# Sparse Graph Label Randomization

October 19, 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.
- $\Pi^m$ 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 $\mathrm{Sim}_1$ and $q_{\lfloor re}$ 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.
- $\mathrm{Sim}_3$ 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

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

Note that in the above definition, we do not give the adversary access to any $\text{SK}_{C_i}$. 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 $\forall v \in V, \deg(v) = d$. 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 $\phi : \{0,1\}^\lambda \times V \to \{0,1\}^{\text{poly}(\lambda)}$ 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 $v$,

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

for function $C_\Gamma$ where $C_\Gamma(\phi(K, u)) = \phi(K, \Gamma(u)_1), \ldots, \phi(K, \Gamma(u)_d)$ if $\Gamma(u) \neq \emptyset$ and otherwise $\Gamma(u)$ returns a $\perp$ string; and, $\mathcal{O}$ represents an indistinguishable obfuscator.

### 2.2 Instantiation

We define

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

For shorthand, we will write $\sigma_v$ to connote an attempted "signature" of $v$ where a correct signature is $F(K, v)$.

We can now define $C_\Gamma$:

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

```
function C_Γ(f(σ_v), v)
    if f(σ_v) ≠ f(F(K, v)) then
        return ⊥
    if Γ(v) = ∅ then
        return ⊥
    u_1, ... u_d = Γ(v)
    return f(F(K, u_1)), f(F(K, u_2)), ..., f(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.** *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 $C_\Gamma$ 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 $v_{\ell+1} = \Gamma(v_\ell)_j$.

□

**Algorithm 2** Circuit for the neighbor function, $C_\Gamma^{w^*, \Gamma(w^*)_1, \dots, \Gamma(w^*)_d}$ with punctured PRF key $K(\{ w^* \})$ and constant $z^*, z_1^*, z_2^*, \dots, z_d^*$

**function** $C_\Gamma(f(\sigma_v), v)$
  **if** $v \neq w$ and $f(\sigma_v) \neq f(F(K, v))$ **then**
    **return** $\perp$
  **if** $v = w$ and $f(\sigma_v) \neq z^*$ **then**
    **return** $\perp$
  **if** $\Gamma(v) = \emptyset$ **then**
    **return** $\perp$
  **if** $v = w$ **then**
    **return** $z_1^*, z_2^*, \dots, z_d^*$
  $u_1, \dots u_d = \Gamma(v)$
  **return** $f(F(K, u_1)), f(F(K, u_2)), \dots, f(F(K, u_d))$

We are going to use a series of inductively built indistinguishable hybrids along with algorithm 2 to show that eq. (4) holds.

- $\texttt{Hyb}_0$: In the first hybrid, the following game is played
  1. $K \xleftarrow{\$} \{0,1\}^{\lambda'}$ and $\phi(K, v_0) = (F(K, v_0), v)$
  2. The challenger generates $\mathcal{O}(C_\Gamma)$ and gives the program to $\mathcal{A}$
  3. The challenger chooses a $v$ and gives the adversary $v$ in plaintext.
  4. $\mathcal{A}$ outputs guess $g$ and wins if $g = \phi(K, v)$

- $\texttt{Hyb}_1$: Let $\mathcal{P}$ be the set of all paths from $v_0$ to $v$. For each path $P \in \mathcal{P}$ where $P$ is an ordered list of connected vertices, we have that the adversary does not know some part of $P$. We can note that this implies that $\mathcal{A}$ does not know $\phi(K, p)$ for all $p \in P$ as then $\mathcal{A}$ can recover $P$. Let $u_P$ be the first vertex in $P$ such that $\mathcal{A}$ does not know *a path* from $v_0$ to $u_P$. Define $\text{Suff}'(P)$ to be the path in $P$ from this $u_P$ to $v$. Then, necessarily, $\mathcal{A}$ does not know $\phi(K, w_P)$ for at least one $w_P \in \text{Suff}'(P)$ as then $\mathcal{A}$ would know a path from $v_0$ to $v$. Now, let $\text{Suff}(P)$ be the path which starts at $w_P$, ends at $v$.

  We now inductively build up a series of hybrids to show that a hybrid distribution which shows $\phi(K, s)$ for $s \in \text{Suff}(P)$ indistinguishable from random. We perform the following procedure for each $P \in \mathcal{P}$. So, for $P \in \mathcal{P}$,

  1. For the base case, let $U = \text{Suff}(P)_1$ where $\text{Suff}(P)_1$ is the first vertex in $P$ such that $\mathcal{A}$ does not know $\phi(K, p)$ for $p \in P$.
     (a) Set $w^* = p$. Then, replace $C_\Gamma$ with $C_\Gamma^{w^*, \Gamma(w^*)_1, \ldots, \Gamma(w^*)_d}$ as defined in algorithm 2. Fix the constant $z^* = f(F(K, p))$ and $z_1^* = f(F(K, \Gamma(w^*)_1)), \ldots, z_d^* = f(F(K, \Gamma(w^*)_d))$.
     (b) Set $z^* = f(t), z_1^* = f(t_1), \ldots z_d^* = f(t_d)$ where $t, t_1, \ldots, t_d$ are chosen at random
  2. For the $\ell$-th inductive step where $1 \leq \ell < |\text{Suff}(P)|$, we are going to assume that we are given a hybrid such that $w^* = \text{Suff}(P)_\ell$ and $z^* = f(t), z_1^* = f(t_1), \ldots z_d^* = f(t_d)$ for random $t, \ldots t_d$ in algorithm 2. Now, we change the hybrid in a similar manner as in the base case:
     (a) Set $w^* = \text{Suff}(P)_{\ell+1}$. Then, replace $C_\Gamma$ with $C_\Gamma^{w^*, \Gamma(w^*)_1, \ldots, \Gamma(w^*)_d}$ as defined in algorithm 2. Fix the constant $z^* = f(F(K, p))$ and $z_1^* = f(F(K, \Gamma(w^*)_1)), \ldots, z_d^* = f(F(K, \Gamma(w^*)_d))$.
     (b) Set $z^* = f(t), z_1^* = f(t_1), \ldots z_d^* = f(t_d)$ where $t, t_1, \ldots, t_d$ are chosen at random

     to puncture on $\text{Suff}(P)_{\ell+1}$ where we update $z^*, \ldots z_d^*$ with new randomness.

Finally, we can note that if $\texttt{Hyb}_0 \overset{c}{\approx} \texttt{Hyb}_1$,

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

where $z^*$ in $C_\Gamma^*$ is the image on a OWF of a randomly chosen point. Thus, if $\mathcal{A}$ can produce $\phi(K, v) = (\sigma_v, v)$, then $\mathcal{A}$ can find a preimage for $z*$ under $f$ and thus break the security of a one way function.

Now, we show that $\texttt{Hyb}_0 \overset{c}{\approx} \texttt{Hyb}_1$.

**Lemma 2.2.** *The game in $\texttt{Hyb}_1(1a)$ is indistinguishable from $\texttt{Hyb}_0$.*

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

**Lemma 2.3.** *The game in $\texttt{Hyb}_1(1b)$ is indistinguishable from $\texttt{Hyb}_1(1a)$.*

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

□

**Lemma 2.4.** *The game in $\texttt{Hyb}_1(2a)$ is indistinguishable from $\texttt{Hyb}_0$ and, by the inductive hypothesis, all previous hybrids.*

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

**Lemma 2.5.** *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 19, 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