POSTER: Toward a Secure and Scalable Attestation

[Extended Abstract]

Moreno Ambrosin[1], Mauro Conti[1], Ahmad Ibrahim[2], Gregory Neven[3], Ahmad-Reza Sadeghi[2], and Matthias Schunter[4]

[1]University of Padua, Italy [2]Technische Universität Darmstadt, Germany
[3]IBM Zurich Research Laboratory, Switzerland [4]Intel Labs, Portland, OR, U.S.A.

{ahmad.ibrahim, ahmad.sadeghi}@trust.tu-darmstadt.de,
{Ambrosin, Conti}@math.unipd.it, matthias.schunter@intel.com,
NEV@zurich.ibm.com

ABSTRACT

Large numbers of smart devices are permeating our environment to collect data and act on the insight derived. Examples of such devices include smart homes, factories, cars, or wearables. For privacy, security, and safety, ensuring correctness of the configuration of these devices is essential. One key mechanism to protect the software integrity of these devices is attestation.

In this paper, we analyze the requirements for efficient attestation of large numbers of interconnected embedded systems. We present the first collective attestation protocol which allows attesting an unlimited number of devices. Simulation results show a run-time of 5.3 seconds in networks of 50, 000 low-end embedded devices.

1. INTRODUCTION

Smart devices are rapidly proliferating into every domain of our life. These devices range from tiny wearables to large industrial installations such as, smart factories. Unlike traditional computers, smart devices usually lack the necessary security capabilities which protect them against attacks. Today, an adversary can easily attack such devices and compromise both privacy and safety [5]. One key mechanism to prevent such attacks and ensure the safe and secure operation of a device, is *remote software attestation*.

While today attestation can be performed on individual smart devices, there is no viable approach to securely scale attestation to *a very large number* of devices. Indeed, the first attempt in this direction, SEDA [1], assumes a software-only attacker, i.e., all the devices in the network are not physically tampered. This assumption is not realistic, in the envisioned large scale deployments.

In this paper we present the first collective attestation scheme for large networks of embedded devices that is: *secure*, *scalable*, and *publicly verifiable*.

2. RELATED WORK

Individual Device Attestation is a well-established research area. The purpose of an attestation protocol is to enable a verifier to verify the software integrity of remote device (denoted by prover). We distinguish three main approaches of attestation: (1) software-based attestation, which requires no secure hardware and does not rely on cryptographic secrets, making it particularly attractive for low-end devices with limited resources. Unfortunately, the security of software-based attestation has been challenged; (2) co-processor-based attestation, which offers improved security guarantees. However, due to their high cost and complexity, they are not suitable for low-end embedded devices; and (3) hardware/software co-design [3], which aims at minimizing the hardware security features required for enabling secure remote attestation. Such security features can be as simple as a Read Only Memory (ROM), and a simple Memory Protection Unit (MPU).

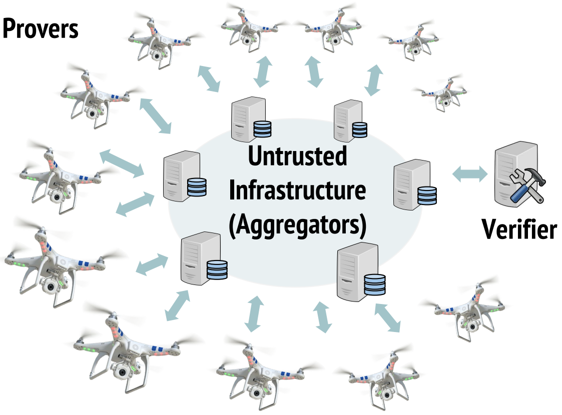

Figure 1: Representation of our system model

Collective Attestation. SEDA [1], made a first step towards a *collective attestation*. However, The main focus of SEDA is efficiency and applicability to low-end embedded devices, rather than security in the presence of a realistic adversary. SEDA is based on neighbors verification and hop-by-hop MACs for authentication. Consequently, every device in SEDA is supposed to be equipped with the minimal hardware required for attestation. Moreover, an adversary which compromises the software of a large number of devices can evade detection by SEDA through physically tampering with one single device in the network. Our proposed solution, which is based on multi-signatures, allows devices with no security hardware to participate in the protocol. Physically tampered devices, on the other hand, can only evade their own detection.

WiSec'16 July 18-22, 2016, Darmstadt, Germany

ACM ISBN 978-1-4503-4270-4/16/07.

DOI: http://dx.doi.org/10.1145/2939918.2942425

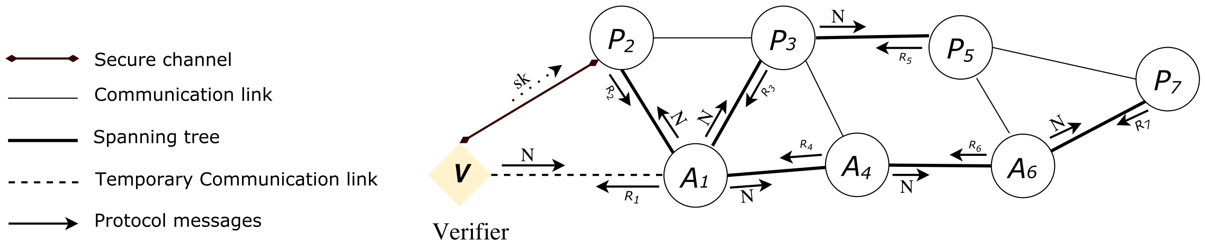

Figure 2: Collective attestation in a network of seven devices (three aggregators and four provers)

Multi Signature. A multi-signature scheme [2] allows n different signers to sign the *same* message m in a constant-size signature, i.e., with signature length independent of n. Most multi-signature schemes also have verification time quasi-independent of n, meaning that the number of core cryptographic operations (e.g., exponentiations or pairing computations) is independent of n.

3. REQUIREMENTS

A collective attestation protocol should satisfy the following requirements:

- *Security:* The protocol should be secure in the presence of a strong adversary capable of physical tampering, i.e., the attestation result of one devices should not be dependent on the hardware security of any other device.
- *Scalablity*, The protocol should efficiently verify the integrity of a large collection of devices. The run-time of the protocol should be at most logarithmic in the size of the network.
- *Public Verifiablity*, i.e., the produced attestation report should be publicly verifiable

4. NETWORK ATTESTATION

Our protocol combines attestation trees with a Boldyreva's multi-signature scheme [2]. It thus provides secure collective attestation with *constant* overhead on the verifier and logarithmic overall run-time. This allows even low power verifier devices, such as a smartphone, to verify the integrity of very large industrial or IoT setups. The collected result is publicly verifiable, and ensures that every devices that is not physically tampered, is also not software-compromised.

The proposed protocol is executed between the following entities: *prover* (P), *aggregator* (A), and *verifier* ($\mathcal{V}$). As shown in Figure 2, each device is initialized before deployment (by $\mathcal{V}$) with a multi-signature secret key, to which $\mathcal{V}$ stores the public key. At attestation time, $\mathcal{V}$ randomly chooses an aggregator device A_1 and sends it a random challenge N. Upon receiving the challenge, each device forwards it to its neighbors, until the challenge is received by every device in the network. Consequently, a spanning tree rooted at A_1 is formed. Finally, starting at leaf nodes in the tree, every prover P_i composes a proof of integrity of its software configuration, (e.g., hash of its binary). It then generates a multi-signature over the software configuration and the received challenge. P_i sends the generated signature to its parent node as an attestation response R_i. Upon receiving all responses from its child nodes, an aggregator A_j aggregates the received multi-signatures according to the definition in [2]. The generated multi-signature R_j is then forwarded to the parent node. As a result, the final multi-signature R_1 is generated by A_1 and then forwarded to $\mathcal{V}$. Having the public keys of all devices in the network, $\mathcal{V}$ can verify the received multi-signature in constant time. If the signature verifies correctly, $\mathcal{V}$ concludes that every (physically untampered) device in the network is not software-compromised.

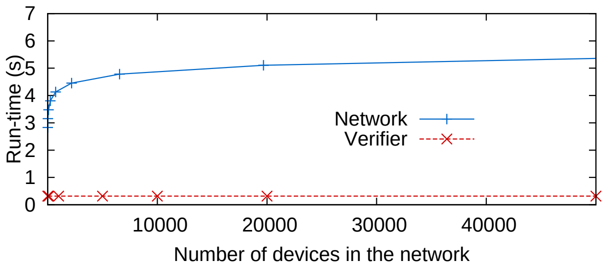

Figure 3: run-time in networks with 4 neighbors per device

5. PERFORMANCE ANALYSIS

We simulated our protocol using the OMNeT++ [4] simulation environment for networks with up to $50,000$. Each device in the network has four neighbors. We based our simulation on measurement of the real execution time on both TyTAN [3] security architecture as prover devices and an Intel Galileo board as the verifier. The run-time of the protocol at both ends (i.e., network and verifier) is shown in Figure 3. As shown in the figure the run-time of the protocol (as function of the network size) is logarithmic at the network's end and constant on the verifier. The overall time required to attest a network of $50,000$ devices is about 5.3 seconds.

6. CONCLUSIONS

Collective attestation is a building block for securing the Internet of Things. For very large numbers of devices, to enables enterprises to validate the configuration and software and ensure that all devices are indeed up-to-date. In this paper, we have proposed the first practical and secure collective attestation scheme. It substantially improves the state of the art (e.g. SEDA [1]) by allowing aggregators that are not equipped with security hardware. our protocol is also resilient to a strong adversary, which is capable of physical tampering. Unfortunately, it only allows collective attestation of homogeneous networks with constant run-time. For future work, we aim to design a collective attestation protocol which enables secure, and efficient attestation of heterogeneous networks.

Acknowledgement

This work has been co-funded by the German Science Foundation as part of project S2 within the CRC 1119 CROSSING, EC-SPRIDE, the European Union's Seventh Framework Programme under grant agreement No. 609611, PRACTICE project, and the Intel Collaborative Research Institute for Secure Computing (ICRI-SC).

7. REFERENCES

[1] N. Asokan et al. Seda: Scalable embedded device attestation. In *ACM CCS'15*.

[2] A. Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme. In *PKC '03*.

[3] F. Brasser et al. Tytan: Tiny trust anchor for tiny devices. In *DAC'15*.

[4] OpenSim Ltd. OMNeT++ discrete event simulator. http://omnetpp.org/, 2015.

[5] J. Vijayan. Stuxnet renews power grid security concerns, 2010.