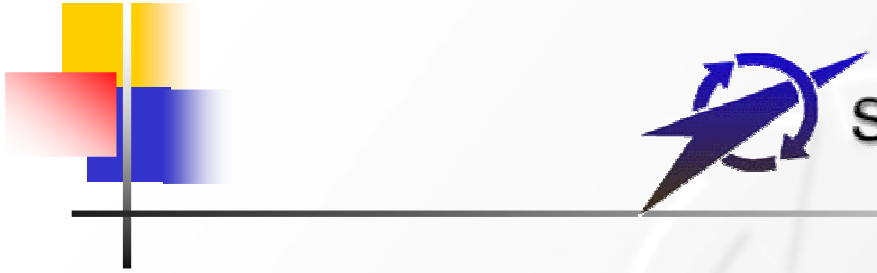
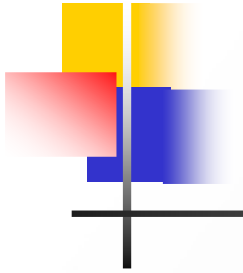


Performance and Scalability of Remote Entrusting Protection

Vasily Desnitsky, Igor Kotenko

Computer Security Research Group,
St. Petersburg Institute for Informatics and Automation of Russian Academy of Sciences



Agenda

- Performance & Scalability
- Problem Statement
- Optimizing protection method implementation
- Security Policies for configuring the protection mechanism
- Technique of performance evaluation
- Technique of security evaluation
- Empirical study

Performance & Scalability

- Remote Entrusting Protection
  - Variety of Tamper Resistance Protection Methods embedded into the mechanism
- Performance & Scalability
- Protection mechanism implementation in practice
- Minimizing of Trusted Server side computations
- Complexity of protection methods
- Some reasonable tradeoff achieving
  - Security quality vs. Scalability

Problem Statement

General task
*to find a set **S** of protection methods that*

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

Reduced to two extreme problems:

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

$$\begin{cases} \text{maximize } \sum_{i \text{ from S}} s(m_i) \\ \sum_{i \text{ from S}} p(m_i) \leq Const \end{cases}$$

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

Performance and Security Evaluation

- Evaluation of
  - resources consumed by each protection method on the TS side
    - $p(m_i) = \langle p1, p2, \ldots \rangle$ - 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 volume of available resources

Protection methods

- RE-TRUST solutions classified as *Remote* ones
  - Barrier Slicing
  - Barrier Slicing with tamper resistant hardware
  - Continuous Replacement
  - Orthogonal Replacement
  - Secure interlocking and authenticity checking
  - Control Flow Checking
  - Invariant Checking
  - Hardware assisted invariants monitoring
  - Remote Attestation with TPM
  - Monitor that performs Checksums on a program
  - …

Protection method analysis

- Trusted Server function to hold the protection method could contain
  - Verification procedure
  - TS side execution of the protection method
  - Replaceable SW component construction
- TS performance mainly depends on
  - Activities on TS which computation complexity grows proportionally the amount of clients

Performance of the protection mechanism

- Heightening performance of the protection mechanism
  - optimizing protection method implementation
  - optimizing configuration of the protection methods by choosing some other combination of these methods

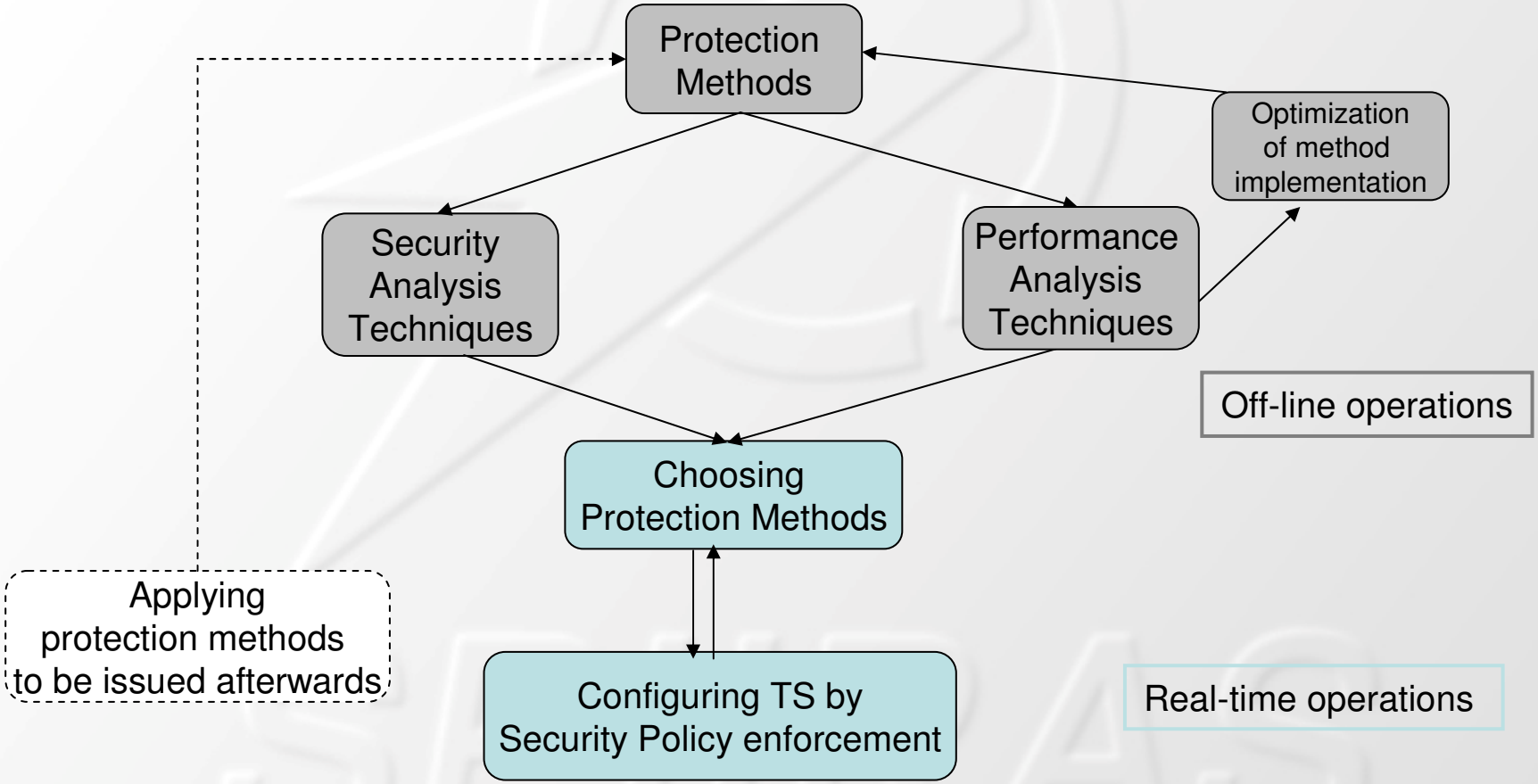
Optimizing protection method implementation

- For each protection method to learn
  - if some activities of the method could be fulfilled *in advance* (before the client programs start)
    - The positive answer would mean the required actions on the trusted entity could be accomplished off-line (*e.g. on the deployment phase of the mechanism*). Therefore these actions will not influence essentially upon the overall performance during client running
    - E.g. methods without replacement: *BS, CFC, IC*
  - If the activities of the method could be carried out for *multiple* clients *at once* (or for some groups of them at least)
    - The positive one would mean the performance of these actions don't depend on the client amount
    - E.g. activities that don't depend on individual client state: *OR*, the methods without remote attestation

Security Policies for configuring the protection mechanism

- Specify policies for a case when TS can't support all the clients correctly because of a lack of available resources when the current method configuration is optimal
- Depending on the target application character to determine *risks* of the program's holder
  - loosing a legal client
    - let some group of clients to work with a reduced protection for a while
  - a malicious client managed to tamper with the program being not revealed
    - discard some clients
- Choosing these client groups
  - according to reputation of separate clients
  - some external data, e.g. user mandates; complete/trial/demo client and other gradations, etc.
  - delegate the process of decision taking from the protection mechanism to the target application

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

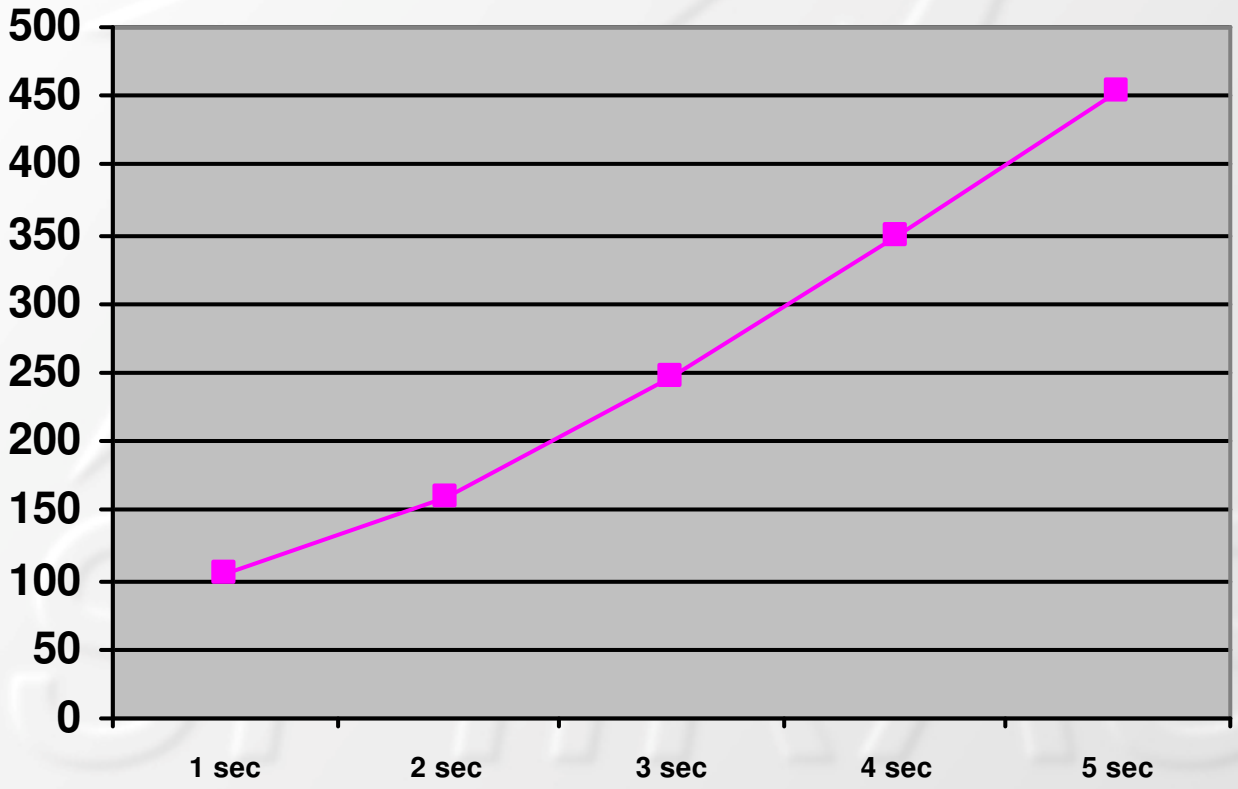
- Main difficulty of security evaluation
  - Essential disparateness and heterogeneity of the protection methods
    - Different object of protection
    - Different theoretic protection principles
- Difficulty of construction of formal evaluation approaches
  - Method strength in many respects is determined 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
  - => we can try 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

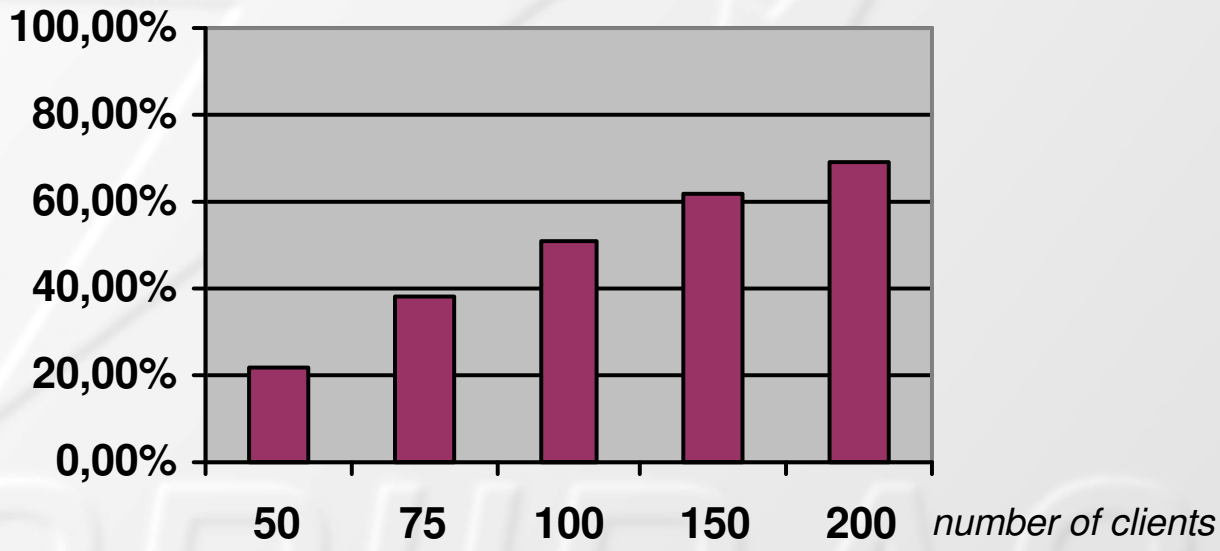
Performance evaluation – experiment results (1/2)

- 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

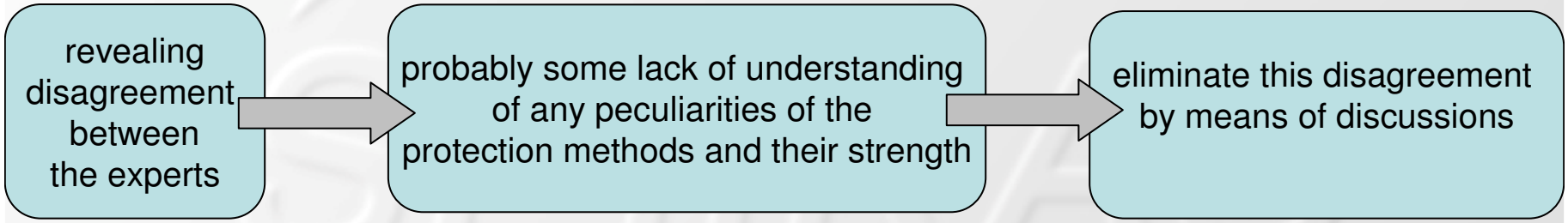


Security Evaluation

- On a basis of expert judgments
  - 9 experts within RE-TRUST community
- Survey task
  - For each protection method
    - Giving weight (*from 1 to 5*)
    - Ranking all the methods
    - 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 known recursive formulas of expert judgment processing

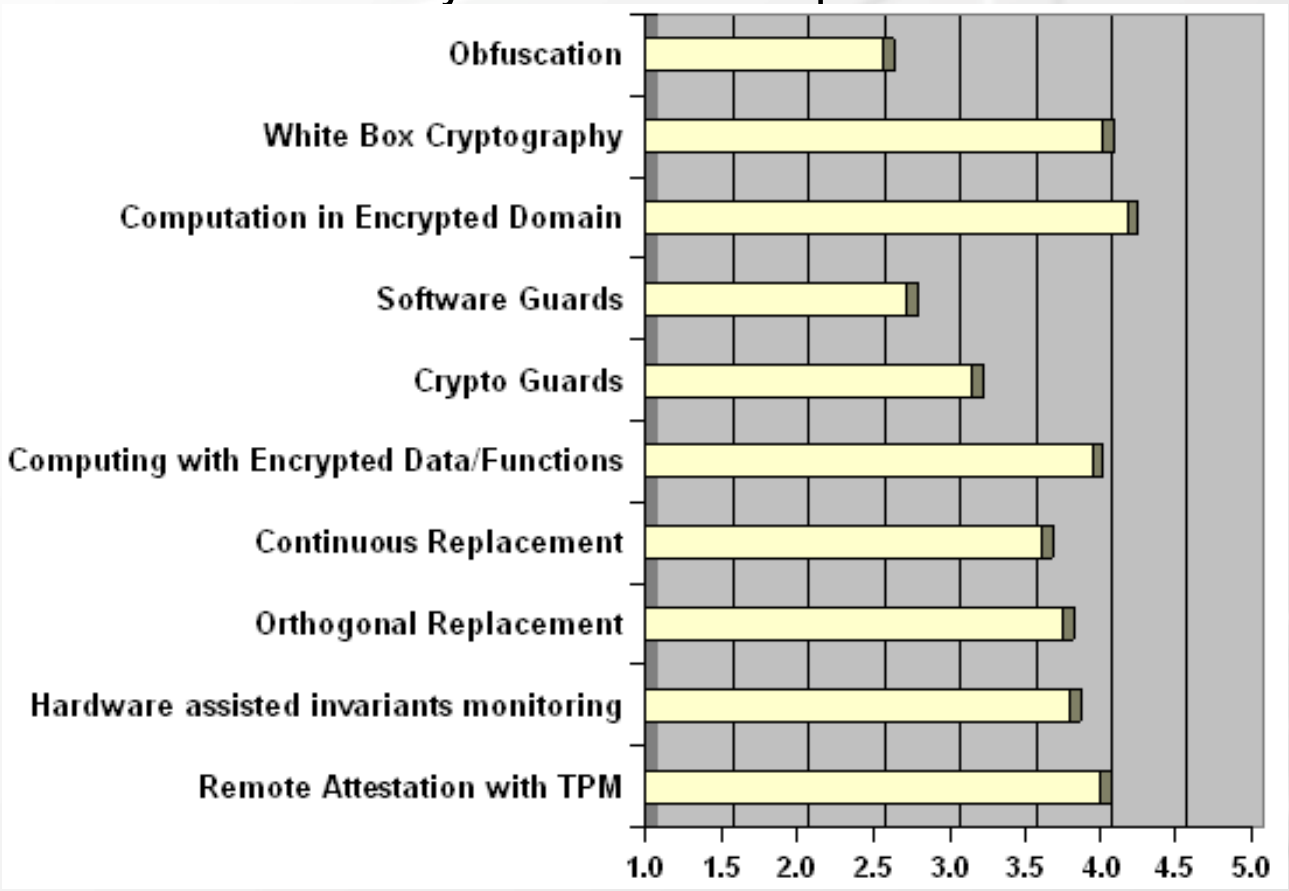
Expert judgment based evaluation technique (1/2)

- 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 protection methods



Conclusion

- As a future activities
  - Searching and construction more precise evaluation approaches for the Remote Entrusting based protection