IBIMA Publishing
Communications of International Proceedings
https://ibimapublishing.com/p-articles/44ICT/2024/4435324/
Vol. 2024 (23), Article ID 4435324

The concept of building a Trusted IoT Environment System*

Sebastian LESKA

Military University of Technology, Warsaw, Poland,
ORCID: 0000-0003-4772-4084

Correspondence should be addressed to: Sebastian LESKA, sebastian.leska@wat.edu.pl

* Presented at the 44th IBIMA International Conference, 27-28 November 2024 Granada, Spain

Abstract

A significant problem when setting up IoT networks and connecting new devices is the issue of secure data exchange between network nodes and ensuring trust between them. The aim of the article is to present the concept of a system for building a trusted environment for IoT devices. The developed system provides for the implementation of a procedure for secure joining nodes to the network and distribution of symmetric cryptographic keys using the TPM module.

Keywords: cryptographic key distribution, security in IoT, trusted IoT environment, secure data exchange, Trusted Platform Module.

Introduction

IoT (Internet of Things) technologies are increasingly used in almost every area of life. The number of IoT devices in operation is growing exponentially, which contributes to the creation of large networks connecting many smart devices. These networks can be built using a wide range of sensors, for example for weather monitoring, city monitoring, public health, industrial networks, etc. In addition, it is possible to connect home devices to the IoT network, which allows them to be automated and built a smart home.

A significant problem accompanying the construction of IoT networks is ensuring security and a trusted environment for IoT devices. According to Atzori et al (2010), many devices do not support any encryption or authentication by default, which significantly exposes them to unauthorized use. This is due to the numerous constrains of IoT devices, which relate to lower computing power, storage resources, energy resources and the use of wireless links for communication, what was described by Furtak et al (2019). Additionally, it is not always possible for IoT networks to be assisted by dedicated servers responsible for managing keys and confirming the identity of network nodes. For this reason, it is necessary to develop a system that allows the IoT network to function securely regardless of access to the Internet, which will be scalable and allow it to function without excessive administrator interaction.

The most important problem in ensuring the security of IoT networks concerns the encryption of transmitted data. The use of radio waves as a transmission medium allows for the mobility of nodes and independence from cable infrastructure, but also facilitates the interception of transmitted data. For this reason, it is necessary to use encryption algorithms. Administrators can choose between asymmetric cryptography, which requires sending only their public key to recipients, but significantly strains the nodes' storage resources, and their use requires more computing power. An alternative is to use symmetric cryptography, in which shorter keys provide a similar level of security as longer asymmetric keys. For example, according to Barker (2020), it is assumed that the strength of the AES-128 cipher is comparable to the strength of the RSA-3072 cipher. However, in case of symmetric cryptography there is the problem of key distribution.

Cite this Article as: Sebastian LESKA, Vol. 2024 (23) "The concept of building a Trusted IoT Environment System" Communications of International Proceedings, Vol. 2024 (23), Article ID 4435324, https://doi.org/10.5171/2024.4435324

When it is necessary to establish a key for a pair of nodes, traditional networks use SSL protocols, which was presented in a research study by Opplinger (2016), IKE developed by Harkins and Carrel (1998) or ISAKMP introduced by Maughan et al (1998), which supports IPSec. In IoT networks, where bandwidth is often limited due to the use of special communication interfaces such as LoRa described by Centenaro et al (2016) or xBee used by Maciaszczyk (2012), the use of these protocols is not always effective. In such situations, it is better to use the ECDH protocol presented by Baker et al in NIST Special Publication (2018) or Diffie-Hellman protocol designed by Diffie and Hellman (1976). However, these solutions have one major disadvantage – the lack of a trusted device to confirm the identity of the other party. For this reason, a lot of research is being conducted on the mechanisms for distributing symmetric cryptographic keys, which, unlike key establishment protocols, are more flexible and require a Central Entity to function. One of the first such solutions is the protocol developed by Needham and Schroeder (1978), which assumes the existence of a Central Entity responsible for generating and transmitting symmetric keys to interested parties. An interesting solution was proposed by Alhasanat et al. (2020), where by using the physical layer of IoT devices, radio waves and modulation, it is possible to distribute many keys of different lengths to many nodes at the same time. Another approach was proposed by Goyal et al (2006), according to which key generation is carried out by using the attributes of given devices. This allows you to encrypt and decrypt messages with different keys.

There are also approaches that involve the use of cryptographic modules, which take over the handling of cryptographic operations in their entirety. Furtak (2023) proposes a system using TPM 2.0 (Trusted Platform Module) to generate and store keys and to encrypt data, where the key is generated by the Central Entity at the request of the client and then transmitted directly to all indicated client nodes. A similar solution is presented by Leska et al (2021), but in this case the generated key is sent to the requesting node, and from there to the other nodes.

The article is structured as follows.. The first part describes the general operation of selected mechanisms for secure joining of nodes to the network and distribution of symmetric keys. Then, the implemented procedures and the implemented system demonstrator were discussed. The last part of the article presents the results and preliminary analysis of the built system and indicates the direction of further research.

The Concept of a Trusted IoT Environment System (TIES)

Procedures for joining nodes to the network

Developing a procedure for securely joining a node to the network is crucial for a properly functioning and trusted environment. In IoT networks, where CA (Certificate Authority) is often absent, there is a problem of confirming the identity of nodes. Connecting a new device to the network is a critical stage of the system's operation, in which it is easiest to plant a sniffer aimed at unauthorized interception of network traffic.

The use of key establishment protocols alone may not be sufficient, because there is no way to confirm that the key is established with our device and not with the adversary's device. Password authentication should also be excluded, because a network like this would be poorly scalable and would require user interaction on client nodes.

The developed concept proposes the use of factory-generated one-time access keys, which will then be entered by the administrator in the Central Entity. These keys will enable the generation of a symmetric key used to secure data transmission between the client device and the Central Entity, so that network communication between the two nodes is encrypted from the very beginning. Implementing such a solution will prevent a potential adversary from planting their own device, because the administrator will not have the appropriate key, and thus will not be able to add such a device to the network. The exact operation of the procedure will be discussed later in the article.

Choosing a key distribution scheme

Baker, in a NIST Special Publication (2020), standardized the KDC (Key Distribution Center) and KTC (Key Translation Center) schemes. The KDC scheme assumes the use of the Central Entity (CE) as a generator of symmetric cryptographic keys. Node N1, which wants to establish connection with node N2, sends a request to the CE to generate a Session Key (SK). This request is encrypted with a Master Key (MK) shared by N1 and the

CE. CE generates SK key and then encrypts it and its copy with the N1 and N2 node Master Keys, respectively. Packages prepared in this way are forwarded to these nodes. As a result, both customers share a Session Key so that data transmission between them can be encrypted.

The KTC scheme also assumes that the Master Key is shared by the Central Entity with each client. The difference between KTC and KDC is that the Session Key is generated by the requesting node N1 and then sent in encrypted form with the Master Key to the CE to be translated for node N2. Translating a Session Key involves decrypting it with the Master Key of node N1 and encrypting it with the Master Key of node N2, and then passing that key only to node N2. The solutions described were tested by Leska in 2023 to compare their performance and safety. The concept of the system described in the following part of the article assumes the use of the KTC scheme for key distribution.

Key Translation Center

KTC enable the distribution of one symmetric key for a pair of nodes as well as one symmetric key for many network nodes. An additional advantage of using this key distribution scheme is that the CE can be used as a trusted third party. In this section, the following symbols are used to describe the schemes:

CE – Central Entity responsible for key translation and distribution;
MK_{Ni} – Master Key of Ni and CE nodes, established by one of the key establishment protocols (e.g. ECDH);
SK – Session Key generated by N_i;
$E(MK_{N1}, [Mi, …, Mj])$ – message encryption [Mi,…, Mj] with a Master Key MK_{N1}, where [Mi,…, Mj] is the concatenation of individual Mi messages.

The use of KTC requires first establishing a Master Key with CE, different for each client node. For this purpose, the Diffie-Hellman protocol or ECDH can be used. This key is only used to secure data transmission between CE and Ni nodes. When the N1 node wants to connect to the N2 node, it generates a symmetric Session Key using local resources, then encrypts this key and the N2 node address with the Master Key and sends this request to the CE. CE decrypts the request, then encrypts the received Session Key and the address of node N1 with the Master Key of node N2 and sends such a packet to node N2. After decrypting the message, the N2 node shares the Session Key with the N1 node.

Transferring the Session Key from CE to N2 can be done in two methods – directly and indirectly. The indirect method involves sending the translated Session Key back to node N1, which then forwards this packet to node N2. In the direct method, the package is sent directly from CE to N2. Figure 1 shows a scheme of Key Translation Center by using the indirect method (Figure 1 A) and the direct method (Figure 1 B).

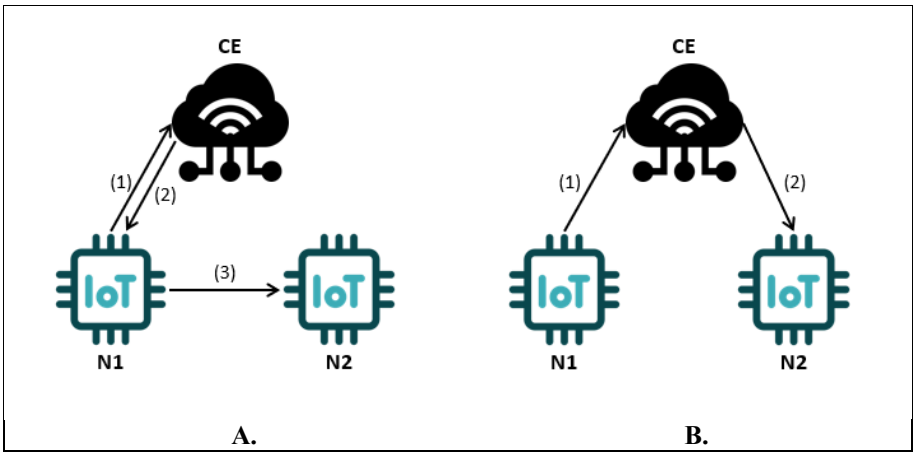

Figure 1 Scheme of KTC key distribution using indirect (A) and direct (B) communication

Key distribution using KTC, as in the case of KDC, takes place in three main stages:

1) **$E(MK_{N1}, SK)$** – the N1 node sends to CE a request to translate the Session Key (SK), encrypted with the Master Key of the N1 node (MK_{N1}).

2) **E(MK_{N2}, SK)** – CE sends a message containing the Session Key encrypted with the Master Key of the N2 node (MK_{N2}).
3) **E(MK_{N2}, SK)** – N1 node sends the package received from CE to N2 node. The package was previously encrypted by CE with the MK_{N2}, therefore only this node will be able to read the read message.

Figure 2 and Figure 3 show the sequence diagrams for the KTC scheme using the indirect method and the direct method, respectively.

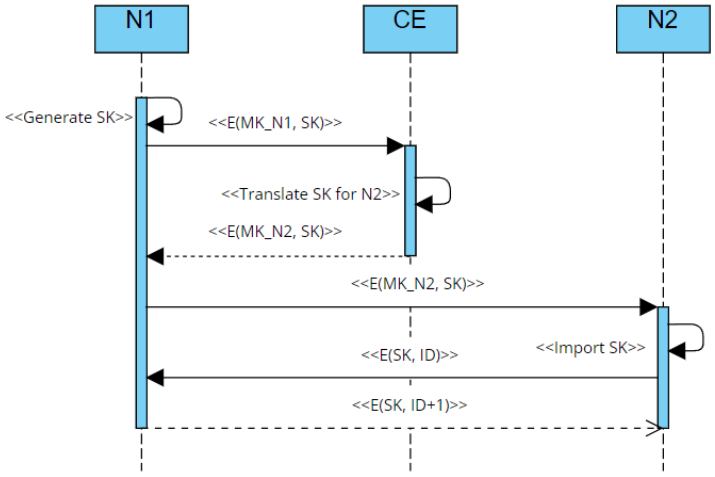

Figure 2 General scheme of symmetric keys distribution by indirect KTC method

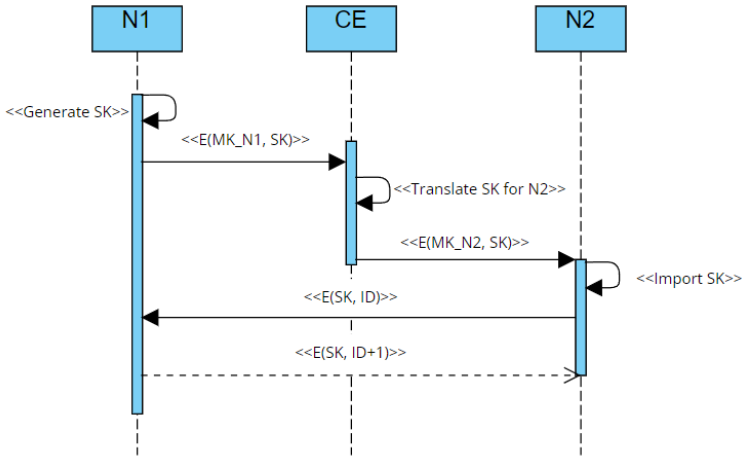

Figure 3 General scheme of symmetric keys distribution by direct KTC method

Proposed Trusted Iot Environment System

Developing a system that provides IoT nodes with a trusted communication environment and security of the processed data requires the implementation of appropriate procedures that will be flexible and scalable. The proposed concept of the system assumes the use of the TPM 2.0 module, which is a hardware cryptographic module. This module uses real physical phenomena to randomize numbers, and as a result, the cryptographic keys it generates have a strong source of entropy, what was described by Arthur et al (2015). In addition, it acts as a safe and it is impossible to export the key from it in plain text. In the designed system, each network node is equipped with a TPM module, which is used to generate and store keys, as well as to encrypt and decrypt data. In this section, the following symbols are used to describe the schemes:

MK_{Ni} – Master Key of Ni and CE nodes;
SK – Session Key generated by N_i;
RK_X – RSA Public Key of X node, generated by TPM module and used for key duplication and distribution.

E(MK_{N1}, [Mi, …, Mj]) – message encryption [Mi,…, Mj] with a Master Key MK_{N1}, where [Mi,…, Mj] is the concatenation of individual Mi messages.

Joining to the network

An important stage of the system's operation is connecting devices to the network. In IoT networks where dozens or hundreds of devices are used, it is necessary to implement procedures that do not require interaction with each device added to the network. It is also important to have control over the devices that are added, so that it cannot happen that among the added devices there is a device planted by a potential adversary, whose purpose is to collect data on the functioning of the network.

For this purpose, the designed system proposes to generate a one-time access key (16-byte random number in hex notation) using the TPM module, which is attached to the device packaging. After placing the device in the target environment, the one-time access key is entered in the Central Entity. This key is used as a seed to generate an AES-256 key, which will be used as the Master Key. In this way, the Central Entity and the client device have a shared key from the beginning, and communication between them is encrypted from the first message. Once the Master Key is determined, the devices proceed to the initial data exchange, during which they exchange RSA public keys generated by the TPM, which are used during the distribution of the key, and then the Central Entity sends the updated list of authorized devices to all nodes. Figure 4 shows the sequence diagrams of attaching a node to the IoT network.

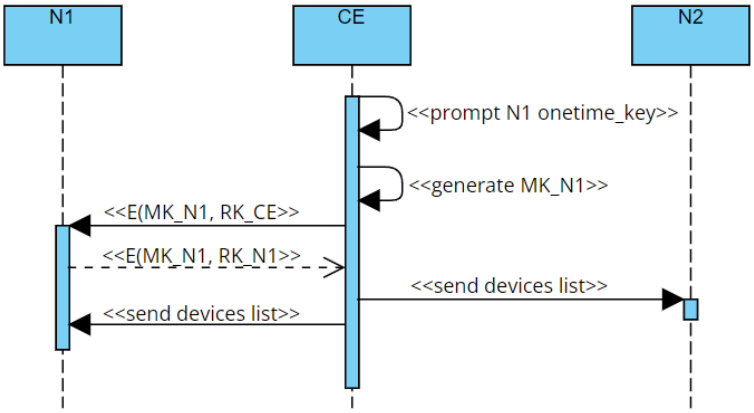

Figure 4 The procedure of attaching a node to the network

Generating and distributing symmetric keys

The most important component of the system is the TPM 2.0 module. This module is a required element of every network node. It supports mechanisms such as the creation of local trusted structures, the generation of symmetric and asymmetric keys, encryption and mutual authentication of nodes. Figure 5 shows a sequence diagram illustrating the operation of key distribution procedure in the Trusted IoT Environment System (TIES) using the TPM module.

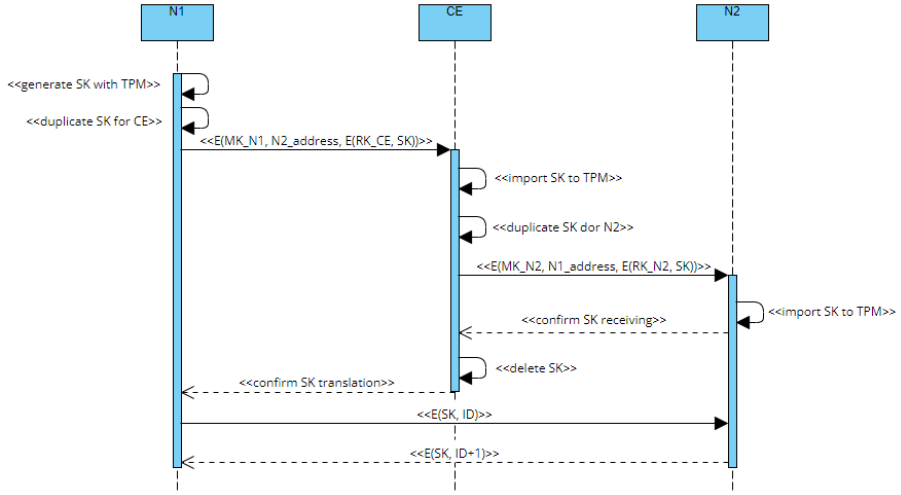

Figure 5 Sequence diagram of key distribution in the proposed TIES System

Node N1 that wants to connect to node N2 generates and duplicates an AES-256 symmetric key (Session Key) using the local TPM. Duplication involves exporting the key stored in the TPM resources and encrypting it with the recipient's RSA public key (in this case, the CE node). A message encrypted with the MK_{N1} key is sent to CE, containing the address of the N2 node and the SK key encrypted with the RSA key.

After receiving the package, CE decrypts the message, verifies who the key comes from and whether there is an N2 node on the list of authorized devices. If successful, the SK key is imported into the TPM, where it is decrypted with the RSA private key, and then the received key is encrypted with the RSA public key of the N2 node and exported. A package encrypted with the MK_{N2} key is sent to the N2 node, which contains the address of the N1 node and the SK key.

The N2 node, after receiving the message, imports the received SK key to the local TPM and confirms the receipt of the package to the CE node. The CE node does not store any Session Keys, therefore it deletes the SK key received from N1 and confirms to the N1 node that the key translation operation has been correctly performed. Node N1, in order to confirm the operation of the key, sends a random number ID to N2 encrypted with the SK key. The N2 node responds with the number of IDs incremented by 1.

TIES System Demonstrator

The concept has not yet been fully tested, but preliminary tests have been carried out. A system consisting of five nodes – the Central Entity and three client nodes – was implemented. Each of the devices is equipped with a TPM 2.0 module responsible for performing cryptographic operations and a LoRa (LongRange) interface, which enables wireless communication over distances of up to several kilometers. Figure 6 shows the hardware components of the system node, and Table 1 shows the specifications of the node components.

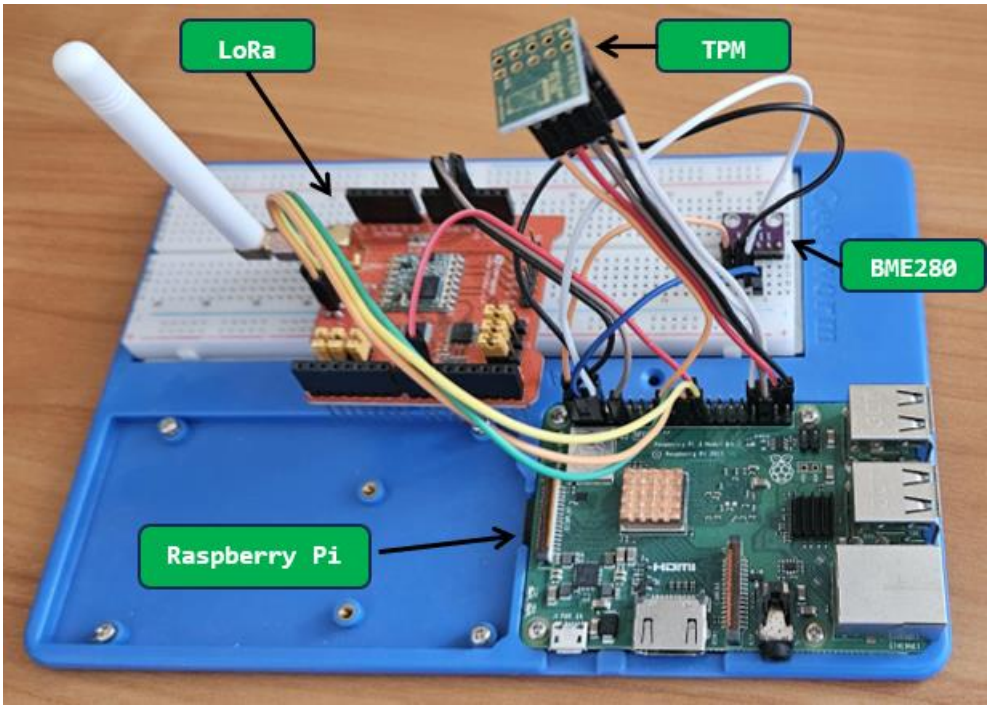

Figure 6 Hardware components of the TIES system node

Table 1 Technical parameters of the TIES system node

Name	Model	Properties
Controller	Raspberry Pi 3B	---
TPM modul	LetsTrust TPM 2.0	Infineon Optiga SLB 9670 TPM 2.0 Firmware >= 7.85 True Hardware Random Number Generator Voltage: 3.3V
Communication module	Dragino LoRa Shield 1.4	Układ LoRa: Semtech SX1278 Częstotliwość: 868 MHz Komunikacja: Half-duplex Voltage: 3.3V
Sensor	BME280	Interface: I2C Measurements: temperature, humidity, pressure Voltage: 3.3V

Experimental results

During the experiments, 5 measurements were carried out for each of the procedures and at each node. During the operation of the system demonstrator, the duration of all procedures was measured. The averages of the collected results are presented in **Error! Not a valid bookmark self-reference.**.

Table 2 Average busy time of TIES nodes

	Busy Time of CE[s]	Busy time of N1 – requesting node [s]	Busy time of N2 – receiving node [s]
Joining to the network	44,7		1,12
Key distribution	47,5	53,7	1,32

First, the procedure for connecting a node to the network was tested. This time is the same for the joining node and the CE node. On average, it takes 44.7 seconds to establish a Master Key, exchange RSA public keys, and

distribute the current list of authorized devices to all nodes. The other nodes, which only receive the new list of devices, require 1.12 seconds.

The next procedure tested was the symmetric key distribution procedure. In this case, the busy time of each node is different. The N1 node, which is the node requesting the key translation, is working the longest. This is due to the need to generate a symmetric key that will be forwarded. Once the key is generated, it is sent to the CE node, which will only start its work at this point. The time it takes to receive, translate, and transmit the key to node N2 and send an acknowledgment to node N1 is 47.5 seconds on average. The N2 node, which only receives the symmetric key, needs the least time. Its operation lasts an average of 1.32 seconds. The total time of operation of the N1 node, from the start of key generation to the receipt of the confirmation of its delivery, is on average 53.7 seconds.

Conclusion

The TIES system meets the assumed goals and allows for secure connection to the network and secure data transmission between network nodes. From the ground up, it is designed to ensure high safety standards, taking into account the possible limitations of the devices. Thanks to the TPM module, the device is relieved of performing cryptographic operations, and in addition, the use of physical phenomena increases the security of the operations performed. Also, the LoRa module, which requires low power consumption, puts much less strain on the system during data exchange.

The system is in the early stages of design, so it is necessary to refine all procedures, mainly in terms of their efficiency and speed. Currently, the system has been designed according to the "need-to-know" principle, so it already ensures the correct level of safety at this stage.

During the work, several problems emerged, mainly concerning the reliability of data exchange. The data transfer speed is slow, and in addition, there is often a situation in which the received packets contain noise and it is necessary to retransmit them. In future research, a lightweight communication protocol for the LoRa interface will be developed that will allow for fast and reliable data exchange.

An important issue is also the issue of ensuring the authenticity of the data, i.e. the ability to confirm that the data comes from a trusted device. This problem is largely solved by network joining procedures and encrypting data with a symmetric key unique to a pair of nodes. Nevertheless, it is worth implementing an additional authentication protocol that will replace the CA found in Internet networks and will be an additional security for the network. Further research will also focus on developing a dedicated authentication protocol.

Acknowledgements

This work was financed/co-financed by Military University of Technology under research project UGB 702/2024

References

- Atzori L., Iera A., Morabito G. (2010), ‘The Internet of Things: A survey’, *Computer Networks*, DOI:10.1016/j.comnet.2010.05.010.
- Furtak J., Zieliński Z. and Chudzikiewicz J. (2019) ‘A Framework for Constructing a Secure Domain of Sensor Nodes’, Sensors 19(12), pp. 2797, DOI:10.3390/s19122797.
- Barker E., (2020), ‘Recommendation for Key Management, Part 1: General’, NIST SP 800-57 Part 1 Rev. 5. DOI:10.6028/NIST.SP.800-57pt1r5.
- Opplinger R. (2016) ‘SSL and TLS. Theory and Practice’, Second Edition, *Artech House* (Verlag), ISBN: 978-1-60807-999-5.
- Harkins D. and Carrel D. (1998) ‘The Internet Key Exchange (IKE)’, RFC 2409, *Internet Engineering Task Force (IETF)*.
- Maughan D., Schretler M., Dchneider M. and Turner J. (1998) ‘Internet Security Association and Key Management Protocol (ISAKMP)’, RFC 2408, *Internet Engineering Task Force (IETF)*.
- Centenaro M., Vangelista L., Zanella A. and Zorzi M. (2016) ‘Long-Range Communications in Unlicensed Bands: The Rising Stars in the IoT and Smart City Scenarios’, IEEE Wireless Communications.
- Maciaszczyk R. (2012), ‘Data transmission from unmanned aerial vehicles using XBee modules’, *Pomiary Automatyka Kontrolna*.

- Baker E., Chen L., Roginsky A., Vassilev A., Davis R. (2018), 'Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography', NIST SP 800-56A Rev. 3, DOI:10.6028/NIST.SP.800-56Ar3
- Diffie W., Hellman M. E. (1976), 'New Directions in Cryptography.' *IEEE Trans. Inf. Theory*. 1976;22:644–654. DOI:10.1109/TIT.1976.1055638.
- Needham R. M., Schroeder M. D. (1978), 'Using encryption for authentication in large networks of computers' Communications of the ACM, vol. 21, no. 12, pp. 993–999.
- Alhasanat, M. et al., A Physical-Layer Key Distribution Mechanism for IoT Networks, Mobile Networks and Applications, 25(1), pp. 173–178, 2020, DOI: 10.1007/s11036-019-01219-5.
- Goyal, V., Pandey O., Sahai A., Waters B. (2006), 'Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data', Proceedings of the 13th ACM conference on Computer and communications security, DOI:10.1145/1180405.1180418
- Furtak J. (2023), 'The Cryptographic Key Distribution System for IoT Systems in the MQTT Environment', Sensors, 23(11), p. 5102, 2023, DOI:10.3390/s23115102.
- Łeska S., Furtak J., (2021), 'System for generating and renewing symmetric cryptographic keys for sensor network nodes using LoRa communication', 38th IBIMA International Conference, Sevilla.
- Leska S. (2022), 'Performance tests of symmetric key distribution systems for IoT networks', 42th IBIMA International Conference, Sevilla.
- Arthur W.,Challener D. (2015) 'A Practical Guide to TPM 2.0 Using the Trusted Platform Module in the New Age of Security', *Apress Open*, New York.