Security and Scalability of Remote Entrusting Protection

Vasily Desnitsky, Igor Kotenko

Laboratory of Computer Security Problems,
St. Petersburg Institute for Informatics and Automation of Russian Academy of Sciences

Agenda

- Security & Scalability
- SW Protection methods and Remote Entrusting principles
- Problem Statement
- Performance and Security evaluation
- Technique of performance evaluation
- Technique of security evaluation
- Empirical study

Security & Scalability

- SW protection against tampering
- Remote Entrusting Protection
  - Client/Trusted Server protection scheme
  - Variety of Tamper Resistance Protection Methods embedded into the mechanism
  - Remote entrusting protection principles
- Security vs. Scalability problem
  - Protection mechanism implementation in practice
  - Minimizing of Trusted Server side computations
- Complexity of protection methods
- The aim is to build a combined SW protection mechanism, achieving
  - Some reasonable tradeoff between security quality and scalability

Protection methods in use

- Tamper resistance SW protection methods
  - Barrier Slicing
  - Barrier Slicing with tamper resistant hardware
  - Continuous Replacement
  - Orthogonal Replacement
  - Control Flow Checking
  - Invariant Checking
  - Hardware assisted invariants monitoring
  - TPM based Remote Attestation
  - SW Monitor performing Checksums on a program
  - Etc.

Remote entrusting protection principles

- Client-Server protection scheme
- Remote attestation (RA)
  - Checking procedures on Client
  - Verification on Trusted Server
- Dynamic replacement (DR)
  - Replaceable SW component construction
  - SW components installation and enforcement
- Slicing (S)
  - TS side execution of a part of a protection method code

Protection methods in use

- Tamper resistance SW protection methods
  - Barrier Slicing – *(S)*
  - Barrier Slicing with tamper resistant hardware – *(S)*
  - Continuous Replacement – *(DR)*
  - Orthogonal Replacement – *(DR)*
  - Control Flow Checking – *(RA)*
  - Invariant Checking – *(RA)*
  - Hardware assisted invariants monitoring – *(RA)*
  - TPM based Remote Attestation – *(RA)*
  - SW Monitor performing Checksums on a program – *(RA)*
  - Etc.

Problem Statement

The aim is
*to find a set **S** of protection methods that*

$$\begin{cases} \text{minimize } \left| \sum_{i \text{ from S}} p(m_i) \right| \\ \text{maximize } \sum_{i \text{ from S}} s(m_i) \end{cases}$$

To be reduced to an extreme problem:

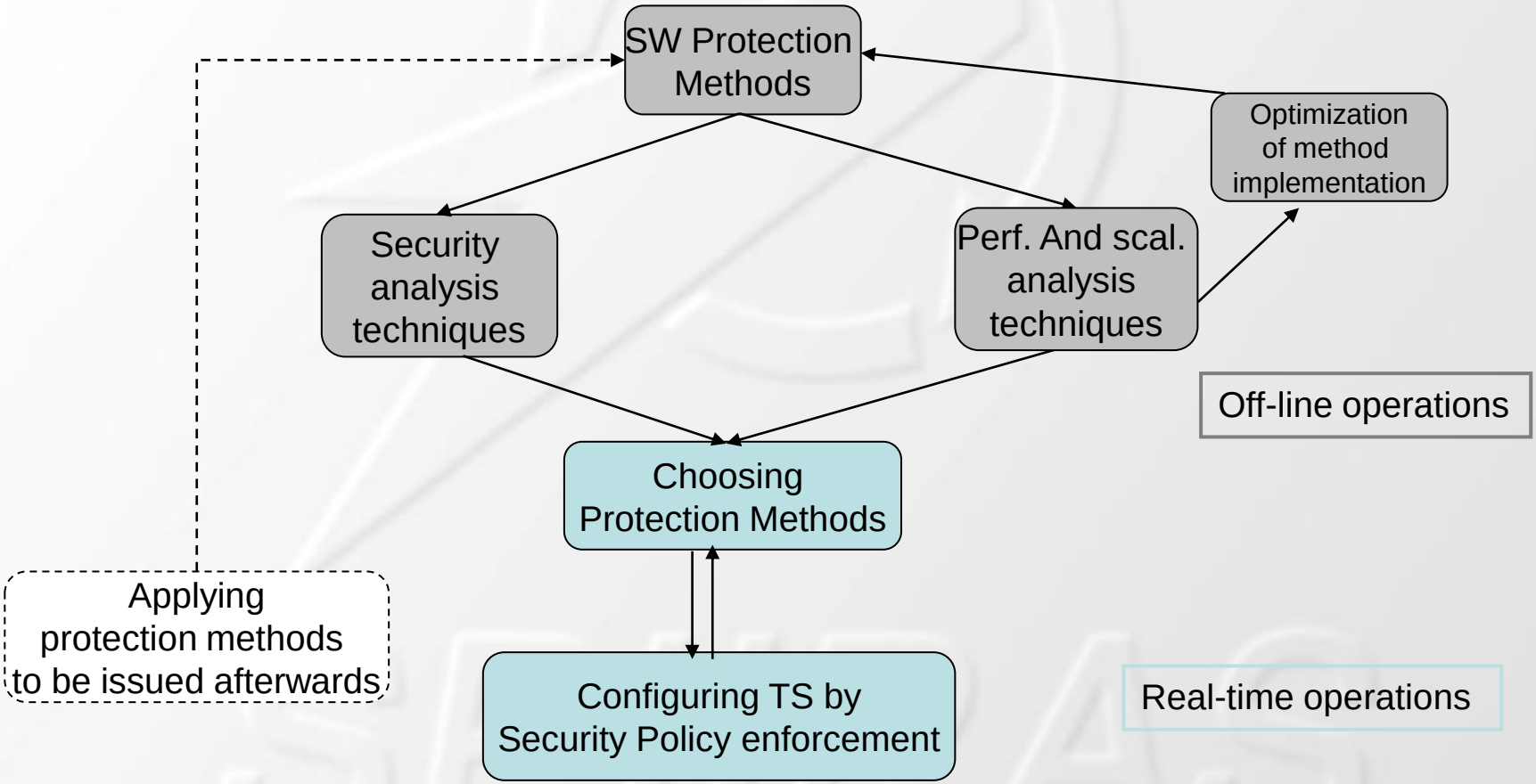
$$\begin{cases} \text{minimize } \left| \sum_{i \text{ from S}} p(m_i) \right| \\ \sum_{i \text{ from S}} s(m_i) \geq Const \end{cases}$$

The extreme problems are solved
on a basis of classical discrete *knapsack problem / exhaustive search*

Performance and Security Evaluation

- Determining metrics for SW protection
- Evaluation of
  - resources consumed by an each protection method
    - $p(m_i) = <p1,p2,\ldots>$ - vector-valued function giving a bundle of metrics for a method m_i
  - security level of each method
    - $s(m_i)$ - specific relative value characterizing strength of protection methods
- Specifying and choosing optimal combination of protection methods depending on volumes of security and resource consumption metrics

Protection mechanism workflow



Technique of performance evaluation (1/2)

- Modeling the protection methods
- Specifying needed performance metrics
  - Metric realizing
  - Using prepared metrics from performance measuring tools
- Simulation of protection method work
  - Simulation of the work of the server and clients and communication between them
  - Computation of metric values for various protection methods and diversity of their parameters
- Analysis of obtained results
  - Comparison of values for a variety of protection methods and/or total values for protection method combinations

Technique of performance evaluation (2/2)

- Specified performance metrics
  - **Workload** – time gap required to accomplish a single unit of the protection method
  - **Throughput** – quantity of the method copies that can be executed on the server concurrently
  - Server load **intensity** of the protection method - amount of computations fulfilled per a specifically allotted time unit

- Evaluation metrics – combined approach
  - *Theoretically* – modeling the most essential resources consuming operations executed on the TS side
  - Empirically – implement the model and measure required data

Technique of security evaluation

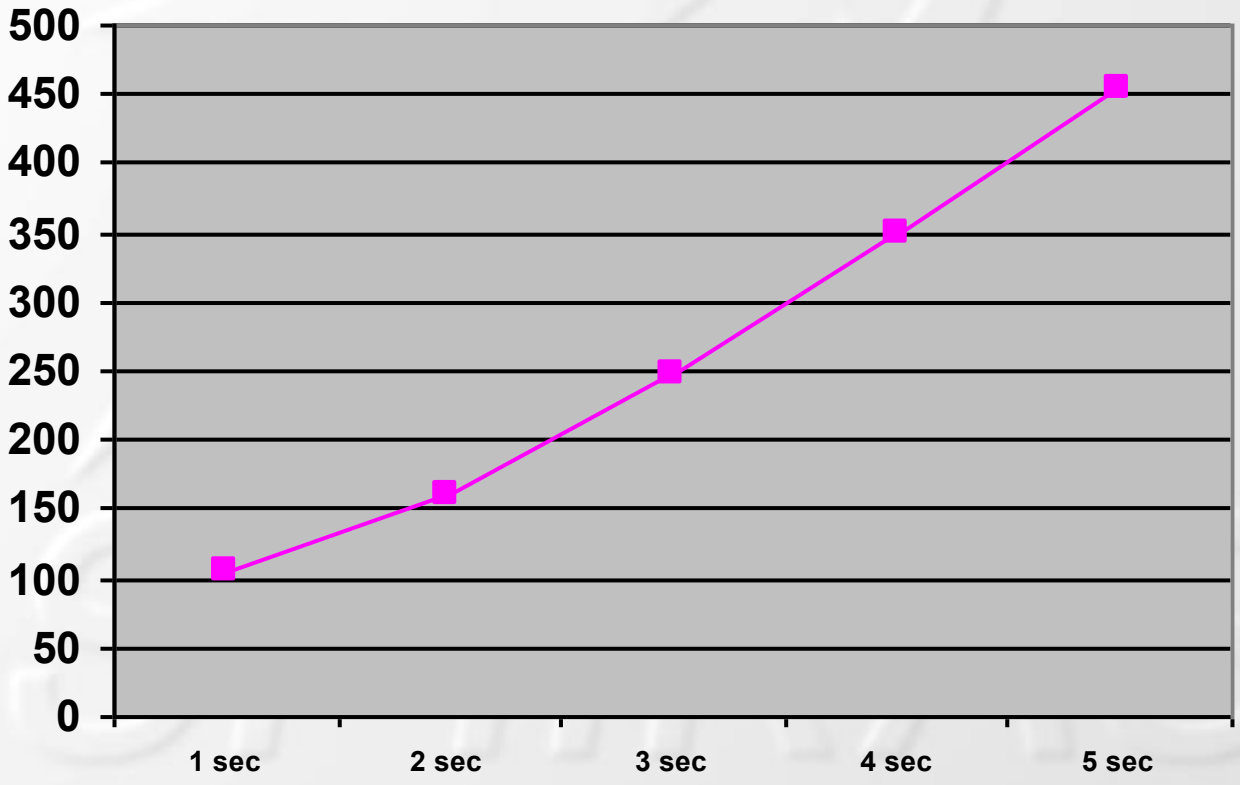
- Main difficulty of security evaluation
  - Essential disparateness and heterogeneity of the protection methods
    - Different object of protection
    - Different theoretic protection principles
- Difficulty of construction of formal evaluation approaches
  - Method strength in many respects is determined reasoning from cognitive abilities of attackers (which may differ drastically for different potential attackers)
  - => such evaluation is very difficult to carry out in a formal way
  - => trying to determine strength of the methods by their **heuristic analysis**
- Security evaluation by heuristic analysis
  - Protection mechanism developer determines strength of all the methods starting from his/her own experience and intuition
- **Expert judgment** approach as an extension of the latter one
  - Surveying a number of security experts
  - Computation of averaged values by expert judgments processing

Empirical study – Performance evaluation

- Modeling of Control Flow Checking method
  - Implementation of the basic operations essential for performance evaluating on TS
  - A test program containing several functions of its business logic to be protected was implemented
  - Limitations – merely correctness of the sequence of beginnings and endings of the functions is checked
- Simulation of Control Flow Checking method
  - Machine *A* simulates the work of Trusted Server side of the protection method
  - Machine *B* simulates the work of a number of clients communicating with the server
  - Measuring values of specified performance metrics
- Modeling and simulation IC and BS methods

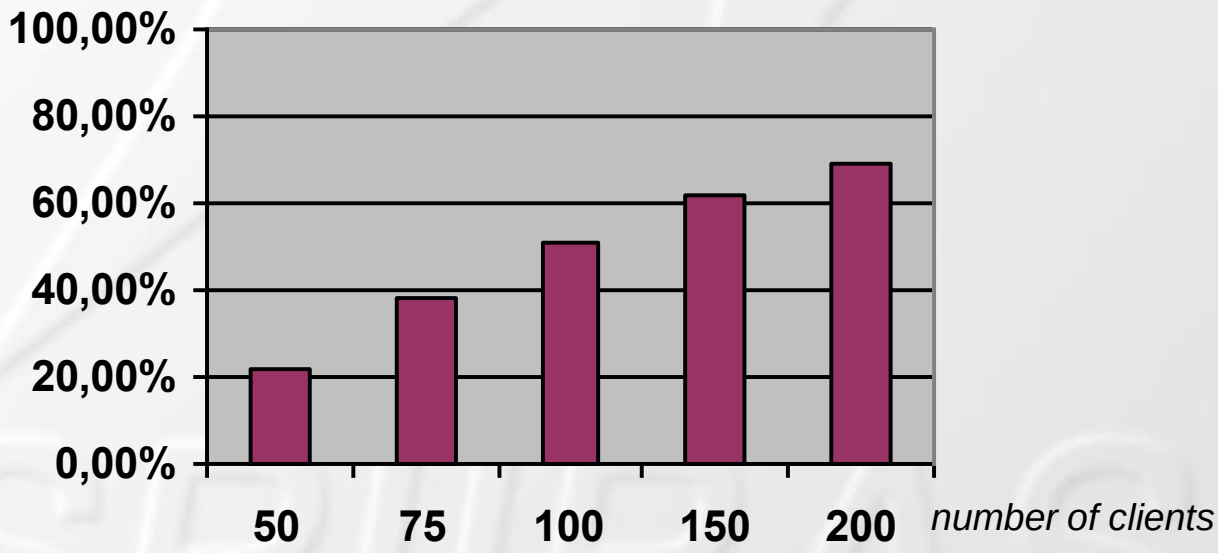
Performance evaluation – experiment results (1/2)

- Dependency between *time* allotted for tag checking on the TS and maximum *amount of clients* carrying out Control Flow Checking model



Performance evaluation – experiment results (2/2)

- Server load intensity for Control Flow Checking model
  - Dependency between server load and client amount

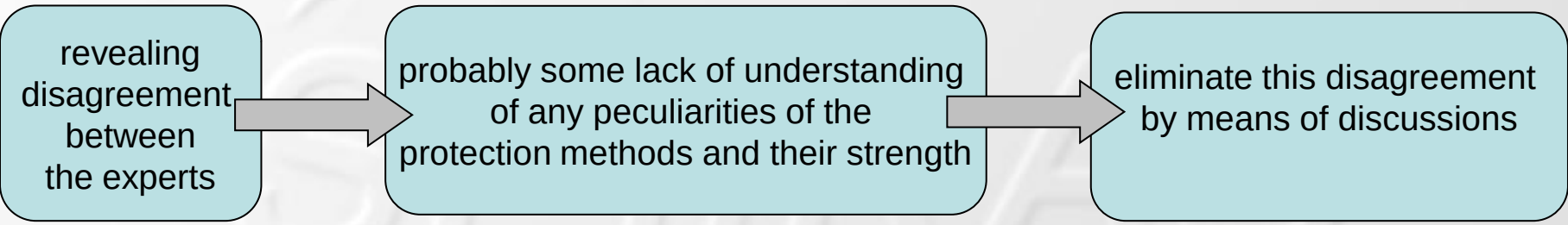


Security Evaluation

- On a basis of expert judgments
  - 10 experts of computer security field
- Survey task
  - For each protection method
    - Giving weight (*from 1 to 10*)
    - With taking into account the method falling into categories
      - Methods with/without *code splitting*, *replacement quality*, *execution on server*
- Competence
  - *A priori* competence determined by each expert him/herself
  - *A posteriori* competence determined by a degree of consistency of the individual expert estimations with the expert group estimation
- Computation using recursive formulas of expert judgment processing

Expert judgment based evaluation technique (1/2)

- Drawbacks and advantages
  - (-) it represents relatively rough solution for the evaluation of protection methods
  - (-) it can not be exploited as a proof of adequacy of the whole protection mechanism
  - (+) it can be regarded as a supplement to security evaluation methods based on formal approaches having their own drawbacks
  - (+) it enables the following scenario:



Expert judgment based evaluation technique (2/2)

- Experiment results
  - Obtained security values for the SW protection methods

TR method	Security level
Barrier Slicing	9,0
Orthogonal Replacement	7,8
Continuous Replacement	7,1
Crypto Guards	6,2
Control Flow Checking	4,3
Invariant Checking	3,3
Obfuscation technique: opaque predicates	1,3

10 grade scale

Conclusion

- As a future activities
  - Searching and construction more precise evaluation approaches for security and performance of the Remote Entrusting protection
  - Conducting and accomplishment empirical studies