RESERVE: Remote Attestation of Intermittent IoT devices

Md Masoom Rabbani
ES&S, imec-COSIC, ESAT, KU Leuven
Diepenbeek, Belgium
mdmasoom.rabbani@kuleuven.be

Edlira Dushku
DTU Compute, Technical University of Denmark (DTU)
Lyngby, Denmark
edldu@dtu.dk

Jo Vliegen
ES&S, imec-COSIC, ESAT, KU Leuven
Diepenbeek, Belgium
jo.vliegen@kuleuven.be

An Braeken
Faculty of Engineering, Vrije Universiteit Brussel (VUB)
Brussels, Belgium
an.braeken@vub.ac.be

Nicola Dragoni
DTU Compute, Technical University of Denmark (DTU)
Lyngby, Denmark
ndra@dtu.dk

Nele Mentens
ES&S, imec-COSIC, ESAT, KU Leuven
& LIACS, Leiden University
Diepenbeek, Belgium
nele.mentens@kuleuven.be

ABSTRACT

Internet of Things (IoT) devices have enveloped our surroundings and have been increasingly deployed in many domains. Even though the IoT has generated unprecedented opportunities, the poorly secured design of IoT devices makes them an easy target for cyber attacks. Aimed at securing IoT devices, Remote Attestation (RA) is a security technique that identifies threat presence in IoT systems. Typically, RA is an atomic procedure that requires uninterrupted connectivity to execute. However, in energy harvesting context where intermittent IoT devices go into sleep mode immediately after regular operations, the atomic property is difficult to achieve. In this paper, we propose RESERVE, a novel lightweight RA protocol designed specifically for Intermittent IoT devices. RESERVE aims to improve the security of intermittent systems by detecting malware presence during online mode and guaranteeing with some probability software legitimacy during offline mode. In particular, RESERVE ensures trustworthiness by organizing the device's software into modules, and after regular operation each device attests as many modules as fit in its energy budget.

CCS CONCEPTS

• **Security and privacy → Network security**; • **Computer systems organization → Embedded systems**.

KEYWORDS

remote attestation, security, intermittent computation

ACM Reference Format:
Md Masoom Rabbani, Edlira Dushku, Jo Vliegen, An Braeken, Nicola Dragoni, and Nele Mentens. 2021. RESERVE: Remote Attestation of Intermittent IoT devices. In *The 19th ACM Conference on Embedded Networked Sensor Systems (SenSys '21), November 15–17, 2021, Coimbra, Portugal.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3485730.3493364

ENSsys, Workshop co-located with ACM SenSys'21, November 15–17, 2021, Coimbra, Portugal

ACM ISBN 978-1-4503-9097-2/21/11…$15.00
https://doi.org/10.1145/3485730.3493364

1 INTRODUCTION

Internet of Things (IoT) devices are permeating our surroundings by increasingly getting deployed in multiple domains ranging from smart homes to smart cities. However, the vast majority of IoT devices lack even basic security properties, and testing of these devices is often overlooked due to their low-cost nature. Thus, attacks like stuxnet [18], Mirai botnet [17], smartTV hack [2], and IoT-ransomware [1], to mention only a few, have exploited IoT vulnerabilities and have shown to be devastating.

To deal with the expanding attack surface in IoT, Remote Attestation (RA) is a well-established security mechanism that detects malware presence in a device. In RA, a trusted party called Verifier verifies the trustworthiness of a potentially untrusted device called Prover. Classically, RA gets executed randomly at unpredictable times and requires an uninterrupted power supply during attestation. In addition, during attestation, the Prover stops its regular operations for a certain period of time to perform RA execution. Thus, RA is an overhead operation. This is a very strong assumption for the energy-harvesting environments which deploy intermittent devices and therefore cannot rely on a continuous power source. To this end, performing RA over a network of devices that work under intermittent connectivity remains an open challenge.

Europe has recently begun the green transition to reduce the global energy footprint and eventually be climate-neutral by 2050. Intermittent IoT devices are increasingly used in different fields such as oil-gas exploration, weather monitoring, and military application. Due to their sensitive mode of operation and deployment in inaccessible terrains, it is essential to guarantee the security of their operations because it frequently results in financial loss. To preserve energy, these devices perform their regular task and switch to sleep mode. Thus, executing uninterrupted RA is challenging. Intermittent IoT systems require the development of novel RA protocols that address the interrupted nature of these systems and yet provide much-needed security.

Contribution of the Paper. In the context of the challenges described above, this paper brings two main contributions:

- To the best of our knowledge, RESERVE is the first RA protocol designed to enable attestation of intermittent IoT systems.
- RESERVE brings novelty in the RA domain by releasing the atomic execution assumption of the state-of-the-art RA protocols and allowing interruptibility in the attestation execution.

2 RELATED WORK

RA is often classified into three main categories based on architectural designs [8]: (1) software-based, (2) hardware-based, and (3) hybrid RA schemes. As there is no requirement for specialized trusted hardware, software-based RA techniques (e.g.,[16]) are less expensive than hardware-based RA techniques, but they provide weaker security guarantees. On the other hand, hardware-based RA protocols (e.g., [4]) rely on the use of a specific hardware platform as a secure execution environment to assure their security. However, low-cost IoT devices cannot employ expensive specialized hardware platforms. Thus, hybrid RA schemes (e.g., [13]) are suitable for IoT devices because they rely on a hardware/software co-design principle consisting of a minimal read-only hardware-protected memory to guarantee uninterrupted, safe, and secure code execution. However, existing RA protocols are not practical for energy-harvesting systems because they require continuous power supply.

3 SYSTEM MODEL

We assume an intermittent IoT system, in which the devices cannot guarantee to perform the entire attestation execution along with their regular operations. Yet, due to the deployment of these devices in safety-critical domains (such as nuclear plants, oil-gas explorations, and military applications), it is indeed crucial to verify the trustworthiness of these devices. To design a RA protocol in this setting, we consider the presence of the following entities as shown in Figure 1.

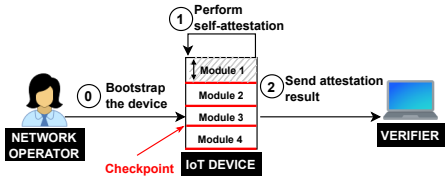

Figure 1: System model of RESERVE protocol

- IoT Device (**Dev**): We assume heterogeneous IoT devices **Dev** deployed in the network. The design of the software running in the **Dev** is based on the modular principle, where each module is associated to a Checkpoint (**ChkPoint**). To preserve energy, **Dev** will run self-attestation whenever it wakes up to perform regular operations.
- Verifier (**Vrf**): The Verifier is an external trusted party that verifies the trustworthiness of **Dev**. We assume that **Vrf** has access to the binaries of each device and has pre-computed the legitimate hash values for each **ChkPoint**. The base stations can act as a **Vrf** since **Dev** connects to them directly.
- Network Operator (**OP**): **OP** guarantees the secure bootstrap of the software deployed on each **Dev** and the secure key distribution among devices at the beginning of the IoT system operation. **OP** is also responsible for ensuring the instrumentation of **Dev** software into **ChkPoint**.

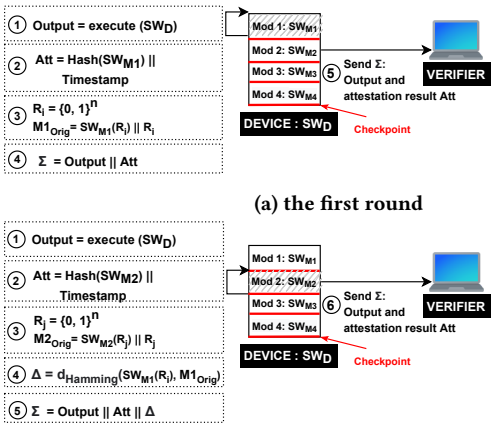

Figure 2: Device Operations of RESERVE protocol

4 ADVERSARY MODEL

Considering the intermittent system described in Sect. 3, we assume the presence of a Software Adversary ($\mathbf{SW_{adv}}$) in the system. The primary objective of a $\mathbf{SW_{adv}}$ is to compromise any **Dev** in the network by injecting malware. The $\mathbf{SW_{adv}}$ can either perform its attack remotely (e.g., remote adversary) or by being present near to the device (e.g., local adversary).

Assumptions. In line with the state-of-the-art RA techniques [5, 9, 11, 12, 15], we keep out of our current scope the physical adversary (capable of capturing the device or performing hardware modifications) and Distributed Denial of Service (DDoS). We also assume that a $\mathbf{SW_{adv}}$ can not modify the hardware of any device. Moreover, in the current scope of the paper we assume that and **Vrf** have established a secure communication for their regular operation. Thus, we do not consider a communication adversary that has full control on the communication channel, e.g., performing packet drop or eavesdrop attack.

Device requirements. In the following we discuss the minimum device requirements, that are also aligned with the existing hybrid RA schemes like [10, 12, 14, 19].

- Read-Only Memory (ROM): It contains the RESERVE software and the 3-bit secret random seed of the Pseudo Random Number Generators (PRNG). This guarantees that a $\mathbf{SW_{adv}}$ cannot modify the attestation measurement.
- Memory Protection Unit (MPU): It guarantees that device key is accessed only by RESERVE protocol resided in ROM.
- Secure writable memory: Only RESERVE has read-write access to this memory region, which is used to securely store checkpoints.
- Real-Time Clock (RTC): Each IoT device needs to have a clock which is not modifiable by software.

5 RESERVE PROTOCOL: THE IDEA

In this section, we present an approach for attestation of intermittent IoT devices. Due to the disruptive nature of the intermittent systems, it is not guaranteed that an IoT device **Dev** can complete the execution of an uninterrupted RA procedure.

As depicted in Figure 2, **Dev**'s software SW_D follows a modular architecture, consisting of many modules, each uniquely identified as SW_{Mi}, for $1 \leq i \leq n$. One module is a standalone software component that is also the minimal unit for which **Dev** should perform an uninterrupted attestation in order to be recognized by the **Vrf**. Given that the **Dev**'s software is instrumented during the bootstrap phase, we assume that each module is associated to a checkpoint **ChkPoint**. One **ChkPoint** contains Program Counter, Stack Pointer, Register File and Main Memory. The **Vrf** has access to the binaries of all the modules and knows in advance the expected legitimate state of each module.

We consider an intermittent system where each **Dev** remains offline and wakes up for a short-period of time to perform its regular operation utilizing a pre-defined energy budget. When the regular operation does not consume the entire allocated energy budget, it will use the remaining energy to accommodate an attestation execution. As soon as the **Dev** completes its regular operation $Output = execute(SW_D)$ (Step (1) in Figure 2a), it self-triggers RA and computes each module's attestation , i.e, performing collision-resistant hash of the software module, $hash(SW_{M1})$. To guarantee the freshness of the measurements, the attestation result Att includes also the timestamp (Step (2)). Once the module attestation is finished, the **Dev** uses a pseudo-random generator (PRNG) to generate a random number (R_i), selects a line of code $SW_{M1}(R_i)$ from the module SW_{M1}, and stores $M1_{Orig} = SW_{M1}(R_i)||R_i$ securely in the **ChkPoint** corresponding to that module (Step (3)). The output of the regular operation $Output$, as well as the attestation responses Att of each individual module, will be stored locally on the **Dev** till it consumes the energy budget. Before switching to sleep mode, the **Dev** computes Σ (Step (4)) by concatenating the $Output$ (Step (1)) along with the attestation response Att (Step (2)) and sends it securely to the **Vrf** as a single response Σ (Step (5)).

When the **Dev** wakes up in the next round, it will begin attestation of the remaining modules that were not verified during the previous round, e.g., SW_{M2} in Figure 2b. To ensure that the attested modules (i.e., SW_{M1}) have not been compromised during the offline mode, the **Dev** will check if the line saved inside the checkpoint $M1_{Orig}$ matches the current state of the module's code $SW_{M1}(R_i)$, by comparing them using Hamming distance $\Delta = d_{Hamming}(SW_{M1}(R_i), M1_{Orig})$, and returning 0 if the module is correct (Step (4) in Figure 2b). Similar to the first round, before switching to sleep mode, the **Dev** computes $\Sigma = Output||Att||\Delta$ (Step (5)) and sends it to the **Vrf** (Step (6)).

6 TECHNICAL DETAILS

Table 1: Simulation setup: Parameters for RESERVE experiments

Parameters	Data-size
Output	8 bit
Attestation Att	32 byte (SHA256 hash algorithm)
Timestamp	8 byte
Seed of PRNG R	2 bit
Random string selected	20 byte
Δ for 10 modules	10 bits

We assume an Internet of Things (IoT) network with intermittent devices (e.g., Tmotesky device). RESERVE uses the IEEE 802.15.4 MAC layer protocol and 6LoWPAN as an adaption layer to communicate between the **Dev** and the **Vrf**. This configuration is very popular in IoT settings [5–7]. The de-facto communication protocols (e.g., 6LoWPAN, ZigBee, etc.) give a maximum packet length of 128 bytes, of which 102 bytes can be used for data transfer in real-world IoT settings [3]. In RESERVE, we consider the following attestation parameters details in Table 1.

7 CONCLUSIONS AND FUTURE WORKS

In this paper, we presented RESERVE, a lightweight and novel RA approach for intermittent IoT systems. RESERVE brings traditional security mechanism like RA to intermittent IoT systems to address the security loopholes. RESERVE lays the basic foundation of applying novel RA protocols for intermittent IoT systems. As future work, we will implement RESERVE on real IoT devices (for e.g., MSP430) and also perform emulations based on the Contiki platform to verify its scalability and efficiency on a large network. In addition, we will also explore other hardware/software implementations.

8 ACKNOWLEDGMENT

This work is supported by Danish Industry Foundation through project "CIDI—Cybersecure IoT in Danish Industry" (project number 2018-0197) and the European Union's Horizon 2020 Research and Innovation program under Grant Agreement No. 952697 (ASSURED). This work is also supported by CyberSecurity Research Flanders with reference number VR20192203.

REFERENCES

[1] 2018. FBI Identifies Biggest Cyber Threats as IoT, Ransomware, Compromised Email. https://governmenttechnologyinsider.com/fbi-identifies-biggest-cyber-threats-as-iot-ransomware-compromised-email/{#}.XG05SPrPw2w
[2] 2019. PewDiePie hackers take over Google smart TV systems. https://www.bbc.com/news/technology-46746592
[3] Ketan Devadiga. 2011. IEEE 802.15.4 and the Internet of Things. (2011).
[4] Arbaugh et al. 1997. A secure and reliable bootstrap architecture. In *SP '97*.
[5] Asokan et al. 2015. SEDA: Scalable Embedded Device Attestation. In *CCS '15*.
[6] Ambrosin et al. 2017. Toward Secure and Efficient Attestation for Highly Dynamic Swarms: Poster. In *WiSec '17*.
[7] Ambrosin et al. 2018. PADS: Practical Attestation for Highly Dynamic Swarm Topologies. In *SIoT*.
[8] Ambrosin et al. 2020. Collective Remote Attestation at the Internet of Things Scale: State-of-the-Art and Future Challenges. *IEEE Commun. Surv. Tutor.* 22, 4 (2020).
[9] Ankergård et al. 2021. State-of-the-Art Software-Based Remote Attestation: Opportunities and Open Issues for Internet of Things. *Sensors* 21, 5 (2021).
[10] Conti et al. 2019. RADIS: Remote Attestation of Distributed IoT Services. In *SDS 2019*. 25–32.
[11] Conti et al. 2020. Attestation-enabled secure and scalable routing protocol for IoT networks. *Ad Hoc Networks* 98 (2020).
[12] Dushku et al. 2020. SARA: Secure Asynchronous Remote Attestation for IoT Systems. *IEEE Trans. Inf. Forensics Secur.* 15 (2020).
[13] Eldefrawy et al. 2012. SMART: Secure and Minimal Architecture for (Establishing Dynamic) Root of Trust.. In *NDSS '12*.
[14] Ibrahim et al. 2017. SeED: Secure Non-Interactive Attestation for Embedded Devices. In *WiSec '17*.
[15] Rabbani et al. 2019. SHeLA: Scalable Heterogeneous Layered Attestation. *IEEE Internet of Things J.* 6, 6 (2019).
[16] Seshadri et al. 2004. SWATT: Software-based attestation for embedded devices. In *IEEE S&P '04*.
[17] Susan Evans. 2019. Mirai Botnet DDoS Attack Type. https://www.corero.com/resources/ddos-attack-types/mirai-botnet-ddos-attack.htm.
[18] Wired. 2010. Stuxnet Incident. https://www.wired.com/2014/11/countdown-to-zero-day-stuxnet.
[19] Østergaard et al. 2021. ERAMO: Effective Remote Attestation through Memory Offloading. In *IEEE CSR*. 73–80.