Different Concepts for Program Obfuscation

Yury Lifshits

Mathematics & Mechanics Faculty
Saint Petersburg State University

Spring 2005 – SETLab

Outline

Applications for Obfuscation

Today: only short overview of applications

In details: Lecture 4 - “Applications for Obfuscation”

Classical Cryptography

What applications in cryptography can we imagine?

- ⇨ Private key cryptosystem → Public key cryptosystem
  It was mentioned even in famous Diffie-Hellman paper.
- ⇨ Homomorphic encoding
- ⇨ Random oracles removing

Software Protection

Situation: we distribute (sell) software products.

Question: Threats and applications you see?

- Competitors threat (reusing your code)
- Intelligent tampering (changing parameters)
- Threat of functionality changes (protection demo-versions)
- Watermarks protection

Concepts of Obfuscation
Yury Lifshits
Applications
Classical Cryptography
Software Protection
Mobile Agents Technology
Other
Main Approaches
Obfuscating Transformations
Blackbox Security
Mobile Cryptography
Aspects of Model
Program Representation
Attacks and Environment
Summary

Mobile Agents Technology

Situation: we distribute programs for our needs.

Question: Threats and applications you see?

- Privacy: e.g. internet-distributed computation
- Keys protection: buying agents.
- Intelligent tampering

Concepts of Obfuscation
Yury Lifshits
Applications
Classical Cryptography
Software Protection
Mobile Agents Technology
Other
Main Approaches
Obfuscating Transformations
Blackbox Security
Mobile Cryptography
Aspects of Model
Program Representation
Attacks and Environment
Summary

Other applications

Question: More applications?

Yes!

- Virus development
- Watermark attacks

Concepts of Obfuscation
Yury Lifshits
Applications
Classical Cryptography
Software Protection
Mobile Agents Technology
Other
Main Approaches
Obfuscating Transformations
Blackbox Security
Mobile Cryptography
Aspects of Model
Program Representation
Attacks and Environment
Summary

An Obfuscator

In details: Lecture 2 - “Obfuscating transformations”

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

Classification of obfuscating transformations

What can we obfuscate in the program?

- Layout transformations
  Change formatting information
- Control flow transformations
  Alter control program and computation
- Aggregation transformation
  Refactor program using aggregation methods
- Data transformations
  Use information encoding

Concepts of Obfuscation
Yury Lifshits
Applications
Classical Cryptography
Software Protection
Mobile Agents Technology
Other
Main Approaches
Obfuscating Transformations
Blackbox Security
Mobile Cryptography
Aspects of Model
Program Representation
Attacks and Environment
Summary

Quality of Obfuscation

How good our obfuscation is?

Strength can be measured by:

- Potency
  $\frac{E(P')}{E(P)} - 1$
- Resilience
  Trivial, weak, strong, full, one-way
- Cost
  Free, cheap, costly, expensive
- Stealthy

Concepts of Obfuscation
Yury Lifshits
Applications
Classical Cryptography
Software Protection
Mobile Agents Technology
Other
Main Approaches
Obfuscating Transformations
Blackbox Security
Mobile Cryptography
Aspects of Model
Program Representation
Attacks and Environment
Summary

Ideal case

What do we want to get?

Very limited information:

- input-output behavior
- running time

Concepts of Obfuscation
Yury Lifshits
Applications
Classical Cryptography
Software Protection
Mobile Agents Technology
Other
Main Approaches
Obfuscating Transformations
Blackbox Security
Mobile Cryptography
Aspects of Model
Program Representation
Attacks and Environment
Summary

Ana and BAna

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

- Ana is a source-code analyzer that can read the program.
  $$Ana(P)$$
- BAna is a black-box analyzer that only queries the program as an oracle.
  $$BAna^P(time(P))$$

Black-Box security

Ana can't get more information than BAna could

Property Hiding

How to formalize property hiding?

Instance: two families of programs Π_1 and Π_2 Adversary task: given a program $P \in \Pi_1 \cup \Pi_2$ to decide whether $P \in \Pi_1$ or $P \in \Pi_2$.

Desirable protection: make adversary task as difficult as well-known computationally hard problem is.

Constant Hiding

How to formalize constant hiding?

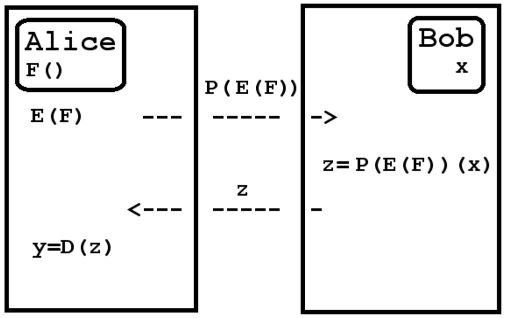
Instance: family of programs $\Pi = \{P \mid P \text{ computes } f(s,x);\ s \in S\}$ Adversary task: given a program $P \in \Pi$ to compute parameter s.

Desirable protection: make adversary task as difficult as well-known computationally hard problem is.

Encrypted Computation

More details: Lecture 5 - “Basic Complexity Results”

What is encrypted computation?



Basic task: keep F unknown to Bob.

Extendings of Encrypted Computation

Additional tasks of encrypted computation model:

- Move difficult computations to Bob
  D is easier than F
- Reduce communication complexity
  In the case $\mathit{sizeof}((F(x)) \ll \mathit{sizeof}(x)$. Example: x is database
- Keep x secret from Alice

Currently studied representations

Obfuscating techniques development depends on used program representation

So what sort of programs are we going to protect?

- Turing Machines / Circuits (function computing)
- C++/Java code
- Assembler code
- Rational function / Matrix representation

Concepts of Obfuscation
Yury Lifshits
Applications: Classical Cryptography, Software Protection, Mobile Agents Technology, Other
Main Approaches: Obfuscating Transformations, Blackbox Security, Mobile Cryptography
Aspects of Model: Program Representation, Attacks and Environment
Summary

Search for other representations

Is it enough?

Not! New models should contain:

- Current state of the program.
- Self-modifiable code
- Notion of computation trace.
- Other formalizations for functionality preserving.

Concepts of Obfuscation
Yury Lifshits
Applications: Classical Cryptography, Software Protection, Mobile Agents Technology, Other
Main Approaches: Obfuscating Transformations, Blackbox Security, Mobile Cryptography
Aspects of Model: Program Representation, Attacks and Environment
Summary

Adversary

What should we specify about adversary?

- Adversary knowledge about protected program
  - Member of family
  - Known function – unknown parameters (data) and state.
- Adversary task (attack)
  - Classification follows in Lecture 4.

Concepts of Obfuscation
Yury Lifshits
Applications: Classical Cryptography, Software Protection, Mobile Agents Technology, Other
Main Approaches: Obfuscating Transformations, Blackbox Security, Mobile Cryptography
Aspects of Model: Program Representation, Attacks and Environment
Summary

Potential for Obfuscation

Is it possible to protect every program?

- How to measure potential of obfuscation?
  - Learnability: black-box learnable functions are impossible to obfuscate.
- What couldn't be protected?
  - Input-Outbut behaviour
  - Traces

Network model

What are interesting network extentions of the model?

- Many programs cooperate
- Programs are migrating
- Programs can be recharged
- Different sources for inputs (outside connections)

Concepts of Obfuscation
Yury Lifshits
Applications
Classical Cryptography
Software Protection
Mobile Agents Technology
Other
Main Approaches
Obfuscating Transformations
Blackbox Security
Mobile Cryptography
Aspects of Model
Program Representation
Attacks and Environment
Summary

Summary

- Rough idea of applications: cryptosystem design, mobile agents technology, software protection.
- Basic approaches: obfuscating transformations, black-box security, encrypted computation.
- Further aspects of the model: program representation, state protection, adversary description, functionality preserving.

Question Time!

Not covered by the talk

Gray & white security
Approximate obfuscators
Operations on obfuscated code
Adversary success
Nondeterministic nature
Modifying algorithm vs. modifying code
Complexity of deobfuscation: NP, NP-hard, undecidable, one-way...
Obfuscation on specification level
Wroblewsky model

Concepts of Obfuscation
Yury Lifshits
Back Up Slides
Not covered by the talk
References

For Further Reading

Yury Lifshits
Program Obfuscation. A Survey [in Russian]
http://logic.pdmi.ras.ru/~yura/of/survey1.pdf

Luis F.G. Sarmenta
Protecting Programs from Hostile Environments
http://bayanihancomputing.net/papers/ae/ae.ps