SEcure DIstributed IoT ManagemENT (SEDIMENT) for OPS-5G

Peraton Labs SEDIMENT Team
PI: David Shur

The views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA).

SEDIMENT Overview

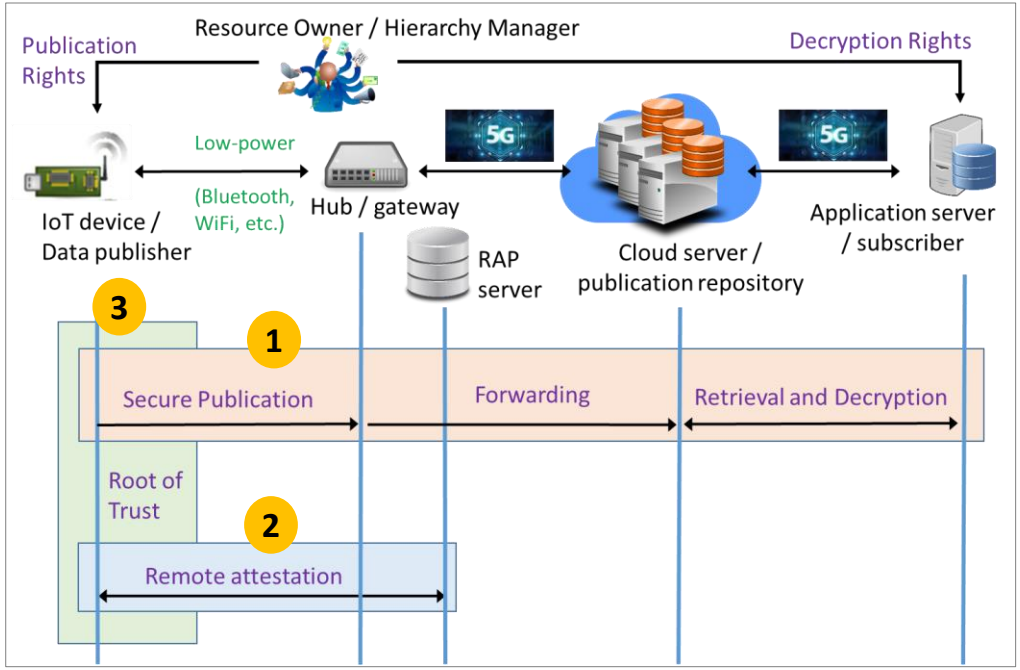
Peraton | LABS

PROBLEM

Resource-constrained devices (IoTs) are a weak link in the 5G security chain. Being easier to compromise, they pose threats to privacy and security of 5G Core, Internet, and critical infrastructure.

SOLUTION

1. **Scalable Cryptography for confidentiality, integrity, and authentication that is end-to-end, one-to-many**
2. **Remote Attestation of device identity, load-time and run-time integrity of software and configuration**
3. **Root-of-Trust, including hybrid options for mid-and low-tier IoT devices**



https://sediment-lfproject.github.io

PROGRAM

DARPA OPS-5G Technical Area 2: Cross-scale 5G node & network security

Phase 1: 10/2020-03/2022
Phase 2: 04/2022-10/2023 ⬅
Phase 3: 11/2023-10/2024

TRANSITION

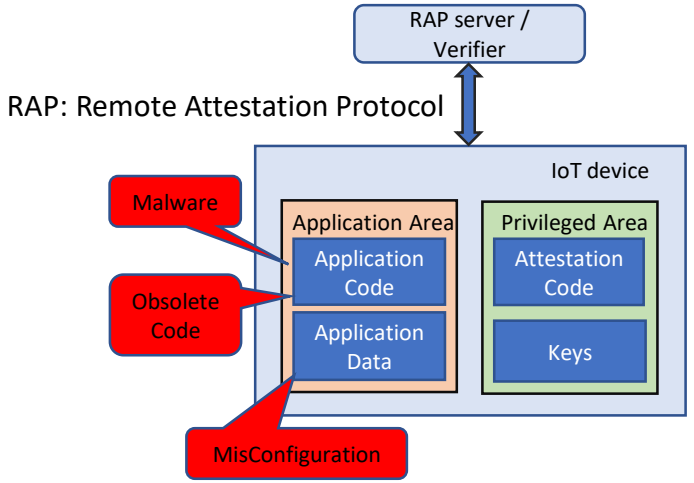
1. **Open-source project under Linux Foundation**
2. **Exploring alignment with 5G Super Blueprint**
3. **IETF RATS engagement being considered**
4. **Demonstration in Peraton Labs 5G testbed (or other LF testbed) in Phase 3**

SEDIMENT provides a combination of software root of trust, remote attestation, and resource-efficient cryptography to build a zero-trust security system that scales across heterogeneous computing platforms.

SEDIMENT Remote Attestation (RA)

Peraton | LABS

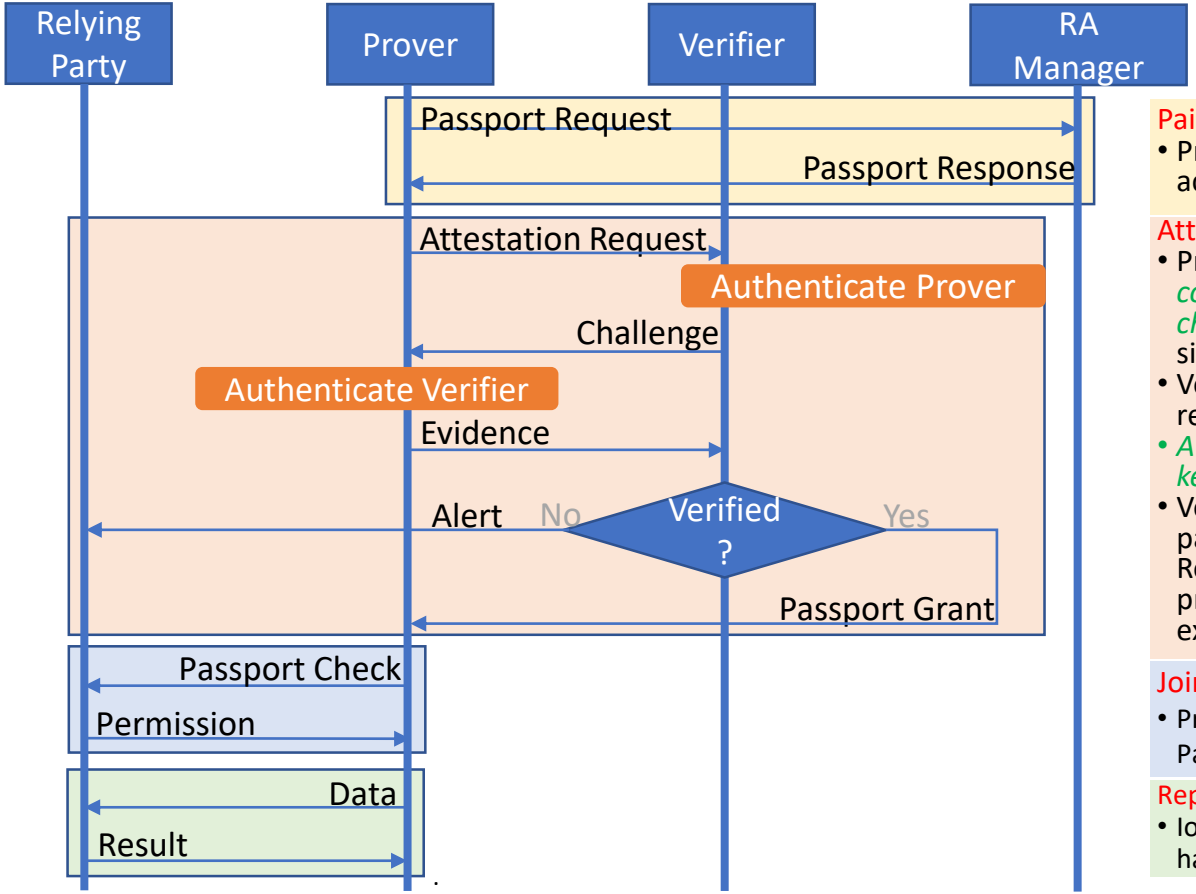
- RA goals
  - Detect presence of malware and software tampering
  - Verify IOT device configuration integrity
  - Authenticate
  - Operate across the entire scale of devices on a 5G network



- Key RA components
  - A protocol (cf. next slide) that supports
    - Authentication and authorization among communicating entities
    - Flexibility in requesting various types of evidences for attestation
      - E.g., firmware fingerprint; software configurations; process status
  - Hardware/software/hybrid Root of Trust (RoT) that ensures the integrity of RA protocol execution [1]
    - Protection of attestation credentials
      - Exclusive access
      - No leaks
    - Safe execution in a protected region
      - Execution atomicity (cannot be interrupted)
      - Immutability (attestation code and everything it depends on cannot be altered after being loaded)
      - Controlled invocation (run attestation code in its entirety)

[1] Karim Eldefrawy, Norrathep Rattanavipanon, and Gene Tsudik. 2017. HYDRA: HYbrid Design for Remote Attestation (Using a Formally Verified Microkernel). *WiSec '17*.

Remote Attestation Protocol Framework



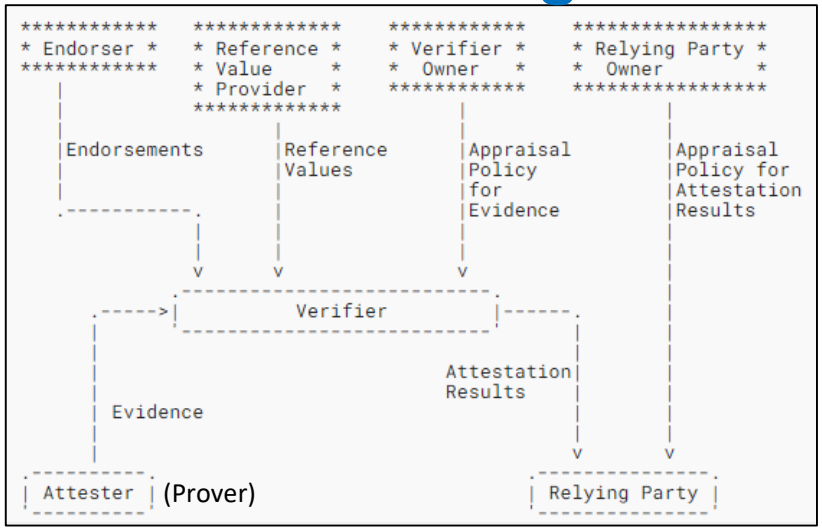
Pairing:
- Prover authenticates itself to RA Manager and acquires Verifier connection information

Attestation:
- Prover initiates attestation, *instead of continuously keeping a communication channel open waiting to be attested,* with significant energy savings.
- Verifier can *issue a variety of challenges* to request Prover to produce evidences.
- *Authentication digests signed with the shared key* enables both parties to authenticate.
- Verifier evaluates evidence. Either grants a passport to Prover or sends an Alert to Relying Party. Prover must restart the procedure after some time if it does not get expected positive response from Verifier.

Join:
- Prover submits granted passport to the Relying Party to gain permission to communicate.

Report:
- IoT device sends application data after Passport has been accepted.

SEDIMENT RA Alignment with IETF RATS Architecture



Conceptual Information Flow https://datatracker.ietf.org/doc/rfc9334/

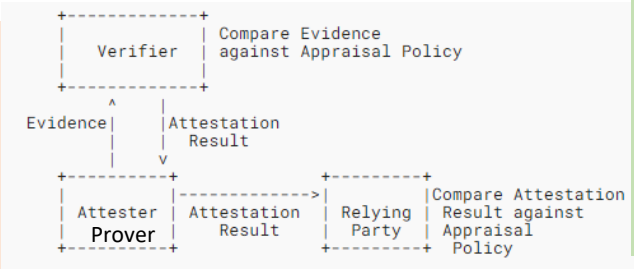
A wide range of evidence types are supported including:

- Full/sparse firmware HMAC
- Application firmware version
- OS/BIOS version
- Boot time; Geolocation
- Software configurations
- User defined functions (UDFs) for run-time integrity checks
  - Running status of specified processes
  - Resource usage
  - Log checks and uptime checks

API features:

- On-demand attestation requests
- IoT device attestation management
  - Access/update of device reference values, appraisal policies

Relying Party: admission;	Network equipment for network access
	Server holding confidential data for release; Entity needing trustworthy remote elements
Evidence:	Claims about the target environment
Reference Values:	Values used in appraising evidences
Endorsements:	Statements that help authenticating device info
Reference models:	Passport; Background-Check



Passport Model

SEDIMENT RA is being aligned with IETF RATS, following the Passport model

SEDIMENT Transition: Linux Foundation Project

Peraton | LABS

- "SEDIMENT, a Series of LF Projects" is being set up with the following mission statement:
  - "to provide open-source software implementing a distributed and scalable security architecture with remote attestation for networked IoT devices"
- Documents have been adopted/executed by LF and Peraton Labs:
  - Technical Charter, Series Agreement, and Contributor's Agreement (word mark transfer)
- Project supporters including one LF member:
  - NIWC (LF member), DARPA, Peraton Labs, UCI, Kryptowire Labs, Aarno Labs, USC/ISI
- Apache v2.0 License for code and CC-BY-4.0 for documentation chosen
- Code and documentation released by DARPA DISTAR and Peraton
- Repositories for code and webpages: https://github.com/sediment-lfproject
- LF Project public announcement on February 22nd, 2023

Engagement with LF 5G SBP

- Develop use cases for RA in collaboration with Muddasar Ahmed and Ranny Haiby
  - Ensure that IoT devices on a network are authentic and have not been tampered with
  - Ensure that IoT devices can be seamlessly maintained and securely updated
- Develop and integrate SEDIMENT RA onto existing 5G SBP lab infrastructure as proof of concept of ways to realizing 5G SBP use cases

Use Case #1: IoT Device Security and Authentication

- Insure that IoT devices on a network are authentic and have not been tampered with.
  - This is particularly sensitive in remote areas that are not often frequented by people.

- How would remote attestation help?
  - RA Verifier is set to periodically "inspect" remote cameras by checking their firmware fingerprint.
    1. RA Verifier confirms that the firmware fingerprint of Remote Camera A is authentic and permits/allows the camera to stay on the network.
    2. RA Verifier determines that the firmware fingerprint of Remote Camera B is not authentic. RA Verifier then alerts Relying Party (e.g., firewall) to deny Remote Camera B access on the network.

Security Enforcement with KubeArmor

- Visibility into SEDIMENT application behavior
  - Identify the process forking behavior of the application
  - Identify sensitive asset access of SEDIMENT
  - Identify network access required by SEDIMENT

- Protection policies for Gateway deploying SEDIMENT Verifier.
  - Process Whitelisting: Do not allow processes to execute within SEDIMENT container outside of the given spec.
  - Network Access: Only allow SEDIMENT binaries to use the network primitives
  - Check SEDIMENT configuration files and create a security net around SEDIMENT's sensitive assets.
  - Use host hardening policies to protect host.