On Obfuscating Point Functions

A recent publication of Hoeteck Wee

Patrick Pletscher

ETH Zurich

April 22, 2005

Goals of today's lecture

Given a program that outputs 1 on exactly one input value, and 0 otherwise. We will study the obfuscation of this program.

Goals of today's lecture

Given a program that outputs 1 on exactly one input value, and 0 otherwise. We will study the obfuscation of this program.

- Is it possible? Yes! Although with some assumptions. So we should understand the implications of the assumptions.

Goals of today's lecture

Given a program that outputs 1 on exactly one input value, and 0 otherwise. We will study the obfuscation of this program.

- Is it possible? Yes! Although with some assumptions. So we should understand the implications of the assumptions.
- How to construct the obfuscator.

Goals of today's lecture

Given a program that outputs 1 on exactly one input value, and 0 otherwise. We will study the obfuscation of this program.

- Is it possible? Yes! Although with some assumptions. So we should understand the implications of the assumptions.
- How to construct the obfuscator.

We as well will hopefully learn more about *one-way functions* and *permutations* as our obfuscator is based on a probabilistic hash function.

Interesting aspects of this paper

- It is the first *positive result* about obfuscators for a general and interesting class of programs.

Interesting aspects of this paper

- It is the first *positive result* about obfuscators for a general and interesting class of programs.
- One of the few instances in cryptography where a random oracle can be replaced by a cryptographic construction.

Overview

Introduction

Short Repetition: Obfuscation

Feasibility of such obfuscators

Constructing Obfuscators for Point Functions

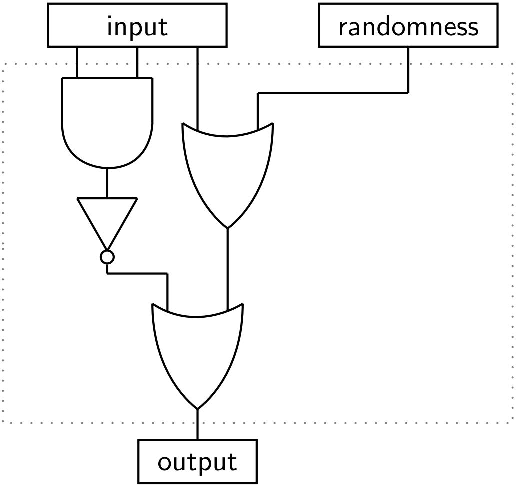
A special one-way permutation

A statistically collision-resistant hash function

Constructing the obfuscator

Short Repetition: (probabilistic) circuits

Circuit: standard boolean circuit with AND, OR and NOT gates.



Short Repetition: Obfuscation (intuitive)

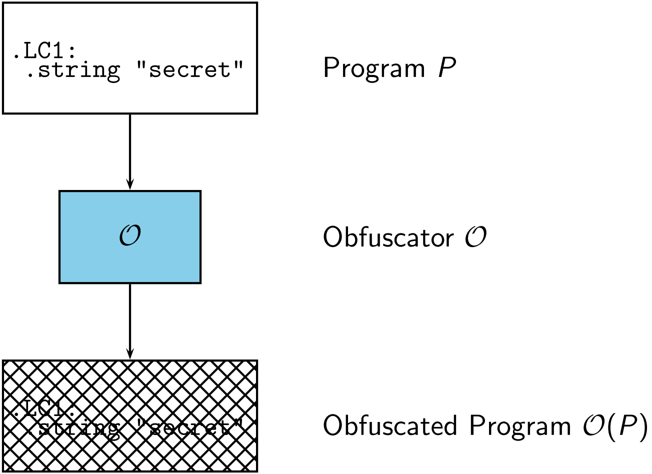


Figure: Idealized obfuscation

Short Repetition: Obfuscation (more formal)

Definition

A probabilistic polynomial-time algorithm $\mathcal{O}$ is an obfuscator for the family of circuits $\mathcal{C}_n$ (where $\mathcal{C}_n$ is the set of circuits in $\mathcal{C}$ that take inputs of length n) if the following three conditions hold:

Short Repetition: Obfuscation (more formal)

Definition

A probabilistic polynomial-time algorithm $\mathcal{O}$ is an obfuscator for the family of circuits $\mathcal{C}_n$ (where $\mathcal{C}_n$ is the set of circuits in $\mathcal{C}$ that take inputs of length n) if the following three conditions hold:

- approximate functionality

Short Repetition: Obfuscation (more formal)

Definition

A probabilistic polynomial-time algorithm $\mathcal{O}$ is an obfuscator for the family of circuits $\mathcal{C}_n$ (where $\mathcal{C}_n$ is the set of circuits in $\mathcal{C}$ that take inputs of length n) if the following three conditions hold:

- approximate functionality
- polynomial slowdown

Short Repetition: Obfuscation (more formal)

Definition

A probabilistic polynomial-time algorithm $\mathcal{O}$ is an obfuscator for the family of circuits $\mathcal{C}_n$ (where $\mathcal{C}_n$ is the set of circuits in $\mathcal{C}$ that take inputs of length n) if the following three conditions hold:

- approximate functionality
- polynomial slowdown
- "virtual black-box" property

Obfuscation: “Virtual black-box” property (I)

For every adversary A, there exists a simulator S, that only has black-box access to the circuit C. It should be indistinguishable whether A runs, or the simulator.

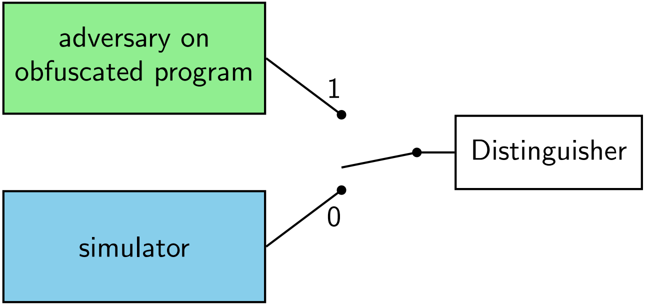


Figure: Indistinguishability of simulator and adversary

“Virtual black-box” property with a weak simulator (I)

Definition

An obfuscator $\mathcal{O}$ for a set of circuits $\mathcal{C}_n$ is (K, s, ϵ)-virtual black-box if it satifies the following: for any probabilistic circuit A of size s, there exists a probabilistic circuit S_A of size K such that for all circuits $C \in \mathcal{C}_n$,

$$\left| \Pr[A(\mathcal{O}(C)) = 1] - \Pr[S_A^C = 1] \right| < \epsilon$$

“Virtual black-box” property with a weak simulator (II)

- We work with circuits.

“Virtual black-box” property with a weak simulator (II)

- We work with circuits.
- Probability not negligible as in last week’s definition.

“Virtual black-box” property with a weak simulator (II)

- We work with circuits.
- Probability not negligible as in last week's definition.
- Adversary's circuit is bounded (by s).

Didn't Barak show the impossibility of such obfuscators?

Didn't Barak show the impossibility of such obfuscators?

- Yes and no.

Didn't Barak show the impossibility of such obfuscators?

- Yes and no.
- Barak et al. showed in their paper that efficient program obfuscators do not exist.

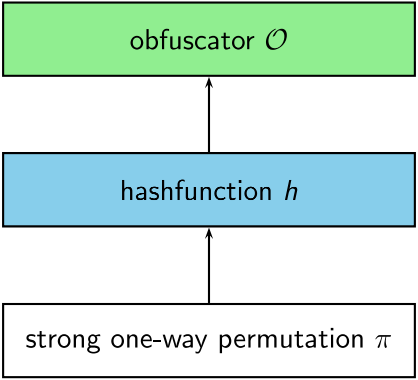
Didn't Barak show the impossibility of such obfuscators?

- Yes and no.
- Barak et al. showed in their paper that efficient program obfuscators do not exist.
- However this proof only holds if we need to obfuscate an arbitrary program.

Didn't Barak show the impossibility of such obfuscators?

- Yes and no.
- Barak et al. showed in their paper that efficient program obfuscators do not exist.
- However this proof only holds if we need to obfuscate an arbitrary program.
- So we can still hope to find an efficient obfuscator for restricted but nonethless useful classes of programs.

Constructing Obfuscators: The big picture



Assumption: A special one-way permutation exists

- strong one-way permutation $\pi : \{0,1\}^n \rightarrow \{0,1\}^n$ wherein any polynomial-sized circuit A of size $p(n)$ inverts the permutation on at most a polynomial number of inputs $q(n)$.

$$\Pr_{x \in U_n}[A_n(\pi(x)) = x] \leq q(n)/2^n.$$

Assumption: A special one-way permutation exists

- strong one-way permutation $\pi : \{0,1\}^n \to \{0,1\}^n$ wherein any polynomial-sized circuit A of size $p(n)$ inverts the permutation on at most a polynomial number of inputs $q(n)$.

$$\Pr_{x \in U_n}[A_n(\pi(x)) = x] \leq q(n)/2^n.$$

- Standard cryptographic assumptions assert hardness with respect to work, i.e. time over success probability.

A statistically collision-resistant hash function

Construction

Let $\pi : \{0,1\}^n \to \{0,1\}^n$ *be a permutation. We define a (public-coin) probabilistic function* $h : \{0,1\}^n \times \{0,1\}^{3n^2} \to \{0,1\}^{3n^2+3n}$ *as follows:*

$$h(x; \tau_1, \ldots, \tau_{3n}) := (\tau_1, \ldots, \tau_{3n}, \langle x, \tau_1 \rangle, \langle \pi(x), \tau_2 \rangle, \ldots, \langle \pi^{3n-1}(x), \tau_{3n} \rangle)$$

A statistically collision-resistant hash function

Construction

Let $\pi : \{0,1\}^n \rightarrow \{0,1\}^n$ *be a permutation. We define a (public-coin) probabilistic function* $h : \{0,1\}^n \times \{0,1\}^{3n^2} \rightarrow \{0,1\}^{3n^2+3n}$ *as follows:*

$$h(x; \tau_1, \ldots, \tau_{3n}) := (\tau_1, \ldots, \tau_{3n}, \langle x, \tau_1 \rangle, \langle \pi(x), \tau_2 \rangle, \ldots, \langle \pi^{3n-1}(x), \tau_{3n} \rangle)$$

$\langle x, y \rangle$ denotes the XOR of the AND of all bits (inner product):

$$\langle x, y \rangle := \bigoplus_{i=1}^{n} x_i \wedge y_i$$

A statistically collision-resistant hash function

Construction

Let $\pi : \{0,1\}^n \to \{0,1\}^n$ be a permutation. We define a (public-coin) probabilistic function $h : \{0,1\}^n \times \{0,1\}^{3n^2} \to \{0,1\}^{3n^2+3n}$ as follows:

$$h(x; \tau_1, \dots, \tau_{3n}) := (\tau_1, \dots, \tau_{3n}, \langle x, \tau_1 \rangle, \langle \pi(x), \tau_2 \rangle, \dots, \langle \pi^{3n-1}(x), \tau_{3n} \rangle)$$

$\langle x, y \rangle$ denotes the XOR of the AND of all bits (inner product):

$$\langle x, y \rangle := \bigoplus_{i=1}^{n} x_i \wedge y_i$$

Constructing the obfuscator: Some (possible) approaches

Random oracle

Apply a random oracle to the secret value and store output. Works. Approach taken by Lynn et al. however not very ingenious as RO has already obfuscator properties.

Constructing the obfuscator: Some (possible) approaches

Random oracle

Apply a random oracle to the secret value and store output. Works. Approach taken by Lynn et al. however not very ingenious as RO has already obfuscator properties.

Pseudorandom function

Use secret value as seed for pseudorandom function and store output of generator.
Will *not work* as pseudorandomness is only guranteed when secret value is chosen uniformly at random.

Constructing the obfuscator: Some (possible) approaches

Random oracle

Apply a random oracle to the secret value and store output.
Works. Approach taken by Lynn et al. however not very ingenious as RO has already obfuscator properties.

Pseudorandom function

Use secret value as seed for pseudorandom function and store output of generator.
Will *not work* as pseudorandomness is only guranteed when secret value is chosen uniformly at random.

The construction of Wee

Won't need a random oracle, but the hash function from the previous slide.

Obfuscators for point functions exist

Theorem

Suppose there exists a $(poly(n, 1/\epsilon)s, \frac{\epsilon K}{16n} \cdot \frac{1}{2^n})$-one-way permutation, then there exists a public-coin obfuscator for the family of point functions $\{I_x\}_{x \in \{0,1\}^n}$ which is $(Kpoly(n), s, \epsilon)$-virtual black-box.

The obfuscator for point functions

Given

- π a $(poly(n, 1/\epsilon)s, \frac{\epsilon K}{16n} \cdot \frac{1}{2^n})$-one-way permutation.
- h: hash function from previous construction based on π.

The obfuscator for point functions

Given

- π a $(poly(n, 1/\epsilon)s, \frac{\epsilon K}{16n} \cdot \frac{1}{2^n})$-one-way permutation.
- h: hash function from previous construction based on π.

The obfuscator $\mathcal{O}(I_x; R)$

1. store $h(x; R)$ (which contains R as a substring)
2. on input y check whether $h(y; R) = h(x; R)$. If so, output 1, 0 otherwise.

The simulator for point functions (I)

Non-uniform advice

Simulator S_A of size $Kpoly(n)$, that has non-uniform advice L about adversary hardwired into it:

$$L = \{x \in \{0,1\}^n : |\Pr_R[A(h(x;R)) = 1] - \Pr[A(U_{3n^2+3n}) = 1]| \geq \epsilon\}$$

The simulator for point functions (I)

Non-uniform advice

Simulator S_A of size $Kpoly(n)$, that has non-uniform advice L about adversary hardwired into it:

$$L = \{x \in \{0,1\}^n : |\Pr_R[A(h(x;R)) = 1] - \Pr[A(U_{3n^2+3n}) = 1]| \geq \epsilon\}$$

What's this set?

The set of all point functions (which are completely determined by x), for which A is able to distinguish the obfuscation of the point function from a random string.

The simulator for point functions (II)

1. S_A queries I_x on each element of L.

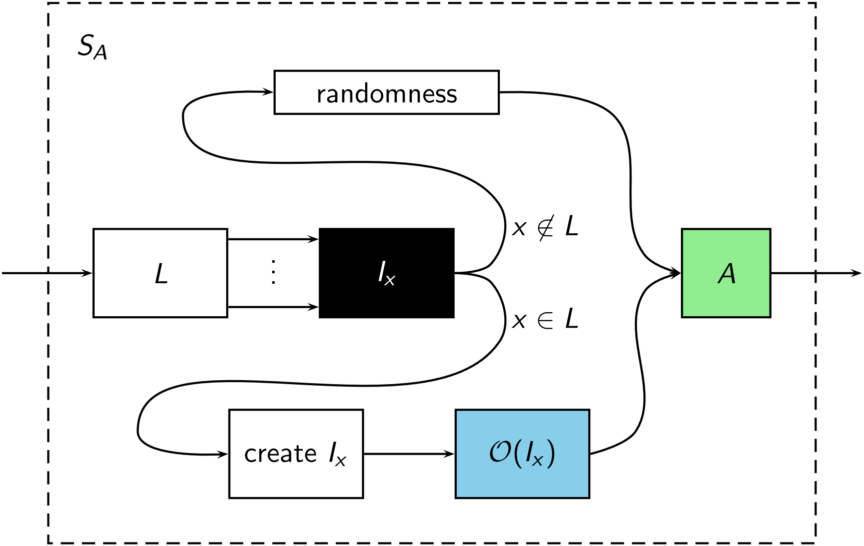
The simulator for point functions (II)

1. S_A queries I_x on each element of L.
2. If $x \in L$.
   2.1 Create the point function circuit.
   2.2 Obfuscate the resulting circuit.
   2.3 Output this obfuscated circuit to the adversary.

The simulator for point functions (II)

1. S_A queries I_x on each element of L.
2. If $x \in L$.
   2.1 Create the point function circuit.
   2.2 Obfuscate the resulting circuit.
   2.3 Output this obfuscated circuit to the adversary.
3. If $x \notin L$:
   3.1 Output a random string of correct length to the adversary.

The simulator for point functions (III)



The obfuscator satisfies "virtual black-box"

We will show

$$|L| \leq K - 1$$

The obfuscator satisfies "virtual black-box"

We will show

$$|L| \leq K - 1$$

But this implies "virtual black-box"

- As $|L| \leq K - 1$ we can hardwire all the advice L into the simulator.

The obfuscator satisfies "virtual black-box"

We will show

$$|L| \leq K - 1$$

But this implies "virtual black-box"

- As $|L| \leq K - 1$ we can hardwire all the advice L into the simulator.
- If $x \in L$, S_A computes a correct obfuscation of the point function I_x. A can by no means distinguish this from an obfuscation of the same point function.

The obfuscator satisfies "virtual black-box"

We will show

$$|L| \leq K - 1$$

But this implies "virtual black-box"

- As $|L| \leq K - 1$ we can hardwire all the advice L into the simulator.
- If $x \in L$, S_A computes a correct obfuscation of the point function I_x. A can by no means distinguish this from an obfuscation of the same point function.
- Else if $x \notin L$, the adversary will succeed with probability less than ϵ by the definition of L.

Excursus: Hybrid argument - Key idea

If m instances of a problem are easy, one instance of the same problem can not be hard.

Excursus: Hybrid argument - Key idea

If m instances of a problem are easy, one instance of the same problem can not be hard.

- Given a distinguisher A for m *instances* of a problem, which is successful with probability ϵ.

Excursus: Hybrid argument - Key idea

If m instances of a problem are easy, one instance of the same problem can not be hard.

- Given a distinguisher A for *m instances* of a problem, which is successful with probability ϵ.
- We can then create a distinguisher A' based on A for *one instance* of the same problem, with advantage ϵ/m.

Excursus: Hybrid argument - Example (semantic security)

Given: Distinguisher A with advantage ϵ

which is able to distinguish m encryptions

$$c_1, \ldots, c_m$$

from random values

$$r_1, \ldots, r_m.$$

Excursus: Hybrid argument - Example (semantic security)

Given: Distinguisher A with advantage ϵ

which is able to distinguish m encryptions

$$c_1, \ldots, c_m$$

from random values

$$r_1, \ldots, r_m.$$

One can create: Distinguisher for only one sample α

1. select a random i for $1 \leq i \leq m$.
2. Create the distribution

$$c_1, \ldots, c_{i-1}, \alpha, r_{i+1}, \ldots, r_m$$

3. Let A run on this distribution. We have advantage ϵ / m.

Proof of $|L| \leq K - 1$: Overview / Getting started

Contradiction if $|L| \geq K$

We can construct a circuit that inverts the one-way function on $\frac{\epsilon K}{16n}$ inputs. In the following we set $|L| = K$.

Proof of $|L| \leq K - 1$: Overview / Getting started

Contradiction if $|L| \geq K$

We can construct a circuit that inverts the one-way function on $\frac{\epsilon K}{16n}$ inputs. In the following we set $|L| = K$.

$$L = \{x \in \{0,1\}^n : |\Pr_R[A(h(x;R)) = 1] - \Pr[A(U_{3n^2+3n}) = 1]| \geq \epsilon\}$$

Proof of $|L| \leq K - 1$: Overview / Getting started

Contradiction if $|L| \geq K$

We can construct a circuit that inverts the one-way function on $\frac{\epsilon K}{16n}$ inputs. In the following we set $|L| = K$.

$$L = \{x \in \{0,1\}^n : |\Pr_R[A(h(x;R)) = 1] - \Pr[A(U_{3n^2+3n}) = 1]| \geq \epsilon\}$$

Getting rid of the absolute value

There exists a subset L' of L of size at least $K/2$ such that for all $x \in L'$:

$$\Pr_R[A(h(x;R)) = 1] - \Pr[A(U_{3n^2+3n}) = 1] \geq \epsilon$$

Proof of $|L| \leq K - 1$: Hybrid argument

for all $x \in L'$:

$$\mathrm{Pr}_R[A(h(x;R)) = 1] - \mathrm{Pr}[A(U_{3n^2+3n}) = 1] \geq \epsilon$$

The hybrid argument

yields an j, for which we can distinguish $\langle \pi^{j-1}(x), \tau_j \rangle$ from a random bit, with advantage (average over $x \in L'$)

$$\geq 1/2 + \epsilon/3n$$

Proof of $|L| \leq K - 1$: Averaging argument

For $x \in L'$ we have advantage (average)

$$\geq 1/2 + \epsilon/3n$$

Now there must be at least a $\epsilon/6n$ fraction of x in L', for which we have advantage

$$\geq 1/2 + \epsilon/6n$$

Proof of $|L| \leq K - 1$: Averaging argument

For $x \in L'$ we have advantage (average)

$$\geq 1/2 + \epsilon/3n$$

Now there must be at least a $\epsilon/6n$ fraction of x in L', for which we have advantage

$$\geq 1/2 + \epsilon/6n$$

To see this, examine the worst case: Nearly all x with probability $1/2 + \epsilon/6n - \delta$ ($\delta \to 0$) and very few with probability 1:

$$\frac{\epsilon}{6n} \cdot 1 + \left(1 - \frac{\epsilon}{6n}\right) \cdot \left(\frac{1}{2} + \frac{\epsilon}{6n} - \delta\right)$$
$$= \frac{1}{2} + \frac{\epsilon}{3n} + \delta\frac{\epsilon}{6n} - \frac{\epsilon}{12n} - \frac{\epsilon^2}{36n^2} - \delta \leq \frac{1}{2} + \frac{\epsilon}{3n}$$

In the worst case we get a smaller average than we should get.

Proof of $|L| \leq K - 1$: Putting it all together

- For a $\epsilon/6n$ fraction of L' we have advantage $\geq 1/2 + \epsilon/6n$.
- An older theorem of Goldreich and Levin proves that we can invert $\pi(x)$ (i.e. find the x) with probability $3/4$, if we can guess $\langle \pi(x), \tau \rangle$ with some advantage.
- $|L'| \geq K/2$.

So the total advantage is:

$$\frac{\epsilon}{6n} \cdot \frac{K}{2} \cdot \frac{3}{4} = \frac{\epsilon K}{16n}$$

I hope you don't feel this way . . .

I hope you don't feel this way ...

Take home message

Obfuscation of point functions is possible, although with some assumptions/simplifications.