Remote Attestation on IoT devices

EDLIRA DUSHKU
Postdoc in Secure Pervasive Computing
Cybersecurity Engineering Section
DTU Compute, Technical University of Denmark

Contact: edldu@dtu.dk

Outline

- **Internet of Things Security**
  - Motivation
  - Overview of Remote attestation
  - History of Remote attestation

- **Remote attestation protocols**
  - Software-based attestation
  - Hybrid attestation
  - Swarm attestation
  - Dynamic attestation
  - Distributed services attestation

- **Research perspectives**
  - Recent research projects
  - Conclusions
  - Open challenges & Future works

Outline

- **Internet of Things Security**
  - Motivation
  - Overview of Remote attestation
  - History of Remote attestation

- **Remote attestation protocols**
  - Software-based attestation
  - Hybrid attestation
  - Swarm attestation
  - Dynamic attestation
  - Distributed services attestation

- **Research perspectives**
  - Recent research projects
  - Conclusions
  - Open challenges & Future works

Internet of Things (IoT) systems

Industrial IoT

IoT for instrustructure

Consumer IoT

Internet of Things (IoT) systems

Cyberattacks on Iran — Stuxnet and Flame

News about Cyberattacks on Iran — Stuxnet and Flame, including commentary and archival articles published in The New York Times.

About 90% of Smart TVs Vulnerable to Remote Hacking via Rogue TV Signals

Oct. 10, 2017

How Israel Caught Russian Hackers Scouring the World for U.S. Secrets

Exploiting the popular Kaspersky antivirus software, Russian hackers searched millions of computers for American intelligence keywords. Israeli intelligence tipped off American officials

ANDY GREENBERG SECURITY 07.21.15 06:00 AM

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

Casino Gets Hacked Through Its Internet-Connected Fish Tank Thermometer

Sunday, April 15, 2018 Wang Wei

Share 9.13k Share Tweet Share

Over 8,600 vulnerabilities found…

FDA recalled half a million pacemakers…

"If you want to keep living, pay a ransom, or die…"

How to improve the situation?

Option 1: Security-by-design

Security-by-design

Security-by-~~design~~: What device producers think it is ?

magic

- No cybersecurity expert
- No additional time/money
- Rush to market

IoT devices prone to cyberattacks

EASY TO EXPLOIT

- Resource-constrained devices with low-cost design
- Do not support complex security techniques

ATTRACTIVE TARGET

- Deployed in safe-critical domains
- Contain sensitive data & control physical environment

AMPLIFY THE ATTACK IMPACT

- Many interconnected devices
- Spread quickly the malware

Option 1: Security-by-design

Difficult: Cannot guarantee that devices do not get compromised

Option 2: Malware detection

Detect compromised device (to isolate from the network)

How to detect malware presence?

Guarantee that the device is
“telling the truth”
even when it is infected by malware

Remote attestation (RA)

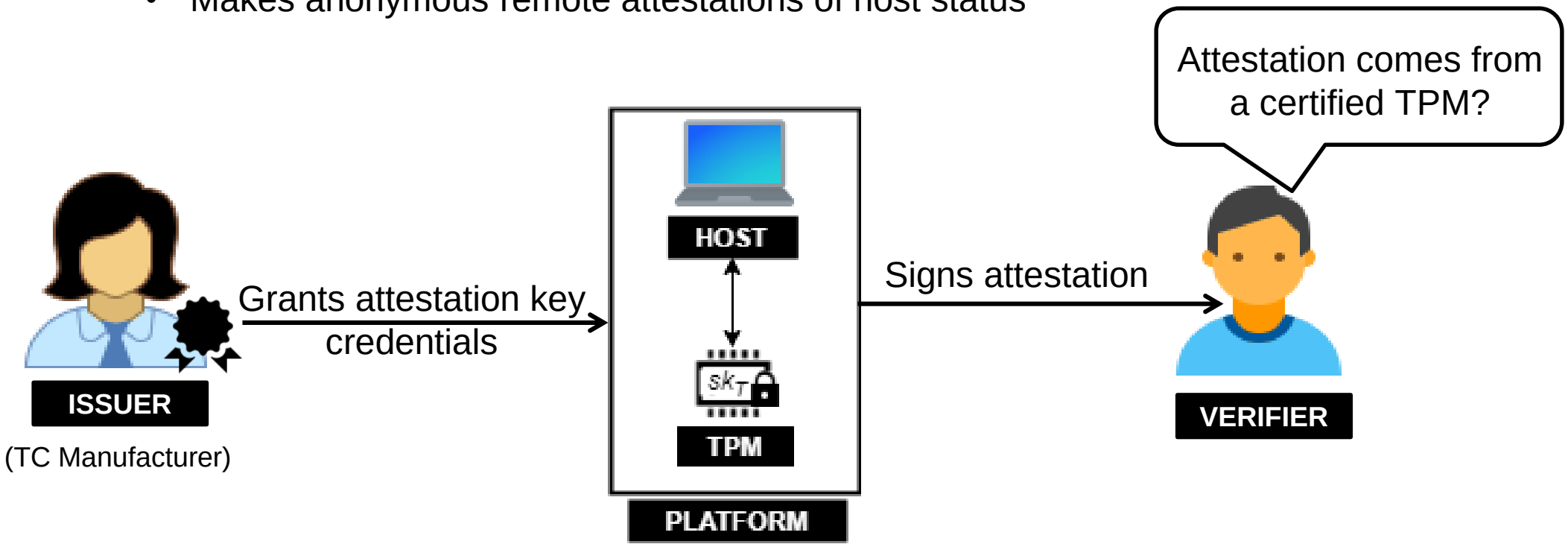
- Two-party Security Protocol
  - **Verifier**: an external trusted entity, not always present, not possible to physically reach a device
  - **Prover**: a (potentially) compromised device

- RA allows the **Verifier** to **guarantee** the **authentication and integrity** of the software running on **Prover**

- Verify that Prover is **NOW** running the initial application

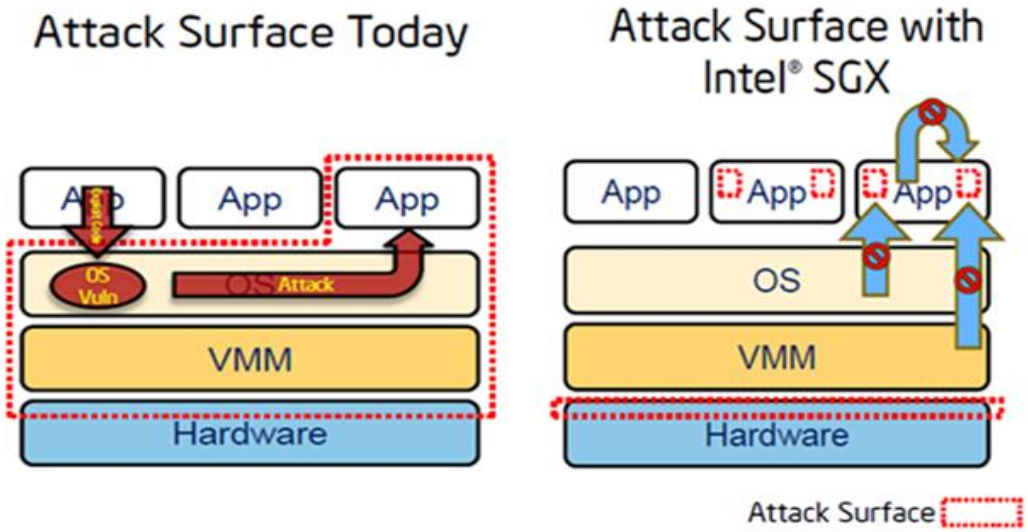
RA in Traditional systems (I): TPM

- **Hardware-based attestation** using a Trusted Platform Module (TPM)
- Secure crypto processor creates, stores, uses cryptographic keys
- Makes anonymous remote attestations of host status



RA in Traditional systems (II): SGX

- Hardware-based memory encryption that isolates specific application code and data in memory.
- Allows user-level code to allocate private regions of memory, called enclaves, which are designed to be protected from processes running at higher privilege levels.



Intel Software Guard Extensions. https://software.intel.com/en-us/sgx

Outline

- **Internet of Things Security**
  - Motivation
  - **Overview of Remote attestation**
  - History of Remote attestation

- **Remote attestation protocols**
  - Software-based attestation
  - Hybrid attestation
  - Swarm attestation
  - Dynamic attestation
  - Distributed services attestation

- **Research perspectives**
  - Recent research projects
  - Conclusions
  - Open challenges & Future works

Overview of Remote attestation

1. Challenge (Executed by Verifier)

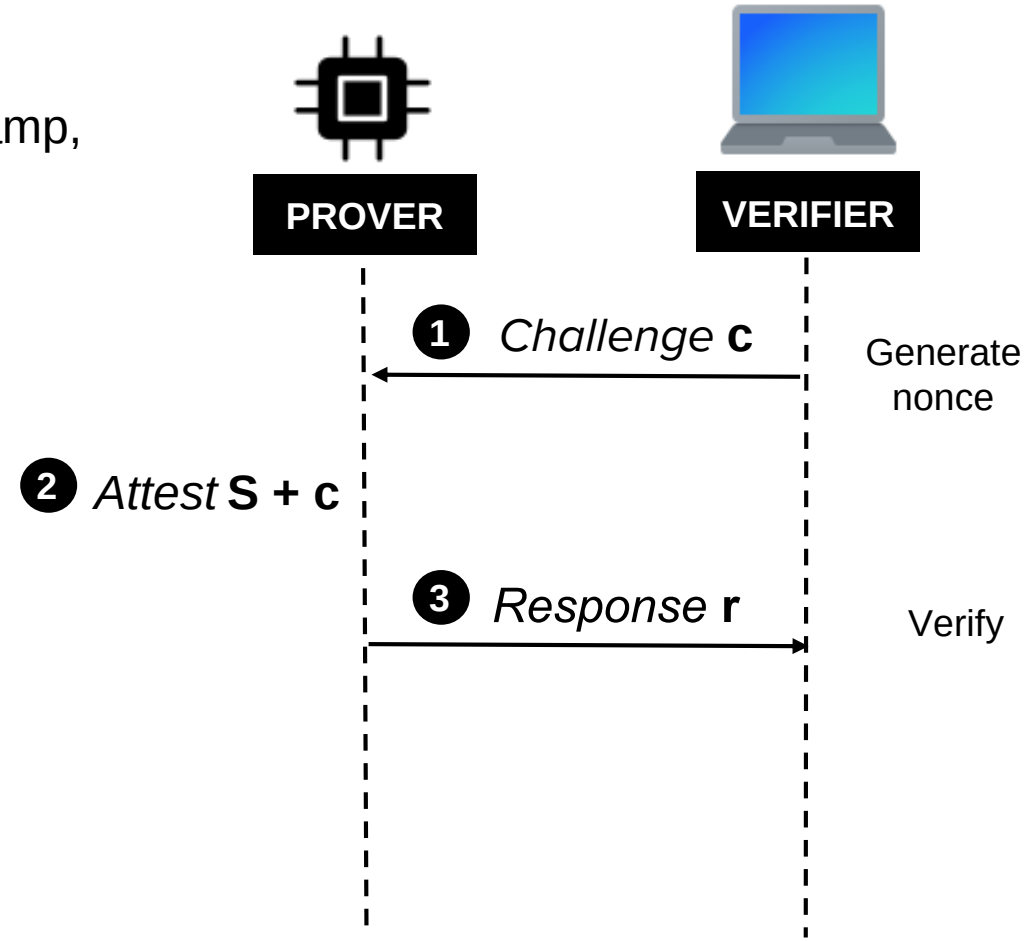
Outputs a random Challenge (nonce, timestamp, memory addresses, attestation routine)

2. Attest (Executed by Prover)

Computes a small attestation response based on internal state **S** (e.g., checksum over memory contents) and challenge **c**

3. Verify (Executed by Verifier)

Compares with the response received from Prover with the expected state



Typical adversary models

1. **Software Adversary**

- **Remote:** Infect device(s) with malware
- **Local:** Learn device secret, impersonate or clone, can launch side channel attack
- **Mobile adversary:** Relocates or deletes itself

2. **Hardware Adversary**

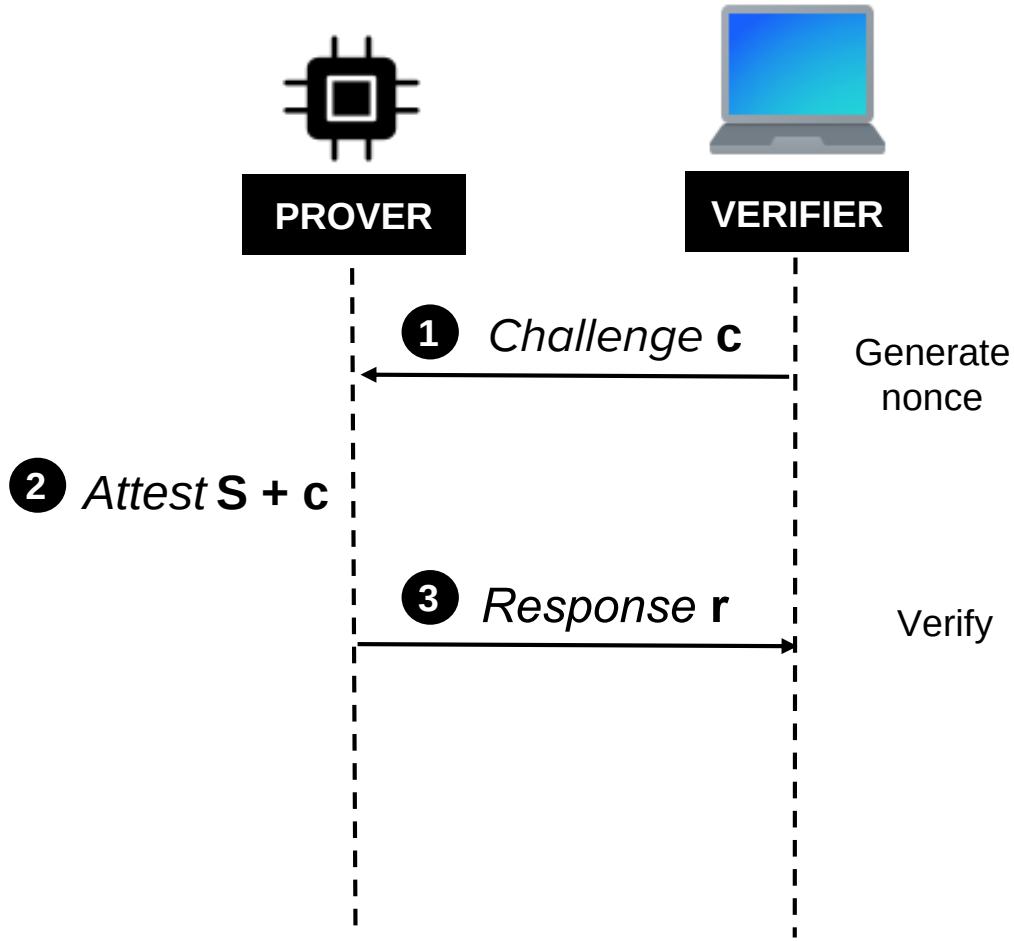
- **Stealthy Physical Intrusive:** Capture device and physically extract secrets, clone device(s)
- **Physical Intrusive:** Capture device and modify contents/components

Requirements of Remote attestation

1. Challenge (Executed by Verifier)
- Authentic, Fresh, Unpredictable

2. Attest (Executed by Prover)
- Authentic, Unforgeable, Dynamic, Deterministic

3. Verify (Executed by Verifier)
- Deterministic



Outline

- **Internet of Things Security**
  - Motivation
  - Overview of Remote attestation
  - **History of Remote attestation**

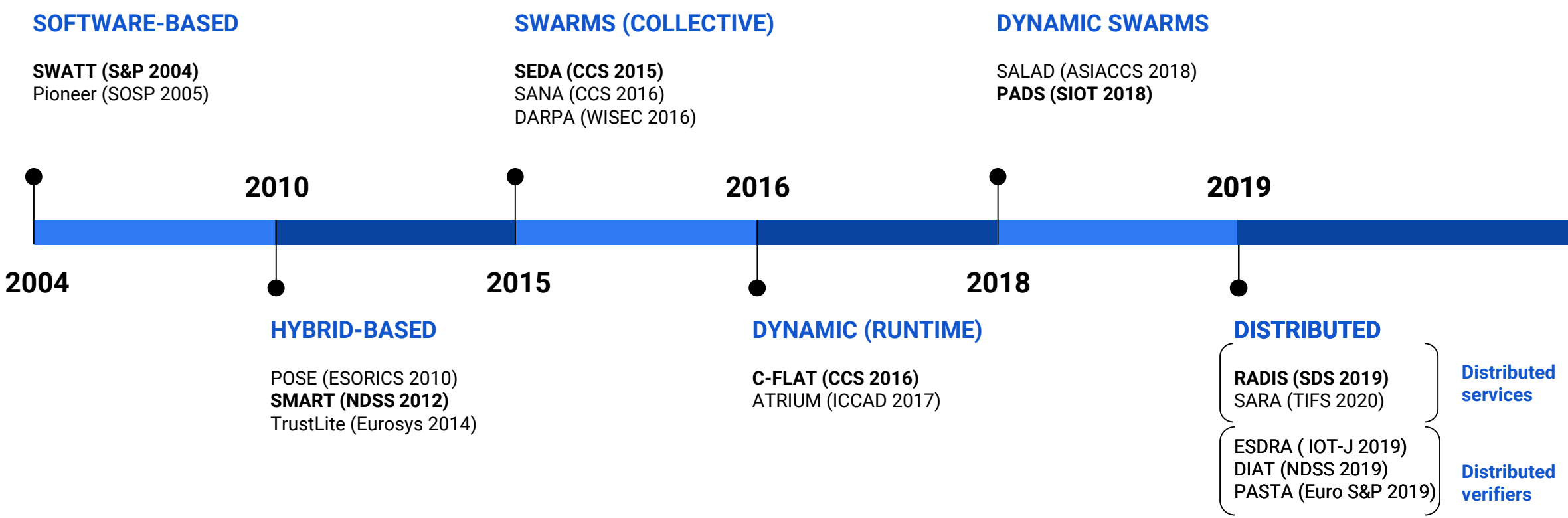
- **Remote attestation protocols**
  - Software-based attestation
  - Hybrid attestation
  - Swarm attestation
  - Dynamic attestation
  - Distributed services attestation

- **Research perspectives**
  - Recent research projects
  - Conclusions
  - Open challenges & Future works

Approaches of Remote attestation

- **Hardware design**
  Hardware-based**,** Software-based, or Hybrid
- **Memory**
  Static vs Dynamic attestation
- **Number of Device**
  Single Device vs Swarms (Collective)
- **Network Topology**
  Static vs Dynamic Swarms
- **Communication data**
  Swarms vs Distributed services

History of Remote attestation



* Protocols in **bold** will be presented in the following slides

Outline

- **Internet of Things Security**
  - Motivation
  - Overview of Remote attestation
  - History of Remote attestation

- **Remote attestation protocols**
  - **Software-based attestation**
  - Hybrid attestation
  - Swarm attestation
  - Dynamic attestation
  - Distributed services attestation

- **Research perspectives**
  - Recent research projects
  - Conclusions
  - Open challenges & Future works

Software-based attestation

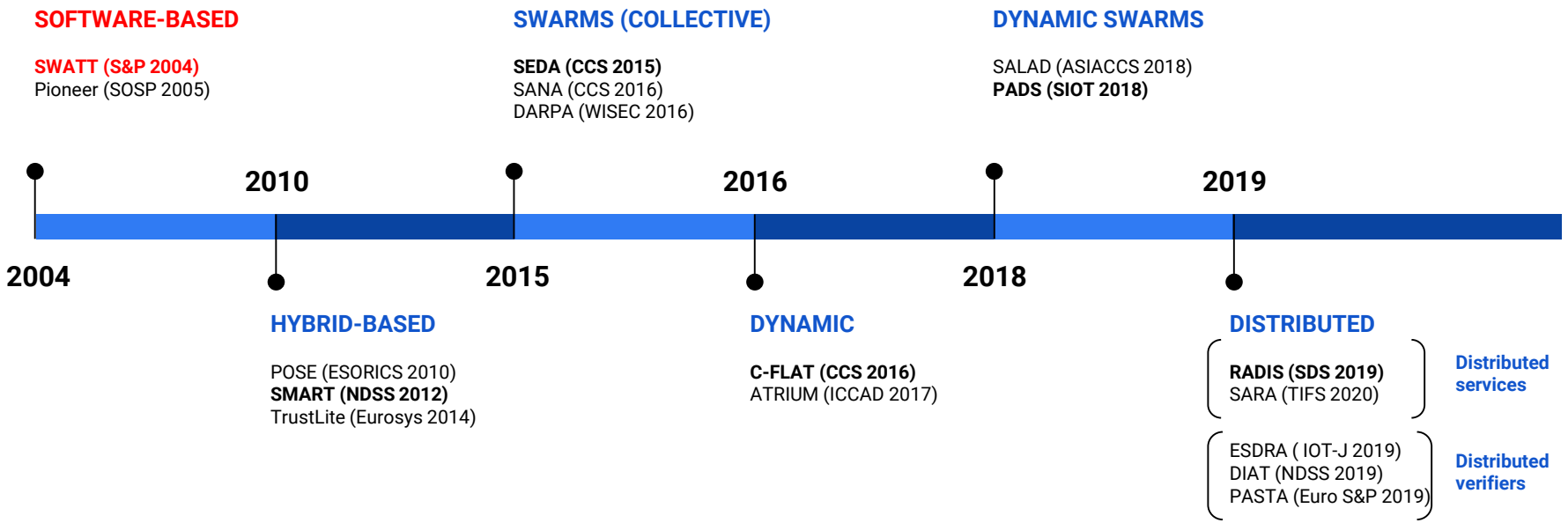
No hardware features to support attestation

- No secrets on Prover (e.g., no Attestation Key)

Relies on two pillars:

- Tight time constraints
- Lack of free space to store malicious code

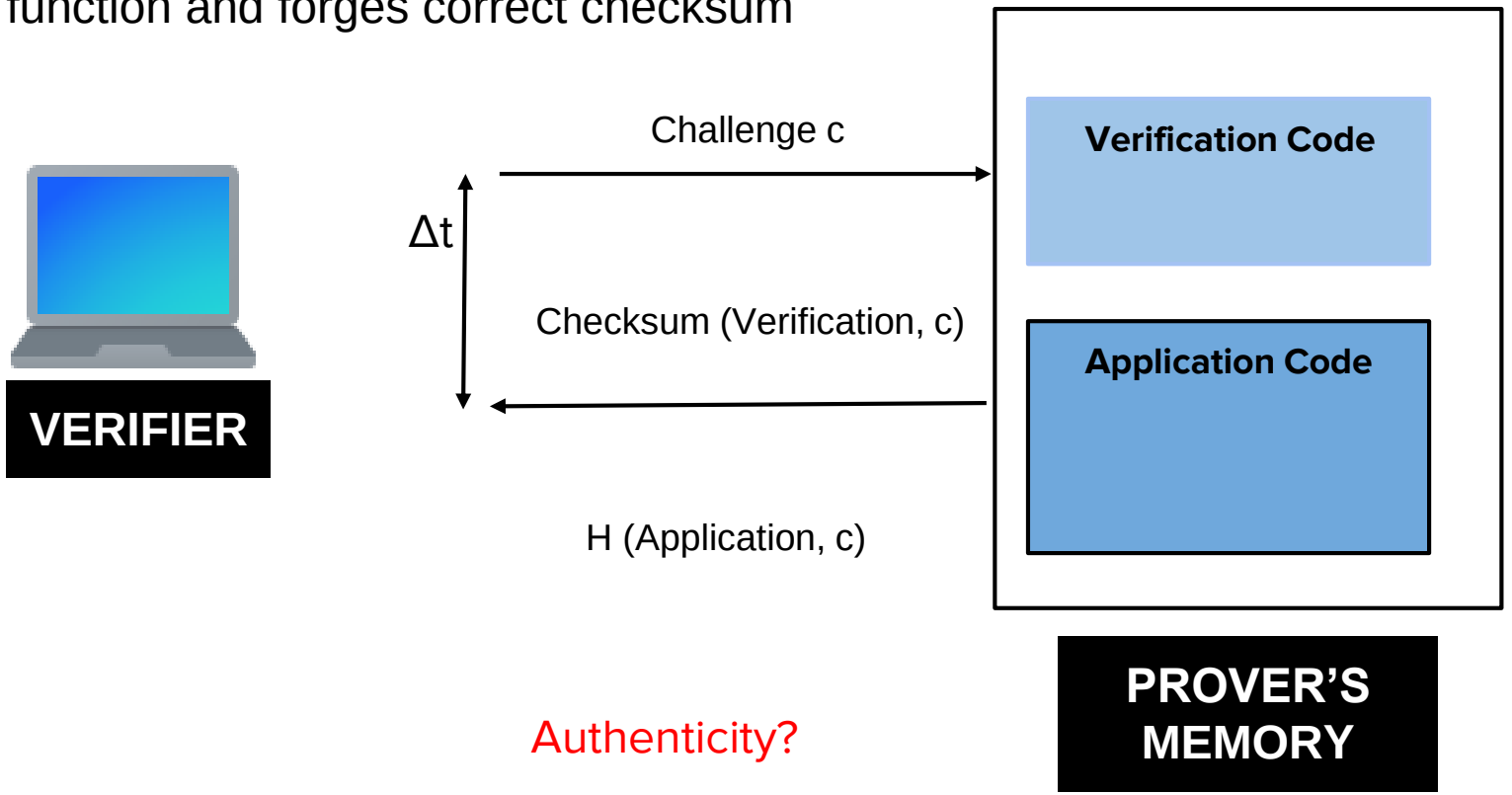
Software-based attestation



A. Seshadri, A. Perrig, L. van Doorn and P. Khosla, "SWATT: softWare-based attestation for embedded devices," IEEE Symposium on Security and Privacy, 2004. Proceedings. 2004, Berkeley, CA, USA, 2004, pp. 272-282

SWATT: softWare-based attestation for embedded devices

Intuition: Checksum is incorrect or checksum computation slows down if attacker modifies verification function and forges correct checksum



Attacks against Software-based attestation

Compression attack
1. Compress code to make space for attack code
2. Decompressed on-the-fly during attestation

Return-oriented rootkit
1. Install a rootkit that hides itself in non-executable memories
2. Use ROP (Return-Oriented Programming) to implement this attack

Castelluccia, C., Francillon, A., Perito, D., Soriente, C.: On the difficulty of software-based attestation of embedded devices. In: Proceedings of 16th ACM Conference on Computer and Communications Security (2009).

Summary of Software-based attestation

Advantages:

- No hardware requirements

Limitations:

- Verifier must know exact hardware configuration
- Difficult to prove time optimality
- Assumes “adversarial silence” during attestation
- Limited to “one-hop” networks

Outline

- **Internet of Things Security**
  - Motivation
  - Overview of Remote attestation
  - History of Remote attestation

- **Remote attestation protocols**
  - Software-based attestation
  - **Hybrid attestation**
  - Swarm attestation
  - Dynamic attestation
  - Distributed services attestation

- **Research perspectives**
  - Recent research projects
  - Conclusions
  - Open challenges & Future works

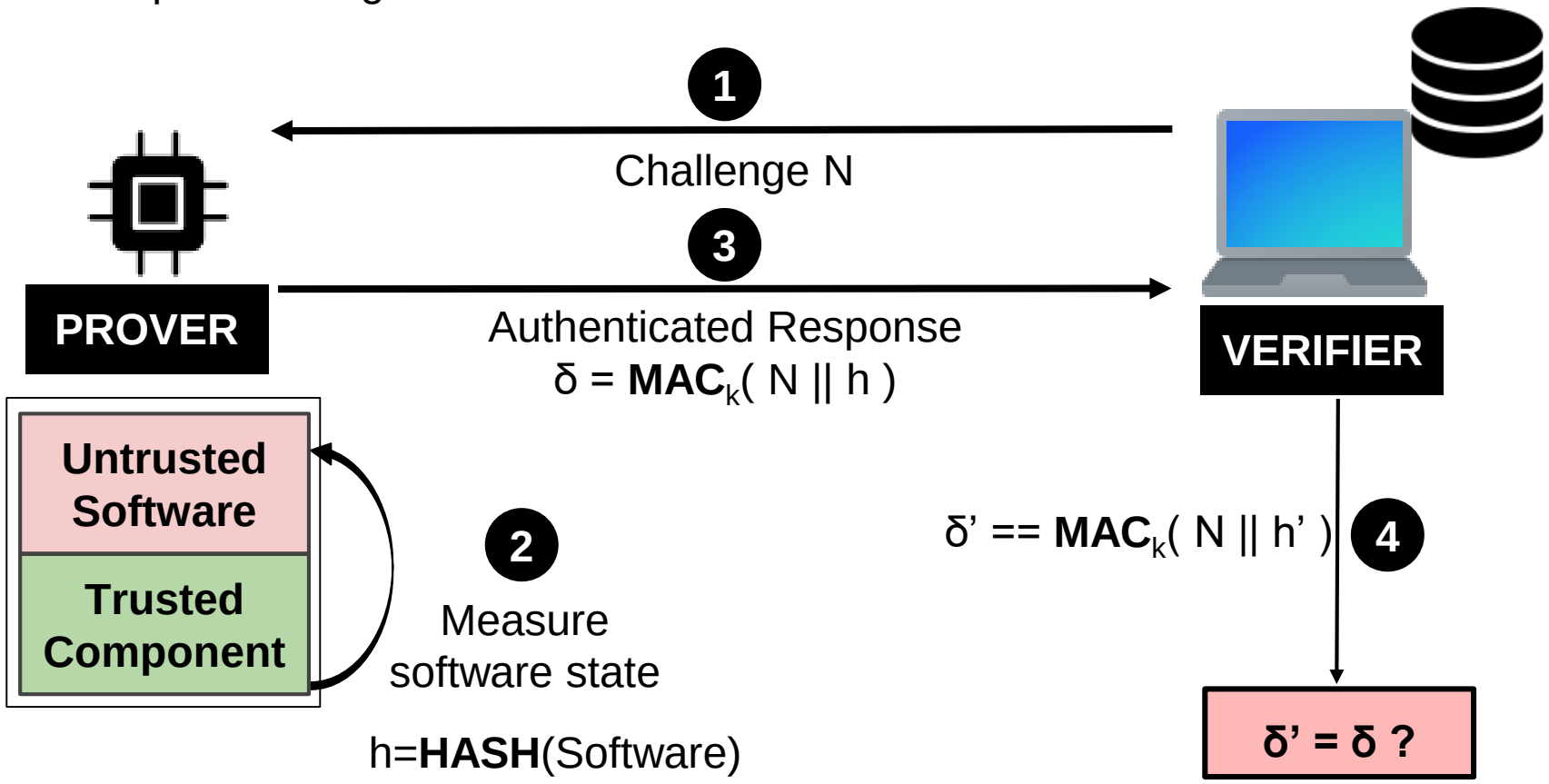
Hybrid attestation

- Minimal trust anchors: small changes to hardware

- Read-only Verification code, secure key storage and atomicity of execution of Verification code

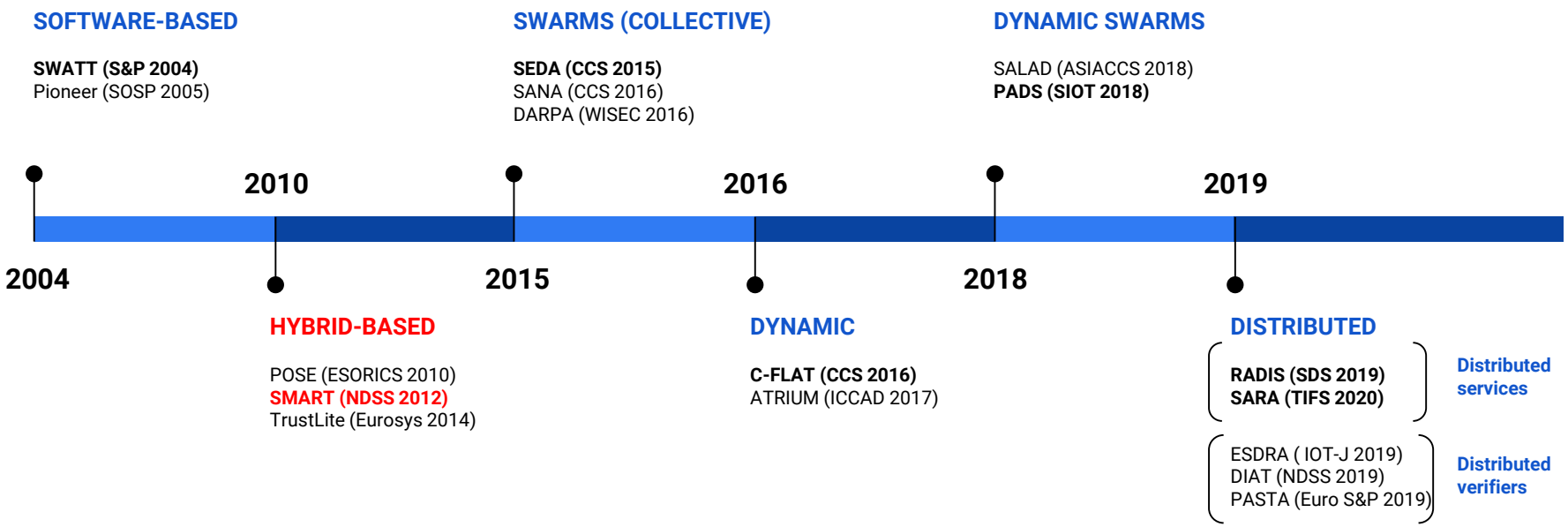
Hybrid attestation (Typical RA paradigm)

- Prover and Verifier share a key *k*
- Verifier expects configuration *h'*



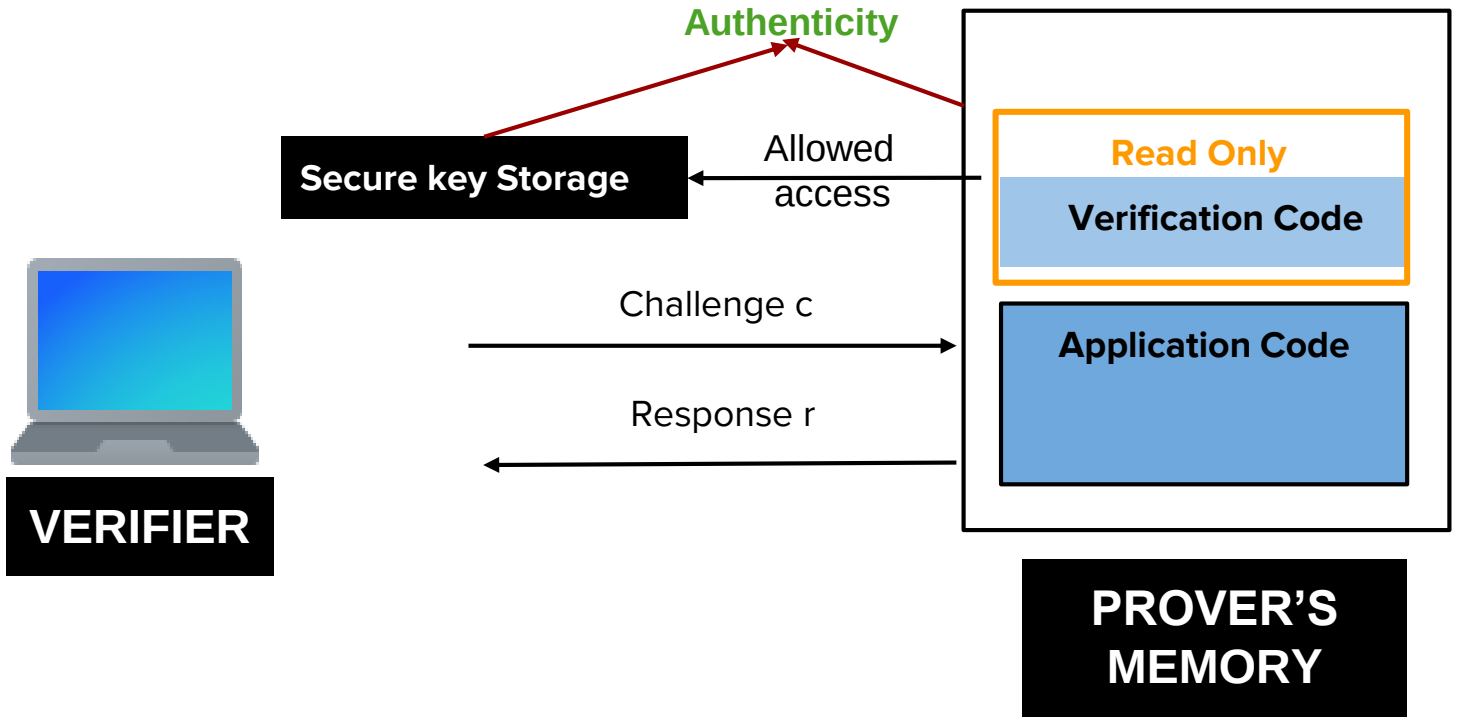
If there is a match, confirm the trustworthy state

Hybrid attestation



Eldefrawy, K., Tsudik, G., Francillon, A., and Perito, D.SMART:Secure and Minimal Architecture for (Establishing Dynamic) Root of Trust. In Proceedings of the 19th Annual Network & Distributed System Security Symposium NDSS '12.(2012).

SMART: Secure and Minimal Architecture



Summary of Hybrid attestation

Advantages

- Can be used across a network / over an untrusted channel
- Verifier does not need to know exact hardware configuration of the Prover

Disadvantages

- Needs additional hardware support

Outline

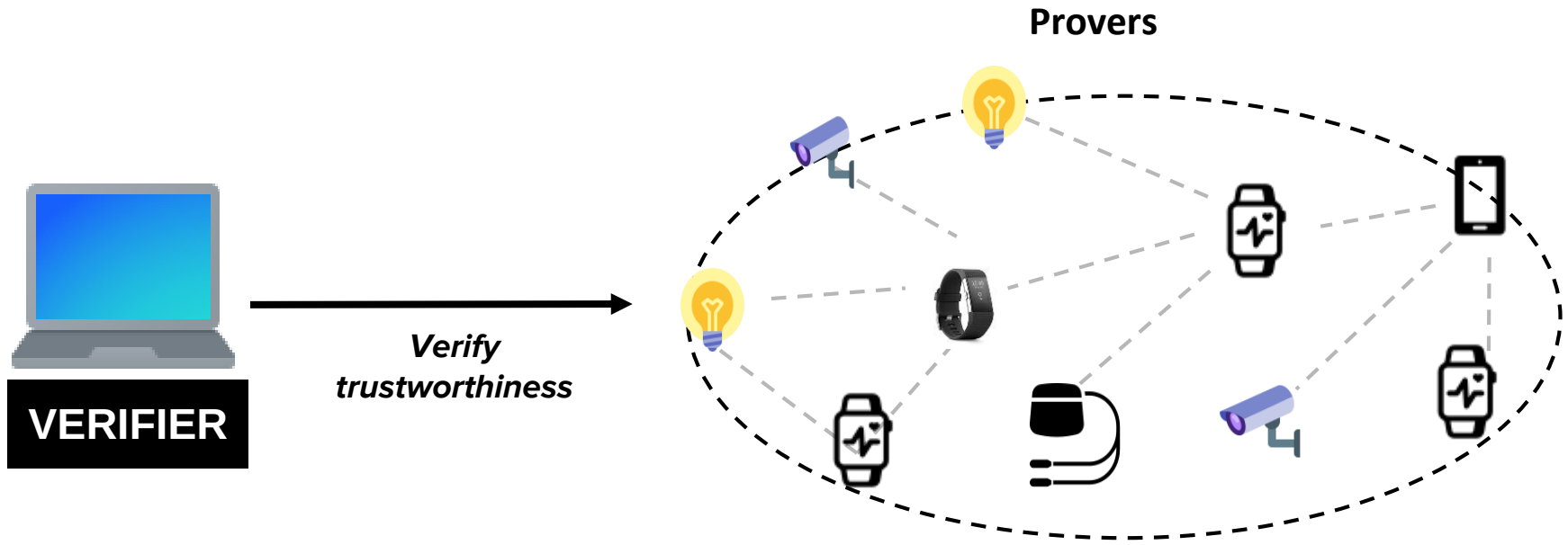
- **Internet of Things Security**
  - Motivation
  - Overview of Remote attestation
  - History of Remote attestation

- **Remote attestation protocols**
  - Software-based attestation
  - Hybrid attestation
  - **Swarm attestation**
  - Dynamic attestation
  - Distributed services attestation

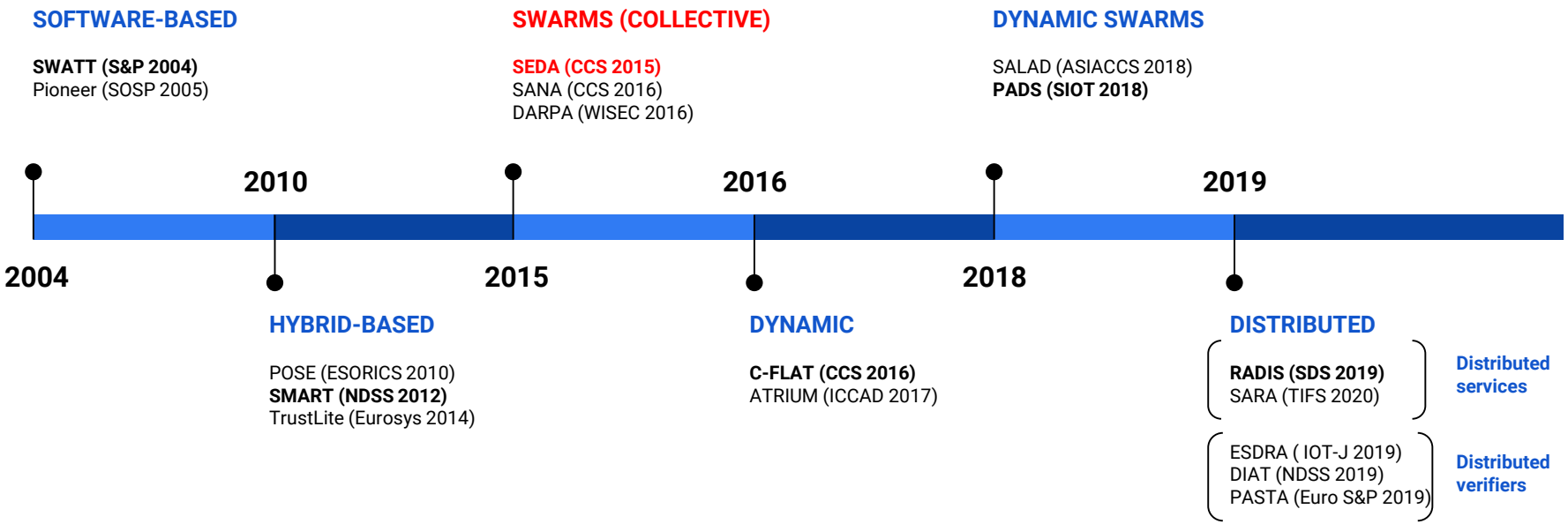
- **Research perspectives**
  - Conclusions
  - Challenges
  - Future works

Swarm attestation (Collective)

- Verify the internal state of a large group of devices
- Should be more efficient than attesting each node individually



Swarm attestation

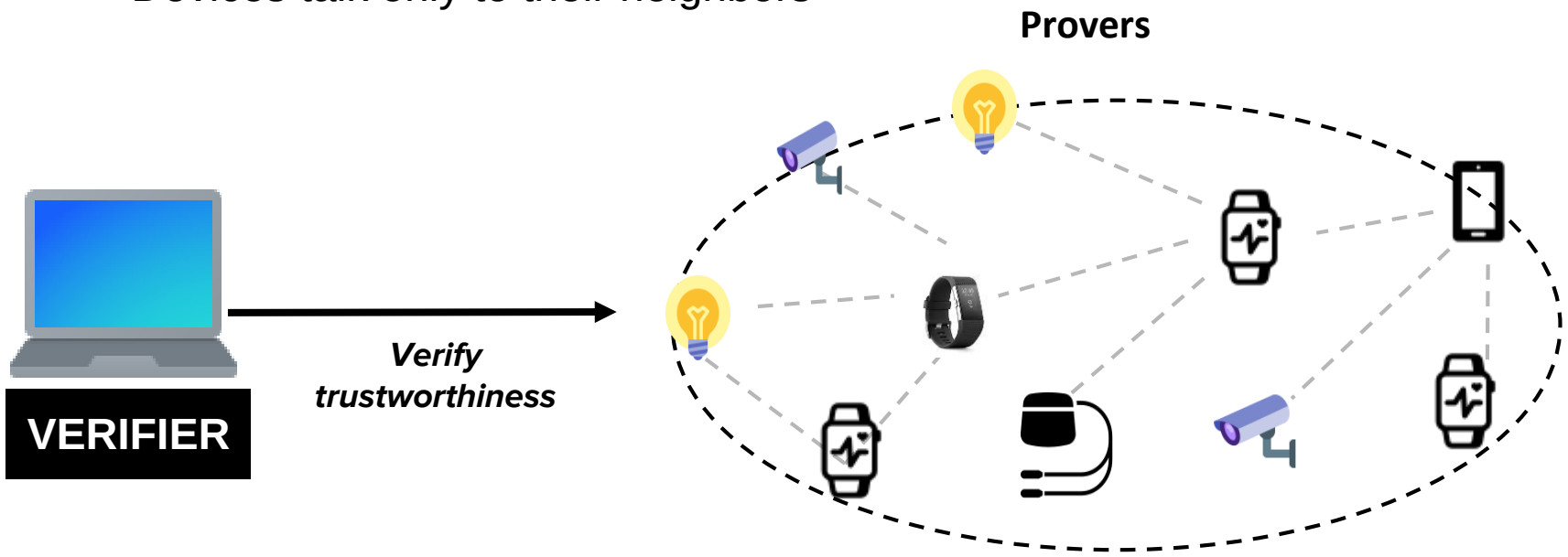


Asokan, N., Brasser, F., Ibrahim, A., Sadeghi, A.R., Schunter, M., Tsudik, G.,Wachsmann, C.: Seda: Scalable embedded device attestation. CCS '15, New York, NY, USA, ACM (2015)

SEDA: Scalable Embedded Device Attestation

System Model and Assumptions

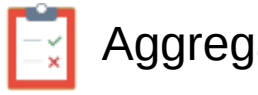
- ALL devices equipped with a trusted component (implementation based on SMART and TrustLite security architectures)
- Devices talk only to their neighbors



SEDA: Scalable Embedded Device Attestation

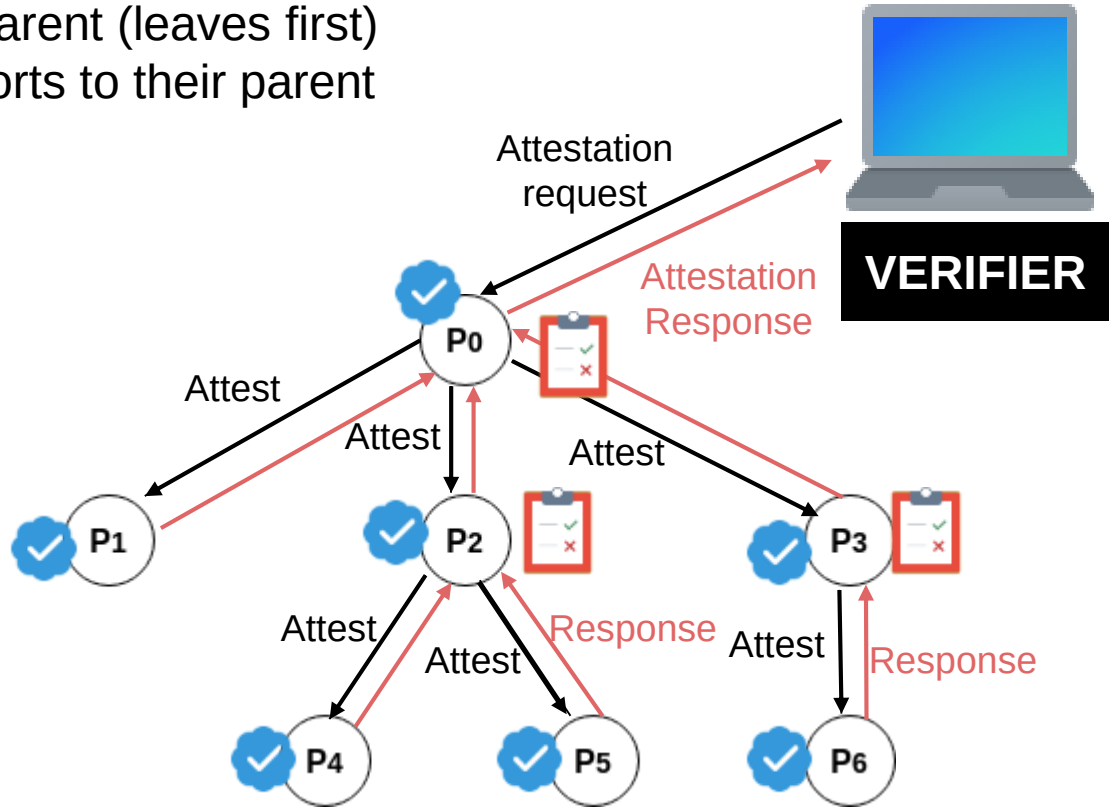
Algorithm logic:

1. Verifier selects random Prover (P_0) initializes attestation
2. Spanning tree is created rooted at P_0
3. Each Prover (device) gets attested by its parent (leaves first)
4. Sub-tree roots accumulate results and reports to their parent
5. P_0 reports overall result to Verifier



Aggregator

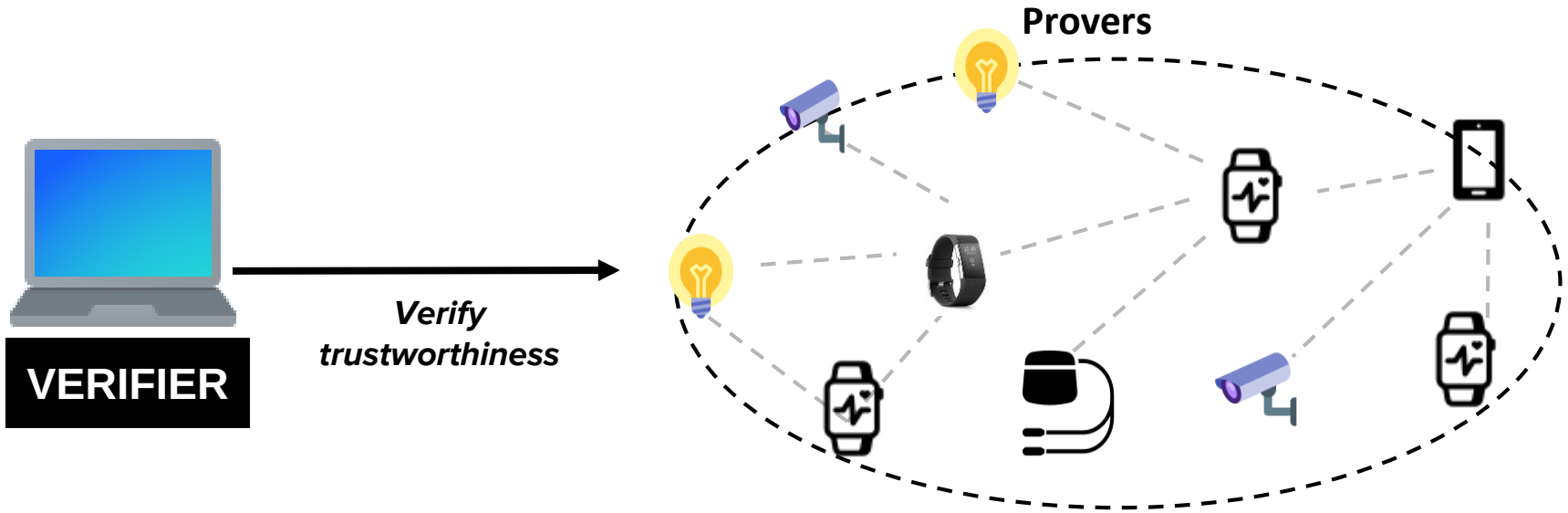
Attested Prover



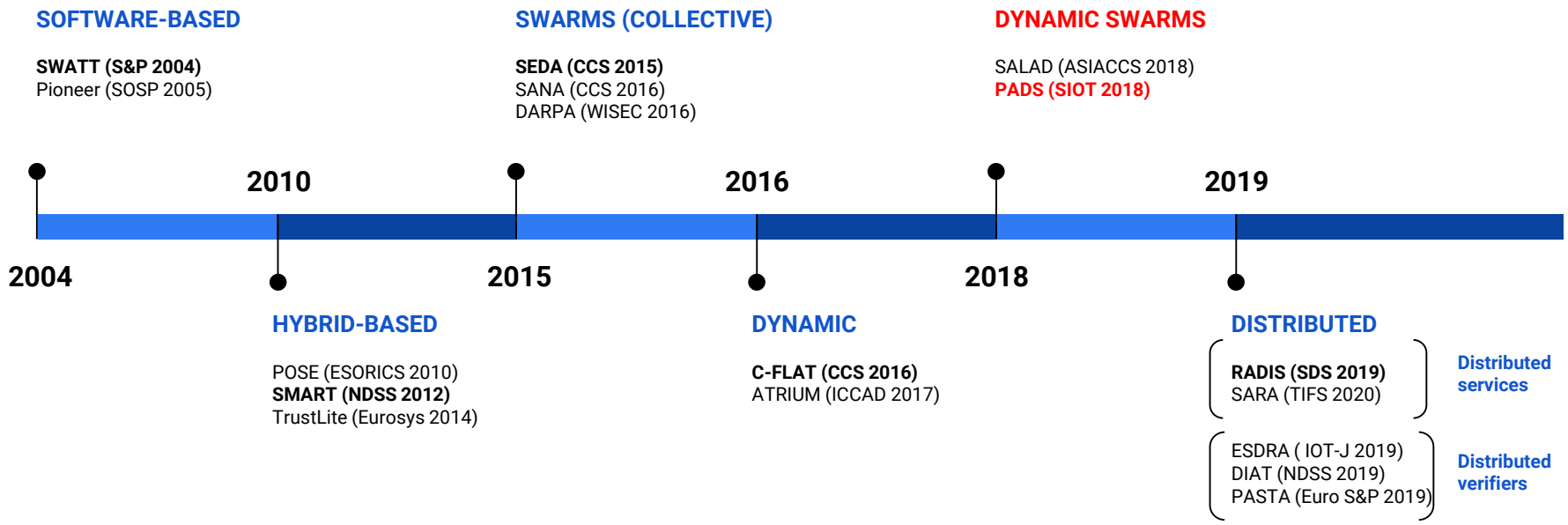
SEDA: Scalable Embedded Device Attestation

Limitations

- Lack of flexibility (ALL devices must participate to attestation), final result is boolean
- Aggregators should be trusted, single point of failure
- Network topology is static



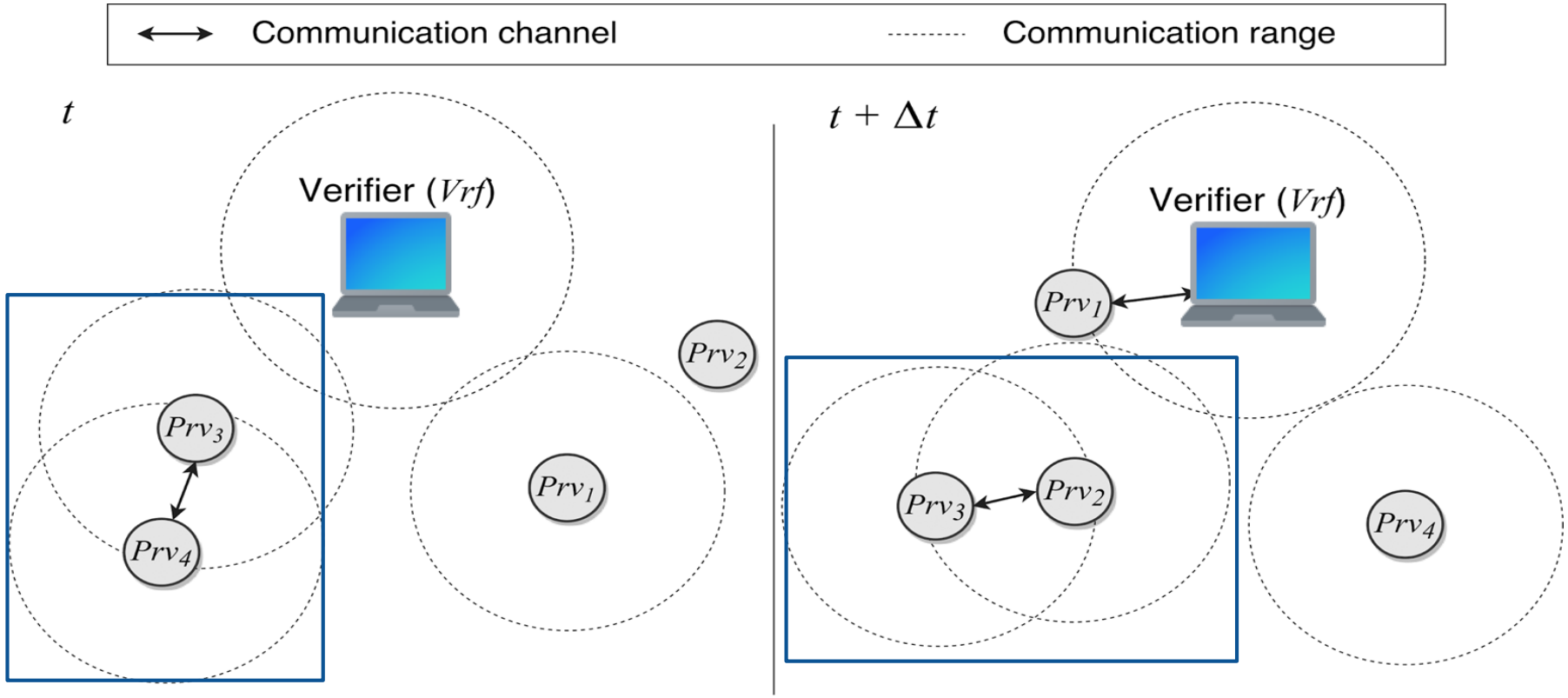
Dynamic swarms



Ambrosin, M., Conti, M., Lazzeretti, R., Masoom Rabbani, M.,and Ranise, S.PADS: Practical Attestation for Highly Dynamic Swarm Topologies. ArXiv e-prints (2018).

PADS: Attestation for Highly Dynamic Swarm

- Heterogeneous mobile devices, devices interact without forming spanning tree
- Use of "Consensus" among devices to corroborate attestation result



PADS: System model

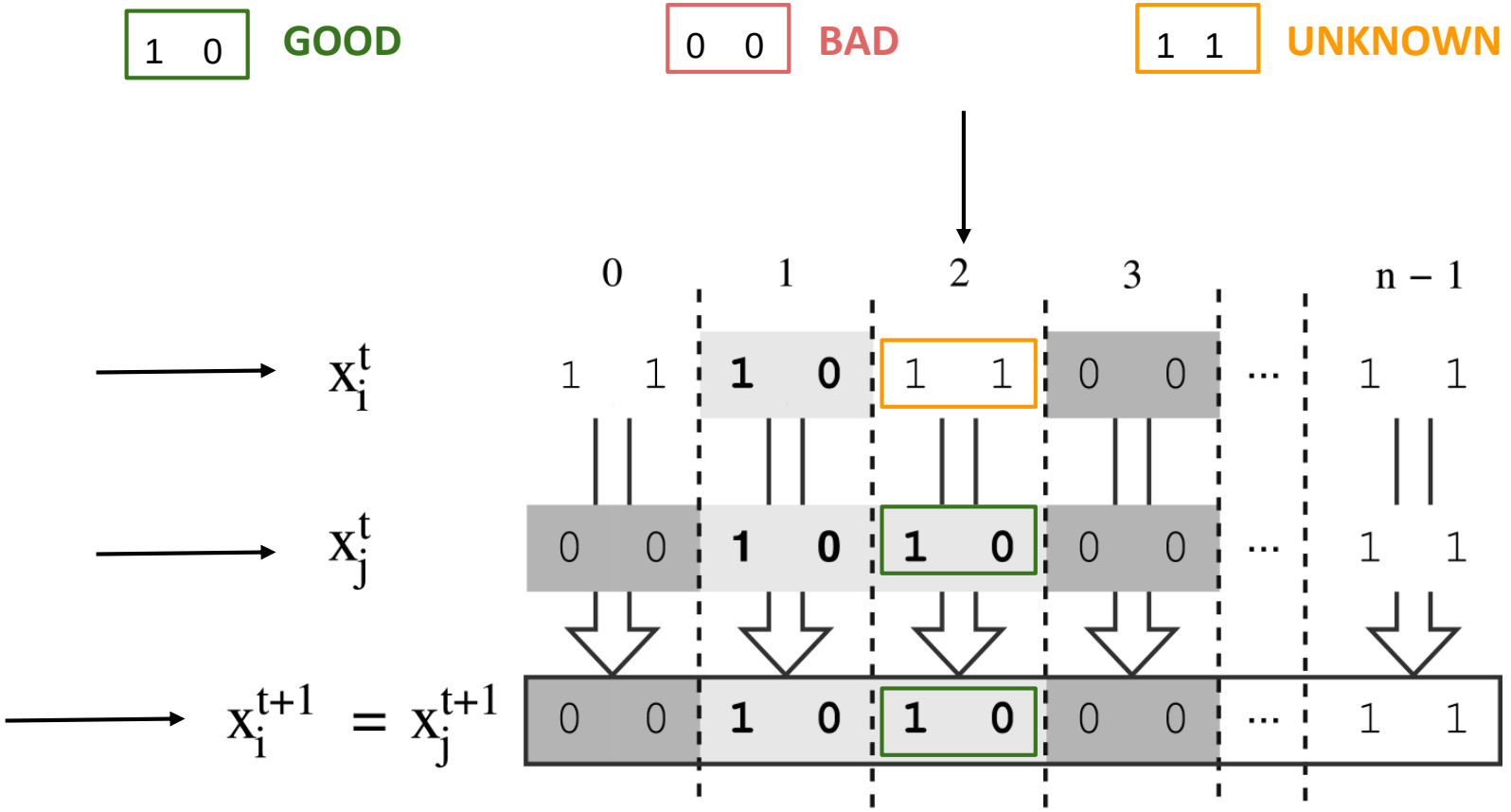
- Only **Provers** (P_j) require a Trusted Execution Environment (TEE)
  - P_j builds an *attestation proof*
    - Contains hash value of the underlying software
    - *Consists of three states (Good-10; Bad-00; Unknown-11)*
    - *Every prover will share its knowledge with other nodes in range*

- **Verifier**
  - Attest individual node before getting its knowledge about the network

- Physically compromised **Provers** can evade detection

PADS: Consensus concept

- Two distinct devices (X_i and X_j) will share there MAC-ed observation for time t
- Consensus among 2 devices will be like



Summary of Dynamic Swarms

- **Advantages**
  - Suitable for dynamic networks
  - Consider device movement during attestation
  - Verifier can have the snapshot of the network at run-time

- **Disadvantages**
  - Complexity of the protocol in terms of both communication and required processing for resource-constrained devices
  - Do not consider the communication data exchanged among devices

Outline

- **Internet of Things Security**
  - Motivation
  - Overview of Remote attestation
  - History of Remote attestation

- **Remote attestation protocols**
  - Software-based attestation
  - Hybrid attestation
  - Swarm attestation
  - **Dynamic attestation**
  - Distributed services attestation

- **Research perspectives**
  - Recent research projects
  - Conclusions
  - Open challenges & Future works

Dynamic attestation

Program Memory Attestation schemes

do not

address runtime attacks

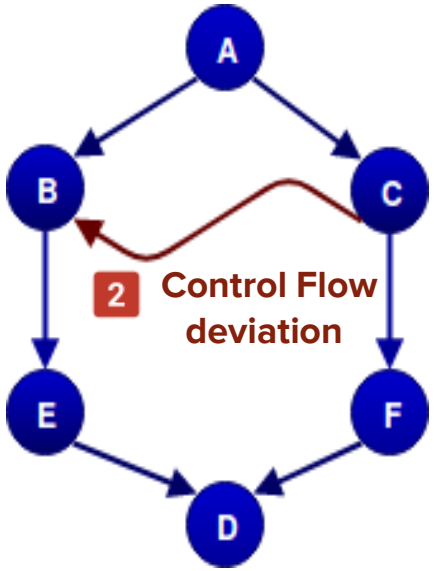
Code reuse attack

Adversary

1 Exploit
Buffer Overflow

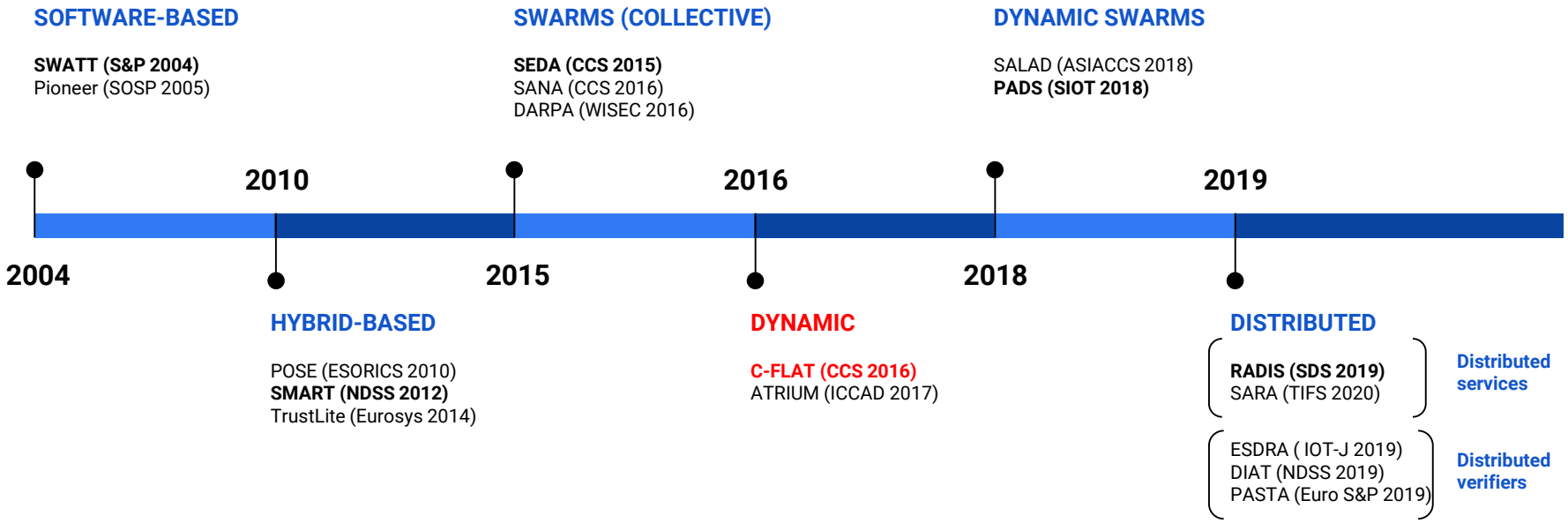
```
A: if(auth==true)
B: then: call privileged()
C: else: call unprivileged()
D: terminate
...
E: privileged{...instructions...}
F: unprivileged{...instructions...}
```

Pseudo-code



Control-flow Graph (CFG)

Dynamic attestation



Abera, T., Asokan, N., Davi, L., Ekberg, J.-E., Nyman, T., Paverd,A., Sadeghi, A.-R., and Tsudik, G.C-FLAT: Control-Flow Attestation for Embedded Systems Software. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security CCS '16.(2016).

C-FLAT: Control-flow attestation

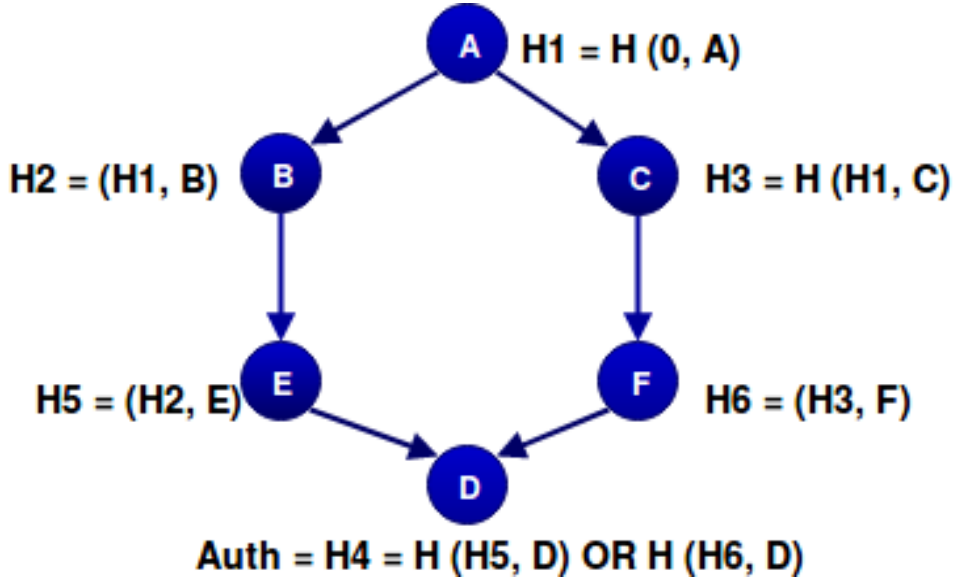
- Proposes a complete attestation of the run-time state of the Prover

- A single hash value that represents the entire control flow of the Prover's state

[Abera et al. C-FLAT, CCS 2016]

C-FLAT: Control-flow attestation

Cumulative Hash Value: H i = H (H i-1 , N)
H i-1 -- previous hash result
N -- instruction block (node) just executed



[Abera et al. C-FLAT, CCS 2016]

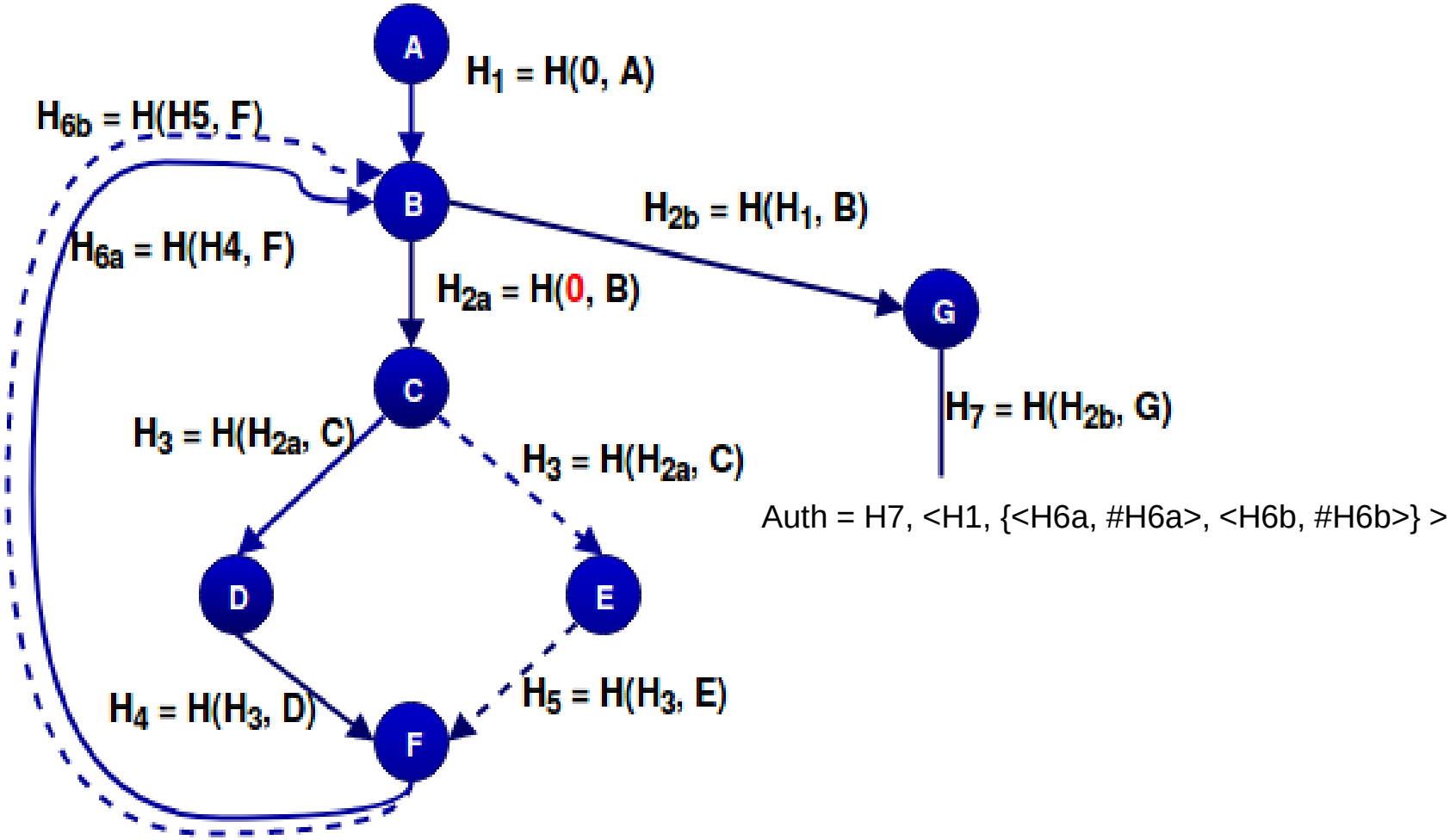
Loops are a challenge!

Different loop paths
and loop iterations lead to many valid hash values

C-FLAT Approach:

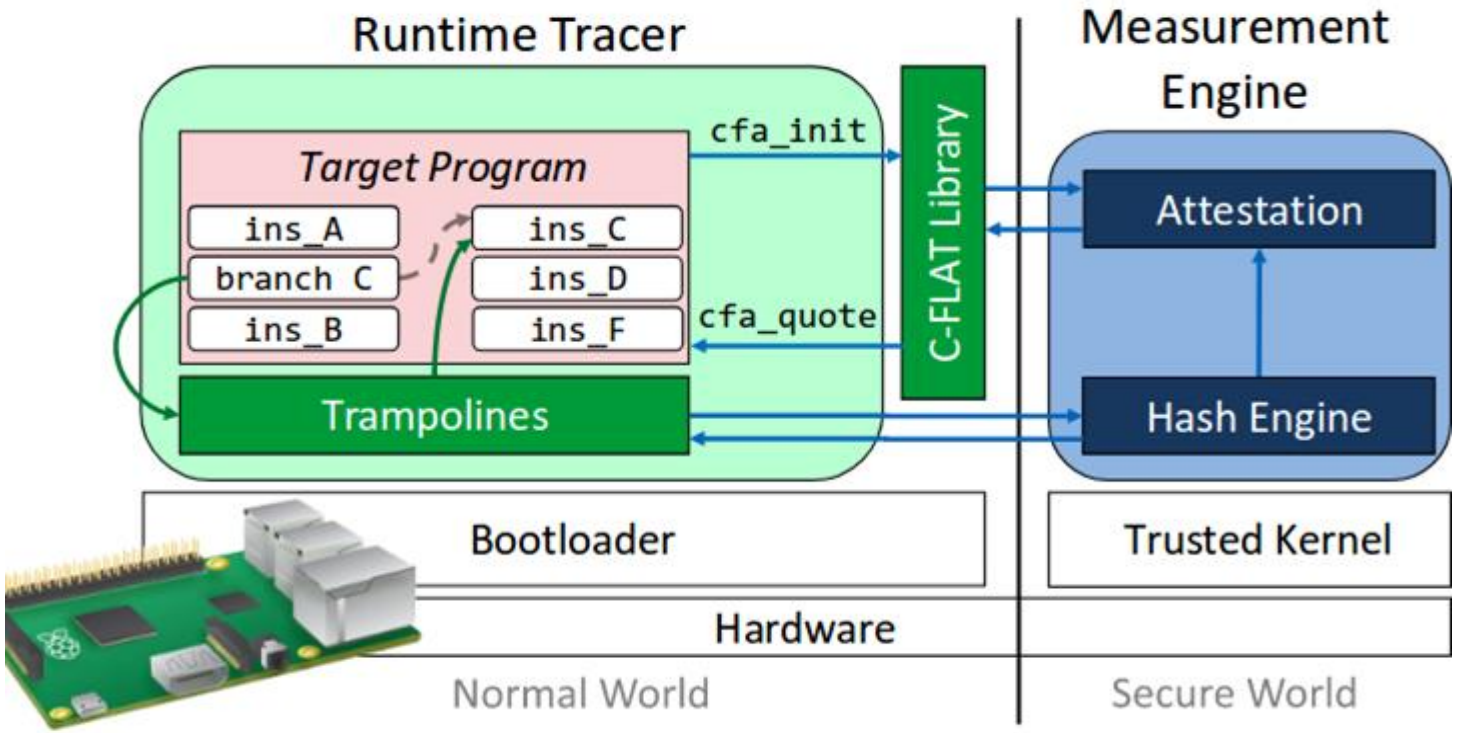
Treat loops as sub-graphs
and report their hash values
and # of iterations separately

C-FLAT approach



Auth = H7, <H1, {<H6a, #H6a>, <H6b, #H6b>} >

C-FLAT implementation



Summary of Dynamic attestation

Advantages

- Better dection level: Detects runtime attacks

Disadvantages

- The protocols rely on customized hardware support
- The computations are not efficient

Outline

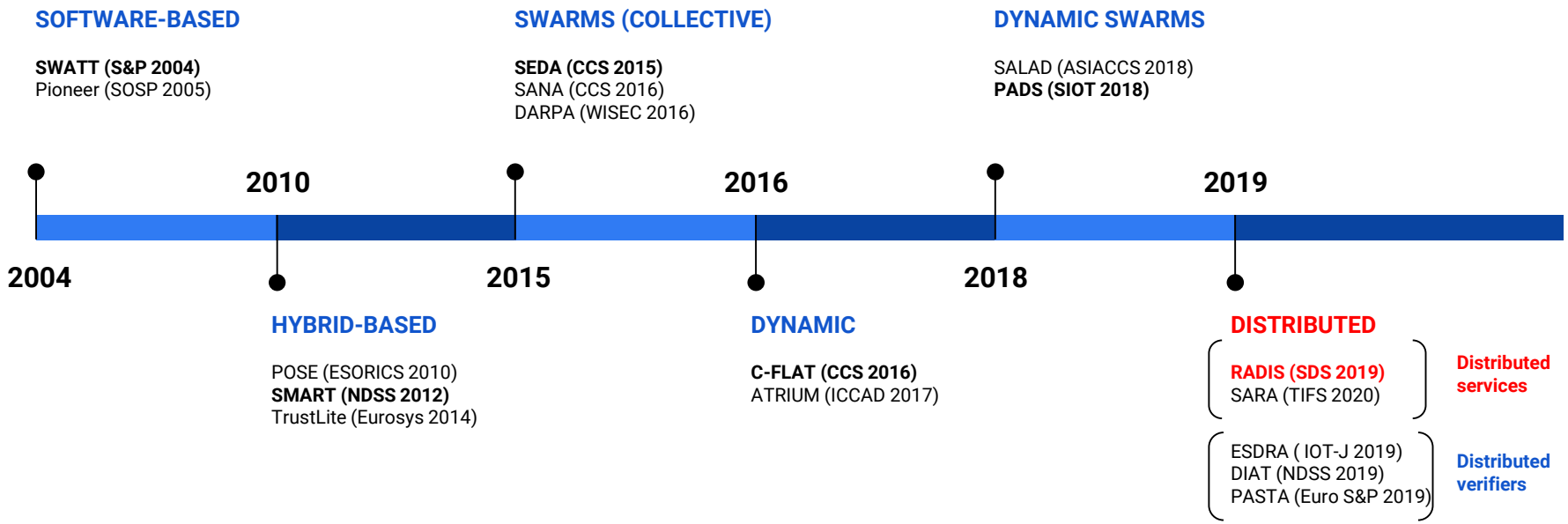
- **Internet of Things Security**
  - Motivation
  - Overview of Remote attestation
  - History of Remote attestation

- **Remote attestation protocols**
  - Software-based attestation
  - Hybrid attestation
  - Swarm attestation
  - Dynamic attestation
  - **Distributed services attestation**

- **Research perspectives**
  - Recent research projects
  - Conclusions
  - Open challenges & Future works

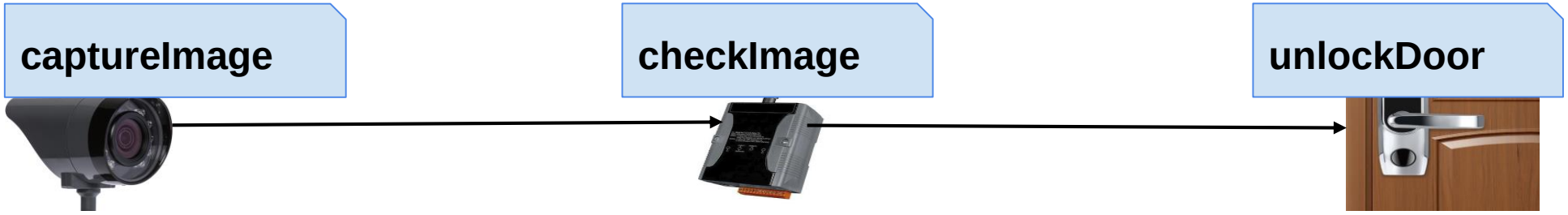
Remote attestation of Distributed Services

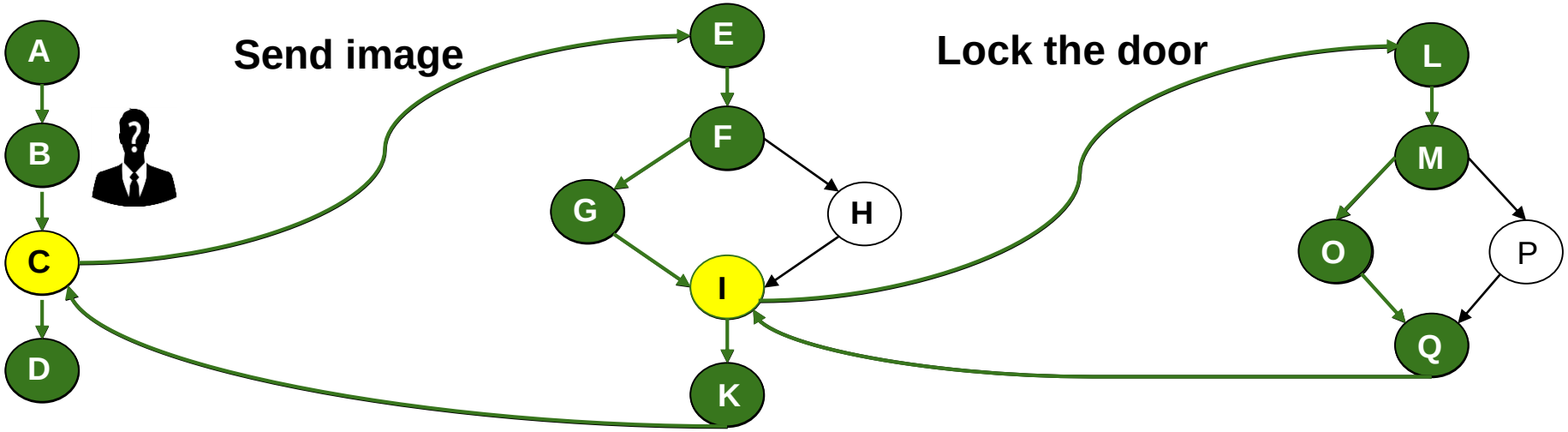
Introduces the service perspective in remote attestation



Conti, M., Dushku, E., and Mancini, L. V. RADIS: Remote Attestation of Distributed IoT Services. In 6th IEEE International Conference on Software Defined Systems, SDS 2019(2019), pp. 25–32.

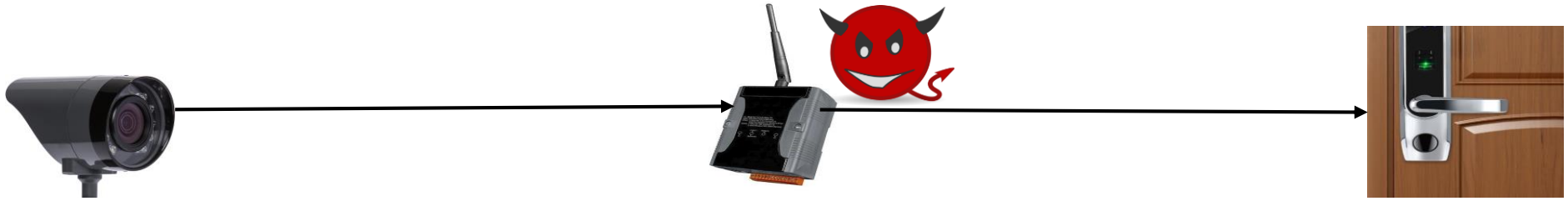
Motivating example: Distributed IoT service (Sync)



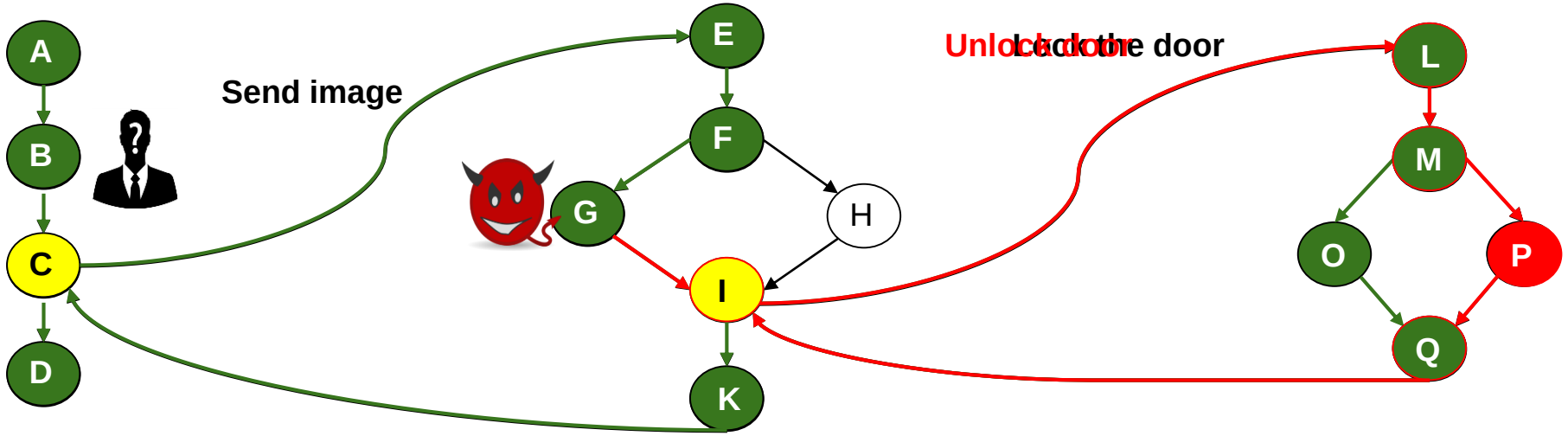


Motivating example: Distributed IoT service (Sync)

Non-Control-Data Attack
modify variable's value/ corrupt data pointer

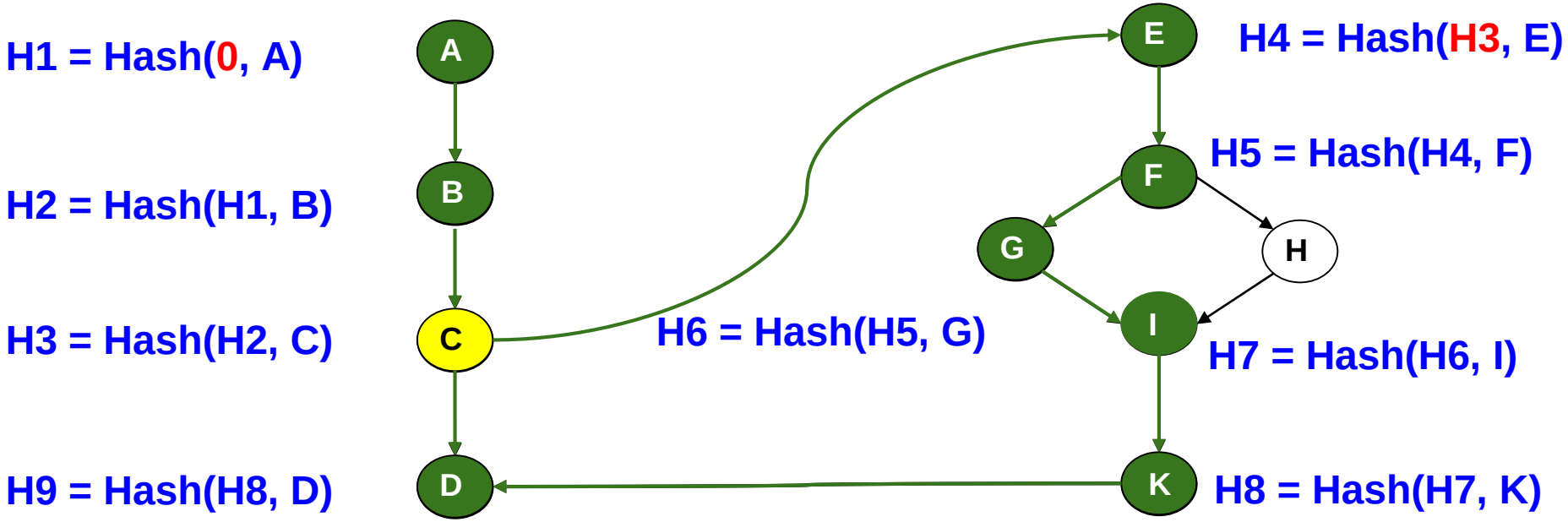


Outdoor camera **Security Monitor** **Smart door**

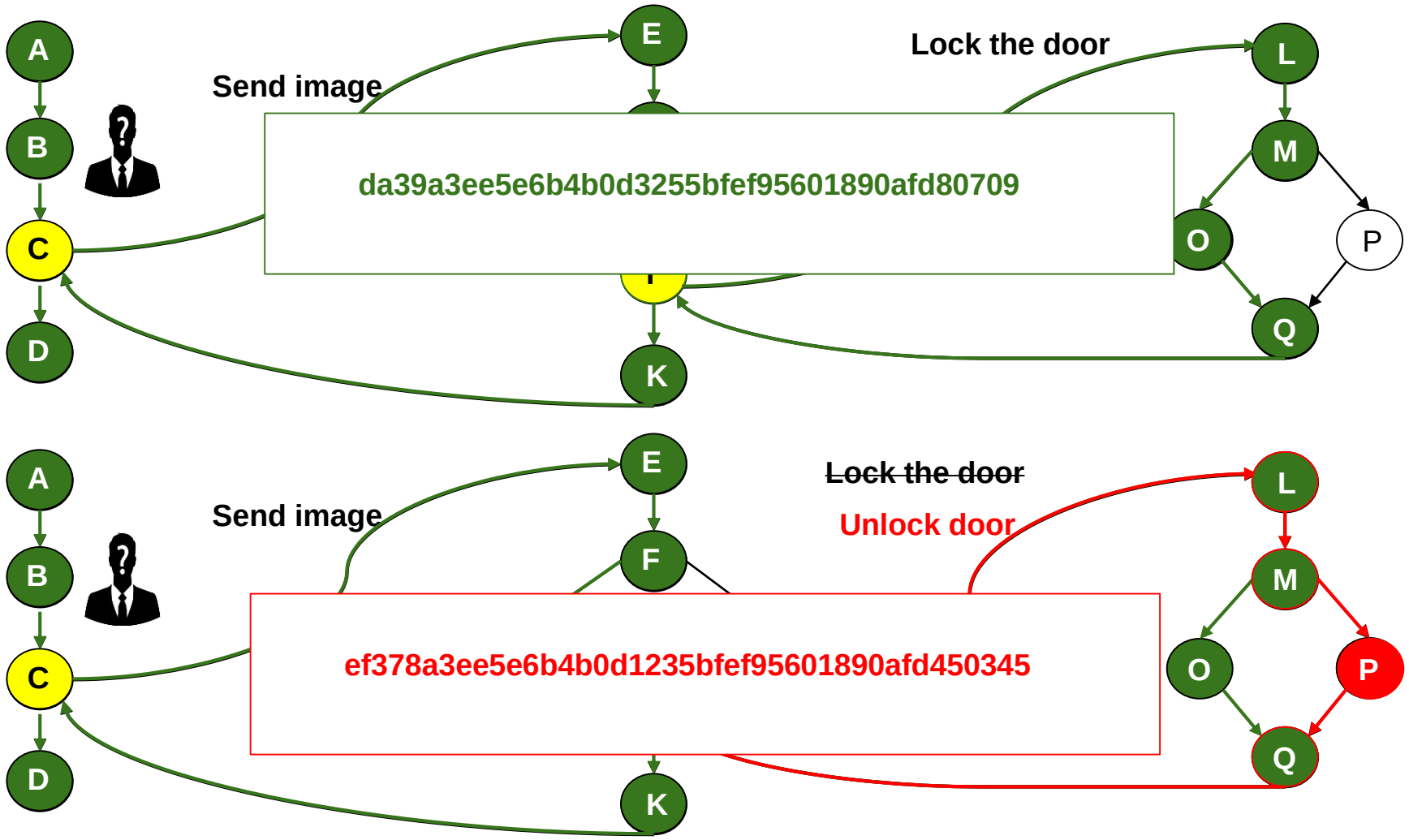


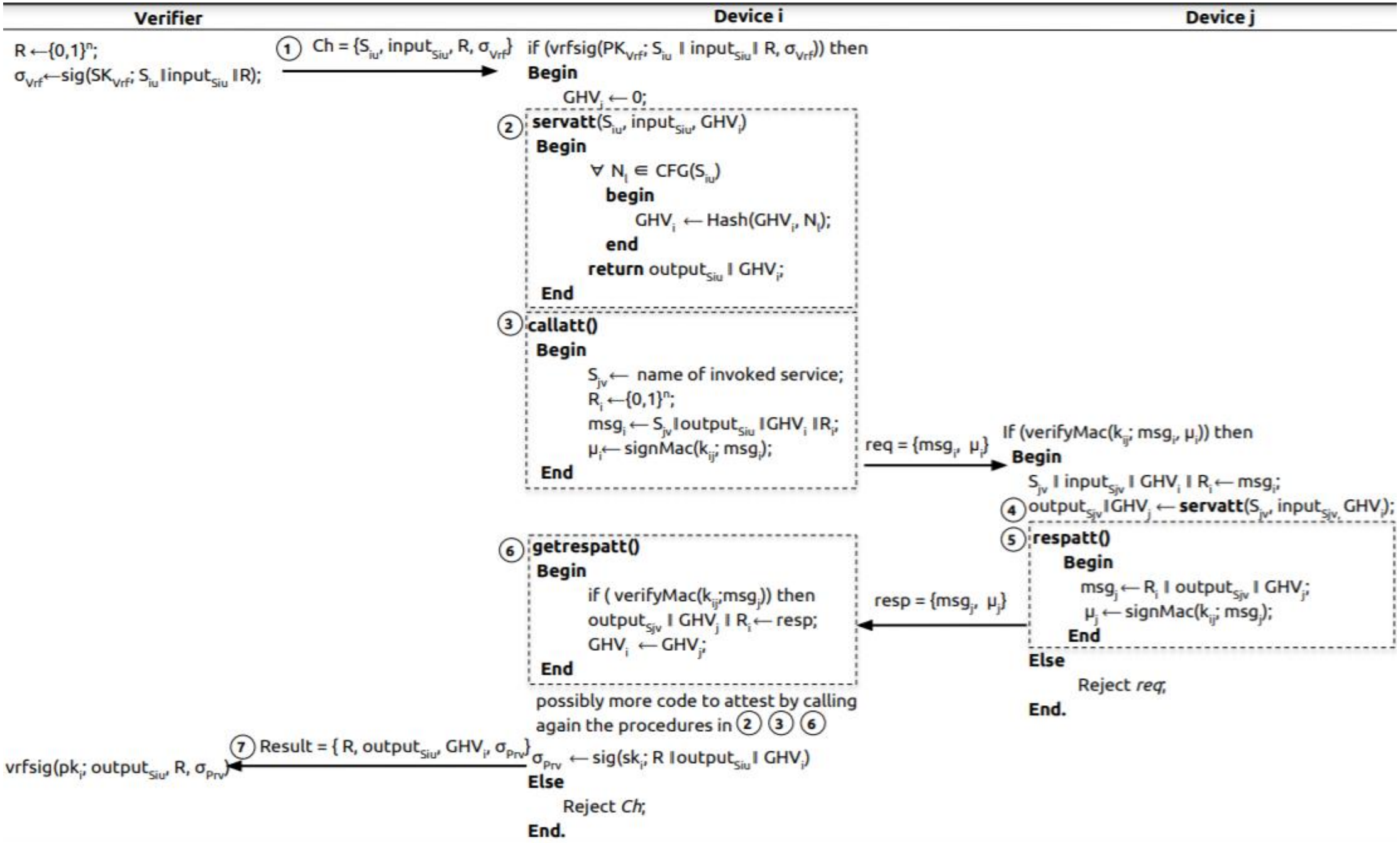
RADIS: Overview

$$H_i = Hash(H_{i-1}, N_{current})$$



Why it works?


A
B
C
D
E
K
L
M
O
P
Q
Send image
Lock the door
da39a3ee5e6b4b0d3255bfef95601890afd80709
A
B
C
D
E
F
K
L
M
O
P
Q
Send image
~~Lock the door~~
Unlock door
ef378a3ee5e6b4b0d1235bfef95601890afd450345


Verifier
Device i
Device j
R ←{0,1}^n;
σ_Vrf ← sig(SK_Vrf; S_iu ‖ input_Siu ‖ R);
1 Ch = {S_iu, input_Siu, R, σ_Vrf}
if (vrfsig(PK_Vrf; S_iu ‖ input_Siu ‖ R, σ_Vrf)) then
Begin
GHV_i ← 0;
2 servatt(S_iu, input_Siu, GHV_i)
Begin
∀ N_l ∈ CFG(S_iu)
begin
GHV_i ← Hash(GHV_i, N_l);
end
return output_Siu ‖ GHV_i;
End
3 callatt()
Begin
S_jv ← name of invoked service;
R_i ←{0,1}^n;
msg_i ← S_jv ‖ output_Siu ‖ GHV_i ‖ R_i;
μ_i ← signMac(k_ij; msg_i);
End
req = {msg_i, μ_i}
If (verifyMac(k_ij; msg_i, μ_i)) then
Begin
S_jv ‖ input_Sjv ‖ GHV_i ‖ R_i ← msg_i;
4 output_Sjv ‖ GHV_j ← servatt(S_jv, input_Sjv, GHV_j);
5 respatt()
Begin
msg_j ← R_i ‖ output_Sjv ‖ GHV_j;
μ_j ← signMac(k_ij; msg_j);
End
Else
Reject req;
End.
resp = {msg_j, μ_j}
6 getrespatt()
Begin
if (verifyMac(k_ij;msg_j)) then
output_Sjv ‖ GHV_j ‖ R_i ← resp;
GHV_i ← GHV_j;
End
possibly more code to attest by calling again the procedures in 2 3 6
σ_Prv ← sig(sk_i; R ‖ output_Siu ‖ GHV_i)
Else
Reject Ch;
End.
7 Result = { R, output_Siu, GHV_i, σ_Prv}
vrfsig(pk_i; output_Siu, R, σ_Prv)

Outline

- **Internet of Things Security**
  - Motivation
  - Overview of Remote attestation
  - History of Remote attestation

- **Remote attestation protocols**
  - Software-based attestation
  - Hybrid attestation
  - Swarm attestation
  - Dynamic attestation
  - Distributed services attestation

- **Research perspectives**
  - Recent research projects
  - Conclusions
  - Open challenges & Future works

ERAMO: Effective Remote Attestation through Memory Offloading

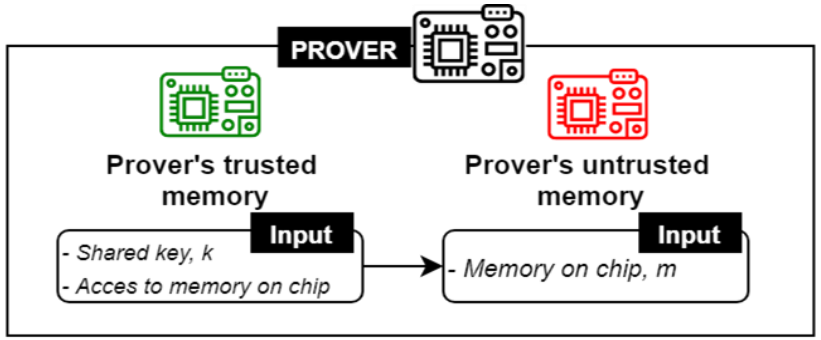
J. H. Østergaard, E. Dushku, and N. Dragoni, **“ERAMO: Effective Remote Attestation through Memory Offloading”**, The IEEE International Conference on Cyber Security and Resilience (IEEE CSR), 2021.

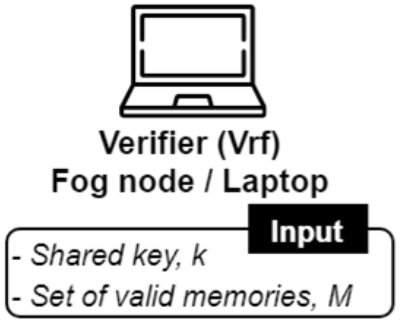
ERAMO Approach

1. Instead of running a complex RA protocol on a resource-constrained IoT device, ERAMO securely offloads memory contents of IoT devices to nearby powerful devices.

2. Allows the Verifier to employ sophisticated methods of attesting the dynamic memory, e.g., memory forensics tools

ERAMO Protocol





N = generateNonce();

1 Request attestation, N

2

Copy memory on chip, *m*

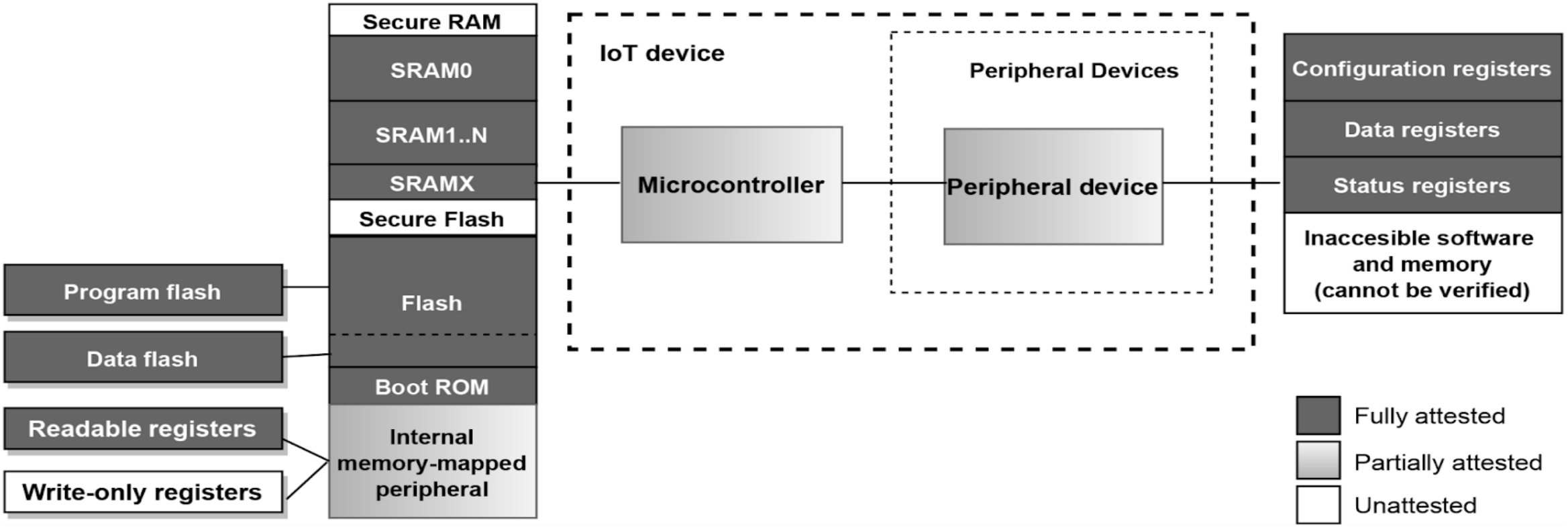
h = hash(m)
s = HMAC(k, (h || N))

3 Send *s and m*

4

h' = hash(m);
s' = HMAC(k, (h' || N));
Verify: s == s' ∧ m ∈ M

Results


Secure RAM
SRAM0
SRAM1..N
SRAMX
Secure Flash
Flash
Boot ROM
Internal memory-mapped peripheral
Program flash
Data flash
Readable registers
Write-only registers
IoT device
Microcontroller
Peripheral Devices
Peripheral device
Configuration registers
Data registers
Status registers
Inaccesible software and memory (cannot be verified)
Fully attested
Partially attested
Unattested

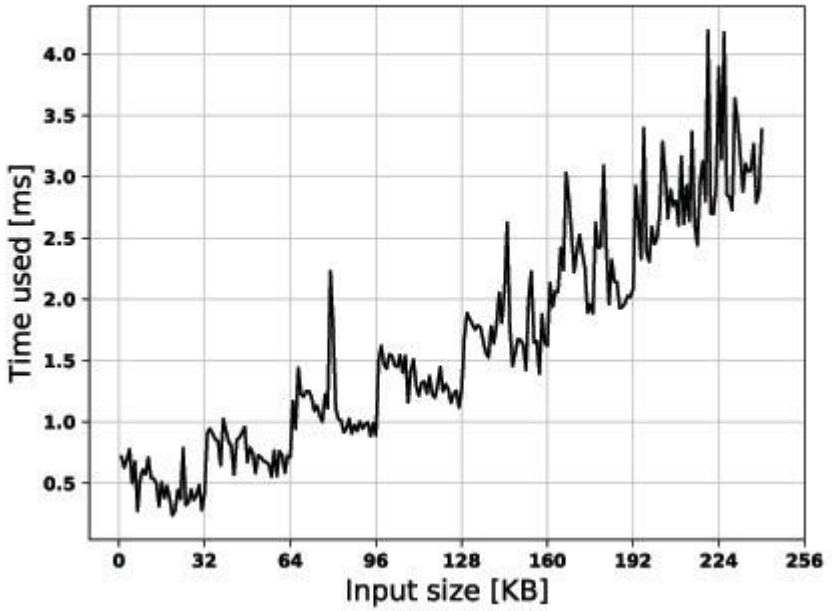
Attestation of non-monolithic systems

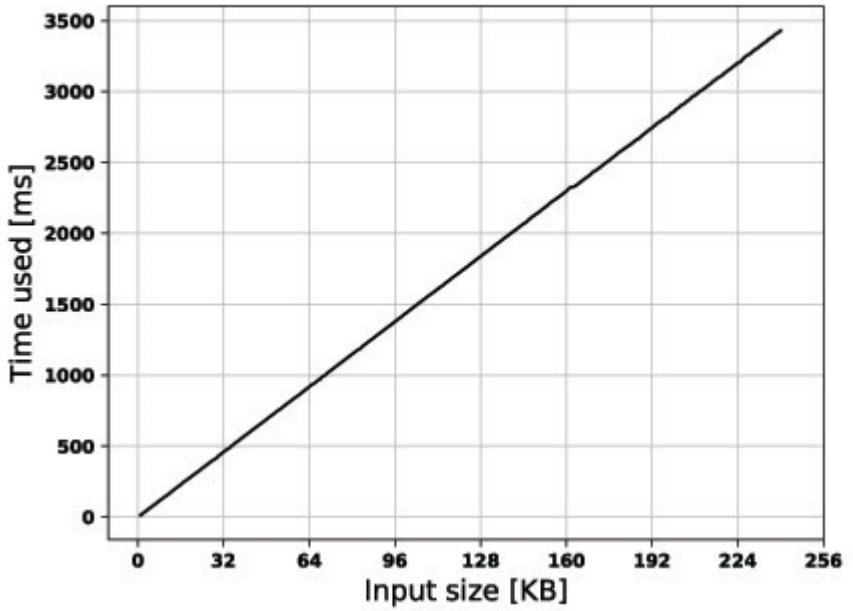
- Extension of attestation
- Systems with additional peripheral devices
- Verify that configurations are correct
- Transmit and verify all peripheral registers

Register Name	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Reset state
hum_lsb	0xFE	hum_lsb<7:0>								0x00
hum_msb	0xFD	hum_msb<7:0>								0x80
temp_xlsb	0xFC	temp_xlsb<7:4>				0	0	0	0	0x00
temp_lsb	0xFB	temp_lsb<7:0>								0x00
temp_msb	0xFA	temp_msb<7:0>								0x80
press_xlsb	0xF9	press_xlsb<7:4>				0	0	0	0	0x00
press_lsb	0xF8	press_lsb<7:0>								0x00
press_msb	0xF7	press_msb<7:0>								0x80
config	0xF5	t_sb[2:0]			filter[2:0]				spi3w_en[0]	0x00
ctrl_meas	0xF4	osrs_t[2:0]			osrs_p[2:0]			mode[1:0]		0x00
status	0xF3					measuring[0]			im_update[0]	0x00
ctrl_hum	0xF2						osrs_h[2:0]			0x00
calib26..calib41	0xE1…0xF0	calibration data								individual
reset	0xE0	reset[7:0]								0x00
id	0xD0	chip_id[7:0]								0x60
calib00..calib25	0x88…0xA1	calibration data								individual

Performance

- Implemented on ARM Cortex M33, 150MHz
- ARM TrustZone for ERAMO and keys
- Performance dependent on hardware





ARCADIS: Control-Flow Attestation of Asynchronous Distributed IoT Services

R. M. Halldórsson, E. Dushku, and N. Dragoni, **“ARCADIS: Control-Flow Attestation of Asynchronous Distributed IoT Services”**, IEEE Access, 2021.

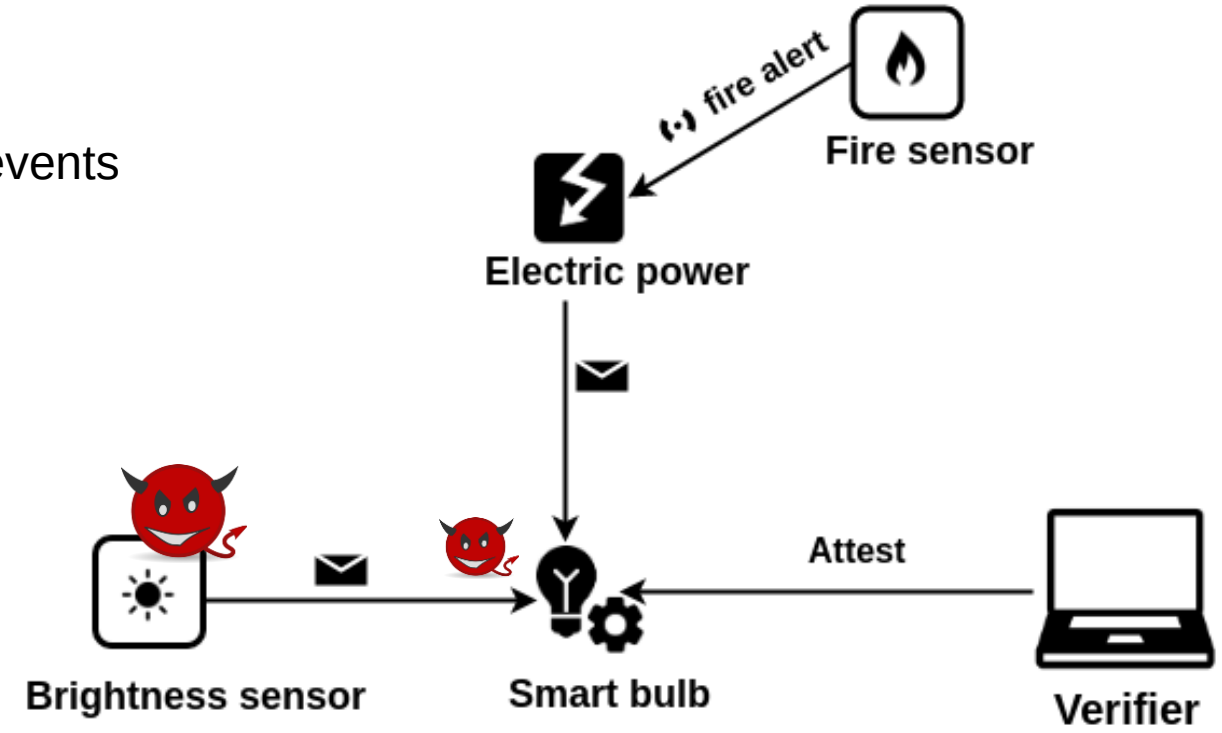
Collective RA

- Attests a group/swarm of IoT devices in a network as a whole
- More efficient than attesting devices one-by-one
- ARCADIS is both a Dynamic scheme and a Collective scheme, focusing on Asynchronous IoT systems

Motivating example: Distributed IoT service (Async)

Legitimate state of Smart bulb is affected by:

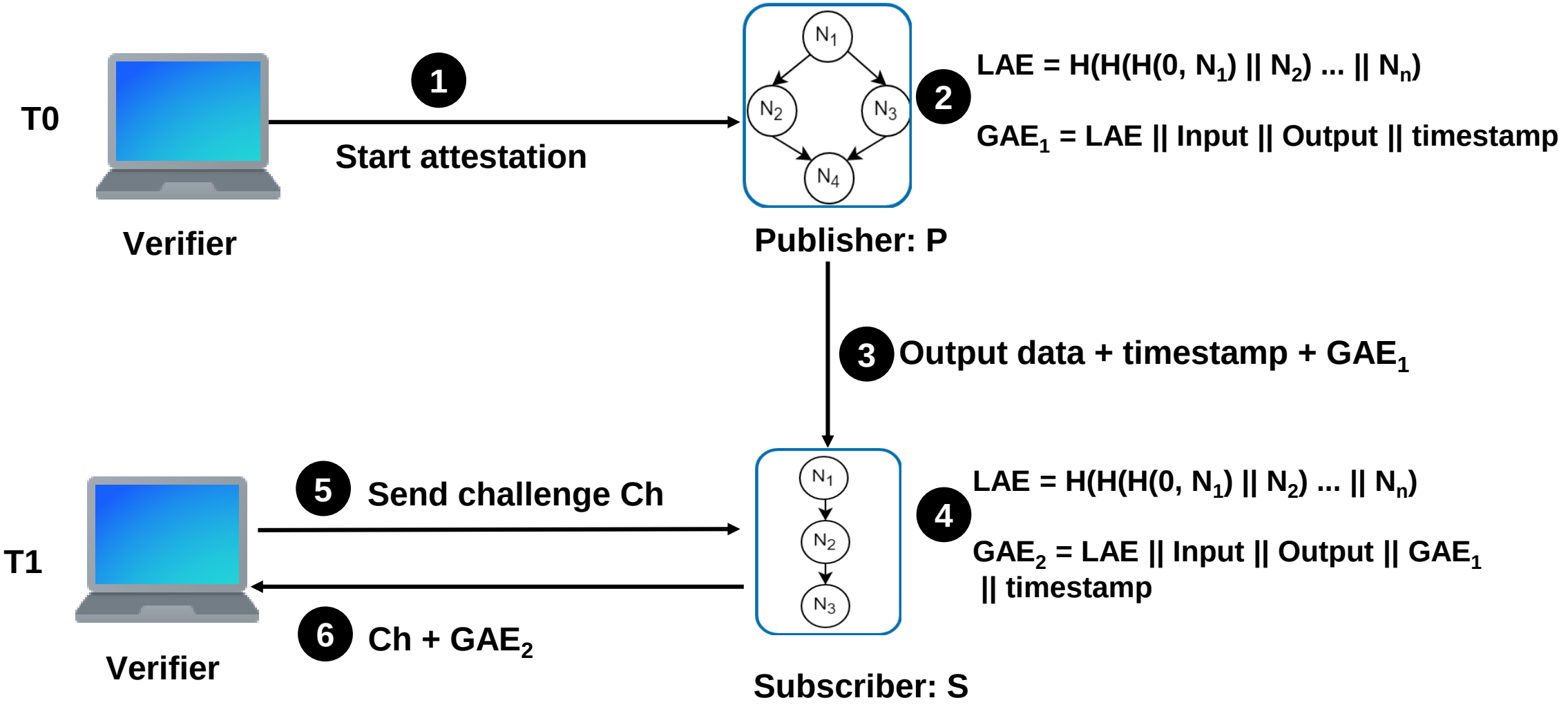
- the data exchanged among devices
- history of the events
- order of occurrence of events



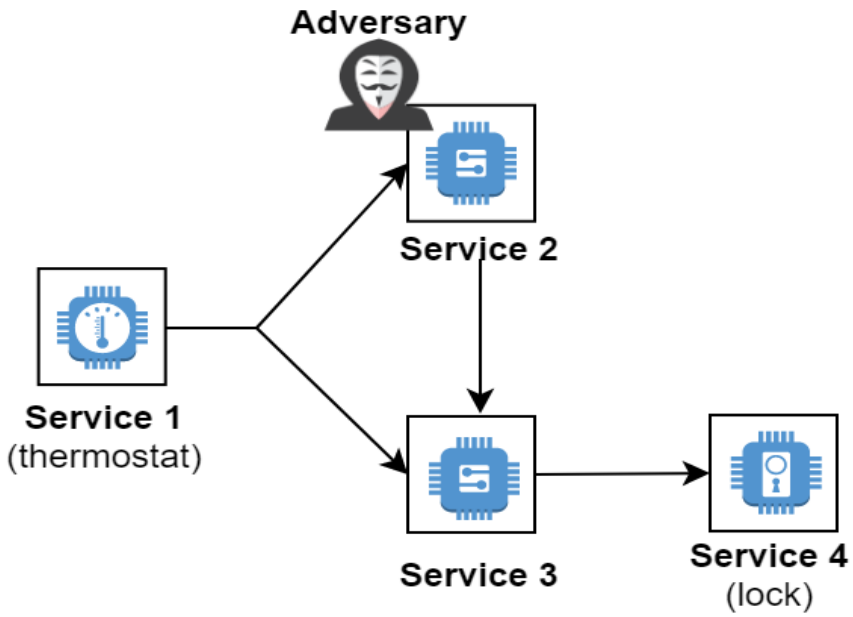
Realistic assumptions

- **Distributed IoT services**
  - Event-driven interactions
- **Distributed Publish/Subscribe pattern**
  - The occurrence of the events in not predictable
- **Clock synchronization**
  - Local clocks on IoT devices are not perfectly synchronized

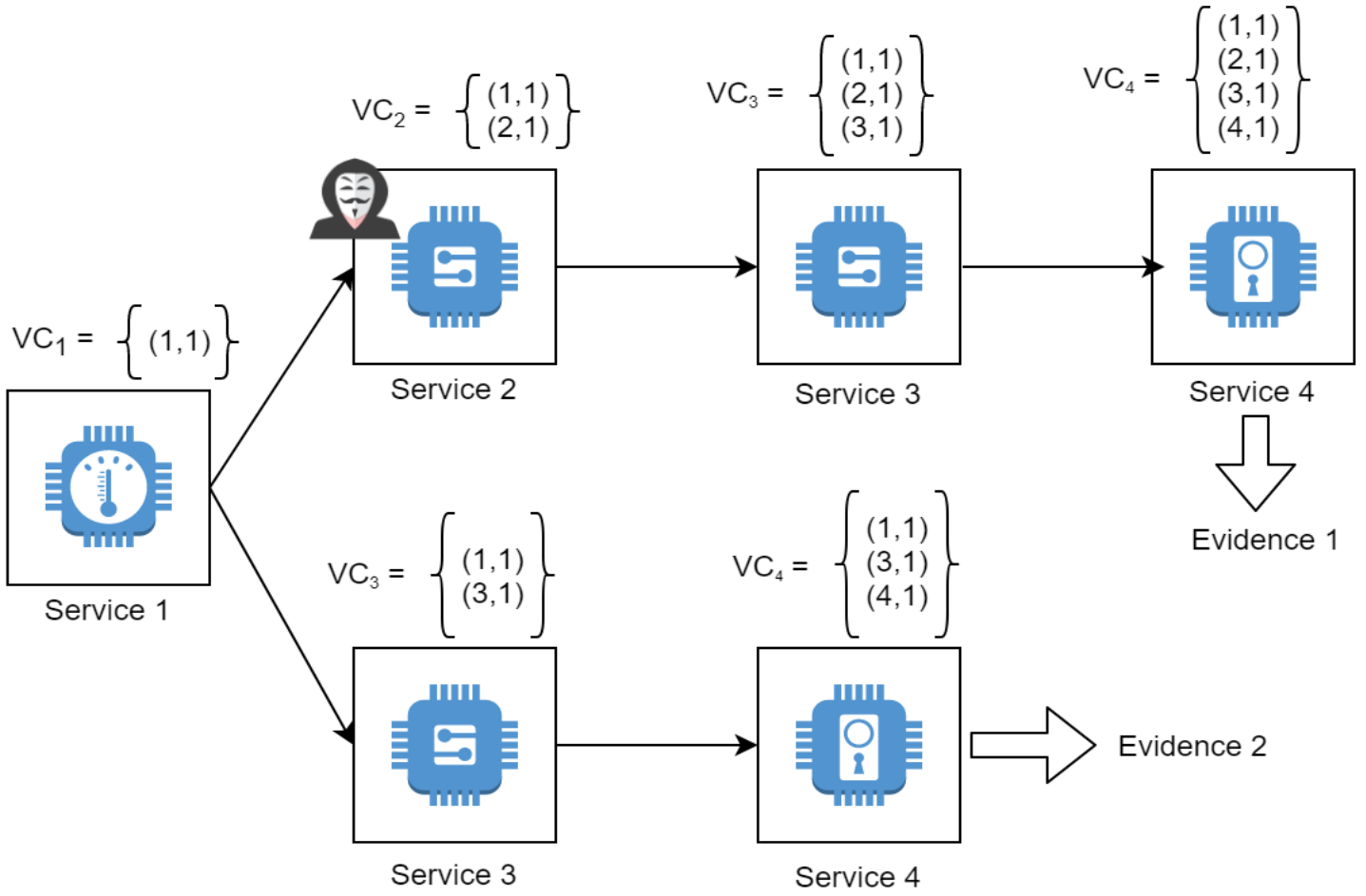
ARCADIS: Overview



ARCADIS: Logical Vector Clock



Scenario

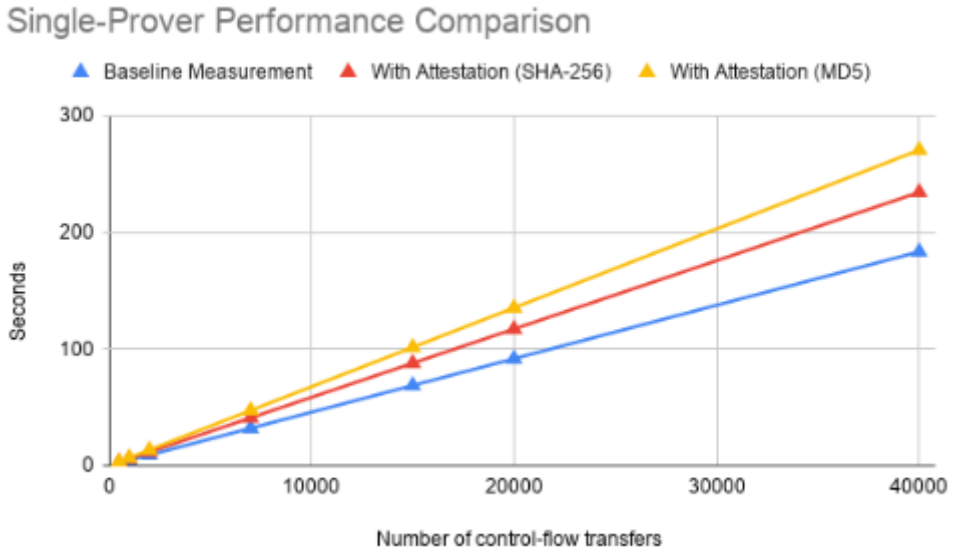


Construction of historical evidence based on Vector Clocks

Simulation details

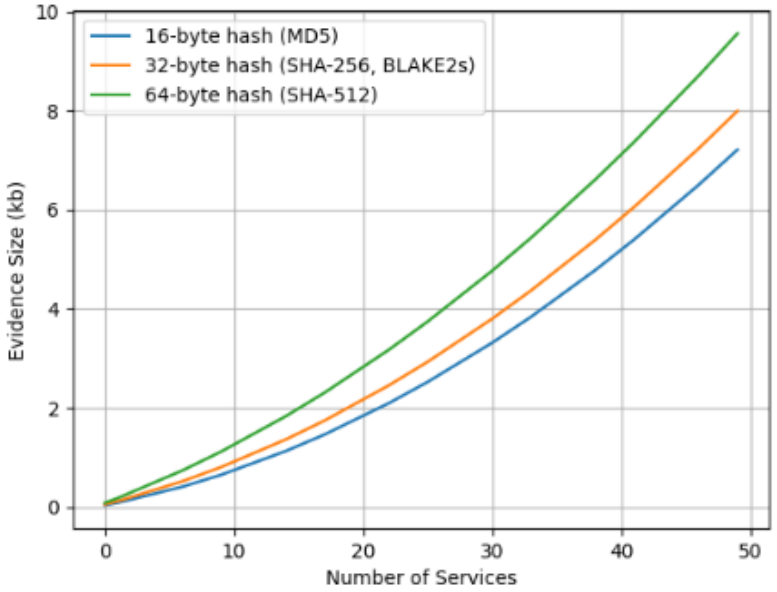
- Implemented for Contiki OS and simulated with the Cooja simulation software
- Simulations on Wismote sensor, TI MSP430 series 5 16-bit CPU, 128/192/256kB flash storage and 16kb SRAM.

Number of control-flow transfers	Embedded program
500	LED Driver
1000	Simple sensor
2000	More complex sensor
7000	Syringe pump
15.000	GPS module
20.000	Gyroscope
40.000	Complex program

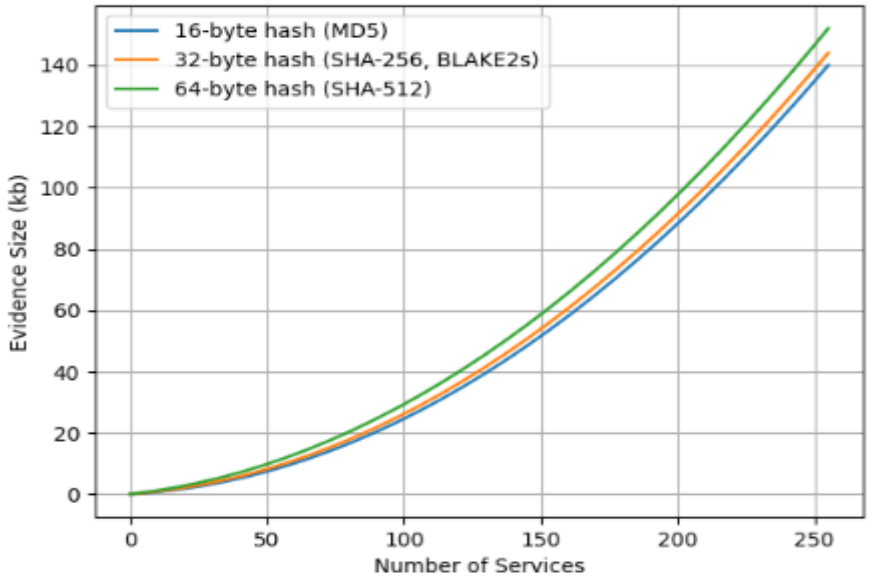


Runtime performance

Practical for smaller networks of about 40-50 Provers



(a) up to 50 services



(b) up to 256 services

Publicly Verifiable Remote Attestation through Blockchain

S. F. J. J. Ankergård, E. Dushku, and N. Dragoni, **“Publicly Verifiable Remote Attestation through Blockchain”**, 14th International Symposium on Foundations & Practice of Security (FPS), 2021.

Our proposal: PERMANENT

Use blockchain technology to make the attestation result **publicly verifiable** and **decentralized**

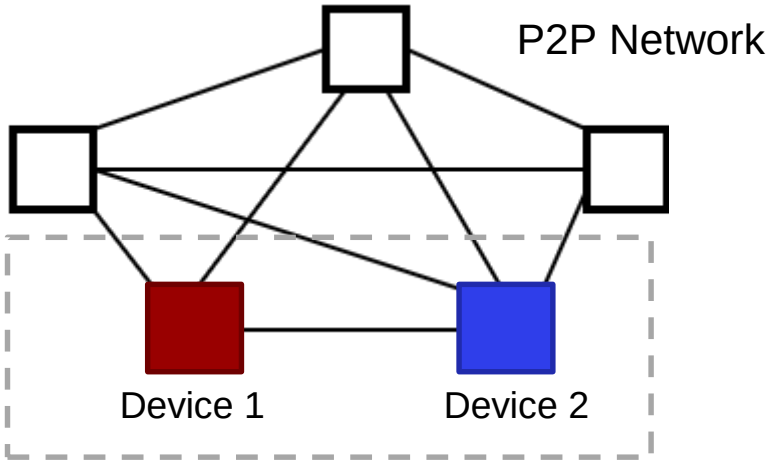
Consensus Algorithms type & requirements

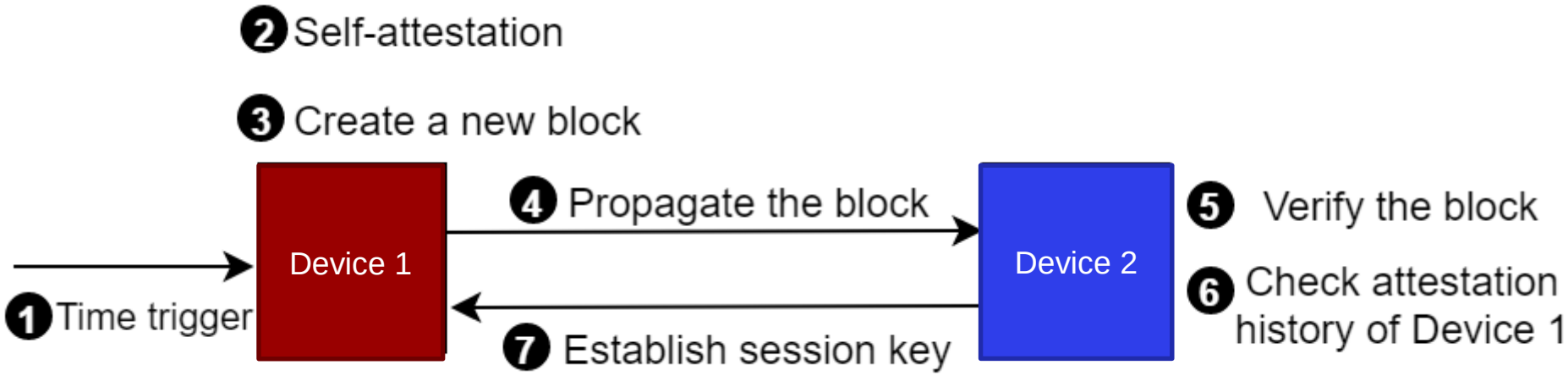
Algorithm	Family	Throughput	Scalability	Overhead
Proof-of-Work (PoW)	Proof-of-X	Low	Low	Computational
Proof-of-Authority (PoA)	Proof-of-X	Low	High	None
Proof-of-Stake (PoS)	Proof-of-X	Low	Low	None
Proof-of-Elapsed-Time (PoET)	Proof-of-X	Low	High	None
Proof-of-Capacity (PoC)	Proof-of-X	Low	Low	None
Proof-of-Burn (PoB)	Proof-of-X	Low	Low	None
Proof-of-Importance (PoI)	Proof-of-X	Low	Low	None
Byzantine Fault Tolerance (BFT)	Voting	High	Low	Communications
Crash Fault Tolerance (CFT)	Voting	High	High	Communications

Consensus Algorithms type & requirements

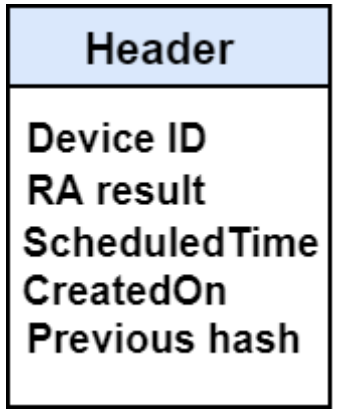
Algorithm	Family	Throughput	Scalability	Overhead
Proof-of-Work (PoW)	Proof-of-X	Low	Low	Computational
Proof-of-Authority (PoA)	Proof-of-X	Low	High	None
Proof-of-Stake (PoS)	Proof-of-X	Low	Low	None
Proof-of-Elapsed-Time (PoET)	Proof-of-X	Low	High	None
Proof-of-Capacity (PoC)	Proof-of-X	Low	Low	None
Proof-of-Burn (PoB)	Proof-of-X	Low	Low	None
Proof-of-Importance (PoI)	Proof-of-X	Low	Low	None
Byzantine Fault Tolerance (BFT)	Voting	High	Low	Communications
Crash Fault Tolerance (CFT)	Voting	High	High	Communications

PERMANENT: System model

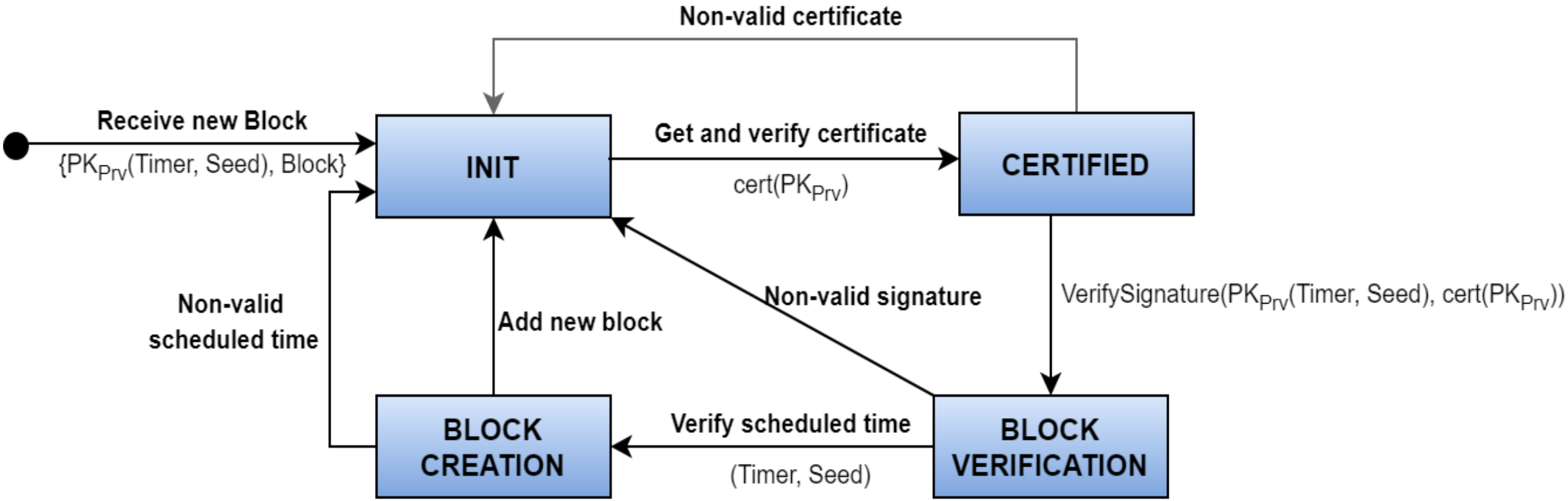




Block design for attestation



PERMANENT: Block Verification

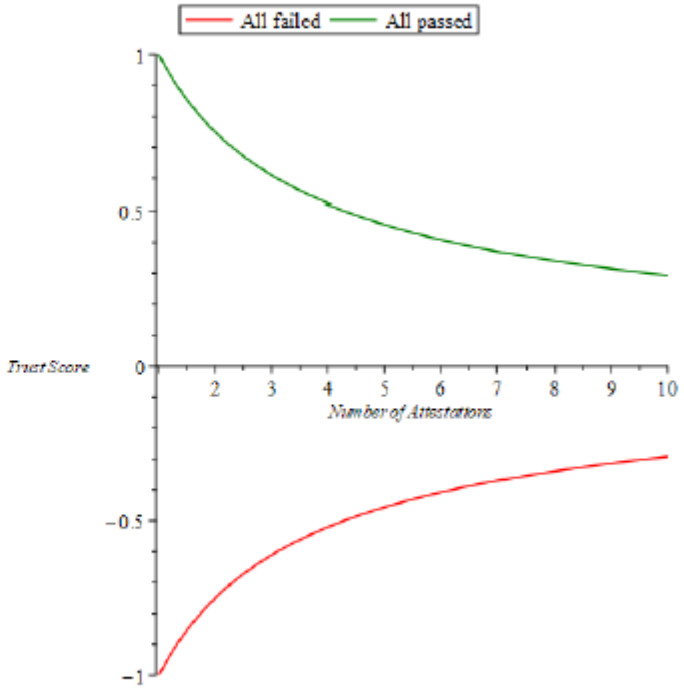


History-based Trust Decision

$$\Phi = \frac{\sum_{i=1}^{n} \left(\frac{CreatedOn_i - CreatedOn_{genesis}}{now - CreatedOn_{genesis}} \right) \times \alpha_i}{n}$$

$$\alpha_i = \begin{cases} 1 \text{ iff Attestation passed} \\ -1 \text{ iff Attestation failed} \end{cases}$$

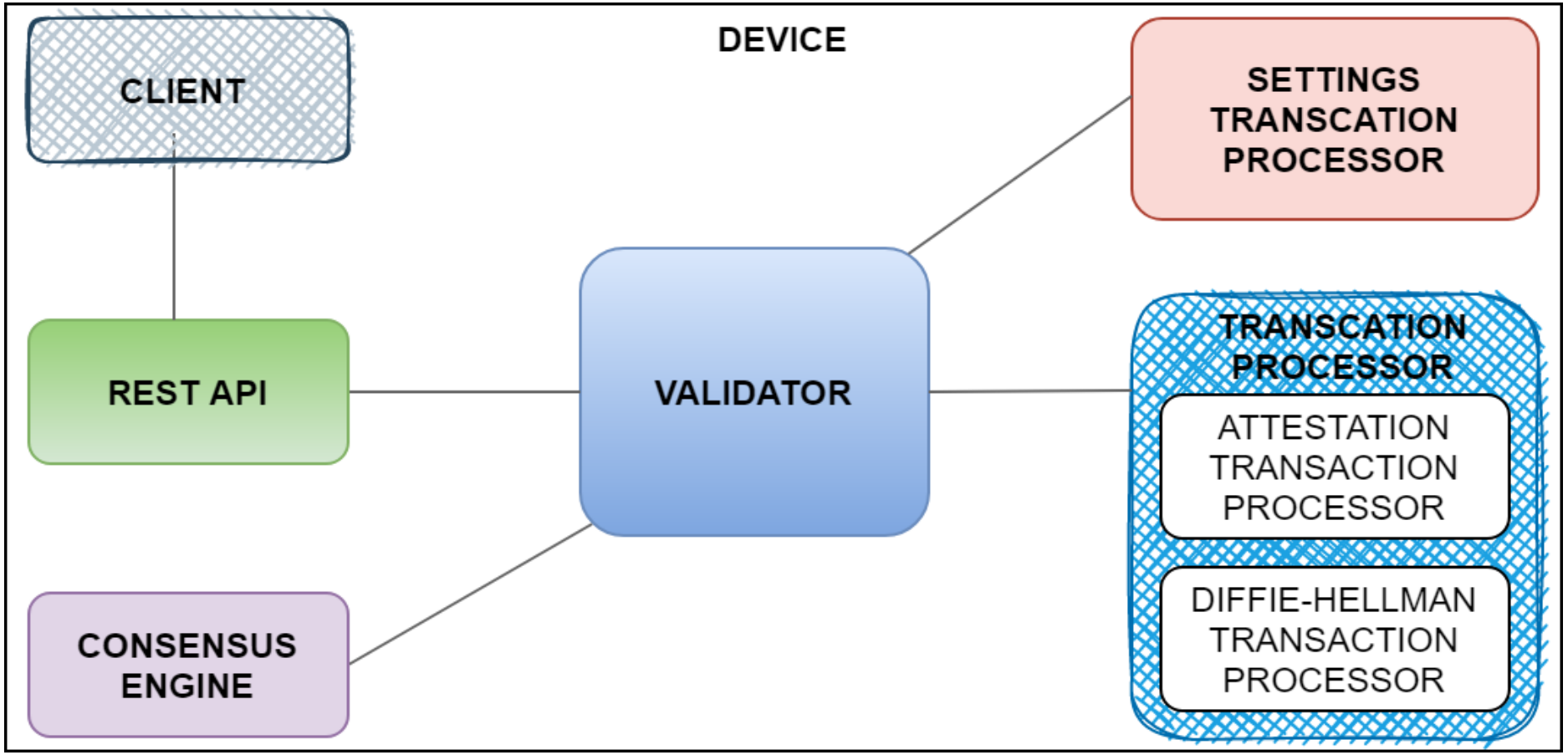
- Failed attestations
- Time weighted average
- Percentage of max score



PERMANENT: Proof-of-concept implementation

- Python, using Hyperledger Sawtooth, PoET consensus algorithm
- Docker has been used to deploy each component in separate containers, simulating a network of devices
- An IoT device can include each component
- The system consists of six components, Client and Transaction Processor are custom and contain the logic of the application.

Proof-of-concept implementation



Outline

- **Internet of Things Security**
  - Motivation
  - Overview of Remote attestation
  - History of Remote attestation

- **Remote attestation protocols**
  - Software-based attestation
  - Hybrid attestation
  - Swarm attestation
  - Dynamic attestation
  - Distributed services attestation

- **Research perspectives**
  - Recent research projects
  - Conclusions
  - Open challenges & Future works

Conclusions

- Introduced RA of IoT devices: Security protocol that guarantees trustworthiness
- Highlighted the need for the attestation of IoT devices. RA can serve as a fundamental building block for other security protocols.
- Presented an overview of the main RA protocols proposed in the literature (software, hybrid, swarm, dynamic, distributed services)

Outline

- **Internet of Things Security**
  - Motivation
  - Overview of Remote attestation
  - History of Remote attestation

- **Remote attestation protocols**
  - Software-based attestation
  - Hybrid attestation
  - Swarm attestation
  - Dynamic attestation
  - Distributed services attestation

- **Research perspectives**
  - Recent research projects
  - Conclusions
  - Open challenges & Future works

Challenges

- Despite many RA approaches, some cyber attacks remain undetected, e.g., data attacks, physical attacks
- New efficient aggregation schemes for swarm attestation and/or distributed service attestation
- Attestation is an overhead operation: device stops the regular work
- Privacy-preserving RA protocol for IoT devices
- There is no generalized remote attestation technique that fits for all the platforms
- There is no RA of large mobile IoT networks, in which nodes join or leave during the remote attestation.

Open Challenges (I)

Detecting physical attacks in swarms with Remote attestation

RA that detect physical attacks rely on the assumption that the adversary needs to shutdown the device for a non-negligible amount time in order to tamper the device.

This assumption relies on the results of a wireless sensors paper [1]. Is this result still valid for the current generation of IoT devices? Can we prove the assumption wrong?

Can we propose a different approach (see [2]) how to detect physical attacks?

[1] Mauro Conti, Roberto Di Pietro, Luigi Vincenzo Mancini, and Alessandro Mei. Emergent properties: detection of the node-capture attack in mobile wireless sensor networks. In ACM WiSec'08.

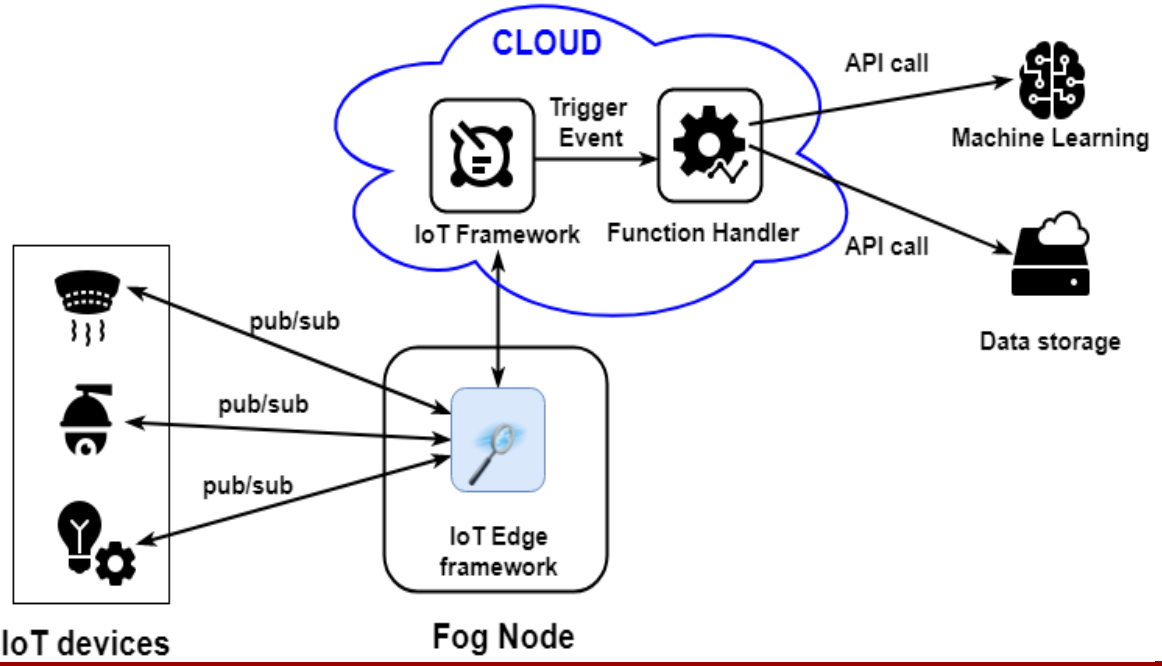
[2] Ibrahim, A., Sadeghi, A.-R., Tsudik, G., and Zeitouni, S. DARPA: Device attestation resilient to physical attacks. In Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks WiSec'16.(2016).

Open Challenges (II)

Extend offloading RA to a large IoT network (Perhaps based on ERAMO)

Exploit the well-established deployment model IoT-Fog-Cloud to implement a new remote attestation approach for low-end IoT devices, which optimizes the remote attestation protocol for IoT devices by securely offloading the attestation computation to the cloud.

.



Open Challenges (III)

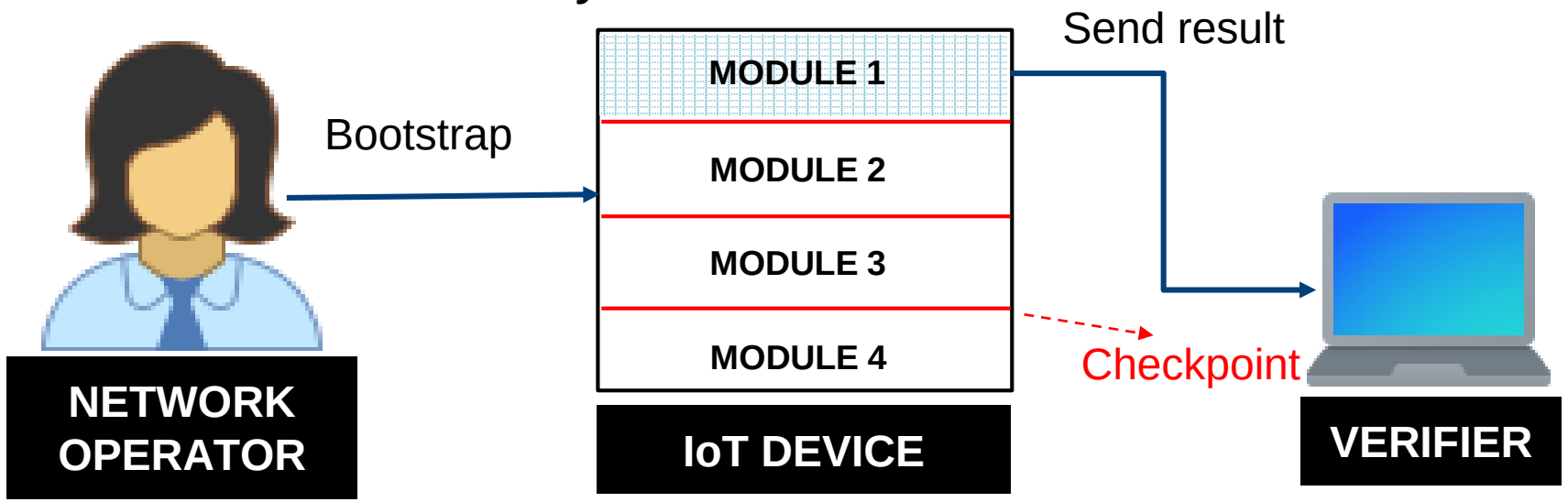
Optimize asynchronous distributed services (Perhaps based on SARA & ARCADIS)

- Create an efficient way to compress the attestation results exchanged among devices Investigate the application of distributed provenance compression schemes on SARA approach. [1]

[1] Chen Chen, Harshal Tushar Lehri, Lay Kuan Loh, Anupam Alur, Limin Jia, Boon Thau Loo, and Wenchao Zhou. 2017. Distributed Provenance Compression. In Proceedings of the 2017 ACM International Conference on Management of Data (SIGMOD '17). ACM, New York, NY, USA, 203-218. DOI: https://doi.org/10.1145/3035918.3035926

Future Works

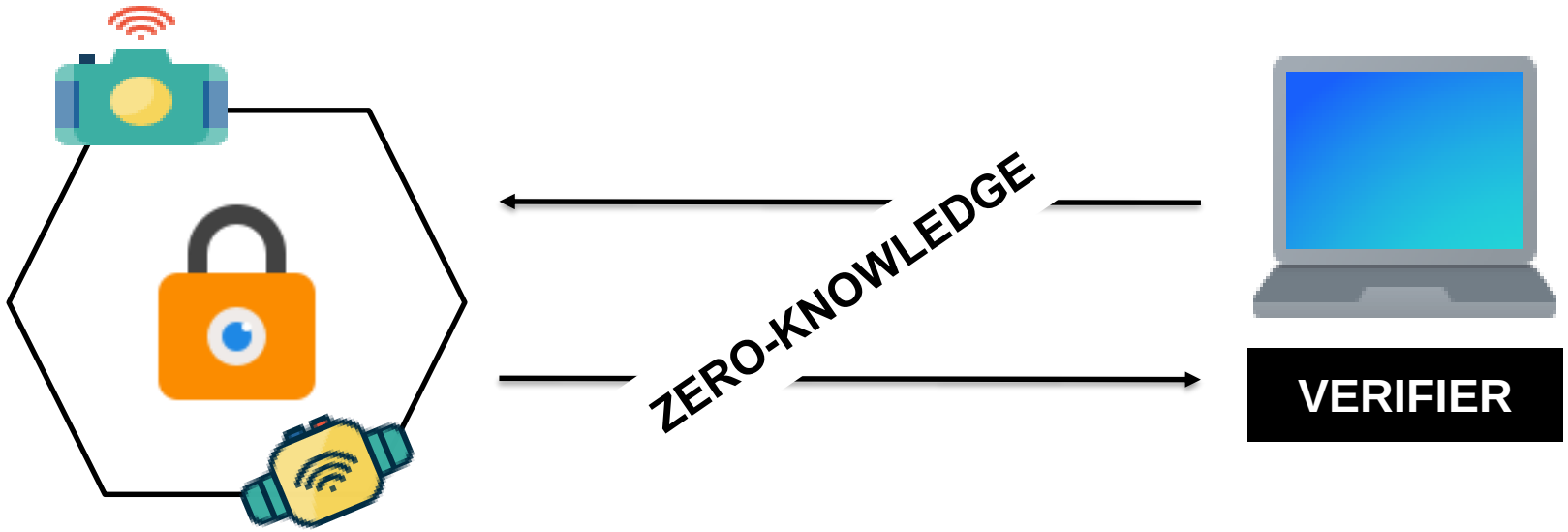
- Lightweight RA operation designed specifically for Intermittent IoT system



RESERVE: Remote Attestation of Intermittent IoT devices
MD M. Rabbani, E. Dushku, J. Vliegen, A. Braeken, N. Dragoni, N. Mentens
In Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems (SenSys '21)

Future Works

- Privacy preserving remote attestation for IoT systems



Recommended reading

- Asokan, N.; Brasser, F.; Ibrahim, A.; Sadeghi, A.R.; Schunter, M.; Tsudik, G.; Wachsmann, C. SEDA: Scalable Embedded Device Attestation. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security CCS ’15, Denver, CO, USA

- Abera, T.; Bahmani, R.; Brasser, F.; Ibrahim, A.; Sadeghi, A.; Schunter, M. DIAT: Data Integrity Attestation for Resilient Collaboration of Autonomous System. In Proceedings of the 26th Annual Network & Distributed System Security Symposium (NDSS), San Diego, CA, USA.

- Dushku, E.; Rabbani, M.M.; Conti, M.; Mancini, L.V.; Ranise, S. SARA: Secure Asynchronous Remote Attestation for IoT Systems. IEEE Transactions on Information Forensics and Security, 2020, 15, 3123–3136.

Questions?

EDLIRA DUSHKU
edldu@dtu.dk