CS 276 – Cryptography Nov 5, 2014

Lecture 19: Indistinguishability Obfuscation

Instructor: Sanjam Garg *Scribe: Jingcheng Liu*

The problem of program obfuscation asks whether one can transform a program (e.g. circuits, Turing machines) to another semantically equivalent program (i.e. having the same input/output behavior), but is otherwise intelligible. It was originally formalized by Barak et al. who constructed a family of circuits that are non-obfuscatable under the most natural virtual black box (VBB) security.

1 VBB Obfuscation

As a motivation, recall that in a private-key encryption setting, we have a secret key k, encryption E_k and decryption D_k. A natural candidate for public-key encryption would be to simply release an encryption $E'_k \equiv E_k$ (i.e. E'_k semantically equivalent to E_k, but computationally bounded adversaries would have a hard time figuring out k from E'_k.

Definition 1 (Obfuscator of circuits under VBB) *O is an* obfuscator *of circuits if*

1. *Correctness:* $\forall c, O(c) \equiv c$.
2. *Efficiency:* $\forall c, |O(c)| \leq \text{poly}(|c|)$.
3. *VBB:* $\forall A, A$ *is PPT bounded,* $\exists$ *Sim (also PPT) s.t.* $\forall c$,

$$\left|\Pr\left[A\left(O(c)\right)=1\right]-\Pr\left[S^c(1^{|c|})=1\right]\right| \leq \text{negl}(|c|).$$

Similarly we can define it for Turing machines.

Definition 2 (Obfuscator of TMs under VBB) *O is an* obfuscator *of Turing machines if*

1. *Correctness:* $\forall M, O(M) \equiv M$.
2. *Efficiency:* $\exists q(\cdot) = \text{poly}(\cdot), \forall M\,(M(x)$ *halts in* t *steps* $\implies O(M)(x)$ *halts in* $q(t)$ *steps).*
3. *VBB: Let* $M'(t,x)$ *be a TM that runs* $M(x)$ *for* t *steps.* $\forall A, A$ *is PPT bounded,* $\exists$ *Sim (also PPT) s.t.* $\forall c$,

$$\left|\Pr\left[A\left(O(M)\right)=1\right]-\Pr\left[S^{M'}(1^{|M'|})=1\right]\right| \leq \text{negl}(\left|M'\right|).$$

Let's show that our candidate PKE from VBB obfuscator O is semantic secure, using a simple hybrid argument.

Proof. Recall the public key $PK = O(E_k)$. Let's assume E_k is a circuit, and we write it as c for short.

$$\begin{aligned}
&H_0 : A(\{(PK, E_k(m_0))\}) && \\
&H_1 : S^c(\{E_k(m_0)\}) && \text{by VBB} \\
&H_2 : S^c(\{E_k(m_1)\}) && \text{by semanti security of private key encryption} \\
&H_3 : A(\{(PK, E_k(m_1))\}) && \text{by VBB}
\end{aligned}$$

■

Now let's show the impossiblity result of VBB.

Theorem 1 *Let O be an obfuscator. There exists PPT bounded A, and a family (ensemble) of functions $\{H_n\}$, $\{Z_n\}$ s.t. for every PPT bounded simulator S,*

$$A\left(O(H_n)\right) = 1 \quad \& \quad A\left(O(Z_n)\right) = 0$$
$$\left|\Pr\left[S^{H_n}\left(1^{|H_n|}\right) = 1\right] - \Pr\left[S^{Z_n}\left(1^{|Z_n|}\right) = 1\right]\right| \leq \text{negl}(n).$$

Proof. Let $\alpha, \beta \xleftarrow{\$} \{0,1\}^n$.
We start by constructing $A', C_{\alpha,\beta}, D_{\alpha,\beta}$ s.t.

$$A'\left(O(C_{\alpha,\beta}), O(D_{\alpha,\beta})\right) = 1 \quad \& \quad A'\left(O(Z_n), O(D_{\alpha,\beta})\right) = 0$$
$$\left|\Pr\left[S^{C_{\alpha,\beta},D_{\alpha,\beta}}(\mathbf{1}) = 1\right] - \Pr\left[S^{Z_n,D_{\alpha,\beta}}(\mathbf{1}) = 1\right]\right| \leq \text{negl}(n).$$

$$C_{\alpha,\beta}(x) = \begin{cases} \beta, & \text{if } x = \alpha, \\ 0^n, & \text{o/w} \end{cases}$$
$$D_{\alpha,\beta}(c) = \begin{cases} 1, & c(\alpha) = \beta, \\ 0, & \text{o/w.} \end{cases}$$

Clearly $A'(X,Y) = Y(X)$ works. Now notice that input length to D grows as the size of $O(C)$. However for Turing machines which can have the same description length, one could combine the two in the following way:
$F_{\alpha,\beta}(b,x) = \begin{cases} C_{\alpha,\beta}(x), & b = 0 \\ D_{\alpha,\beta}(x), & b = 1 \end{cases}$.
Let $OF = O(F_{\alpha,\beta})$, $OF_0(x) = OF(0,x)$, similarly for OF_1, then A would be just $A(OF) = OF_1(OF_0)$.
Now assuming OWF exists, specifically we already have priavte-key encryption, we modify D as follows.

$$D_k^{\alpha,\beta}(1,i) = \text{Enc}_k(\alpha_i)$$
$$D_k^{\alpha,\beta}(2,c,d,\odot) = \text{Enc}_k(\text{Dec}_k(c) \odot \text{Dec}_k(d)), \text{where } \odot \text{ is a gate of AND, OR, NOT}$$
$$D_k^{\alpha,\beta}(3,\gamma_1,\cdots,\gamma_n) = \begin{cases} 1, & \forall i, \text{Dec}_k(\gamma_i) = \beta_i, \\ 0, & \text{o/w.} \end{cases}$$

Now the adversary A just simulate $O(C)$ gate by gate with a much smaller $O(D)$, thus we can use the combining tricks as for the Turing machines. ■

2 Indistinguishability Obfuscation

Definition 3 (Indistinguishability Obfuscation) iO$(\cdot)$ *is an* indistinguishability obfuscation *if* $\forall c_1, c_2$ *such that* $|c_1| = |c_2|$ *and* $c_1 \equiv c_2$, *we have*

$$\text{iO}(c_1) \overset{c}{\approx} \text{iO}(c_2).$$

Recall the witness encryption scheme, with which one could encrypt a message m to an instance x of an NP language L, such that $\mathrm{Dec}\left(x, w, \mathrm{Enc}\left(x, m\right)\right) = \begin{cases} m, \mathrm{if}(x, w) \in L, \\ \perp, \mathrm{o/w} \end{cases}$

Proposition 1 *Indistinguishability obfuscation implies witness encryption.*

Proof.
Let $C_{x,m}(w)$ be a circuit that on input w, outputs m if and only if $(x, w) \in L$.
Now we construct witness encryption as follows: $\mathrm{Enc}\left(x, m\right) = \mathrm{iO}\left(C_{x,m}\right), \mathrm{Dec}\left(x, w, c\right) = c(w)$.
Semantic security follows from the fact that, for $x \notin L$, $C_{x,m}$ is just a circuit that always output $\perp$, and by indistinguishability obfuscation, we could replace it with that constant circuit (padding if necessary), and then change the message, and change the circuit back, and we are done. ■

Proposition 2 *Indistinguishability obfuscation and OWF implies public key encryption.*

Proof.
We'll use a length doubling PRG $F : \{0,1\}^n \to \{0,1\}^{2n}$, together with a witness encryption scheme (E, D). The NP language for the encryption scheme would be the image of F.

$$
\begin{aligned}
\mathrm{Gen}(1^n) &= (PK = F(s), SK = s), s \xleftarrow{\$} \{0,1\}^n \\
\mathrm{Enc}\left(PK, m\right) &= E(x = PK, m) \\
\mathrm{Dec}\left(e, SK = s\right) &= D(x = PK, w = s, c = e).
\end{aligned}
$$

■

Proposition 3 *Every best possible obfuscator could be equivalently achieved with an indistinguishability obfuscation (up to padding and computationally bounded).*

Proof.
We prove by hand-waving.
Consider circuit c, the *best possible obfuscated* $BPO(c)$, and c' which is just padding c to the same size of $BPO(c)$. Computationally bounded adversaries cannot distinguish between $\mathrm{iO}\left(c'\right)$ and $\mathrm{iO}\left(BPO(c)\right)$.
Note that doing iO never decreases the "entropy" of a circuit, so $\mathrm{iO}\left(BPO(c)\right)$ is at least as secure as $BPO(c)$. ■