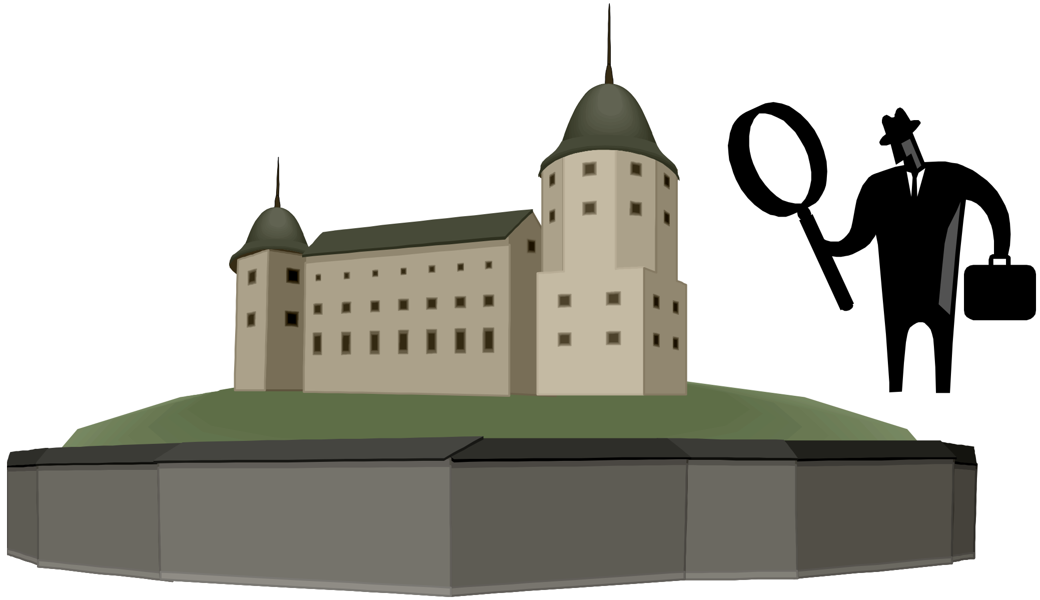


Towards Analysis of Various Protection Techniques

Brecht Wyseur, KULeuven
December 15th 2008, Paris

Overview

- Software protection techniques
  - Crypto guards
  - Obfuscation techniques
  - Fuzzing (analysis technique)
  - White-Box …

- Hardware assisted software protection techniques
  - Remote attestation with a TPM on a legacy OS
  - Physically Observable Cryptography (POC)

WP2

WP3

WP2 techniques

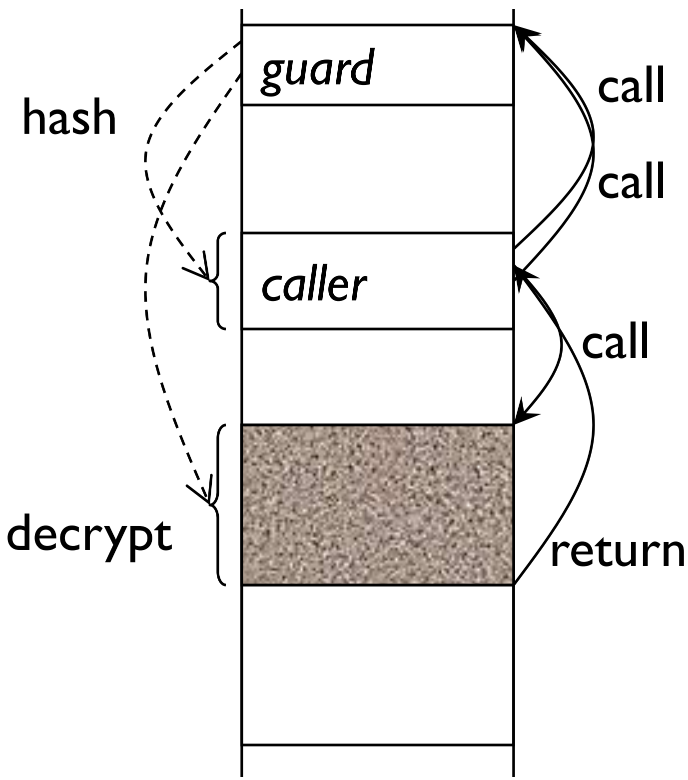
Crypto guards

Obfuscation techniques

White-box …

Crypto guards

- Construction [1]



- Goal: protect software implementations against analysis and tampering.
- On demand encryption
- A *crypto guard* is a small piece of code, that dynamically decrypts code with a key derived from other code bytes
- Idea: deploy a large network of nested code guards to make life of an adversary hard.

- [1] J. Cappaert, B. Preneel, B. Anckaert, M. Madou, and K. De Bosschere, "Towards Tamper Resistant Code Encryption: Practice and Experience", 2008

Crypto guards -- analysis

- Implementation on SPEC CPU2006 test suite
- Experiments to measure cost in execution time
  - 1. Bulk encryption
  - 2. On demand encryption

Program	Total func	On demand	Speed cost	# guard
Mcf	22	20	1.09	28
Milc	159	146	8.17	543
Hmmer	234	184	3.20	873
Lbm	19	12	1.00	20
Sphinx_livepretend	210	192	6.65	1277

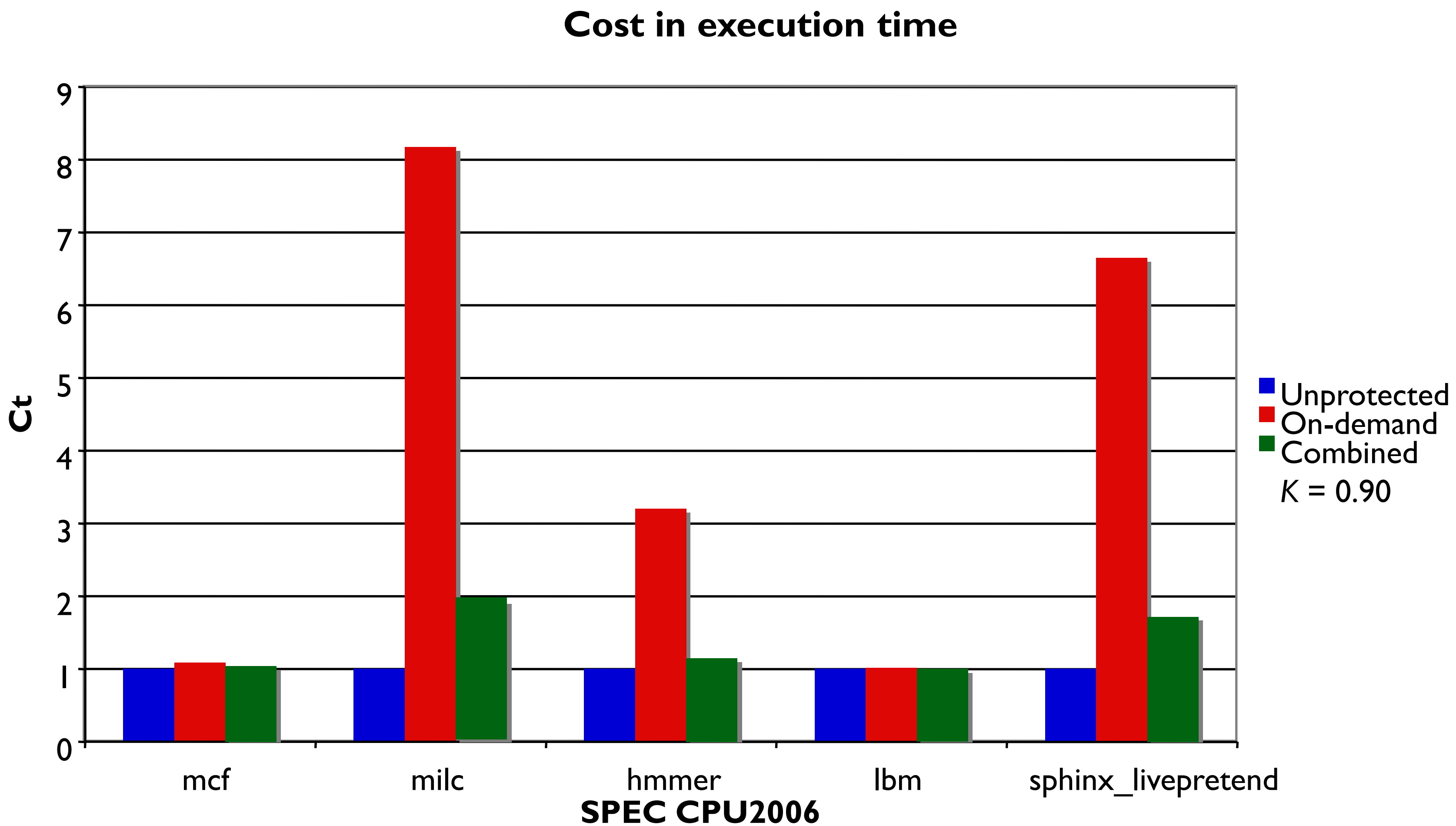
Crypto guards -- analysis

- 3. Performance vs. security trade-off
  - Hot code heuristic
    hot code = code that is frequently called (k% times)
  - Exp. 3: for k = 0.90: bulk encryption for hot code; on demand for remainder.

Program	Total func	On demand	Speed cost	# guard
Mcf	22	19	1.04	24
Milc	159	135	1.95	486
Hmmer	234	183	1.15	862
Lbm	19	8	1.00	17
Sphinx_livepretend	210	181	1.72	1257

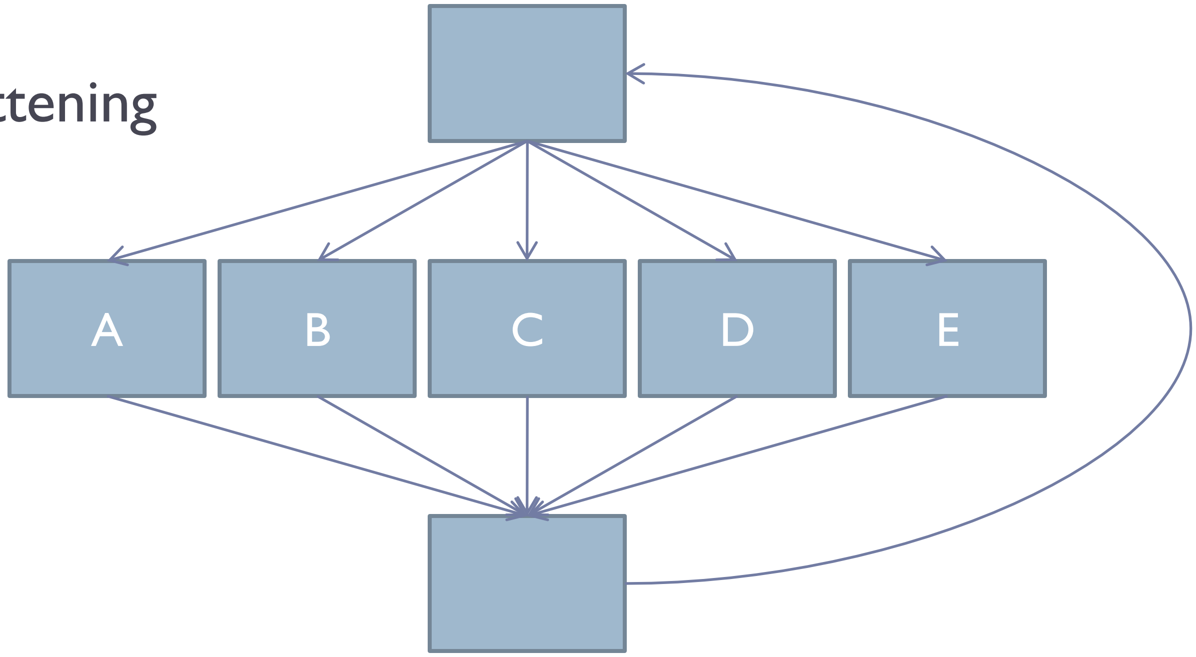
Crypto guards -- analysis

- Experiment results



Obfuscation techniques

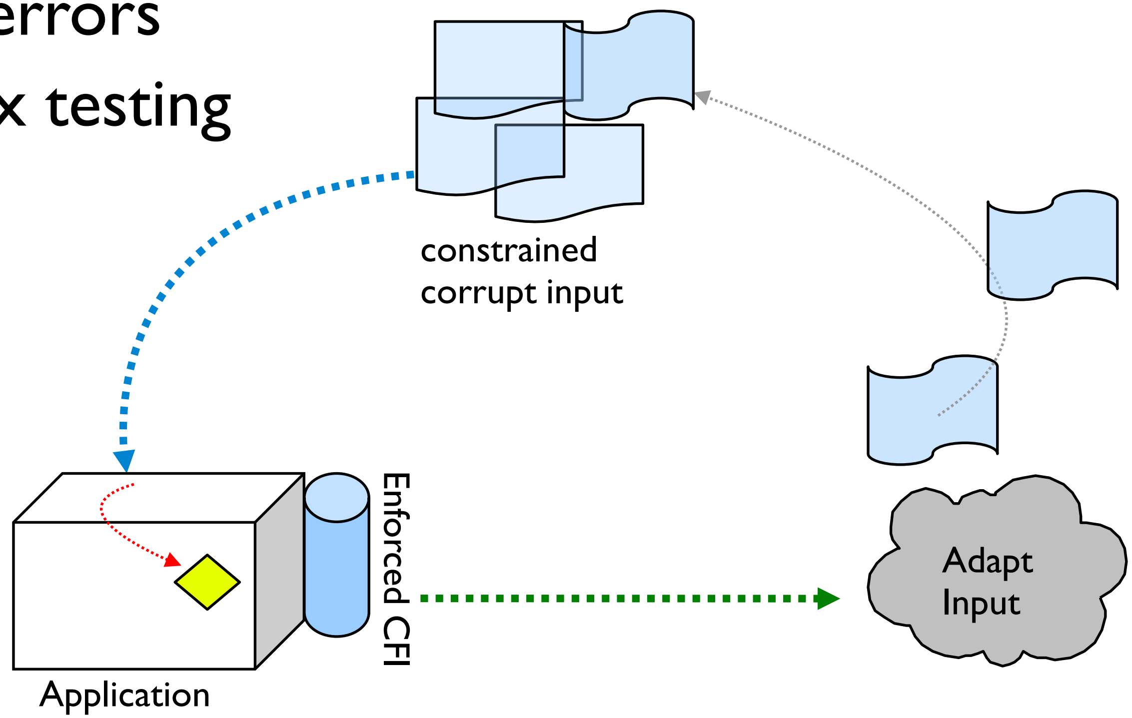
- Goal: make static and dynamic analysis difficult
- Techniques
  - Control Flow Flattening



  - Opaque predicates

Fuzzing [2]

- Submit random/unexpected data to an application, and monitor resulting errors
- Adaptive white-box testing technique



- In initial phase.
- Seems suitable to assess "invariants monitoring" techniques (invariants = constraints)

- [2] N. Kisserli, B. Preneel, "Surgical fuzzing of open source applications using static analysis", COSIC internal report, 5 pages, 2008

White-Box Cryptography

- Goal: implement cryptographic primitives in such a way that they remain secure in a white-box attack context.

- How to assess the security of WBC?
  - WBC techniques are very custom designed per primitive
  - Assess security of $O(E_k)$
  - $O(E_k)$ is secure => E_k is secure (in black-box context)

- Traditional assessment of security in cryptography
  - Direct proof (information theoretically secure)
  - Proof by reduction (to some hard problem)
  - Ad-hoc security

Security analysis of WBC

Black-Box

- Ad-hoc security
  - Block ciphers
  - Process of scrutinizing
    - Cryptanalysis
    - Design criteria (S-boxes, avalanche effect, diffusion properties, MDS, …)

White-Box

- White-boxed block cipher
- Metrics (diversity, ambiguity)
- Process of scrutinizing
  - Cryptanalysis
  - WB design criteria (differential properties, no MDS, …)

Security analysis of WBC

Black-Box

- Proof by reduction
  - Typical for asymmetric crypto
  - Provably secure based on some problems *believed to be hard.*
  - Define model + attack goals: security notions
- Direct proof

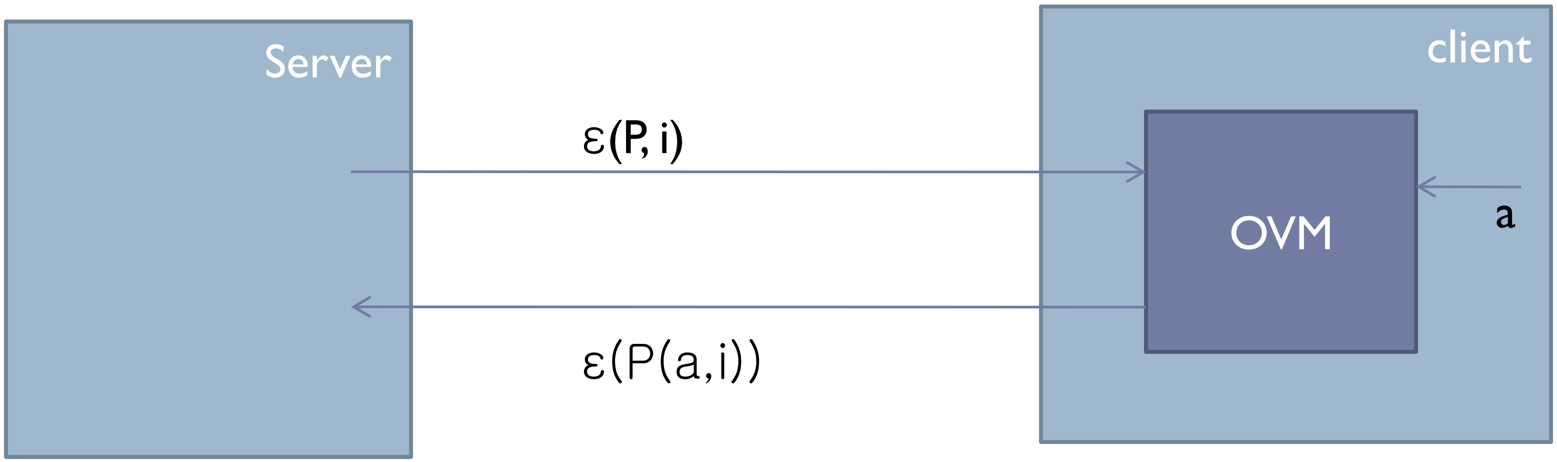
White-Box

- Symmetric ciphers from asymmetric crypto (cheating?)
- Model: def. obfuscation + context [3].
- P = NP?

- [3] A. Saxena, B. Wyseur, "On White-Box Cryptography and Obfuscation", Cryptology ePrint Archive, Report 2008/273, 2008

White-Box Remote Program Execution

- Framework:



- Goals
  - *Obfuscated Virtual Machine (OVM)* able to execute generic programs (note: Barak *et al.*; Goldwasser *et al.* do not apply)
  - program obfuscation as secure as underlying cipher
    E.g.: level of "trust" in integrity of execution: $1 - 2^{-m}$, where
    $\varepsilon : GF(2)^n \rightarrow GF(2)^{n+m}$

WBRPE – security analysis

- Problem: the *Obfuscated Virtual Machine (OVM)* leaks *EVERY* computation (CPU and memory calls)

- How to make a secure OVM?
  - From SFE (as presented at RE-TRUST 2008)
    - Problem: size for *reasonable* circuits.
  - Create a custom secure building block (towards a TM)
    - Then, composing building blocks
    - We are able to construct a secure VM for a narrow set of circuits
    - Generalizing: universally composable cryptography (Canetti 2001)
  - In practice (for now) – augment a VM (e.g., JVM)
    - Deploy obfuscation techniques

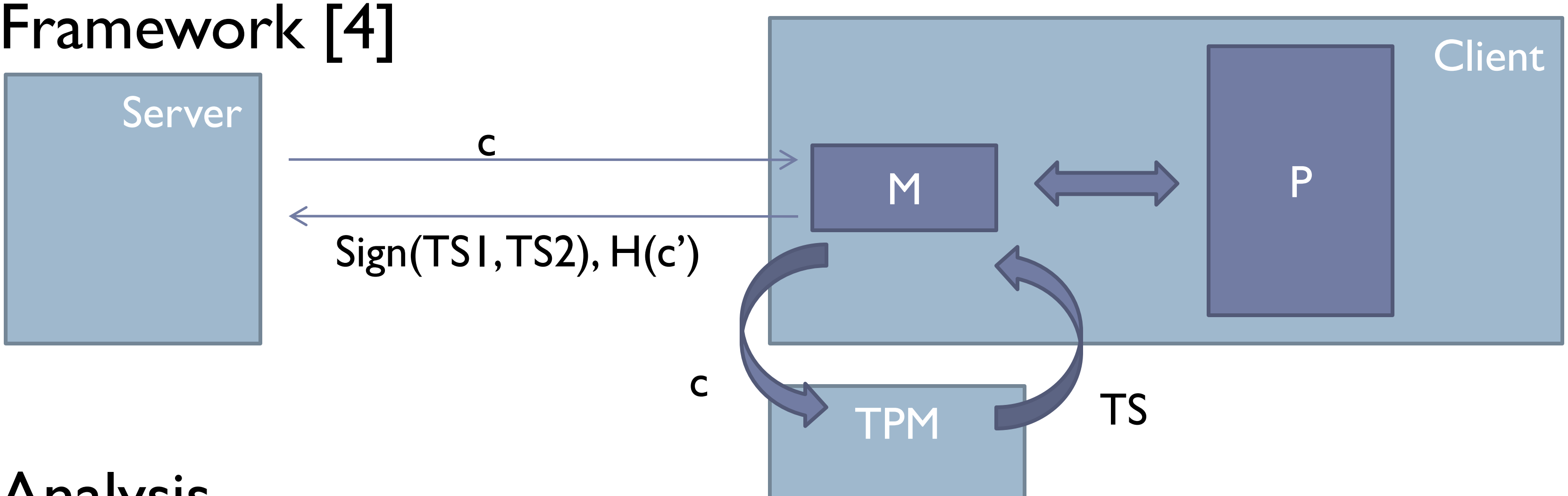
WP3 techniques

Remote attestation

Physically observable cryptography

Remote attestation with a TPM

- Framework [4]



- Analysis
  - Requirements
    - Requirements on PIONEER (unpredictable, optimal checksum function, unpredictable random walk through memory); TEAS (unpredictable, well obfuscated checksum function)
    - Trusted bootloader; trusted clock ticker
  - Efficiency

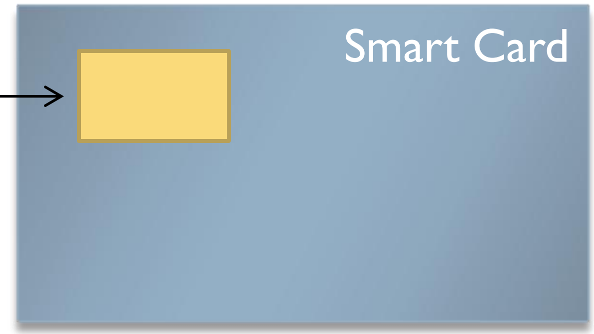
[4] D. Schellekens, B. Wyseur, B. Preneel, "Remote attestation on Legacy Operating Systems with Trusted Platform Modules", REM 2007

Physically Observable Cryptography

- Goal: model a side-channel adversary, and attempt to obtain (provable) security on circuit implementations
- Models
  - Micali & Reyzin
    - Reduction proofs
    - Problem: for each extension, new assumptions required
  - Ishai, Sahai, and Wagner
    - Private circuits I: probing attacks
    - Private circuits II: tamperable circuits
    - Problem: realistic assumptions?
- Future research
  - Improved models
  - New constructions

Physically Observable Cryptography

- Micaly & Reyzin model
  - Secure basic primitive
  - Reduce security of other constructions to security of the basic primitive



- Micali & Reyzin studied basic theoretic constructions
  - (PO) OWF → (PO) PRNG
  - Disadvantage: inefficient, not used in practice
- KUL
  - Study of practical constructions (RSA-CPA; RSA-OAEP; RSA-FDH)
  - Problem: requirements needed for each step
- Future work
  - Change model
  - Develop new schemes (not likely; will face similar problems)

Physically Observable Cryptography

- Ishai-Sahai-Wagner model (*"Private circuits I"*)
  - Boolean circuit implementation
  - Adversary can probe t wires
  - t-security: adversary does not gain any information
- Construction
  - Based on *secret sharing*
  - Result: circuit with $O(nt^2)$ gates
- Problem: controversial model – 'normal' adversaries do not probe, but measure power consumption

Conclusion

- Assess the security of techniques
  - Metrics, empirical studies, fuzzing
    - Obfuscation techniques
  - Scrutinizing
    - White-box implementations of block ciphers
    - Hash functions (remote attestation)
  - Provable security (reduction proofs)
    - White-box implementations from asymmetric primitives
    - White-Box Remote Program Execution (WBRPE)
    - Physically Observable Cryptography (POC)
- Assumptions
  - Trusted TPM, Trusted bootloader
  - Model (obfuscation definition; leakage model)

Conclusions

- Our main question: how well are our defenses against ‘real-world’ adversaries?