Access Control Integration in Sparkplug-Based Industrial Internet of Things Systems: Requirements and Open Challenges

Pietro Colombo[a] and Elena Ferrari[b]
Department of Theoretical and Applied Science, University of Insubria, Via O. Rossi, 9 - 21100 Varese, Italy
{*pietro.colombo, elena.ferrari*}*@uninsubria.it*

Keywords: Industrial Internet of Things, Sparkplug, MQTT, Fine-Grained Access Control, Access Control Policies.

Abstract: Sparkplug (ISO, 2023) is an emerging open-source software specification that supports integrating applications and devices in Industrial Internet of Things (IIoT) systems and eases data integration. Although recently introduced, Sparkplug has rapidly gained popularity, and its specifications have already been defined as an ISO standard. However, the basic data protection features it supports could hinder the adoption of Sparkplug on a large scale. Efficient access control solutions for Sparkplug-based IIoT systems open new research challenges. In this position paper, we take the first step to fill this void by presenting key requirements for integrating access control into Sparkplug systems and discussing significant issues for developing a suitable access control framework.

1 INTRODUCTION

Industrial Internet of Things (IIoT) is an emerging paradigm that combines IoT technologies and real-time data analytics to enable device connectivity, favor faster business decisions, and maximize revenue growth (Ahmed et al., 2023; Boyes et al., 2018). Recent surveys claim that manifold industrial fields benefit from IIoT technologies and that the IIoT market is rapidly expanding. See, for instance, https://www.grandviewresearch.com/industry-analysis/industrial-internet-of-things-iiot-market).

A key issue for the IIoT domain is the interoperability of applications and heterogeneous mission-critical devices from different vendors that characterize an industrial environment (Hazra et al., 2021), which can be addressed by relying on a standard communication protocol.

In recent years, MQTT (OASIS, 2019) has become a reference communication protocol for IoT applications. MQTT is often used in IIoT frameworks as the adopted topic-based publish/subscribe communication overcomes the limits of poll/response protocols used in Industrial Control systems (Nast et al., 2022). However, MQTT lacks a standardized data format, leading to interoperability problems. Moreover, MQTT lacks a unified, systematic approach to managing the state of devices, which IIoT control applications must continuously monitor.

Sparkplug (ISO, 2023) is a recent open specification designed to favor the interoperability of heterogeneous devices and data integration. Sparkplug extends MQTT while maintaining compatibility with its infrastructure. Indeed, a Sparkplug system is composed of MQTT servers and specialized MQTT clients. Sparkplug standardizes state management and message topic structures and rules out message payload formatting and encoding, providing vendor-neutral specifications for data formats. Although recently proposed, Sparkplug appears as an up-and-coming IIoT technology and has already been standardized (ISO, 2023). However, the Sparkplug's diffusion might be hindered by the poor security features it provides. Critical security issues can lead to equipment damage, regulatory problems, and personal safety hazards. Like MQTT, Sparkplug supports basic access control mechanisms. Sparkplug specification suggests using access control lists (ACLs), relying on the MQTT server's ability to enforce them. However, ACLs ensure a coarse-grained level of protection based on message topic matching. Policies specify the read/write topics authorized to a client, and access is granted for any message with a compliant topic. More effective mechanisms are needed. Access control should be enforced at a finer-grained granularity level, granting permissions based on message properties, and the access request context.

[a] https://orcid.org/0000-0003-4617-5247
[b] https://orcid.org/0000-0002-7312-6769

Colombo, P. and Ferrari, E.
Access Control Integration in Sparkplug-Based Industrial Internet of Things Systems: Requirements and Open Challenges.
DOI: 10.5220/0013055700003825
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 20th International Conference on Web Information Systems and Technologies (WEBIST 2024)*, pages 380-384
ISBN: 978-989-758-718-4; ISSN: 2184-3252
Proceedings Copyright © 2024 by SCITEPRESS – Science and Technology Publications, Lda.

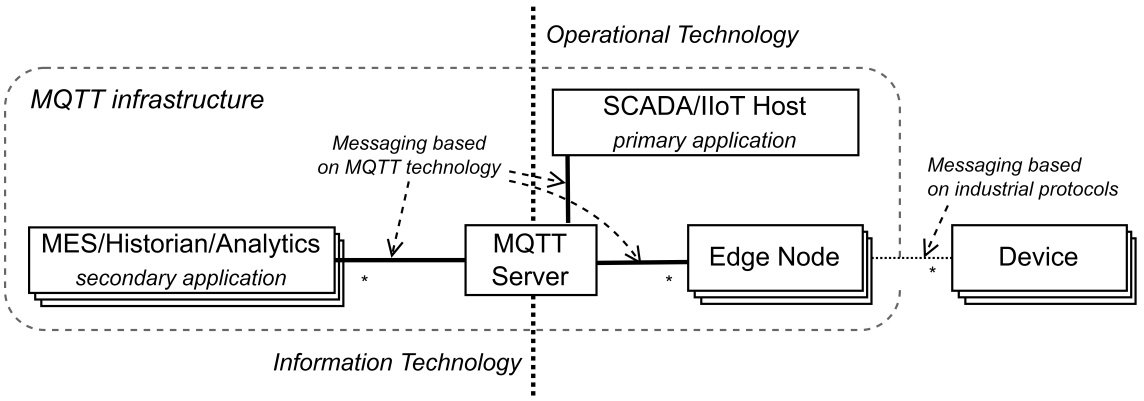

Figure 1: Sparkplug system architecture.

Access control is a crucial security feature for IIoT systems and has been thoroughly investigated (e.g., see Salonikias et al., 2019). Access control frameworks for the IoT can be classified based on the characteristics of the adopted access control model and the enforcement mechanisms they use. In a large class of frameworks, which include, for instance, the work by Xu et al. (2018), access authorizations are materialized into capability tokens assigned to subjects, who must prove their authorizations by presenting the received token. Another framework family, which comprises the work by La Marra et al. (2018), grants authorizations when access control conditions that refer to mutable subjects, objects, and environment attributes are satisfied. Authorizations are adapted to attribute changes. On the other hand, access control frameworks like that proposed by Gouglidis and Mavridis (2012), grant privileges to users via role assignments. Finally, frameworks such as the one proposed by Han et al. (2022), authorize access when conditions that refer to subjects, objects, and environment attributes are satisfied.

Despite the vast literature on access control, only a few works have focused on integrating access control in MQTT systems (e.g., La Marra et al., 2018; Colombo et al., 2021), and we are unaware of works on Sparkplug. Although Sparkplug systems rely on MQTT as base messaging technology, access control approaches for MQTT are unsuited for being used with Sparkplug, and cannot be easily adapted to Sparkplug systems. A new, dedicated approach is needed, to be designed by taking into account distinctive Sparkplug features, such as the Report by Exception communication and the standardized message payload definition (ISO, 2023). To fill this void, in this paper, we present essential requirements for integrating access control in Sparkplug-enabled IIoT systems and discuss open research problems for developing an access control framework.

The remainder of the paper is organized as follows. Section 2 overviews basic Sparkplug concepts. Section 3 introduces essential requirements for defining an access control framework for Sparkplug systems. Section 4 discusses research challenges related to the framework development, and finally, Section 5 concludes the paper.

2 SPARKPLUG

A Sparkplug system includes an MQTT server that routes messages issued by specialized MQTT clients operating as edge nodes with related devices, primary and secondary applications. An example of Sparkplug architecture is shown in Figure1. Edge nodes are gateways for devices (such as sensors, PLCs, and RTUs) that cannot communicate via Sparkplug. Edge nodes allow them to communicate with the rest of the infrastructure and control their states. Primary applications, are SCADA/IIoT hosts that monitor the state of edge nodes and devices and issue commands in response to a state change. Finally, secondary applications, like Manufacturing Execution Systems (MES), Historians, and Analytics, receive and process data generated by edge nodes and devices and possibly send commands.

Communication among applications and edge nodes is achieved by exchanging Sparkplug messages, namely MQTT application messages, with specific payload and topic structures. The payload of a Sparkplug message aggregates properties referred to as metrics that typically denote diagnostics, current state values, and metadata. The topic of a message specifies the components to which that message should be addressed and the message type.

Supported message types are listed in Table 1.

Table 1: Sparkplug message types.

Message type	*Semantics*
NBIRTH/DBIRTH	Notifies that an edge node/device is online. The payload metrics denote information related to the component interface and the data it handles.
NDEATH/DDEATH	Notifies that an edge node/device went offline.
NDATA/DDATA	Notifies that one/multiple metrics handled by an edge node/device changed value.
STATE	Notifies a state change from online to offline or vice versa of the primary application.
NCMD/DCMD	Notifies the request by primary/secondary applications to modify the value of metrics controlled by an edge node/device.

3 REQUIREMENTS

An access control framework for Sparkplug-enabled IIoT applications should ensure fine-grained access control with customizable protection levels and a low enforcement overhead.

State-of-the-art fine-grained access control approaches for MQTT systems operate at the message granularity level (e.g., Colombo et al., 2021). However, in Sparkplug systems, the granularity level should be finer-grained for some message types. NBIRTH/DBIRTH and NDATA/DDATA messages aggregate metrics, each denoting a distinct data property. By enforcing access control at the message level, a Sparkplug message addressed to a rightful subscriber is blocked or routed with all its metrics. However, some of the metrics could be necessary for a subject and not needed for others, and based on the Principle of Least Privilege (Saltzer and Schroeder, 1975), subjects should receive the minimum privileges to achieve their tasks. The same considerations apply to NCMD/DCMD messages that allow multiple write requests to be issued simultaneously. For instance, a SCADA/IIoT host should access all the metrics edge nodes and devices manage and issue control signals to manage these components. In contrast, a secondary application performing analytics should not access the metrics allowing device control. To enforce access control at the metrics granularity level, authorized views of Sparkplug messages should substitute the original messages, from which the unauthorized metrics will be removed/obfuscated. For instance, any metric that allows modifying a receiving device's state should be pruned out from the view of a DCMD message issued by a secondary application.

Additionally, to favor customization, access control should consider any available information related to the requesting subject, the protected object, and the request context. In Sparkplug systems, the subjects are referred to by identifiers. In contrast, metrics specify the timestamp at which they have been set and can also be labeled with additional metadata. A discretionary access control model should, therefore, be able to regulate subjects' accesses based on object and environment attributes. More precisely, to exploit the Sparkplug mentioned above features, access should be granted to subjects based on the satisfaction of conditions that refer to the access request time and metric properties. For instance, assume that the metric property sensitivity is used to denote the sensitivity level of the metrics. A secondary application could be authorized to access data generated by a device at a specific time interval, but only if none of the metrics included in the message payloads has been marked as sensitive.

The enforcement monitor should be implemented in such a way as to intercept any Sparkplug message addressed to the MQTT server and issued by the MQTT server to edge nodes and applications. Depending on the applicable access control policies, the monitor could either block the message transit or remove unauthorized metrics from the message payload (if any) and consent to the transit of the derived view.

4 OPEN CHALLENGES

In this section, we discuss open research challenges implied by the requirements presented in Section 3.

The peculiarities of Sparkplug make the specification of an access control enforcement mechanism at the metric granularity level for the access control model mentioned in Section 3 a challenging task. In particular, Sparkplug recommends adopting a report-by-exception (RbE) communication strategy to save transmission bandwidth. According to RbE, data and state information are only published when they change. For instance, suppose *sa* is a secondary application that subscribed to the reception of messages by an edge node *en*. Suppose *sa* is allowed to see metric *m1* within messages published by *en* during the time

intervals *[t1..t2]* and *[t4..t5]*. Based on the RbE approach, whenever *m1* changes value, *en* publishes an NDATA message that includes *m1*. During the interval *[t1..t2]*, authorized views of these NDATA messages, including *m1*, must be sent to *sa*. Suppose now that *en* updates *m1* at time *t3* immediately after *t2* but before *t4*. Based on the above-mentioned policy, *sa* cannot see the update at *t3*. However, the access is newly allowed at *t4*. Therefore, at *t4*, *sa* should be made aware of the value of *m1* published by *en* at *t3*, as it is the most recent one. An open issue is, therefore, to find an efficient approach to communicate the last *m1* value. One possible solution could be forcing *en* to generate an NBIRTH message that includes *m1*, with a forged NCMD message requesting the *en* rebirth. However, the exemplified scenario refers to a single case of interaction, but additional cases could be identified. A thorough analysis is therefore needed, as well as the study of the related enforcement mechanisms.

Another challenge is related to policy specification. The enforcement of access control at the metric level could cause significant growth of the policy set size. However, it could be reasonable to assume that within a message payload, the sensitive metrics that require dedicated access control policies could be a minority. Approaches to minimizing the policy number and easing policy administration must be investigated. A viable strategy could be using a policy model that protects classes of messages satisfying a topic filter with an exception list that specifies the metrics to be removed. For instance, referring to the previous example, one could specify a policy *p1* that matches any NBIRTH and NDATA message published by *en*, with an exception list that includes *m1* and a condition specifying the time interval *(t2..t4)* and a policy *p2* with the same topic filter, an empty exception list, and a condition always true.

Many policies could be specified for primary and secondary applications that could subscribe to receiving data from any edge node and device in a Sparkplug system. A key requirement is that the access control policies that apply to an access request are identified with minimal delay. For instance, policy retrieval could benefit from efficient in-memory NoSQL databases like Redis (https://redis.io), which ensures the execution of lookup-by-key operations with submillisecond latency. However, data modeling strategies for access control policies and indexing strategies for such databases should be studied to minimize the selection time of access control policies applicable to an access request.

The efficiency of policy evaluation could depend on the language employed to specify AC conditions. It could be possible to rely on platforms like GraalVM (https://www.graalvm.org), which allow code execution in multiple programming languages with high performance. However, an empirical evaluation is needed to assess the performance and to see whether such a solution is oversized.

Finally, the enforcement monitor could either be developed as an independent component operating as a proxy of the MQTT server or integrated into the MQTT server itself, exploiting the leading brokers' extensions mechanism, like those of HiveMQ (https://www.hivemq.com), EMQX (https://www.emqx.com) or Mosquitto (https://mosquitto.org). Both approaches should be studied by assessing aspects of performance and scalability.

5 CONCLUSIONS

Despite the vast literature on access control for the IoT domain, up to now no access control framework has targeted Sparkplug systems.

In this paper, we take a step to fill this void, presenting key requirements for integrating fine-grained access control in Sparkplug-based IIoT systems and discussing research issues to be addressed to build an effective access control framework for this use case.

ACKNOWLEDGEMENTS

This work was supported in part by project SERICS (PE00000014) under the NRRP MUR program funded by the EU-NGEU.

REFERENCES

Ahmed, S. F., Alam, M. S. B., Hoque, M., Lameesa, A., Afrin, S., Farah, T., Kabir, M., Shafiullah, G. M., and Muyeen, S. M. (2023). Industrial Internet of Things enabled technologies, challenges, and future directions. *Computers and Electrical Engineering*, 110.

Boyes, H., Hallaq, B., Cunningham, J., and Watson, T. (2018). The industrial internet of things (IIoT): An analysis framework. *Computers in Industry*, 101.

Colombo, P., Ferrari, E., and Tümer, E. D. (2021). Regulating data sharing across MQTT environments. *Journal of Network and Computer Applications*, 174.

Gouglidis, A. and Mavridis, I. (2012). DomRBAC: An access control model for modern collaborative systems. *Computers and Security*, 31(4).

Han, D., Zhu, Y., Li, D., Liang, W., Souri, A., and Li, K. C. (2022). A Blockchain-Based Auditable Access Con-

trol System for Private Data in Service-Centric IoT Environments. *IEEE Transactions on Industrial Informatics*, 18(5).

Hazra, A., Adhikari, M., Amgoth, T., and Srirama, S. N. (2021). A Comprehensive Survey on Interoperability for IIoT: Taxonomy, Standards, and Future Directions.

ISO (2023). Sparkplug® version 3. ISO/IEC 20237:2023 - Information technology.

La Marra, A., Martinelli, F., Mori, P., Rizos, A., and Saracino, A. (2018). Introducing usage control in MQTT. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 10683 LNCS.

Nast, M., Raddatz, H., Rother, B., Golatowski, F., and Timmermann, D. (2022). A Survey and Comparison of Publish/Subscribe Protocols for the Industrial Internet of Things (IIoT). In *ACM International Conference Proceeding Series*.

OASIS (2019). MQTT Version 5.0 OASIS Standard.

Salonikias, S., Gouglidis, A., Mavridis, I., and Gritzalis, D. (2019). Access control in the industrial internet of things. In Alcaraz, C., editor, *Security and Privacy Trends in the Industrial Internet of Things. Advanced Sciences and Technologies for Security Applications.* Springer, Cham.

Saltzer, J. H. and Schroeder, M. D. (1975). The Protection of Information in Computer Systems. *Proceedings of the IEEE*, 63(9).

Xu, R., Chen, Y., Blasch, E., and Chen, G. (2018). BlendCAC: A smart contract enabled decentralized capability-based access control mechanism for the IoT. *Computers*, 7(3).

SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS