Standards-based Remote Attestation for Internet-of-Things Swarms

Yuxuan Song, Mališa Vučinić, Thomas Watteyne

▶ **To cite this version:**

Yuxuan Song, Mališa Vučinić, Thomas Watteyne. Standards-based Remote Attestation for Internet-of-Things Swarms. IEEE International Conference on Robotics and Automation (ICRA), Breaking Swarm Stereotypes workshop, May 2024, Yokohama, Japan. hal-04630429

HAL Id: hal-04630429
https://inria.hal.science/hal-04630429

Submitted on 1 Jul 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Standards-based Remote Attestation for Internet-of-Things Swarms

Yuxuan Song, Mališa Vučinić, Thomas Watteyne
Inria Paris, France
first.last@inria.fr

***Abstract*—Remote attestation is a security service to verify and confirm the integrity and trustworthiness of each robot in the swarm. We developed an efficient method of doing remote attestation in parallel with the network access authentication, utilizing a newly standardized key exchange protocol named EDHOC. We demonstrate the feasibility of this method through benchmarks on the hashing time of 520 kB under constrained scenarios, resulting in times of 780 ms on the nRF52840 and 728 ms on the nRF5340.**

I. Introduction

To mitigate the risk of cyberattacks on robotic swarms, only legitimate robots should be admitted into the swarm. Let's imagine a robot in which an attacker has altered the firmware, but which still has valid authentication credentials such as a digital certificate and the corresponding private key. This allows it to still authenticate to the swarm operator. *How to ensure that only robots with verified and trustworthy firmware are allowed to join the swarm?*

The solution is to do a remote attestation. Attestation service running on a robot enables it to generate the evidence about the state of its software and hardware. By sending this evidence to a remote entity called Verifier, the swarm operator can assess whether the robot is compromised.

For a swarm of robots, the remote attestation procedure should be both energy and bandwidth efficient. A lightweight security protocol named Ephemeral Diffie-Hellman over COSE (EDHOC) [1] is standardized recently. EDHOC is a highly compact and efficient protocol that enables authenticated key exchange in constrained scenarios.

In this paper, we present a lightweight method to perform the remote attestation using EDHOC. The attestation procedure is integrated into EDHOC handshake session. An evaluation is achieved on a swarm robotic platform to prove the feasibility of this method. We benchmarked the time to hash the bytes in firmware image size on nRF52840 microcontroller and nRF5340 microcontroller. The results are 780 ms on the nRF52840 and 728 ms on the nRF5340 for hashing 520 kB, which shows that this method can be deployed even on the constrained platforms.

II. A Primer on Remote Attestation

Remote attestation [2] is a security process that can help the swarm central server establish a level of trust in the robot before allowing the robot to join the swarm.

A standardized way of doing remote attestation involves three entitites. The *Attester* provides reliable evidence about the state of itself. The *Verifier* evaluates the evidence and produces attestation results. The *Relying Party* consumes the attestation results to execute application-specific actions.

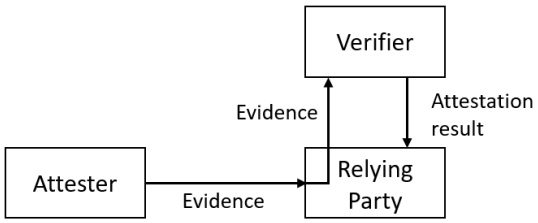

Fig. 1. Remote attestation architecture.

The architecture for doing remote attestation is shown in Fig. 1. The Relying Party conveys the evidence from the Attester to the Verifier. The Verifier then sends back the attestation result to the Relying Party.

III. Ephemeral Diffie-Hellman over COSE (EDHOC) protocol

EDHOC [1] was recently standardized in the Internet Engineering Task Force (IETF). It is a key exchange protocol based on ephemeral elliptic curve Diffie-Hellman (ECDH) keys. The execution of EDHOC takes place between an Initiator and a Responder. The protocol consists of three mandatory messages(message_1, message_2, and message_3) and an optional fourth message(message_4).

EDHOC also allows external security applications to be integrated in the handshake by defining the External Authorization Data (EAD) field. EADs are sent in dedicated fields of the EDHOC messages: EAD_1, EAD_2 and EAD_3.

Fedrecheski*et al.* [3] have compared EDHOC with Datagram Transport Layer Security (DTLS) protocol 1.3, a widely used protocol for key exchange on the Internet. Their study comprises several metrics, including message footprint, time, energy and memory. Results indicate that, in constrained IoT networks, EDHOC outperformed DTLS in all metrics. Compared to DTLS, EDHOC exhibits a 7× reduction in time-on-air, a 1.44× reduction in handshake duration, a 2.79× reduction in energy consumption, and a 4× reduction in flash and RAM memory usage.

IV. Remote Attestation over EDHOC

We propose to do remote attestation in parallel with network access authentication. In that setting (in Fig. 2), we propose

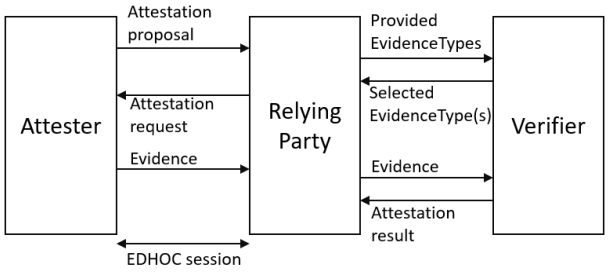

Fig. 2. Overview of message flow.

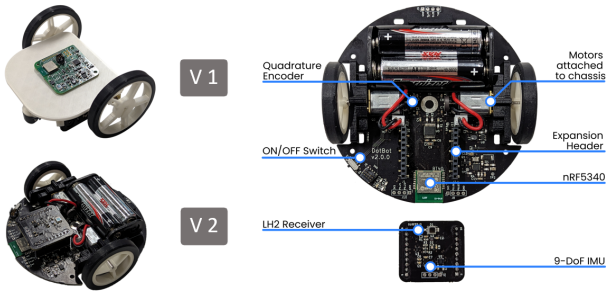

Fig. 3. Dotbot platform.

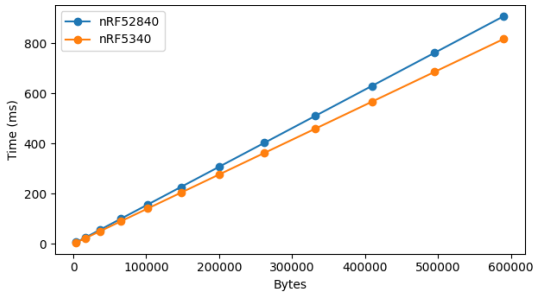

Fig. 4. Benchmark of a software implementation of SHA-256 on a constrained swarm robotic platform DotBot.

to map the EDHOC Initiator to the Attester, and the EDHOC Responder to the Relying Party. The Verifier is hosted on the central swarm server.

The Attester and the Relying Party communicate by transporting messages within EDHOC's External Authorization Data (EAD) fields. The Attester sends an attestation proposal in EDHOC's EAD_1 field, which contains all the supported evidence types by the Attester. The evidence type indicates the format type and an appropriate profile of the evidence. For example, an evidence type indicates that the evidence is formatted as CBOR Web Token (CWT), with a profile of Platform Security Architecture (PSA) claims [4].

If the Verifier can support at least one of the provided evidence types, the Relying Party signals to the Attester an attestation request in EDHOC's EAD_2 field. The Verifier specifies a selected evidence type, and a Nonce in order to guarantee the message freshness.

The Attester conveys the evidence to the Relying Party in EDHOC's EAD_3 field. The evidence is defined as an attestation token [5], an attested claims set. The information about the state of robot softwares, such as the active firmware image or the software configurations are carried within the evidence. The value of the measurement must be a hash of 32, 48 or 64 bytes. This hash, together with other useful information is then signed or symmetrically protected with a MAC to construct the evidence. The evidence is then sent to and evaluated by the Verifier.

V. Evaluation

To evaluate the feasibility of doing remote attestation in a constrained swarm, we benchmark the execution of a SHA-256 hash function on the *DotBot* swarm robot (Fig. 3). DotBot is a cm-scale micro-robot that is developed for education and research purposes. All DotBot developments are done under an open source hardware and software licenses. The first version runs on the nRF52840 microcontroller and the second version runs on the nRF5340 microcontroller.

We evaluate the time to hash a string of bytes about the same size as a firmware image. The maximum size of a firmware image is 520 kB. The results are shown in Fig. 4: the time to hash increases linearly with number of bytes. We can see that the hash of 520 kB takes approx. 780 ms on the nRF52840, approx. 728 ms on the nRF5340. This result shows the feasibility of hashing the entire firmware image even on the constrained platforms. This does not need to be done in real time; attestation is expected to be done only a handful of times throughout the lifetime of the robot.

VI. Conclusions

Remote attestation is an essential process for a remote entity to verify the trustworthiness of a device joining the swarm. Remote attestation also ensures that the device remains uncompromised and untampered while within the swarm. In this paper we present a secure and lightweight approach to perform remote attestation over EDHOC, a recently standardized lightweight key exchange protocol.

Acknowledgements

This project has received funding from the European Union's Horizon Europe Framework Programme under Grant Agreement No. 101093046.

References

[1] G. Selander, J. Preuß Mattsson, and F. Palombini, *Ephemeral Diffie-Hellman Over COSE (EDHOC)*, Internet Engineering Task Force (IETF) Std. RFC9528, 2024.

[2] H. Birkholz, D. Thaler, M. Richardson, N. Smith, and W. Pan, *Remote ATtestation procedureS (RATS) Architecture*, Internet Engineering Task Force (IETF) Std. RFC9334, 2023.

[3] G. Fedrecheski, M. Vučinić, and T. Watteyne, "Performance Comparison of EDHOC and DTLS 1.3 in Internet-of-Things Environments," in *IEEE Wireless Communications and Networking Conference (WCNC)*, Dubai, United Arab Emirates, 21-24 Apr 2024.

[4] H. Tschofenig, S. Frost, M. Brossard, A. L. Shaw, and T. Fossati, *Arm's Platform Security Architecture (PSA) Attestation Token*, Internet Engineering Task Force (IETF) Std. draft-tschofenig-rats-psa-token-22, 2024.

[5] L. Lundblade, G. Mandyam, J. O'Donoghue, and C. Wallace, *The Entity Attestation Token (EAT)*, Internet Engineering Task Force (IETF) Std. draft-ietf-rats-eat-25, 2024.