REMOTE ATTESTATION OF HETEROGENEOUS CYBER PHYSICAL SYSTEMS

(THE AUTOMOTIVE USE CASE)

KARIM ELDEFRAWY (HRL LABORATORIES)
GAVIN HOLLAND (HRL LABORATORIES)
KMELDEFRAWY@HRL.COM, GHOLLAND@HRL.COM

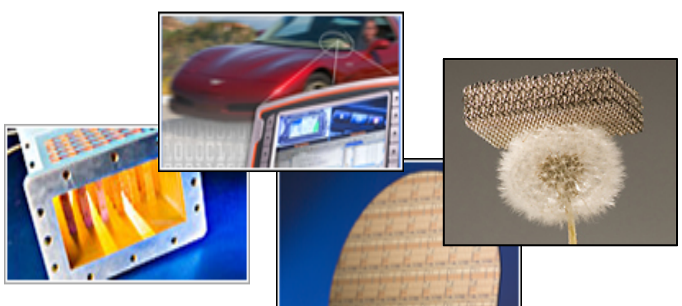
GENE TSUDIK (UC IRVINE)
GTS@ICS.UCI.EDU

This talk is largely based on pervious work by the authors and others in the following papers:
- El Defrawy, Francillon, Perito, Tsudik, Secure and Minimal Architecture for Establishing Dynamic Root of Trust, NDSS 2012.
- Francillon, Nguyen, Rasmussen, Tsudik, A Minimalist Approach to Remote Attestation, DATE 2014.

Overview of HRL

HRL Laboratories, LLC

- Formerly Hughes Research Laboratories (est. 1948)
- Formed as a Limited Liability Company (LLC) , 1997
- R&D for The Boeing Company and General Motors
- Government and commercial contracts
- AS9100 accredited / DoD Trusted Foundry
- 250,000 square feet of lab space
- 10,000-square-foot Class 10 clean room
- Located on 72 acres in Malibu, CA



General Motors

- General Motors Corp. est. in 1908
- #2 in sales globally (7.5M vehicles in 2009)
- 200,000+ employees worldwide, 200+ facilities
- GM R&D: world's first automotive research ctr.
- Milford Proving Grounds: industry's first dedicated automobile testing facility
- Long history in new technologies and breakthrough innovations, dating back to 1920
- 1,123 US patents in 2011 alone
- 1st place team in DARPA Urban Challenge

http://media.gm.com/product/public/us/en/gmfacts/history/timeline.html

Outline

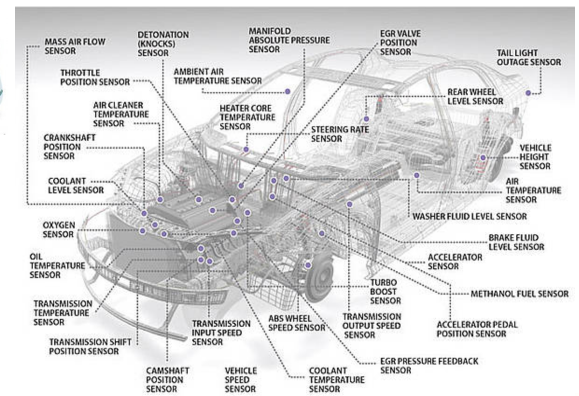
- Introduction and Motivation
- Prelims for Remote Attestation
- Secure and Minimal Architecture for (Dynamic) Root of Trust (SMART)
- Future Directions

Widening Range of Specialized/Embedded Devices

Smartphones and Watches

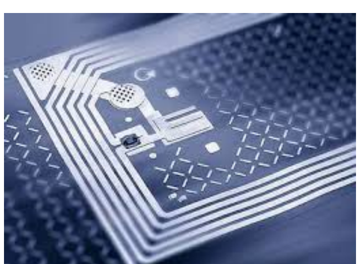
Connected Devices

Sensors and Actuators



SmartCards

RFIDs



Appliances

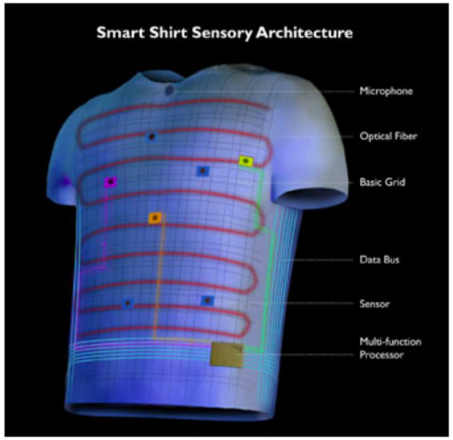
Automotive Systems

Industrial Systems

Already Here or Coming Soon

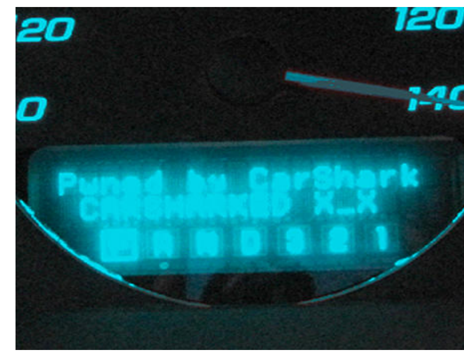
- Smart watches, e.g., Apple, Samsung

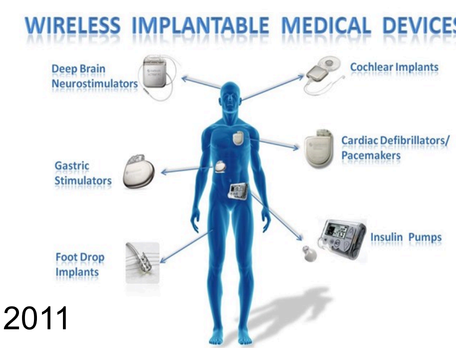
- Smart glasses and personal (VR) displays, e.g., Google Glass, Oculus Rift, Samsung
- Smart footwear, e.g., Nike+
- Smart clothes and garments (outer and under)



Notable Attacks on Embedded Systems

- Stuxnet [1] (also DUQU)
  - Infected controlling windows machines
  - Changed parameters of the PLC (*programmable logic controller)* used in centrifuges of Iranian nuclear reactors
- Attacks against automotive controllers [2]
  - Internal controller-area network (CAN)
  - Exploitation of one subsystem (e.g., bluetooth) allows access to critical subsystems (e.g., braking)
- Medical devices
  - Insulin pumps hack [3]
  - Implantable cardiac defibrillator [4]
- Most effective CPS attacks are **remote** infestations, i.e., **<u>not physical attacks</u>**





[1] W32.Stuxnet Dossier, Symantec 2011
[2] Comprehensive Experimental Analyses of Automotive Attack Surfaces, USENIX 2011
[3] Hacking Medical Devices for Fun and Insulin: Breaking the Human SCADA System, Blackhat 2011
[4] Pacemakers and Implantable Cardiac Defibrillators: Software Radio Attacks and Zero-Power Defenses, S&P 2008

Specialized/Embedded Devices in the Automotive Domain

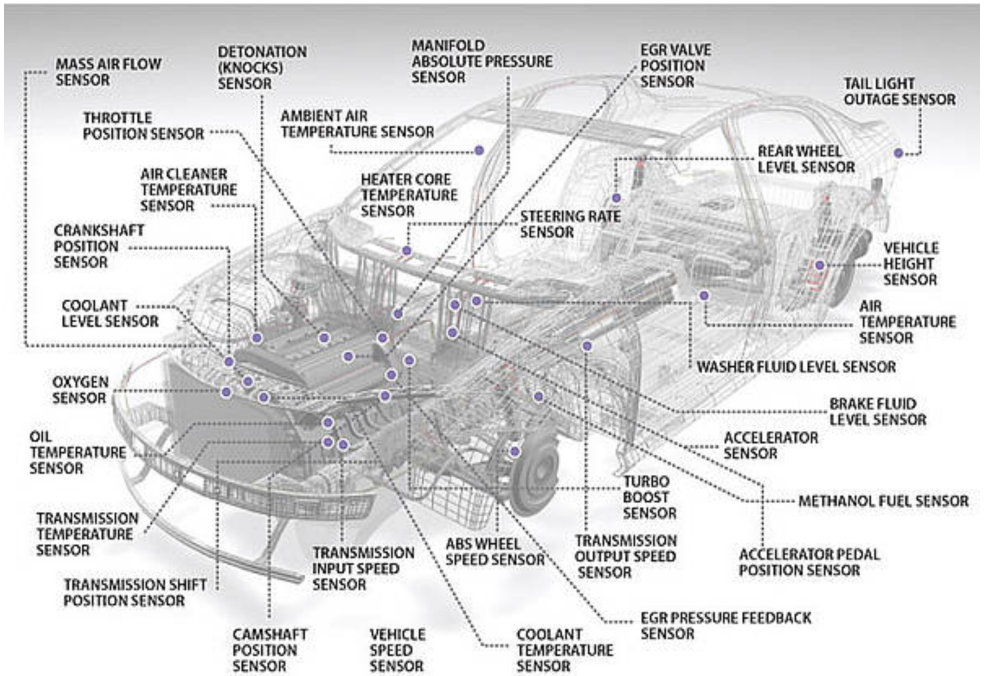


Figure source: http://www.can-cia.org/index.php?id=1691

According to sensormag.com
http://www.sensorsmag.com/product/automotive-sensor-market-worth-3578-billion-2022

Market report on "Automotive Sensor Market by Product (Pressure, temperature, level, speed, MEMS, oxygen, Nox), Application (powertrain, safety & control, vehicle security, alternative fuel, telematics) and Geography - Forecast & Analysis to 2013 – 2022" is expecting market to grow at a **CAGR of 8.6% from 2014 to 2022** and reach $35.78 Billion in 2022.

"Modern cars have become complex digital devices, which can contain over 70 electronic control units (ECUs) …"
https://www.escar.info/escar-usa.html

Trusted Computing Group (TCG) Automotive Focus

"Given the diverse use cases inside the vehicle, it is reasonable to describe a vehicle as a composite industrial control system network with one or more Internet Gateways and one or more human user interfaces." -- TCG TPM 2.0 Automotive Thin Profile, March 16th 2015

TCG TPM 2.0 Automotive Thin Profile

TCG TPM 2.0 Automotive Thin Profile

Family "2.0"

Level 00 Version 1.0

March 16, 2015

Contact: admin@trustedcomputinggroup.org

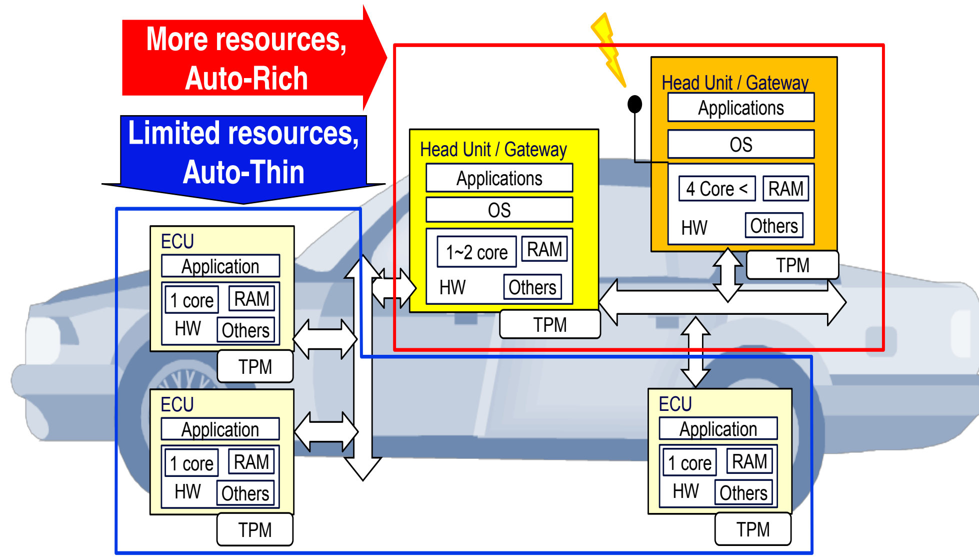


Figure source: TCG TPM 2.0 Automotive Thin Profile, March 2015
http://www.trustedcomputinggroup.org/resources/tcg_tpm_20_library_profile_for_automotivethin

Trusted Computing Group (TCG) Automotive Focus

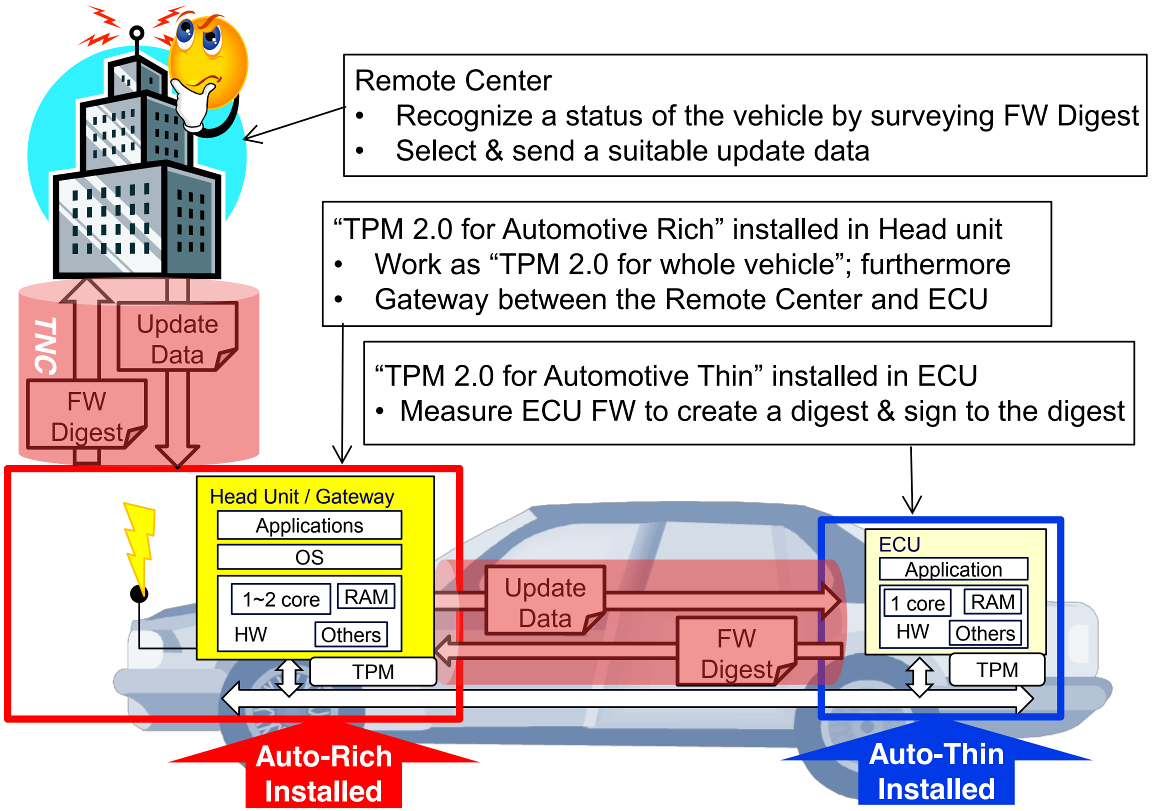


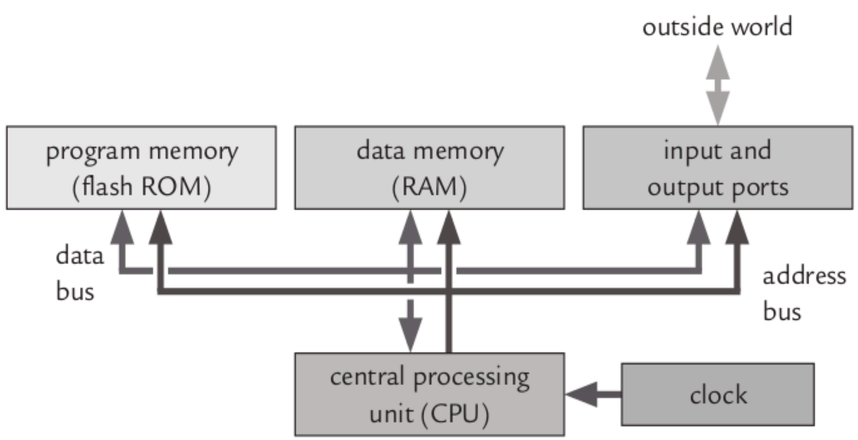
Figure above shows the message flow for each component (Head Unit/Gateway or ECU) for remote maintenance handled by Automotive-Rich, and -Thins.

Figure source: TCG TPM 2.0 Automotive Thin Profile, March 2015
http://www.trustedcomputinggroup.org/resources/tcg_tpm_20_library_profile_for_automotivethin

Low-end Embedded Devices (Automotive-Thin in TCG Language)

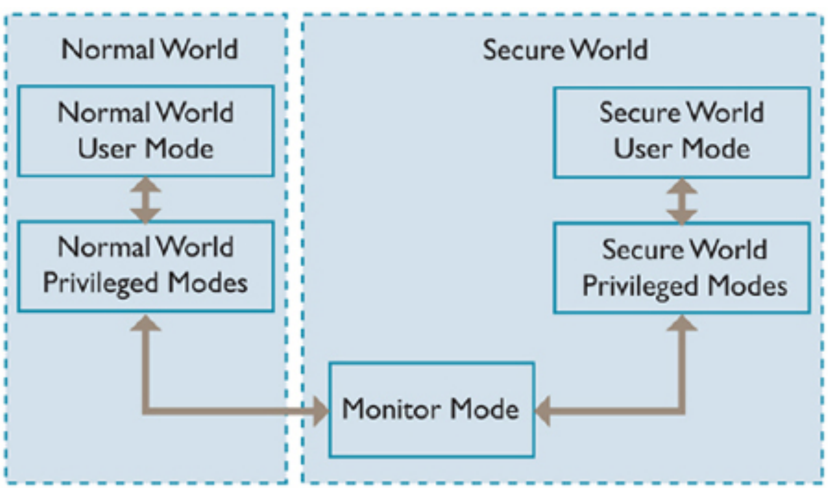
- Memory: program (e.g., 128KB Flash) and data (e.g., 4KB SRAM)
- Typically built around an MCU (serving as CPU), Integrated clock
- As well as:
  - Communication interfaces (USB, CAN, Serial, Ethernet, etc.)
  - Analog to digital converters

- Examples: TI MSP430, Atmel AVR, Raspberry Pi



High-end Embedded Devices (Automotive-Rich in TCG Language)

- **Contrast with high-end processors, e.g., ARM, Intel**
- **Possibly built-in cryptographic support/functions, e.g., TPM, secure boot, HW-based isolation**
- **Notable example: ARM Trustzone**



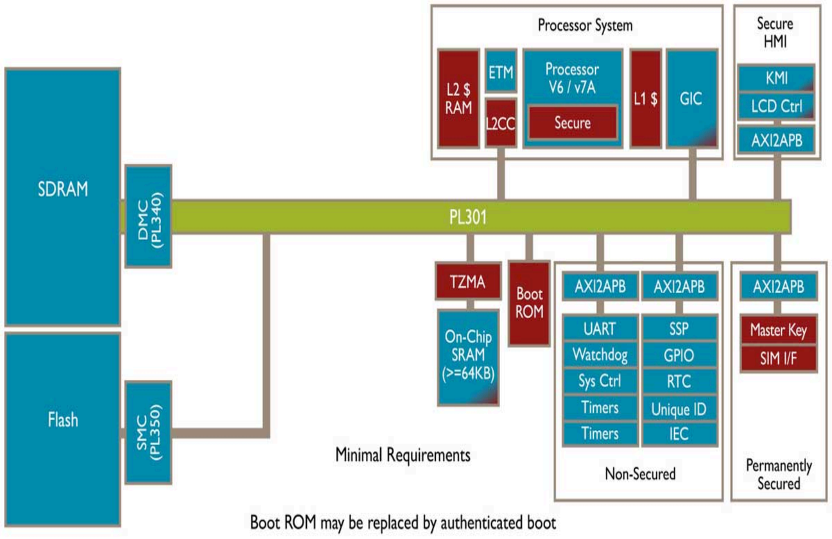


Figure sources: http://www.arm.com/products/processors/technologies/trustzone/index.php

Issues in a (Heterogeneous) CPS

- **Unrealistic that every processor will be Trustzone-like, maybe (at most) 1 or 2 in the system**
- **Cost is a serious limitation**
- **"Trust Anchors" in a large CPS can be built on more powerful CPUs, they can attest other CPUs/MCUs**

CPS Definition: "A cyber-physical system (CPS) is a system where there is tight coordination of the system's computational and physical elements, though sensors and actuators"

Disclaimer: This Talk is …

- Not a final solution for securing heterogeneous CPS, more research still needed (also ongoing work to standardize it)
- A description of design and performance of an essential component for remote attestation for low-end (automotive-thin) embedded devices
- A blueprint for how the entire system can be attested
- Outline of future direction and research

Outline

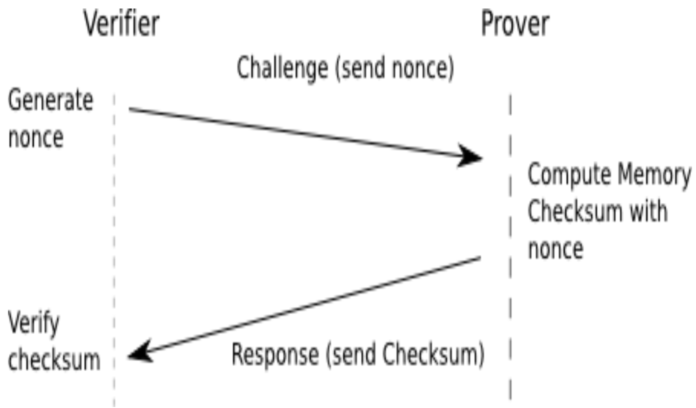
- Introduction and Motivation
- Prelims for Remote Attestation
- Secure and Minimal Architecture for (Dynamic) Root of Trust (SMART)
- Future Directions

Remote Attestation and Required Security Goals

Remote Attestation Definitions

– Two party protocol between trusted verifier and untrusted prover

– Remotely verify the internal state of the prover



Internal state of prover is composed of: code, registers, data memory, i/o

Three types of attestation:

– Hardware-based: secure hardware supported (e.g., TPM)

– Software-based attestation: does not support multi-hop communication

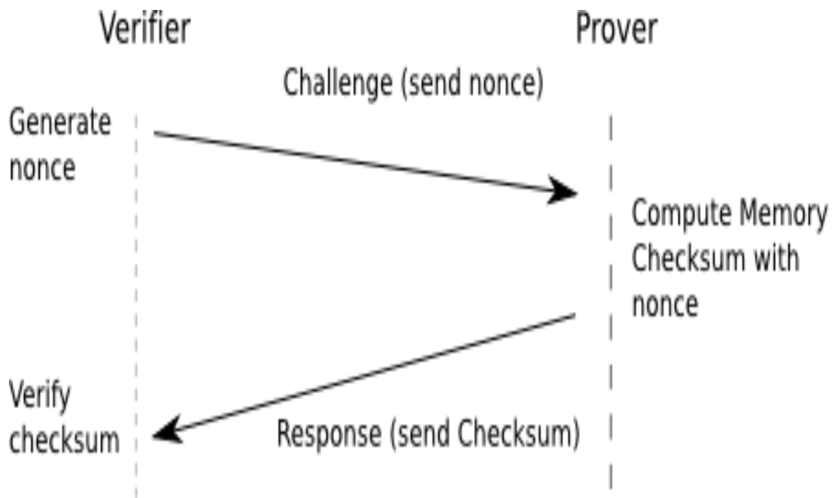
– Hybrid: minimal hardware support and changes (this talk)

Formalizing Remote Attestation (1)

- **Malicious software will lie about the software state of the prover**
- **Need to have guarantees that the device is not lying**

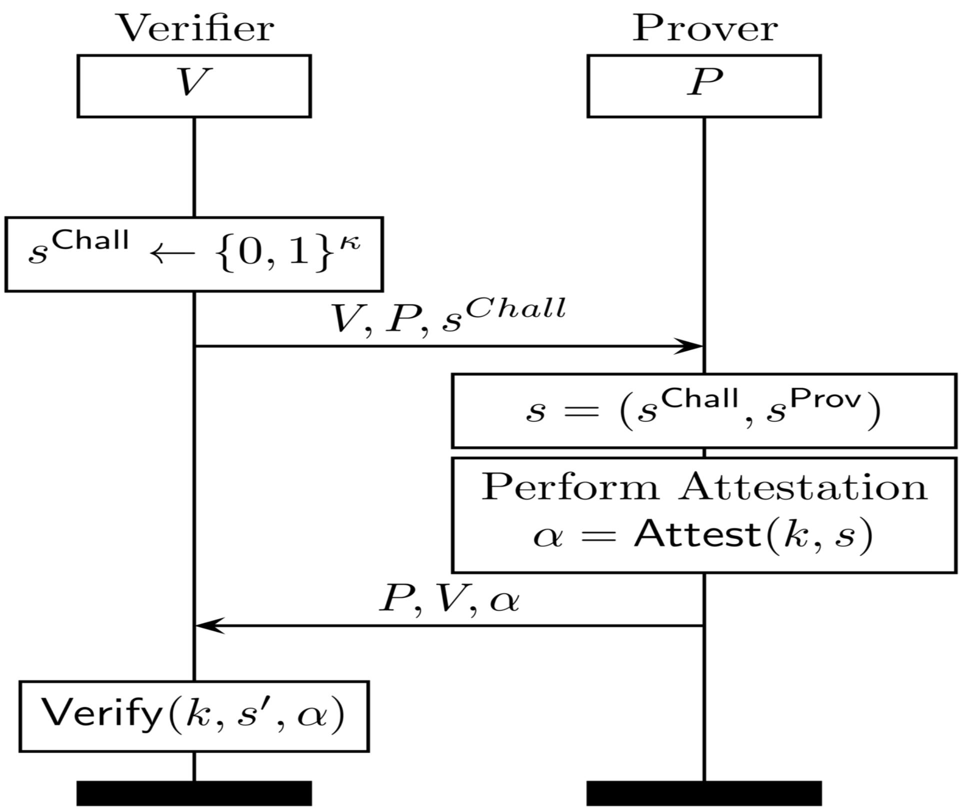
Attestation protocol *P = (Setup, Attest, Verify)*:

- ***k = Setup(1^{κ})***
  a setup procedure to generate a shared key
- ***α = Attest(k, s)***
  Key, Device state => Attestation token
- ***output = Verify(k, s, α)***
  Key, Expected state, Token => Yes/No



Formalizing Remote Attestation (2)

(1) Pick a random challenge



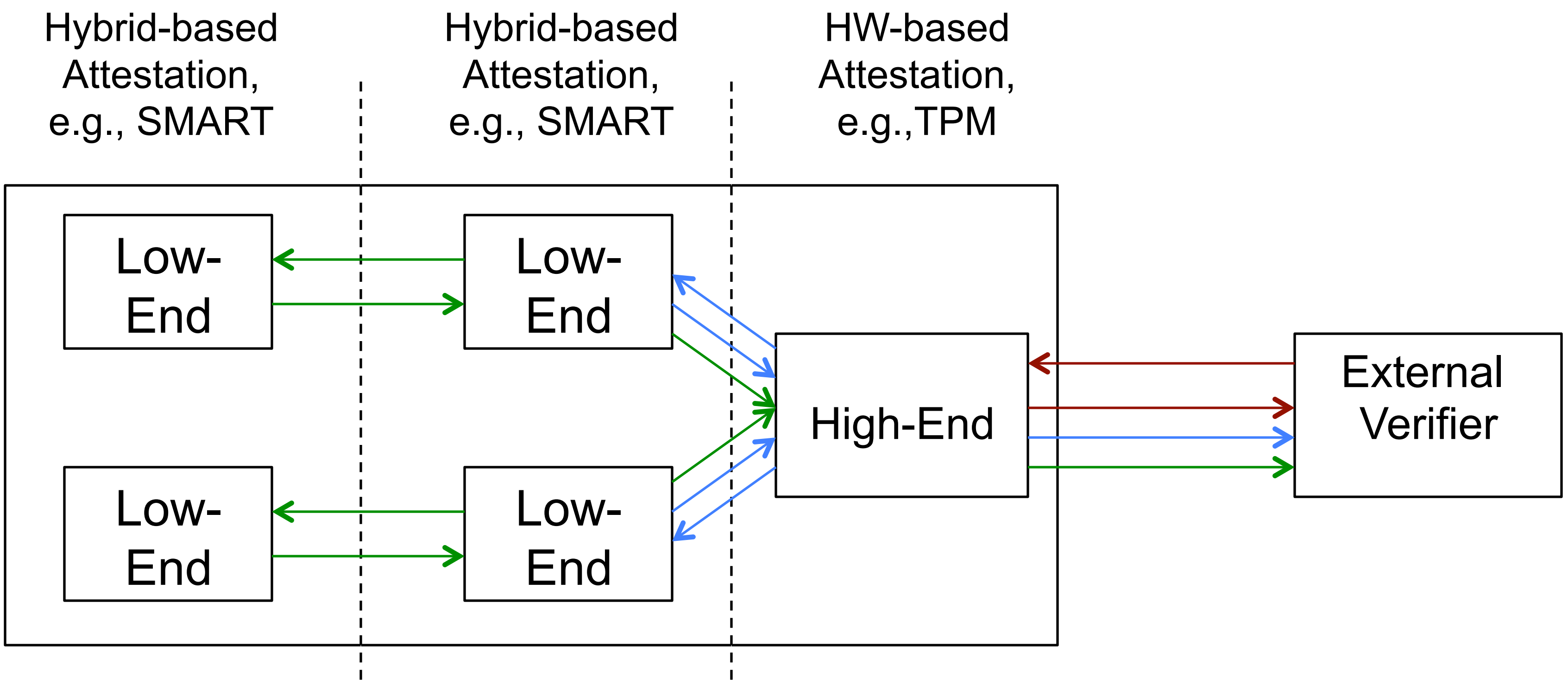
(2) State of prover + challenge + key

(3) Verify that the returned output corresponds to expected state

Attestation protocol may also return the exact state

Attestation of a Heterogeneous CPS

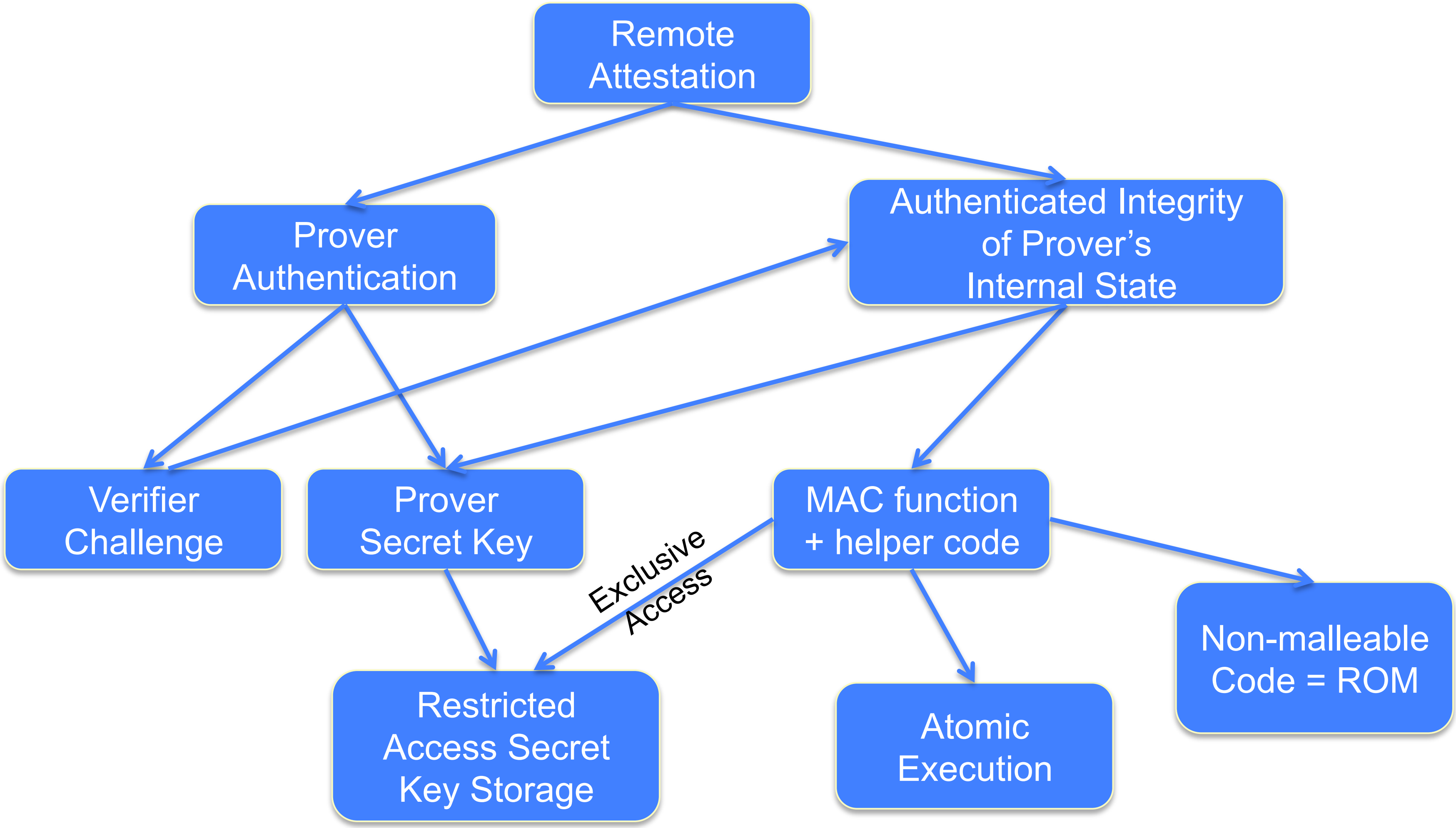
- Assume there are two types of devices:
  - High-end CPUs with TPM and hardware functionalities
  - Low-end CPUs without TPM (SMART could be a solution)
  - Otherwise: use SW-based attestation on Low-end (weak security guarantees)



Outline

- Introduction and Motivation
- Prelims for Remote Attestation
- Secure and Minimal Architecture for (Dynamic) Root of Trust (SMART)
- Future Directions

Systematic Analysis of Required Features



Building Blocks

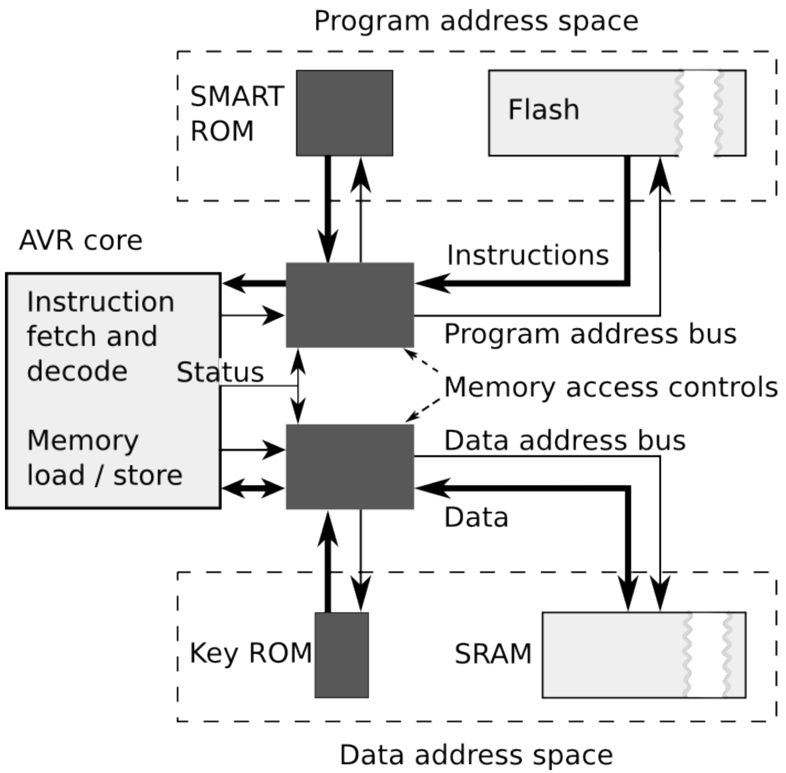
1. Secure Key Storage (as little as 180 bits)
   – Required for remote Prover
   – Enables Prover authentication

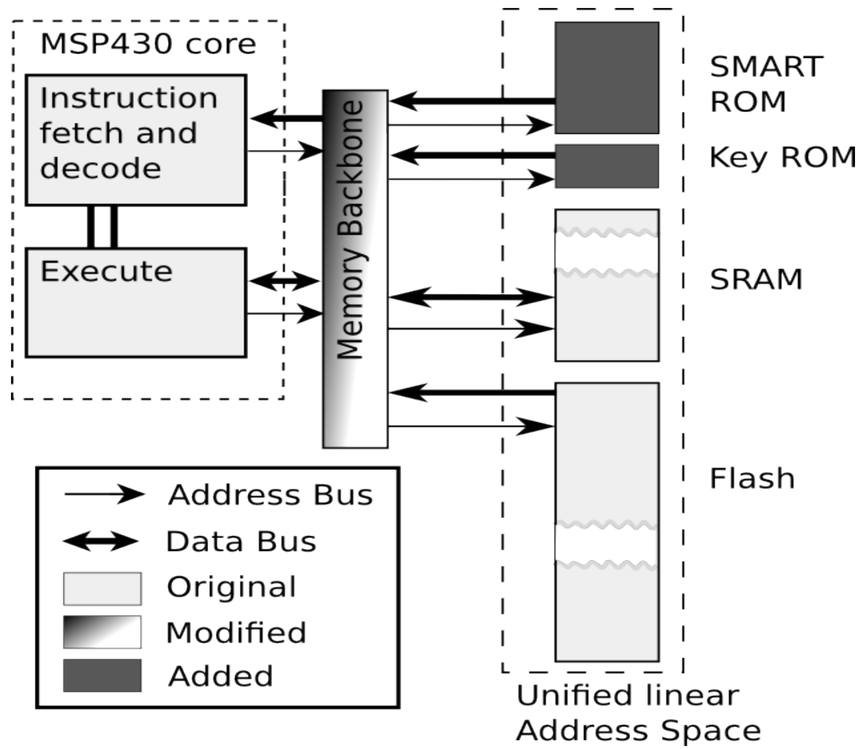
2. Trusted ROM code memory region
   – Read-only means integrity: computes response
   – Accesses/uses key (exclusively)

3. MCU access control
   – Grants access to key from within ROM code only

4. Atomicity of ROM code execution
   – Disable/enable interrupts
   – No invocation other than from the start

Scope of Modifications to MCUs

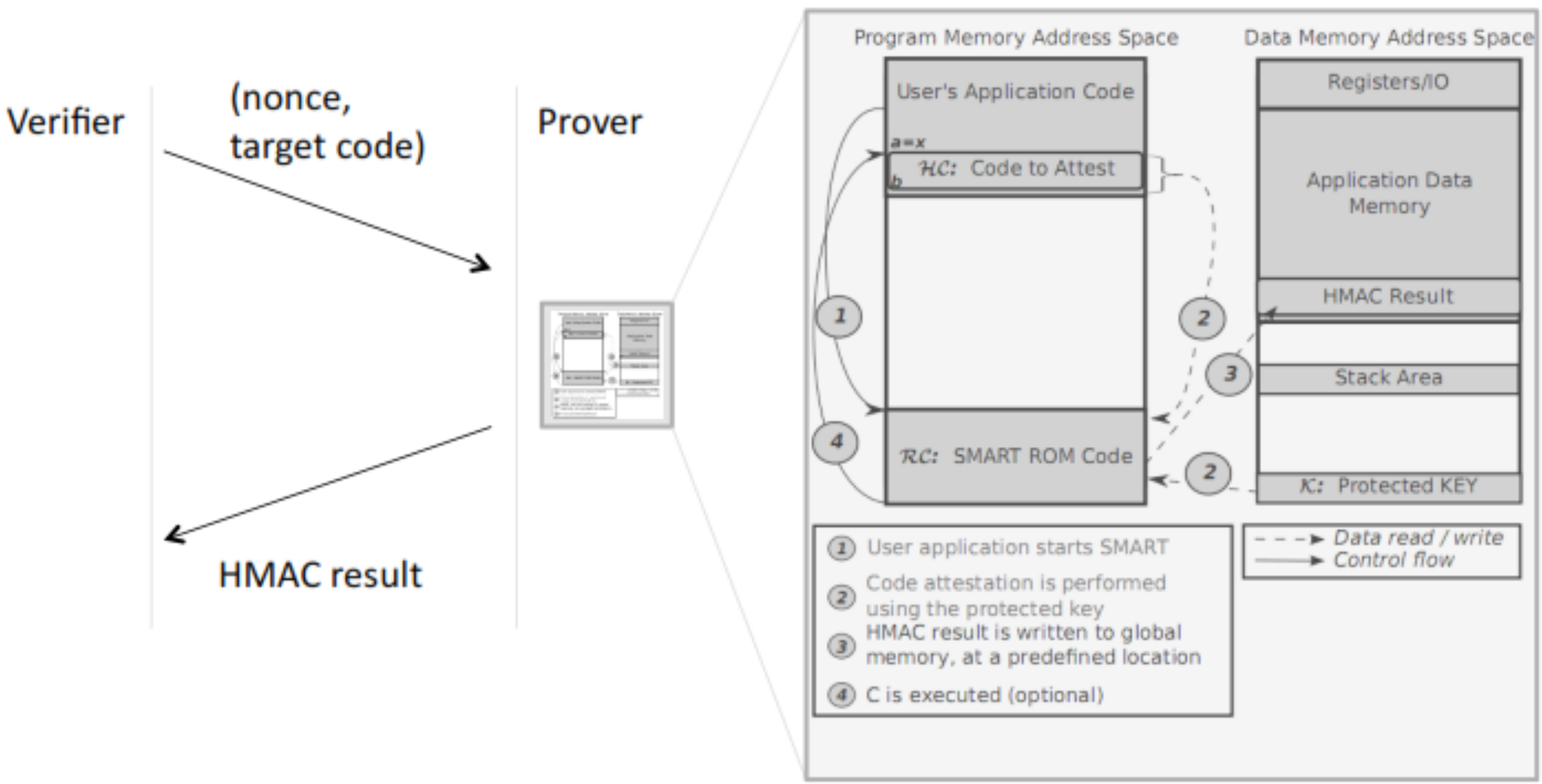




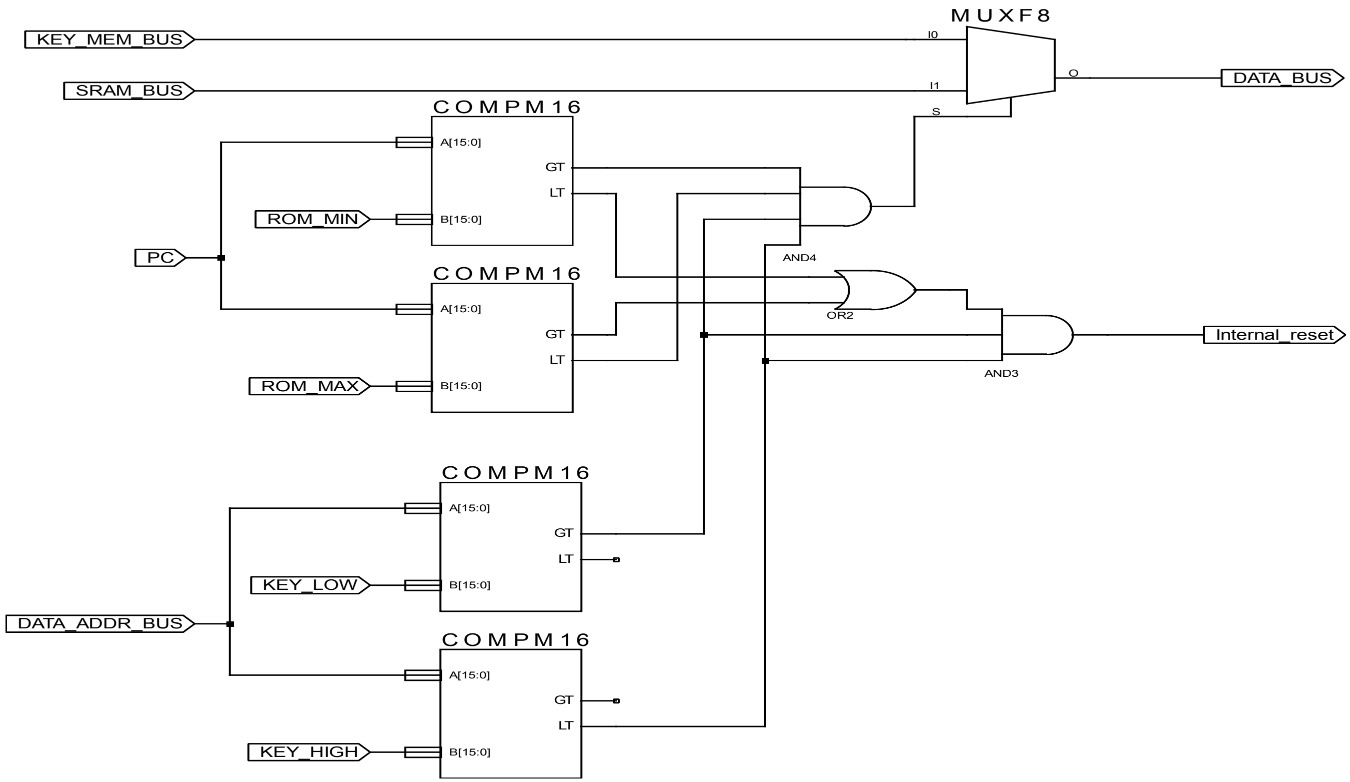
AVR: Dark gray boxes represent logic added to the processor. Core control signals provide information about internal processor status to memory bus controls.

MSP430: Memory backbone was modified to control access to ROM and key. MSP430 is based on Von Neumann architecture, concurrent access can occur to different memory parts (e.g., instruction fetch and read data). In that case, memory backbone arbitrates bus access and temporarily saves/restores data.

The Complete SMART Protocol



Required Access Control Logic for Secret Key



Design and Operation Issues

- If Prover infected, ROM code and malware share the same MCU resources
  - Malware can set up execution environment to compromise ROM code and extract key
  - Malware can schedule interrupts to occur asynchronously while key (or some function thereof) is in main memory
  - Malware can use code gadgets in ROM to access key
    - Return-Oriented Programming (ROP)
  - ROM code might leave traces of key in memory after its execution

Countermeasures

- Atomic ROM code execution: enforced in hardware
  - Enter at first instruction
  - Exit at last instruction
- ROM code instrumented to check for memory safety
  - Used DEPUTY
  - Upon detecting error reboot and clear memory
- Interrupts disabled immediately upon ROM entry
  - Before key usage (enabled upon exit)
  - DINT instruction must itself be atomic
- Erase key-related material before end of execution

Cost of Adding ROM and Access Control

Implemented on two commodity low-end MCU platforms (AVR and MSP430)

Component		Original Size in kGE	Changed Size in kGE	Ratio
AVR MCU		103	113	10%
Core		11.3	11.6	2.6%
Sram	4 kB	26,6	26.6	0%
Flash	32 kB	65	65	0%
ROM	6 kB	-	10.3	-
MSP430 MCU		128	141	10%
Core		7.6	8.3	9.2%
Sram	10 kB	55.4	55.4	0%
Flash	32 kB	65	65	0%
ROM	4 kB	-	12.7	-

Comparison of chip surface required by each component of original MCU to SMART-modified version. kGE stands for thousands of Gate Equivalents (GE-s). One GE is proportional to the surface of the chip and computed form the module surface divided by the surface of a NAND2 gate, 9,37 x 10^{-6} mm^2 with this library.

HMAC Performance and Effort to Implement

HMAC is the most expensive operation to perform attestation in SMART

Data Size	Cycles	Time at 8MHz
1 KByte	2302281	287 ms
512 Bytes	1281049	160 ms
32 Bytes	387471	48 ms

Changes made (in # of HDL lines of code) in AVR and MSP430 processors, respectively, excluding comments and blank lines.

Component	Original	Changed	
	Lines	Lines	Ratio
AVR, core (VHDL)	3932	151	3.84%
AVR, tests	2244	760	
MSP430, core (Verilog)	4593	182	3.96%
MSP430, tests	17665	1122	

Secure Remote Attestation for Low-end Devices

- **Introduction and Motivation**
- **Prelims for Remote Attestation**
- **Secure and Minimal Architecture for (Dynamic) Root of Trust (SMART)**
- **Future Directions**

Open Issues and Research Directions

- Asymmetric vs symmetric cryptography on Prover
- Automated synthesis of attestation and formal verification of implementation
- Platform for more sophisticated or specialized services: secure code update, secure erasure, secure boot
- More experiments and implementation on various platforms/CPS
- Verifier Authentication (very relevant to mitigate denial-of-service)

Questions?

SMART as an API

Algorithm SMART usage to attest a memory range.

input : n nonce sent by $\mathcal{VRF}$ → challenge
a start address to attest
b end address to attest } → memory range to attest
H HMAC result (global variable)

output: HMAC output

begin
  SMART (a, b, $\emptyset$, $False$, n, &H, $\emptyset$) ;
  Send(H);
end

For More Details

K. Eldefrawy, A. Francillon, D. Perito and G. Tsudik, SMART: Secure and Minimal Architecture for Establishing Dynamic Root of Trust, NDSS 2012.

A. Francillon, Q. Nguyen, K. Rasmussen and G. Tsudik, A Minimalist Approach to Remote Attestation, DATE 2014,

- **full version in Crypto ePrint Archive: Report 2012/713.**