On the Feasibility of Remote Attestation for IoT Devices

Yong-Hyuk Moon, Jeong-Nyeo Kim, and Yong-Sung Jeon
Hyper-connected Communication Research Laboratory
Electronics and Telecommunication Research Institute (ETRI)
Daejeon, Republic of Korea
email: {yhmoon, jnkim, ysjeon}@etri.re.kr

***Abstract*—This paper reviews practical difficulty of deploying conventional remote attestation mechanisms into Internet-of-Things. We then suggest a new research direction for highly feasible attestation in terms of six identified perspectives.**

Keywords-remote attestation; code integrity; device security.

I. INTRODUCTION

These days, device security is a growing concern with proliferation of low-power embedded devices. Especially, malware injection has become a critical threat even to small footprint devices, e.g., Internet-of-Things (IoT). Once a device is infected or compromised, unauthorized software can send confidential data to an external entity, force the device to operate abnormally, and induce harmful activities in an unpredictable manner. This creates several challenges, so that flawless design and implementation remains a crucial issue in practical system. In this paper, we confine our focus to three objectives: *i*) a brief review on the existing attestation approaches in Section II, *ii*) identifying requirements from challenging issues of attestation in Section III, and *iii*) setting a research direction towards a highly feasible attestation for IoT devices in Section IV.

II. EXISTING APPROACHES TO ATTESTATION

Three lines of attestation schemes have been proposed to convince a verifier of a current system state of device.

A. Hardware Based Attestation

Trusted platform module (TPM) [1], a chip connecting to the microcontroller unit (MCU), is widely used to ensure that a system platform has loaded properly (e.g., secure booting). For this, TPM as the root of trust for measurement offers isolated storage to maintain asymmetric keys and platform configuration registers (PCRs). However, attestation based on such hardware trusted computing base (TCB) is most suitable for more-capable computing devices.

B. Software Only Attestation

As an early effort, PIONEER [2] offers primitive design principles and operations in order to externally verify a code at runtime. On the other hand, a software attestation protocol could be unfeasible due to the three common assumptions: *i*) a target device has been authenticated; thus, means for encrypted communication, secure key storage and so forth are given, *ii*) trustworthiness of prover relies on the predefined time bound for a response to a verifier's challenge, and *iii*) a prover process is strongly protected.

C. Hybrid Approaches

New approaches have been recently developed for establishing a dynamic root of trust with minimal modifications to standard built-in hardware. SMART [3] changes access logic to memory bus in the existing MCU, so that particular read only memory (ROM) resident code only accesses to a protected key for computing measurement. However, memory access violation is not concerned in this scheme. Unlike SMART, memory protection unit (MPU) enforces that only a trustlet constructing an attestation mechanism can access to its data for execution in TrustLite [4]. Secure inter-process communication issue is still a remaining issue.

III. CHALLENGING ISSUES OF DEVICE ATTESTATION

IoT devices are commonly resource-constrained; thus, installing TCB increases the costs of device production and requires additional software (e.g., driver, library). This strategy also increases the overall system complexity and is utterly opposed to the things' characteristics.

A software process loaded on memory can be identified by comparing the measured hash values in attestation with reference data, called a list of reference integrity measurements (RIMs). Despite the simple matching, creating and maintaining RIMs is a challenging task. Furthermore, measurement represents not a security state of code but its execution state. Although a platform state relies on different software configurations, a binary decision of attestation only implies whether measured hash values are correct. Thus, the RIM-based technique may not be valid for detecting buffer overflow and return-oriented programing (ROP) attacks.

On the one hand, a prover can be replaced by malicious codes and its invocation can be hijacked. Precomputation of measured integrity value is also possible. To guarantee the secure state of prover as well as reliability of response, it is required to separate a prover's work space from the other memory regions in a strict manner. Intuitively, it is difficult to verify the large number of devices one by one, that is, considerably time-consuming. Further, a verifier needs to handle devices, which operate on heterogeneous system platforms allowing various software configurations. Conventional attestation is insufficient in terms of scalability.

 ISBN: 978-1-61208-516-6

Since verifier impersonation could be a trivial attack to devices, if a prover believe that a bogus verifier is genuine, fake attestation requests easily invoke the measuring process of prover at any time. This situation acts as Denial of Service (DoS) attack. Thus, software only attestation is especially vulnerable to this setting.

IV. TOWARDS HIGHLY FEASIBLE ATTESTATION

With respect to the aforementioned major concerns, we discuss candidate solutions that can be applied to design a highly feasible remote attestation mechanism for IoT devices.

A. Authentic Requests

In the context of IoT devices, computing a message authentication code is time-consuming and asymmetric key cryptography based on X.509 certificates requires large computational complexity. A recent solution [5] mitigates this limitation by applying nonces, counters and timestamps to the process of authenticating verifier requests in attestation. These data can be effective in detecting reply attacks, reordered requests and delayed requests, respectively, if non-volatile memory is supported and provides a sufficient space.

B. Meausrement Assurance

A measurement result must not be compromised even in a tempered device. To this end, reference data and keys must be protected in the isolated memory space. One possible solution is to use the internal inaccessible ROM in which a bootloader is located. However, such a type of ROM may not be a built-in component to some devices. MPU could be another countermeasure to enforce rules of controlling memory access and permission. Fortunately, this hardware chip is provided by widely used commodity MCU products.

C. Prover Protection

To satisfy minimal hardware support, MPU could be used for prover protection by making a specific region of memory isolate. An isolated region is only accessible by a system module with a privileged mode, so that a set functions of MPU could not be called by a user process. In addition to that, one region can be divided into several blocks according to specific purposes. One critical drawback is caused by the fact that some IoT operating systems do not provide any barrier or means (e.g., system call interfaces) to differentiate user mode and kernel mode.

D. Verfication Flexibility

Since conventional attestation depends on cryptographic algorithms, such as hashing, it is very effective in ensuring whether a binary code running on a device is exactly same as that a verifier expects. Its all-or-nothing strategy does not allow the existence of devices with various degree of trustworthiness, cannot distinguish between identification and behavior of codes, and locks a device into a limited platform. One ultimate goal of new attestation is to obtain a strong evidence that a program on a remote device purely behaves according to a given security policy.

E. Control Flow

To measure and verify the runtime state of particular codes, every control flow of program including stack usage should be traced by TCB. In case of detecting ROP attacks, the last branch record (LBR) may be required to monitor the abnormal branch instructions to some gadget (a small piece of codes). Low-power MCU, such as ARM cortex family is not capable of maintaining the overall history of these instructions due to the absence of LBR. To overcome this problem, a prover can accumulate addresses of source and target of every branch instruction by building a hash chain of branch path, i.e., control flow.

F. Scalability

One common limitation of remote attestation is that a verifier certainly suffers from a performance bottleneck since it cannot scale to diversity of devices. A simple and straightforward approach to mitigate this problem is to attest a group of devices (swarms) instead of dealing with a single device at time of attestation [6]. Devices, meanwhile, can be also verified by rapidly investigating consistency of their relationship, which is created in the form of clique [7]. The matter to consider is that these attestation schemes may be subject to the construction types of topologies.

V. CONCLUSIONS

In this paper, we have reviewed the existing attestation schemes with respect to their limitations. Future research directions and advanced solutions have been also discussed for designing a highly feasible attestation in the IoT system.

ACKNOWLEDGMENT

This work was supported by Institute for Information and communication Technology Promotion (IITP) grant funded by the Korea government (MSIP) [B0190-16-2032, Development of Operating System Security Core Technology for the Smart Lightweight IoT Devices].

REFERENCES

[1] Trusted Computing Group. TPM Main Specification Level 2 Version 1.2, Revision 116, March 1 2011.

[2] Arvind Seshadri et al., “Pioneer: Verifying Code Integrity and Enforcing Untampered Code Execution on Legacy Systems,” SOSP’05, pp. 1-16, October 23-26, 2005, United Kingdom.

[3] E. Karim, F. Aurélien, P. Daniele, and T. Gene, “SMART: Secure and Minimal Architecture for (Establishing a Dynamic) Root of Trust,“ NDSS’12, February 5-8, USA.

[4] P. Koeberl, S. Schulz, A.-R. Sadeghi, and V. Varadharajan, “TrustLite: A Security Architecture for Tiny Embedded Devices,” EuroSys’14, April 13-16, 2014.

[5] F. Brasser, K. B. Rasmussen, A.-R. Sadeghi, and G. Tsudik, “Remote Attestation for Low-End Embedded Devices: the Prover’s Perspective,” DAC ’16, June 05-09, 2016, USA.

[6] N. Asokan et al., “SEDA: Scalable Embedded Device Attestation,” CCS’15, pp. 964-975, October 12–16, 2015.

[7] Y.-H. Moon and Y.-S. Jeon, “A Functional Relationship Based Attestation Scheme for Detecting Compromised Nodes in Large IoT Networks,” CUTE’15, vol. 373, pp. 713-721, December 2015.

 ISBN: 978-1-61208-516-6