

Sparse Graph Obfuscation

October 5, 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, \dots, C_d .

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

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

Our modified security definition is as follows:

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

we also copy the admissibility constraints of [GGLW22]:

1.2 Non-malleable Bounded FE

Here, we introduce the notion of non-malleable bounded functional encryption. While we make the definition explicit (in terms of its non-malleability), we prove that simulation-secure bounded FE is equivalent to simulation secure non-malleable bounded FE.

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

1. $(\text{MPK}, \text{MSK}) \leftarrow \text{FE.Setup}(1^\lambda)$
2. $\text{SK}_{C_i} \leftarrow \text{FE.Keygen}(\text{MSK}, C_i)$ for $i \in [d]$
3. $\text{CT}_1, \dots, \text{CT}_q \leftarrow \text{FE.Enc}(\text{MPK}, m_1), \dots, \text{FE.Enc}(\text{MPK}, m_q)$

4. $c'_1, \dots, c'_\ell \leftarrow \mathcal{A}(\text{CT}, 1^{|m|})$
5. $m'_i \leftarrow \perp$ is $c_i = c_j$ for $j \in [q]$ 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 negl such that for all $\{m\}_0, \{m\}_1 \in \{0, 1\}^{nq}$, we have

$$|\Pr[\mathcal{D}(\text{NM}(\{m\}_0, \mathcal{A})) = 1] - \Pr[\mathcal{D}(\text{NM}(\{m\}_1, \mathcal{A})) = 1]| \leq \text{negl}. \quad (2)$$

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

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

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 non-malleable.

and note that if SK_{C_i} is given to the adversary such that $\text{FE.Dec}(\text{SK}_{ID}, C_i(x)) \notin R(m)$ for all $x \in \{0, 1\}^n$, then the scheme is still non-malleable even if the adversary is given access to SK_{C_i} . This is because we can use the FE simulator to replace FE.Enc with a simulator Sim_{FE} which relies solely on public parameters and $C_i, C_i(x)$. Indeed, as the

Thus, we can see that given the simulator SK_{C_i} is useless as $\text{FE.Dec}(C_i(x)) \notin R(m)$.

TODO: CHANGE NM definition to be from <https://www.cs.cornell.edu/rafael/papers/PSV06a.pdf>... the adversary gets the public key

2 A sketch for the boys

2.1 DAG 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, Γ , which can be computed in polynomial time. We define a (pseudo) randomized and keyed labelling function $\phi : V \times \{0, 1\}^\lambda \rightarrow \{0, 1\}^{\text{poly}(\lambda)}$ 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 ,

$$\Pr[\mathcal{A}(C_\Gamma, v_0, v, \phi(K, v_0)) \in \text{Image}(\phi(K, v))] \leq O(v)\epsilon \quad (3)$$

for some fixed $\epsilon \leq \text{negl}(\lambda)$ and 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 0 string of length $d|\phi(K, \cdot)|$.

2.2 Instantiation

We define $\phi(K, v)$ to be as follows:

1. Let $r_1, r_2 \xleftarrow{\$} \{0, 1\}^\lambda$ or r_1, r_2 is drawn from a pseudorandom distribution.
2. Return $\text{FE.Enc}(\text{MPK}, (K, v, r_2))$ where encryption is done with randomness from r_1 .

We can now define, C_Γ .

Algorithm 1 The circuit for the neighbor function, C_Γ .

```

1: function  $\text{INNER}_i(K, v, r)$ 
2:   if  $\Gamma(v) = \emptyset$  then
3:     return  $0 \in \{0, 1\}^*$ 
4:    $u_1, \dots, u_d = \Gamma(v)$ 
5:    $u = u_i$ 
6:    $r_1, r_2 = \text{PRG}(r)$ 
7:   return  $\text{FE.Enc}(\text{MPK}, (u, K, r_2))$  where we encrypt with randomness from  $r_1$ .
8: function  $C_\Gamma(\phi(K, v))$ 
9:   for  $i \in [d]$  do
10:     $u_i = \text{Dec}(\text{SK}_{\text{inner}_i}, \phi(K, v))$ 
11:  return  $(u_1, \dots, u_d)$ 

```

Now, we will prove [eq. \(3\)](#). For convenience, we will restate [eq. \(3\)](#):

$$\Pr[\mathcal{A}(C_\Gamma, v_0, v, \phi(K, v_0)) \in \text{Image}(\phi(K, v))] \leq O(v)\epsilon$$

for all PPT \mathcal{A} .

Proof of [eq. \(3\)](#). Let P be the set of all paths from v_0 to v . Then, as the adversary does not know any path from v_0 to v , we have that for each $p \in P$ where $p = v_0, \dots, v$, there is some suffix of p , p' , which the adversary does not know.

We are going to use a similar proof technique as in ?? where we use a hybrid argument.

- Hyb_0 : In the first hybrid, the following game is played

1. $K \xleftarrow{\$} \{0, 1\}^{\lambda'}$ and $\text{MPK}, \text{SK} \leftarrow \text{FE.Setup}(1^{\lambda'})$.

2. The challenger generates $\text{SK}_{\text{inner}_i} \leftarrow \text{FE.Keygen}(\text{MSK}, \text{inner}_i)$ for $i \in [d]$ and gives these keys to \mathcal{A}
 3. The challenger chooses a v and gives the adversary v in plaintext.
 4. The challenger picks random $r_1, r_2 \xleftarrow{\$} \{0, 1\}^{\lambda'}$ and generates $\phi(K, v_0) = \text{FE.Enc}(\text{MPK}, (K, v_0, r_2))$ using r_1 as the random coins and gives $\phi(K, v_0)$ to \mathcal{A} .
 5. \mathcal{A} outputs guess g and wins if $g \in \phi(K, v)$
- **Hyb₁**: We replace $\phi(K, v_0)$ with a simulated cipher-text using the simulator Sim_2 MPK with its simulated counterpart using Sim_0 , and $\text{SK}_{\text{inner}_i}$ with its simulated counterpart using Sim_3 as defined in [eq. \(1\)](#).
 - **Hyb_{2a}**: For any input into Sim_2 via $\Pi^{K, w, r}$ where $w \in \mathbb{W}$ and r is random, we replace the output of inner_i with inner'_i which uses true randomness r_1^*, r_2^* in stead of r_1, r_2 . For any call to $C_\Gamma(\phi(K, w))$ by \mathcal{A} for $w \in V$, we replace the output of inner_i with inner'_i which uses true randomness r_1^*, r_2^* in stead of r_1, r_2 . This is equivalent to changing Π^m to $\Pi^{m'}$ in [eq. \(1\)](#) where $\Pi^{m'}$ is the list $(\text{inner}_1, \text{inner}'_1(\cdot), \dots, \text{inner}_d, \text{inner}'_d(\cdot))$. Note that this gives us that $\text{inner}'_i(K, w, r) = \phi(K, u) = \text{FE.Enc}(\text{MPK}, (K, u, r_2^*))$ where $u = \Gamma(w)_i$.
 - **Hyb_{2b}**: For any call by \mathcal{A} to $\text{inner}'_i(K, w, r) = \text{CT}$, we replace CT with CT' where CT' is the output of Sim_2 with input $\Pi^{(K, u, r)'}$ where $u = \Gamma(w)_i$.
- Note that the replacement of **Hyb_{2a}** and **Hyb_{2b}** are repeated multiple times. Specifically, these replacements are repeated at most α times where α is the number of unique times \mathcal{A} runs $\text{FE.Dec}(\text{SK}_{\text{inner}_i}, \phi(K, w))$.
- **Hyb₃**: 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} never queries $\text{inner}_i(w^u)$ where $u = \Gamma(w^u)_i$ for some $u \in P$ and the adversary knows a path from v_0 to w . We can see this because if there is no $u \in P$ such that \mathcal{A} never queries $\text{inner}_i(w^u)$, then the adversary knows a path from v_0 to v . Define $\text{Suff}(P)$ to be the path which starts at u , ends at v , and is a suffix of P . We now inductively build up a series of hybrids to show that a hybrid distribution which “erases” $\phi(K, v)$ from inner_i is indistinguishable from the above hybrid.

- For the base case, let $U = \{u_1, \dots, u_{\|\mathcal{P}\|}\}$ where u is the first vertex in P such that \mathcal{A} never queries $\text{inner}_i(w^u)$ as defined above. Then, replace $\text{inner}'_i(\cdot)$ with $\text{inner}^*_i(\cdot)$ in Π_m such that $\text{inner}^*_i(w) = \text{inner}'_i(w)$ if $w \neq w^u$ for $u \in U$ and otherwise $\text{inner}^*_i(w^u) = \perp$. We can note that this hybrid is indistinguishable as inner'_i only changes for input ciphertexts which the adversary never queries.
- For the ℓ -th inductive step, we are going to assume that we are given a hybrid such that inner^ℓ_i such that $\text{inner}^\ell_i(w^u) = \perp$ for $u \in U^\ell$ where U^ℓ where $U^\ell = \bigcup_{P \in \mathcal{P}} \text{Suff}(P)_1, \dots, \text{Suff}(P)_\ell$ and otherwise $\text{inner}^\ell_i(\cdot) = \text{inner}'_i(\cdot)$. We now show that if \mathcal{A} can distinguish between a hybrid with $\text{inner}^\ell_i(\cdot)$ and $\text{inner}^{\ell+1}_i(\cdot)$, then the adversary can break the non-malleability of the FE scheme. We defer this proof to [lemma 2.2](#).

Finally, we can note that by the indistinguishability of **Hyb₀** and **Hyb₅**,

$$\Pr[\mathcal{A}(C_\Gamma, v_0, v, \phi(K, v_0)) \in \text{Image}(\phi(K, v))] \approx \Pr[\mathcal{A}(C'_\Gamma, v_0, v, \phi(K, v_0)) \in \text{Image}(\phi(K, v))]$$

where C'_Γ is C_Γ except that C'_Γ uses inner_i^p where $p = \max_{P \in \mathcal{P}} |P|$. We can note that C'_Γ returns \perp for any query on $\phi(K, w^v)$ where $w^v \in \Gamma^{-1}(v)$. Using [lemma 2.3](#), we have that

$$\Pr[\mathcal{A}(C'_\Gamma, v_0, v, \phi(K, v_0)) \in \text{Image}(\phi(K, v))] \leq \text{negl}(\lambda).$$

□

Lemma 2.1. $\text{Hyb}_0 \stackrel{c}{\approx} \text{Hyb}_2$.

Proof. First we show that $\text{Hyb}_0 \stackrel{c}{\approx} \text{Hyb}_1$. Note that if \mathcal{A} can distinguish between Hyb_0 and Hyb_1 then an adversary can distinguish between an FE scheme and its simulated counterpart where m is fixed to (K, v_0, r) .

Then, if \mathcal{A} can distinguish Hyb_1 and Hyb_2 ,

□

Lemma 2.2. Let \mathcal{A} be a PPT adversary and assume that we have a non-malleable and simulation secure FE scheme. Then, we have that the inductive step of Hyb_3 holds.

Proof. We construct an adversary \mathcal{B} that can break NM security using \mathcal{A} if \mathcal{A} can distinguish between the hybrids in the inductive step. Note that in order to distinguish between the hybrids, \mathcal{A} must have queried inner_i^ℓ or inner_{i+1}^ℓ on $\phi(K, w^u)$ where $u \in \{\text{Suff}(P)_{\ell+1} \mid P \in \mathcal{P}\}$ as this is the only difference between the hybrids. Thus, we see that \mathcal{A} is able to produce to produce $\text{CT} \in \phi(K, w^u)$. By definition of inner_i^ℓ though, we know that $\text{inner}_i^\ell(\phi(K, q)) \neq \phi(K, w^u)$ for any $q \in V$ as we define $\text{inner}_i^\ell(K, q) = \perp$ if $\text{inner}_i'(K, q) = \phi(K, w^u)$. Thus, the adversary has to be able to produce $\text{CT} \in \phi(K, w^u)$ without calling C_Γ .

Thus, if $\mathcal{A}(w^u, v_0, C_\Gamma, \phi(K, v_0))$ can produce $\text{CT} \in \phi(K, w^u)$, we can have $\mathcal{B}(\phi(K, v_0), \phi(K, q_1), \dots, \phi(K, q_{\text{poly}(\lambda)}))$ produce $\phi(K, w^u)$ where $q_1, \dots, q_{\text{poly}(\lambda)}$ are all the vertices that \mathcal{A} has queried C_Γ on. \mathcal{B} simply has to invoke Sim_1 to create a simulated set of function keys for inner_i for all $i \in [d]$ and can then simulate C_Γ .

We can then have \mathcal{B} invoke Sim_1 to create a simulated function key for $\text{SK}'_{\text{inner}_i}$ and thus a simulated C'_Γ . \mathcal{B} then gives $\mathcal{A}(w^u, v_0, C'_\Gamma, \phi(K, v_0))$. we can then break ?? as \mathcal{A} is able to create an encryption of $\phi(K, w^u)$ with non-negligible probability while the simulator cannot. □

Lemma 2.3. Define C'_Γ where C'_Γ is defined as in [algorithm 1](#) except that for some set $U \subset V$, $C'_\Gamma(w^u)_i = \perp$ for all $w^u \in V$ such that $u = \Gamma(w^u)_i$ for some $u \in U$. In words, the parent of all $u \in U$ do not return $\phi(K, u)$ when queried on C'_Γ . Then, assuming the non-malleability and simulation security of FE, we have that for all PPT \mathcal{A} and all $u \in U$,

$$\Pr[\mathcal{A}(C'_\Gamma, v_0, u, U, \phi(K, v_0)) \in \text{Image}(\phi(K, u))] \leq \text{negl}(\lambda). \quad (4)$$

Proof. We construct an adversary \mathcal{B} that can break NM security using \mathcal{A} if \mathcal{A} can produce $\text{CT} \in \phi(K, u)$ for some $u \in U$.

If $\mathcal{A}(w^u, v_0, C_\Gamma, u, U, \phi(K, v_0))$ can produce $\text{CT} \in \phi(K, u)$, we can have $\mathcal{B}(\phi(K, v_0), \phi(K, q_1), \dots, \phi(K, q_{\text{poly}(\lambda)}))$ produce $\phi(K, u)$ where $q_1, \dots, q_{\text{poly}(\lambda)}$ are all the vertices that \mathcal{A} has queried C'_Γ on. \mathcal{B} simply has to invoke Sim_1 to create a simulated set of function keys for inner_i for all $i \in [d]$ and can then simulate C'_Γ with these function keys.

We can then have \mathcal{B} invoke Sim_1 to create a simulated function key for $\text{SK}'_{\text{inner}_i}$ and thus a simulated C'_Γ . \mathcal{B} then gives $\mathcal{A}(w^u, v_0, C'_\Gamma, \phi(K, v_0))$. We can then break ?? (this is supposed to be the NM relationship equation) as \mathcal{A} is able to create an encryption of $\phi(K, w^u)$ with non-negligible probability while the simulator cannot. □

October 5, 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](#), [1.1](#)
- [Pas06] Rafael Pass. Lecture 16: Non-malleability and public key encryption, October 2006. [1.2](#), [1.2](#)