An Enrolment Gateway for Data Security in Heterogeneous Industrial Internet of Things

Fergal Martin-Tricot, Cédric Eichler, Pascal Berthomé

▶ **To cite this version:**

Fergal Martin-Tricot, Cédric Eichler, Pascal Berthomé. An Enrolment Gateway for Data Security in Heterogeneous Industrial Internet of Things. 29th IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises - WETICE 2020, 2020, Bayonne (Virtual Conference), France. hal-02960931

HAL Id: hal-02960931
https://hal.science/hal-02960931

Submitted on 8 Oct 2020

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.

An Enrolment Gateway for Data Security in Heterogeneous Industrial Internet of Things

Fergal MARTIN-TRICOT, Cédric EICHLER, Pascal BERTHOMÉ
LIFO - INSA Centre-Val de Loire
88 Boulevard Lahitolle
18000 BOURGES, France
E-Mail: firstname.lastname@insa-cvl.fr

***Abstract*—Industry includes more and more *IoT* components to have a better control on production and logistic processes.**

Unfortunately, growing network openness in a formerly isolated world induces major security risks which are especially critical in an industrial context. These risks are exacerbated by the highly fragmented nature of the industrial *IoT* market which imposes interoperability management and challenges security.

We propose an approach to enable end-to-end data security within a heterogeneous *IoT* deployment. Interoperability is ensured by a central network powered by oneM2M interacting with various tier protocols. In this paper, we focus on secure communication with ZigBee and discuss how it can be transposed to other protocols, namely Z-Wave and Thread.

I. Introduction

During the last years, the industry adopted *Machine to Machine* (***M2M***), a paradigm of communication that allows direct machine to machine exchanges [1]. This approach provides tools to have a better real-time vision and therefore control on procedures. The next step of this transformation is now on track: Industry 4.0 [2]. The idea is to use the *Internet of Things* (***IoT***) to interconnect every component of the industrial process and thus have a real-time vision of it.

The industrial context implies some paramount security requirements induced by the criticality of processes in terms of safety and industrial secrecy. For this reason, protocols used for industrial *IoT* must be blameless on the security mechanism employed. Unfortunately, *IoT* protocols, mainly because of their devices' limited resources, have made some trade-off on the security.

Furthermore, there is a lot of protocols that address different needs. The most interesting ones for industrial concerns can be divided in two main categories: *Wireless Local Area Network* (***WLAN***) (with mesh wireless network [3] and classical wireless ones) and *Low Power Wide Area Network* (***LPWAN***) [4].

The first one is suitable to make a communication network at room or building scale. This is the perfect choice to have, for example, a sensor network to monitor a facility production. This category comprises classical technologies such as WiFi or Bluetooth but also Mesh networking that offers some interesting properties like better range and energy efficiency [5]. The most used and developed mesh *IoT* technologies are ZigBee [6], Thread [7] and Z-Wave [8].

On the other hand, *LPWAN* usually rely on a network widely deployed and proposed by a service provider. The main difference with 3G or 4G networks is that the protocol is more optimized to reduce power consumption. These protocols can thus work in a big area and allow a device to be connected anywhere [9]. In an industrial approach, it can be used in shipment and supply chain.

The multiplicity of manufacturers, protocols, and their complementarity pushes us to consider that a realistic use case is necessarily heterogeneous in terms of protocols. It raises a known problem in computer science: protocol interoperability. When two different technologies are involved, data structures are different and complicates communication. In this context, the security management is still an issue [10] as different approaches must coexist and raises the question of the interface security.

A. Scenario

Our scenario takes place in a factory which needs to interconnect *IoT* devices. A central network, dedicated to the control and supervision of all devices, communicates with different protocols.

Since interoperability is centred around this central network, it is critical to ensure the data security between an end device which is a part of a tier protocol and a device inside the central network.

According to our scenario, a major constraint we are putting is the **impossibility to alter the operation of an end-device in a tier protocol**. Indeed, we consider that it could be a closed device based on proprietary technologies.

B. Key challenge and article overview

The first constraint is to find a solution that could operate the central network and thus be able to communicate with every protocol. An existing specification, oneM2M, seems to fit these requirements and will be presented in the next section.

Our proposition is an approach to enable communication between a oneM2M node with an other one using a tier protocol with end-to-end ciphering of the data exchanged. As an illustration, our approach will use ZigBee as tier protocol.

II. Related Work

As previously stated, the security of *IoT* protocols has been well studied. This is particularly true for the protocols we are interested in: *Mesh Local Area Network*.

ZigBee is probably the most mature and represented of such protocols. Modern security was introduced in version 3.0 of ZigBee and has since been largely analysed, for example by Zillner [11] and Fan et al. [12]. More recent protocols inspired by ZigBee such as Thread and Z-Wave have also been the target of in-depth security analysis. One may cite for example Dinu and Kizhvatov [13] on Thread and Rouch et al. [14] regarding Z-Wave. The consensus on these protocols is that security is acceptable in most cases [15] because most security breaches are coming from implementation issues.

In spite of the variety of proposals addressing *IoT* interoperability, solutions are largely converging. They usually imply a Gateway (e.g. [16], [17]) which has to translate data and instructions between different protocols. Regardless of the implementation and varying details, to the best of our knowledge, all of these kinds of approaches rely on a central translator interfacing the different protocols. They thus impose the same trust model: the interoperability-enabling component, generally unique and central, is necessarily trusted.

The security of a trusted interoperability gateway is discussed, for example by Tarouco et al. in [18] or by Martino et al. in [19]. The data security is an inescapable issue within the industrial *IoT*. It is thus necessary to limit the number of trusted components and their trust level. In 2017, Mukherjee et al. proposed a solution to implement end-to-end communication for Cloud-Fog communication [20] through a pre-shared key approach. To the best of our knowledge, no work guarantee end-to-end data security in heterogeneous *IoT* communication from the enrolment to the communication phase.

III. An interoperability solution: *oneM2M*

oneM2M is a standard initially proposed by the *ETSI* (European Telecommunication Standard Institute) and nowadays supported by a dozen of national and international standardization institutes and professional forums. It aims to provide a common framework to operate in *Machine to Machine* (*M2M*) and *IoT* paradigm [21]. In particular, oneM2M proposes a mature approach for interoperability management and numerous internal security measures [22].

A. Interoperability management in oneM2M

The oneM2M's interoperability management relies entirely on a specific entity: the *Interworking Proxy Entity* (***IPE***). The idea is to create a node supporting a non-oneM2M communication interface and a oneM2M's one.

The *IPE* is thus connected to a Third-Party protocol and it has the following main functions:

- It gets data from the network it is connected to.
- It translates data between the tier network and oneM2M's
- It ensures the integrity of the data it transmits into the oneM2M's network - using oneM2M's security measures.

Every specific protocol can be made compatible with oneM2M by creating a dedicated *IPE*.

Because this approach is generic, it can work with most of the existing protocols. However, data security can become a problem because every data must transit through the *IPE*. A vulnerability into the *IPE* would give read and write access to every data in the third-party network.

B. Introduction to oneM2M's internal data security

To securely communicate within the oneM2M network, a oneM2M device can follow a procedure defined in the *Remote Security Provisioning Framework* (***RSPF***). Its aim is to provide a node with the keys it needs to authenticate and thus communicate securely within the network. A specific entity is defined into the specification to ensure the key provisioning during enrolment: the *M2M Enrolment Function* (***MEF***).

In particular, two oneM2M nodes may communicate securely with end-to-end encryption if they have exchanged an *ESData* key. It can be provisioned during *RSPF* inside the two nodes which need to communicate.

With this approach, data transmitted can be totally secured between two oneM2M nodes. Our aim is thus to provide end-to-end security from the tier network up to the management network and negating the flaws identified in the previous subsection. To do so, we must assume that **we can modify the *IPE* and the oneM2M's end device which wants to communicate with the tier network**.

IV. A third-Party protocol as an example: ZigBee

To design and to assess the feasibility of our approach, we rely on a real protocol to find a working technical solution.

A. Introduction to ZigBee

ZigBee is a specification used to create *Wireless Personal Area Network* (***WPAN***) using radio communication. Introduced for the first time in 1998 and standardized in 2003, the current version, *ZigBee 3.0* [23], proposes an open and secure *WPAN* technology using a mesh network architecture.

We are basing our work on this technology because of its openness specification and the ease to find test devices - like XBee. Moreover, the others *WPAN* that we studied (Z-Wave, Thread) are working in a similar way, in regard to enrolment, as ZigBee and our work could be adapted on an other protocol.

B. ZigBee with a oneM2M classic deployment

In this part, we will present an architecture using ZigBee and oneM2M, corresponding to a "vanilla" oneM2M's deployment (depicted in Fig. 1), it raises issue with three main security breaches:

- The *IPE* is the end of the oneM2M's security measures
- The Gateway is the end of the ZigBee's security measures
- The communication between both can also be a problem

For the *IPE* to be able to act on enrolment, we need to make the assumption that **we can modify the ZigBee gateway**.

C. ZigBee security model

Two different security models for the enrolment exist in ZigBee 3.0: centralized and distributed. In this section, we will present the security used in a ZigBee network based on the centralized one, as it corresponds to our use-case.

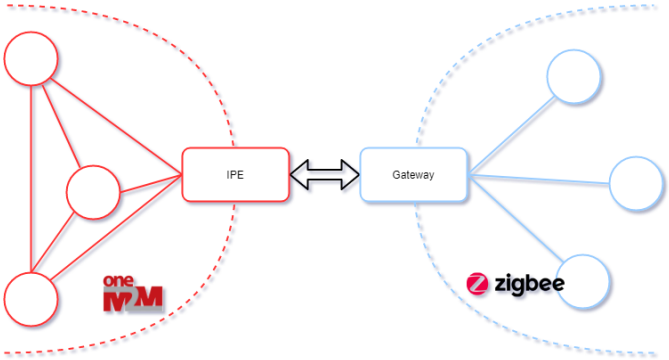

Fig. 1. ZigBee-oneM2M deployment

1) Keys used: The protocol defines two main symmetric ciphering keys that are used though all the lifespan of a node.

The first one is the install_code, a symmetric key included inside the device during manufacturing. It cannot be changed, should be unique and is used for the authentication during the enrolment of the node.

The second one is the Network Key. It is shared by every node of a specific network and used to cipher every communication inside it.

2) Enrolment: The enrolment, like in every wireless protocol, is a critical phase during which the key used to cipher all communications must be transmitted to a new device.

The key point is the usage of a specific ciphering symmetric key to authenticate the new device to the *Trust Centre*: the install_code. Once this key is shared with the *Trust Centre* (QR code scanning for example), a secure communication is done to provision the *Network Key* to the new device.

After the enrolment phase, a node is able to communicate securely using the *Network Key* with every device of the network.

V. Ensuring End-to-End Security in Heterogeneous Network

A. Assumptions and Objective

Our main objective is to propose some minimal network architecture alterations that should enable end-to-end data security between a oneM2M node and a ZigBee one.

B. The working principle

The *IPE* and the ZigBee gateway have total access on data that transit through them. To protect the data, we must cipher it from the sender to the receiver. To do so, we propose a way to divert the enrolment principle of ZigBee to deploy a ciphering key into a ZigBee node, from the oneM2M network.

Our proposal relies on the creation of a oneM2M's special *MEF* which would receive and deal with ZigBee's enrolment procedure. It becomes possible to deliver a specific ciphering key for a ZigBee node.

After the provisioning of this key, the ZigBee node will continue to work normally but the message sent will be ciphered using a key only known by the recipient within the oneM2M's network.

C. Enrolment in detail

Here are the involved actors and their names, as shown in Fig. 2:

- A ZigBee node, denoted **A**, which is not yet enrolled in the ZigBee network and needs to join it. Its install_code is denoted **IC-A**
- The ZigBee gateway, which has the *Trust Centre* role and is connected to the *IPE* through an out-of-scope medium (HTTP for example). This part has to be modified.
- The *IPE*, which will be modified to basically only forward ciphered messages.
- A oneM2M's node, denoted **B**, which wants to communicate with **A** to get some data. We assume that it can be modified, at least at an application level.
- Our custom *MEF* entity, named **MEF Gateway**, is made to manage ZigBee enrolment from the oneM2M network. It should be made to work with a specific protocol.

As presented in Section IV-C2, node enrolment in ZigBee is authenticated and ciphered thanks to a classical *Pre-Shared Key* approach. This choice should offer an appropriate protection against man-in-the-middle attacks.

Thanks to this security mechanism, we can consider moving the *Trust Centre* functionalities in our *MEF Gateway* without the need to trust intermediate nodes.

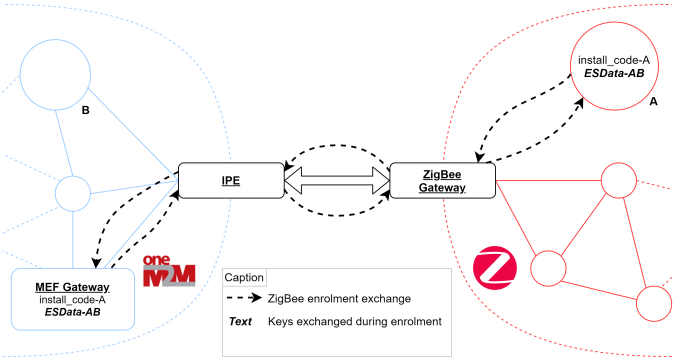

Fig. 2. Enrolment of a ZigBee node with our *MEF* gateway

As in a classical ZigBee network, to enrol a node, our *MEF Gateway* must be provided with the *install_code* of **A**.

After this phase, as depicted in Fig. 2 all the standard ZigBee enrolment exchange will be transmitted to our *MEF Gateway*. By doing this, we can have a total control over the enrolment procedure and thus the provisioned keys.

This phase will provide **A** with the *Network Key*. With this approach, the network key is used for end-to-end communication and will be known only by **A** and **B** (and our *MEF*, as with a classic oneM2M's *MEF*). Neither the *IPE* nor the Gateway is able to read or modify the messages sent by **A**.

The same *ESData-AB* key is thus provisioned thanks to standard oneM2M's *RSPF* procedure in node **B**.

D. Communication between A and B

By overriding the original *IPE*, the translation of communications between oneM2M's and ZigBee's data ontologies must be done elsewhere. We are therefore splitting the role of the

IPE into several devices. The network interface stays within the "original" *IPE* and is named *IPE-Network Interface* (**IPE-NI**) whereas the ontology and ciphering ones are now within the oneM2M's end device and are named *IPE-Ontology and Ciphering Interface (**IPE-OCI**)*.

To allow transparent and secured communication between **A** and **B**, we rely on the modular approach of oneM2M. Indeed, in oneM2M, an *Application Service Node* is divided into the application part and the *Common Service Entity* which manages communication with the oneM2M network. We therefore propose to add the *IPE-OCI* module, into the *CSE* of a node which needs to communicate with ZigBee.

With our solution, a data exchange between a oneM2M and a ZigBee node is as follows (as depicted in figure 3):

1) The application, deployed on Node B, needs to get a data from Node A. It sends the corresponding request to its *CSE*.
2) The *IPE-OCI*, inside the *CSE*, translates this request with the ZigBee data structure and cipher it using *ESData-AB* key.
3) This message is sent to the *IPE-NI*, the gateway, and finally, **A**. When the request is received by **A**, the data will be sent through the *Gateway* and the *IPE-NI* to be received by **B**
4) The message is deciphered and translated by the *IPE-OCI* before being sent back to the application.

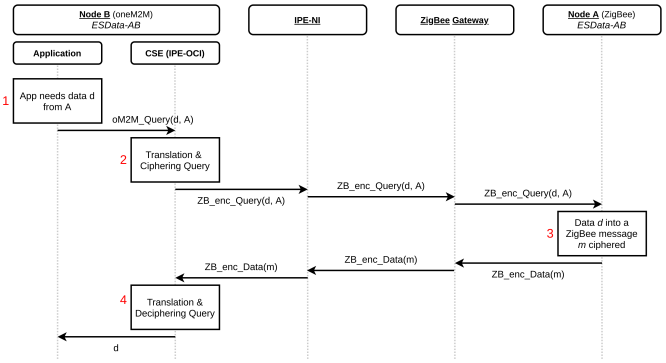

Fig. 3. Data exchange with our solution

In this way, communication is well ciphered from end-to-end and only devices A and B have access to the data.

VI. Conclusion

Integration of *IoT* within the industry, as the foundation of the 4th industrial revolution, is crucial. Yet, it poses new security threats, especially when considering a heterogeneous, interoperable environment.

In this paper, we address the issue of end-to-end data security within a heterogeneous industrial *IoT* architecture. Our proposal is based on a promising standard handling interoperability, oneM2M. Using ZigBee as a use-case, we interface both internal security mechanisms to handle ZigBee node enrolment within oneM2M and thus centralize key management. In addition to its wide use, the security model of ZigBee is close to others, such as Z-Wave's and Thread's, making our solution adaptable to these kinds of protocols.

The next step is now to implement our proposal into a hardware solution or a simulated environment.

References

[1] J. Latvakoski, A. Iivari, P. Vitic, B. Jubeh, M. B. Alaya, T. Monteil, Y. Lopez, G. Talavera, J. Gonzalez, N. Granqvist, M. Kellil, H. Ganem, and T. Väisänen, "A survey on M2M service networks," *Computers*, vol. 3, no. 4, pp. 130–173, 2014.

[2] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Business & Information Systems Engineering*, vol. 6, no. 4, pp. 239–242, Aug 2014.

[3] Y. Liu, K.-F. Tong, X. Qiu, Y. Liu, and X. Ding, "Wireless mesh networks in IoT networks," in *2017 International Workshop on Electromagnetics: Applications and Student Innovation Competition*. IEEE, 2017, pp. 183–185.

[4] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of LPWAN technologies for large-scale IoT deployment," *ICT Express*, vol. 5, no. 1, pp. 1–7, mar 2019.

[5] C. Gomez and J. Paradells, "Wireless home automation networks: A survey of architectures and technologies," *IEEE Communications Magazine*, vol. 48, no. 6, 2010.

[6] ZigBee, "ZigBee: Securing the IoT," ZigBee Alliance, Tech. Rep., 2017.

[7] Thread, "Thread technical overview," Thread, Tech. Rep., 2015.

[8] ABR, "Introduction to the Z-Wave security ecosystem," Sigma Design, Tech. Rep., 2016.

[9] K. E. Nolan, W. Guibene, and M. Y. Kelly, "An evaluation of low power wide area network technologies for the internet of things," in *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, sep 2016.

[10] M. Elkhodr, S. Shahrestani, and H. Cheung, "The internet of things: New interoperability, management and security challenges," *International Journal of Network Security & Its Applications*, vol. 8, no. 2, pp. 85–102, mar 2016.

[11] T. Zillner and S. Strobl, "ZigBee exploited - the good, the bad and the ugly," *Black Hat*, 2015.

[12] X. Fan, F. Susan, W. Long, and S. Li, "Security analysis of Zigbee," *Massachusetts Institute of Technology*, 2017.

[13] D. Dinu and I. Kizhvatov, "EM analysis in the IoT context: Lessons learned from an attack on Thread," *IACR*, pp. 73–97, 2018.

[14] L. Rouch, J. François, F. Beck, and A. Lahmadi, "A Universal Controller to Take Over a Z-Wave Network," in *Black Hat Europe 2017*, pp. 1–9.

[15] D. Celebucki, M. A. Lin, and S. Graham, "A security evaluation of popular internet of things protocols for manufacturers," in *2018 IEEE International Conference on Consumer Electronics*, 2018, pp. 1–6.

[16] H. Derhamy, J. Eliasson, and J. Delsing, "IoT interoperability—on-demand and low latency transparent multiprotocol translator," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1754–1763, oct 2017.

[17] M. Blackstock and R. Lea, "IoT interoperability: A hub-based approach," in *2014 International Conference on the Internet of Things (IOT)*, oct 2014.

[18] L. M. R. Tarouco, L. M. Bertholdo, L. Z. Granville, L. M. R. Arbiza, F. Carbone, M. Marotta, and J. J. C. de Santanna, "Internet of things in healthcare: Interoperability and security issues," in *2012 IEEE International Conference on Communications (ICC)*, 2012.

[19] B. D. Martino, M. Rak, M. Ficco, A. Esposito, S. Maisto, and S. Nacchia, "Internet of things reference architectures, security and interoperability: A survey," *Internet of Things*, vol. 1-2, pp. 99–112, sep 2018.

[20] B. Mukherjee, R. L. Neupane, and P. Calyam, "End-to-end IoT security middleware for cloud-fog communication," in *2017 IEEE 4th International Conference on Cyber Security and Cloud Computing*, 2017.

[21] *Functional Architecture - TS-0001 - v3.12.0*, oneM2M, 2018.

[22] *Security Solutions - TS-0003 - v3.8.0*, oneM2M, 2018.

[23] *ZigBee 3.0 Stack User Guide*, NXP Semiconductors, Sep. 2018.