[BGI+12]
Barak et al. "On the (Im)possibility of Obfuscating Programs".
CRYPTO 01 and Journal of the ACM 2012.

SAPIENZA
UNIVERSITÀ DI ROMA

1

Obfuscation.

Goal of obfuscation: Make a program "unintelligible" while preserving its functionality. Intuitively obfuscator = virtual black-box: anything one can compute from it, one can also compute from the input-output behaviour of the program. So an obfuscator O is a "compiler" that given a program (or circuit) P outputs a new program $O(P)$ such that:

-) FUNCTIONALITY. $O(P)$ computes the same function as P.
-) VBB. Anything one can compute from $O(P)$, can be eff. computed given oracle access to P.

Applications of obfuscation:
1) Software protection. Protect against reverse engineering. Factoring example. Watermarking.
2) Replacing a RO. Obfuscate a family of PRFs.
3) Converting SKE into PKE. Obfuscate $Enc(K, \cdot)$ of the SKE.
4) Homomorphic encryption. Obfuscate $Enc(K, Dec(K, \cdot) + Dec(K, \cdot))$.

Main result: general purpose VBB obfuscation is impossible.

1. PLAN:
-) Definitions of obfuscation.
-) The main impossibility proof.
-) Impossibility of applications.
-) Discussion.

2. DEFINITIONS. Main definition for circuits and TMs. A circuit C computes $f: \{0,1\}^n \to \{0,1\}^m$ if $\forall x \in \{0,1\}^n$, $C(x) = f(x)$. AND, OR, NOT and randomness gates.
Oracle access: $A^P(x)$ output of algorithm A on input x, given

oracle access to P. If P is a TM we actually mean A is allowed to run P on some input x for say t steps getting output y.

Main definitional issue: How to formalize "what the adversary can learn". Strongest would be computational indistinguishability: given oracle access to P, produce an output distribution that is $\approx_c$ from what the adversary computes given $O(P)$. Here is a distinguisher: $D(P, P')$ accepts if P' and P agree on many randomly chosen inputs (with $P' \approx O(P)$).

Weakest: restrict to satisfy a relation to one bit functions.

DEF (TM obfuscator). A PPT algorithm O is a TM obfuscator for a family $\mathcal{F}$ of TMs if it satisfies:

-) FUNCTIONALITY. $\forall M \in \mathcal{F}$, $O(M)$ describes a TM computing M.

-) POLYNOMIAL SLOW-DOWN. Length and running time of $O(M)$ at most polynomially longer than that of M: $\exists$ poly$(\cdot)$ s.t. $\forall M \in \mathcal{F}$, $|O(M)| \le \text{poly}(|M|)$ and if M halts after t steps, $O(M)$ halts after poly(t) steps (on input x).

-) VBB. $\forall$ PPT A, $\exists$ a PPT S and a negligible function ε s.t. $\forall M \in \mathcal{F}$ we have $|\Pr[A(O(M))=1] - \Pr[S^M(1^{|M|})=1]| \le \varepsilon(|M|)$.

Circuit obfuscator: as above, but $\mathcal{F}$ is a collection of circuits.

PROP. If a TM obfuscator exists, then a circuit obfuscator exists.

PROOF. Can emulate a circuit via a TM but not the other way.

3. IMPOSSIBILITY. We just show the essence of it. We start with a definition.

DEF (2-TM obfuscator). As a TM-obfuscator, but:

-) VBB. $\forall$ PPT A, $\exists$ a PPT S and $\varepsilon(\cdot)$ such that $\forall M, N \in \mathcal{F}$ we have

$$|\Pr[A(O(M), O(N))=1] - \Pr[\cancel{A}\, S^{M,N}(1^{|M|+|N|})=1]| \le \varepsilon(\min\{|M|, |N|\})$$

LEMMA. Neither 2-TM nor 2-circuit obfuscator exists.

Intuition: There is a difference between having BB access to a program and getting access to the program itself no matter how obfuscated. The proof will use a function that is hard to learn via oracle queries.

PROOF. Suppose $\exists$ a 2-TM obfuscator $\mathcal{O}$. For $\alpha, \beta \in \{0,1\}^k$ define a TM

$$C_{\alpha,\beta}(x) = \begin{cases} \beta & \text{if } x = \alpha \\ 0^k & \text{otherwise} \end{cases}$$

2

SAPIENZA
UNIVERSITÀ DI ROMA

We assume that on input x, $C_{\alpha,\beta}$ runs in time $10 \cdot |x|$ (convention).
Now consider a TM $D_{\alpha,\beta}$, as follows:

$$D_{\alpha,\beta}(C) = \begin{cases} 1 & \text{if } C(\alpha) = \beta \\ 0 & \text{otw} \end{cases}$$

where C is the code of a TM. The above is not well-defined and later we will just run a modified $D_{\alpha,\beta}$ that runs $C(\alpha)$ for $\mathrm{poly}(k)$ steps and outputs 0 in case C does not halt within that many steps.
Consider the following adversary A: given two TMs C, D, adversary A outputs $A(C, D) = D(C)$. (Again later run D on C for poly steps.)
Now, $\forall \alpha, \beta \in \{0,1\}^k$ we have

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

Let Z_k be a TM always outputting 0^k. Then $\forall$ PPT S we must have:

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

The above prob. is taken over the choice of $\alpha, \beta \leftarrow \{0,1\}^k$ and the coins of S.
Finally,

$$\Pr[A(O(Z_k), O(D_{\alpha,\beta})) = 1] = 2^{-k}$$

as $D_{\alpha,\beta}(Z_k) = 1$ iff $\beta = 0^k$. Note that (1)+(2)+(3) yields the lemma.
Note that Eq. (1) involves $A(O(D_{\alpha,\beta}), O(C_{\alpha,\beta}))$ which is equivalent to $O(D_{\alpha,\beta})(O(C_{\alpha,\beta}))$, which is equivalent to $D_{\alpha,\beta}(O(C_{\alpha,\beta}))$ which requires executing $O(C_{\alpha,\beta})(\alpha)$ which is equivalent to ~~running~~ $C_{\alpha,\beta}(\alpha)$.
By polynomial slow-down the latter requires $\mathrm{poly}(10 \cdot k) = \mathrm{poly}(k)$ steps.
Thus A needs only to run for $\mathrm{poly}(k)$ steps. ■
For the circuit case, it suffices to assume that $D_{\alpha,\beta}$ have input of size $\mathrm{poly}(k)$.
Next, we present the main result.
THM. No TM-obfuscator exists.
Proof. For TMs $f_0, f_1 : X \to Y$, define $f_0 \# f_1 : \{0,1\} \times X \to Y$ as
$(f_0 \# f_1)(b, x) = f_b(x)$

Assume that there is a TM-obfuscator O. For $\alpha, \beta \in \{0,1\}^k$ let $C_{\alpha,\beta}$ and $D_{\alpha,\beta}$ as above, and Z_k as above. Consider $F_{\alpha,\beta} = C_{\alpha,\beta} \# D_{\alpha,\beta}$ and $G_{\alpha,\beta} = Z_k \# D_{\alpha,\beta}$. Consider adversary A that given a TM $F : \{0,1\} \times X \to Y$ first decomposes F into $F_0 \# F_1$ and then outputs $F_1(F_0)$. (As before A runs in poly$(|F|)$ time). Let S be the PPT simulator for A, we have (as before)

$$\Pr[A(O(F_{\alpha,\beta}))=1] - \Pr[A(O(G_{\alpha,\beta}))=1] = 1-2^{-k}$$

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

This is a contradiction. ∎

In the circuit settings things get more complicated. Note that above the adversary A evaluates $F_1(F_0)$ where $F_0 \# F_1 = O(F_{\alpha,\beta}) = O(C_{\alpha,\beta} \# D_{\alpha,\beta})$. For this to make sense the size of F_0 must be at most the size of F_1. But since the output of $F_0 \# F_1$ can be polynomially longer than its input $C_{\alpha,\beta} \# D_{\alpha,\beta}$, there is no such guarantee.

However, one can get around this and show the following theorem (complicated):

<u>THM</u>. If one-way functions exist, no circuit obfuscator exists.

The above result is actually unconditional, since:

<u>THM</u>. If efficient circuit obfuscator exists, then one-way functions exists.

Proof. For $\alpha \in \{0,1\}^k$ and $b \in \{0,1\}$, let $C_{\alpha,b} : \{0,1\}^k \to \{0,1\}$ be defined as

$$C_{\alpha,b}(x) = \begin{cases} b & \text{if } x = \alpha \\ 0 & \text{otw} \end{cases}$$

Define $f_k(\alpha, b, r) = O(C_{\alpha,b}; r)$. Show that $f = \bigcup_{k\in\mathbb{N}} f_k$ is a owf. It is efficient. Also b is determined by $f_k(\alpha, b, r) \Rightarrow$ if b is a hard-core bit for f_k, then f_k is hard to invert. To prove that b is a hard-core bit first note that for all PPT S:

$$\Pr_{\alpha,b}[S^{C_{\alpha,b}}(1^k) = b] \leq 1/2 + negl(k).$$

By VBB property of O, it follows that for all PPT A:

$$\Pr_{\alpha,b,r}[A(f(\alpha,b,r)) = b] = \Pr_{\alpha,b}[A(O(C_{\alpha,b})) = b] \leq 1/2 + negl(k).$$ ∎

<u>COROLLARY</u>. Efficient circuit obfuscator do not exist.

4. <u>IMPOSSIBILITY OF SOME APPLICATIONS</u>. Until now we ruled out some basic generic definitions for obfuscation. Here we actually even rule out the applications.

(a) PKE from SKE. Given $SKE = (G, E, D)$ publish $PK = O(E(K, \cdot))$.

<u>THM</u>. If there exists secure probabilistic SKE schemes, then there exists ones that are unobfuscatable.

Unobfuscatable means that $\exists$ a PPT A such that $\forall K \leftarrow G(1^k)$ and $\forall \tilde{E}(K, \cdot) = O(E(K, \cdot))$ such that $\tilde{E}(K, \cdot)$ and $E(K, \cdot)$ are stat. close, $A(\tilde{E}(K, \cdot)) = K$.

(b) Fiat-Shamir signatures.

$$\begin{array}{ccc} P_r & \xrightarrow{\text{com}} & V_r \\ & \xleftarrow{\text{ch}} & \\ & \xrightarrow{\text{resp}} & \end{array} \Rightarrow$$

$SS = (G, S, V)$

$\sigma = (com, ch, resp) = S(sk, m)$

$com \leftarrow P_1(1^k)$ $\quad ch = \mathcal{H}(m, com, pk)$

$resp \leftarrow P_2(ch, com, sk)$

<u>THM</u>. Assume that OWF exists. There exists $\{h_{K}\}_{K \in \{0,1\}^*}$ s.t. replacing $\mathcal{H}$ by an obfuscation of h_K yields an insecure scheme.

5. <u>DISCUSSION</u>. Where do we go from here? Can we obtain any positive result about cryptographic obfuscation? Proposal for next talks:

1. Barak et al. only rule out VBB obfuscation for *all* programs.
Natural direction: obfuscate *specific* programs or (possibly weakening the security definition in a way that still allows for applications).
Lynn, Prabhakaran, Sahai. "Positive results and techniques for obfuscation". EC 04.
Wee. "On obfuscating point functions". STOC 05.

2. Constructions for weaker definitions of obfuscation (general purpose).

<u>DEF</u>. (iO). $\forall$ PPT A, $\exists$ negl. func. ε such that $\forall C_1, C_2$ that compute the same function and are of the same size k:

$$|\Pr[A(O(C_1)) = 1] - \Pr[A(O(C_2)) = 1]| \leq \varepsilon(k).$$

<u>DEF</u>. (eO, a.k.a. diO). $\forall$ PPT A, $\exists A'$ and $\varepsilon(\cdot)$ s.t. the following holds. Suppose C_1, C_2 are circuits of size k, s.t.

$$\varepsilon' = |\Pr[A(O(C_1)) = 1 - \Pr[A(O(C_2)) = 1]| > \varepsilon(k).$$

Then $\forall$ C_1', C_2' of size κ s.t. C_i' computes the same as C_i, we have that $A'(C_1', C_2')$ outputs an input on which C_1, C_2 differ in time poly$(\kappa, 1/(\varepsilon' - \varepsilon(\kappa)))$.

Garg, Gentry, Halevi, Raykova, Sahai, Waters. "Candidate Ind. obfuscation and functional encryption for all circuits". FOCS 2013.

Boyle, Chung, Pass. "On Extractability (a.k.a. Differing-Inputs) Obfuscation". TCC 2014.

Pass, Seth, Telang. "Ind. Obfuscation from Semantically-Secure Multilinear Encodings". CRYPTO 2014.

Ananth, Gupta, Ishai, Sahai. "Optimizing Obfuscation: Avoiding Barrington's Theorem". CCS 2014.

3. Applications of iO.

Sahai, Waters. How to use iO: deniable encryption and more. STOC 2014.

*[GGHR14]. "Two-round secure MPC from iO". TCC 2014.

[HSW14]. "Replacing a RO: FDH from iO". EC 2014.

[BZ13]. "Multiparty KE, efficient traitor-tracing, and more from iO". CRYPTO14.

[BST13]. "Poly-many hard-core bits for any one-way function". IACR ePrint. ACM.

[MR13]. "There is no iO in Pessiland". IACR ePrint.

[BM14]. "Using iO via UCEs". AsiaCrypt 2014.

For 1. above mention also:

[HRsV07] "Securely obfuscating Re-encryption". TCC 2007.

[Had10] "Secure obfuscation for encrypted signatures". EC 2010.

[GGHW14] "On the implausibility of Differing-Inputs obfuscation and extractable witness encryption with auxiliary input".