

On the (Im)possibility of Obfuscating Programs

Barak, Goldreich, Impagliazzo et al.

October 21, 2004

- 1 Introduction
 - What is program obfuscation?
 - Why is obfuscation desirable
- 2 Formalization
 - Obfuscators
 - Impossibility Results
- 3 Discussion

What is program obfuscation?

Informally:

- Making a program *unintelligible*.
- Functionality must be preserved.

Many heuristic approaches, but little theoretical work.
For example, *Twelve Days of Christmas*.

Why is obfuscation desirable?

- Software Protection
- Homomorphic Encryption
- Removing Random Oracle
- Private-key Encryption \rightarrow Public-key Encryption

Turing Machine Obfuscator

Definition

A *probabilistic algorithm* \mathcal{O} is a TM obfuscator if:

functionality For every TM M , the string $\mathcal{O}(M)$ describes a TM that computes the same function as M .

poly slowdown The length and running time of $\mathcal{O}(M)$ are at most polynomially larger than that of M .

black box For any PPT A , there is a PPT S and a negligible function α such that for all TMs M

$$|\Pr[A(\mathcal{O}(M)) = 1] - \Pr[S^M(1^{|M|}) = 1]| \leq \alpha(|M|)$$

2-TM Obfuscator

Definition

A 2-TM obfuscator like a TM obfuscator, except the black box property:

black box For any PPT A , there is a PPT S and a negligible function α such that for all TMs M, N

$$\begin{aligned} & |Pr[A(\mathcal{O}(M), \mathcal{O}(N)) = 1] - Pr[S^{M,N}(1^{|M|+|N|}) = 1]| \\ & \leq \alpha(\min(|M|, |N|)) \quad (1) \end{aligned}$$

Theorem

2-TM obfuscators do not exist.

Proof Outline

- Assume there exists a 2-TM obfuscator \mathcal{O} .
- Define 2 families of TMs:

$$C_{a,b}(x) \stackrel{\text{def}}{=} \begin{cases} b & x = a \\ 0^k & \text{otherwise} \end{cases} \quad D_{a,b}(C) \stackrel{\text{def}}{=} \begin{cases} 1 & C(a) = b \\ 0^k & \text{otherwise} \end{cases}$$

for strings $a, b \in \{0, 1\}^k$.

- Define Z_k to be a TM that always outputs 0^k .

Proof Outline (cont.)

- Consider adversary A : $A(C, D) = D(C)$, the following holds:

$$\Pr[A(\mathcal{O}(C_{a,b}), \mathcal{O}(D_{a,b})) = 1] = 1 \quad (2)$$

- For every PPT S :

$$|\Pr[S^{C_{a,b}, D_{a,b}}(1^k) = 1] - \Pr[S^{Z_k, D_{a,b}}(1^k) = 1]| \leq 2^{\Omega(k)} \quad (3)$$

where the probabilities are taken over a, b , and coin tosses of S .

- By definition of A :

$$\Pr[A(\mathcal{O}(Z_k), \mathcal{O}(D_{a,b})) = 1] = 0 \quad (4)$$

- 2 3 4 contradict the assumption.

Combining functions and programs

- For functions or TMs $f_0, f_1 : X \rightarrow Y$, define their combination $f_0 \# f_1 : \{0, 1\} \times X \rightarrow Y$ by $(f_0 \# f_1)(b, x) \stackrel{\text{def}}{=} f_b(x)$.
- Given any TM $C : \{0, 1\} \times X \rightarrow Y$, we can efficiently decompose C into $C_0 \# C_1 : C_b(x) \stackrel{\text{def}}{=} C(b, x)$.
- Having oracle access to $f_0 \# f_1$ is equivalent to having oracle access to f_0 and f_1 individually.

Impossibility of TM Obfuscator

Theorem

TM obfuscators do not exist.

Proof sketch:

- Assume \mathcal{O} is a TM obfuscator, and let $C_{a,b}$, $D_{a,b}$, and $Z_{a,b}$ be the TMs defined before.
- Define TMs $F_{a,b} = C_{a,b} \# D_{a,b}$ and $G_{a,b} = Z_{a,b} \# C_{a,b}$.
- On input a TM F , adversary A first decomposes F into $F_0 \# F_1$ and then outputs $F_1(F_0)$.

As in the previous proof, we have:

- $\Pr[A(\mathcal{O}(F_{a,b})) = 1] = 1$
- $\Pr[A(\mathcal{O}(G_{a,b})) = 1] = 0$
- $|\Pr[S^{F_{a,b}}(1^k) = 1] - \Pr[S^{G_{a,b}}(1^k) = 1]| \leq 2^{-\Omega(k)}$

where the probability is taken over a, b , and the coin tosses of A, S , and \mathcal{O} , which contradicts the assumption.

Discussion

- The paper showed that the *virtual black box* paradigm for obfuscators is inherently flawed. There may still be methods to make programs *untelligible* in a meaningful and precise sense.
- The proof relies on the construction of “unnatural” function families. (Does a more intuitive proof exist?)