Remote Attestation

Subjects: Computer Science, Information Systems

Contributor: Edlira Dushku

Remote Attestation (RA) is a security mechanism to remotely detect adversarial presence on untrusted devices in order to guarantee their trustworthiness. RA runs as a two-party security protocol in which a trusted party (i.e., verifier) assures the integrity of the untrusted remote device (i.e., prover). Software-based RA approaches aim at verifying device integrity without relying on specialized hardware components. Despite their limited security guarantees, software-based RA approaches bring opportunities in attesting legacy and resource-constrained Internet of Things (IoT) devices, in which the presence of a hardware root-of-trust is not always a realistic assumption.

remote attestation software-based attestation legacy Internet of Things Battery-Free IoT

Internet of Things security IoT Trust

1. Introduction

With the Internet of Things (IoT) revolution, IoT devices are experiencing an exponential growth, becoming pervasive in infrastructure and industrial systems (e.g., digital transportation, smart cities, automated factories), and emerging as an integral part of our everyday life (e.g., smart home, wearable devices). According to Statista (https://www.statista.com/statistics/976313/global-iot-market-size/ (accessed on 31 December 2020)), the global IoT market is expected to reach around 1.6 trillion dollars in market revenue by 2025. However, the enormous expansion of interconnected IoT devices that perform safety-critical operations and contain sensitive information, combined with their limited capabilities to implement advanced security techniques, makes IoT devices a prominent target of a broad range of malicious exploitations [1][2][3].

Aimed at securing IoT devices, Remote Attestation (RA) has been proposed as a valuable security technique that allows a trusted party (i.e., verifier) to assure the integrity of the untrusted IoT device (i.e., prover). During the attestation, the prover sends proofs about its current state of the memory (typically a hash of the memory) to the verifier, whereas the verifier matches the received evidence with the expected legitimate state (known in advance) of the prover, and according to that it validates whether the prover is trustworthy or not.

Based on their architectural design, RA schemes can broadly be classified into three main categories: (1) Software-based RA (e.g., Seshadri et al. [4][5]) which provides security guarantees based on strict running time constraints of the verification procedure; (2) Hardware-based RA (e.g., Sailer et al. [6], Tan et al. [7]) which uses a tamper-resistant hardware module as a secure execution environment; and (3) Hybrid RA (e.g., Eldefrawy et al. [8], Brasser et al. [9])

which rely on a minimal read-only hardware-protected memory. Due to the lack of requirements for a specialized tampered-resistance hardware, software-based RA schemes are low-cost solutions in comparison with hardware-based RA. However, using a secure execution environment such as Trusted Platform Module (TPM) [10], ARM TrustZone [11], and Intel Software Guard Extensions (SGX) (https://software.intel.com/en-us/sgx (accessed on 31 December 2020)), hardware-based RA provides high-security guarantees, that protects RA protocol execution from compromised software. Nevertheless, classic low-cost IoT devices do not support the requirements of hardware-based schemes for costly specialized hardware-protected modules. To ensure uninterrupted, safe and secure code execution of the RA protocol, hybrid RA schemes depend on the existence of a minimal read-only hardware-protected memory. However, the assumption made by hardware-based RA and hybrid RA of a specialized hardware is not a trivial requirement for many IoT devices with limited computational power which do not support any specialized hardware, such as battery-free, energy harvesting IoT devices [12].

Considering that there is a great number of legacy IoT devices already deployed without a specialized hardware support, it is difficult (if not impractical) to customize the hardware and redeploy these devices. Due to the cost, it is also not a viable option to replace them all with new devices relying on specialized hardware. In addition, many IoT devices are designed to be small, cheap, and battery-free, thus, introducing new and specialized hardware could potentially not only increase the cost and size of the devices but also deviate from the energy harvesting feature of their design. Nevertheless, it is crucial to provide security protections on such low-cost devices. In this context, software-based RA can be considered a very promising approach. However, to the best of our knowledge, a comprehensive analysis of existing software-based RA schemes in order to investigate their advantages and disadvantages along with the opportunities that they offer for attesting legacy and/or resource-constrained IoT systems is still missing in the literature.

2. Opportunities of Software-Based RA Schemes

Software-based RA protocols have been abandoned in the most recent RA proposals as they are considered deprived of necessary security guarantees. However, the lightweight design of such protocols could be of great value for various already-deployed IoT solutions or new commercial IoT products. In the following, we discuss some opportunities that software-based RA approaches bring in enabling attestation on different categories of very lightweight IoT devices.

2.1. Legacy Devices

With the large number of IoT devices deployed over the past years, many IoT devices currently in use are legacy devices. Most legacy IoT devices were designed to operate unconnected, standalone, and the adoption of novel security solutions are often impractical for such devices. Considering the unique characteristics of legacy IoT devices that typically lack complete and accurate documentation, it becomes crucial to bring RA’s benefits to such legacy devices without disrupting their existing operations. In this context, the adoption of hardware or hybrid RA schemes requiring specialized hardware support or customized hardware configuration is impractical for legacy IoT devices. In contrast, the software-based RA approaches are suitable for legacy devices as they rely only on software. Even though software-based RA protocols are vulnerable to sophisticated attacks, software-based RA protocol could still provide some degree of integrity guarantees in these devices. Under certain assumptions such as legacy devices deployed in a private and relatively-small network, the software-based approaches such as SWATT [4], Pioneer [5] and LRMA [13] are a promising solution for the missing security mechanisms present on resource-constrained legacy IoT devices.

2.2. Battery-Free Devices

Europe has recently entered into the green transition, which aims at lowering global energy footprint towards achieving the ultimate goal of being climate-neutral by 2050. As a result, the deployment of battery-free IoT devices [12] is expected to be increased in the upcoming years. In this context, the RA protocols that rely on customized hardware not cause an increased cost and size of any resource constraint IoT devices and deviate from the initial core objective of the original energy-harvesting design of battery-free IoT devices. While typically the IoT networks of such tiny devices adopt correlated information to detect compromised devices, such battery-free devices could benefit from software-based RA schemes as an integrity check mechanism. However, the software-based RA protocols that perform expensive computational operations and rely on strict time constraints could be heavy for such devices. The most suitable protocols for energy harvesting devices could be the software-based RA protocols that rely on loosely time constraints listed in Table 2 such as [14].

2.3. Fog Computing

Due to strict time constraints, software-based RA schemes have been considered limited to a one-hop network setting and unsuitable for attestation of large networks with multi-hop distance between the verifier and provers. However, with the emerging paradigm of Fog computing (https://www.openfogconsortium.org/ (accessed on 31 December 2020)), there comes the opportunity to introduce single-hop attestation schemes between these devices and a connected Fog node, that can act as a verifier. The software-based RA schemes have been considered impractical due to the strong assumptions of the required verifier's knowledge to validate the legitimate state of IoT devices, for instance knowing the exact hardware configuration. Table 1 presents an overview of the required knowledge by the verifier. In a Fog computing infrastructure, each Fog node serves as a distributed verifier, the assumption that each Fog node has all the required knowledge of the devices connected to the Fog node seems realistic. Thus, each Fog node may attest its device by performing a software-based RA scheme. However, software-based RA schemes are challenging in mobile networks in which devices frequently join and leave different Fog nodes.

Table 1. Overview of software-based RA schemes w.r.t. required verifier knowledge.

Scheme	Mem. Cont.	Exact HW Config.	Network Delay	Used Mem.	Checksum
Reflection [15], Dataguard [16]	✓	✗	✗	✗	✗
SWATT [4], Pioneer [5], LRMA [13]	✓	✓	✓	✗	✗
PIV [17]	✓	✗	✗	✗	✗
Self-Modifying Code [18]	✗	✓	✗	✗	✗
Proactive [19], Distributed 1 [20], USAS [21]	✗	✗	✗	✓	✗
Distributed 2	✗	✗	✗	✗	✓
Memory Filling [14]	✗	✓	✗	✓	✗
Lightweight [22]	✗	✓	✗	✓	✗

2.4. IoT Applications

Software-based RA schemes serve as building blocks for other crucial software-based security mechanisms such as key establishment [23], security software update [24], recovery [25] and secure erasure [26]. With the IoT devices playing a remarkable role in many domains such as healthcare, vehicles and transportation systems, industrial appliances, and smart homes, the cutting edge of security is continually being pushed. Recent works in the literature have integrated RA with Blockchain to provide stronger security guarantees (e.g., decentralization, traceability, anonymity and non-repudiation) for critical real-time infrastructures such as Vehicle-to-Vehicle communications [27]. Other promising applications include the trustworthy collaboration among Automated Guided Vehicles in the mobile and collaborative Smart Factory context [28].

References

1. Kolias, C.; Kambourakis, G.; Stavrou, A.; Voas, J. DDoS in the IoT: Mirai and Other Botnets. Computer 2017, 50, 80–84.
2. Fernandes, E.; Jung, J.; Prakash, A. Security Analysis of Emerging Smart Home Applications. In Proceedings of the 2016 IEEE Symposium on Security and Privacy SP ’16, San Jose, CA, USA, 22–26 May 2016.
3. Neshenko, N.; Bou-Harb, E.; Crichigno, J.; Kaddoum, G.; Ghani, N. Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT

Exploitations. IEEE Commun. Surv. Tutor. 2019, 21, 2702–2733.

4. Seshadri, A.; Perrig, A.; van Doorn, L.; Khosla, P. SWATT: SoftWare-based ATTestation for embedded devices. Proc. IEEE Symp. Secur. Privacy 2004, 2004, 272–282.
5. Seshadri, A.; Perrig, A.; Luk, M.; van Doom, L.; Shi, E.; Khosla, P. Pioneer: Verifying code integrity and enforcing untampered code execution on legacy systems. Operating Syst. Rev. (ACM) 2005, 39, 1–16.
6. Sailer, R.; Zhang, X.; Jaeger, T.; van Doorn, L. Design and Implementation of a TCG-based Integrity Measurement Architecture. In Proceedings of the 13th Conference on USENIX Security Symposium SSYM’04, San Diego, CA, USA, 9–13 August 2004.
7. Tan, H.; Hu, W.; Jha, S. A Remote Attestation Protocol with Trusted Platform Modules TPMs in Wireless Sensor Networks. Sec. Commun. Netw. 2015, 8, 2171–2188.
8. Eldefrawy, K.; Tsudik, G.; Francillon, A.; 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, San Diego, CA, USA, 5–8 February 2012.
9. Brasser, F.; El Mahjoub, B.; Sadeghi, A.R.; Wachsmann, C.; Koeberl, P. TyTAN: Tiny trust anchor for tiny devices. In Proceedings of the 52nd Design Automation Conference, San Francisco, CA, USA, 8–12 June 2015; pp. 1–6.
10. Arthur, W.; Challener, D. A Practical Guide to TPM 2.0: Using the Trusted Platform Module in the New Age of Security; Apress: Berkeley, CA, USA, 2015.
11. Yiu, J. ARMv8-M Architecture Technical Overview; ARM White Paper; ARM: Cambridge, England, UK, 2015.
12. Koetsier, J. Battery-Free IoT: These Tiny Printable Computers Harvest Energy From Radio Waves. 2020. Available online: (accessed on 31 December 2020).
13. Yang, X.; He, X.; Yu, W.; Lin, J.; Li, R.; Yang, Q.; Song, H. Towards a low-cost remote memory attestation for the smart grid. Sensors 2015, 15, 20799–20824.
14. AbuHmed, T.; Nyamaa, N.; Nyang, D.H. Software-based remote code attestation in wireless sensor network. In Proceedings of the GLOBECOM - IEEE Global Telecommunications Conference, Honolulu, HI, USA, 30 November–4 December 2009.
15. Spinellis, D. Reflection as a Mechanism for Software Integrity Verification. ACM Trans. Inf. Syst. Secur. 2000, 3, 51–62.
16. Zhang, D.; Liu, D. DataGuard: Dynamic Data Attestation in Wireless Sensor Networks. In Proceedings of the 2010 IEEE/IFIP International Conference on Dependable Systems & Networks (DSN), Chicago, IL, USA, 28 June–1 July 2010; pp. 261–270.

17. Park, T.; Shin, K.G. Soft tamper-proofing via program integrity verification in wireless sensor networks. IEEE Trans. Mobile Comput. 2005, 4, 297–309.
18. Shaneck, M.; Mahadevan, K.; Kher, V.; Kim, Y. Remote Software-Based Attestation for Wireless Sensors In Security and Privacy in Ad-hoc and Sensor Networks; Molva, R., Tsudik, G., Westhoff, D., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; pp. 27–41.
19. Ahn, S.; Chong, K. Requirements Change Management on Feature-Oriented Requirements Tracing; Springer: Berlin/Heidelberg, Germany, 2007; Volume 4706 LNCS, pp. 296–307.
20. Yang, Y.; Wang, X.; Zhu, S.; Cao, G. Distributed Software-based Attestation for Node Compromise Detection in Sensor Networks. In Proceedings of the 2007 26th IEEE International Symposium on Reliable Distributed Systems (SRDS 2007), Beijing, China, 10–12 October 2008; pp. 219–230.
21. Jin, X.; Putthapipat, P.; Pan, D.; Pissinou, N.; Makki, S.K. Unpredictable software-based attestation solution for node compromise detection in mobile WSN. In Proceedings of the 2010 IEEE Globecom Workshops, GC’10, Miami, FL, USA, 6–10 December 2010; pp. 2059–2064.
22. Kiyomoto, S.; Miyake, Y. Lightweight attestation scheme for wireless sensor network. Int. J. Secur. Its Appl. 2014, 8, 25–40.
23. Seshadri, A.; Luk, M.; Perrig, A. SAKE: Software attestation for key establishment in sensor networks. Ad Hoc Netw. 2011, 9, 1059–1067.
24. Seshadri, A.; Luk, M.; Perrig, A.; van Doorn, L.; Khosla, P. SCUBA: Secure Code Update By Attestation in Sensor Networks. In Proceedings of the 5th ACM Workshop on Wireless Security; Association for Computing Machinery: New York, NY, USA, 2006; pp. 85–94.
25. Pietro, R.D.; Ma, D.; Soriente, C.; Tsudik, G. POSH: Proactive co-Operative Self-Healing in Unattended Wireless Sensor Networks. In Proceedings of the 2008 Symposium on Reliable Distributed Systems, Naples, Italy, 6–8 October 2008; pp. 185–194.
26. Perito, D.; Tsudik, G. Secure Code Update for Embedded Devices via Proofs of Secure Erasure. In Computer Security—ESORICS 2010; Gritzalis, D., Preneel, B., Theoharidou, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 643–662.
27. Xu, C.; Liu, H.; Li, P.; Wang, P. A Remote Attestation Security Model Based on Privacy-Preserving Blockchain for V2X. IEEE Access 2018, 6, 67809–67818.
28. Fortino, G.; Messina, F.; Rosaci, D.; Sarné, G.M.L.; Savaglio, C. A Trust-Based Team Formation Framework for Mobile Intelligence in Smart Factories. IEEE Trans. Ind. Inform. 2020, 16, 6133–6142.