construct a larger version of $C_1^{s'}$ by obfuscating a program which calls C_0^s internally and returns $\perp$ for the j-th input of w. $\mathcal{A}$ also then produces C_0^s which is C^0 padded out to the size of $C_1^{s'}$. Note that we define $q(\mathcal{I}+1, \lambda) - q(\mathcal{I}, \lambda)$ in section 2.2 to be at least as large as the difference between C_0^s and $C_1^{s'}$.

Now, assume towards contradiction that there exists an adversary $\mathcal{A}$ that can distinguish $C_0^{s'}$ and $C_1^{s'}$ with advantage $\epsilon' > \epsilon/10d$ in $O(T')$ time with polynomial advantage $\epsilon' > \epsilon/10d$. 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 $X_i = 1$.
We can see that $C_0^{s'}$ and $C_1^{s'}$ differ on at most 1 input which we will call 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 $C_i^{\text{Mid}} = C_0^{s'}$ or $C_i^{\text{Mid}} = C_1^{s'}$ with probability γ, we have that $\mathcal{B}(C_0^{s'}, C_1^{s'})$ 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. We then have that because $\alpha \in H$ and the initial circuit has size $s \geq q(\mathcal{I}, \lambda)$, $\mathcal{B}$ 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 a random obfuscation of $C_b^{s'}$ via re-randomizing the obfuscating of the existing $C_b^{s'}$
   (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. $\square$

Lemma 2.5. *The game in* $\mathtt{Hyb}_{2,\deg^{-1}(w^*)}$ *is indistinguishable from* $\mathtt{Hyb}_3$ *with probability at most* $\epsilon/10$.

Proof. As with lemma 2.3, the indistinguishably follows directly from the definition of indistinguishable obfuscation. $\square$

Lemma 2.6. *The game in* $\mathtt{Hyb}_3$ *is indistinguishable from* $\mathtt{Hyb}_4$.

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 $\mathtt{Hyb}_{2,\deg^{-1}(w^*)}$ except that $z^* = f(a)$. If a is the output of the PRF, then we are in $\mathtt{Hyb}_{2,\deg^{-1}(w^*)}$, if a is the output of a random function, then we are in $\mathtt{Hyb}_3$. □

Algorithm 2 Circuit for the neighbor function, C_Γ^S with PRF key K and constant w^*, z^*

```
function C_Γ^S(X, v)
    if v ≠ w and f(X) ≠ f(F(K, v)) then
        return ⊥
    if v = w and f(X) ≠ z* 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 3 Circuit for the neighbor function, C_Γ^S with punctured PRF key $K(S)$ and constant w^*, Y^*, J^*, z^*

```
function C_Γ^S(X, v)
    if v ≠ w and f(X) ≠ f(F(K, v)) then
        return ⊥
    if v = w and f(X) ≠ z* then
        return ⊥
    if Γ(v) = ∅ then
        return ⊥
    u_1, ... u_d = Γ(v)
    while ∃j* ∈ [d], (j*, u_j) ∈ Y* do
        Set F(K, u_{j*}) = ⊥
    return F(K, u_1), F(K, u_2), ..., 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 $C^{\mathrm{Mid}} = C_0$, 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 $\delta = \frac{\epsilon'}{2\alpha} - 1$

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

So we finally have that

$$\mathbf{Pr}[\epsilon'_1 > \epsilon'_0] \geq 1 - \exp\left(-\frac{I\epsilon'}{12}\right) - \exp\left(-\frac{I\epsilon'}{12}\right) \geq 1 - 2\exp\left(-\frac{I\epsilon'}{12}\right). \tag{3}$$

Setting $I \geq \frac{12(\ln 2 + \ln n - \ln(1-\epsilon) + \ln 2)}{\epsilon'} \in \mathrm{poly}(n, 1/\epsilon, 1/\epsilon')$, 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.