Analysis of security protocols used in the Internet of Things

Iskandar Olimov [a)], Nuriddin Jabbarov

Tashkent University of Information Technologies named after Muhammad Al-Khwarizmi, Tashkent, Uzbekistan

[a)] *Corresponding author: olimoviskandar89@gmail.com*

Abstract. This article analyzes security protocols used in the Internet of Things (IoT). It provides a detailed overview of the protocols used in IoT environments. Additionally, the study examines encryption algorithms, hash functions, and a number of other key characteristics used in these protocols.

INTRODUCTION

In the era of current information technology development, all work is being automated and various conveniences are being created. At the same time, the dwellings necessary for human living create a special impression on people, especially with the help of these technologies. In a modern house, it is required to incorporate several modern requirements, such as automated devices, air conditioning, ventilation systems, and so on.

In today's information society, the Internet of Things is becoming more relevant and popular as a result of the expansion of its fields of application. IoT devices collect data from the real environment and transmit it over networks. The Internet of Things (IoT) has now become globalized and has become a dominant research field due to its applications in various fields [1,2]. Because smart transportation, smart logistics, smart healthcare, smart environment, smart infrastructure (smart cities, smart homes, smart offices, smart shopping centers, industry), smart agriculture, etc. contribute to the development of society, it connects many IoT devices to the real world, from tiny sensors to servers, with a number of challenges.

FIGURE 1. Internet of Things

As data flows across billions of smart devices running on different platforms during the transition from sensors to servers, there will be a variety of unprecedented problems for their owners or users (for example, security, privacy, interoperability, durability, support, etc.) that arise with the technology's weaknesses [1,2].

Analyzing and improving the protocols used to ensure the confidentiality and integrity of data in the Internet of Things is one of today's urgent tasks.

EXPERIMENTAL RESEARCH

The protocols used in the Internet of Things can be divided into the following categories according to their general status:

TABLE 1. Protocols used in the Internet of Things

Standardized protocols	These protocols are widely adopted and standardized by international organizations or industry associations, have well-defined specifications, and are commonly used in various IoT applications	MQTT CoAP Zigbee Z-Wave Thread
Industry-Specific protocols	These protocols are designed for specific networks or applications and may not be widely used outside of these domains. They consider the unique requirements and constraints of specific sectors	BACnet KNX DNP3 MODBUS OPC-UA
Cellular communication protocols	Various cellular communication protocols are used to enable IoT devices to connect to the Internet and transmit data	NB-IoT LTE-M 5G
Classic protocols	Classical protocols refer to classical communication protocols that were not originally designed for IoT, but are still used in some applications	HTTP TCP/IP SNMP UDP
Proprietary and low-power protocols	Proprietary protocols are developed by specific companies or organizations and are not open or standardized. These protocols will be designed for specific IoT platforms or devices	LORA Sigfox EnOcean INSTEON

MQTT (Message Queuing Telemetry Transport) protocol. MQTT is a communication protocol designed to facilitate dialogue between two or more systems or devices.

That is, whether it is done by software or hardware (or both), it carries out the transfer of data by various means and with a specified format [4,6].

In the Internet of Things, two or more devices communicate with through a set of established standards and protocols to communicate and understand each other. Due to the large number of Internet of Things devices, these protocols must meet the requirements beyond the bandwidth, speed, and other limits, that is, it is necessary to be able to add more connected devices without affecting the global system.

MQTT functions as a lightweight messaging protocol which supports Internet of Things applications while offering multiple security measures to protect message confidentiality, integrity, and authentication.

The mechanisms include TLS (Transport Layer Security)/SSL (Secure Sockets Layer), which function as encryption and authentication protocols for MQTT. The system ensures complete security for all communications between brokers and MQTT clients by preventing unauthorized monitoring and alteration of data. The system uses certificates to authenticate servers during MQTT over TLS/SSL operations. It also allows client authentication in specific situations [3,4,10].

Authentication and Authorisation: The application of authentication methods is one of the ways MQTT brokers can verify the identity of the clients. The methods can be as straightforward as the use of a username and a password or can be more secure like client certificates. By means of access control lists (ACLs), it is determined the clients' rights over the topics for publishing or subscribing.

Quality of Service (QoS) levels: The MQTT protocol has established its own criteria regarding the delivery of messages among the different QoS levels. QoS 0 (at most once) is the most primitive of the three, followed by QoS 1 (at least once), and QoS 2 (exactly once), the most reliable one, as the third. The three QoS levels bear different degrees of reliability. The proper QoS meets the requirement of the significance of the data being sent.

Message Encryption: Besides the use of TLS/SSL, MQTT ensures the end-to-end encryption of the payloads. In other words, the actual data being sent through MQTT messages gets encrypted; otherwise unauthorized persons would be able to access it in case the connection is broken.

Broker Security: The security of an MQTT broker is of utmost importance; reason why one should not only implement strong authorization and password controls having their minimum lengths set, but also keep the broker updated regularly as part of known vulnerability management.

Network Segmentation: A very effective way to protect against intrusion is to properly segment IoT networks in such a way that they isolate MQTT traffic and networks; it could help limit an attack.

Security Testing: Regularly you would conduct testing and vulnerability assessments of the MQTT infrastructure, make the findings known and do the necessary remediation. An example of this would be penetration testing.

By applying these security measures, MQTT communication can be made more secure and reliable, which serves to ensure the confidentiality, integrity, and authenticity of data transmitted in IoT applications.

Constrained Application Protocol (CoAP). It is a crypto-based web communication protocol designed specifically for constrained devices and low power networks, mostly in IoT applications. Its main application is to very effectively provide communication among the devices which have low power, low processing speed, and small memory.

CoAP is touted as a lightweight substitute for the conventional protocols like HTTP. It brings in UDP which is the underlying transport protocol rather than TCP thus firstly by nature, reduces the overhead that is accompanied with connection establishment and secondly with reliable data transmission.

CoAP includes security features that allow communication to be secured between the devices in the Internet of Things applications during the whole process [5,6]. The security features in CoAP are put in place to prevent unauthorized access and data corruption and to keep the communications' integrity and confidentiality. CoAP protocol security has the following main components:

DTLS (Datagram Transport Layer Security): CoAP might get DTLS as a communication security which is a slower but safer way of the UDP transport protocol usage. DTLS cuts off the listening and modification of messages but it also prohibits the unauthorized persons to be the recipients of CoAP messages. DTLS is built on the TLS protocol and aims at being seamlessly operated in low-resource settings.

Authentication and authorization: CoAP supports multiple authentication mechanisms to validate the identities of the clients and servers. These methods include PSK (Pre-Shared Keys), where shared secret keys are used between clients and servers, and Raw Public Key (RPK) certificates that allow asymmetric key authentication. CoAP also permits the use of external authentication protocols such as OAuth or OpenID Connect for user verification [4,5].

Modes of secure communication: CoAP provides modes for secure communication. The modes are "noSec" mode, where no security measures are applied at all, "PSK" mode, where Pre-Shared Keys are used as a means of authentication and encryption, and "certificate" mode, where digital certificates are used for both authentication and encryption.

Control of resource access: CoAP has built-in access control provisions, which enable the restriction of client access to selected resources. Access controls can be based on either authentication credentials (like username-password or tokens) or on the access control lists (ACLs) connected to the resources.

The implementation of security in CoAP-enabled systems must be a process that takes into account the particular security needs, the capabilities of the devices, and the nature of the deployment environment. Key management, certificate management, and secure configurations must be done properly to ensure that the security of CoAP protocol is not compromised.

ZigBee. It is a low-power LAN protocol that complies with the IEEE802.15.4 standard. The ZigBee technology doubles as a wireless communication technology that is characterized by short-range, low-complexity, low-power, low-speed, and low-cost features. The technology is mainly employed in various electronic devices that have a short range of operation, consume little power, and have low data transmission rates [4,7] as well as in routine applications having periodic data, uninterrupted data, and low-time data transmission.

Zigbee, the wireless communication protocol that is very much in use in the Internet of Things applications, has provided several security mechanisms so that the communication between the devices is secure and reliable. Zigbee security is based on several key elements, namely data confidentiality and authentication.

Authentication: At the start of the network setup, the devices exchange cryptographic keys to verify one another's identity. This way, unauthorized devices are kept out of the network and, thereby, are prevented from taking part in the communications.

Key setup: Zigbee uses the TCLK (Trust Center Link Key) protocol. This protocol allows devices to securely exchange encryption keys with the Trust Center, which is the central part responsible for network security. A trust center creates and distributes keys to network devices, ensuring secure communication [1,7].

Z-wave. This protocol is a popular wireless communication protocol designed specifically for Internet of Things applications in the home automation sector. It offers several features and benefits suitable for IoT deployments [7]. How to use the Z-Wave protocol in the Internet of Things:

– Home automation;
– Wireless connection;
– Compatibility;
– Security;
– Energy efficiency.

Z-Wave protocol affords an array of security features which not only make the IoT environment secure but also that of the users' privacy. The following are some of the major Z-Wave protocol aspects vis-à-vis security matters for IoT:

Encryption: Z-Wave is the one that implements the AES-128 (Advanced Encryption Standard) symmetric encryption approach thus, ensuring the devices' communication is secured. In fact, encryption does not allow the data flow over a network to be detected and thus it can neither be intercepted nor access.

Authentication: Z-Wave utilizes authentication systems to establish that devices are who or what they say they are within the network. Each Z-Wave device has a unique node ID and a secure node authentication key.Devices use NAK to authenticate and communicate securely with other devices on the network.

Secure join: When a new device is added to a Z-Wave network, a secure join process is performed to ensure that only authorized devices join the network. This process involves exchanging cryptographic keys and securely authenticating the new device before allowing it to participate in the network [3,4].

TABLE 2. Encryption algorithms used in Internet of Things protocols

Internet of Things	Encryption algorithms
MQTT	AES, RSA, ECC
CoAP	AES, RSA, ECC
DTLS	AES, RSA, ECC
Zigbee	AES, ECC
Z –Wave	AES

BACnet (Building Automation and Control Networks). This is a communication protocol specially developed for building automation and control systems in the Internet of Things. It allows devices and systems within the building to exchange information, such as HVAC (heating, ventilation, and air conditioning) systems, lighting, access control, and energy management systems [9].

The BACnet protocol has various features and functions explicitly designed for building automation in the IoT and these features and functions are listed below:

Device discovery and management: BACnet has device discovery mechanisms allowing for devices to be automatically discovered and added to a network. It has mechanisms to discover and manage devices by means of reading and writing device properties, inquiring what services may be available, and discovering the status of devices.

Network topologies: BACnet can be used with various network topologies and can be used for both wired and wireless systems, that is, it can be used over Ethernet, RS-485, IP and other types of communication media that occur within building automation systems. This flexibility in the BACnet protocol helps it to be open and flexible when it comes to adapting to and potentially integrating with existing infrastructure.

Security: BACnet supports various security mechanisms to protect the integrity and confidentiality of data exchanged between devices. It includes authentication mechanisms for device identification and authorization, as well as encryption methods for secure communication. However, the level of security implementation may vary depending on the specific BACnet devices and systems in use.

KNX. It is a standardized communication protocol used in building automation and control systems. It is designed to provide interoperability and integration between various devices and systems within a building, including lighting, HVAC, security systems, energy management, and more. KNX is widely used and supported by many manufacturers in the building automation industry [8].

Security features of the KNX protocol in the Internet of Things:

Secure device configuration: Proper configuration and protection of KNX devices within the system is essential. This includes setting strong passwords, disabling unnecessary services or features, and keeping devices and software updated to address known vulnerabilities [5,8].

Authentication and access control: One of the key features that KNX has is the ability to support authentication and this is among the reasons why it is widely used in smart building automation. Device identity verification by

means of authentication not only allows communication with devices within the KNX system but also interaction with the system. Also, access control measures will be imposed wherein the access to vital functions or data on the KNX network will be restricted.

Secure communication: It is highly advisable to employ more encryption methods besides the existing ones to secure the data that is being transmitted over the KNX network. Encrypted communication can either be implemented using IPsec or TLS (Transport Layer Security) or the data may pass through the KNX systems unencrypted.

Physical security: There is no doubt that insulating the physical infrastructure of the KNX system is just as important. The protection of devices, cables, and network gear should be such that nobody unauthorized can touch them.

Network segmentation: One of the ways that separating the KNX network into logical sub-networks can enhance security is by having each zone with its own specified policy and controls that enforce restrictions depending on the communication type between trust levels or device roles. This not only aids in the prevention of security breaches but also in the limitation of their effects on the whole network.

Intrusion detection and monitoring: The installation of an Intrusion Detection System (IDS) or security monitoring software can be very helpful in spotting and notifying you of any unattended or harmful activities in the KNX system.

Regular updates and audits: Manufacturers provide security and updates that are incorporated into the devices so that the KNX devices operate with the latest technology.

Even though it is easy to think of security as a single-layered approach it really consists of several layers and the use of extra security measures beyond those offered by the protocol itself is important for a strong and secure KNX IoT system.

DNP3 (Distributed Network Protocol). DNP3 is a way for devices to talk to each other in industrial automation, control systems, and even some Internet of Things stuff. It started in the electricity biz but now is in water, oil, gas, and transport too.

For security, DNP3 has tools to keep data safe and private. It uses authentication and encryption to secure how devices communicate. Plus, it can spot and stop any meddling or unauthorized peeps.

DNP3 gets used a lot in important setups and real-time management. Because it plays nice with other systems and has good security, it's great for IoT in lots of industries. It helps devices and systems chat easily in industrial setups, so you can control and watch industrial stuff with no problem.

MODBUS. It is a communication protocol widely used in industry that is very easy to understand and use, making it easy to connect devices in the same network.

Communication methods: MODBUS has several communication methods. There are Modbus RTU (which uses serial communication hardware like RS-232 or RS-485), Modbus ASCII (which transmits messages in readable text format), and Modbus TCP (which transmits over the Internet via Ethernet with the same protocols).

Addresses: Every device that connects on the MODBUS network has a unique address which is like a house number. This way, the master computer is informed clearly about which device it is communicating with. The address is placed at the beginning of the message. Securing it: In case of using MODBUS TPC, it is better to apply message encryption with TLS or IPsec. This way, the unauthorised can neither eavesdrop nor tamper with the data transfers during the communication.

Secure communication channels: In the case of using MODBUS over TCP/IP (MODBUS TCP), communication channels should definitely be secured with encryption protocols like TLS (Transport Layer Security) or IPsec. This will effectively protect the data coming from and going to the devices against snooping and unauthorized alteration.

TABLE 3. Industry-specific and cellular communication protocols

Protocol	Inherent Encryption Support	Encryption Approach
BACnet	No	TLS, IPsec
KNX	No	VPNs, IPsec, OpenVPN
DNP3	Yes	Message-level encryption
MODBUS	No	TLS, IPsec, Manual encryption

NB-IoT (Narrowband Internet of Things). This is a specifically tailored protocol for the communication of devices over low-power wide-area networks (LPWAN) in IoT. It also allows providing a connectivity solution that is both efficient and reliable for numerous IoT applications.

Sometimes, security related to the data transmission of the NB-IoT network is considered to be the main requirement. Security measures such as encryption through IPsec or TLS (Transport Layer Security) should be taken up so as to create secure communication channels among the devices and the network. This will, in turn, make sure that no one else but the end-users of the network have access to the data, and that the data are not altered in or during the process.

HTTP (Hypertext Transfer Protocol). It is the most frequently employed application-layer protocol for browsing the Web. HTTP has enabled web browsing originally but now it is also taking part in IoT by making data exchange between IoT devices and servers possible.

In the IoT scenario, HTTP usage raises the security issue first and foremost. HTTP is susceptible to eavesdropping and tampering because it transmits data in a legible form. To get around the issue, switching to HTTPS (HTTP Secure) is the most viable option as it relies on SSL/TLS to keep the data secure during transmission between the devices and the server. Furthermore, setting up secure authentication and authorization methods is also necessary to make sure that only the devices intended for that purpose can reach and communicate with the server [6,12].

HTTPS (HTTP Secure): Always use HTTPS over HTTP whenever possible. The main point of HTTPS is that it scrambles and so protects through SSL/TLS the data transfer between IoT devices and the server from interception and other kinds of attacks. The costs associated with HTTPS setup consist of obtaining and installing an SSL/TLS cert on the server that then enables secure communication using port 443 – the same port assigned to regular HTTP.

TCP/IP (Transmission Control Protocol/Internet Protocol). TCP/IP is a computer networking protocol also called Internet Protocol (IP), as it is synonymous with the Internet. The identification of the computer that is part of the network is done by an IP address. This protocol suite also encompasses the User Datagram Protocol, commonly abbreviated as UDP.

IoT devices cannot connect to the Internet without TCP/IP. They can be allocated addresses in either IPv4 or IPv6 form and thus routed via data packets through specific networks. Therefore, these protocols are the ones that make it possible for not only mobile devices to talk to one another but also to get connected to the Internet [12,13].

The TCP/IP suite consists of TCP and UDP. TCP is the protocol that gives reliable communication and thus is known as connection-oriented; TCP's reliability relies on the acknowledgments, retransmissions, and flow control that are its inherent characteristics. Conversely, UDP is a light-weight and non-intrusive communication channel that is best suited for those applications where the speed is preferred to reliability.

SNMP. SNMP is the protocol most commonly used for the management and monitoring of the network devices. In addition, with SNMP the Internet of Things can be made alive because the IoT devices will be managed, monitored, and their operational data collected using this protocol.

The security issue of sending SNMP over the Internet of Things is of major concern now. SNMPv3, the most recent version of the protocol comes with a lot of security features improvements such as authentication, encryption, and access control that can be utilized for securing the communication between the IoT devices and control systems.

LoRa. LoRa. LoRa. LoRa is the network that is commonly referred to as a low power wide area network (LPWAN) protocol that is best suited for long-distance communication in IoT applications. This means that the communication among devices will be wireless and over long distances but the power consumption will be very low, which makes it the best choice for devices that rely on batteries and for applications that require low data rates and long battery life.

LoRa has a number of security features that it uses to ensure the safety of data transmissions and the authentication of the devices. For example, it allows the use of AES-128 encryption to ensure that the communication between the devices and the gateways is secure.

Sigfox. Sigfox is the low power wide area network (LPWAN) technology that the devices use to connect. Therefore, in this case, the devices are able to transmit tiny amounts of data over long distances while consuming a very low power [6,7].

Sigfox has a star network topology where IoT devices connect to Sigfox base stations directly, and this centralized architecture simplifies network installation and connectivity complexity, as the devices do not directly connect to one another.

RESEARCH RESULTS

The following section provides a comparative study of popular IoT communication protocols in terms of the security-related architectural and cryptographic features. In contrast to typical comparisons that only cover encryption algorithms, the proposed evaluation framework covers authentication and authorization mechanisms, message-level

confidentiality and integrity, key management strategies, network segmentation capabilities, and considerations on availability.

The results are summarised in Tables. The strongest encryption alone cannot guarantee the security of End-to-End transmission; protocols that could provide integrated authentication, dynamic key management, integrity protection, and logical network isolation have higher resistance against the most common IoT attacks, such as impersonation, replay attacks, and denial-of-service attacks. This comparison provides a ground for choosing protocols based on the need for security rather than merely communication efficiency. It is the basis of further discussion in the next section.

TABLE-4. Comparative analysis of security features in IoT communication protocols

Protocol	Auth & Authorisation	Message Encryption	Integrity & Anti-Replay	Key Management	Network Segmentation	Availability
MQTT	Cert / ACL	TLS (AES)	HMAC	Manual / TLS	Broker-based	Medium
CoAP	PSK / Cert	DTLS (AES-CCM)	AEAD	Dynamic	Gateway-based	Medium
DTLS	Mutual Cert	AES-GCM	AEAD	Dynamic	Transport-level	High
Zigbee	Network Key	AES-128	MIC	Centralized	PAN Segmentation	Medium
Z-Wave	Secure Inclusion	AES-128	MAC	Controller-based	Home ID Isolation	Medium

Encryption algorithms play a crucial role in securing the communication protocols used by IoT devices. They help protect data transmitted between IoT devices and provide privacy and integrity.

Table 5. Analysis of IoT protocols

IoT protocol	Authentication	Encryption algorithms	Hash algorithms	OSI layer
MQTT	TLS (X.509 certificates), SASL	AES, RSA, ECC	SHA-256, MD5	Application, Transport, Network
CoAP	DTLS (X.509 certificates), OAuth, Pre-Shared Key (PSK)	AES, RSA, ECC	SHA-256, MD5	Application, Transport, Network
HTTPS	TLS (X.509 certificates), OAuth, API kalitlari	AES, RSA, ECC	SHA-256, MD5	Application, Transport, Network, Physical
DTLS	DTLS (X.509 certificates), PSK (pre-shared key)	AES, RSA, ECC	SHA-256, MD5	Transport, Network, Physical
Zigbee	ECDSA	AES, ECC	SHA-256, MD5	Application, Network
Z-wave	ECDSA	AES	SHA-256, MD5	Application, Network, Physical
LoRaWAN	LoRaWAN MAC-layer security, Application-layer security	AES	SHA-256, MD5	Application, Network, Physical
Ip	ECDH), AES-CCM	AES, ECC	SHA-256, MD5	Application, Network,Physical

Table 6. Security analysis by protocols

Protocols	Basic security mechanisms	Key vulnerabilities/attack vectors
MQTT	TLS (MQTT over TLS), username/password, client certificates, broker access control list, token-based authentication.	If TLS is absent — MITM, interception; incorrect ACL/credential management in the broker.
CoAP	DTLS (transport) or OSCORE (throughput application layer), raw COSE/CBOR.	Overhead costs for DTLS handshakes create a problem when reconnecting; loss of DTLS packets via UDP; risk of incorrect proxy/translation.
Zigbee	AES-128 (network key, connection key, TC), security control center, setup codes, key transport.	Older versions: unencrypted key transfer, default keys; DoS attacks and key extraction attacks.
Z-Wave	S0/S2 security (AES-128), ECDH for S2 key exchange, authenticated enable.	S0 weakness — outdated; attacks on controller implementation.
BACnet	BACnet/SC (TLS/WebSocket + X.509), BACnet authentication extensions	Outdated BACnet over UDP/TCP often uses plain text; building controllers are vulnerable.
KNX	KNX Secure: KNX Data Secure + KNX IP Secure (AES-128, update counters, authentication)	Mixed networks with outdated devices; misconfiguration vulnerabilities.
DNP3	DNP3-SA (secure authentication), AEAD AES-GCM support (newer versions)	Outdated DNP3 has no inherent security; misconfigured request/response; difficulty with broadcast authentication.
MODBUS (RTU/TCP)	NO built-in support; Recommendations: VPN, TLS wrappers, gateways	Well known: lack of default authentication/encryption → MITM, command injection, replay.
NB-IoT / LTE-M	3GPP security (mutual authentication, NAS security, encryption/integrity), SIM/security context	Low-cost devices may skip updates, weak credential lifecycle.
5G	5G-AKA, SUCI (hidden SUPI), higher subscriber privacy, NAS/AS security	Complex architecture; misconfiguration/outdated roaming attacks possible. Plain HTTP, misconfigured TLS, DNS/TCP-based attacks, UDP spoofing.
HTTP / TCP / UDP / IP	TLS (HTTPS) for HTTP; IPsec for network; TCP/UDP without built-in encryption	ABP is less secure (no attachment), countermeasures against reuse, potential key management vulnerabilities.
LoRa / LoRaWAN	OTAA/ABP attachment procedures, NwkSKey/AppSKey (AES-128), MIC	Limited payload, limited end-to-end security; some deployments depend on network security.

CONCLUSIONS

The Internet of Things system is developing today, which in turn leads to increased requirements for its security. This article describes in detail the classification of protocols used in the Internet of Things by their functions. At the same time, the article presents an analysis several security features of the protocols used by the Internet. As a result of the analysis, it is possible to achieve high efficiency by improving the encryption, authentication, and hash algorithms in the existing security protocols.

REFERENCES

1. O. I. Salimbayevich, B. Y. Absamat ugli, S. M. Akmuratovich and K. S. Jaloldin ugli, "Internet of things architecture and security challenges," 2020 International Conference on Information Science and Communications Technologies (ICISCT), Tashkent, Uzbekistan, 2020, pp. 1-4, doi: 10.1109/ICISCT50599.2020.9351495.
2. S. M. Akmuratovich, O. I. Salimboyevich, K. A. Abdusalomovich, T. O. O. Ugli, Y. S. Botirboevna and T. U. Usmonjanovna, "A Creation Cryptographic Protocol for the Division of Mutual Authentication and Session Key,"

2021 International Conference on Information Science and Communications Technologies (ICISCT), Tashkent, Uzbekistan, 2021, pp. 1-6, doi: 10.1109/ICISCT52966.2021.9670057.
3. A. Ayoub, A. Abdallah, M. Ayyash. Security Analysis of Smart Home Protocols: Z-Wave and Thread. IEEE Access (2022) 10: 44321–44335. https://doi.org/10.1109/ACCESS.2022.3167894.
4. M. Humayed, J. Lin, F. Li, B. Luo. Cyber-Physical Systems Security—A Survey. IEEE Internet of Things Journal (2022) 9/4: 2553–2578. https://doi.org/10.1109/JIOT.2021.3074590.
5. R. Langner, P. Smith. Security Vulnerabilities in BACnet and KNX Based Building Automation Systems. Automation in Construction (2022) 140: 104354. https://doi.org/10.1016/j.autcon.2022.104354.
6. Y. Zhang, X. Lin, W. Yu. A Comprehensive Security Analysis of Industrial IoT Communication Protocols. IEEE Communications Surveys & Tutorials (2023) 25/2: 1341–1372. https://doi.org/10.1109/COMST.2023.3241198.
7. A. Farooq, M. S. Khan, N. Javaid. Security Assessment of LPWAN Technologies: LoRa and Sigfox. Sensors (2023) 23/4: 2156. https://doi.org/10.3390/s23042156.
8. S. Alzahrani, K. Salah. Security Evaluation of MQTT and CoAP in Large-Scale IoT Deployments. Computer Networks (2024) 236: 109963. https://doi.org/10.1016/j.comnet.2023.109963.
9. M. Conti, A. Dehghantanha, K. Franke, S. Watson. Internet of Things Security and Forensics: Challenges and Opportunities. Future Generation Computer Systems (2024) 149: 377–391. https://doi.org/10.1016/j.future.2023.09.012.
10. J. Wang, H. Chen, Y. Li. Security and Trust Management in Industrial IoT Protocols. IEEE Transactions on Industrial Informatics (2024) 20/1: 112–124. https://doi.org/10.1109/TII.2023.3298814.
11. H. Nguyen, T. Vo, D. Kim. Security Analysis of Industrial Protocols Modbus and DNP3 in Smart Grid Environments. International Journal of Critical Infrastructure Protection (2025) 38: 100540. https://doi.org/10.1016/j.ijcip.2024.100540.
12. L. Morello, A. Bianchi, R. Carbone. Next-Generation IoT Protocol Security: Challenges and Directions. IEEE Access (2025) 13: 21540–21558. https://doi.org/10.1109/ACCESS.2025.3341127.
13. P. Kasinathan, M. Pastrone, M. A. Spirito. Analysis of MQTT Security for Constrained Internet of Things Devices. Ad Hoc Networks (2023) 139: 102902. https://doi.org/10.1016/j.adhoc.2022.102902
14. E. Sisinni, A. Saifullah, S. Han, U. Jennehag, M. Gidlund. Industrial Internet of Things: Challenges, Opportunities, and Directions. IEEE Transactions on Industrial Informatics (2022) 18/7: 4724–4736. https://doi.org/10.1109/TII.2021.3122597.