Sparse Graph Obfuscation

October 4, 2023

1 Preliminaries

1.1 Bounded Functional Encryption

We will use the notation of static, bounded functional encryption as presented in [AR17].

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 $C_1, \ldots, C_d$.

Algorithm 1 $\mathrm{Exp}^{\mathrm{real}}_{\mathcal{F},\mathcal{A}}(1^\lambda)$

1: $(\mathrm{MPK}, \mathrm{MSK}) \leftarrow \mathrm{FE.Setup}(1^\lambda)$
2: $\mathrm{SK}_{C_i} \leftarrow \mathrm{FE.Keygen}(\mathrm{MSK}, C_i)$ for $i \in [d]$
3: $x_i \leftarrow \mathcal{A}(\mathrm{SK}_{C_i})$
4: $\mathrm{CT}_i \leftarrow \mathrm{FE.Enc}(\mathrm{MPK}, x_i)$
5: $\alpha \leftarrow \mathcal{A}(\mathrm{CT}_1, \ldots, \mathrm{CT}_d)$
6: **return** $x_1, \ldots, x_d, \alpha$

Algorithm 2 $\mathrm{Exp}^{\mathrm{ideal}}_{\mathcal{F},\mathrm{Sim}}(1^\lambda)$

1: $(\mathrm{MPK}, \mathrm{MSK}) \leftarrow \mathrm{FE.Setup}(1^\lambda)$
2: $\mathrm{SK}_{C_i} \leftarrow \mathrm{FE.Keygen}(\mathrm{MSK}, C_i)$ for $i \in [d]$
3: $x_i \leftarrow \mathcal{A}(\mathrm{SK}_{C_i})$
4: $\mathrm{CT}_i \leftarrow \mathrm{Sim}(1^\lambda, 1^{|x_i|}, \mathrm{MPK}, C_i, \mathrm{SK}_{C_i}, C_i(x_i))$
5: $\alpha \leftarrow \mathcal{A}(\mathrm{CT}_1, \ldots, \mathrm{CT}_d)$
6: **return** $x_1, \ldots, x_d, \alpha$

TODO: change to use this definition https://eprint.iacr.org/2019/314.pdf because in the above we can *simulate* SK_{C_i}. Actually no: use https://eprint.iacr.org/2021/847.pdf because statefulness / simplicity

Note that the adversary $\mathcal{A}$ and simulator are stateful but we do not include this in the above notation for simplicity.

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 $NM(m_1, \ldots, m_q, \mathcal{A})$ be a game as follows for $q = \mathrm{poly}(\lambda)$:

1. $(\mathrm{MPK}, \mathrm{MSK}) \leftarrow \mathrm{FE.Setup}(1^\lambda)$
2. $\mathrm{SK}_{C_i} \leftarrow \mathrm{FE.Keygen}(\mathrm{MSK}, C_i)$ for $i \in [d]$
3. $\mathrm{CT}_1, \ldots, \mathrm{CT}_q \leftarrow \mathrm{FE.Enc}(\mathrm{MPK}, m_1), \ldots \mathrm{FE.Enc}(\mathrm{MPK}, m_q)$
4. $c'_1, \ldots, c'_\ell \leftarrow \mathcal{A}(\mathrm{CT}, 1^{|m|})$
5. $m'_i \leftarrow \perp$ is $c_i = c_j$ for $j \in [q]$ and $\mathrm{FE.Dec}(\mathrm{SK}_{\mathrm{identity}}, c_i)$ 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 $\mathtt{negl}$ such that for all $\{m\}_0, \{m\}_1 \in \{0,1\}^{nq}$, we have

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

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

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

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 $\mathrm{FE.Dec}(\mathrm{SK}_{ID}, C_i(x)) \notin R(m)$ 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 $\mathrm{FE.Dec}(C_i(x)) \notin R(m)$.

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 $\phi : V \times \{0,1\}^\lambda \to \{0,1\}^{\text{poly}(\lambda)}$ 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,

$$\mathbf{Pr}[\mathcal{A}(C_\Gamma, v_0, v, \phi(K, v_0)) \in \text{Image}(\phi(K, v))] \leq O(v)\epsilon \tag{2}$$

for some fixed $\epsilon \leq \texttt{negl}(\lambda)$ and function C_Γ 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 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 FE.Enc(MPK, (K, v, r_2)) where encryption is done with randomness from r_1.

We can now define, C_Γ.

Algorithm 3 The circuit for the neighbor function, C_Γ.

function $\texttt{INNER}_i(K, v, r)$
  if $\Gamma(v) = \emptyset$ **then**
    return $0 \in \{0,1\}^*$
  $u_1, \ldots, u_d = \Gamma(v)$
  $u = u_i$
  $r_1, r_2 = \texttt{PRG}(r)$
  return FE.Enc(MPK, (u, K, r_2)) where we encrypt with randomness from r_1.
function $C_\Gamma(\phi(K, v))$
  for $i \in [d]$ **do**
    $u_i = \texttt{Dec}(\text{SK}_{\texttt{inner}_i}, \phi(K, v))$
  return $(u_1, \ldots, u_d)$

Before showing that our definition of ϕ and C_Γ satisfy eq. (2), we first must show that an adversary cannot find K.

Lemma 2.1. *Let $\mathcal{A}$ be a PPT adversary which can find K with probability $N\epsilon$. Then, there exists a PPT adversary, $\mathcal{B}$ which can break the FE scheme with probability ϵ. Or, given that the FE scheme is ϵ secure, then,*

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

Proof. We first prove the above but in the case of selective security. I.e. the adversary has to fix its query path at the start. We then use standard complexity leveraging techniques to achieve adaptive security.

We proceed via a series of hybrids. Note that for $|V| \geq \exp(\lambda)$, we require exponential hardness for the FE scheme. As such, let modified security parameter $\lambda' = \lambda + O(\log |V|)$.

- $\texttt{Hyb}_0$: 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 $\text{SK}_{\texttt{inner}_i} \leftarrow \text{FE.Keygen}(\text{MSK}, \texttt{inner}_i)$ for $i \in [d]$ and gives these keys to $\mathcal{A}$
  3. 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 $\mathcal{A}$.
  4. $\mathcal{A}$ outputs guess K' and wins if $K = K'$
- $\texttt{Hyb}_1$: We replace the *entire graph* with its simulated counterpart, starting at the root and moving to the leaves. TODO: we basically replace the ct with simulated, replace the output with true randomness, giving us a valid ct_next, then we move on to the next ct, etc.
- $\texttt{Hyb}_1$: We replace the *entire graph* with its simulated counterpart. We start with the leaves and then "work backwards" doing two steps at a time. Let $U_1, \ldots, U_D$ be a partition of V such that for all $u \in U_i$, the minimum distance from u to a the root is i. Starting with $j = D$, we do the following
  1. For vertex $u \in U_j$, we invoke the FE simulator to get
  $$\text{CT}'_u \leftarrow \text{Sim}\big(1^\lambda, 1^{TODO}, \text{MPK}, (\texttt{inner}_1, \ldots, \texttt{inner}_d), (\text{SK}_{\texttt{inner}_1}, \ldots \text{SK}_{\texttt{inner}_d}), (\texttt{inner}'_1(u), \ldots, \texttt{inner}'_d(u))\big)$$
  where $\texttt{inner}'_i$ is the same as $\texttt{inner}_i$ except that instead of using r_1 as randomness for FE.Enc, we use r'_1 which is true randomness that is fixed for u. Moreover, $\texttt{inner}'_i$ uses true, fixed randomness, r'_2, rather than r_2.
  2. For all $w \in V$ where $\Gamma(w)_i = u$, we replace $\texttt{inner}'_i(\phi(K, w))$ with CT'_u.
  3. If $j \neq 1$, we go back to step 1 with $j = j - 1$.

Assuming $\texttt{Hyb}_1$ and $\texttt{Hyb}_0$ computational indistinguishable, if we can guess K, then we can build an adversary $\mathcal{B}$ which can show that $\texttt{Hyb}_0$ and $\texttt{Hyb}_1$ are not indistinguishable. An adversary $\mathcal{B}$ can simply feed in the observables of $\texttt{Hyb}_0$ into $\mathcal{A}$ and $\texttt{Hyb}_1$ into $\mathcal{A}$. As $\texttt{Hyb}_1$ is independent of K, we necessarily have that $\mathbf{Pr}[\mathcal{A}(\texttt{Hyb}_1) = K] \leq \texttt{negl}(\lambda)$. Then, we can note that if $\mathcal{A}$ can guess K then, $\mathbf{Pr}[\mathcal{A}(\texttt{Hyb}_0) = K] > \texttt{negl}(\lambda)$. Thus, we have that $\mathbf{Pr}[\mathcal{A}(\texttt{Hyb}_0) = K] - \mathbf{Pr}[\mathcal{A}(\texttt{Hyb}_1) = K] > \texttt{negl}(\lambda)$ and thus $\texttt{Hyb}_0$ and $\texttt{Hyb}_1$ are not computationally indistinguishable.

Now we just have to show that $\texttt{Hyb}_0 \overset{c}{\approx} \texttt{Hyb}_1$. First, we can note that $\texttt{inner}_i(\phi(K, u)) \overset{c}{\approx} \texttt{inner}'_i(\phi(K, u))$ as if these distributions are indistinguishable, we can break the security of the PRG used in line 6 of algorithm 3. Now, we can note that if $\text{CT}_u = \phi(K, u)$, we have that $\text{CT}_u \overset{c}{\approx} \text{CT}'_u$ are indistinguishable if $\texttt{inner}_i(\text{CT}_u) \overset{c}{\approx} \texttt{inner}_i(\text{CT}'_u)$ for all $i \in [d]$. We can note that this holds true for u if u is a leaf as $\texttt{inner}_i(u) = 0$ and thus we can invoke the security of the FE simulator. Then, by step 2 in the above hybrid, we inductively create the hybrid such that $\texttt{inner}_i(\text{CT}_w) \overset{c}{\approx} \texttt{inner}_i(\text{CT}'_w)$. Thus, as we work backwards from the leaves, we have that $\text{CT}_u \overset{c}{\approx} \text{CT}'_u$ for all $u \in V$. Assuming that the FE scheme is ϵ secure and the distance from the distributions of $\texttt{inner}_i(\text{CT}_u)$ and $\texttt{inner}'_i(\text{CT}_u)$ is at most ϵ, then by the triangle inequality, we have that the distance between $\texttt{Hyb}_0$ and $\texttt{Hyb}_1$ is at most $O(|V|)\epsilon$. More formally, we have that for any PPT adversary, $\mathcal{B}$,

$$\big|\mathbf{Pr}[\mathcal{B}(\texttt{Hyb}_0) = 1] - \mathbf{Pr}[\mathcal{B}(\texttt{Hyb}_1) = 1]\big| \leq O(|V|)\epsilon.$$

□

Now, we will prove eq. (2). For convenience, we will restate eq. (2):

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

for all PPT $\mathcal{A}$.

Proof of eq. (2). 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 $p = v_0, \ldots, v$, there is some suffix of p, p', which the adversary does not know.

We are going to use a similar proof technique as in lemma 2.1 where we use a hybrid argument.

- $\texttt{Hyb}_0$: 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 $\text{SK}_{\texttt{inner}_i} \leftarrow \text{FE.Keygen}(\text{MSK}, \texttt{inner}_i)$ 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 $g \in \phi(K, v)$
- $\texttt{Hyb}_1$: We replace the entire graph with its simulated counterpart, such that the output of $\texttt{inner}_i$ is replaced with $\texttt{inner}'_i$ which encrypts its output ciphertext with true randomness.
- $\texttt{Hyb}_{1.5}$: We replace SK_{C_i} with its simulated equivalent
- $\texttt{Hyb}_2$: Consider this hybrid to be a series of hybrids indexed by ℓ:
  - For each ℓ, we replace the pre-neighb of first node in p' with $\perp$. We then note that by NM (see next lemma), we cannot get the next ct We then replace the next pre-img with bot and continue

Lemma 2.2. *Let $\mathcal{A}$ be a PPT adversary, then assuming NM of scheme we have that $\texttt{Hyb}_2$ step holds.*

Proof. Assume that adversary $\mathcal{A}$ can distinguish between $\texttt{Hyb}_1/\texttt{Hyb}_2^{\ell-1}$ and $\texttt{Hyb}_2^{\ell}$ with non-negligible probability. Then, we know that the adversary must have queried the pre-neighb of the node in p' as otherwise the hybrids are the same. But, this implies that $\mathcal{A}$ can generate the pre-image ciphertext CT_ℓ. We know show that we can create and adversary $\mathcal{B}$ which can break the NM of the scheme. For all ciphertexts that $\mathcal{A}$ has seen via graph traversal, we fix $\mathcal{A}_0$ in the NM encryption to produce these ciphertexts. We then note that we know $C_\Gamma(\phi(K, v_k))$ for $k \in \text{poly}(\lambda)$ and can then invoke the simulator S_1 to create a simulated function key $\text{SK}^*_{\texttt{inner}_i}$. We can then note that because the view of $\mathcal{B}$ (the NM advers) is enough to simulate $\text{SK}^*_{\texttt{inner}_i}$, we can then have $\mathcal{B}$ invoke $\mathcal{A}$ to produce, with non-negligible probability, CT_ℓ.

□

□

October 4, 2023

Abstract

References

[AR17] Shweta Agrawal and Alon Rosen. Functional encryption for bounded collusions, revisited. In *Theory of Cryptography Conference*, pages 173–205. Springer, 2017. 1.1

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