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. $\forall c$,

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

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: $\exists q(\cdot) = \text{poly} (\cdot), \forall M (M(x) \text{ halts in } t \text{ steps } \implies O(M)(x) \text{ halts in } q(t) \text{ steps})$.
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. $\forall c$,

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

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.

$$H_0 : A(\{(PK, E_k(m_0))\})$$
$$H_1 : S^c(\{E_k(m_0)\}) \quad \text{by VBB}$$
$$H_2 : S^c(\{E_k(m_1)\}) \quad \text{by semanti security of private key encryption}$$
$$H_3 : A(\{(PK, E_k(m_1))\}) \quad \text{by VBB}$$
Now let’s show the impossibility 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 \( S \),

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

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

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

\[
C_{\alpha,\beta}(x) = \begin{cases} 
\beta, & \text{if } x = \alpha \\
0^n, & \text{o/w}
\end{cases}
\]

\[
D_{\alpha,\beta}(c) = \begin{cases} 
1, & c(\alpha) = \beta \\
0, & \text{o/w}
\end{cases}
\]

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 private-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.

\[\blacksquare\]

## 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

\[
iO(c_1) \approx iO(c_2).
\]
Recall the witness encryption scheme, with which one could encrypt a message \( m \) to an instance \( x \) of an NP language \( L \), such that \( \text{Dec} (x, w, \text{Enc} (x, m)) = \begin{cases} m, & \text{if} (x, w) \in L, \\ ⊥, & \text{o/w} \end{cases} \)

**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: \( \text{Enc} (x, m) = \text{iO} (C_{x,m}), \text{Dec} (x, w, c) = c(w) \).
Semantic security follows from the fact that, for \( x \not\in L \), \( C_{x,m} \) is just a circuit that always output \( ⊥ \), 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 \( F \).

\[
\begin{align*}
\text{Gen}(1^n) &= (PK = F(s), SK = s), s \leftarrow \{0,1\}^n \\
\text{Enc} (PK, m) &= E(x = PK, m) \\
\text{Dec} (e, SK = s) &= D(x = PK, w = s, c = e).
\end{align*}
\]

**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 \( \text{BPO}(c) \), and \( c' \) which is just padding \( c \) to the same size of \( \text{BPO}(c) \). Computationally bounded adversaries cannot distinguish between \( \text{iO}(c') \) and \( \text{iO}(\text{BPO}(c)) \).
Note that doing \( \text{iO} \) never decreases the “entropy” of a circuit, so \( \text{iO}(\text{BPO}(c)) \) is at least as secure as \( \text{BPO}(c) \).