Program Obfuscation

2021年5月6日 13:32

Intuition: to make a program unreadable, without changing its functionality.

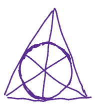
Def. An Obf.

1. $\forall$ TM P. $Obf(P) \to P^*$

   $\forall x \in D$. $P(x) = P^*(x)$ (may be relaxed to negl. difference)

2. $|P^*| \in poly(|P|)$, $Runtime(P^*) \in poly(Runtime(P))$

3. Security Non trivial.



→ Define as "Strong Virtual-Black-Box" (2000)

$Obf(P) \approx_c Sim^{P(\cdot)}(1^{|P|})$ (what you can learn from $Obf(P)$ is only oracle)

$\exists$ ppt Sim, such that $\forall P$, the above holds.

→ Define as "(normal) Virtual-Black-box". (2001)

for all ppt Adv. $\exists$ ppt Sim

$\{Adv(Obf(P)) = 1\} \approx \{Sim^{P(\cdot)}(1^{|P|}) = 1\}$

Looser def. no need to output a whole program

Only need to hide 1-bit information.

→ Yet Another Weaker Def:

Indistinguishability Obfuscation (IO)

if P_1, P_2 have the same input-output behaviour

$iO(P_1) \approx iO(P_2)$

Impossibility of (even normal) VBB:

THM: $\exists$ TM P. s.t. it is impossible to be VBB obfuscated.

proof. $P_k(x) = \begin{cases} \text{if } x = a. \text{ outputs } b. \\ \text{if } X(a) = b. \text{ (take TM of code } X\text{). outputs } c \\ \text{else output } 0. \end{cases}$

$k = a, b, c \xleftarrow{\$} U_n$

Remark: However, this doesn't work for $\{0,1\}^n \to \{0,1\}^n$. Since $|P^*_{a,b,c}| \notin \{0,1\}^n$

When given $\underbrace{Obf(P_k)}_{= P_k^*}$ Run: $P_k^*(P_k^*) = c$. $\Rightarrow c$ is learnable

However, c not learnable from oracle access $Sim^{P_k(\cdot)}(1^{|P_k|})$.

$\exists$ TM: $\{0,1\}^n \to \{0,1\}^n$ as well:

proof. Assume OWF. $\Longrightarrow$ private key Enc.

Another Counter Example

VBB of all P $\xrightarrow{\text{implies}}$ secret-key Fully homo Enc.

$Enc_k(m) \to C$. $Eval(C_1, C_2)$:

VBB $\left(\begin{array}{l} 1. \text{ Decrypt: } Dec_k(C_1) = m_1 \\ \qquad Dec_k(C_2) = m_2 \\ 2. \text{ Compute } m_1 \text{ NAND } m_2 = m \\ 3. \text{ Output } Enc_k(m) \end{array} \right)$

Remark: Secret-key FHE implies Public-key FHE

$P_{a,b,m,k}(x)$:

$\begin{cases} \text{if } Dec_k(x) = b, \text{ outputs } m \\ \text{if } x = 0^n. \text{ output } Enc_k(a) \\ \text{if } x = a. \text{ output } b. \\ \text{if } x = \text{"Eval"} | C_1 | C_2. \text{ Runs}(\cdot) = Eval(C_1, C_2) \end{cases}$

$\Rightarrow P^*(0^n) \to Enc_k(a)$

Homomorphically, Run P^* on $Enc_k(a) \to Enc_k(b)$

Enc_k as the function f

$P^*(Enc_k(b)) = m$

Witness Encryption:

(Enc, Dec)

- NP language L.
- $Enc(x, m \in \{0,1\}) = Enc_x(m) \mid x$
- $Dec(w, c) = \begin{cases} \text{if } C(x, w) = 1. \text{ output } m \\ \text{if not, output } \perp. \end{cases}$

Security: if $x \notin L$. $Enc(x, 0) \approx Enc(x, 1)$

Construct from iO:

$Enc(x, m) = iO\left(\begin{array}{l} \text{input: } w \\ \text{if } C(x,w) = 1: \text{ outputs } m \\ \text{else outputs } \perp \end{array} \right)$

Construction of iO.

1. for special functions

   - Point function $P_a(x) = \begin{cases} 1 & \text{if } x = a \\ 0 & \text{otherwise} \end{cases}$ $(a \leftarrow U(\{0,1\}^n))$

   use a PRG G, $y = G(a)$

   $iO(P_a)$: compute $G(x) := z$

   if $z = y$, $\Rightarrow 1$.

   otherwise 0.

   [Actually, this is even a (strong) VBB obfuscator, since a simulator outputs $y' \xleftarrow{\$} U(\{0,1\}^m)$. $y' \approx y$.]

   for $a \leftarrow D$ (any distribution)

   use random oracle H.

   $H, H(a) \approx H, y \leftarrow U_m$

   Generalize:

   $P_{\vec{a}}(x) = \begin{cases} 1 & \text{if } \langle a, x \rangle = 0 \mod q \\ 0 & \text{else.} \end{cases}$

   $\vec{a} \leftarrow \mathbb{Z}_q^n$. $(q = \exp(n))$

Generalize:

$$P_{\vec{a}}(x) = \begin{cases} 1 & \text{if } \langle a, x \rangle = 0 \mod q \\ 0 & \text{else.} \end{cases}$$

$\vec{a} \leftarrow \mathbb{Z}_q^n$. $(q = \exp(n))$

$\Rightarrow iO(P_{\vec{a}})$:

discrete-log $g^{a_1}, \dots g^{a_n}$

$g^{a_1 x_1}, \dots, g^{a_n x_n} = g^0$ or not.

2. For general function (a candidate)

From (Barrington 1986.)

For any $C \in NC^1$ (boolean circuit with log-depth)

$\downarrow$ matrix branching program

0	M_1^0	M_2^0	...			...
1	M_1^1	M_2^1	..			...
step	1	2	3	4	5	

$M_i^b \in \{0,1\}^{5\times5}$ suffices

and M_i^b permutation matrix

if input $\in \{0,1\}^4$. say $x = 1011$,

$\Rightarrow M_{out} = M_1^1 M_2^0 M_3^1 M_4^1 \vdots M_5^1 M_6^0 \cdots$

if $M_{out} = M_0 \Rightarrow 0$

$M_1 \Rightarrow 1$.