Support Remote Attestation for Decentralized Robot Operating System using Trusted Execution Environment

Author1
Address line 1
Address line 2
Address line 3
Address line 4
author1_email

Author2
Address line 1
Address line 2
Address line 3
Address line 4
author2_email

Author3
Address line 1
Address line 2
Address line 3
Address line 4
author3_email

***Abstract*— The surge in autonomous robot deployments across diverse domains is undeniable. The Robot Operating System (ROS) stands out as the prevailing standard for robotics systems, with ROS 2 emerging as its revitalized version. ROS 2 uses Data Distribution Service (DDS) as its communication middleware, aligning itself with the blossom of decentralized and distributed smart systems. However, the security of ROS 2 is dependent on the implementation of the DDS security plugins, which provides domain-level access protection under the assumption of trust in local machines. The problem arises when a malicious ROS 2 node, compromised by malware, can disseminate false information or pilfer sensitive data from other legitimate nodes within the system. To address this vulnerability, this paper harnesses the Trusted Execution Environment (TEE) to build a trustworthy ROS 2 platform with remote attestation. The proposed solution not only verifies the identity but also ensures the integrity of ROS 2 nodes before they provide/consume data and/or collaborate with each other. Our design establishes trust between communication parties and improves the ROS 2 security by incorporating the hardware level protection.**

Keywords—Remote Attestation, Robot Operating System, Data Distribution Service, Trusted Execution Environment

I. Introduction

Smart and autonomous systems usually involve massive and heterogeneous devices for seamless communication and collaboration [1]. In the realm of robotics, the Robot Operating System (ROS) has emerged as one of the most foremost machine-to-machine communication protocol [2]. ROS 2 is evolved to build upon the data-centric middleware, i.e. Data Distribution Service (DDS), and consequently making its security highly linked with the DDS security model [3] which defines five plugins: Authentication, Access Control, Cryptographic, Logging and Data Tagging.

The first three plugins play a crucial role in authorizing actions undertaken by ROS 2 nodes and secure data exchange between communication entities. Below is a concise overview of the vital components required in these three plugins.

- Identity Certificate Authority (CA): in the Authentication plugin, it authenticates the identity of ROS 2 nodes/participants before granting them to access a specific domain.
- Permissions CA: within the Access Control plugin, it defines permissions of authenticated ROS 2 nodes regarding reading/writing specific information within the domain.
- Key generation and management: the Cryptographic plugin includes key generation, encryption, decryption, sign/verify identity certificates and all cryptographic operations associated with the aforementioned plugins.
- Security configuration file: it specifies parameters for the security plugins, such as CA's public key, signed permission files, node's certificate and private key.

The DDS security provides a domain-level protection from unauthorized outsider access and during data transmission. However, it assumes that the local machine is trusted, making it susceptible to various types of software exploits, including Denial of Service through the publication of numerous fake data [4], injected malware and masquerading credentials [5]. Moreover, the identity authentication of a ROS 2 node occurs before it provides and/or consumes data, without ongoing checks. This vulnerability exposes ROS 2 nodes to potential repurpose and limits the deployment of ROS 2 especially for the applications with stringent privacy and confidential computing requirements.

Hardware-assisted security solutions, such as Trusted Execution Environment (TEE) [6], have been proposed to enhance protection against software exploits. TEE establishes a secure and isolated area within the operating system, safeguarding data and code from unauthorized access or modification, even in compromised or privilege-escalated system scenarios. This technology holds the potential to improve ROS 2 security by providing identity attestation and ensuring data integrity and confidentiality.

SCONE [7] is a container-style TEE middleware using Intel's Software Guard eXtensions (SGX) [8], which provides a remote attestation mechanism to verify the authenticity of a user/application, thereby establishing trust among peers/clients. This paper chooses SCONE to build a secure and trusted ROS 2 communication platform due to its ease-of-use feature. The design of SCONE has the capability to enhance the implementation of components in the existing DDS security plugins.

- SCONE Configuration and Attestation Service (CAS): can undertake the role of Identity CA, providing attestation for ROS 2 nodes and conducting continuous integrity check.
- SCONE Policy: encompasses all security related details. It can function as the Permissions CA and facilitate the sharing of keys and secrets among ROS 2 nodes.
- SCONE File System Shield: encrypts data source, files and code. It could be utilized to secure the variables and files required by DDS configuration file

and performs the encryption and decryption operations specified in the Cryptographic plugin.

- SGX enclave: an encrypted memory region. With DDS installed inside the enclave, only the DDS protocol have the right to access and retrieve information within. Adversaries attempting software attacks or holding hypervisor rights cannot read or interfere with the enclave.

II. Platform Overview

This paper proposes a TEE (i.e. SCONE) based ROS 2 platform for trusted communication and task execution. The design motivation is to safeguard the DDS protocol which is the internal communication middleware for ROS 2 inside SGX enclave using SCONE. Thus, the integrity of the ROS 2 nodes can be continuously checked throughout their active time in DDS domain(s), preventing any unauthorized alterations to their registered services.

As shown in Fig.1, each ROS 2 node initiates a SCONE container to encapsulate the DDS protocol and a list of available services. An ID & Permission Manager component is devised to provide remote attestation for ROS 2 nodes and control security policies based on the agreement of all ROS 2 nodes. The Manager is operating within an enclave and can be attested by all ROS 2 nodes. Notably, the ID & Permission Manager can be deployed on the Cloud while maintaining its integrity and preserving all associated resources.

- Security Policy

A primary goal of ROS 2 security is to define permissions on nodes for producing and/or consuming specific data. The SCONE Policy design perfectly matches this requirement. For instance, a policy owner may give explicit permission to an application with the specific MRENCLAVE value by specifying the attestation method in the policy. Additionally, a secret value can be exported from one node' policy and then imported into another's for data retrieval or service execution. More details of SCONE Policy definition can be found in [9]. Leveraging the SCONE policy, ROS 2 nodes could consent on a service provision plan, including data publisher, data consumer, service provider, etc. The proposed ID & Permission Manager strictly enforces the plan. Consequently, publishers gain control over the consumption of their data and subscribers can effortlessly verify the source and integrity of the received information.

- Trust on ROS 2 nodes

In the initial setup, each ROS 2 node initiates a SGX enclave, specifically SCONE, to install DDS protocol for node discovery, deploy the Service Function Loader for managing ROS 2 applications and define encrypted parameters and files for executing ROS 2 services. With the protection of TEE, even in an untrusted host environment outside SCONE, two crucial security aspects are ensured. First, it thwarts attackers and unauthorized applications from eavesdropping on ROS 2 data. Second, it prevents root users or system administrators from maliciously altering originally registered services, such as providing fake data. Additionally, this TEE-based approach instills trust among ROS 2 nodes from the discovery phase onward. ROS 2 nodes have the flexibility to publish, subscribe, or engage in bidirectional data exchange. The SCONE Policy empowers ROS 2 nodes to govern the consumption of their data. Each ROS 2 node can create and

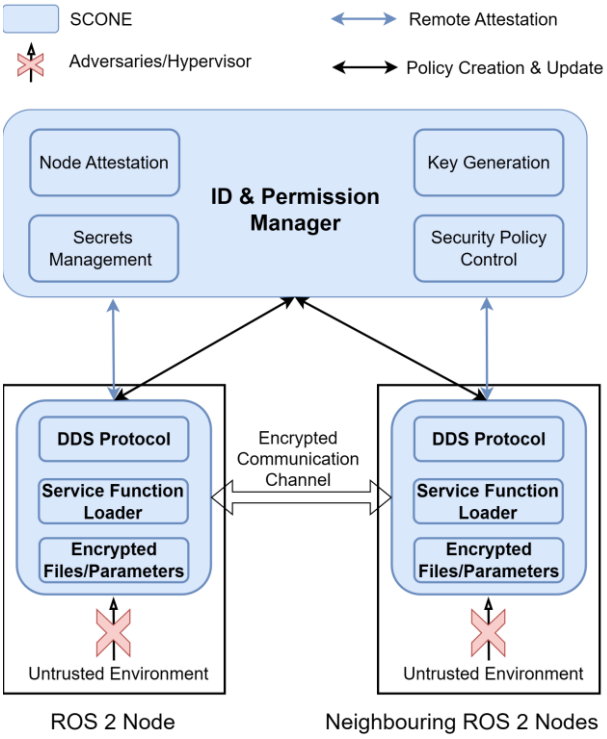

Fig.1. TEE (SCONE) based ROS 2 Platform with Remote Attestation

upload its policy to the ID & Permission Manager. Only the policy owner with the correct predecessor policy hash value is allowed to update the policy. Data subscribers can access the data after successfully passing the remote attestation process.

- Remote Attestation

The implementation of the ID & Permission Manager relies on the SCONE built-in CAS functionality. This component plays a pivotal role in attesting the compliance of ROS 2 nodes before permitting communication and collaboration, while also managing all policies uploaded by ROS 2 nodes. Noteworthy is that the Manager, as per the proposed design, cannot see the details of security policies. The ID & Permission Manager is also responsible for generating keys to encrypt/decrypt files for ROS 2 nodes and securely pass secrets between them according to their policy definitions after performing the remote attestation. Integrating TEE with ROS 2 ensures that nodes can trust the integrity of their working peers.

III. Conclusion

Beyond industrial applications, more robots are being deployed in society including autonomous vehicles [10], humanoids [11] and healthcare [12]. ROS 2 plays a pivotal role in facilitating the seamless sharing of confidential information and collaboration on intricate tasks for robotics systems. Recognizing the imperative need for high-level security and trust assurances in such diverse applications, this paper presents a novel approach by integrating TEE with ROS 2. The primary goal is to enable remote attestation for robots and devices, thereby establishing a foundation for a secure and trusted machine-to-machine communication platform. The proposed design not only safeguards data against unauthorized access but also ensures the integrity of ROS 2 nodes through attestation before granting permission for data read/write operations.

REFERENCES

[1] H. Araujo, M. R. Mousavi, and M. Varshosaz, 'Testing, Validation, and Verification of Robotic and Autonomous Systems: A Systematic Review', ACM Trans. Softw. Eng. Methodol., vol. 32, no. 2, p. 51:1-51:61, Mar. 2023, doi: 10.1145/3542945.

[2] J. Zhang, X. Yu, S. Ha, J. P. Queralta, and T. Westerlund, 'Comparison of DDS, MQTT, and Zenoh in Edge-to-Edge and Edge-to-Cloud Communication for Distributed ROS 2 Systems'. Sep. 28, 2023. doi: 10.48550/arXiv.2309.07496.

[3] J. Kim, J. M. Smereka, C. Cheung, S. Nepal, and M. Grobler, 'Security and Performance Considerations in ROS 2: A Balancing Act'. arXiv, Sep. 24, 2018. doi: 10.48550/arXiv.1809.09566.

[4] B. Breiling, B. Dieber, and P. Schartner, 'Secure communication for the robot operating system', in 2017 Annual IEEE International Systems Conference (SysCon), Apr. 2017, pp. 1–6. doi: 10.1109/SYSCON.2017.7934755.

[5] V. DiLuoffo, W. R. Michalson, and B. Sunar, 'Credential Masquerading and OpenSSL Spy: Exploring ROS 2 using DDS security'. arXiv, Apr. 27, 2019. doi: 10.48550/arXiv.1904.09179.

[6] P. Jauernig, A.-R. Sadeghi, and E. Stapf, 'Trusted Execution Environments: Properties, Applications, and Challenges', IEEE Security & Privacy, vol. 18, no. 2, pp. 56–60, Mar. 2020, doi: 10.1109/MSEC.2019.2947124.

[7] S. Arnautov et al., 'SCONE: secure Linux containers with Intel SGX', in Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation, in OSDI'16. USA: USENIX Association, Nov. 2016, pp. 689–703.

[8] V. Costan and S. Devadas, 'Intel SGX Explained'. 2016.

[9] 'Policy Language - Confidential Computing'. Accessed: Dec. 18, 2023. [Online]. Available: https://sconedocs.github.io/CAS_session_lang_0_3/

[10] A. Elmquist and D. Negrut, 'Modeling Cameras for Autonomous Vehicle and Robot Simulation: An Overview', IEEE Sensors Journal, vol. 21, no. 22, pp. 25547–25560, Nov. 2021, doi: 10.1109/JSEN.2021.3118952.

[11] J. A. Rojas-Quintero and M. C. Rodríguez-Liñán, 'A literature review of sensor heads for humanoid robots', Robotics and Autonomous Systems, vol. 143, p. 103834, Sep. 2021, doi: 10.1016/j.robot.2021.103834.

[12] H. S. Ahn, M. H. Lee, and B. A. MacDonald, 'Healthcare robot systems for a hospital environment: CareBot and ReceptionBot', in 2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), Aug. 2015, pp. 571–576. doi: 10.1109/ROMAN.2015.7333621.