Partially Reconfigurable TPM Architectures as the Security Anchors of Future Embedded IT Systems

CASED

S. Malipatlolla, T. Feller, and S. A. Huss

Integrated Circuits and Systems Lab
Technische Universität Darmstadt
and
Center for Advanced Security Research Darmstadt
Germany

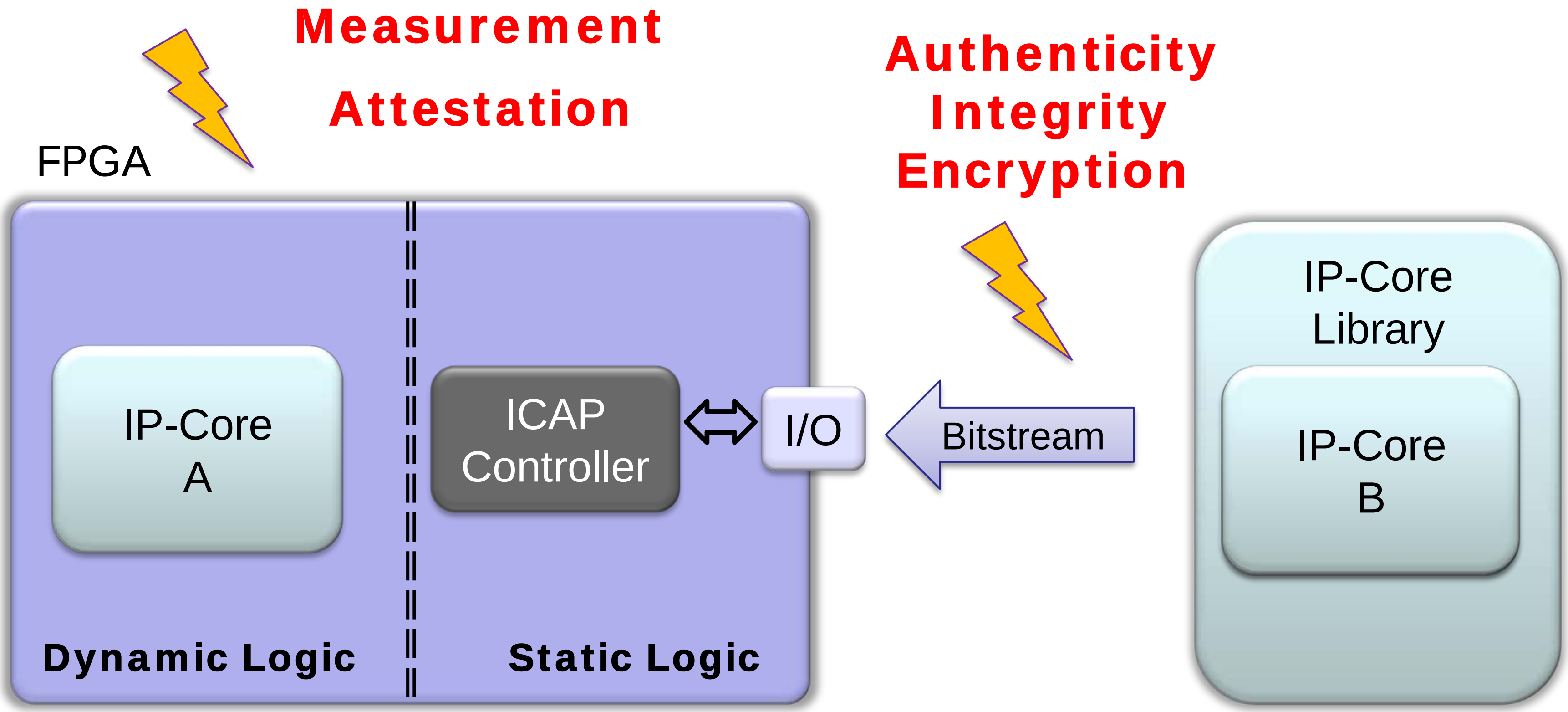
TECHNISCHE UNIVERSITÄT DARMSTADT Fraunhofer SIT h_da HOCHSCHULE DARMSTADT UNIVERSITY OF APPLIED SCIENCES LOEWE – Landes-Offensive zur Entwicklung Wissenschaftlich-ökonomischer Exzellenz

Motivation / Problem Statement

Main Idea:

Trustworthy Operation

IP Protection



Related Work

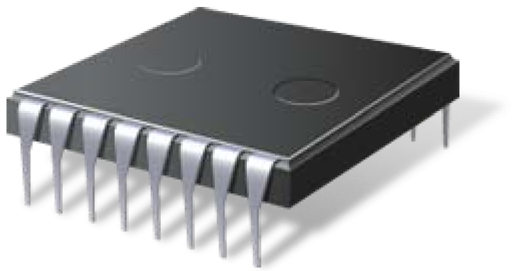
CASED

Xilinx and Altera [3,4]

- Static bitstream encryption provided
- No integrity and authenticity verification of IP

Glas et al. [5,6]

- External TPM (Infineon 1.2)
- Trustworthy reconfigurable embedded system



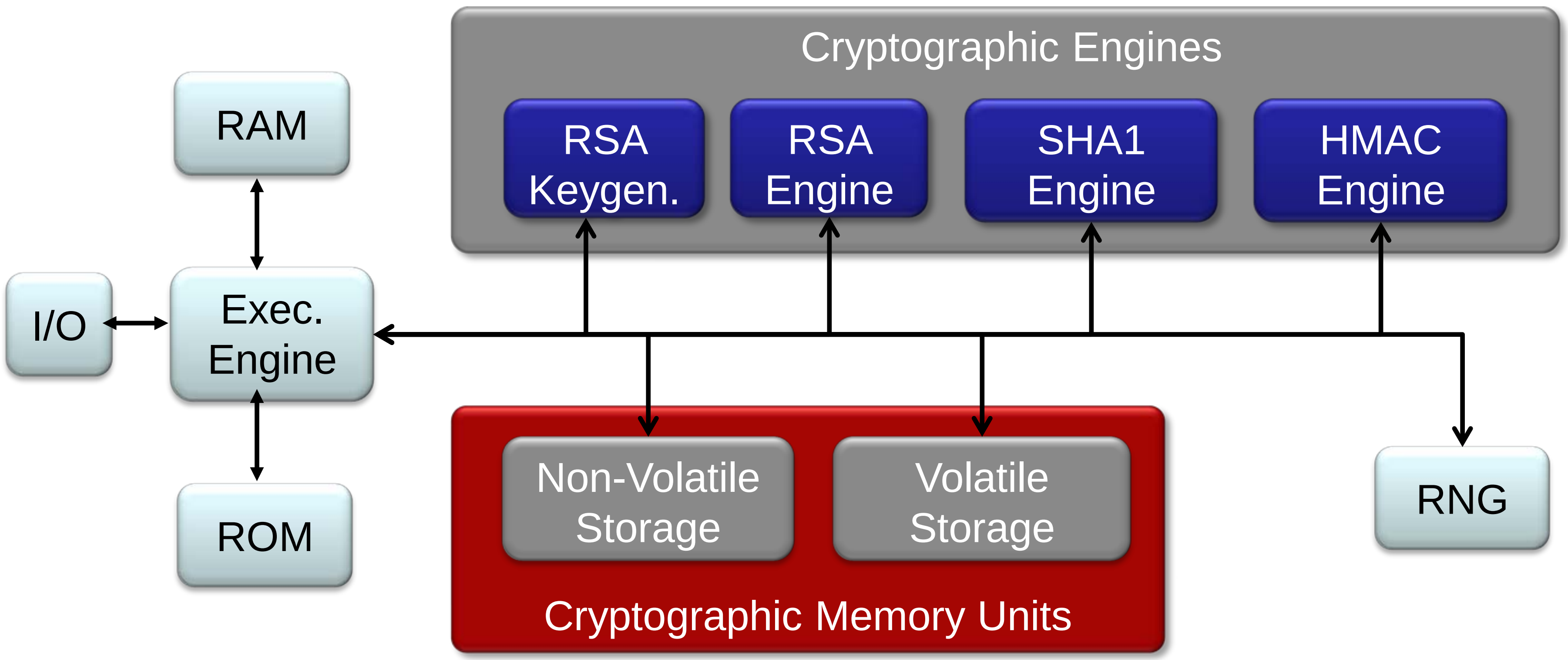
Eisenbarth et al. [7]

- Trusted Computing on FPGA
- TPM as a full bitstream

TECHNISCHE UNIVERSITÄT DARMSTADT Fraunhofer SIT h_da HOCHSCHULE DARMSTADT UNIVERSITY OF APPLIED SCIENCES LOEWE – Landes-Offensive zur Entwicklung Wissenschaftlich-ökonomischer Exzellenz

Conventional TPM Architecture

CASED



Drawbacks of Current TPMs

- Lack of flexibility in current TPM design because of ASIC design style

- Increasing threats on internal TPM engines
  - Collision search attacks on SHA-1 [8]
  - Side-channel attack on RSA key generator [9]

Future requirements:

- NIST recommendation on key lengths [11]

- TPM.next specification by TCG [10]

Main Characteristics of proposed Solution

CASED

Availability of both static and updatable regions

Implementation options:

- Structured-ASICs: Only mask programmable, i. e., not updateable
- FPGAs: In general no non-volatile memory (NVM) available

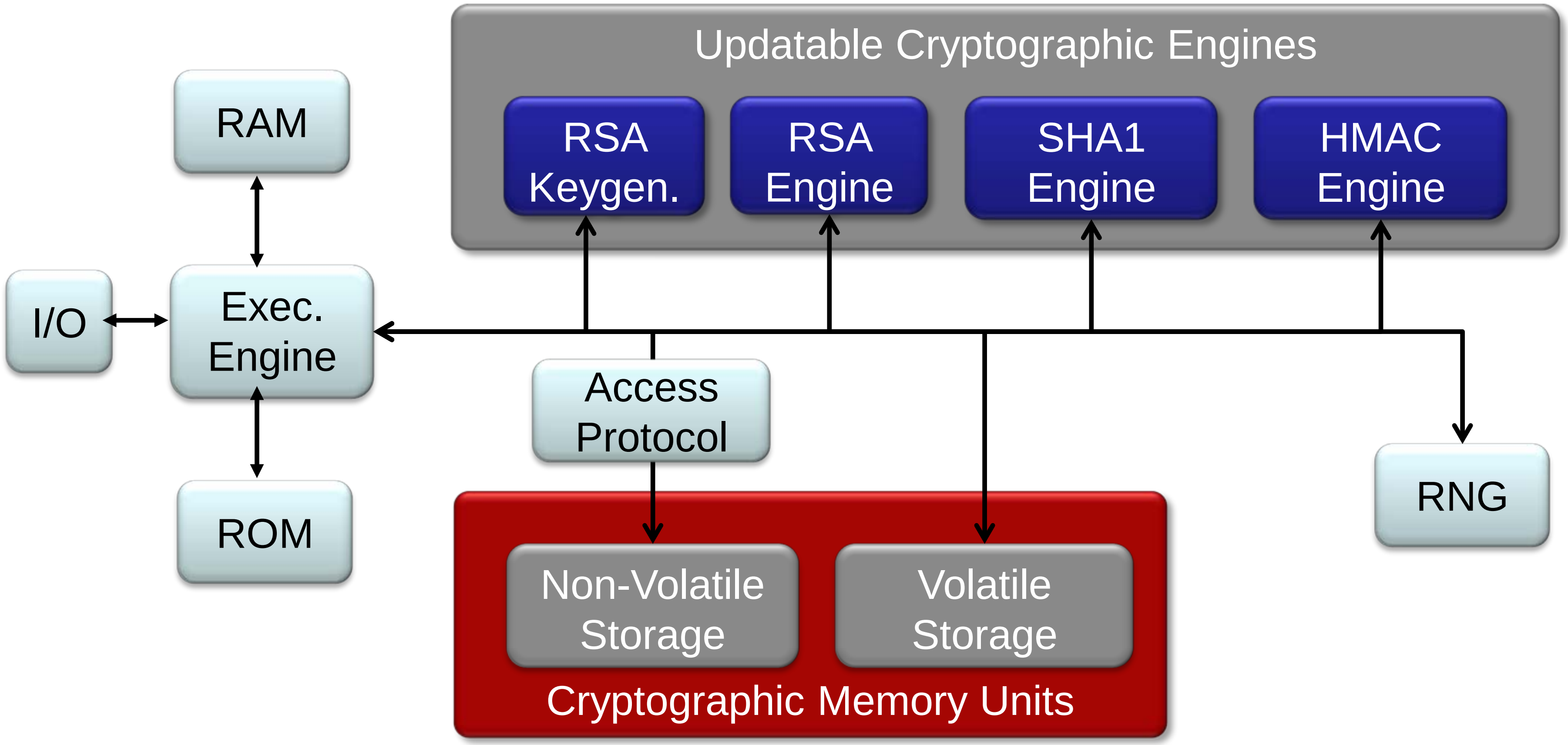
Possible target devices:

- Xilinx Spartan3AN, Microsemi ProASIC3 [13]
  - On-board Flash memory available for NVM implementation, but no support of partial reconfiguration (PR), i. e., not updateable

Proposed solution: Sustainable TPM

- FPGA with PR property as target device
- External NVM with secure access protocol (cf. Schellekens et al. [14])

Novel Architecture: Sustainable TPM



Outline of NVM Access Protocol

Access Protocol: A challenge-response scheme that establishes a secure communication between the external NVM and the STPM

- The protocol uses a MAC algorithm keyed with a shared secret K_Auth.
- K_Auth is present in both the STPM and the external NVM and is derived from an intrinsic physical unclonable function (PUF).
- A PUF is a physical structure inserted into an integrated circuit, which exploits variations in the manufacturing process.
- Dedicated `Read` and `Write` operations for NVM access are required.

Access Protocol Operations

CASED

STPM | **Extenal NVM**

$r \in_R Z_n$ — $r||i$ →

$M_i \leftarrow \text{Read}(i)$

$c \leftarrow \text{Read}(0)$

← $M_i||c||H_{K\_Auth}(r||i||M_i||c)$

Verify MAC

Read

Access Protocol Operations

CASED

STPM | **Extenal NVM**

Read

STPM: $r \in_R Z_n$

STPM → NVM: $r || i$

NVM: $M_i \leftarrow Read(i)$

NVM: $c \leftarrow Read(0)$

NVM → STPM: $M_i || c || H_{K\_Auth} (r || i || M_i || c)$

STPM: Verify MAC

Read

Write

NVM: $c \leftarrow Read(0)$

STPM → NVM: $i || M_i'' || H_{K\_Auth} (i || M_i'' || c)$

NVM: Verify MAC

NVM: $Write(i, M_i'')$

NVM: $Write(0, c+1)$

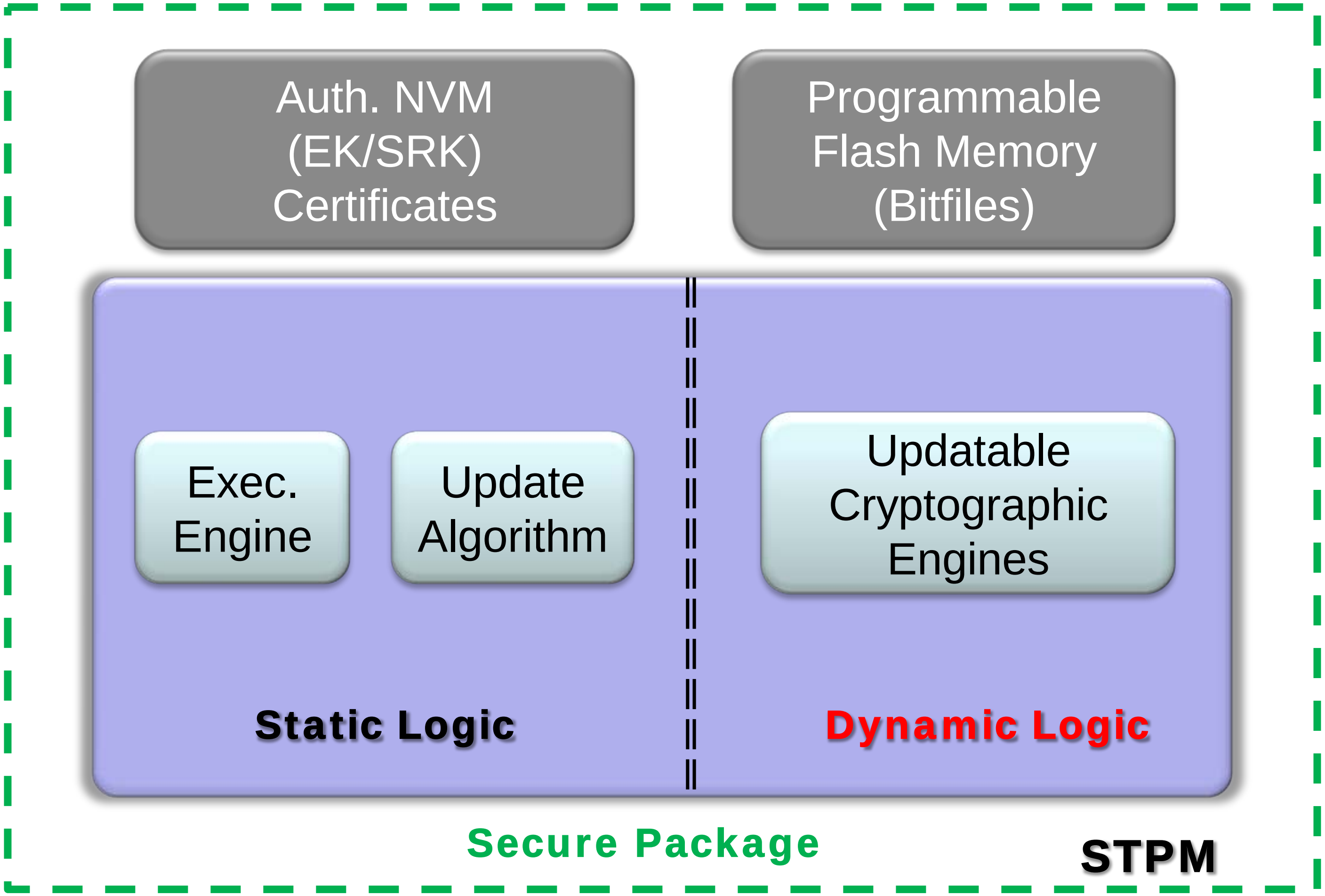
NVM → STPM: $H_{K\_Auth} (c+1)$

STPM: Verify MAC

Write

STPM on Top of partially reconfigurable FPGAs

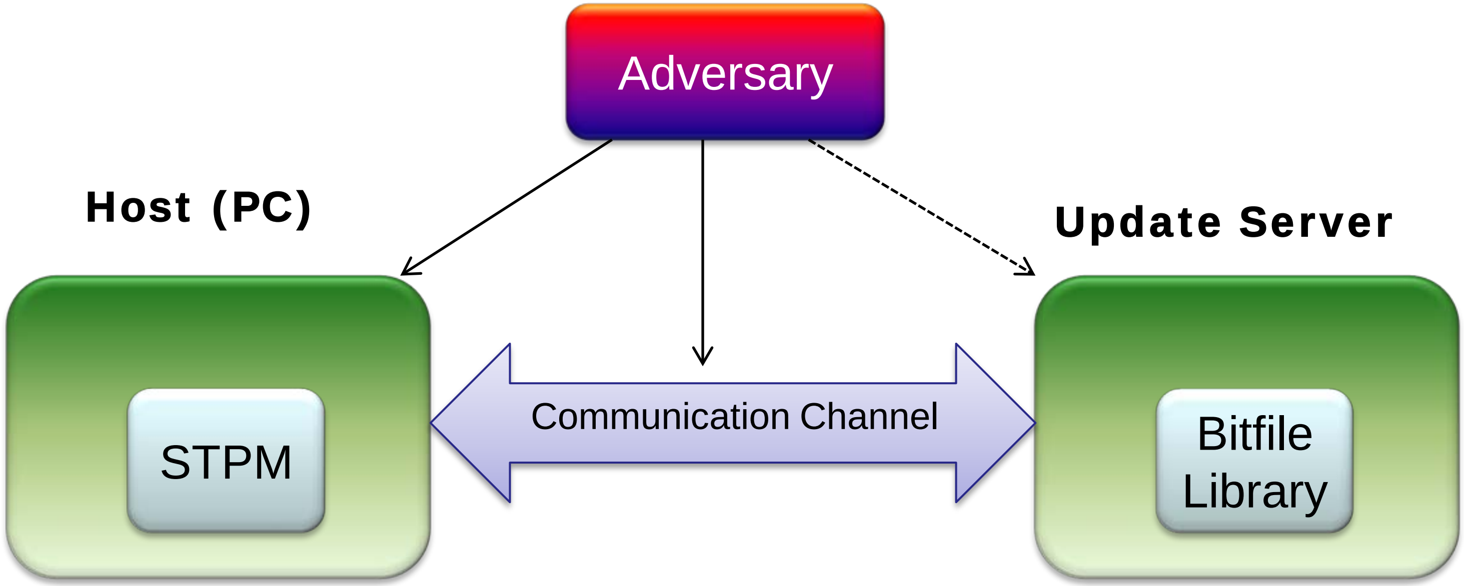
CASED



TECHNISCHE UNIVERSITÄT DARMSTADT
Fraunhofer SIT
h_da HOCHSCHULE DARMSTADT UNIVERSITY OF APPLIED SCIENCES
LOEWE – Landes-Offensive zur Entwicklung Wissenschaftlich-ökonomischer Exzellenz

Adversarial Model

CASED

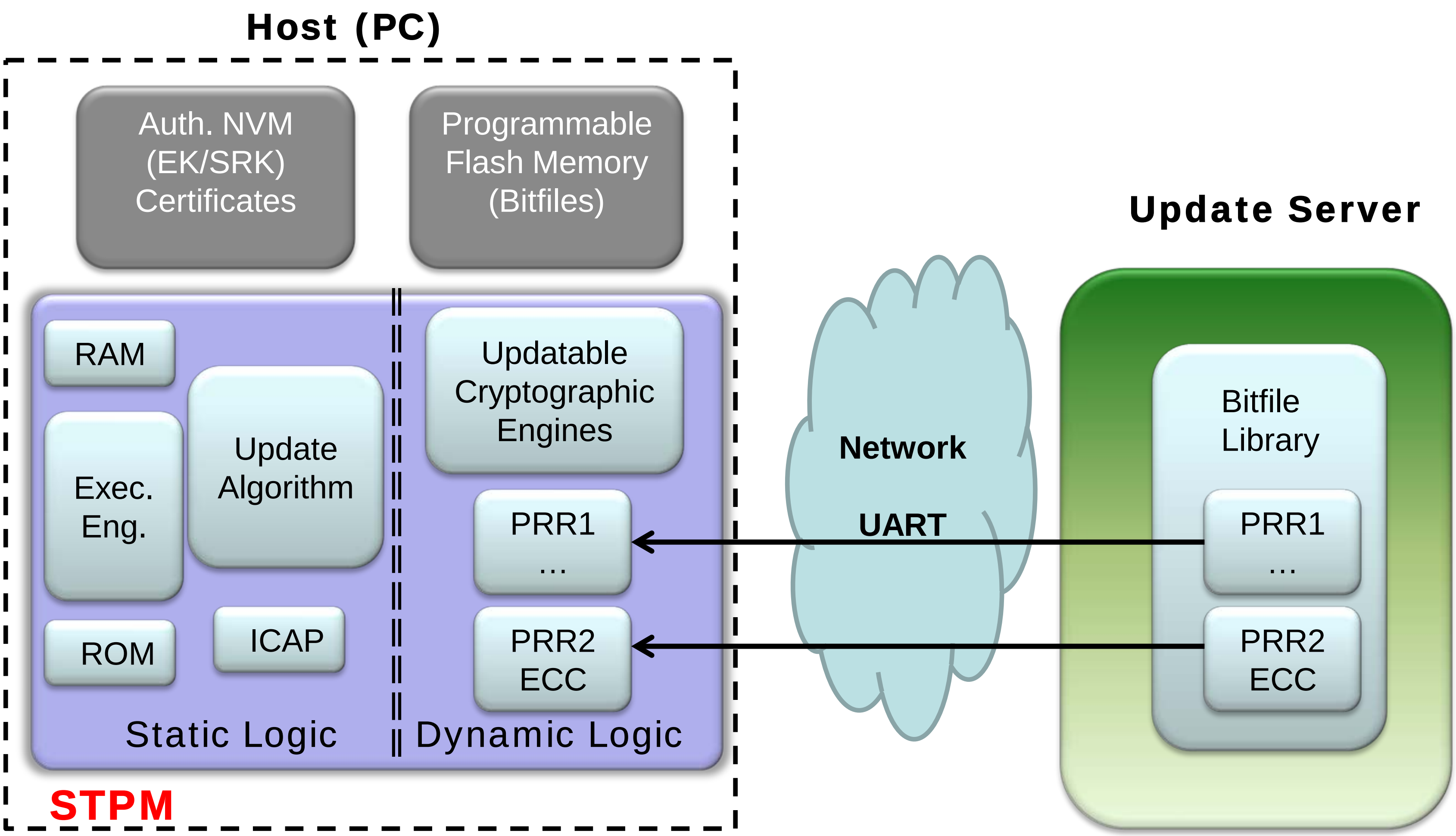


The assumptions on the adversarial model and the update algorithm description are detailed in the paper [12]

Implications of a Compromised RSA Engine

Affected Components and Confidential Data	Key Hierarchy Signatures Bound/Sealed Data
Trust Recovery Strategy	Data Recovery (back-up) Update RSA Engine Sign EK by Trusted Third Party Take Ownership Bind/Seal New Data

Replacing broken RSA Module by an ECC Engine



Implementation

STPM architecture implemented as a proof-of-concept on a Xilinx Virtex-5 LX110T FPGA platform

- MicroBlaze soft-core processor used for control flow and command execution.
- Xilinx ISE (version 11.5) suite and Mentor ModelSim simulator exploited for the design.
- Presented results produced from a synthesis run (using XST).
- Ressource consumption of both static and dynamic regions of the STPM is as follows.

Results (1 / 2)

Module	Registers	LUTs	BRAMs (36 Kbit)
AES-128 Hash-Core HMAC-Core Controller PR ICAP	524 289 294 662 170	899 138 184 453 168	5 0 0 0 2
Update Algorithm	1939	1842	7
Execution Engine	2326	2704	4
Total Static Logic	4265	4546	11

Results (2 / 2)

Crypto Engine	Registers	LUTs	BRAMs (36Kbit)	Frequency (MHz)
RSA	12341	18501	0	17.48
ECC	8851	12925	1	166
SHA-1	1013	1754	0	156.14
HMAC	1722	2353	0	156.14
RNG	1424	1248	5	283.68
TPM Engines	25531	36781	6	

Regaining Trust

CASED

Re-establishing trust in the information processing system is mandatory after an update

- A corrupted RSA engine leads to a compromise of
  - Security functions
  - Keys and signatures
  - Data and commands

- Remote Attestation, Binding, and Sealing are the affected security functions, so following actions are required:
  - Reconstruction of TPM key hierarchy
  - Protection of data by means of new keys

TECHNISCHE UNIVERSITÄT DARMSTADT Fraunhofer SIT h_da HOCHSCHULE DARMSTADT UNIVERSITY OF APPLIED SCIENCES LOEWE – Landes-Offensive zur Entwicklung Wissenschaftlich-ökonomischer Exzellenz

TPM Key Hierarchy

- There exists a fixed key hierarchy in every TPM with the Endorsement Key (EK) and Storage Root Key (SRK).
- The TPM internally generates the additional keys required for various operations using above keys and RSA key generator.
- The keys of the key hierarchy are utilized to encrypt and decrypt the user data and keys.
- A compromised RSA algorithm implies compromised keys and loss of all the data protected by those keys too.
- Thus, a new key hierarchy is to be built after replacing the RSA with an ECC engine.

Remote Attestation and Signatures

- A trusted platform is able to attest the current system state to any requester.
- Usually, the trusted system states are stored in Reference Measurement Lists (RMLs) produced by the IT department.
- In case of a compromised engine, the signatures generated during the remote attestation process become invalid.
- Therefore, all these signatures have to be recomputed by means of the updated asymmetric engine.
- Utilize these new signatures for later operations.

Binding and Sealing

CASED

- Binding means to encrypt the data by keys bound to a specific platform.
- Sealing means to bind the data at a given platform state.
- Once a weakness of RSA has been discovered, all key material is obsolete along with the data encrypted by it.
- Thus, the data bound or sealed to a platform, can not be recovered securely because of the compromised RSA keys.
- However, the new data can be protected by taking advantage of the updated asymmetric engine and the new platform state.

Recovery Strategy Steps (1/2)

CASED

- Check for an availability of the back-up of the data. All data must be restored before updating the engine.
- Perform an update of the broken engine with a new asymmetric engine by utilizing the defined update algorithm.
- A compatible EK with the new asymmetric engine must be loaded/generated and then signed by a trusted party.
- Take ownership of the system to generate a new SRK, which is the root key of the TPM key hierarchy.

Recovery Strategy Steps (2/2)

CASED

- Generate a new key hierarchy utilizing the new SRK and the new asymmetric engine.
- Signatures must be produced utilizing the new asymmetric engine for use in the Remote Attestation procedure
- Bind/Seal the new data with the new keys of the generated key hierarchy.
- Old data may also be secured utilizing the new keys.

Conclusion

CASED

Main contribution:

Novel updatable TPM architecture (STPM) providing:

- Flexible and secure update of cryptographic engines
- Re-establishing the trust in the system after an update
- IP protection and trustworthy attestation

Advantage:

Future embedded systems may be secured and trustworthily operated utilizing the proposed STPM as their security anchor.

Thank you for your attention!

Secret Key Storage Using PUF

Intrinsic-ID™ Quiddikey™ Product: Secure Key Storage

- A key storage product that extracts the key derived from the PUF
- Protects the device and its content against counterfeiting and cloning
- Available as an IP core, can be integrated into any chip design

  Advantage: The key is not present when the device is powered off
- K_Auth as required in NVM access protocol: May be derived from this product and shared between STPM and the external NVM
- Quiddikey™ implementation is available for the Microsemi ProASIC3E FPGA device

Quiddikey^{TM} on ProASIC3E FPGA