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

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

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

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

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

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

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

$$\begin{aligned} & \text{Compose}(\cdots \text{Compose}(\text{Obfuscate}(1^\lambda, C_0^0), C_1^0), \cdots, C_k^0) \\ \stackrel{c}{\approx} \ & \text{Compose}(\cdots \text{Compose}(\text{Obfuscate}(1^\lambda, C_0^1), C_1^1), \cdots, C_k^1). \end{aligned}$$

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.

$$C_\Gamma^S(\phi(K, v), v) = \underbrace{C_P \circ \cdots \circ C_P}_{S \text{ times}} \circ C_\Gamma(\phi(K, v)) \tag{2}$$

where C_Γ is defined in Algorithm 1 and C_P is defined in Algorithm 2. We will use the shorthand $\mathrm{H}i\mathcal{O}(C_\Gamma^S)$ to denote $C_P \circ \cdots \circ C_P \circ\ \mathrm{H}i\mathcal{O}(C_\Gamma)$.

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 $Hi\mathcal{O}$ 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*

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

then

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

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

2.3 Proof of Theorem 1

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

[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

[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 ??

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 **??**.

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

$$\mathbf{Pr}[\epsilon_1' > \epsilon_0'] \geq 1 - \exp\left(-\frac{I\epsilon'}{12}\right) - \exp\left(-\frac{I\epsilon'}{12}\right) \geq 1 - 2\exp\left(-\frac{I\epsilon'}{12}\right). \tag{5}$$

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.