PUFs: Anchors of Trust in Resource Constrained Environments

Georgios Selimis(IID), Roel Maes(IID), Erik van der Sluis(IID), Frans Willems(TUE) and Martin Deutschmann(TEC)

The E! 9629 PATRIOT project has received funding from the Eurostars-2 joint programme with co-funding from the European Union Horizon 2020 research and innovation programme.

Project & team

- PATRIOT- **P**UFs: **A**nchors of **T**rust in **R**esource ConstraI**n**ed EnvirOnmenTs
  - Goal: Prototype for IoT Security based on PUF-SRAM technology.
  - Focusing on authentication and security for resource constrained IoT devices.
  - EUREKA Eurostars project (October 2015-September 2017).

INTRINSIC ID

TU/e Technische Universiteit Eindhoven University of Technology

TECHNIKON

IoT = Internet of Threats?

UAV relies on sensors to keep flying

Errors in sensor inputs can trigger major consequences in hubs

Industry needs reliable supply

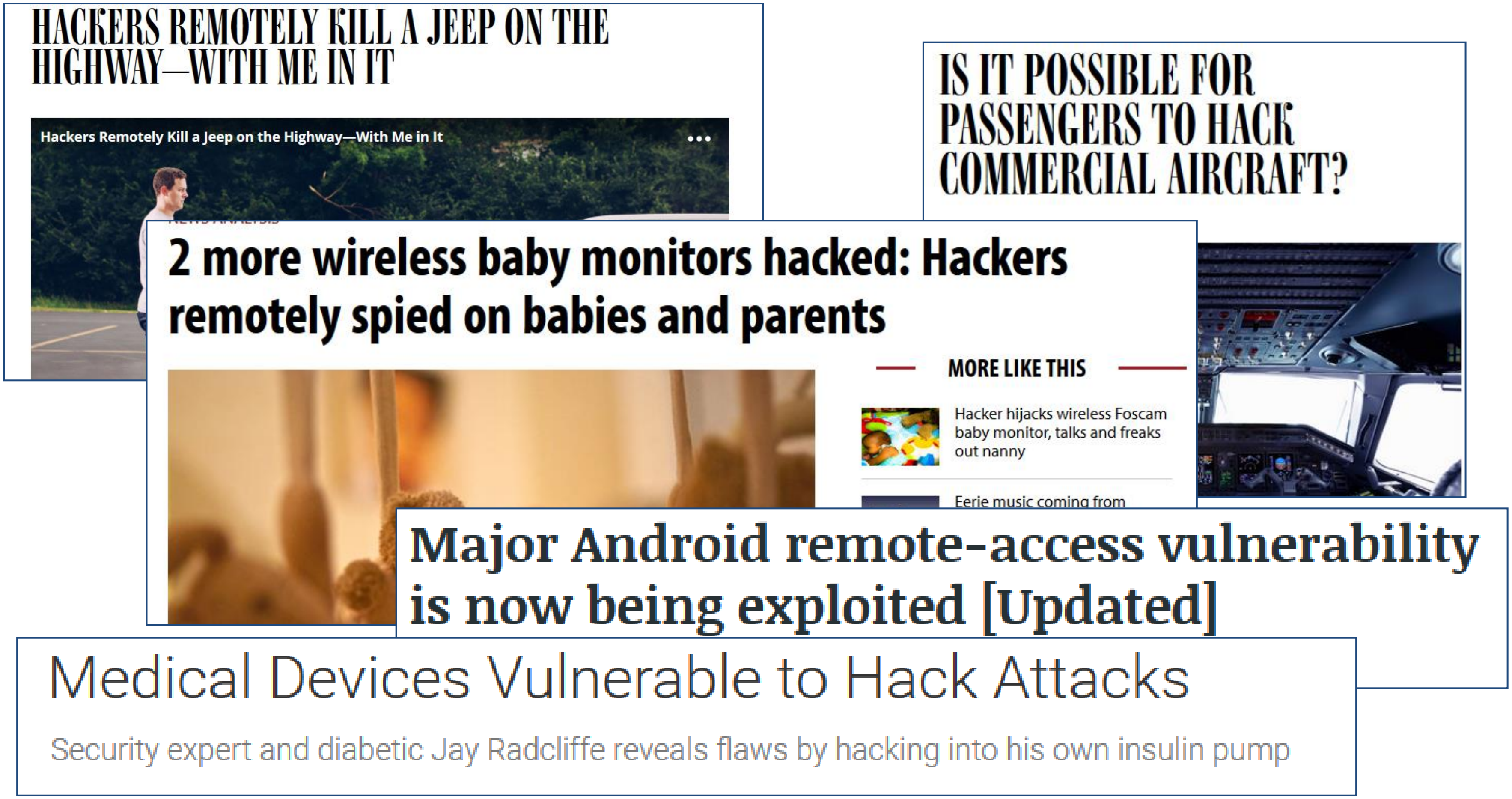
Medical sensors drive automated devices (pacemaker, insulin pump)

(semi) autonomous vehicles rely on sensors for safety

Users have an incentive to hack their own meters

Physical access to remote and distributed infrastructure cannot be prevented

IoT hacks on the news


HACKERS REMOTELY KILL A JEEP ON THE HIGHWAY—WITH ME IN IT

Hackers Remotely Kill a Jeep on the Highway—With Me in It

IS IT POSSIBLE FOR PASSENGERS TO HACK COMMERCIAL AIRCRAFT?

2 more wireless baby monitors hacked: Hackers remotely spied on babies and parents

MORE LIKE THIS

Hacker hijacks wireless Foscam baby monitor, talks and freaks out nanny

Eerie music coming from

Major Android remote-access vulnerability is now being exploited [Updated]

Medical Devices Vulnerable to Hack Attacks

Security expert and diabetic Jay Radcliffe reveals flaws by hacking into his own insulin pump

PATRIOT security challenge

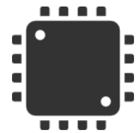
Need for security

Challenging environment:

- Critical data
- Connectivity
- No user trust
- 24/7 autonomous running

No Flash

No secure storage



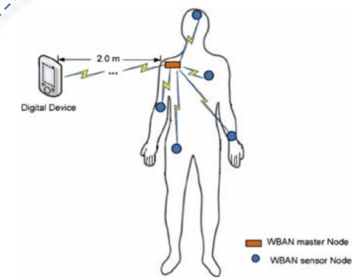
Few MHz CPU

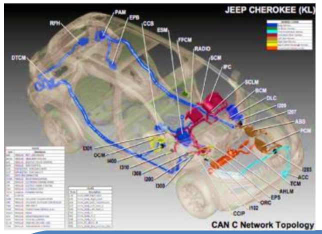
Limited RAM Few kbyte

Resource constrained

Limited resources

- No NVM, no secure storage
- Limited Resources (power, memory, processing)





Complexity

Complexity:

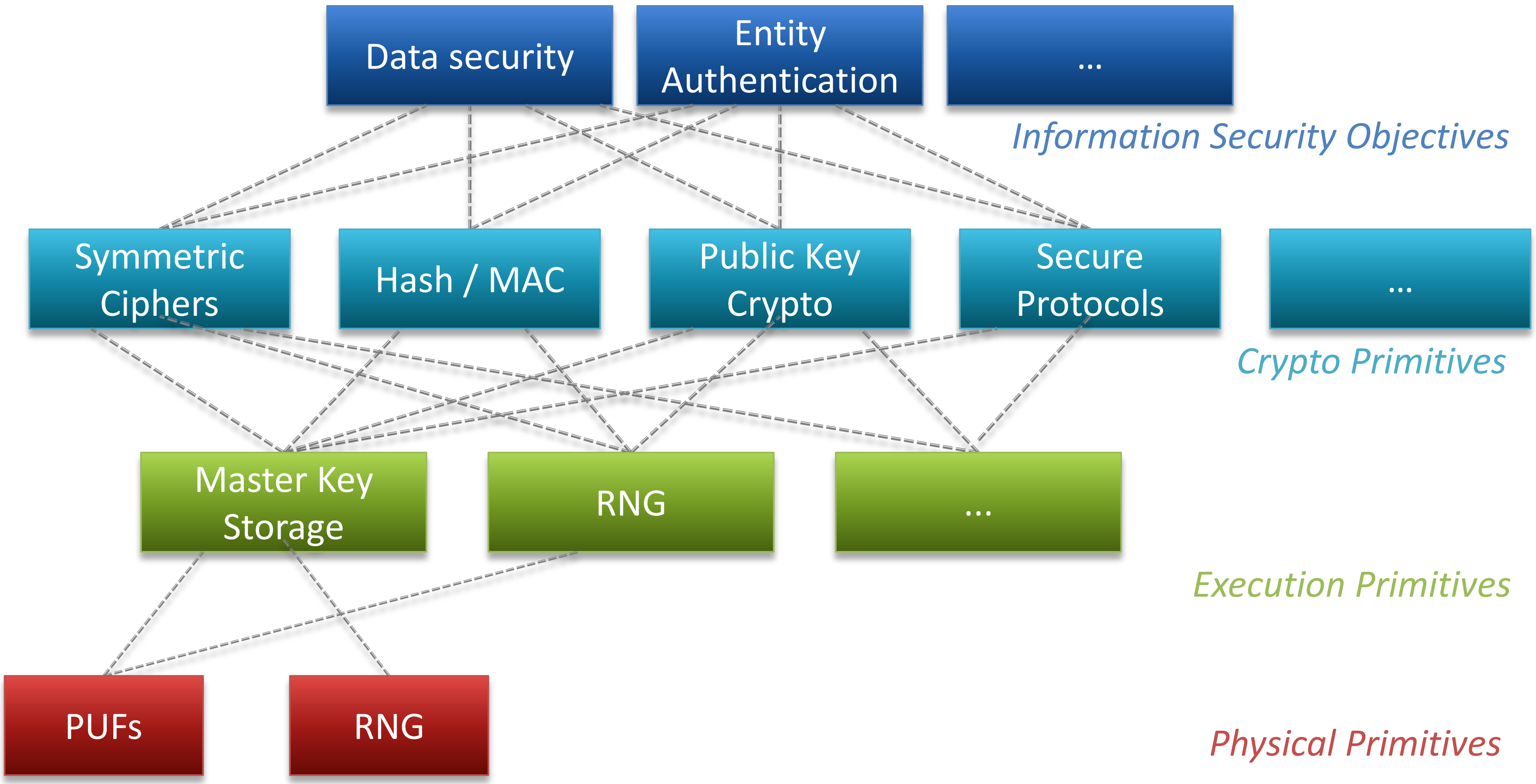
- High number of interconnected devices → Need for better key management mechanisms

Root of Trust (RoT) for resource constrained devices

- What is RoT?
  - Embedded systems: Multiple layers of abstraction, such as hardware, firmware, operating system and applications.
  - The higher layers should trust the lower layers.
  - The initial source of trust at the bottom of the system is called RoT.
- Existing solution: Trusted Platform Module
  - Best suited for high-end devices, such as laptops and smartphones but rely on added hardware.
- Our RoT solution
  - Binding the sensitive IoT information such as the root key, to existing on chip SRAM hardware components that can be found in any microprocessor (MPU) or microcontroller (MCU).
  - SRAM - PUF technology for secure key generation, key storage, key management and randomness extraction.

PATRIOT

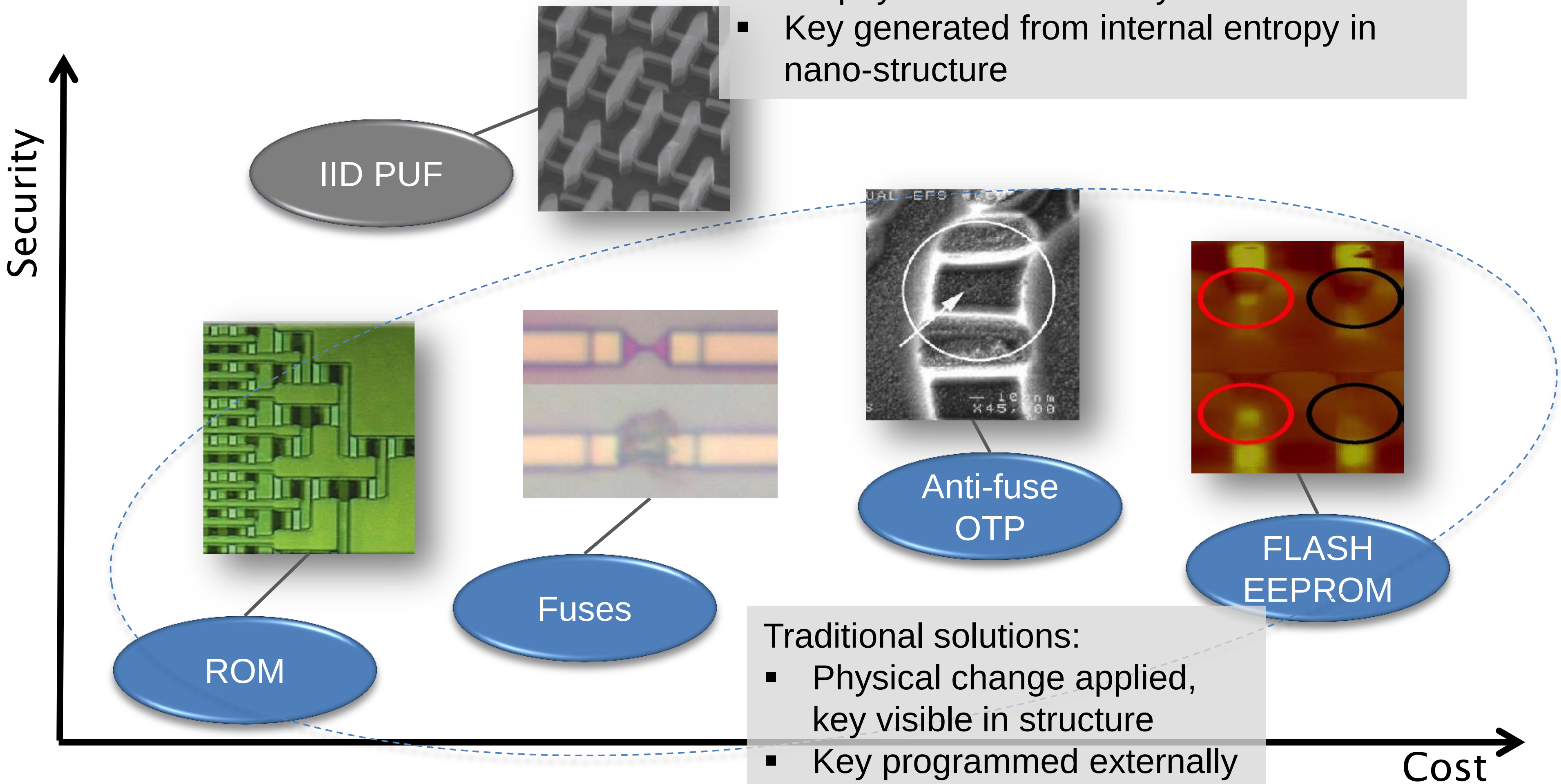
RoT in Embedded devices



Key Storage Security

PUF based key storage:

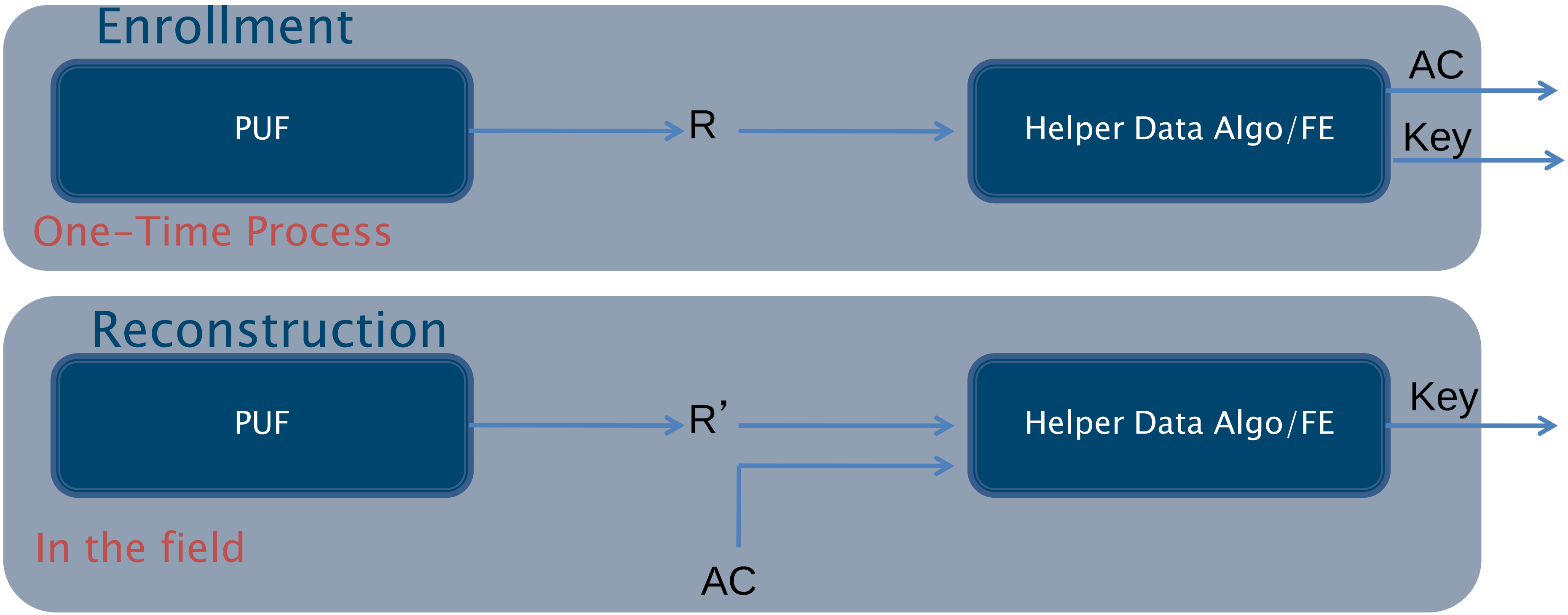
- No physical traces of any sensitive data
- Key generated from internal entropy in nano-structure



Traditional solutions:

- Physical change applied, key visible in structure
- Key programmed externally

Key Storage with SRAM PUFs

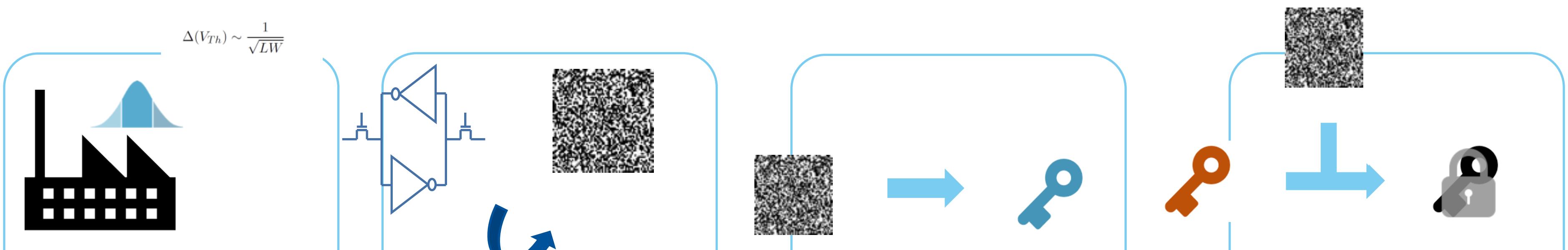


$I(AC, Key) < \varepsilon$ $P[\text{Key not Correct}] < \delta$

Main advantages

- **Availability**: uninitialized SRAM memory is present in almost every device
- **Flexibility**:
  - Implementation in hardware, software or both.
  - Allows for secure operations without requiring embedded NVM.
- **Security**: Strong protection against physical attacks, no keys permanently stored

SRAM-PUF Intrinsic-ID Technology



Uncontrollable deep sub-micron process variations

Start–up SRAM values establish a unique, silicon fingerprint

Turned into unique device key and ID

Program and store user keys

SRAM Reliability

- Aging: 25yr +
- Voltage +/- 20%
- T°: -50°C - 150°C
- Humidity Test
- EMC Test

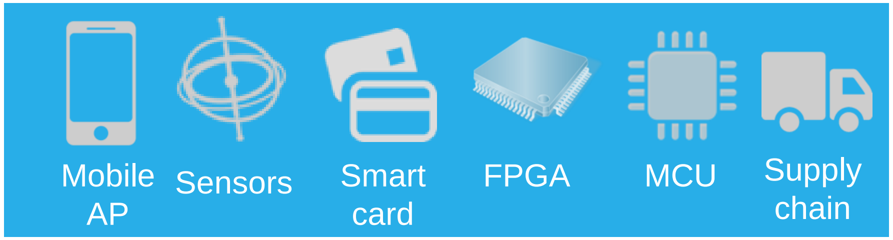
Worst case key reconstruction error rate < 10^{-9}

Foundries Tested Across the World

Process Nodes Tested

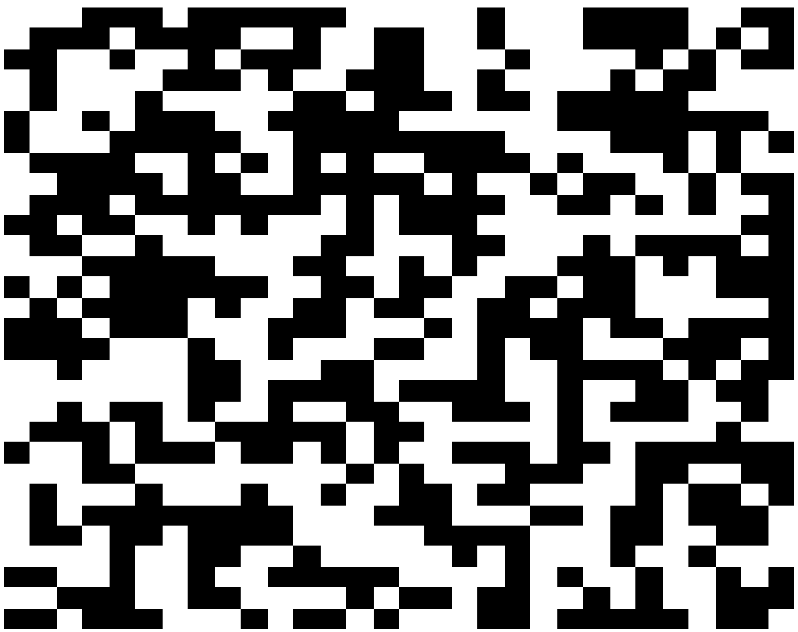
180 nm	160 nm	150 nm
130 nm	90 nm	65 nm
45 nm	40 nm	28 nm
20 nm	16 nm	14 nm

Deployed

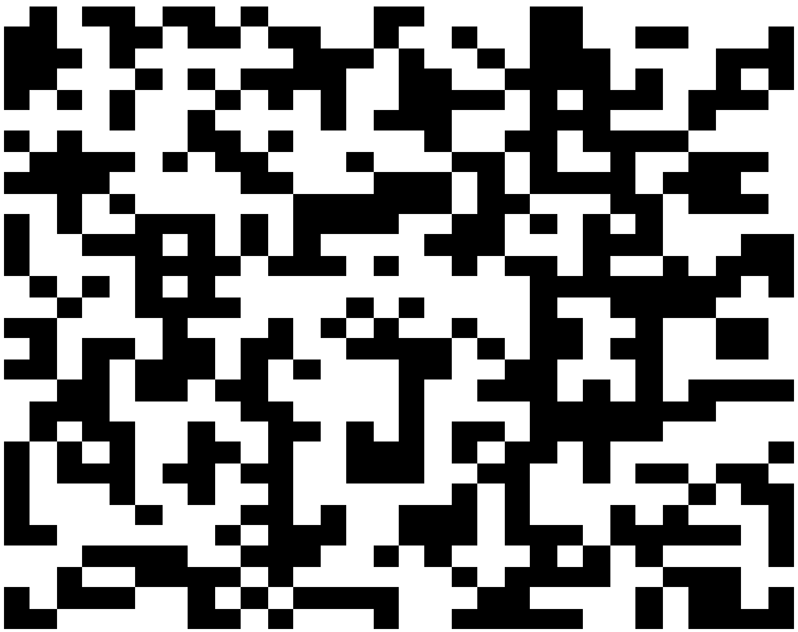


SRAM Fingerprint Uniqueness

Device 1



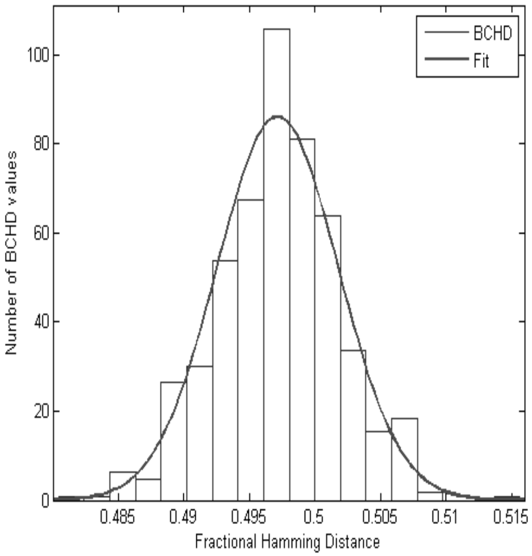
Device 2



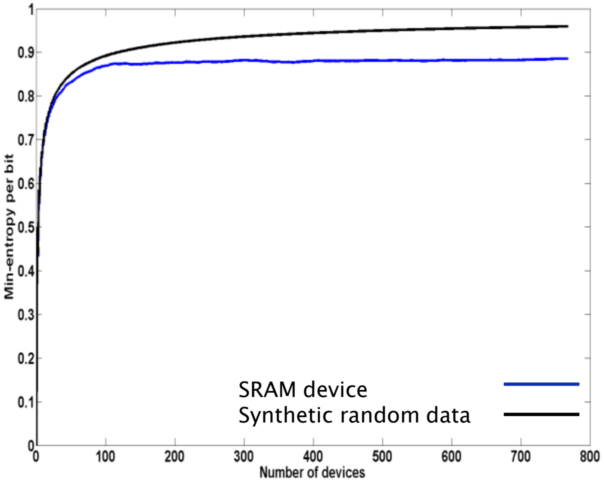
~ 50% difference

Security

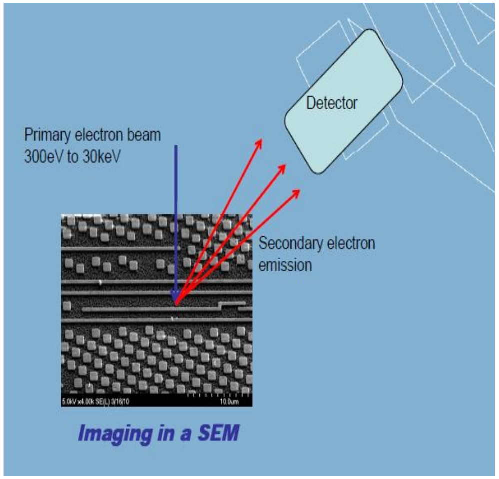
Unique



High Entropy



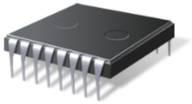
Attack robustness



Security for resource constrained devices

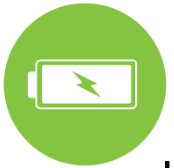
- RoT and Secure storage
  - SRAM-PUF secure and low-cost solution.

- Resource efficient security protocol
  - Move the complexity to the peers with more processing capabilities.



- Resource efficient crypto hardware
  - Design for optimizing area, latency, lifetime.

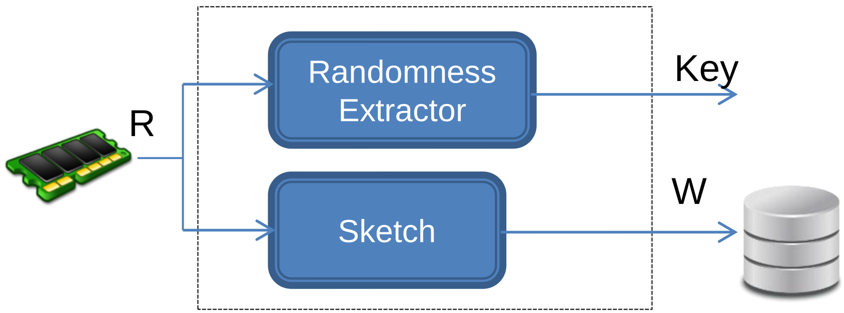
- Resource efficient crypto software
  - SW for less memory accesses, latency, code size.



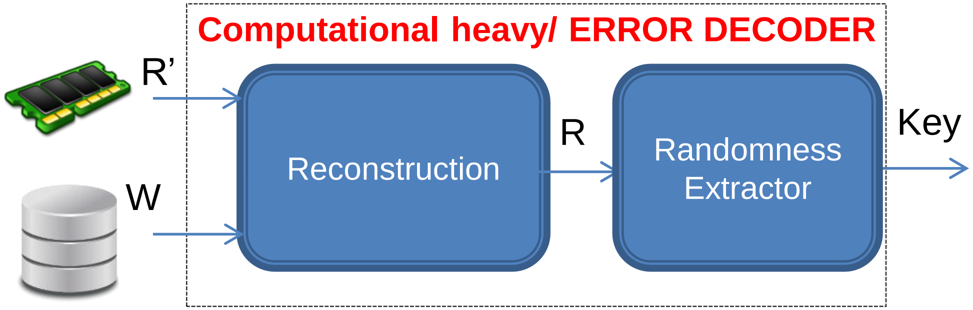
- Power management
  - Optimize the power consumption when the component is on sleep mode.

Fuzzy Extractor

Enrollment phase



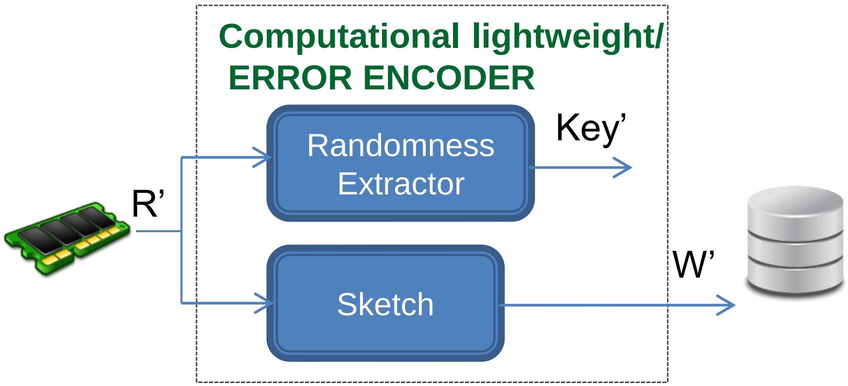
Reproduction phase



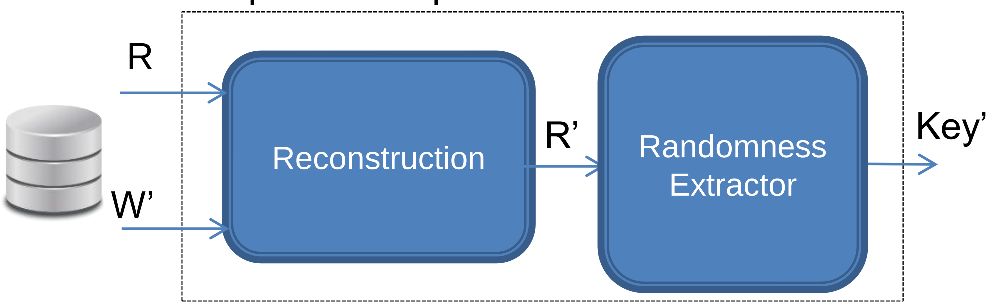
Reverse Fuzzy Extractor

Enrollment phase

Reproduction phase 1



Reproduction phase 2



Van Herrewege, A., Katzenbeisser, S., Maes, R., Peeters, R., Sadeghi, A.R., Verbauwhede, I. and Wachsmann, C., 2012. “*Reverse fuzzy extractors: Enabling lightweight mutual authentication for PUF-enabled RFIDs*”. in Financial Cryptography and Data Security (pp. 374-389). Springer Berlin Heidelberg.

Reverse Fuzzy Extractor: Working Principle

(Trusted) Back-end Database | **One-time trusted enrollment phase** | **PUF-Equiped HW Token**

(Trusted) Back-end Database	Message	PUF-Equiped HW Token
	Enroll →	**PUF → R** **(Block future enrollments)**
Store (ID, R) in DB indexed by ID	← **ID,R**	

In-the-field mutual authentication phase

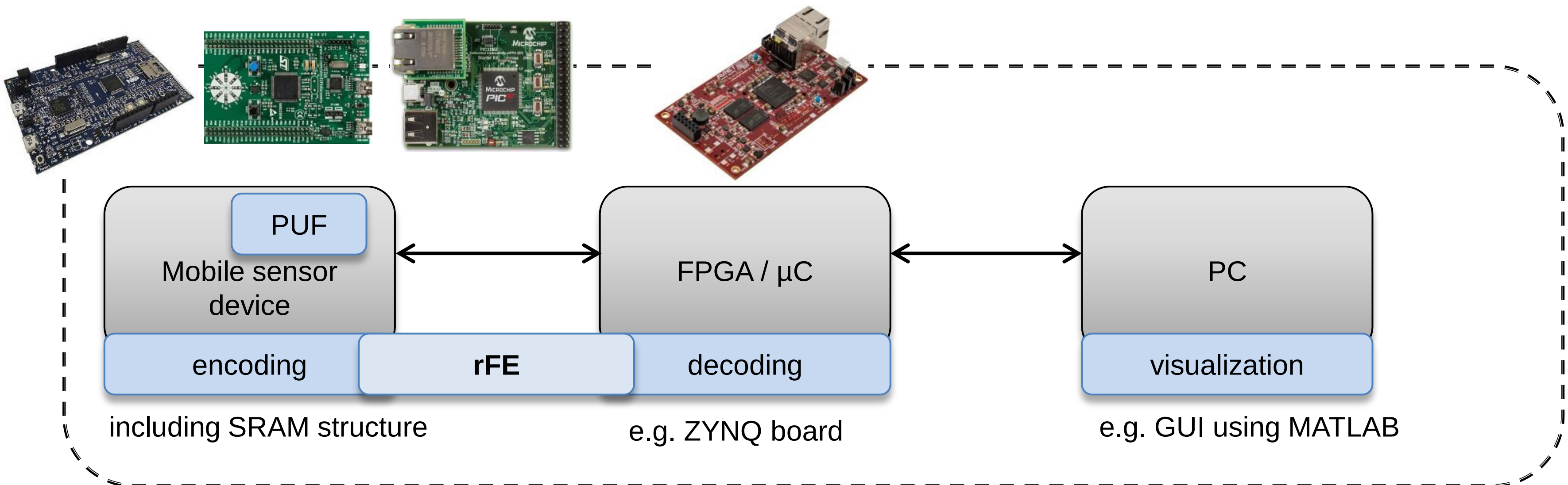
(Trusted) Back-end Database	Message	PUF-Equiped HW Token
TRNG → N_1	**"Authenticate", N_1** →	**PUF → R'** **TRNG → S, N_2** **C = Encode(S)** **W' = R' + C** **T_1 = Hash(S, ID, N_1, N_2)**
Lookup (ID, R) in DB **C' = W' + R** **S* = Decode(C')**	← **ID, W', T_1, N_2**	
Verify: **T_1 == Hash(S*, ID, N_1, N_2)** **T_2 = Hash(S*, ID, N_2, N_1)**	**T_2** →	**Verify:** **T_2 == Hash(S, ID, N_2, N_1)**

Research challenges

- Proof that authentication remains secure even after knowledge of $(W'_1,\ldots,W'_N)$ for arbitrary N
  - Proof that $H(S \mid W'_1,\ldots,W'_N) > 0$
- PUF post-processing
  - Implementation simple and lightweight.
  - Reliability very high [failure rate < 1.00E-09].
  - Keep the redundancy low (code rate ~ 1/20).
  - Consider entropy loss (e.g. due to bias).

Requirements	
Bit Error Rate	**25%**
Bias (50% +/- …)	**30%**
Failure Rate (FRR)	**1.00E-09**
Security Level	**256**
PUF Size	**1kB**
Key Gen./Rec.	**Yes**
Key Storage	**Yes**
Entity Authentication	**Yes**
Data Authentication	**Yes**
Data Confidentiality	**Yes**

Prototype: Implementation details



Conclusions: Secure Design for limited resources

- Prototype demonstrating our security solution for IoT
- Resource efficient security protocol
  - rFE: Move the complexity to the peers with more processing capabilities.
- RoT
  - Using existing SRAM.
  - Secure storage.
- Efficient Design
  - Resource efficient crypto hardware.
  - Resource efficient crypto software.
- Research
  - Secure Reverse Fuzzy Extractor.
  - Efficient PUF post-processing signal processing.

Questions?