Program Obfuscation and Related Topics
Applications and Perspectives

Yury Lifshits

yura@logic.pdmi.ras.ru

Faculty of Mathematics and Mechanics
Saint-Petersburg State University

Spring 2005 – Intel
Outline

Basics of Obfuscation
Outline

Basics of Obfuscation

Perspective Directions
Outline

Basics of Obfuscation

Perspective Directions

State of the Art
Talk Objectives

⇒ Short overview of applications and results
Talk Objectives

 gönderilő Short overview of applications and results
 gönderilő Search for topic with common interest
Talk Objectives

- Short overview of applications and results
- Search for topic with common interest
- Search for new problems and ideas
Talk Objectives

- Short overview of applications and results
- Search for topic with common interest
- Search for new problems and ideas
- And ask for your intuition about the topic
Main Concept

So, what is Obfuscator?
So, what is Obfuscator?

"clear" \[\rightarrow\] Obfuscator \[\rightarrow\] "unreadable"
So, what is **Obfuscator**?

- Functionality preserving
- Increase of code size, time & space requirements are restricted (usually by constant factor)
- Obfuscated program is **not readable** (not understandable)
Some facts:

- First mention — famous Diffie-Hellman paper (1976)
- More than 30 publications, several Ph.D. thesises
- More than 25 Java obfuscators
- International Contests (C, Perl, PostScript, Ruby)
- Famous universities involved (Weizmann, Stanford, Princeton, MSU)
- Famous companies involved (Sun, Microsoft)
General Source-to-Source Obfuscators

Observations:

- Long list of tricks (layout, data, control flow)
- Commercial potential
- No guaranteed security
- Static analysis of obfuscated program is computationally hard
- Arms race against hackers
General Source-to-Source Obfuscators

Observations:
- Long list of tricks (layout, data, control flow)
- Commercial potential
- No guaranteed security
- Static analysis of obfuscated program is computationally hard
- Arms race against hackers

Conclusion:
Obfuscators are necessary (in some cases) even without guaranteed security
Low-level Obfuscators

Observation:
Disassembling and decompilation tools are not perfect

Low-level obfuscation:
⇒ Making exact disassembling hard
⇒ Making exact decompilation hard

Same story — arms race with adversary:
Low-level Obfuscators

Observation:
Disassembling and decompilation tools are not perfect

Low-level obfuscation:
- Making exact disassembling hard
- Making exact decompilation hard

Same story — arms race with adversary:

New protection $\Rightarrow$ new analysis $\Rightarrow$ new protection $\ldots$
Low-level Obfuscators

**Observation:**
Disassembling and decompilation tools are not perfect

Low-level obfuscation:
- Making exact disassembling hard
- Making exact decompilation hard

Same story — arms race with adversary:

New protection $\Rightarrow$ new analysis $\Rightarrow$ new protection ...

**Conclusion:**
Future obfuscators will combine source-to-source and low-level tricks
Hardware-based program protection

Good recent news:

⇒ Some promising solutions are already presented (XOM, 2004)
⇒ Model: memory is accessible to adversary, processor is not
⇒ To achieve the best level of security program should be obfuscated in special way
⇒ Security analysis is not ready yet
Hardware-based program protection

Good recent news:

⇒ Some promising solutions are already presented (XOM, 2004)
⇒ Model: memory is accessible to adversary, processor is not
⇒ To achieve the best level of security program should be obfuscated in special way
⇒ Security analysis is not ready yet

Conclusion:
There is a potential for hardware and/or interpretation level of software protection
RTL-model Obfuscation

New threat: bookmark insertion during chip manufacturing

Solution: chip obfuscation

Most appropriate level for obfuscation usage
[Zakharov, 2005] — RTL model of chip
RTL-model Obfuscation

New threat: bookmark insertion during chip manufacturing

Solution: chip obfuscation

Most appropriate level for obfuscation usage
[Zakharov, 2005] — RTL model of chip

**Conclusion:**
Obfuscation could be done but effectiveness is not studied yet
Specific Protection

What type of attacks are we going to resist?
Specific Protection

What type of attacks are we going to resist?

- Key’s extraction
- Modification:
  - Add
  - Delete
  - Edit
  - Reuse
- Vulnerability search
- Bookmarks insertion
- Program state attack
More Applications

Other applications?
More Applications

Other applications?

- Mobile agents protection
- White Box Encoding and DRM applications
- Digital watermarks
Current Achievements

Most significant results to the moment:

- A lot of obfuscators. Static analysis is now really hard
- Definition of “ideal” security
- Parameter hiding based on classical cryptography
- Hardware solutions (in theory?)
- Huge list of tricks/ideas without security proof
Our Contribution

What have our SPRINT Lab group already done?
Our Contribution

What have our SPRINT Lab group already done?

 DISCLAIMS

⇒ Theoretical models of:
  • Program Slowdown
  • Secure Function Sharing
  • Fully Encrypted Computation
  • Condition-protection

⇒ Hardware methods survey

⇒ Low-level obfuscation survey (+ some original tricks)
Theoretic View

Main questions for obfuscation theory:
- Find all obfuscatable programs?
- List of modelling examples which require obfuscation (benchmarks)?
- Protection against specific attacks?
- Hardware/interpretation protection of programs?
- Quality of obfuscation?
- Power of deobfuscation (program understanding)?
What Do We Learn Today?

- Obfuscating transformations should make programs harder to understand, analyse and modify
What Do We Learn Today?

- Obfuscating transformations should make programs harder to understand, analyse and modify.
- There is a long list of threats based on program understanding.
What Do We Learn Today?

- Obfuscating transformations should make programs harder to understand, analyse and modify.
- There is a long list of threats based on program understanding.
- Hope for new protection methods.
What Do We Learn Today?

- Obfuscating transformations should make programs harder to understand, analyse and modify
- There is a long list of threats based on program understanding
- Hope for new protection methods

Thanks for your attention! Questions?

- New problems?
- New ideas?
- Critique?
For Further Reading

Yury Lifshits.
*Lecture Notes on Program Obfuscation*
http://cs-seminar.spb.ru/，“Reports” section

Yury Lifshits
*Program Obfuscation. A survey*