GATEWAY FOR SECURE IIoT INTEGRATION IN INDUSTRIAL CONTROL APPLICATIONS

Oana CHENARU[1]

With their increasing popularity in the Smart domain, IoT devices are also penetrating the industrial market under the IIoT paradigm. At the same time, migration of SCADA applications from the centralized approach to the Cloud requires development of security mechanisms adapted to the particular characteristics of such systems. This paper details the requirements, design and implementation of a gateway which uses FIDO2 as secure authentication mechanism for Cloud communication. The gateway uses MQTT and OPC UA interfaces for connecting to IoT devices and local SCADA applications and enables protocol conversion and data aggregation for increasing efficiency in data communication.

Keywords: secure authentication, industrial gateway, IIoT, communication

1. Introduction

Developed initially with the main purpose of providing a unified architecture to connect things able to collect and transmit data, IoT emerged towards a ubiquitous connectivity method, found in various application areas. This increasing popularity revealed important limitations and challenges from the privacy and security points of view, allowing access to sensitive data. Weak protocols, unconscious use of IoT devices and limited security guidelines were identified as most common causes allowing access to malicious applications [1]. Despite the traditional isolated nature of industrial automation and control applications, the need for increased connectivity and access to higher processing resources pushed the adoption of IoT in this domain under the term of IIoT (Industrial IoT) [2]. While there are numerous definitions available for this concept, we consider as most accurate the one proposed in [2]: IIoT is … "A system comprising networked smart objects, cyber-physical assets, associated generic information technologies and optional cloud or edge computing platforms, which enable real-time, intelligent, and autonomous access, collection, analysis, communications, and exchange of process, product and/or service information, within the industrial environment, so as to optimize overall production value". Thus, the particularity of IIoT is that of connecting smart and embedded objects

[1] Lect., Faculty of Automatic Control and Computers, University POLITEHNICA of Bucharest, Romania, e-mail: oana.chenaru@upb.ro

and plant assets to cloud computing platforms with the aim of providing new services able to improve the operational activity and process awareness for the monitored industrial environment. This empowers a wide range of intelligent applications like predictive maintenance, Digital Twin (DT), energy efficiency, lifecycle management, planning to benefit from increased real-time information. Also, new architectures like Cloud SCADA (Supervisory Control And Data Acquisition) are seen as a practical and cost-effective solution to provide flexibility in accessing process real-time data by different teams of the industrial process, enabling better communication and increased operational efficiency [3].

A drawback in wide and fast adoption of IIoT in industrial application was the need to adapt the industrial network architectures which had SCADA servers as the central point for process monitoring and control and the lack of dedicated security measures and policies for vulnerabilities and threats analysis [1], [2]. A major requirement identified in [3] was the integration of old and new technologies, as not even for new applications these new approaches are not ready to replace the traditional architectures.

To provide answers to these limitations current research papers mainly address vulnerabilities and threats identification in IIoT applications and provide some best practices or mitigation measures [3]. The scope of our approach is to provide a practical solution for secure integration of old and new technologies available in industrial applications, facilitating their integration with a Cloud level. For this we define a IIoT gateway implementing a secure authentication mechanism which ensures compatibility with existing industrial applications, while being able to collect data from IoT sensors using MQTT (Message Queuing Telemetry Transport) protocol and to forward them using OPC UA (Open Platform Communications United Architecture).

2. Related work in IIoT Security

As identified in [4] a typical IoT architecture includes three main layers: hardware, network and application. **The hardware layer** is represented by the IoT sensing elements. In the migration towards IIoT field equipment other types of sensors, actuating elements, data acquisition modules or controllers must be integrated in the architecture.

The network layer is responsible for providing the medium and resources for data communication between the sensing elements, the local servers, and the user applications. It is represented by the merge of IoT with industry-specific protocols, providing data access on a per-device basis.

The application layer includes user services for processing and visualization, but its organization as a cloud layer leverages implementation of advanced intelligence, storage and privacy and security compliance services. The intelligence sublayer is specific for each application, implementing desired analytics functions.

Enabling security in IIoT applications must be analyzed at each of these layers, starting from the design phase. Research papers addressing security challenges in such applications usually propose measures from the IT domain applied to SCADA networks: use of firewalls and intrusion detection systems, use of secured communication channels like IP-Sec or SSL/TLS, user based authentication and encryption [3]. To facilitate integration of such approaches to address specific SCADA data-access, reliability, and availability vulnerabilities the authors in [5] propose a quantitative risk modelling architecture to provide a quantitative measure to prioritize such measures. Still, such measures are addressing the industrial system as a whole, while to support the variability and heterogeneity of new IIoT applications each layer should implement specific measures.

From the IIoT perspective, **security at the hardware layer** must address the rules based on which a device can access a network, which can be achieved by applying the confidentiality security measures like encryption, access control, authentication and network isolation [6]. While these concepts were initially used for information security, with the switch towards IIoT their role comes to substitute typical firewall and antivirus solutions which usually require more resources than available in the low-power devices.

Authentication in such applications is a key aspect of security measures as in IIoT this must allow interaction between devices, machines, users and applications while not adding significant overhead in processing or data transfer [6]. Novel authentication schemes were proposed in literature to access IoT devices and a detailed review using hashing, multiple factor protection, context-based identity provisioning or signature provisioning was presented in [7]. In [8] a ticketing model using encryption, digital signature and hashing is presented, applicable in various smart domains. Paper [9] proposes a method for separating access roles to IoT devices by implementing an identity management scheme allowing Single sign-on for temporary access. The limitation of these approaches is the focus only on the authentication and management of different users, which does not respond to IIoT requirements.

Security at the network layer is usually limited by the wide adoption of open protocols in IIoT applications, such as MQTT and OPC UA, which usually use basic security measures and plain-text authentication methods. One solution here is to add security features on the transport layer like TLS (Transport Layer Security) or IPSec (Internet Security) [6]. Another solution is to combine the protocol mechanisms with encryption or access control lists, as shown in [10] for MQTT.

From the **application layer** perspective, security must address the integrity and authenticity of collected data. For this, many of the research papers address anomaly detection and mitigation through Machine Learning or Deep Learning techniques which analyze data accuracy, either individually, or at a contextual level, exploiting the inherent correlations between measured data in industrial applications. A review of such approaches is detailed in [7].

3. Secure gateway design

Security measures for the hardware layer and networks presented in Section 2 are rarely available for typical devices and machinery from the industrial sector, especially as there is no single provider for all the hardware and software components of an application. The presented research papers propose solutions which can address security issues considering new application and devices, but industrial applications are usually characterized by machinery or controllers with proprietary operating systems, where access to interfaces which would be required to implement security as the ones presented above is not possible.

We build our IIoT security-enabling solution starting from the interconnectivity requirements of a typical industrial application as illustrated in Fig 1. where various local sensing and control elements need to send data to a cloud server in a secure way. The local sensing elements and process data are represented by:

a) IoT devices communicating over MQTT one or more measured parameters with a specific timestamp. We consider in this architecture generic IoT devices, thus facilitating their integration in a common data repository.

b) wired sensors communicating over standardized interfaces like 4..20 mA, 0..10V or industrial protocols. A PLC (Programmable logic controller) might be required to enable the physical interfaces required to collect this data or to implement control actions.

We considered in our application also a SCADA server, typically found in industrial applications handling data processing relevant to be analyzed at the cloud level. The cloud is enabling data acquisition, long-term data storage and sending commands to the industrial application. The cloud provides the possibility to connect over an X509 encrypted MQTT protocol.

We identified the following secure connectivity requirements:

- Mechanism for restricting access of local devices and SCADA application to the cloud level, while ensuring data acquisition and commands

- Mechanism for the management of access control of field devices. We addressed processes where the network of elements is not dynamic and they can only be added manually
- Encrypted data transfer to and from the cloud over MQTT
- Secure access of field operators to monitor and manage network devices

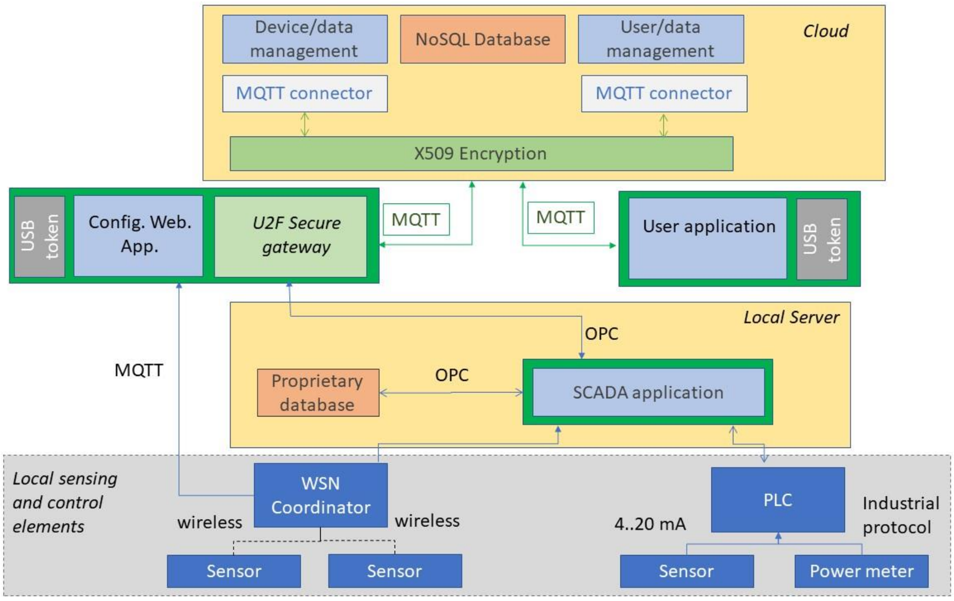


Fig. 1. Use-case architecture

We designed a secure gateway to act as a trusted device between local equipment and machinery, server running SCADA application, and cloud application. The main components of the secure gateway are illustrated in Fig. 2. The aim is to separate the local level of sensing and control devices, from the upper level of user data to increase local security and provide user privacy. We equipped the gateway with OPC UA and MQTT communication interfaces, considered as representative for the industrial control applications and, respectively, for the IoT sector, but the proposed architecture supports future compatibility with a wide range of protocols.

This gateway operates at the hardware layer, implementing a secure authentication mechanism, and can also handle encryption at the network layer for the MQTT protocol through the X509 method provided by the cloud. Access control is achieved by allowing the operator to keep a map of registered devices, to monitor and manage their activity.

In implements an authentication mechanism presented in [11] which we adapted to be compatible to FIDO2 and added the applicability for the IIoT domain. We propose the use of a combination of hardware/software tokens to enable/disable communication between the different layers according to access rights to selected services. The token connects to the cloud platform to evaluate the correspondence between the hardware device and its access rights. The token will enable on-demand communication for the user.

To implement the token mechanism we use a U2F (Universal 2nd Factor) based on FIDO2 (Fast ID Online) library [12] as a provisioning scheme for IoT devices and SCADA applications. We provide a solution to make the U2F device that has been trusted by the IoT cloud in an initial authentication-based registration step, to provision the new IoT devices. All subsequent device settings modification, setting update, and owner transfer can also be performed by using the U2F token that has been trusted to improve security and provide a better user experience.

Three types of actions are possible: configuration actions (OPC and MQTT data setup, IoT device registration, network settings update etc.), smart gateway authentication (linking a smart gateway to a user, enabling access from a cloud server to multiple locations) and operation actions (transmitting data the cloud or receiving data or commands from the cloud and send them to physical devices/local servers).

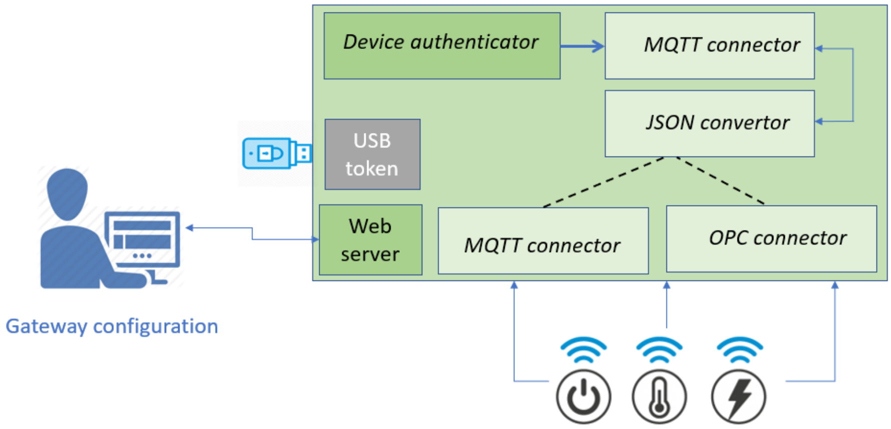


Fig. 2. Secure gateway design

To be able to communicate with local sensing and control elements, the OPC UA tags and MQTT connections will be configured through a locally accessible web-based configuration interface. This ensures local data is kept locally, maintaining the existing local security level (usually consisting in a username and password). This interface also allows FIDO2 registration of U2F token(s) which will be used to connect to their cloud account. The user will be able to manage authenticated gateways and registered tokens, as well as authentication actions associated with personal tokens. During the device configuration stage, the user will initiate an IoT device registration and binding request or configuration update with the help of the secure gateway. This will involve mapping an input MQTT or OPC input interface data to an output MQTT interface.

To ensure user privacy, a detailed procedure was defined and implemented at the cloud level. External access is provided only to data, according to the rights given by a registered token. Data can be organized on owners, and on locations (given by gateways). An owner can have multiple assigned U2F tokens, each token linking one or multiple gateways. We consider each gateway as representing a site or location. Depending on the site type which initiated the authentication, the gateway will have different access rights. Devices could be configured to connect to a restricted area of a cloud user account. They will not have access or any knowledge of the owner data or other information not related with the scope of the application. If multiple such gateways are connected to the same owner account, they can be configured so that all have access to the same data values. If the authentication was initiated by a gateway for an industrial application, a dedicated storage area will be assigned for that specific application (a sub cloud). That storage area will not be linked to a user but to a specific token. This way we can have multiple users from the same owner can be linked to the same application (each with his own token(s)), as well as multiple applications linked to the same user without affecting privacy.

3. Implementation and results

For testing the proposed secure gateway we used a PC running Linux for the FIDO web server, providing the user application and secure authentication mechanism, and a Raspberry PI as the secure gateway. We used a push-button U2F token with FIDO2 library from Feitian as the user device enabling gateway authentication. The secure authentication scrips are based on Python open source FIDO2 libraries from Yubico [12]. We used a private cloud to check the feasibility of enabling MQTT communication. A demo OPC UA client application was used to check communication and data conversion. The access to the cloud is restricted through the use of a token associated to a set of user credentials, username and password.

The message exchange for the different implemented operations is illustrated in Fig. 4 and Fig. 5. From the user interface, authentication is possible either using a set of credentials, or with an already registered token. When a user will first try to authenticate using his cloud credentials, the user interface will check if this data is valid in the cloud. If so, he will be able to access the interface for visualizing the data or to register a new token.

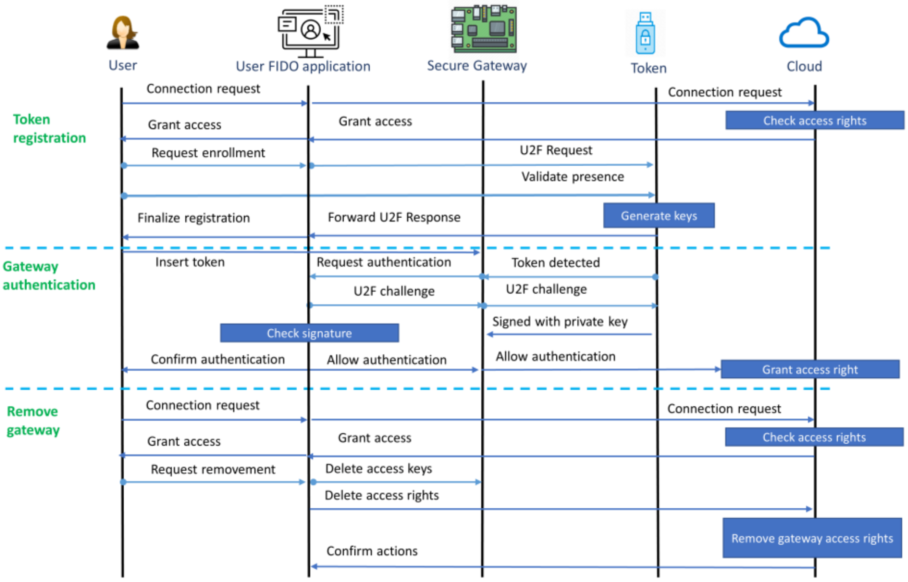


Fig. 4. Message exchange for token registration and gateway authentication

To register a new token, a user needs to initiate this request from the web-based application and will be requested to plug in the U2F token. This will initiate a certification request in the FIDO server. After this, the token LED will start to blink, requesting confirmation for user presence. After the user confirms his presence, a set of public and private keys is generated, and the public key is stored in the user application while the private key will reside on the secure element (token). Fig. 6. shows an example of the certificate generated by the FIDO server. A key handle of the U2F token is be assigned to the user’s account.

To be able to communicate with the cloud, a gateway needs to be authenticated using a secure token. Once such a token is inserted in the gateway it automatically initiates a token-user binding process. The gateway sends an authentication request to the FIDO server providing the user application. The server sends a challenge to the token and the user needs to confirm his presence to accept the authentication by pressing the button of the secure element. This way the token will provide its secure key to the FIDO server. After this step the gateway receives the user’s cloud access information, making it act as a “trusted device” and enabling communication exchange.

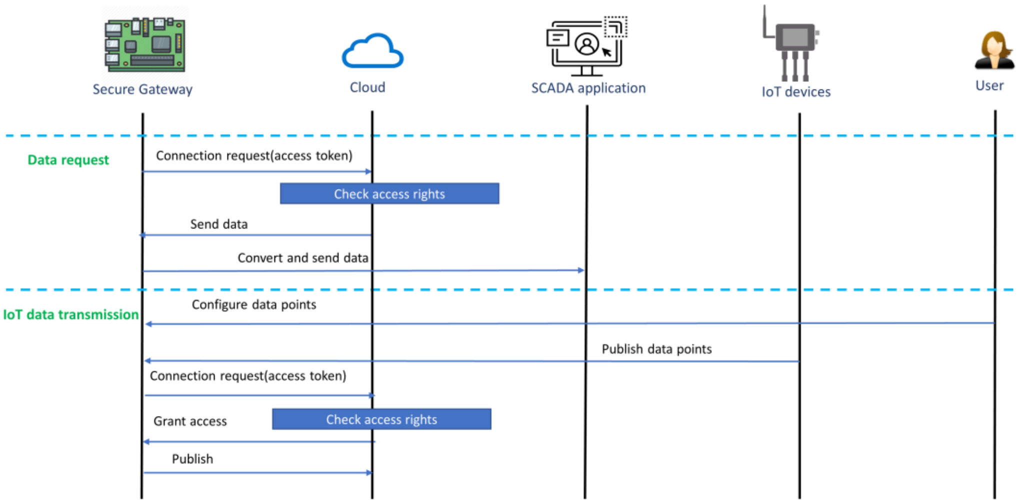
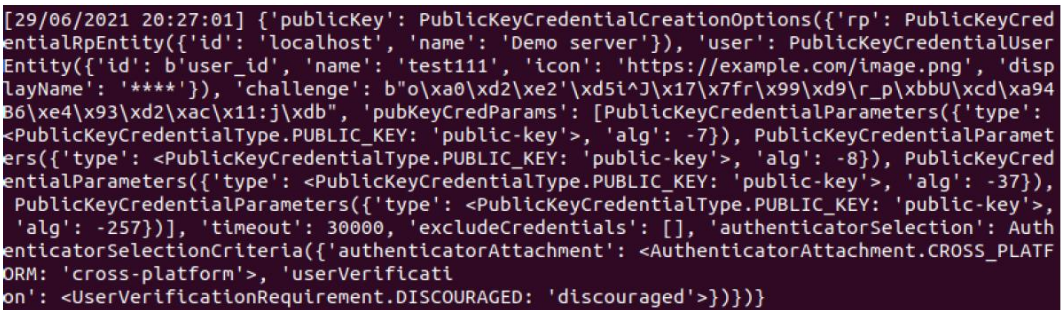


Fig. 5. Message exchange for data send/request (gateway is authenticated)



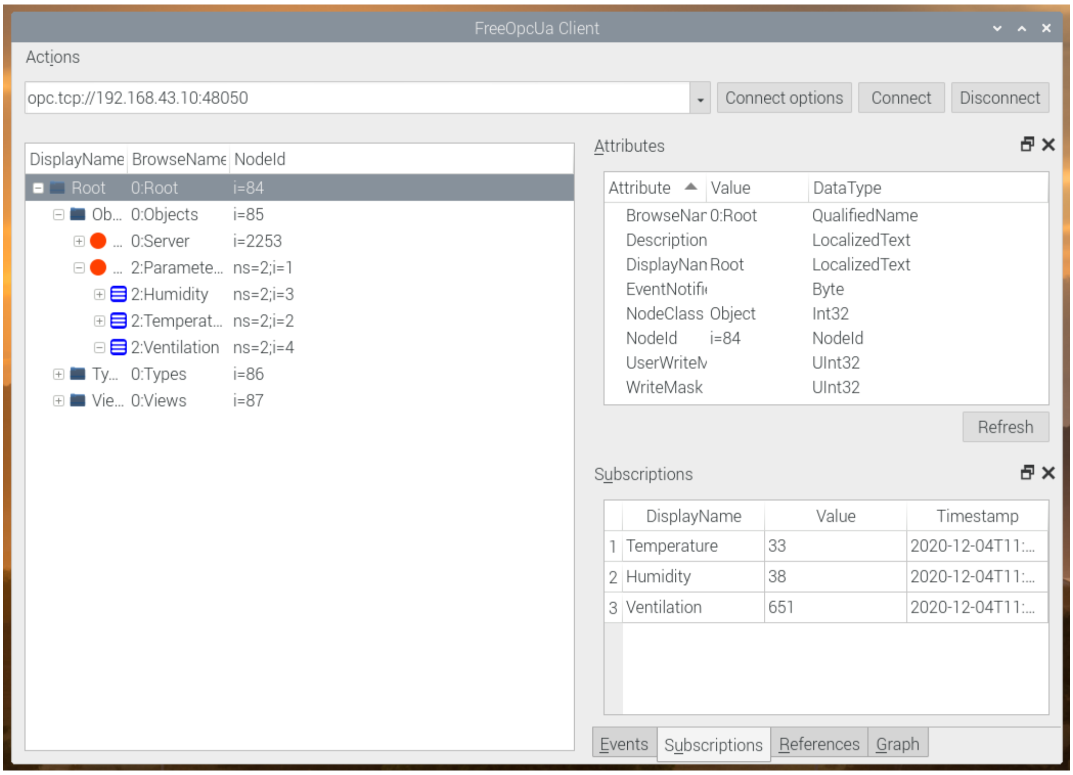
```
[29/06/2021 20:27:01] {'publicKey': PublicKeyCredentialCreationOptions({'rp': PublicKeyCred
entialRpEntity({'id': 'localhost', 'name': 'Demo server'}), 'user': PublicKeyCredentialUser
Entity({'id': b'user_id', 'name': 'test111', 'icon': 'https://example.com/image.png', 'disp
layName': '****'}), 'challenge': b"o\xa0\xd2\xe2'\xd5i^J\x17\x7fr\x99\xd9\r_p\xbbU\xcd\xa94
B6\xe4\x93\xd2\xac\x11:j\xdb", 'pubKeyCredParams': [PublicKeyCredentialParameters({'type':
<PublicKeyCredentialType.PUBLIC_KEY: 'public-key'>, 'alg': -7}), PublicKeyCredentialParamet
ers({'type': <PublicKeyCredentialType.PUBLIC_KEY: 'public-key'>, 'alg': -8}), PublicKeyCred
entialParameters({'type': <PublicKeyCredentialType.PUBLIC_KEY: 'public-key'>, 'alg': -37}),
 PublicKeyCredentialParameters({'type': <PublicKeyCredentialType.PUBLIC_KEY: 'public-key'>,
 'alg': -257})], 'timeout': 30000, 'excludeCredentials': [], 'authenticatorSelection': Auth
enticatorSelectionCriteria({'authenticatorAttachment': <AuthenticatorAttachment.CROSS_PLATF
ORM: 'cross-platform'>, 'userVerificati
on': <UserVerificationRequirement.DISCOURAGED: 'discouraged'>})})}
```

Fig. 6. Example of security certificate generated by the FIDO server in the registration step

Once an IoT device was authenticated and cloud access details were received, it starts to transmit data continuously without further user or secure gateway actions (the link between the input and output interfaces will remain active). This link is visible in the user application and can be disabled by the user from the device management section. The same way, an authenticated gateway is able to send commands to physical devices, by mapping the tags of cloud data (MQTT topics) to local OPC tags (data addresses in a local OPC server). To enable these two operating actions a script converts from the data format of local devices to the JSON data format characteristic for MQTT communication.

For the implementation of the MQTT and OPC UA interfaces publicly available resources were used. The MQTT interface was configured by defining the cloud server URL, and the authentication mechanism consisting in username and password. The conversion between OPC UA and MQTT is achieve through mapping of the corresponding variables. MQTT evaluation was done using open MQTT-Spy tool.

Evaluation of the OPC UA interface was done using a demo client application. As illustrated in Fig. 7, it was able to connect to process parameters for monitoring and control. The monitored parameters were then sent over MQTT to the cloud server (Fig. 8)and no accuracy or integrity problems were identified.



opc.tcp://192.168.43.10:48050

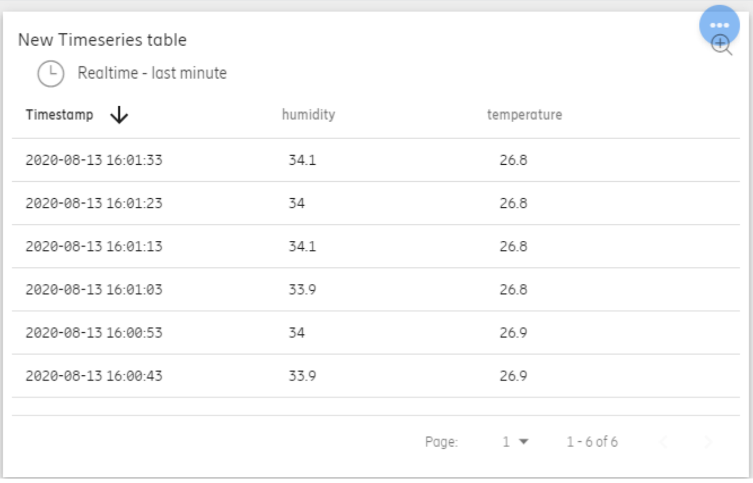
DisplayName	BrowseName	NodeId
Root	0:Root	i=84
Ob...	0:Objects	i=85
...	0:Server	i=2253
...	2:Paramete...	ns=2;i=1
	2:Humidity	ns=2;i=3
	2:Temperat...	ns=2;i=2
	2:Ventilation	ns=2;i=4
Ty...	0:Types	i=86
Vie...	0:Views	i=87

Attribute	Value	DataType
BrowseNar	0:Root	QualifiedName
Description		LocalizedText
DisplayNan	Root	LocalizedText
EventNotifi		Byte
NodeClass	Object	Int32
NodeId	i=84	NodeId
UserWriteN		UInt32
WriteMask		UInt32

	DisplayName	Value	Timestamp
1	Temperature	33	2020-12-04T11:...
2	Humidity	38	2020-12-04T11:...
3	Ventilation	651	2020-12-04T11:...

Fig. 7. OPC-UA interface for data visualization

The experimental setup allowed us to evaluate all functionalities of the secure gateway including access control restriction, secure authentication using U2F for gateway configuration and management, functionality of the user interface, gateway to cloud connection and data acquisition and conversion from OPC UA to MQTT. The flexibility in the design will allow the use of such an approach in various applications and adding additional tools to increase security according also to available cloud mechanisms. The novelty of this solution is represented by the use of an authentication mechanism based on U2F to allow devices and applications from the industrial domain to connect securely oven an encrypted link to a cloud server. The proposed secure gateway addresses the practical limitations and requirements of the industrial sector regarding available interfaces and processing functions available, interconnectivity aspects and flexibility needs.



New Timeseries table

Realtime - last minute

Timestamp ↓	humidity	temperature
2020-08-13 16:01:33	34.1	26.8
2020-08-13 16:01:23	34	26.8
2020-08-13 16:01:13	34.1	26.8
2020-08-13 16:01:03	33.9	26.8
2020-08-13 16:00:53	34	26.9
2020-08-13 16:00:43	33.9	26.9

Page: 1 ▾ 1 - 6 of 6

Fig. 8. Cloud visualization of real-time data

6. Conclusions

This paper addressed the security and privacy of IIoT applications from the secure authentication perspective, considering the functional requirements of traditional SCADA systems. The presented steps showed how a secure authentication mechanism based on FIDO2 can be used to enable device authentication, to allow its connection to a remote or cloud server. This approach is an extension of typical secure authentication mechanisms, available only to identify user access, to respond to the secure communication requirements of industrial control applications.

Future steps will involve updating the cloud and gateway applications by using X509 certificates for encryption of data from MQTT devices. It is a digital certificate that uses the widely accepted international X. 509 public key infrastructure (PKI) standard to verify that a public key belongs to the user, computer or service identity contained within the certificate. Also, a setup will be configured to simulate various security breaches and evaluate the vulnerabilities of the proposed solution.

Acknowledgment

The work has been partially funded by the Operational Programme Human Capital of the Ministry of European Funds through the Financial Agreement 51675/09.07.2019, SMIS code 125125, and by UEFISCDI Romania through

ITEA3 programme project PARFAIT - Personal dAta pRotection FrAmework for IoT, project id: PN-III-P3-3.5-EUK-2017-02-0063.

REFERENCES

[1] *L. Tawalbeh, F. Muheidat, M. Tawalbeh, M. Quwaider,* “IoT Privacy and Security: Challenges and Solutions”, in Applied Sciences. 2020, 10, 4102.

[2] *H. Boyes, B. Hallaq, J. Cunningham, T. Watson*, “The industrial internet of things (IIoT): An analysis framework”, in Computers in Industry, vol. 101, pp.1-12, 2018.

[3] *A. Sajid, H. Abbas, K. Saleem*, "Cloud-Assisted IoT-Based SCADA Systems Security: A Review of the State of the Art and Future Challenges," in IEEE Access, vol. 4, pp. 1375-1384, 2016.

[4] *P. Sethi, S. Sarangi*, "Internet of Things: Architectures, Protocols, and Applications", Journal of Electrical and Computer Engineering, vol. 2017, Article ID 9324035, 25 pages, 2017.

[5] *G. Falco, C. Caldera, H. Shrobe,* "*IIoT Cybersecurity Risk Modeling for SCADA Systems*," in IEEE Internet of Things Journal, vol. 5, no. 6, pp. 4486-4495, Dec. 2018.

[6] *K. Tange, M. De Donno, X. Fafoutis, N. Dragoni*, "A Systematic Survey of Industrial Internet of Things Security: Requirements and Fog Computing Opportunities," in IEEE Communications Surveys & Tutorials, vol. 22, no. 4, pp. 2489-2520, Fourthquarter 2020, doi: 10.1109/COMST.2020.3011208.

[7] *P. Jayalaxmi, R. Saha, G. Kumar, N. Kumar T. -H. Kim*, "A Taxonomy of Security Issues in Industrial Internet-of-Things: Scoping Review for Existing Solutions, Future Implications, and Research Challenges," in IEEE Access, vol. 9, pp. 25344-25359, 2021, doi: 10.1109/ACCESS.2021.3057766.

[8] *A. Alhothaily, C. Hu, A. Alrawais, T. Song, X. Cheng, D. Chen,* "A Secure and Practical Authentication Scheme Using Personal Devices," in IEEE Access, vol. 5, pp. 11677-11687, 2017.

[9] *A. Witkovski, A. Santin, V. Abreu, J. Marynowski*, "An IdM and Key-Based Authentication Method for Providing Single Sign-On in IoT," 2015 IEEE Global Communications Conference (GLOBECOM), San Diego, CA, 2015, pp. 1-6, doi: 10.1109/GLOCOM.2015.7417597.

[10] *S. Katsikeas* et al., "Lightweight & secure industrial IoT communications via the MQ telemetry transport protocol," 2017 IEEE Symposium on Computers and Communications (ISCC), pp. 1193-1200, 2017, doi: 10.1109/ISCC.2017.8024687.

[11] W. Kang, “U2Fi: A Provisioning Scheme of IoT Devices with Universal Cryptographic Tokens”, arXiv:1906.06009, 2019.

[12] FIDO U2F, accessible online: https://www.yubico.com/authentication-standards/fido-u2f/ (last access nov 2020)