Introduction to Obfuscation. Black-box Security

Yury Lifshits
Steklov Institute of Mathematics, St.Petersburg, Russia

Introduction to Obfuscation

Tartu University
13/03/2006

Outline

1. Applications of Obfuscation
   - Classification of Threats
   - Applications in Software Protection
   - Applications in Mobile Agents
   - Applications in Cryptography
   - More Applications

2. Blackbox Secure Obfuscation
   - Defining Security of Obfuscation
   - Impossibility Result

Applications in Software Protection

Situation: company distribute (sell) software products.

Question: Threats and applications you see?
- Integrity protection
  - Against decomposition and reusing code fragments
  - Against adding new functionalities
  - Against changing the order of computation
- Protection of internal constraints on:
  - Usage time
  - Input data
  - Availability of customization
  - Quality of performed tasks
  - Number of runs
- Watermarks protection
  - Deleting watermarks in obfuscated program is much harder

Protection of IF Operator

Consider a program containing the following construction:

```
If (some condition) then
  do something important
else do nothing (or some not interesting things)
```

Adversary attack: destroy this IF operator i.e. get a program with unconditional important module.

Impossibility Result

Defining Security of Obfuscation

Outline

Different Types of Attacks

How can adversary act with program?
- Study program (extracting knowledge)
- Decompose program (reusing code/algorithms of it)
- Change program behavior (making illegal modifications)

More attacks?
Mobile Agents Technology

Situation: author distribute programs for his own needs.

Question: Threats and applications you see?
- Privacy of data in mobile agents
- Illegal agent modification
- Sending hard computational task to untrusted cluster
- Auxiliary computing devices for smart cards
- Network monitoring system
- Keys protection
- Buying agents

Buying Agent

Another important example is buying agent.

What do we have: a set of “sellers” with installed buying agents. These agents have a task to purchase a specific good if some conditions (usually on price) holds

Aspects:
- Buying agents have keys to the credit card or electronic money.
- Adversary is always able to delete an agent.
- Agents owner wants to prevent key’s extraction and changing conditions of purchase or even buying wrong good.

Network Monitoring Systems

First interesting example of mobile agent needed protection is network monitoring and management systems.

We have: a huge network consisting of nodes, and a monitoring agent installed on each node.

Some observations:
- Agents interacts with their hosts
- Agents interacts with central (the only trusted) node. We call it control center.
- We can’t protect agents against just deleting (uninstalling them)
- We want to protect the “state” of agents and their proper execution

Applications in Cryptography

What applications in cryptography can we imagine?
- Private key cryptosystem → Public key cryptosystem
  - It was mentioned even in famous Diffie-Hellman paper
- Constructing homomorphic encryption schemes
- Realizing random oracles in cryptosystems

New Public-Key Cryptosystems

General idea: given a private-key (symmetric) cryptosystem publish obfuscated encryption algorithm $O(E_k)$ as a public key.

Analysis:
- We must be sure that key extraction of $O(E_k)$ is computationally hard
- Moreover, rewriting $O(E_k)$ to any efficient program computing $D_k$ must be computationally hard

Conclusion: starting symmetric cryptosystem should have sufficient difference in encrypting and decrypting algorithms

Constructing Homomorphic Encryption

Given good enough obfuscator it’s easy to construct a homomorphic encryption.

Question: Any ideas how to do this?

Construction: as such homomorphic encryption we can take just any public key cryptosystem:

```
Input: E(x), E(y)
Program algorithm: using private key decrypt x and y, compute x + y (respectively xy), then encrypt it.
Output: E(x + y) (respectively, E(xy))
```

If we are able to obfuscate $P$ and $Q$ in the way that extracting private key and intermediate results (x and y) is computationally hard than we are done!

More Applications

- Diversity producing (every user receive his own version)
  Makes virus attacks harder
- Guaranteed slowdown of encrypting procedure in cryptosystems
  Makes brute-force attacks harder
- Digital Rights Management software
  Protection against extracting secret keys from players for copyrighted media files

Question: Your ideas of applications?
Security Definitions in Cryptography (1)

- Define adversary’s inputs
- Define adversary’s goal
- Security = achieving goal is computationally hard

**Proof instrument:**
Reduction: “If somebody can break this new system then he also able to solve some well-known hard problem”

**Example:** security of pseudorandom generators

---

**Ana and BAna**

We are interested in 2 types of polynomial-time analyzers:

- **Ana** is a source-code analyzer that can read the program.
  \[ \text{Ana}(P) \]

- **BAna** is a black-box analyzer that only queries the program as an oracle.
  \[ \text{BAna}^P(\text{time}(P)) \]

**Black-Box security**

**Ana** can’t get more information than **BAna** could

---

**Unobfuscatable Function Family**

Family \( \mathcal{H} = \cup \mathcal{H}_k \)

- \( h \in \mathcal{H}_k \) computable in poly\((k)\) time
- \( \exists x : \mathcal{H} \rightarrow \{0,1\} \) such that
  - \( |Pr\{\mathcal{S}^k(x) = \pi(h)\} - 1/2| = \nu(k) \)
  - \( \exists A \) such that for every TM \( M \) computing \( h \), \( A(M) = \pi(h) \)

---

**Counterexample**

Cannibalistic construction:

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

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

\[ Z_{\alpha}(x) = 0^x \]

**Intuition:** it is difficult to distinguish pairs \( C_{\alpha,\beta}, D_{\alpha,\beta} \) from pair \( Z_{\alpha} \), \( D_{\alpha,\beta} \) given only black box access to these programs.

---

Security Definitions in Cryptography (2)

- Define ideal model
- Security = adversary cannot compute more than in ideal model

**Proof instrument:**
Simulation: “For any property that could be extracted from the new system almost the same property can be extracted from the ideal model”

**Example:** security of zero-knowledge proofs

---

**Black-box Security**

Randomized algorithm \( O \) is an Obfuscator if three following conditions hold:

- (functionality) \( \forall \text{TM } M : O(M) \approx M \)
- (effectiveness) \( \exists \rho : M(x) \text{ terminates in } t \text{ steps } \Rightarrow O(M)(x) \text{ terminates in } \rho(t) \text{ steps.} \)
- (black-box security) For every PPT \( A \) there exists PPT \( S \) such that for all TMs \( M \):
  \[ |Pr\{A(O(M)) = 1\} \setminus Pr\{S^M(1^{|M|}) = 1\}| = \nu(|M|). \]

---

**Unobfuscatable 2-Functions Family**

Family \( \mathcal{G} = \cup \mathcal{G}_k \)

\( \mathcal{G}_k \) is a set (distribution) of pairs of functions \( B^n \rightarrow B^m \)

- \( (g_1, g_2) \in \mathcal{G}_k \) computable in poly\((k)\) time
- \( \exists \pi : \mathcal{G} \rightarrow \{0,1\} \) such that
  - \( |Pr\{\mathcal{S}^k(\pi) = \pi(g_1, g_2)\} - 1/2| = \nu(k) \)
  - \( \exists A \) such that for every TMs \( M_1, M_2 \) computing \( g_1, g_2 \),
    \( A(M_1, M_2) = \pi(g_1, g_2) \)

Existence of unobfuscatable function families and 2-function families. What follows from what?

---

**Technical Details**

We leave out technical details:

- Truncated version of \( D \)
- Combining pair of functions into a single one.
Extensions of Impossibility Result

More impossibilities of obfuscation:
- Unobfuscatable functional properties (not only predicates)
- Computationally easy but still unobfuscatable programs (in $\text{TC}_0$ class)
- Attack (deobfuscation algorithm) is known in advance
- Obfuscator might preserve functionality only approximately
- Impossibility of obfuscation for sampling algorithms

Summary

Main points:
- Rough idea of applications: cryptosystem design, mobile agents technology, software protection.
- Black-box security: obfuscated program tells no more than input-output behaviour.
- There exists unobfuscatable function families

Reading List

- B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S., Vadhan, K. Yang
  On the (Im)possibility of Obfuscating Programs, 2001.
- K. Yang
  Talk on “(Im)possibility of Obfuscating”, 2001.
- N. Kuzyurin, N. Varnovsky and V. Zakharov
- Yu. Lifshits
  Lecture Notes on Program Obfuscation, 2005.
  http://logic.pdmi.ras.ru/~yura/obfuscation.html

Thanks for attention. Questions?