

# Data Exchange Protocol for Cryptographic Key Distribution System Using MQTT Service

Janusz Furtak

Military University of Technology  
 ul. Kaliskiego 2, 00-904 Warsaw,  
 Poland  
 Email: janusz.furtak@wat.edu.pl

□ **Abstract**— There is an increasing demand for capturing reliable data from IoT network devices. Due to the limited capabilities of such devices to process and store sensitive data and the range and performance of the communication link, it is a significant challenge to develop a secure solution for symmetric key distribution. This paper presents a secure data exchange protocol for a cryptographic key generation, renewal, and distribution (KGR) system. The Trusted Platform Module (TPM) supports the trust establishment, key generation, all cryptographic procedures of the KGR system and secure data exchange in the described protocol. The protocol uses the MQTT (Message Queuing Telemetry Transport) service, which IoT devices widely use.

## I. INTRODUCTION

IN many systems where symmetric cryptography has been decided upon, the challenge of securely generating, renewing and distributing symmetric cryptographic keys must be met. For traditional IT systems, solutions of this type are familiar. A more significant challenge is to develop a solution for scenarios where IoT sensor nodes are the data source. The reason is that sensor nodes are usually built on devices with limited capabilities. These limitations include memory size, processing power, limited power source, and the short range of the wireless link used. Using a certification authority (CA) to build trust structures in such systems is difficult or impossible. Hence, in the cryptographic key distribution system and in systems that receive those keys, trust structures must be created locally with the support of electronic circuits such as TPM or Zymkey<sup>1</sup>.

Large networks of sensor nodes are usually divided into groups of cooperating devices to find a compromise between the limited capabilities of IoT devices and security requirements (e.g., confidentiality, integrity, availability, etc.). This group of devices uses group key management (GKM) for key distribution. Dammak et al. in [1] presented an extensive analysis of the properties of GKM systems

depending on their applications. The analysis included: wireless body area networks (WBANs) [2], wireless sensor networks (WSNs) [3], cloud computing [4], wireless IPv6 networks [5], and IoT [6][7]. The result of this analysis is the following conclusions. Performance degrades quickly when many devices form a single group and when group members change frequently. The problem of key renewal when devices join/leave a group is not fully resolved. The Decentralized Lightweight Group Key Management architecture for Access Control, presented in [1], attempts to solve the problems mentioned in the above analysis.

The article presents a secure data exchange protocol, the most critical component of the cryptographic key generation, renewal, and distribution (KGR) system. The developed security mechanisms use a hardware-based Trusted Platform Module (TPM) to establish local trust structures and ensure secure storage and data exchange on IoT network node resources. The developed protocol guarantees security regardless of the protection applied to the MQTT service used, which is an essential advantage for applications in federated environments.

## II. CONCEPT OF A DATA EXCHANGE PROTOCOL

### A. Characteristics of the cryptographic key generation and renewal system

The system of generation, renewal, and distribution of cryptographic keys (KGR) for a hybrid environment, in which a protected exchange of data between domains of sensor nodes and traditional systems (IT systems) will be required, is presented in figure Fig. 1.

Clients of the KGR system can be sensor node networks and traditional IT systems. Sensor node networks are organized as secure sensor node domains [9][10]. Only representatives (Gateway nodes) of secure sensor node domains and representatives of other systems participate in distributing cryptographic keys in the KGR system. Each gateway node works as a sink node for data originating from domain nodes or the IT system and as an emitter of data

□ This work was supported by the UGB-798 university project.

<sup>1</sup> <https://www.zymbit.com/zymkey>

originating outside for domain nodes or nodes of the IT system.

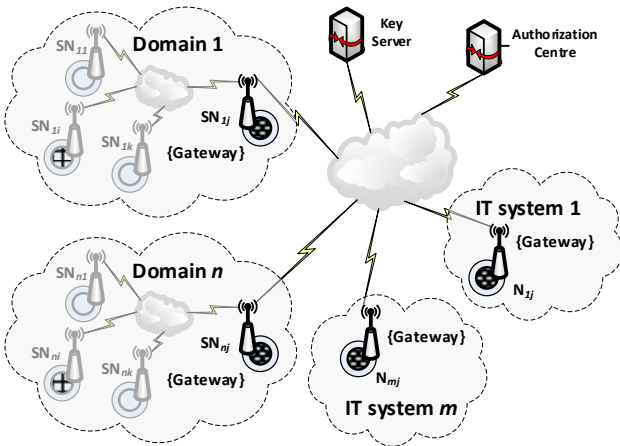


Fig. 1 The way various domains and IT systems cooperate with the key server

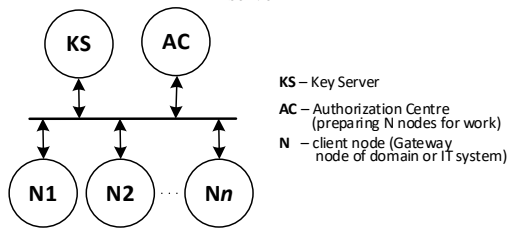


Fig. 2 Components of the KGR system

In the KGR system (Fig. 2), the source of cryptographic keys will be a separate node KS (Key Server) to which the  $N_1, N_2, \dots, N_n$  nodes (gateway nodes of domains and IT systems) will have access. The AC (Authorization Center) node is also a system component. It will be responsible for managing the KS node and adding new identifiers for the authorized  $N$  nodes in the KS node resources.

The MQTT protocol will be used to exchange data to facilitate key distribution for IoT networks. The protocol is event-driven, and data exchange between nodes is based on the publish /subscribe (Pub/Sub) pattern. The sender (Publisher) and receiver (Subscriber) are isolated from each other and use so-called "topics" to communicate. The intermediary of data exchange is a node called an MQTT broker.

### B. Assumptions for the KGR system

Since one of the data exchange parties may be a mobile IoT network node, which is classified as a restricted device [11], it is assumed that the KGR system should satisfy the following assumptions:

- for the generation of symmetric cryptographic keys, the KGR system will use a high entropy random number generator (e.g., quantum random number generator),
- the KGR system is available on the Internet,
- the KGR system only handles requests from authorized clients,
- the KS node will obtain data about authorized clients from the AC node,

- the client can be implemented in hardware, but can also be a software component that is installed on a node that acts as a Gateway,
- the MQTT protocol will be used to distribute cryptographic keys,
- each KGR node uses a local trust structure that is constructed using the TPM module,
- sensitive data stored in the resources of each node and data transmission between nodes are cryptographically protected,
- each of the  $N$ -type nodes must be appropriately prepared and then registered in the KGR system before it can start normal working - registered nodes before they begin their activities are authenticated,
- a hardware TPM module supports all cryptographic procedures in the KGR system. TPM is an implementation of a standard developed by the Trusted Computing Group [12].

### C. Procedures implemented in the KGR system

When creating a secure system and hardware and software configurations, secure procedures for using such a system and procedures for decommissioning it must be prepared. Procedures in the KGR system are implemented in two phases.

In the first phase, procedures are executed that prepare the KS node and  $N$  nodes for cooperation. These procedures include, but are not limited to, initializing the KS node, preparing credentials for the  $N$  nodes, and initializing and registering  $N$  nodes in the KS node resources.

In the second phase, procedures for generating and distributing symmetric keys for  $N$  nodes, renewing these keys, and secure data exchange between  $N$  nodes are performed. Only the procedures of the second phase will be described in detail later.

Fig 3 shows how data is exchanged between the nodes of the KGR system during the procedures mentioned above. The MQTT service is used in the KGR system only as an intermediary for data exchange and is neither the source nor the destination of the generated keys. Fig 4 shows the communication structure of the KGR system.

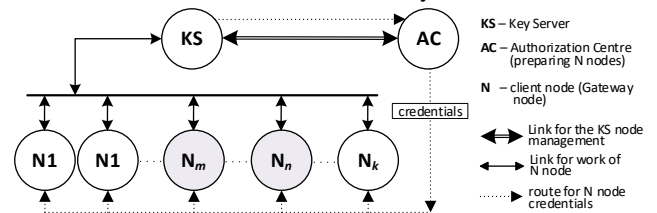


Fig. 3 The method of data exchange in the KGR system

MQTT service to exchange data requires defining a string called "topic". The main task of the MQTT broker is to forward messages published in a given "topic" by one client to clients subscribing to data with this "topic". To increase the security level in the KGR system, the content of each "topic" is random and known only to both parties of the

message exchange. In the description, strings related to the "topic" will be marked as TOPICn. Table 1 shows the purpose of the "topic" strings used.

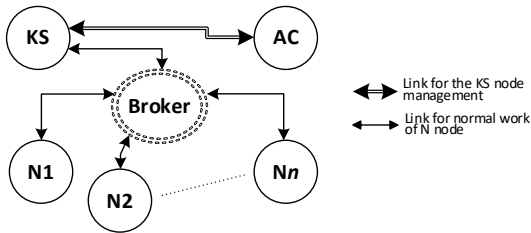


Fig. 4 The communication structure of the KGR system

TABLE 1.  
LIST OF TOPICS USED BY THE NODES.

topic	node	purpose
TOPIC0	KS	for each N node for the first request during the registration procedure
TOPIC1	KS	for subsequent requests from the given N node during the registration procedure
TOPIC2	N	for requests from KS node (i.e., published by KS node)
TOPIC3 <sub>m</sub>	N <sub>m</sub>	for requests from N <sub>m</sub> node (published by N <sub>m</sub> node)
TOPIC4 <sub>m</sub>	N <sub>m</sub>	for publishing to N <sub>m</sub> node

The rest of the paper describes the symmetric key generation and distribution procedure in detail.

### III. DATA EXCHANGE PROTOCOL FOR THE KEY GENERATION AND DISTRIBUTION PROCEDURE

#### A. Procedure description

For simplicity in the following description, we will assume that the key set will be generated for a pair of nodes N1 and N2. Node N1 will initiate the key generation procedure.

The procedure for generating and distributing symmetric keys consists of three stages:

- requesting symmetric keys,
- delivery of symmetric keys to the requesting and destination nodes,
- confirmation of delivery of keys to the destination node.

This procedure requires the following conditions, which are provided in the first phase of KGR system preparation (not described here):

- the KS node is initialized and subscribes to TOPIC0 and TOPIC1 topics,
- the N-type nodes for which keys are to be generated must be registered in the KS node's resources, and each node subscribes to a TOPIC2 topic.

Meeting these requirements means that each N-type node has a local trust structure generated and has keys to secure data exchange between these nodes and the KS node.

An Encrypt-then-MAC (EtM) [13] approach requiring two keys will be used to secure data exchange. The AES algorithm will be used for encryption, and the HMAC algorithm for hash determination - both of which are supported on hardware by the TPM module. During one key generation operation, the node KS will prepare the

symmetric key NNSK (*Node to Node Security Key*), the initialization vector for this key, and the key NNSKsign (*Node to Node Security Key for signing*). Both these keys will be used only by nodes N1 and N2. A single key generation and distribution operation require performing the following actions:

- node N1 sends a request to node KS to generate a symmetric key pair for nodes N1 and N2,
- the KS node generates the NNSK key and the NNSKsign key and temporarily stores them,
- KS node sends the generated data to N1 and N2 nodes,
- the N2 node sends an acknowledgement of receipt of the keys to the N1 node via the KS node,
- after sending this acknowledgement, the KS node removes the NNSK and NNSKsign keys from its resources.

Fig. 5 shows the sequence diagram for these activities.

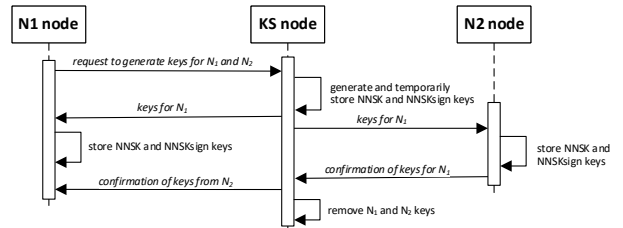


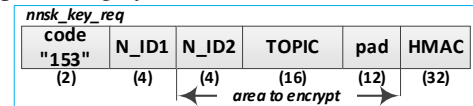
Fig. 5 Sequence diagram of key generation and distribution for node pair N1 and N2

The procedure results are the secure distribution of the generated keys and the contents of TOPIC3 and TOPIC4 to the N1 and N2 nodes. Nodes N1 and N2 will use the topics for subsequent secure data exchange among themselves using the MQTT service.

#### B. Frame structure description

The key generation and distribution procedure uses a data exchange protocol that utilizes six types of frames. For authenticated encryption data exchange will be used Encrypt-then-MAC (EtM) approach using the key pair, which are known only to the pair of nodes N<sub>i</sub> and KS and was established during the registration procedure of node N<sub>i</sub>. The list of these frames is as follows:

- nnsk\_key\_req** - request to generate a new set of keys for N1 and N2 nodes - sending TOPIC generated by N1 for publishing by N2 node, which will act as TOPIC3,



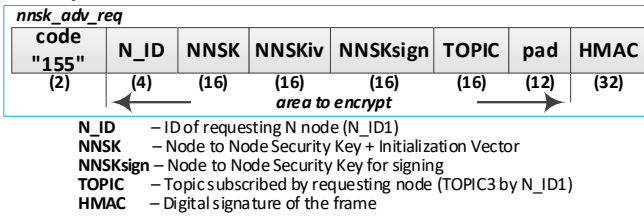
N\_ID1 - ID of requesting N node  
 N\_ID2 - ID of destined N node  
 TOPIC - Topics subscribed by N\_ID1 node (TOPIC3)  
 HMAC - Digital signature of the frame

- nnsk\_key\_ans** - response to keys request - sending a set of keys and initialization vector,

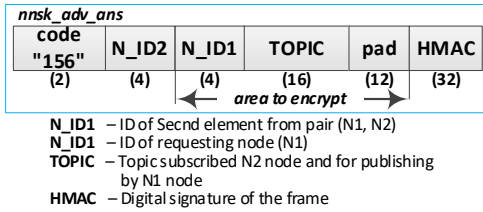


N\_ID1 - ID of requesting N node  
 N\_ID2 - ID of destined N node  
 NNSK - Node to Node Security Key + Initialization Vector  
 NNSKsign - Node to Node Security Key for signing  
 HMAC - Digital signature of the frame

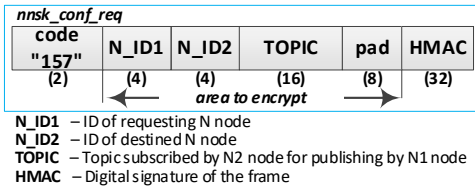
- **nnsk\_adv\_req** - advertisement of generating a new set of keys on the initiative of node N1 - sending a set of keys, initialization vector, and TOPIC3,



- **nnsk\_adv\_ans** - confirmation of receipt of the set of keys - sending TOPIC generated by N2 for publishing by N1 node, which will act as TOPIC4,



- **nnsk\_conf\_req** - confirmation of delivery of a set of keys for N2 - sending TOPIC generated by N2 for publishing by N1 node,



- **nnsk\_conf\_ans** - confirmation of the end of key distribution for a pair of nodes (N1, N2),

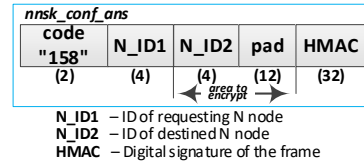


Fig. 6 shows the basic sequence diagram illustrating the operation of the data exchange protocol during the generation and distribution of the key set for a pair of nodes (N1, N2) for a use case in which the distribution procedure proceeds correctly. Diagram in Fig. 6 also includes the actions performed by the nodes involved in the data exchange during this procedure.

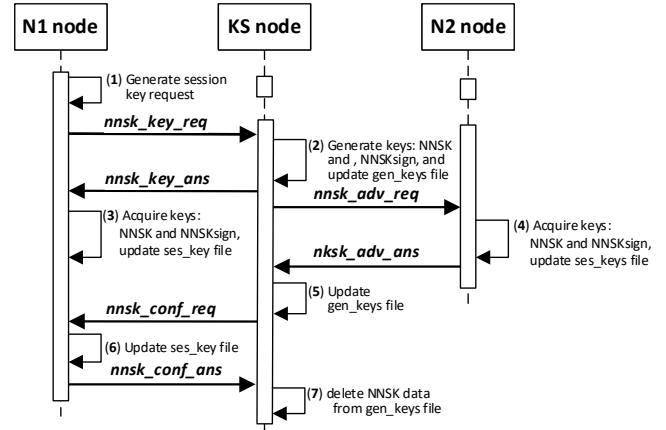


Fig. 6 Sequence diagram for key generation and distribution protocol

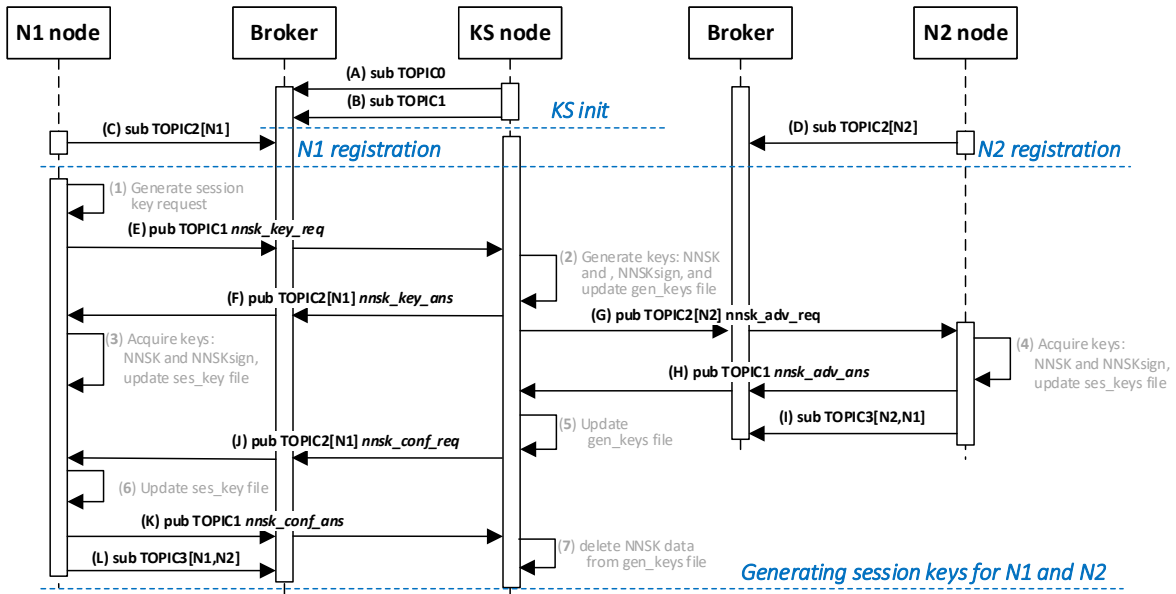


Fig. 7 The sequence diagram of the data exchange in the MQTT service for the generation and distribution key procedure.  
 Legend: **sub TOPIC** - subscription to the topic broker "TOPIC", which is known to all nodes  
**sub TOPIC[Ni]** - subscription to the "TOPIC" topic broker, which is known only to Ni and KS nodes  
**sub TOPIC[N1, N2]** - subscription to the "TOPIC" topic broker, which is known only for N1 and N2 nodes  
**pub TOPICi nnsk\_xxx\_yyy** - publishing a frame "nnsk\_xxx\_yyy" with the topic "TOPICi", where xxx={key, adv, conf}, yyy={req, ans}



In the MQTT service used to exchange data via the described protocol, a service broker plays the role of an intermediary. An important role is played in this service by appropriately using topics when sending data. Fig. 8 shows a sequence diagram illustrating data exchange in the MQTT service for the cryptographic key set distribution protocol.

### C. Experiment description

To verify the operation of the KGR system, in particular the protocol for generating and distributing cryptographic keys, a lab bench was used that included four nodes (Fig. 9). One node played the role of KS, and two nodes played the role of N. The fourth node was a broker for the MQTT service. The experiments used Mosquitto 1.5.1 software, which implements the MQTT 3.1.1 protocol.

Each node of the lab bench was implemented in Python on a Raspberry Pi 3 platform with the Raspbian Buster system. The KS, N1, and N2 nodes were equipped with a Trusted Platform Module (TPM) 2.0 (the LetsTrust TPM). In the KGR system, TPM modules were the source of random numbers. It supported cryptographic procedures to generate keys, secure data exchange between nodes, and protect data stored in the resources of the KS, N1, and N2 nodes.

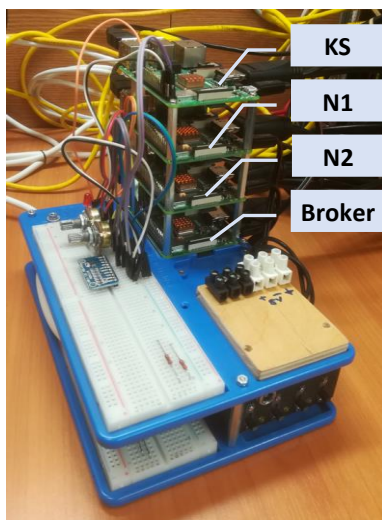


Fig. 8 View of lab bench for testing the KGR system.

During the testing of the KGR system, several experiments were conducted to confirm the correctness of the system's operation. They covered the procedures for preparing the KS node, distributing credentials for N-type nodes, registering N-type nodes in the KS node resources, and the procedure for generating and distributing keys for N-type nodes described in this paper.

## IV. CONCLUSIONS

The described protocol for secure data exchange is the main component of the KGR system. This protocol provides secure distribution of cryptographic keys over the MQTT service IoT devices often use. This protocol can be used for

IoT devices and other IT systems that interact with IoT devices. Advantages of the protocol include:

- data exchange between the components of the KGR system is cryptographically secured,
- topics used in the MQTT service have random values and are known only to the nodes that use these topics - the KGR system also provides secure distribution of these topics,
- sensitive data of the KGR system nodes are cryptographically secured,
- the TPM module supports all cryptographic procedures,
- MQTT service broker is only an intermediary of data exchange and does not participate in any other way in procedures performed in the KGR system.

### ACKNOWLEDGMENT

The presented study is the result of the author R&D activity in the IST-176 Research Task Group on Federated Interoperability of Military C2 and IoT Systems.

### REFERENCES

- [1] Dammak, M.; Senouci, S.M.; Messous, M.A.; Elhdhili, M.H.; Gransart, C. Decentralized Lightweight Group Key Management for Dynamic Access Control in IoT Environments. *IEEE Trans. Netw. Serv. Manag.* 2020, 1–15, DOI: 10.1109/TNSM.2020.3002957.
- [2] Tan, H.; Chung, I. A Secure and Efficient Group Key Management Protocol with Cooperative Sensor Association in WBANs. *Sensors* 2018, 18, 3930, DOI: 10.3390/s18113930.
- [3] Zhong, H.; Luo, W.; Cui, J. Multiple multicast group key management for the Internet of People. *Concurr. Comput. Pract. Exp.* 2016, 29, e3817, DOI: 10.1002/cpe.3817.
- [4] Ding, W.; Hu, R.; Yan, Z.; Qian, X.; Deng, R.H.; Yang, L.T.; Dong, M. An Extended Framework of Privacy-Preserving Computation with Flexible Access Control. *IEEE Trans. Netw. Serv. Manag.* 2020, 17, 918–930, DOI: 10.1109/TNSM.2019.2952462.
- [5] Mehdizadeh, A.; Hashim, F.; Othman, M. Lightweight decentralized multicast-unicast key management method in wireless IPv6 networks. *J. Netw. Comput. Appl.* 2014, 42, DOI: 10.1016/j.jnca.2014.03.013.
- [6] Kung, Y.; Hsiao, H. GroupIt: Lightweight Group Key Management for Dynamic IoT Environments. *IEEE Internet Things J.* 2018, 5, 5155–5165, DOI: 10.1109/JIOT.2018.2840321.
- [7] Abdmeziem, M.R.; Tandjaoui, D.; Romdhani, I. A Decentralized Batch-Based Group Key Management Protocol for Mobile Internet of Things (DBGK). In *Proceedings of the IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, Liverpool, UK, 2015; 1109–1117, DOI: 10.1109/CIT/IUCC/DASC/PICOM.2015.166
- [8] Yao, W.; Han, S.; Li, X. LKH++ Based Group Key Management Scheme for Wireless Sensor Network. *Wirel. Pers. Commun.* 2015, 83, 3057–3073.
- [9] J. Furtak, Z. Zieliński and J. Chudzikiewicz, Procedures for sensor nodes operation in the secured domain, *Concurr. Comput. Pract. Exp.* 2019, 32, e5183, DOI: 10.1002/cpe.5183.
- [10] J. Furtak, Z. Zieliński and J. Chudzikiewicz, A Framework for Constructing a Secure Domain of Sensor Nodes, *Sensors* 2019, 19, 2797, DOI: 10.3390/s19122797.
- [11] Borman C., Ersue M., Keranen A., Terminology for Constrained-Node Networks, RFC 7228, Internet Engineering Task Force (IETF), May 2014.
- [12] Trusted Computing Group. TPM Main Part 1 Design Principles. Specification Version 1.2, Revision 116; Trusted Computing Group: Beaverton, OR, USA, 2011.
- [13] Information technology - Authenticated encryption. 1977:2020. ISO/IEC. Retrieved November 2020.