

# Sparse Graph Label Randomization

October 22, 2023

# 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 PRFs  $F$  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

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

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

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

## 1.2 Indistinguishable Obfuscation

We will use the definition of indistinguishable obfuscation as presented in [GGH<sup>+</sup>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

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

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

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

## 2 Using Weak Extractible Obfuscation

### 2.1 Graph Randomized Traversal

Say that we have a sparse, potentially exponentially sized, graph  $\mathcal{G} = (V, E)$  and  $\forall v \in V, \deg(v) = d$ . Moreover, if the graph is a DAG, for simplicity, assume that for all  $v$ ,

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

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,  $\Gamma$ , which can be computed in polynomial time. We define a randomized and keyed labelling function  $\phi : \{0, 1\}^\lambda \times V \rightarrow \{0, 1\}^{\text{poly}(\lambda)}$  such that given,  $\phi(K, v_0)$  for root  $v_0$ , an adversary,  $\mathcal{A}$ , which does not know a path from  $v_0$  to  $v$ ,

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

for function  $C_\Gamma$  where  $C_\Gamma(\phi(K, u)) = \phi(K, \Gamma(u)_1), \dots, \phi(K, \Gamma(u)_d)$  if  $\Gamma(u) \neq \emptyset$  and otherwise  $\Gamma(u)$  returns a  $\perp$  string; and,  $\mathcal{O}$  represents an indistinguishable obfuscator. We fix  $\epsilon \leq \text{negl}(\lambda)$ .

### 2.2 Instantiation

We define

$$\phi(K, v) = F(K, v).$$

We can now define  $C_\Gamma$ :

---

**Algorithm 1** The circuit for the neighbor function,  $C_\Gamma$ .

---

```

1: function  $C_\Gamma(X, v)$ 
2:   if  $f(X) \neq f(F(K, v))$  then
3:     return  $\perp$ 
4:   if  $\Gamma(v) = \emptyset$  then
5:     return  $\perp$ 
6:    $u_1, \dots, u_d = \Gamma(v)$ 
7:   return  $F(K, u_1), F(K, u_2), \dots, F(K, u_d)$ 

```

---

We are going to show that [eq. \(1\)](#) holds 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  $v$ . 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 right-hand side of the cut.

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

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

We define  $C_V \subset V$  to be

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

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  $v$  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  $C_\Gamma$  and create circuit  $\mathcal{O}(C_\Gamma)$  as well as  $\phi(K', v_0)$
2. Run  $\mathcal{A}(\mathcal{O}(C_\Gamma), v_0, v, \phi(K', v_0))$  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  $C_\Gamma(\phi(K', v_\ell), v_\ell)_j \in \{\phi(K', v_0), \dots, \phi(K', v)\}$  and then setting  $v_{\ell+1} = \Gamma(v_\ell)_j$ .

□

We can look at [lemma 2.1](#) 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 BLAH Then, for any  $v \notin H$  and  $w \in \Gamma(h)$  for all  $h \in H$ , we have that*

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

*Proof.* We are going to use a series of indistinguishable hybrids along with the circuit defined in [2](#) to show the above

- **Hyb<sub>0</sub>**: In the first hybrid, the following game is played
  1.  $K \leftarrow \{0, 1\}^{\lambda'}$  and  $\phi(K, v_0) = (F(K, v_0), v)$  where  $K$  is some fixed secret drawn from a random distribution
  2. The challenger generates  $\mathcal{O}(C_\Gamma)$  and gives the program to  $\mathcal{A}$
  3. The challenger gives the adversary  $w^*$  in plaintext.
  4.  $\mathcal{A}$  outputs guess  $g$  and wins if  $g = \phi(K, w^*)$
- **Hyb<sub>1</sub>**: We replace  $C_\Gamma$  with  $C_\Gamma$  as defined in [2](#). Fix the constant  $z^* = f(F(K, w^*))$
- **Hyb<sub>2,1</sub>**: We replace [algorithm 2](#) with [algorithm 3](#) where we set  $Y^* = (1, y)$  such that  $\Gamma(y)_1 = w^*$ . So then, we have that  $F(K, \Gamma(y)_1) = \perp$ . Moreover, we set the punctured set,  $S$  to  $\emptyset$  (i.e. we do not puncture the PRF).
- **Hyb<sub>2,j</sub>** for  $j \in 2, \dots, \deg^{-1}(w^*)$ : 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  $\perp$ .

- $\text{Hyb}_3$ : We puncture the PRF at  $w^*$  and set  $S = \{w^*\}$ .
- $\text{Hyb}_4$ : Set  $z^* = f(t)$  where  $t$  is chosen at random

Finally, we can note that if  $\text{Hyb}_0 \stackrel{c}{\approx} \text{Hyb}_2$ ,

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

where  $z^*$  in  $C_\Gamma^*$  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  $\text{Hyb}_0$  and  $\text{Hyb}_3$  is at most  $\epsilon/2$ . Thus, if  $\mathcal{A}$  can produce  $\phi(K, v) = (\sigma_v, v)$  with advantage  $\epsilon/2$  in  $\text{Hyb}_3$ , 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  $\text{Hyb}_0$  cannot be more than  $\epsilon$ .  $\square$

**Lemma 2.3.**  *$\text{Hyb}_0$  and  $\text{Hyb}_1$  are distinguishable with advantage at most  $\epsilon/10$ .*

*Proof.* Note that for all inputs  $(z, v)$  to  $C_\Gamma$  as defined in [algorithm 1](#) and [algorithm 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  $\epsilon/8$ .  $\square$

**Lemma 2.4.** *Each hybrid from  $\text{Hyb}_1$  to  $\text{Hyb}_{2,1}$  and  $\text{Hyb}_{2,j-1}$  to  $\text{Hyb}_{2,j}$  for  $j \in 2, \dots, \deg^{-1}(w^*)$  is distinguishable with advantage at most  $\epsilon/(10d)$ . Thus,  $\text{Hyb}_1$  and  $\text{Hyb}_{2, \deg^{-1}(w^*)}$  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 is that if a hybrid is distinguishable with advantage more than  $\epsilon/10d$ , then  $\mathcal{A}$  can produce a label  $\phi(K, h)$  for  $h \in H$ .

First, assume towards contradiction that there exists an adversary  $\mathcal{A}$  that can distinguish two consecutive hybrids with polynomial advantage  $\epsilon' > \epsilon/10d$ . Following the proof sketch in [\[IPS15\]](#) say that the input size to  $C_\Gamma$  is  $n$ . Also, let  $C_0$  be the circuit from the first hybrid and  $C_1$  the one from the second. Let  $C_i^{\text{Mid}}$  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  $C_0$  and  $C_1$  differs on at most 1 input (which is the appended vertex  $y$  to  $Y^*$ ); call this input  $\alpha$ . 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  $C_1$  with probability  $\gamma$ , we have that  $\mathcal{B}(C_0, C_1)$  can tell if  $\alpha_i$  is 0 or 1 with probability  $\gamma$ . Thus,  $\mathcal{B}$  can reconstruct  $\alpha$  with probability at least  $\gamma^n$ . Note that this implies that  $\mathcal{B}$  can learn  $\phi(K, y)$  where  $y \in H$  by construction and thus gives our desired contradiction. Moreover, we have that  $\mathcal{A}(C_0)$  (where  $C_0 = C_\Gamma$ ) can construct  $C_1$  (and thus  $C_i^{\text{Mid}}$ ) by obfuscating a program which calls  $C_\Gamma$  internally and returns  $\perp$  for the  $j$ -th input if the input vertex is  $u$  such that  $(j, u) \in Y^*$ .

So now, we just need to build  $\mathcal{B}$  to tell if  $C_i^{\text{Mid}} = C_0$  or  $C_1$  with probability  $\gamma^n \geq \frac{\epsilon}{10d}$ .

Then,  $\mathcal{A}$  can distinguish between  $C^M$  via the following:

1. Run  $I = \left\lceil \frac{12(\ln n - \ln(1 - \frac{\epsilon}{10d}) + \ln 2)}{\epsilon'} \right\rceil$  iterations of the following experiment to estimate advantage  $\epsilon'_b$  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  $C_i^{\text{Mid}}$  via re-obfuscating  $C_i^{\text{Mid}}$
  - (c) Have  $\mathcal{A}$  distinguish between  $C_b$  and  $C^{\text{Mid}}$
  - (d) Output 1 if successful.

Note that we can estimate  $\epsilon'_b$  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,  $C^{\text{Mid}} = C_1$ .

We differ the proof that  $\gamma^n \geq \frac{\epsilon}{10d}$  to [appendix A](#). □

**Lemma 2.5.** *The game in  $\text{Hyb}_{2, \deg^{-1}(w^*)}$  is indistinguishable from  $\text{Hyb}_3$ .*

*Proof.* The indistinguishability follows directly from the definition of indistinguishable obfuscation. □

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

*Proof.* We now show that if the advantage of  $\mathcal{A}$  is greater than  $\epsilon/8$ , then we can create a reduction,  $\mathcal{B}$ , which can break the selective 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  $\text{Hyb}_{2, \deg^{-1}(w^*)}$  except that  $z^* = f(a)$ . If  $a$  is the output of the PRF, then we are in  $\text{Hyb}_{2, \deg^{-1}(w^*)}$ , if  $a$  is the output of a random function, then we are in  $\text{Hyb}_3$ . □

---

**Algorithm 2** Circuit for the neighbor function,  $C_\Gamma$  with PRF key  $K$  and constant  $w^*, z^*$

---

```

1: function  $C_\Gamma(X, v)$ 
2:   if  $v \neq w$  and  $f(X) \neq f(F(K, v))$  then
3:     return  $\perp$ 
4:   if  $v = w$  and  $f(X) \neq z^*$  then
5:     return  $\perp$ 
6:   if  $\Gamma(v) = \emptyset$  then
7:     return  $\perp$ 
8:    $u_1, \dots, u_d = \Gamma(v)$ 
9:   return  $F(K, u_1), F(K, u_2), \dots, F(K, u_d)$ 

```

---



---

**Algorithm 3** Circuit for the neighbor function,  $C_\Gamma$  with punctured PRF key  $K(S)$  and constant  $w^*, Y^*, J^*, z^*$

---

```

1: function  $C_\Gamma(X, v)$ 
2:   if  $v \neq w$  and  $f(X) \neq f(F(K, v))$  then
3:     return  $\perp$ 
4:   if  $v = w$  and  $f(X) \neq z^*$  then
5:     return  $\perp$ 
6:   if  $\Gamma(v) = \emptyset$  then
7:     return  $\perp$ 
8:    $u_1, \dots, u_d = \Gamma(v)$ 
9:   if  $\exists j \in [d], (j^*, v^*) \in Y^*$  then
10:     Set  $F(K, u_{j^*}) = \perp$ 
11:   return  $F(K, u_1), F(K, u_2), \dots, F(K, u_d)$ 

```

---

## 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<sup>+</sup>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  $I = \left\lceil \frac{\ln 2 \cdot 12 (\ln n - \ln(1 - \frac{\epsilon}{10d}))}{\epsilon'} \right\rceil$  where  $I$  is the number of iterations of the experiment define in lemma 2.4.

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

$$\begin{aligned} \gamma = \Pr[\epsilon'_1 > \epsilon'_0] &= \Pr \left[ \sum_{j \in [I]} S_{1,j} > \sum_j S_{0,j} \right] \\ &\geq \Pr \left[ \sum_{j \in [I]} S_{1,j} > \frac{I\epsilon'}{2} \right] \cdot \Pr \left[ \sum_{j \in [I]} S_{0,j} < \frac{I\epsilon'}{2} \right]. \end{aligned}$$

We then have that

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

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

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

So we finally have that

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

Setting  $I \geq \frac{12(\ln n - \ln(1 - \frac{\epsilon}{10d}) + \ln 2)}{\epsilon'} \in \text{poly}(n, 1/\epsilon, 1/\epsilon')$ , we have that

$$\begin{aligned} \gamma^n &\geq \left( 1 - 2 \exp \left( -\frac{I\epsilon'}{12} \right) \right)^n \\ &\geq 1 - 2n \cdot \exp \left( -\frac{I\epsilon'}{12} \right) = 1 - 2n \cdot \exp \left( -\ln n + \ln \left( 1 - \frac{\epsilon}{10d} \right) - \ln 2 \right) \\ &= 1 - \left( 1 - \frac{\epsilon}{10d} \right) = \frac{\epsilon}{10d} \end{aligned}$$

as desired.