Sparse Graph Obfuscation

October 3, 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 $C_1, \ldots, C_d$.

See page 8 of [AR17] for now. I'll put in the actual definition later.

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

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

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

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

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

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,

$$\mathbf{Pr}[\mathcal{A}(C_\Gamma, v_0, v, \phi(K, v_0)) \in \text{Image}(\phi(K, v))] \leq \epsilon \tag{1}$$

for some fixed $\epsilon \leq \texttt{negl}(\lambda)$ 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 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 $0 \in \{0,1\}^*$
  $u_1, \dots, u_d = \Gamma(v)$
  $u = u_i$
  $r_1, r_2 = \texttt{PRG}(r)$
  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 $(u_1, \dots, u_d)$

Before showing that our definition of ϕ and C_Γ satisfy eq. (1), 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,*

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

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. 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 check whether or not we are in $\texttt{Hyb}_0$ or $\texttt{Hyb}_3$. $\mathcal{B}$ simply feeds in the distribution 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 we can distinguish $\texttt{Hyb}_0$ and $\texttt{Hyb}_1$.

Now we just have to show that $\texttt{Hyb}_0 \stackrel{c}{\approx} \texttt{Hyb}_1$. First, we can note that $\texttt{inner}_i(\phi(K, u)) \stackrel{c}{\approx} \texttt{inner}'_i(\phi(K, u))$ as if these distributions are indistinguishable, we can break the security of the $\texttt{PRG}$ used in line 6 of algorithm 3. Now, we can note that if $\text{CT}_u = \phi(K, u)$, we have that $\text{CT}_u \stackrel{c}{\approx} \text{CT}'_u$ are indistinguishable if $\texttt{inner}_i(\text{CT}_u) \stackrel{c}{\approx} \texttt{inner}_i(\text{CT}'_u)$ 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 $\texttt{inner}_i(\text{CT}_w) \stackrel{c}{\approx} \texttt{inner}_i(\text{CT}'_w)$. Thus, as we work backwards from the leaves, we have that $\text{CT}_u \stackrel{c}{\approx} \text{CT}'_u$ 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, $\mathcal{B}$,

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

$\square$

Lemma 2.2.

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

Proof.

- We think of giving the "null key" to the adversary. I.e. we give a func which always eval to null
- We then replace ϕ with its FE simulated counterpart
- We then say that $\phi(K, v_0)^*$ is independent of $\phi(K, v)$ b/c non-malleable

Assume that we can break the above hybrid. We then have that either break non-malleableness of you can find K □

Claim 2.3. *eq. (1) holds for any PPT adversary, $\mathcal{B}$ when C_Γ is implemented as in algorithm 3.*

Proof of Claim 2.3. We proceed via a hybrid argument and then show that if there exists an adversary $\mathcal{B}$ that can beak eq. (1), then we can build an adversary $\mathcal{A}$ which can distinguish between the hybrids.

- $\texttt{Hyb}_0$: Guess game. Win if guess
- $\texttt{Hyb}_1$: As the above but we replace $\phi(K, v_0)$ with its FE simulated counterpart as in the above lemma
- $\texttt{Hyb}_2$: We replace $\texttt{inner}_i$ ($\texttt{inner}_i^*$) to output $\perp$ if it is supposed to output v
- $\texttt{Hyb}_3$:

Now, note that if $\mathcal{B}$ can distinguish between **??**, then we can build adversary $\mathcal{A}$ to distinguish between $\texttt{Hyb}_0$ and $\texttt{Hyb}_2$ by invoking $\mathcal{B}$ to distinguish □

October 3, 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, 1.1