Recent Results in Code Obfuscation

Alexander Antrobus

Gatsby Unit
University College London

March 11th 2016

What does this code do?

```
/*
  LEAST LIKELY TO COMPILE SUCCESSFULLY:
  Ian Phillipps, Cambridge Consultants Ltd., Cambridge, England
*/

#include <stdio.h>
main(t,_,a)
char
*
a;
{
    return!

0<t?
t<3?

main(-79,-13,a+
main(-87,1-_,
main(-86, 0, a+1 )

+a)):

1,
t<_?
main(t+1, _, a )
:3,

main ( -94, -27+t, a )
&&t == 2 ?_
<13 ?
```

```
main ( 2, _+1, "%s %d %d\n" )

:9:16:
t<0?
t<-72?
main( _, t,
"@n'+,#'/*{}w+/w#cdnr/+,{}r/*de}+,/*{*+,/w{%+,/w#q#n+,/#{l,+
#q#n+,/+k#;*+,/'r :'d*'3,}{w+K w'K:'+}e#';dq#'l q#'+d'K#!/+k
q#'r}eKK#}w'r}eKK{nl]'/#;#q#n'){)#}w'){){nl]'/+#n';d}rw' i;#
r{#w'r nc{nl]'/#{l,+'K {rw' iK{;[{nl]'/w#q#\
\
n'wk nw' iwk{KK{nl]!/w{%'l##w#' i; :{nl]'/*{q#'ld;r'}{nlwb!/
{nl'-{}rw]'/+,}##'*}#nc,',#nw]'/+kd'+e}+;\
#'rdq#w! nr'/ ') }+}{rl#'{n' ')# }'+}##(!!/")
:
t<-50?
_==*a ?
putchar(31[a]):

main(-65,_,a+1)
:
main((*a == '/') + t, _, a + 1 )
:

0<t?

main ( 2, 2 , "%s")
:*a=='/'||

main(0,

main(-61,*a, "!ek;dc i@bK'(q)-[w]*%n+r3#l,{}:\nuwloca-O;m .v

,a+1);}
```

It's *C* and generates...

On the First day of Christmas my true love sent to me a Partridge in a Pear Tree.

On the Second day of Christmas my true love sent to me Two Turtle Doves and a Partridge in a Pear Tree.

On the Third day of Christmas my true love sent to me Three French Hens, Two Turtle Doves and a Partridge in a Pear Tree.

. . . . . (12 verses)

(winner of International Obfuscated C Code Contest 1988, written by Ian Phillipps)

Code Obfuscation

...is the deliberate act of creating obfuscated code, i.e. source or machine code that is difficult for humans to understand. Like obfuscation in natural language, it may use needlessly roundabout expressions to compose statements.

Useful?

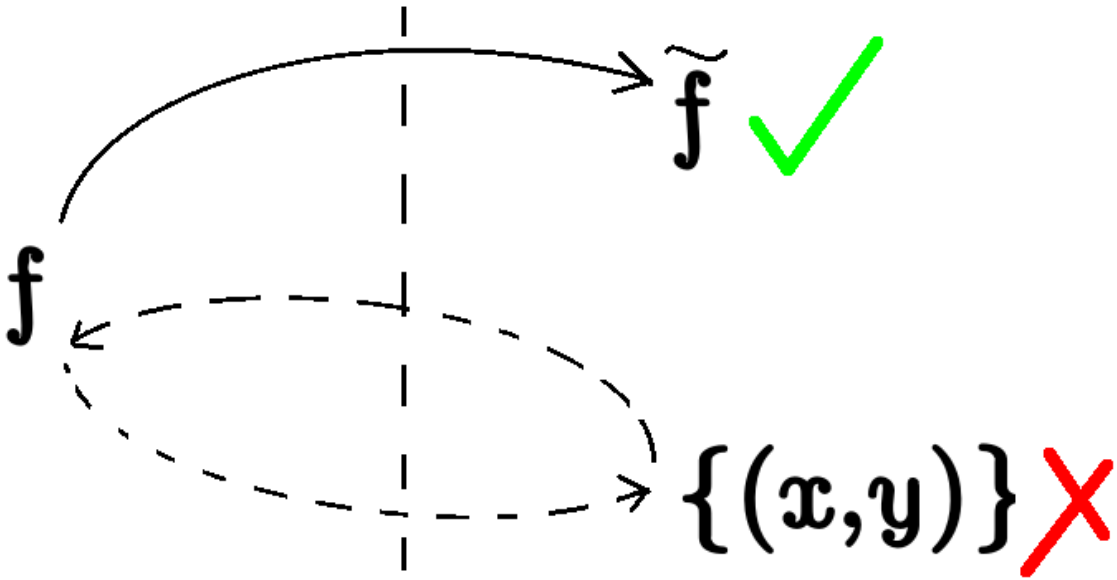
- Protect intellectual property (hard to copy programme)
- sending 'agents' onto malicious machines... robust to 'interrogation'.

History

In 2001, Barak, Goldreich and others...showed

(universal) Obfuscation Impossibility

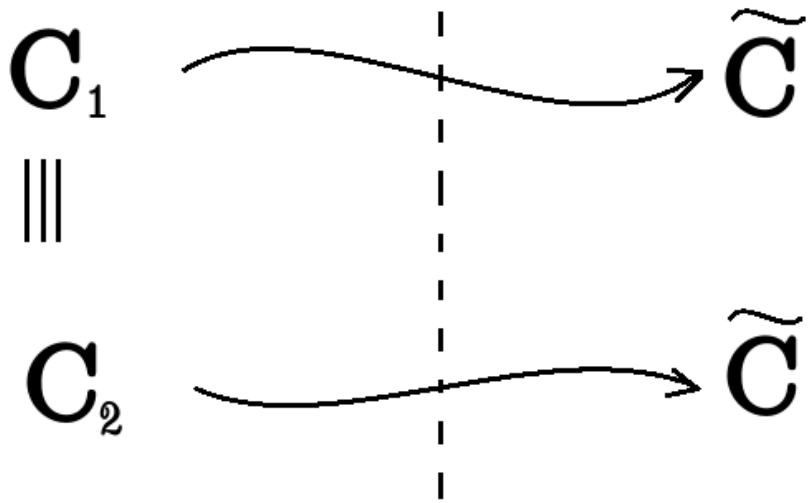
There exists a family of functions $\mathcal{F}$ with indicator function $s : \mathcal{F} \to \{0, 1\}$ such that, given *any* circuit which implements a $f \in \mathcal{F}$, there exists an efficient algorithm to compute $s(f)$, even though, given *oracle access* to f, $s(f)$ cannot be determined in any efficient sense.



Indistinguishability Obfuscation

"Indistinguishability obfuscation requires that given any two equivalent circuits C_0 and C_1 of similar size, the obfuscations of C_0 and C_1 should be computationally indistinguishable"

- Garg et al. http://eprint.iacr.org/2013/451.pdf



In case...

Circuit

A circuit is a triple (M, L, G), where

- M is a set of values,
- L is a set of gate labels, each of which is a function from M^i to M for some non-negative integer i (where i represents the number of inputs to the gate), and
- G is a labelled directed acyclic graph with labels from L. The vertices of the graph are called gates. For each gate g of in-degree i, the gate g can be labeled by an element ℓ of L if and only if ℓ is defined on M^i.

https://eprint.iacr.org/2013/451.pdf

Constructed an algorithm which provides indistinguishability obfuscation for all polynomial-size, log-depth circuits.[a]

[a] Relying on an assumption about algebraic hardness...

Catches...

- The (naively) obfuscated versions of simple codes are...

Catches...

- The (naively) obfuscated versions of simple codes are... **HUGE!**

Catches...

- The (naively) obfuscated versions of simple codes are... **HUGE!**
- Relies on the assumption that a certain decision problem about randomly garbled branching decision processes is as hard as it currently seems...

Catches...

- The (naively) obfuscated versions of simple codes are... **HUGE!**
- Relies on the assumption that a certain decision problem about randomly garbled branching decision processes is as hard as it currently seems...*very likely.*

Catches...

- The (naively) obfuscated versions of simple codes are... **HUGE!**
- Relies on the assumption that a certain decision problem about randomly garbled branching decision processes is as hard as it currently seems...*very likely.*
- Has been shown that this concept of 'indistinguishability obfuscation" is the *best* possible form of obfuscation possible.

Catches...

- The (naively) obfuscated versions of simple codes are... **HUGE!**
- Relies on the assumption that a certain decision problem about randomly garbled branching decision processes is as hard as it currently seems...*very likely.*
- Has been shown that this concept of 'indistinguishability obfuscation" is the *best* possible form of obfuscation possible.

END

- On universal obfuscation: https://www.iacr.org/archive/crypto2001/21390001.pdf
- on indistinguishability obfuscation: http://eprint.iacr.org/2013/451.pdf