Program Obfuscation

Rui Zhang
Nov 14, 2018

Program Obfuscation

- To *intentionally* make a program *unintelligible*, while preserving its functionality.

- **Goal**: Change program so still has same I/O behavior but is impossible to understand.
Example of obfuscated C code.
Applications of Obfuscation

• **For Software Protection**

  Software vendors want to prevent users from reverse-engineering executable code.

  Obfuscation helps to bypass antivirus, delay security research response.

• **For Cryptography**

  Many applications: fully homomorphic encryption, public key encryption, zero knowledge proofs, etc.
Obfuscation Definition

- **Definition (VBB Secure Obfuscation)**: A compiler $O$ is a virtual black box (VBB) secure obfuscator if it satisfies the following conditions:
  
  - **Functionality**: For every program (Turing Machine/Circuit) $P$, the string $O(P)$ describes a program that computes the same function as $P$.
    
    $$P(x) = (O(P))(x) \text{ for every } x \in \{0,1\}^n$$
  
  - **Polynomial Slowdown**: There is a polynomial $p$ such that for every program $P$,
    
    $$|O(P)| \leq p(|P|)$$
  
  - **Virtual Black Box Property**: For every efficient adversary $A$ mapping $\{0,1\}^*$ to $\{0,1\}$, there exists an efficient simulator $S$ such that for every program $P$, the following random variables are computationally indistinguishable.
    
    $$|\Pr[A(O(P)) = 1] - \Pr[S^{P}(1^{|P|}) = 1]| \leq a(|P|)$$
Virtual Black Box Property: for every efficient adversary $A$ mapping $\{0,1\}^*$ to $\{0, 1\}$, there exists an efficient simulator $S$ such that for every program $P$, the following random variables are computationally indistinguishable.

$$|\Pr[A(O(P)) = 1] - \Pr[S^P(1^{|P|}) = 1]| \leq a(|P|)$$
The Goal of Adversary

• What is the adversary trying to achieve?
  • A program that produces the same output as $P$.
  • A program that produces output with some relation to the output of $P$.
  • A function that computes some function of $P$.
  • Decide some property of $P$.

• The last achievement is the weakest, we want to prove that it is impossible.
Impossibility of Obfuscation

• VBB obfuscation is impossible in general.

• **Theorem:** There do not exist VBB obfuscators for Turing Machines.
2-TM Obfuscator

- **Definition (2-TM Obfuscator):** A 2-TM Obfuscator is defined in the same way as a TM obfuscator, except that the “virtual black box” property is strengthened as follows:

  - **Virtual Black Box Property:** For any adversary \( A \), there is an efficient simulator \( S \) such that for all TMs \( M \) and \( N \), the following random variables are computationally indistinguishable.

  \[
  |\Pr[A(O(M), O(N)) = 1] - \Pr[S^{M,N}(1^{\|M\|+\|N\|}) = 1]| \leq a(\min\{|M|,|N|\})
  \]
Obfuscating two TMs

- **Proposition**: 2-TM obfuscators do not exist.

- **Proof Sketch**:

  \[
  C_{\alpha,\beta}(x) = \begin{cases} 
  \beta & \text{if } x = \alpha \\
  0 & \text{if } x \neq \alpha
  \end{cases}
  \]

  \[
  D_{\alpha,\beta}(C) = \begin{cases} 
  1 & \text{if } C(\alpha) = \beta \\
  0 & \text{if } C(\alpha) \neq \beta
  \end{cases}
  \]

  Consider an adversary A, which, given two (obfuscated) TMs as input, simply run the second TM on the first one. That is 
  \[ A(C, D) = D(C). \]

  \[ \Pr[A(O(C_{\alpha,\beta}), O(D_{\alpha,\beta})) = 1] = 1 \]
Obfuscating two TMs

• Any simulator $S$ given oracle access to $C_{\alpha,\beta}$ and $D_{\alpha,\beta}$, cannot find an accepting input for either $C_{\alpha,\beta}$ or $D_{\alpha,\beta}$.

$$|\Pr[S^{C_{\alpha,\beta},D_{\alpha,\beta}}(1^k) = 1] - \Pr[S^{Z_k,D_{\alpha,\beta}}(1^k) = 1]| \leq 2^{-\Omega(k)}$$

• By Definition of $A$, we have:

$$\Pr[A(O(Z_k), O(D_{\alpha,\beta})) = 1] = 0$$

• The combination of these 3 equations contradict the fact that $O$ is a 2-TM obfuscator.
Theorem: TM obfuscators do not exist.

Proof Sketch:

$$F_{\alpha,\beta}(b, x) = \begin{cases} 
C_{\alpha,\beta}(x) & \text{if } b = 0 \\
D_{\alpha,\beta}(x) & \text{if } b = 1 
\end{cases}$$

$$G_{\alpha,\beta}(b, x) = \begin{cases} 
Z_k & \text{if } b = 0 \\
D_{\alpha,\beta}(x) & \text{if } b = 1 
\end{cases}$$

$$\Pr[A(O(F_{\alpha,\beta})) = 1] = 1$$

$$\Pr[A(O(G_{\alpha,\beta})) = 1] = 0$$

$$|\Pr[S^{F_{\alpha,\beta}}(1^k) = 1] - \Pr[S^{G_{\alpha,\beta}}(1^k) = 1]| \leq 2^{-\Omega(k)}$$
Impossibility for Circuits

- **Theorem**: VBB obfuscation for circuits does not exist.
  - In the case of boolean circuits, the previous proof does not work because the input for circuits is small.
  - The proof makes use of a type of homomorphic encryption scheme based on one-way functions.
  - They show that the one-way functions are implied by VBB obfuscators for circuits.
  - This contradiction proves unconditionally that VBB obfuscators for circuits do not exist.
Summary

• **Obfuscation Definition**
  • Functionality
  • Polynomial Slowdown
  • Virtual Black Box Property

• **Impossibility of Obfuscation**
  • Impossibility of VBB Obfuscation for Turing Machines
  • Impossibility of VBB Obfuscation for Circuits