Sparse Graph Obfuscation

September 29, 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 $\mathrm{Exp}^{\mathrm{real}}_{\mathcal{F},\mathcal{A}}(1^\lambda)$

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** $x_1, \ldots, x_d, \alpha$

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 $C_\Gamma(\phi(K, u)) = \phi(K, \Gamma(u)_1), \ldots, \phi(K, \Gamma(u)_d)$ 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. (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 $N \geq \exp(\lambda)$, we require exponential hardness for the FE scheme.

- $\texttt{Hyb}_0$: In the first hybrid, the following game is played
  1. K is chosen at random, MPK, SK are generated for the functional encryption.
  2. The challenger generates $\text{SK}_{\texttt{inner}_i}$ for $i \in [d]$ and gives these keys to $\mathcal{A}$
  3. The challenger picks random r_1, r_2 and generates $\phi(K, v_0) = \text{FE.Enc}(\text{MPK}, (K, v_0, r_2))$ and gives this to $\mathcal{A}$.
  4. $\mathcal{A}$ outputs guess K' and wins if $K = K'$
- $\texttt{Hyb}_1$: We replace $\phi(K, v_0)$ with its FE simulated counterpart
- $\texttt{Hyb}_2$: When giving $\texttt{inner}_i(K, v_0, r_2) = \text{FE.Enc}(K, u_i, r_2')$ to the simulator and adversary, we replace r_2' with truly random $r_2'^*$ and the encryption to be done with true randomness r_1^*. Define $\texttt{inner}_i^*$ to be $\texttt{inner}_i$ except that r_1, r_2 are chosen at random.
- $\texttt{Hyb}_3$: Define an ordering for $\mathcal{G}$, $u_1, \ldots, u_g$ where u_1 is the root and $g = |V|$. Then, starting with $j = g$ and decrementing to $j = 1$, replace ϕ. Replace $\texttt{inner}_i$ with $\texttt{inner}_i^*$, we can use the same argument as above. Now, replace $\texttt{inner}_i^* *$ with $\texttt{inner}_i^*$ where $\texttt{inner}_i^* * (K, v, r) = \text{Sim}(blah, blah, K, v_i', r_2')$ TODO: fix for $v_i' = \Gamma(v)_i$ if $\Gamma(v) \neq \emptyset$ and otherwise with the 0 string. We can note that as $\mathcal{G}$ is a DAG, every vertex has a path to a leaf. As a leaf's output on $\texttt{inner}_i$ is 0, the simulation of a leaf's ciphertext is independent of K. Thus, as we work backwords, updating $\texttt{inner}_i^*$ with $\texttt{inner}_i^{**}$, we can note that the simulation for the input ciphertext to $\texttt{inner}_i^{**}$ is independent of K.

Now, if we can guess K, then we can check whether or not we are in $\texttt{Hyb}_0$ or $\texttt{Hyb}_3$ via feeding in things from $\texttt{Hyb}_0$/ $\texttt{Hyb}_3$ into $\mathcal{A}$. As $\texttt{Hyb}_3$ is indep of K, we are negligible in success prob definitionally, if we are in $\texttt{Hyb}_0$ then we can guess K with non-neg prob. Thus we gucci. We also have to specify hardness distance of like $|V|$ and say triangle inequality. □

Claim 2.2. *eq. (1) holds for any PPT adversary, $\mathcal{B}$ when C_Γ is implemented as in algorithm 3.*

Proof of Claim 2.2. We proceed via a hybrid argument and then show that if there exists an adversary $\mathcal{B}$ that can beak **??**, then we can build an adversary $\mathcal{A}$ which can distinguish between the hybrids.

- $\texttt{Hyb}_0$: As the LHS of the FE game
- $\texttt{Hyb}_1$: As the RHS of the FE game
- $\texttt{Hyb}_2$: As the above but we replace $\Pi_m = \big((\texttt{inner}_1, \texttt{inner}_1(v, K'), \ldots, (\texttt{inner}_d, \texttt{inner}_d(v, K')\big)$ for all $v \in \{\, v_1, ..., v_p \,\}$

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 □

September 29, 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