Sparse Graph Label Randomization

October 20, 2023

1 Preliminaries

1.1 Bounded Functional Encryption

We will use the notation of static, bounded functional encryption as presented in [GGLW22].

Security

We will slightly weaken the security notion such that the adversary does not choose which circuits it can learn the functional secret key for. Indeed, this is a weaker notion of functional encryption which fixes the adversary's output circuit. We will assume that we get circuit $C_1, \ldots, C_d$.

For completeness, we have the original security definition of [GGLW22] below:

$$\left\{ \mathcal{A}^{\mathrm{KeyGen}(\mathrm{MSK},\cdot)}(\mathrm{CT}) \quad \begin{array}{l} (1^n, 1^q) \leftarrow \mathcal{A}^{(1)} \\ (\mathrm{MPK}, \mathrm{MSK}) \leftarrow \mathrm{Setup}\,(1^n, 1^q) \\ m \leftarrow \mathcal{A}^{\mathrm{KeyGen}(\mathrm{MSK})}(\mathrm{MPK}) \\ \mathrm{CT} \leftarrow \mathrm{Enc}(\mathrm{MPK}, m) \end{array} \right\}_{\lambda \in \mathbb{N}}$$
$$\overset{c}{\approx}$$
$$\left\{ \mathcal{A}^{\mathrm{Sim}_3^{U_m(\cdot)}}(\mathrm{CT}) \quad \begin{array}{l} (1^n, 1^q) \leftarrow \mathcal{A}(1^\lambda) \\ (\mathrm{MPK}, \mathbf{st}_0) \leftarrow \mathrm{Sim}_0\left(1^\lambda, 1^n, q\right) \\ m \leftarrow \mathcal{A}^{S_1(\mathbf{st}_0)}(\mathrm{MPK}) \\ (\mathrm{CT}, \mathbf{st}_2) \leftarrow \mathrm{Sim}_2(\mathbf{st}_1, \Pi^m) \end{array} \right\}_{\lambda \in \mathbb{N}}$$

whenever the following admissibility constraints and properties are satisfied:

- $\mathrm{Sim}_1, \mathrm{Sim}_3$ are stateful in that after each invocation, they updated their states $\mathbf{st}_1, \mathbf{st}_3$ respectively which is carried over to the next invocation.
- Π^m contains a list of functions f_i queried by $\mathcal{A}$ in the pre-challenge phase along with their output on the challenge message m. That is, if f_i is the i-th function queried by $\mathcal{A}$ to oracle Sim_1 and $q_{[re}$ be the number of queries $\mathcal{A}$ makes before outputting m, then $\Pi^m = \big((f_1, f_1(m)), \ldots, (f_{q_{pre}}, f_{q_{pre}}(m))\big)$.
- $\mathcal{A}$ makes at most q queries combined tote key generation oracle in both games.
- Sim_3 for eac queried function f_i, in the post challenge phase, makes a single query to its message oracle U_m on the same f_i itself.

Our modified security definition is as follows:

$$\left\{ \mathcal{A}^{\mathrm{KeyGen}(\mathrm{MSK},\{C_1,\ldots,C_d\})}(\mathrm{CT}) \quad \begin{array}{l} (1^n, 1^q) \leftarrow \mathcal{A}^{(1)} \\ (\mathrm{MPK}, \mathrm{MSK}) \leftarrow \mathrm{Setup}\,(1^n, 1^q) \\ m \leftarrow \mathcal{A}(\mathrm{MPK}, \mathrm{SK}_{C_1}, \ldots, \mathrm{SK}_{C_d}) \\ \mathrm{CT} \leftarrow \mathrm{Enc}(\mathrm{MPK}, m) \end{array} \right\}_{\lambda \in \mathbb{N}}$$
$$\overset{c}{\approx} \tag{1}$$
$$\left\{ \mathcal{A}^{\mathrm{Sim}_3^{U_m(\{C_1,\ldots,C_d\})}}(\mathrm{CT}) \quad \begin{array}{l} (1^n, 1^q) \leftarrow \mathcal{A}(1^\lambda) \\ (\mathrm{MPK}, \mathbf{st}_0) \leftarrow \mathrm{Sim}_0\left(1^\lambda, 1^n, q\right) \\ m \leftarrow \mathcal{A}^{S_1(\mathbf{st}_0)}(\mathrm{MPK}, C_1, \ldots, C_d) \\ (\mathrm{CT}, \mathbf{st}_2) \leftarrow \mathrm{Sim}_2(\mathbf{st}_1, \Pi^m) \end{array} \right\}_{\lambda \in \mathbb{N}}$$

where the admissibility constraints remain the same.

1.2 Non-malleable Bounded FE

Here, we introduce the notion of non-malleable bounded functional encryption.

We define non-malleable security of bounded functional encryption in almost the exact notion of [Pas06] for public key encryption. First, let $NM(m_1, \ldots, m_q, \mathcal{A})$ be a game as follows for $q = \text{poly}(\lambda)$:

1. $(\text{MPK}, \text{MSK}) \leftarrow \text{FE.Setup}(1^\lambda)$
2. $\text{CT}_1, \ldots, \text{CT}_q \leftarrow \text{FE.Enc}(\text{MPK}, m_1), \ldots \text{FE.Enc}(\text{MPK}, m_q)$
3. $\text{CT}'_1, \ldots, \text{CT}'_\ell \leftarrow \mathcal{A}(\text{MPK}, \text{CT}_1, \ldots, \text{CT}_q, 1^{|m|})$
4. $m'_i \leftarrow \perp$ is $\text{CT}_i = \text{CT}'_j$ for any $i \in [q]$, $j \in [\ell]$ and $\text{FE.Dec}(\text{SK}_{\text{identity}}, c_i)$ otherwise.

Then, we say that a bounded functional encryption scheme is non-malleable if for all PPT $\mathcal{A}$ and every PPT $\mathcal{D}$, there exists a negligible function $\texttt{negl}$ such that for all $\{\, m \,\}_0, \{\, m \,\}_1 \in \{0,1\}^{nq}$, we have

$$\left| \mathbf{Pr}[\mathcal{D}(NM(\{m\}_0, \mathcal{A})) = 1] - \mathbf{Pr}[\mathcal{D}(NM(\{m\}_1, \mathcal{A})) = 1] \right| \leq \texttt{negl}. \tag{2}$$

As outlined in [Pas06], we can equivalently define non-malleability in terms of a PPT recognizable relation R such that

$$\Bigg| \mathbf{Pr}\left[NM\left(m_1, \ldots m_q, \mathcal{A}(z)\right) \in \bigcup_{m \in \{m\}} R(m) \right] - \tag{3}$$
$$\mathbf{Pr}\left[c \leftarrow \text{Sim}_{NM}(1^n, z); m' = \text{FE.Dec}(\text{SK}_{\text{identity}}, c); m' \in \bigcup_{m \in \{m\}} R(m) \right] \Bigg| \leq \texttt{negl}(\lambda).$$

Note that in the above definition, we do not give the adversary access to any SK_{C_i}. We simply require that the scheme is public key (many message) non-malleable.

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,

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

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, an adversary, $\mathcal{A}$, which does not know a path from v_0 to v,

$$\mathbf{Pr}[\mathcal{A}(\mathcal{O}(C_\Gamma), v_0, v, \phi(K, v_0)) = \phi(K, v)] \leq \epsilon \tag{4}$$

for function C_Γ 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 $\epsilon \leq \texttt{negl}(\lambda)$.

2.2 Instantiation

We define

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

We can now define C_Γ:

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)
```

We are going to show that eq. (4) 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 $\mathbf{Pr}[E(\Gamma, v_0, v) = P] \geq \frac{1}{p(\lambda)}$ 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*

$$\mathbf{Pr}[\mathcal{A}(\mathcal{O}(C_\Gamma), v_0, v, \phi(K, v_0)) \in \phi(K, C_V)] \leq \texttt{negl}(\lambda). \tag{5}$$

We define $C_V \subset V$ to be

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

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. (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 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 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 $\phi(K', v_0), \ldots \phi(K', v)$ 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 $C_\Gamma(\phi(K', v_\ell), v_\ell)_j \in \{\, \phi(K', v_0), \ldots, \phi(K', v) \,\}$ and then setting $v_{\ell+1} = \Gamma(v_\ell)_j$.

$\square$

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 $H = C_V$. Then, for any $v \notin H$ and $w \in \Gamma(h)$ for all $h \in H$, we have that*

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

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. $K \leftarrow \{0,1\}^{\lambda'}$ 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 $g = \phi(K, w^*)$
- $\mathtt{Hyb}_1$: We replace C_Γ with C_Γ as defined in 2. Fix the constant $z^* = f(F(K, w^*))$
- $\mathtt{Hyb}_{2,1}$ We replace algorithm 2 with algorithm 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 $\perp$.

- $\texttt{Hyb}_3$: 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 $\texttt{Hyb}_0 \overset{c}{\approx} \texttt{Hyb}_2$,

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

where z^* in C_Γ^* 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. Note that for all inputs (z, v) to C_Γ 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 $\epsilon/8$. □

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 follow very closely the simple case of weak extractible obfuscation as defined in (TODO: cite). 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 (TODO: cite), say that the input size to C_Γ 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 C_0 and C_1 differs 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_1 with probability γ, we have that $\mathcal{B}$ 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 $y \in H$ by construction and thus gives our desired contradiction. 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 $I = \left\lceil \frac{\ln 2 \cdot 96 \left(\ln n - \ln\left(1 - \frac{\epsilon}{10d}\right)\right)}{\epsilon'} \right\rceil$ 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, $C^{\text{Mid}} = C_1$.

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

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

We then have that

$$\mathbf{Pr}\left[\sum_j S_{1,j} > I\epsilon' \cdot \tfrac{1}{2}\right] \geq 1 - \exp\left(-\frac{I\epsilon'}{2^2 \cdot 3}\right) = 1 - \exp\left(-\frac{I\epsilon'}{96}\right). \qquad \text{(by the Chernoff bound)}$$

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

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

So we finally have that

$$\mathbf{Pr}[\epsilon_1' > \epsilon_0'] \geq 1 - \exp\left(-\frac{I\epsilon'}{12}\right) - \exp\left(-\frac{I\epsilon'}{96}\right) \geq 1 - 2\exp\left(-\frac{I\epsilon'}{96}\right). \tag{6}$$

Setting $I \geq \frac{\ln 2 \cdot 96\left(\ln n - \ln\left(1 - \frac{\epsilon}{10d}\right)\right)}{\epsilon'} \in \text{poly}(n, 1/\epsilon, 1/\epsilon')$, we have that

$$\begin{aligned}\gamma^n &\geq \left(1 - 2\exp\left(-\frac{I\epsilon'}{96}\right)\right)^n \\ &\geq 1 - 2n \cdot \exp\left(-\frac{I\epsilon'}{96}\right) = 1 - 2n \cdot \exp\left(-\left(\ln n + \ln\left(1 - \frac{\epsilon}{10d}\right)\right) \cdot 2\right) \\ &= 1 - \left(1 - \frac{\epsilon}{10d}\right) = \frac{\epsilon}{10d}\end{aligned}$$

as desired. ☐

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

Proof. 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. 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 $\texttt{Hyb}_{2,\deg^{-1}(w^*)}$ 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_Γ with PRF key K and constant w^*, z^*

function $C_\Gamma(X, v)$
  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_Γ with punctured PRF key $K(S)$ and constant w^*, Y^*, J^*, z^*

function $C_\Gamma(X, v)$
  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 $\perp$
  $u_1, \ldots u_d = \Gamma(v)$
  if $\exists j \in [d], (j^*, v^*) \in Y^*$ **then**
    Set $F(K, u_{j^*}) = \perp$
  return $F(K, u_1), F(K, u_2), \ldots, F(K, u_d)$

Lemma 2.7. *The game in $\mathtt{Hyb}_1(1a)$ is indistinguishable from $\mathtt{Hyb}_0$.*

Proof. As the functionality of C_Γ in $\mathtt{Hyb}_0$ equals that of $\mathtt{Hyb}_1(1a)$, we have indistinguishable simply from the definition of indistinguishable obfuscation. □

Lemma 2.8. *The game in $\mathtt{Hyb}_1(1b)$ is indistinguishable from $\mathtt{Hyb}_1(1a)$.*

Proof. Here we argue that if the game in $\mathtt{Hyb}_1(1b)$ is distinguishable from $\mathtt{Hyb}_1(1a)$, then we can construct an adversary, $\mathcal{B}$, which can break the security of the PRF at the punctured point. □

Lemma 2.9. *The game in $\mathtt{Hyb}_1(2a)$ is indistinguishable from $\mathtt{Hyb}_0$ and, by the inductive hypothesis, all previous hybrids.*

Proof. Again, we have that the circuit for C_Γ is the same in $\mathtt{Hyb}_0$ and $\mathtt{Hyb}_1(2a)$. Thus, by the definition of indistinguishable obfuscation, these games are indistinguishable. □

Lemma 2.10. *The game in $\mathtt{Hyb}_1(2b)$ is indistinguishable from $\mathtt{Hyb}_1(2a)$ and, by the inductive hypothesis, all previous hybrids.*

Proof. TODO: PRF security + extractor part □

October 20, 2023

Abstract

References

[GGLW22] Rachit Garg, Rishab Goyal, George Lu, and Brent Waters. Dynamic collusion bounded functional encryption from identity-based encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 736–763. Springer, 2022. 1.1, 1.1

[Pas06] Rafael Pass. Lecture 16: Non-malleability and public key encryption, October 2006. 1.2, 1.2