IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 3, 2021, accepted June 22, 2021, date of publication June 24, 2021, date of current version July 6, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3092203

How to Survive Identity Management in the Industry 4.0 Era

JASONE ASTORGA[1], MARC BARCELO[2], AITOR URBIETA[2], AND EDUARDO JACOB[1], (Senior Member, IEEE)

[1]Department of Communications Engineering, University of the Basque Country (UPV/EHU), 48013 Bilbao, Spain
[2]Ikerlan Technology Research Centre, Basque Research and Technology Alliance (BRTA), 20500 Arrasate/Mondragón, Spain

Corresponding author: Jasone Astorga (jasone.astorga@ehu.eus)

This work was supported in part by the Spanish Ministry of Science and Innovation through the National Towards zeRo toUch nEtwork and services for beyond 5G (TRUE-5G) Project under Grant PID2019-108713RB-C53, in part by the European Commission through the Electronic Components and Systems for European Leadership-Joint Undertaking (ECSEL-JU) 2018 Program under the framework of key enabling technologies for safe and autonomous drones' applications (COMP4DRONES) Project under Grant 826610, with the national financing from France, Spain, Italy, The Netherlands, Austria, Czech, Belgium, and Latvia, in part by the Ayudas Cervera para Centros Tecnológicos Grant of the Spanish Centre for the Development of Industrial Technology (CDTI) through the Project EGIDA under Grant CER-20191012, and in part by the Basque Country Government through the Creating Trust in the Industrial Digital Transformation (TRUSTIND) ELKARTEK Program Project under Grant KK-2020/00054.

This work did not involve human subjects or animals in its research.

ABSTRACT Industry 4.0 heavily builds on massive deployment of Industrial Internet of Things (IIoT) devices to monitor every aspect of the manufacturing processes. Since the data gathered by these devices impact the output of critical processes, identity management and communications security are critical aspects, which commonly rely on the deployment of X.509 certificates. Nevertheless, the provisioning and management of individual certificates for a high number of IIoT devices involves important challenges. In this paper, we present a solution to improve the management of digital certificates in IIoT environments, which relies on partially delegating the certificate enrolment process to an edge server. However, in order to preserve end-to-end security, private keys are never delegated. Additionally, for the protection of the communications between the edge server and the IIoT devices, an approach based on Identity Based Cryptography is deployed. The proposed solution considers also the issuance of very short-lived certificates, which reduces the risk of using expired or compromised certificates, and avoids the necessity of implementing performance expensive protocols such as Online Certificate Status Protocol (OCSP). The proposed solution has been successfully tested as an efficient identity management solution for IIoT environments in a real industrial environment.

INDEX TERMS Automatic enrolment, IIoT, Industry 4.0, SCEP, X.509.

I. INTRODUCTION

The next industrial revolution is known as Industry 4.0 and mainly seeks to enhance the efficiency, security and reliability of manufacturing processes, resulting in higher quality products, greater traceability and reduction of manufacturing costs. The main idea to materialize these goals is to replace slow, costly and error-prone manual procedures with reliable and efficient computer-based automatic systems. In order to achieve this aim, massive data gathering supported by Industrial Internet of Things (IIoT) devices and networks plays a key role. These data could then be used to feed specific machine learning, automation and orchestration algorithms targeted at improving efficiency, predicting failures, etc. This is graphically represented in Figure 1, where all manufacturing processes in the outermost circle are monitored with IIoT devices. Then, the data analysis and decision-making processes in the middle circle make use of the data gathered by the IIoT devices. And in the centre of it all stands identity management, as an essential ingredient to make the whole system trustworthy.

In such a context, where the information gathered by IIoT devices commands production level decisions, strong security is a must [1], [2]. This includes reliable authentication and access control mechanisms, as well as data encryption. In order to implement such security services, two main approaches are possible: (1) to distribute shared secret keys and use symmetric cryptographic algorithms; or (2) solutions

The associate editor coordinating the review of this manuscript and approving it for publication was Congduan Li.

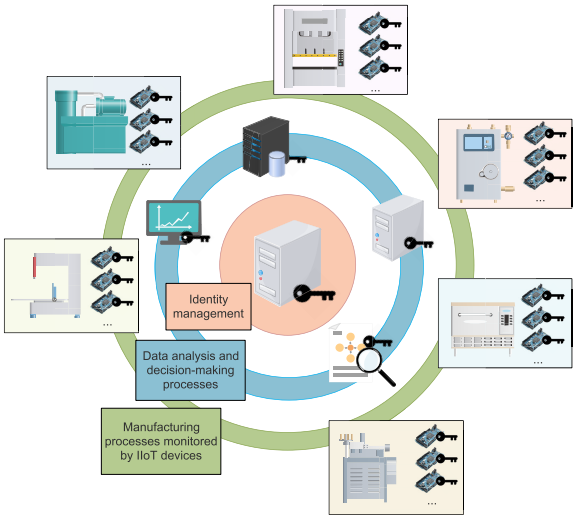


FIGURE 1. Industry 4.0 manufacturing processes based on data gathered by IIoT devices.

based on Public Key Infrastructures (PKIs) and digital certificates, which rely on public key cryptographic mechanisms. The distribution of shared secret keys implies scalability issues, specially when all the involved parties do not belong to the same organization. For this reason, PKI-based alternatives are preferred to implement strong security mechanisms in open environments. However, the utilization of PKIs implies also important challenges, specially when massively deployed resource-deprived devices are involved, as in the case of IIoT-based industrial scenarios.

On the one hand, the implementation and execution of public key cryptographic algorithms requires extensive use of CPU and memory; resources that are only limitedly available in most IIoT devices [3], [4]. For the majority of IIoT devices, such as Programmable Logic Controllers (PLCs), this results in long computation times of public key cryptographic operations. On the other hand, apart from the performance-related problems inherent to the hardware characteristics of IIoT devices, another important challenge is related to their massive deployment [5], [6], as they are extensively used to gather data of all the possible aspects of manufacturing processes. When relying on PKI-based security mechanisms, each individual entity and service, including IIoT devices, must own a digital certificate that reliably links its identity with its corresponding public key. The management of thousands of digital certificates implies an important organizational challenge for current industrial plants, specially when most IIoT devices do not have a screen and keyboard to make certificate request easier; or a clock with the actual time and date, to aid in the certificate renewal process.

Another issue related to the use of digital certificates is associated with certificate revocation and checking. In fact, existing mechanisms to protect from compromised certificates such as the use of Certificate Revocation Lists (CRLs) or the Online Certificate Status Protocol (OCSP) have demonstrated to be highly resource-expensive and of limited effectiveness [7]. Such problems are even more troublesome in IIoT environments, where networks are more dynamic, with devices entering or leaving the network more frequently, and where these little devices are more exposed to security attacks. As a consequence, the size of CRLs grows and their management, update and query become important challenges. For all these reasons, most novel approaches are promoting the issuance of ever more short-lived certificates [8], [9]. The use of short-lived certificates eliminates the need of managing and checking revoked certificates and enhances the overall network security, by minimizing the possibility of accepting compromised certificates before they expire. However, short-lived certificates present also performance and organizational-level challenges. In contexts where a high number of certificates must be issued every few days or hours, certificate (re)enrolment processes might become a bottleneck [10].

Therefore, the main contribution of this paper is a reliable and efficient identity management system for IIoT devices. The main goal of this system is to simplify the identity management for IIoT devices, moving the complexity associated to this process from the resource-deprived IIoT devices to a more resource-rich edge server. More specifically, the proposed solution relies on issuing X.509 digital certificates, which allow to implement strong security mechanisms both within the organization and also with entities outside it. Additionally, the proposed solution considers the issuance of certificates with very short lifespans. In fact, short-lived certificates improve the overall network security, as they reduce the possibility of having compromised and not revoked certificates. In order to enhance the performance and scalability of IIoT identity management, the proposed approach delegates the certificate lifecycle management to a resource-rich edge server known as the Certificate Lifecycle Management (CLM) server, which is located in the boundary between the IIoT network and the corporate data network. The CLM keeps an always up-to-date and consistent inventory of organizational-level digital certificates, which improves the security management of the organization, and facilitates data searching, report generation, etc. This enhances the overall security level of the whole organization, as a good planning and management of security procedures and cryptographic material is one of the most important pillars of network security.

One important aspect of the proposed solution is that although certificate management procedures are delegated to the CLM, the private keys of the IIoT devices never leave these devices. That is, private keys are only known by the owner IIoT device and never communicated to the CLM. This allows the CLM to be a just partially trusted entity and avoids the possibility of the CLM impersonating any of the IIoT devices.

On the other hand, taking into account that the proposed solution targets industrial scenarios, the use of widely tested mechanisms and standard compliance become essential

factors. For this reason, we have designed our solution fully based on established standards and security infrastructures, facilitating its easy and quick adoption. More specifically, the communications between the IIoT devices and the CLM are based on the Constrained Application Protocol (CoAP) [11] and Raw Public Key (RPK)-Datagram Transport Layer Security (DTLS) [12], [13]. To make the distribution and verification of RPKs more efficient, an Identity Based Cryptography (IBC) [14] solution is proposed, where the public key of each entity is directly derived from its identity. On the other hand, the enrolment and renewal of digital certificates is carried out by means of the Simple Certificate Enrollment Protocol (SCEP) [15], which is currently the most widely supported protocol for automatic enrolment and renewal of digital certificates.

In order to guarantee the viability of the proposed solution, a benchmarking prototype has been deployed in a real industrial environment, specifically in the Aeronautics Advanced Manufacturing Center (CFAA) [16]. The CFAA was launched by the University of the Basque Country along with a wide range of companies involved in manufacturing and Industry 4.0 and it is financially supported by the local and autonomous government. The final aim of the CFAA is to raise the industrial competitiveness of the region by promoting research and experimentation in a real environment with the most novel machinery and manufacturing techniques. Therefore, its R+D+I activity is focused on TRLs 5 to 7, allowing fast transfer to the industrial fabric. The obtained results show that the proposed system is suitable for the targeted environments both from the functional and performance points of view.

Therefore, the main contributions of this paper could be summarized as follows:

- A novel, efficient and scalable identity management system for IIoT environments based on X.509 digital certificates, including the necessary procedures and messages.
- A real-world implementation of the proposed system in an industrial environment that acknowledges the feasibility and suitability of the solution.

The rest of the paper is structured as follows. Section II summarizes the most important literature related to the researched topic. Then, Section III reviews the most important concepts of the technologies upon which the proposed system is built, namely Identity Based Cryptography and automatic enrolment protocols. Section IV defines the general architecture of the proposed system, including all the involved entities, while Section V details the procedures and message exchanges designed for its operation. Next, Section VI considers the security issues that might affect the proposed system and how they are avoided. Section VII details the testbed implemented and the measures carried out to evaluate the performance of the proposed system and assess the factors that have most impact on it. Finally, Section VIII gathers the main conclusions of the work and future research lines.

II. RELATED WORK

The introduction of PKIs in IoT scenarios presents important challenges, mainly related to the resource limitations of the involved devices and the typically very high number of devices involved. Apart from the obvious difficulties related to the execution of asymmetric encryption algorithms in devices with severe limitations regarding memory and CPU, the lifecycle management of the X.509 certificates used to distribute public keys in a trustworthy manner presents also an important challenge, mainly due to scalability reasons [10], [17]. In the same way, downloading CRLs or executing the OCSP protocol to check the revocation status of each received certificate are also costly tasks for IoT devices [18], [19].

The issues related to the use of PKIs by resource-limited IoT devices have been extensively studied in the literature [20]–[22], and many solutions have been proposed, mainly based on the delegation of the most consuming tasks to a resource-rich and trusted centralized entity [23]–[26]. However, the issues related to the lifecycle management of certificates in IoT environments have received little attention so far. In this section, we review the existing proposals to make the lifecycle management of certificates lighter for IoT environments.

The authors in [10] propose PKI4IoT, a contribution to make IoT devices compatible with current PKIs. For this purpose, the paper proposes a lightweight profile for the compression and encoding of X.509 certificates and an automatic and light enrolment protocol, suitable for massively deployed resource-deprived devices. The proposal assumes that IoT devices are preloaded when manufactured with a factory certificate of the device, the corresponding private key and at least one trusted Certificate Authority (CA) certificate. However, this proposal is based on tailor-made protocols, avoiding the use of standard mechanisms, which hinders its easy adoption by industry.

Taking into account the difficulty of certificate lifecycle management in IoT scenarios, the authors in [27] propose TwinPeaks, an alternative to current PKIs built on Certificateless Public Key Cryptography (CL-PKC). In short, in order to remove the need for PKIs and digital certificates, the proposal is based on using public identities which are directly linked to public keys. More specifically, public keys are generated taking as input the corresponding public identities, such as a fully qualified domain name, IP address, etc. One important drawback of this approach is that both public and private keys are generated by a centralized server, known as the Private Key Generator (PKG). This Trusted Third Party (TTP) gets to know the private keys of all the entities it serves, which results in the so-called key escrow problem. Another drawback is that for its operation, the solution requires also the implementation of a new key server hierarchy and it defines new specific messages for querying these key servers. In fact, when a source A wants to obtain the public key of a destination B, first it uses B's identity to obtain its domain

name, then it queries the DNS for the IP address of the key server of B's domain and finally, it contacts the corresponding key server in order to obtain B's public key.

On the other hand, in order to make the enrolment process lighter for IoT devices, the work in [28] proposes to replace long passphrases with short pre-shared keys or PINs in the security mechanisms of the SCEP message exchange. These short PINs will then feed a Password Authenticated Key Exchange (PAKE) protocol, which in turn will derive a one-time-secret to authenticate the actual certificate enrolment process. As the proposed security mechanism is designed like a wrapper around the standard SCEP protocol, it can be easily implemented and deployed.

The paper in [29] presents the design, implementation and evaluation of an automatic enrolment mechanism for IoT devices. The proposed mechanism is mainly based on the implementation of EST over CoAP/DTLS. First a PSK based DTLS channel is established between the IoT device and the CA. Over this secure channel, the EST PKCS #10 and PKCS #7 request/response messages are sent. The CA verifies the request automatically, and the obtained certificate is then used to establish certificate-based DTLS sessions between the IoT devices and Internet hosts. The proposed mechanism has been developed for the Contiki operating system and successfully implemented in a device with a 32 kB RAM and about 200 kB ROM. Therefore, this work mainly addresses the performance issues of automatic enrolment processes linked to the hardware limitations of IoT devices, but does not consider management issues when a high number of IoT devices are involved.

Focused on the problem of checking the revocation status of each certificate, the work in [30] deals with the issue of long CRLs, which exceed the available memory in IoT devices. The authors defend the use of CRLs for checking certificate revocation as a better option than using OCSP for 3 reasons: (1) OCSP requires the IoT device being online; (2) if the number of OCSP queries becomes very large, the OCSP responder might become overwhelmed; and (3) OCSP queries leave a trace of the IoT devices activity, which risks its privacy. The authors propose two alternative protocols to the conventional CRL protocol, which allow to greatly reduce the amount of downloaded information, reducing in this way, the used RAM and bandwidth. The first option is based on a generalized Merkle hash tree and the second option is based on a Bloom filter. Both protocols are probably secure and allow parametrization to personalize the trade-off between efficiency and security under various conditions.

Similarly, the work in [18] presents a solution to the problem of having to transmit and store very big CRLs, which is especially problematic in the case of resource-deprived IoT devices. The proposed approach heavily builds on a data structure called Distributed Hash Table (DHT), which allows quick lookups when data is distributed among multiple devices. The proposal is specifically tailored to smart meter networks. According to the proposed solution, the CRL is split into N portions, being N the number of smart meters in the network, and each smart meter only stores a part of the whole CRL. In order to distribute the revoked certificates' identifiers among the N portions, first, each smart meter is identified by a hash of its IP address and its public key and a hash of each revoked certificate is also computed. Then, the hash of each certificate is compared with the hash of the smart meters. For this purpose, a special hash function is used, known as Consistent Hash Function, which ensures uniform distribution of key and value pairs on an imaginary ring. The network gateway acts as the distributor of CRL portions. Therefore, each smart meter just stores one portion of the CRL, and a finger table to find other CRL portions when they are required. To guarantee the security of the CRL portion distribution process, these portions must be signed with the gateway's private key.

III. BACKGROUND TECHNOLOGIES

This section deals with the main technologies that act as foundations for the proposed solution, namely Identity-Based Cryptography and automatic certificate enrolment.

A. IDENTITY-BASED CRYPTOGRAPHY

Identity-Based Cryptography (IBC) is a particular type of asymmetric cryptographic schema where public-private key pairs are not randomly generated. Instead, each entity's public key is directly derived from its identity, such as email address, social security number, etc. This asymmetric cryptographic schema was first proposed by Shamir in 1984 [31] and its main goal is to avoid the necessity of issuing and signing digital certificates to reliably bind an identity to a public key; and therefore, also the necessity of building and maintaining CA hierarchies. That is, in IBC anyone can send a confidential message to a specific destination without needing to download and check digital certificates, just by using its identity as public key. However, it was not until 17 years after Shamir published the IBC concept that the first fully-functional IBC implementation was developed by Boneh and Frankil [32].

For its operation, IBC requires the existence of a trusted third party acting as a PKG, which is able to compute the private key corresponding to each public key. For this aim, the PKG holds a Master Public Key and a Master Secret Key and it implements specific elliptic curves based on bilinear pairings, such as the widely used Weil pairing [33].

IBC gathers the concepts of Identity-based Encryption (IBE) and Identity-based Signature (IBS) and different international standardization organizations such as IETF, IEEE and ISO/IEC have standardized several IBE and IBS algorithms [34]–[36].

B. AUTOMATIC ENROLMENT MECHANISMS

Certificate enrolment is the process by which a user (human, software, etc.) obtains a valid certificate from a CA. This process is initiated by a user request including the user's public key, a proof of knowledge of the associated private key and other enrolment information. When a CA receives such a

request, if verifies the conveyed information and compares it to its established policy rules. If the verification is successful, the CA creates the requested certificate, posts it and sends an identifying certificate to the user. Certificate enrolment might be directly performed with a CA or through a Registration Authority (RA).

Currently, a number of automatic enrolment protocols exist, but without a doubt, the most widely deployed one is the Simple Certificate Enrollment Protocol (SCEP) [15]. This protocol, originally defined by Cisco, allows a device to easily obtain a digital certificate from a trusted CA by using a URL and a secret shared with the CA, which implies that users must be pre-registered with the CA (share secret keys). Then, PKI services can be configured to automatically accept all certificate requests or to send some of them for approval by an administrator. Although it is the most widely used automatic enrolment protocol, it is not a standard, but an IETF draft. The operations supported by SCEP include certificate enrolment, certificate renewal, certificate queries, CRL queries and distribution of CA public keys.

As an evolution of SCEP, the Enrolment over Secure Transport (EST) [37] protocol was defined by Cisco, Akayla, and Aruba Networks and standardized in 2013. The main difference with respect to SCEP is that instead of using shared secrets to guarantee the security of the communications between clients and the CA, EST implements standard TLS as the transport security layer, which implies the necessity of distributing and managing digital certificates. Another difference of EST with respect to SCEP is that it supports server-side key generation, which is important in the case of IoT devices with not enough power or entropy source to generate a random private key.

Another alternative for automatic certificate enrolment is Certificate Management over CMS (CMC) [38]. This protocol is very similar to SCEP, but it implements a wider range of certificate management operations that go beyond the certificate provisioning of SCEP and EST, such as certificate revocation, status, batch requests, etc. The specification defines a message format, message control and data structures. In the same way as CMC, Certificate Management Protocol (CMP) [39] also goes beyond certificate provisioning and it also defines its own message format. The transport mechanism for CMP is HTTP.

On the other hand, Automated Certificate Management Environment (ACME) [40] is a communication protocol designed for automating the issuance and domain validation procedures. Designed by Internet Security Research Group (ISRG) [41] for their Let's Encrypt service, ACME allows to set up an HTTPS server and make it automatically obtain a browser-trusted certificate without any human intervention. The communication between the client and the server is based on JSON messages over HTTPS.

Finally, Enrolment with Application Layer Security (EALS) [42] defines a certificate enrolment protocol specifically tailored to IoT constrained devices. For this reason, it runs over CoAP and the used data format is CBOR. Specifically, EALS defines a certificate enrolment mechanism based on CMC messages and secured at the application layer by means of OSCORE [43]. For the OSCORE implementation, it requires that the communicating entities share a secret key, known as the ''master secret''. For the establishment of this pre-shared key, the use of EDHOC [44] is proposed.

IV. GENERAL ARCHITECTURE

The aim of this section is to describe the proposed edge-based certificate management architecture for IIoT scenarios. The proposed architecture with the modules loaded in each entity is graphically depicted in Figure 2. A key element of this architecture is the Certificate Lifecycle Management (CLM) server, located in the edge between the IIoT network and the traditional corporate data communication network. This is a most suitable position for the CLM to perform certificate enrolment and renewal on behalf of the resource deprived IIoT devices. Then, specific modules must also be loaded in the endpoint IIoT devices as well as in the corporate CA. Additionally, a PKG is also introduced to support the implementation of secure communications based on IBC. Given the criticality of the services provided by this entity and the information it stores, it is envisioned that this PKG is not connected to the corporate network and it operates offline.

Next, the features and functions of the involved entities are detailed.

A. CERTIFICATE LIFECYCLE MANAGEMENT (CLM) SERVER

The CLM is the key element of the proposed security management system and its overall goal is to manage the certificates of all IIoT devices within the corporate network in a secure and well-structured way. For this aim, some certificate management functions are delegated from the endpoint IIoT devices to the CLM. However, an important feature is that IIoT devices' private keys are never delegated to the CLM and they never leave the owner IIoT device. This is an essential point in order to preserve the end-to-end security of the communications protected afterwards by those private keys.

The CLM stores and manages a consistent inventory with information about all IIoT devices registered in the corporate network, including device identity, IBC public key, digital certificate, expiration date, and a flag indicating whether the IIoT device is operational or not. When a IIoT device is tagged as ''operational,'' it means that it has already obtained a valid digital certificate from the corporate CA; in an opposite case, it is tagged as ''not operational''. In this way, the overall security architecture of the organization is improved as the certificate inventory is unique, consistent and always up-to-date; and the CLM, being a resource-rich device, is in better position to manage the organization's certificates' lifecycle: check renewal dates, start re-enrolment process when needed, store certificates, etc.

Communications between the CLM and the endpoint IIoT devices are based on CoAP and specifically, the CLM implements two CoAP clients: one to request a CSR to

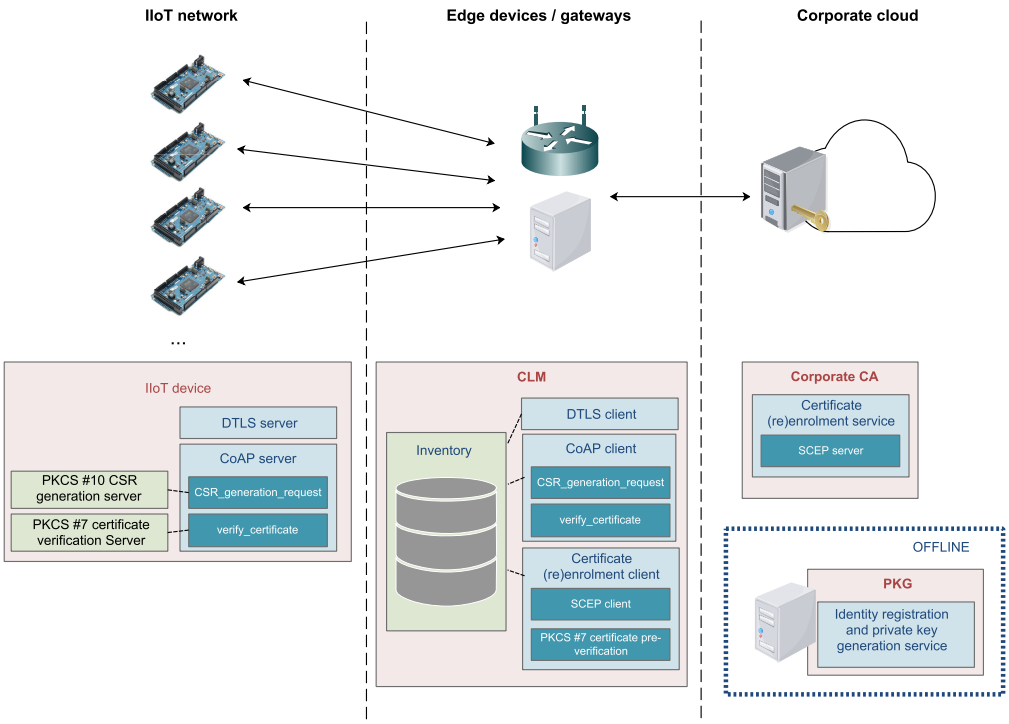


FIGURE 2. General architecture of the proposed system.

be generated and signed by the corresponding IIoT device (*CSR_generation_request* client) and a second one to provide the IIoT device with the CA signed certificate for its verification and storage (*verify_certificate* client). Additionally, these CoAP communications between the CLM and the IIoT devices must be appropriately protected. For this aim, a RPK-based DTLS channel is established, in which the CLM acts as DTLS client, and therefore, it starts the DTLS handshake against the DTLS servers running on the endpoint IIoT devices. The used RPKs are based on IBC, securely linking in this way each public key with its corresponding identity and removing the necessity for initial certificate distribution at the commissioning phase.

The CLM must also communicate with the corporate CA for the automatic (re)enrolment of digital certificates on behalf of endpoint IIoT devices. For this aim, a solution based on the SCEP protocol has been designed. Specifically, the CLM implements a SCEP client which generates SCEP requests including the CSRs previously generated by the endpoint IIoT devices. Once the CA responds with the SCEP response including the generated and signed X.509 certificate, the CLM performs a first verification of the provided certificate, which has been called the pre-verification step. At this step, the CLM verifies the correct structure of the received certificate, the included public information and the CA signature. That is, the CLM performs all the verifications that do not require owning the IIoT device's private key, since this private key never leaves the IIoT device. This pre-validation step allows filtering invalid or malformed certificates before they reach the actual IIoT devices, therefore saving resources in the limited IIoT devices.

B. IIoT DEVICES

For the IIoT devices to be accepted in the proposed system and their certificates managed by the CLM, they must own some root security-related data and must implement some specific services.

First, the IIoT devices must be configured with their identity, their IBC public and private keys and also the identity and IBC public key of the CLM. All these data are loaded in the commissioning phase, before the IIoT devices are actually deployed into the network. Due to the criticality of the stored data, this process is performed offline.

Additionally, the IIoT devices must also implement two CoAP services. The first one is used to generate and sign the corresponding CSR when the CLM asks to do so (*CSR_generation_request* server). This service is programmed so that IIoT devices generate CSRs in PKCS #10 format. The second CoAP service is used to receive the X.509 certificates issued by the corporate CA and forwarded by the CLM, and to verify their correctness (*verify_certificate* server). This verification implies operations that require knowledge of the IIoT device's private key, such as the verification of the public key included in the certificate.

As previously explained, these CoAP communications between the CLM and the IIoT devices are protected by means of a RPK-based DTLS channel, for which the IIoT devices must implement a DTLS server.

C. CORPORATE CA

For automatic certificate (re)enrolment, the CA implements a SCEP server, which verifies each received SCEP request, and in a successful verification case, it generates and signs the corresponding certificate in PKCS #7 format. Then, it sends this certificate to the CLM in a SCEP response message. In our current implementation, the validation of the requests sent by the CLM is based on a pre-configured password, although more advanced mechanisms could be designed and implemented.

D. PRIVATE KEY GENERATOR (PKG)

For the sake of scalability, the DTLS sessions between the CLM and each IIoT device are authenticated by means of raw public keys. These raw public keys have a special feature as they are based on an IBC asymmetric cryptographic scheme. More specifically, these raw public keys are directly an identifier of each user, and the corresponding private keys are created biding this identifier with a system master secret. This approach simplifies the distribution of root authentication credentials to each IIoT device at commissioning phase, while it still allows to establish secure communications within the organization based on public key cryptography.

Therefore, before a new IIoT device is deployed in the network, the PKG computes its corresponding public/private key pair taking as input the identity of the IIoT device and a master secret key owned by the PKG. The solution is based on using the well known pairing-friendly curve BN254. Then, the private and public keys are copied to the IIoT device, while the identity of the new IIoT device along with its public key are registered in the CLM.

As the PKG is a critical entity responsible for generating the pairs of public and private keys that constitute the foundation of the security of the proposed solution, it is an entity that does not even need to be online.

V. LIFE-CYCLE MANAGEMENT

This section details the procedures defined in the proposed system for the life-cycle management of digital certificates, which are also graphically represented in Figure 3.

A. COMMISSIONING PHASE

In the case of IIoT devices, the commissioning phase is the last phase in the manufacturing process, or the first phase before the actual deployment, where IIoT devices are provided with the necessary configuration to successfully operate in the targeted scenario and network. In the case of the solution proposed in this paper, the commissioning phase is related to registering the new device in the CLM, as well as providing the device with an identity, an IBC public/private key pair and basic networking configurations, such as the identity and public key of the CLM. Additionally, if it has not been done during the manufacturing and software image installation, the two CoAP services designed for CSR generation and certificate verification are also installed.

The identity configuration is performed offline and aided by the IBC PKG. In this way, the new IIoT device is provided with an identity which the PKG uses as source information to create the corresponding private and public keys. Then, the new device is registered in the CLM using as entry information its identity and its public key, and the device is marked as ''not operational,'' as it does not own yet a valid digital certificate that would allow it to securely communicate with any other entity within or outside the corporate network.

At this point, the CLM initiates the delegated enrolment process in order to allow the device to move from ''not operational'' to ''operational'' state, which is achieved when the device owns a valid digital certificate.

B. CERTIFICATE ENROLMENT AND RENEWAL

After the commissioning phase, the new IIoT device is registered in the CLM with a ''not operational'' state. So the CLM initiates the enrolment process as detailed in Figure 3. For this aim, the CLM first establishes a RPK-based DTLS channel with the affected IIoT device, using as the receiver's public key, the IBC public key computed from the identity of the corresponding IIoT device. Then, the CLM uses the just established DTLS channel to securely send a query to the *CSR_generation_request* CoAP service running within the IIoT device. This request conveys the CA certificate and might also enclose the information to include in the CSR request.

Upon reception of a *CSR_generation_request*, the IIoT device creates a new public/private key pair and generates a new PKCS #10 CSR, signed with its private key. The information included in the CSR can be either the default information configured in the IIoT device or the information sent along with the query message. Once the signed PKCS #10 CSR is generated, it is sent to the CLM as a response to the received *CSR_generation_request*.

When the CLM receives the PKCS #10 CSR signed by the IIoT device, it encloses it in a new SCEP client request and sends it to the CA. The CA, in turn, verifies the request against its enrolment policy, and if the verification is successful, it signs the requested certificate and returns it to the CLM in PKCS #7 format. Although more sophisticated enrolment policies can be developed, in our current approach, communications between the CLM and the CA are password-protected; and the CA takes as trustworthy any enrolment request coming from the CLM and protected with the right password.

When the CLM receives the PKCS #7 signed certificate issued by the CA, if verifies the format of the certificate and the signature of the CA in a pre-verification process. However, it cannot verify any of the fields protected with the public key of the endpoint IIoT device, as it does not own the IIoT device's private key. Therefore, it forwards the

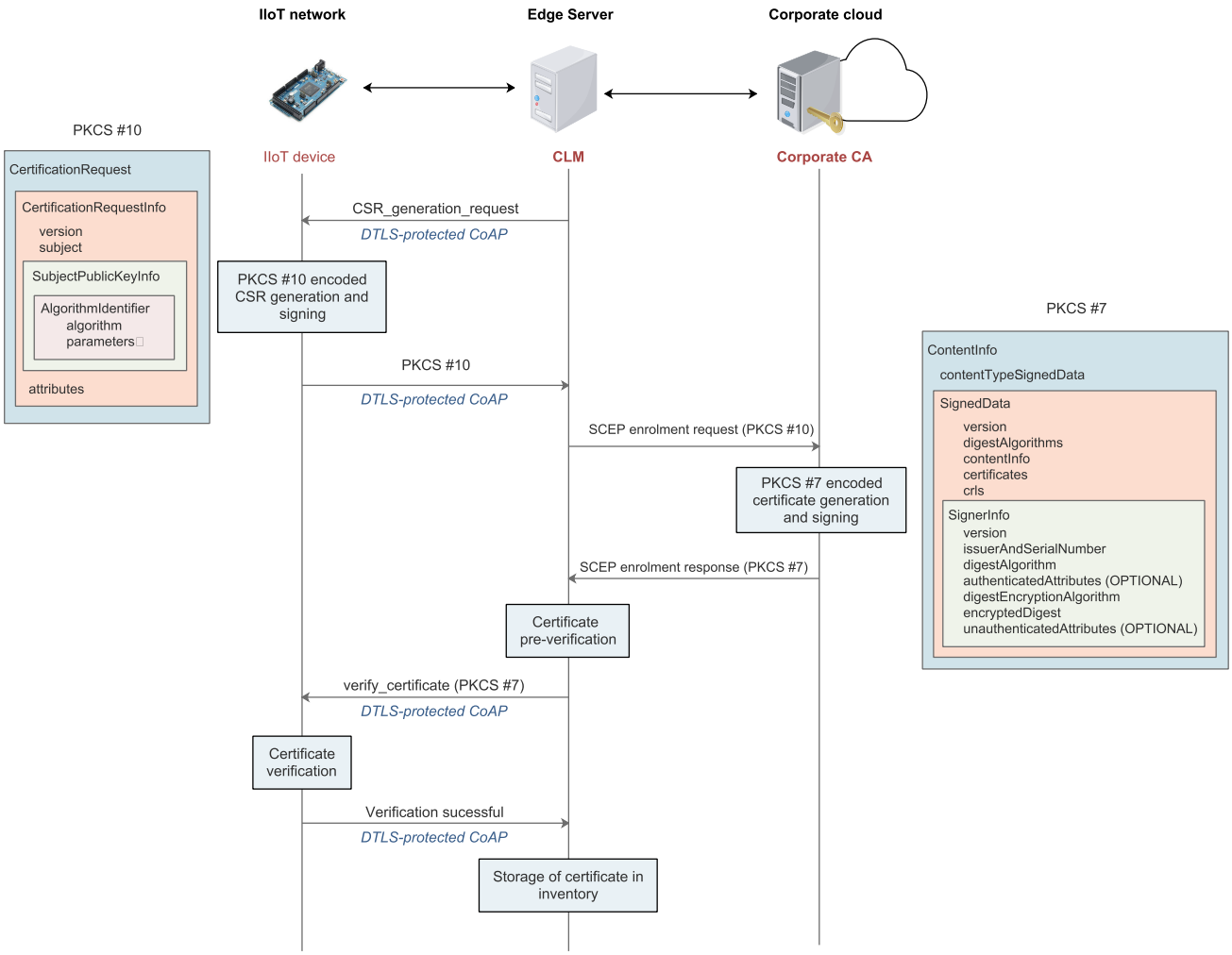


FIGURE 3. Message exchange of the proposed system.

obtained PKCS #7 signed certificate to the IIoT device for its final verification. For this aim, the CLM first establishes a RPK-based DTLS channel with the targeted IIoT device, using as public key the IBC key derived from the IIoT device's identity. Then, it uses the established secure channel to invoke the *verify_certificate* CoAP service, including in the request message the just obtained PKCS #7 certificate. Finally, the IIoT device verifies the PKCS #7 certificate using the public key of the CA and its own private key and responds to the CLM with a code that indicates if the verification has been successful or not.

If the verification of the obtained certificate ends up successfully, the CLM marks the IIoT device as "operational" and registers the enrolment time and the validity period of the certificate, in order to start the renewal process when necessary. In the case that the verification failed, the CLM restarts the enrolment process up to a pre-established number of attempts, after which a notification is sent to the network administrator.

The renewal process consists basically of a new enrolment process, triggered when the certificate expiration date approaches. This process is controlled by the CLM, releasing the resource-limited IIoT devices from the responsibility of permanently checking the validity of their certificates. This is a suitable solution because many IIoT devices do not even have a clock with the current time to check the expiration date of their certificates.

VI. SECURITY CONSIDERATIONS

The proposed solution is heavily based on standard and long-tested technologies, which have been extensively analysed from the security point of view. Nevertheless, the modifications introduced to enable the delegated automatic enrolment may incur in new vulnerabilities, which must also be considered. Therefore, in this section we focus on identifying and analysing the potential security risks that may arise from the new entities and services defined in the proposed system. The goal is not perform a low-level formal security evaluation of the whole system, but to assess from a high-level point of view the security of the newly introduced elements and services. Additionally, we do not consider security risks associated to the operating systems used for testbed

implementation or security vulnerabilities associated to the specific hardware used.

A. CLM

The proposed approach consists of delegating the enrolment request to an edge server known as the CLM. However, the CLM may behave maliciously and try to snoop on transmitted messages or modify them. The first type of attack related to traffic monitoring does not constitute a real risk, since the request information transmitted by the end IIoT device and the certificate generated by the CA as a response, contain both of them, public information. Regarding the potential risk of the CLM modifying or injecting new messages, this is avoided by the fact that private keys are never delegated to the CLM or any other third party. This means that transmitted messages are verified end-to-end by the end IIoT device and the CA, and the CLM is not able to modify any message without it being easily detected when the signature is verified. That is, each party owns a private key which never leaves the owner device. Additionally, the request sent by the client IIoT device is signed using its private key, which removes the possibility of any intermediate entity, including the CLM, modifying the request message in a way that it goes undetected. Similarly, the response message generated by the CA is also signed by the issuing entity. As signatures are verified in the receiving entities, any modification to either the request or response message would be detected.

B. NEW CoAP SEVICES

In order to support the proposed solution, IIoT devices are configured at commissioning phase with two new CoAP services: one, to generate a new PKCS #10 CSR; and the second one, to validate the PKCS #7 format response provided by the CA. These two CoAP services are protected by means of RPK-based DTLS connections. As the DTLS connections are established using IBC-based public keys, each endpoint's public key is directly linked to its identity, which allows to automatically authenticate each endpoint's identity. Additionally, once the DTLS connection is established, the CoAP information transmitted over it is automatically encrypted and integrity-protected by means of the DTLS record protocol. In this way, the use of RPK-based DTLS connections for the protection of CoAP services implicitly removes the potential vulnerability of these services, as DTLS is an extensively validated standard security protocol, which additionally has been specifically designed to protect CoAP communications.

C. IIoT DEVICES

IIoT devices are traditionally difficult to protect. On the one hand, the resource limitations of these devices imply that most of the current security mechanisms, designed for resource-rich devices, are not directly applicable to the IIoT world. Additionally, the massive deployment of IIoT devices makes harder the management of the security in these environments. In this context, the solution presented in this paper entails a step forward towards the protection and availability of strong security mechanisms in IIoT devices. On the one hand, the proposed solution allows IIoT devices to securely obtain a digital certificate, which is a key element for the subsequent implementation of authentication and encryption-key negotiation mechanisms. In this way, IIoT devices are able to implement strong security mechanisms, which reduce their vulnerability window and exposure to attacks. On the other hand, the proposed solution also improves the management of the security in scenarios with a high number of deployed IIoT devices, since certificate enrolment and renewal are managed in a consistent way by a resource-rich edge server.

VII. FUNCTIONAL AND PERFORMANCE EVALUATION

In order to assess the feasibility and suitability of the proposed solution, we have deployed it in the CFFA [16], an advanced manufacturing center launched by the University of the Basque Country along with a wide range of companies of the sector and supported by the local and Basque governments. The CFAA is connected to the Faculty of Engineering of Bilbao (EIB) (about 18 Kms away) and the headquarters of the University of the Basque Country in Leioa (about 15 Kms away) by means of the SN4I [45] experimental facility. SN4I is an experimental network based on Network Function Virtualization (NFV), Software Defied Networking (SDN) and 5G technologies aimed at supporting innovative Industry 4.0 developments and deployments. SN4I interconnects the three premises at Layer 2 and at a data rate of 10 Gbps, by means of fiber optic links and a set of OpenFlow switches. Additionally, the Faculty of Engineering and CFAA sites include 5G access. On the other hand, in each location a virtual infrastructure node, managed by the OpenStack Virtual Infrastructura Manager (VIM), has been deployed; and the three OpenStack nodes are managed by an ETSI Open Source MANO (OSM) [46] Management and Orchestration (MANO) system located in the premises of the Faculty of Engineering of Bilbao.

In such a context, the deployed testbed consists of a Raspberry Pi acting as IIoT device and the CLM and CA services deployed as two Virtual Network Functions (VNFs) in the OpenStack node of the CFAA and managed by the ETSI OSM MANO. The selection of the Raspberry Pi as IIoT device has been motivated by the flexibility it provides for research tasks, specifically the wide availability of software and cryptographic libraries and the easy installation and programming of new services and functionalities.

Apart from providing a real-world functional validation of the proposed solution, the implemented testbed is also used to carry out a performance evaluation, mainly focused on measuring the operational delays incurred by the different phases of the proposed system. In fact, the proposed solution is based on issuing short-lived certificates in order to minimize the possibility of compromised certificates being used as valid. This means that the issued certificates have short lifespans, so that they expire before they could be compromised. As a result, certificate re-enrolments occur very frequently.

For this reason, it is important to assess that these re-enrolments do not entail a bottle neck.

A. TESTBED IMPLEMENTATION

This section details the setting environment for the performance evaluation carried out. In order to assess the feasibility of the proposed architecture, we have implemented a testbed consisting of three machines: a Raspberry Pi 3 Model B acting as IIoT device, a virtual machine running Ubuntu 20.04 to implement the CLM, and a second virtual machine running Debian 10.6 in order to execute the CA. Communications between them are carried out through a SN4I infrastructure, where the raspberry accesses the infrastructure by means of WiFi. The overall implemented testbed is depicted in Figure 4.

For the implementation of the CA functions, we have selected the OpenXPKI - Version 3.8.0 CA [47], because to the best of our knowledge, it is the only open source alternative that implements the standard SCEP automatic enrolment protocol. We have created a RA and a CA hierarchy consisting of two CAs.

The CLM implements the delegated enrolment functionality based on the CertNanny SCEP client software, Version 1.2.0 [48]. In fact, this client software has been selected because the OpenXPKI CA has been tested to run successfully with it. However, in order to allow for a delegated enrolment, the original CertNanny software has been modified at two specific points: (1) CSR obtaining and (2) certificate verification. The first modification involves the CertNanny client establishing a CoAP communication with the corresponding IIoT device in order to obtain a CSR signed by the device. Then, the CertNanny client uses this CSR obtained from the IIoT device to generate the SCEP request and sends it to the CA, following its normal operation. Once it receives the SCEP response from the CA, the CertNanny client pre-verifies the enclosed PKCS #7 certificate with the public information it has available. The whole verification of the obtained PKCS #7 certificate implies the second modification to the original SCEP client program. More specifically, at this point, the SCEP client establishes a second CoAP connection with the IIoT device, forwards the PKCS #7 certificate and waits for a response from the IIoT device, indicating if the verification has been successful or not. In order to perform specific cryptographic operations, such as the pre-validation of the certificate received from the CA or the storage of the certificates successfully validated by the IIoT device, the modified SCEP client program invokes the corresponding OpenSSL commands.

Additionally, in order to establish the aforementioned CoAP communications, the CLM implements the corresponding CoAP clients, which first invoke a DTLS client in order to secure these communications.

The IIoT device implements two CoAP servers with their corresponding DTLS servers in order to protect these communications. The first CoAP server receives CSR generation requests sent by the CLM and invokes the corresponding OpenSSL commands to generate the required private/public key pair and the PKCS #10 CSR. Then, it responds to the CoAP request by sending the generated CSR. The second CoAP server receives the PKCS #7 certificate generated by the CA and forwarded by the CLM and validates the information contained in it. For this aim, it invokes the necessary OpenSSL commands. Then, it responds to the CLM by indicating if the validation has been successful or not.

For the implementation of the secure CoAP communications between the IIoT device and the CLM, the Eclipse Californium Java-based CoAP framework [49], Version 2.5.0, has been used for the development of the corresponding CoAP and DTLS clients and servers. Finally, for the implementation of the PKG in charge of generating the public/private key pairs used in IBC, the MIRACL Crypto SDK C software library [50], Version 4.0, has been used.

B. EXPERIMENTS

In order to assess the performance of our identity management solution, we have measured the time taken by the process at its different stages. We have split the delegated enrolment process in the following steps, which are also graphically represented in Figure 4 with square boxes of dashed lines:

1) CoAP *CSR_generation_request*: this step corresponds to the time taken by the CoAP communication between the CLM and the IIoT device in order to request the IIoT device to generate a new CSR and send it to the CLM. The main contribution to this time is the DTLS handshake process.
2) Private key and CSR generation: this step is performed at the IIoT device after the reception of a request to its *CSR_generation_request* CoAP service. It implies, first, the generation of a public/private key pair, which is the most costly operation; and then, the generation and signing of the corresponding PKCS #10 CSR.
3) Delegated enrolment: this steps gathers the time taken by the execution of the CertNanny SCEP client at the CLM. It starts with the generation of the SCEP request based on the PKCS #10 CSR generated by the IIoT device and finalises once the PKCS #7 certificate has been received from the CA and pre-validated.
4) CoAP *verify_certificate*: this step corresponds to the second CoAP communication between the CLM and the IIoT device, where the CLM sends to the IIoT device the PKCS #7 certificate provided by the CA. As in the case of the first CoAP communication, the main time contribution is also the associated DTLS handshake.
5) Certificate verification: this step accounts for the time taken by the IIoT device in order to validate the information contained in the PKCS #7 certificate generated by the CA and forwarded by the CLM.

Additionally, we have considered different cryptographic conditions, in order to evaluate the impact of each of them

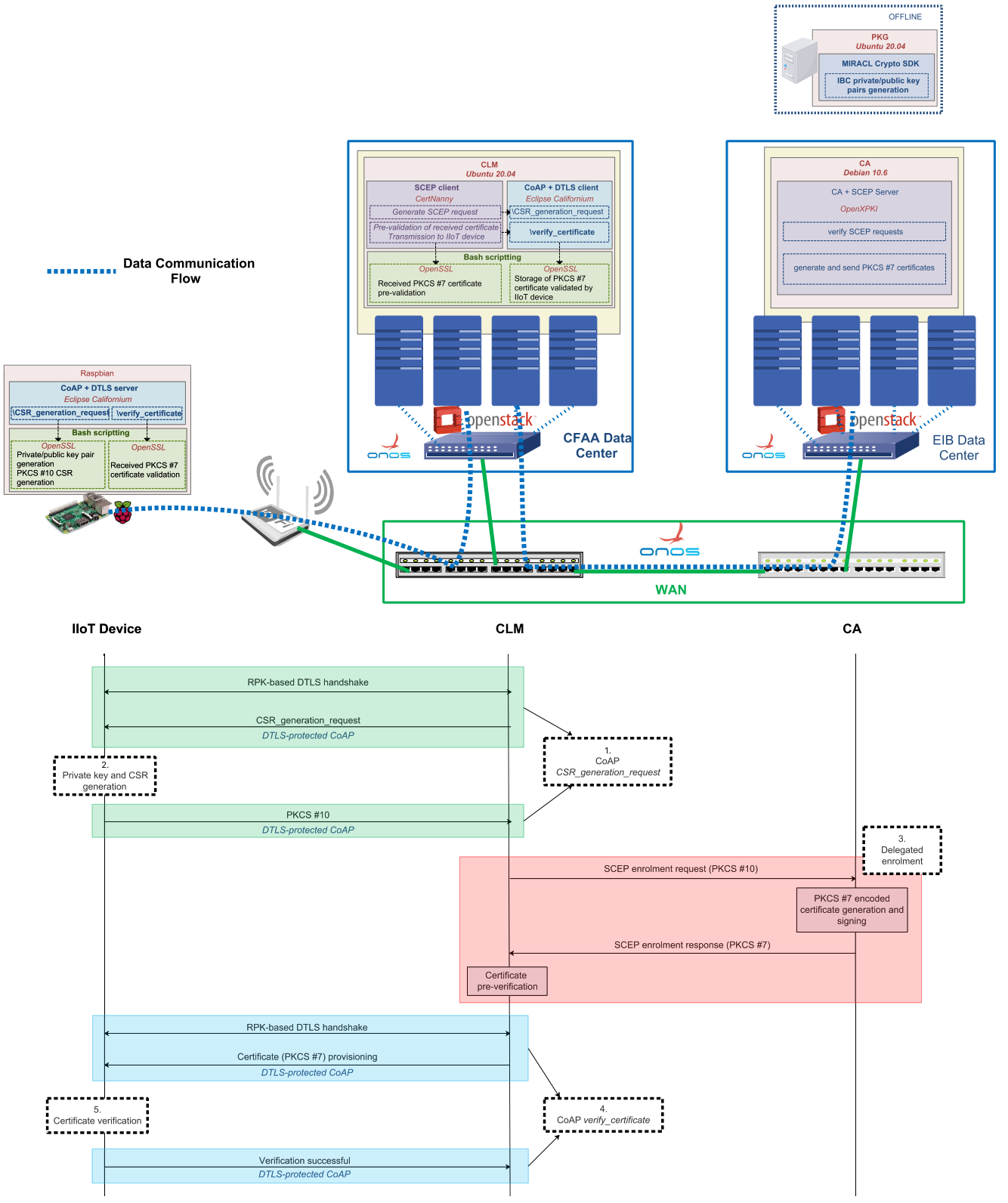


FIGURE 4. Implemented testbed and performed experiments.

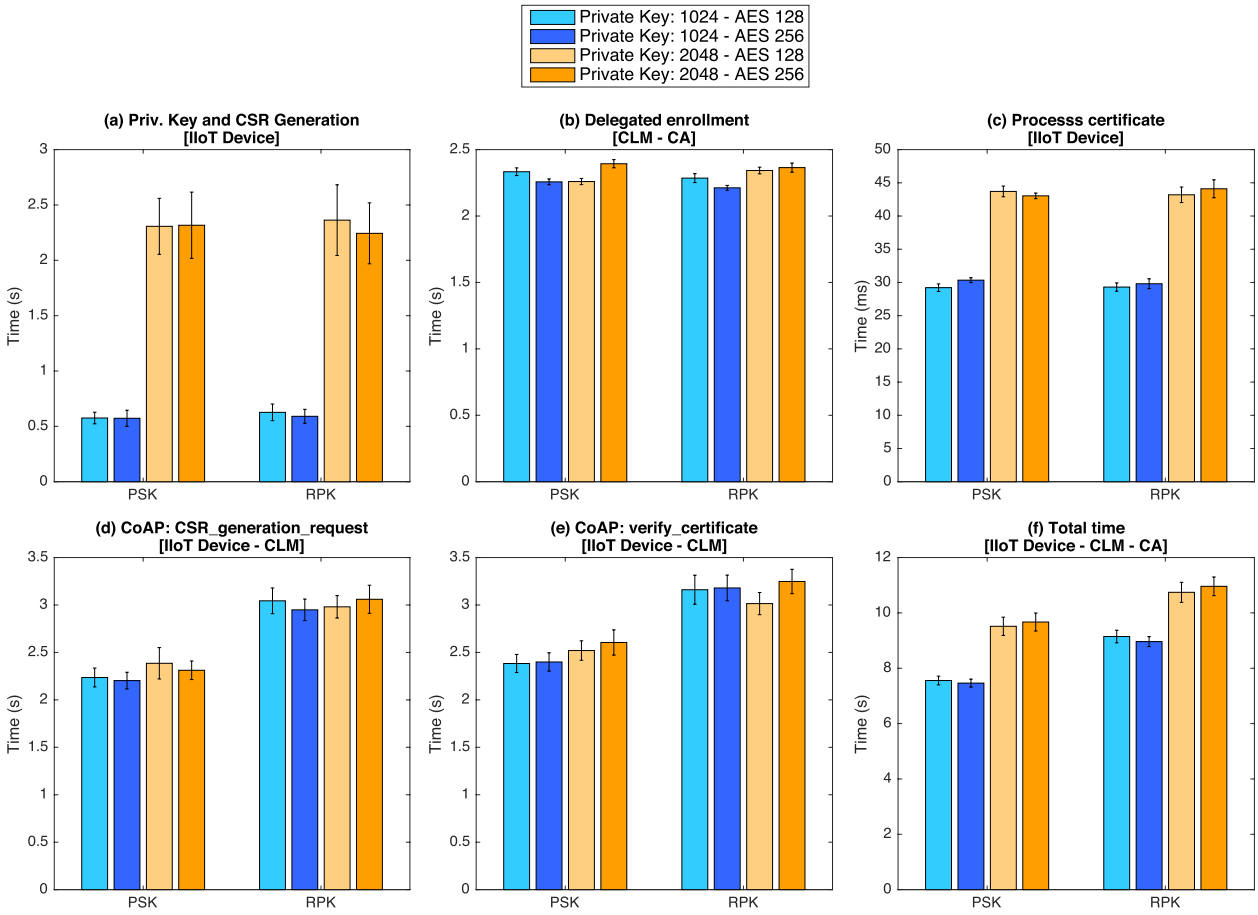


FIGURE 5. Results of the performance evaluation for above described experiments and different configurations of the cryptographic mechanisms. Each chart specifies performed operation and involved entities.

in the performance of the solution. Specifically, we have considered different values for the following parameters:

- Authentication mechanism used in the DTLS handshake: we have considered two different authentication mechanisms for the DTLS connections established between IIoT devices and the CLM. Specifically, we have considered the use of PSKs and RPKs in order to assess the impact of using public key cryptography with respect to the most efficient mechanisms of using symmetric key cryptography.
- Length of the AES key used for symmetric encryption in the DTLS record protocol: Once the DTLS handshake has finished, AES GCM-based symmetric key encryption is used by the DTLS record protocol in order to protect the confidentiality and integrity of the CoAP communications. In this case, we have considered the use of 128 and 256 bit-length symmetric keys.
- Length of the public/private key pairs generated by the IIoT: we have considered the use of 1024 and 2048 bit lengths for the keys to be included in the requested certificates.

C. OBTAINED RESULTS AND DISCUSSION

The implemented testbed has demonstrated that the proposed solution is feasible, easily deployable, secure, efficient and highly scalable. The results of the tests carried out have been gathered in Figure 5 and Table 1, where the execution times measured for the previously defined steps and depending on the selected parameters have been collected. In order to obtain the represented results, experiments with each specific configuration have been repeated 100 times. Figure 5 allows an easy visual interpretation of the obtained mean values and 95% confidence intervals. Table 1, in turn, provides more precise numerical results, including also standard deviations.

The measured results show that the factor that has a greater impact on the execution times is the length of the asymmetric keys used for the certificate generation; specially when it is the resource-deprived IIoT device the entity that must compute it and use it to perform cryptographic operations. In fact, as shown in Figure 5a generating a 2048 bit private key in the IIoT device implies nearly 5 times more time than generating a 1024 bit private key. Similarly, the results in Figure 5c show that using a 2048 bit length private key implies an increase of about 40% in the time needed by the IIoT device to verify the PKCS #7 certificate. On the other hand, Figure 5a also shows that the creation of the private key in the IIoT device is also the operation that entails a greater time variability, which is more evident in the case of 2048 bit length keys.

Another important conclusion of the obtained results is that the use of RPKs for the DTLS handshake implies similar

TABLE 1. Obtained results of the performance evaluation. Mean value, standard deviation and 95% confidence interval of 100 measures for each specific configuration and each measured parameter.

Obtained numerical results								
DTLS auth. mechanism	**PSK**				**RPK**			
Private key length (bits)	**1024**		**2048**		**1024**		**2048**	
AES enc. key length (bits)	**128**	**256**	**128**	**256**	**128**	**256**	**128**	**256**
Private key and CSR creation time (s)								
Mean	0.575	0.5731	2.307	2.317	0.627	0.591	2.363	2.244
Std. Dev.	0.265	0.368	1.276	1.509	0.380	0.317	1.610	1.387
95% Confidence Interval	0.523 - 0.628	0.500 - 0.646	2.054 - 2.560	2.017 - 2.616	0.551 - 0.702	0.528 - 0.654	2.044 - 2.682	1.968 - 2.519
Delegated enrolment time (s)								
Mean	2.333	2.257	2.260	2.394	2.285	2.212	2.343	2.364
Std. Dev.	0.146	0.112	0.118	0.156	0.169	0.092	0.127	0.174
95% Confidence Interval	2.304 - 2.362	2.235 - 2.280	2.236 - 2.283	2.363 - 2.425	2.252 - 2.319	2.194 - 2.231	2.317 - 2.368	2.330 - 2.390
Process certificate time (ms)								
Mean	29.211	30.345	43.706	43.039	29.304	29.809	43.197	44.103
Std. Dev.	2.864	1.841	4.075	2.154	3.183	3.775	5.914	6.829
95% Confidence Interval	28.643 - 29.780	29.980 - 30.710	42.897 - 44.514	42.611 - 43.466	28.673 - 29.936	29.060 - 30.558	42.024 - 44.370	42.748 - 45.458
First CoAP communication time: CSR_generation_request (s)								
Mean	2.236	2.203	2.386	2.312	3.044	2.949	2.981	3.060
Std. Dev.	0.503	0.446	0.834	0.493	0.684	0.573	0.597	0.747
95% Confidence Interval	2.136 - 2.335	2.115 - 2.292	2.220 - 2.551	2.214 - 2.410	2.908 - 3.180	2.835 - 3.063	2.862 - 3.099	2.912 - 3.208
Second CoAP communication time: verify_certificate (s)								
Mean	2.383	2.399	2.520	2.605	3.161	3.180	3.015	3.248
Std. Dev.	0.481	0.488	0.516	0.673	0.772	0.684	0.589	0.648
95% Confidence Interval	2.288 - 2.479	2.303 -2.496	2.417 - 2.622	2.471 - 2.738	3.008 - 3.314	3.044 - 3.315	2.898 - 3.131	3.119 - 3.376
Total time (s)								
Mean	7.557	7.463	9.516	9.670	9.146	8.962	10.744	10.960
Std. Dev.	0.798	0.735	1.662	1.644	1.153	0.907	1.826	1.705
95% Confidence Interval	7.399 - 7.715	7.317 - 7.609	9.186 - 9.846	9.344 - 9.996	8.918 - 9.375	8.782 - 9.142	10.382 - 11.106	10.622 - 11.299

delays to using PSKs, as demonstrated in Figures 5d and 5e. This means that the proposed usage of asymmetric cryptography for authentication during the DTLS handshakes does not have a significant negative impact over the performance of the overall solution with respect to the most efficient solution: the use of symmetric key cryptography based on pre-shared secrets. Similarly, the use of 128 or 256 bit AES symmetric keys for the execution of the DTLS record protocol does not have a big impact on the duration of the protected CoAP communications. Therefore, the most secure configuration based on 256 bit keys should be the preferred option.

It is also relevant to note the high contribution of the two CoAP communications in the overall execution time of the solution (see Figures 5d, 5e and 5f). The performance of the CoAP communications is mainly dominated by the handshake processes of the DTLS connections established to protect them, which emphasizes the high performance cost of the DTLS protocol, specially when resource-deprived devices are involved.

All in all, the whole enrolment process takes about 10 seconds in the worst case, which is a very short time that endorses the feasibility of performing frequent re-enrolments in order to support the use of very short-lived certificates, increasing in this way the overall security of the network.

VIII. CONCLUSION AND FUTURE WORK

Industry 4.0 is the era of IIoT and interconnected machines and processes; and its success will not be possible if network and data security are not guaranteed. In this context, identity management, commonly linked to the distribution and management of X.509 digital certificates, presents important challenges for the organizations. Individual digital certificates must be issued for each user, machine or process within the organization and therefore, their management is specially troublesome in scenarios involving massive IIoT deployments, such as Industry 4.0.

In order to solve this problem, in this paper we present an edge-based certificate management system, which relies on the delegation of the certificate (re)enrolment processes from IIoT devices to a resource-rich edge server, called CLM. The CLM is responsible for maintaining an up-to-date and consistent inventory of issued certificates, checking expiration times and triggering renewal processes when necessary. In this way, the complexity of certificate management is concentrated on the CLM; and the operation of the resource-limited IIoT devices is simplified, relieving them from complex processing tasks, which frequently require checking current time and date.

Additionally, the used certificates are short-lived, which simplifies certificate management by eliminating the need to notify compromised certificates, maintain CRLs and use protocols such as OCSP, which have proved to be resource-expensive and of limited effectiveness. Instead, the use of short-lived certificates results in frequent certificate renewal, which inherently reduces the possibility of certificates being compromised before they expire. In this way, limited IIoT devices do not need to store long CRLs or to execute complex protocols, such as OCSP.

On the other hand, the proposed solution is fully based on standards and widely tested technologies, which perfectly fits the industrial environment.

In order to proof the feasibility of our solution and assess its performance, we have implemented a testbed in a real industrial environment specifically devoted to innovation and experimentation. The obtained results show that the proposed system is fully adequate for the targeted scenarios, easy to

deploy, secure and scalable. Therefore, the proposed system provides an efficient mechanism to support large scale IoT identity management in the Industry 4.0 era.

The work gathered here presents an initial prototype which will be enhanced in future work with more advanced mechanisms to authenticate the enrolment request issued by the CLM, which is now based on a pre-shared password between the CLM and the CA. Additionally, support for additional automatic enrolment protocols, such as EST, will be added.

REFERENCES

[1] L. Tawalbeh, F. Muheidat, M. Tawalbeh, and M. Quwaider, "IoT privacy and security: Challenges and solutions," *Appl. Sci.*, vol. 10, no. 12, p. 4102, Jun. 2020, doi: 10.3390/app10124102.

[2] *IoT Privacy, Data Protection, Information Security*, Eur. Commission, Brussels, Belgium, 2013.

[3] S. Surendran, A. Nassef, and B. D. Beheshti, "A survey of cryptographic algorithms for IoT devices," in *Proc. IEEE Long Island Syst., Appl. Technol. Conf. (LISAT)*, May 2018, pp. 1–8.

[4] M. Suárez-Albela, P. Fraga-Lamas, and T. Fernández-Caramés, "A practical evaluation on RSA and ECC-based cipher suites for IoT high-security energy-efficient fog and mist computing devices," *Sensors*, vol. 18, no. 11, p. 3868, Nov. 2018, doi: 10.3390/s18113868.

[5] S. Haseeb, A. Hashim, O. Khalifa, and A. Ismail, "Connectivity, interoperability and manageability challenges in Internet of Things," in *Proc. AIP Conf.*, vol. 1883, Sep. 2017, Art. no. 020004.

[6] W. Z. Khan, M. H. Rehman, H. M. Zangoti, M. K. Afzal, N. Armi, and K. Salah, "Industrial Internet of Things: Recent advances, enabling technologies and open challenges," *Comput. Electr. Eng.*, vol. 81, Jan. 2020, Art. no. 106522.

[7] E. Topalovic, B. Saeta, L.-S. Huang, and C. Jackson, "Towards short-lived certificates," in *Proc. IEEE Oakland Web Secur. Privacy (W SP)*, May 2012, pp. 1–9.

[8] *SSL/TLS Certificate Validity Capped at a Maximum of Two Years*. Accessed: Jun. 29, 2021. [Online]. Available: https://www.globalsign.com/en/blog/ssl-certificate-validity-capped-at-maximum-two-years

[9] *Apple Limits Validity Period of TLS Certificates to 398 Days*. Accessed: Jun. 29, 2021. [Online]. Available: https://support.apple.com/en-us/HT211025

[10] J. Höglund, S. Lindemer, M. Furuhed, and S. Raza, "PKI4IoT: Towards public key infrastructure for the Internet of Things," *Comput. Secur.*, vol. 89, Feb. 2020, Art. no. 101658. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404819302019

[11] Z. Shelby, K. Hartke, and C. Bormann, *The Constrained Application Protocol (CoAP)*, document IETF, RFC 7252, 2014.

[12] E. Rescorla, H. Tschofenig, and N. Modadugu, *The Datagram Transport Layer Security (DTLS) Protocol Version 1.3*, document IETF Internet-Draft, 2020.

[13] E. H. Wang, Y. Yang, X. Kang, and Z. Cheng, *Using Identity as Raw Public Key in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)*, document IETF Internet-Draft, 2019.

[14] L. Chen, "An interpretation of identity-based cryptography," in *Foundations of Security Analysis and Design IV*, A. Aldini and R. Gorrieri, Eds. Berlin, Germany: Springer, 2007, pp. 183–208.

[15] P. Gutmann, *Simple Certificate Enrolment Protocol*, document IETF, RFC 8894, 2020.

[16] *Aeronautics Advanced Manufacturing Center*. Accessed: Jun. 29, 2021. [Online]. Available: https://www.ehu.eus/en/web/cfaa/home

[17] S. Magnusson, "Evaluation of decentralized alternatives to PKI for IoT devices. A litterature study and proof of concept implementation to explore the viability of replacing PKI with decentralized alternatives," M.S. thesis, Dept. Electron., KTH. Skolan för Elektroteknik och Datavetenskap, Stockholm, Sweden, 2018.

[18] M. Cebe and K. Akkaya, "Efficient certificate revocation management schemes for IoT-based advanced metering infrastructures in smart cities," *Ad Hoc Netw.*, vol. 92, Sep. 2019, Art. no. 101801. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870518307844

[19] J. Zhang, H. Zhong, J. Cui, Y. Xu, and L. Liu, "An extensible and effective anonymous batch authentication scheme for smart vehicular networks," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3462–3473, Apr. 2020.

[20] D. Kelly and M. Hammoudeh, "Optimisation of the public key encryption infrastructure for the Internet of Things," in *Proc. ICFNDS*. New York, NY, USA: ACM, 2018, doi: 10.1145/3231053.3231098.

[21] M. E. S. Saeed, Q.-Y. Liu, G. Tian, B. Gao, and F. Li, "Remote authentication schemes for wireless body area networks based on the Internet of Things," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4926–4944, Dec. 2018.

[22] D. Diaz-Sanchez, A. Marín-Lopez, F. A. Mendoza, P. A. Cabarcos, and R. S. Sherratt, "TLS/PKI challenges and certificate pinning techniques for IoT and M2M secure communications," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3502–3531, May 2019.

[23] J. Park and N. Kang, "Lightweight secure communication for CoAP-enabled Internet of Things using delegated DTLS handshake," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2014, pp. 28–33.

[24] J. Han and D. Kim, "A back-end offload architecture for security of resource-constrained networks," in *Proc. IEEE 15th Int. Symp. Netw. Comput. Appl. (NCA)*, Oct. 2016, pp. 383–387.

[25] E. Cho, M. Park, H. Lee, J. Choi, and T. T. Kwon, "D2TLS: Delegation-based DTLS for cloud-based IoT services," in *Proc. Int. Conf. Internet Things Design Implement. (IoTDI)*. New York, NY, USA: ACM, 2019, pp. 190–201, doi: 10.1145/3302505.3310081.

[26] F. Marino, C. Moiso, and M. Petracca, "PKIoT: A public key infrastructure for the Internet of Things," *Trans. Emerg. Telecommun. Technol.*, vol. 30, no. 10, Oct. 2019, Art. no. e3681. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.3681

[27] E. Cho, J. Kim, M. Park, H. Lee, C. Hamm, S. Park, S. Sohn, M. Kang, and T. T. Kwon, "TwinPeaks: An approach for certificateless public key distribution for the Internet and Internet of Things," *Comput. Netw.*, vol. 175, Jul. 2020, Art. no. 107268. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128618314051

[28] M. Rossberg and M. Theil, "Secure enrollment of certificates using short PINs," in *Proc. 12th Int. Conf. Availability, Rel. Secur. (ARES)*. New York, NY, USA: ACM, Aug. 2017, pp. 1–9, doi: 10.1145/3098954.3098988.

[29] Z. He, M. Furuhed, and S. Raza, "Indraj: Digital certificate enrollment for battery-powered wireless devices," in *Proc. 12th Conf. Secur. Privacy Wireless Mobile Netw.* New York, NY, USA: ACM, May 2019, pp. 117–127.

[30] L. Duan, Y. Li, and L. Liao, "Flexible certificate revocation list for efficient authentication in IoT," in *Proc. 8th Int. Conf. Internet Things*. New York, NY, USA: ACM, Oct. 2018, pp. 1–8, doi: 10.1145/3277593.3277595.

[31] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology*, G. R. Blakley and D. Chaum, Eds. Berlin, Germany: Springer, 1985, pp. 47–53.

[32] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Advances in Cryptology—CRYPTO 2001*, J. Kilian, Ed. Berlin, Germany: Springer, 2001, pp. 213–229.

[33] A. Weil, "Sur les fonctions algébriques à corps de constantes fini," *CR Acad. Sci. Paris*, vol. 210, no. 1940, pp. 592–594, 1979.

[34] X. Boyen and L. Martin, *Identity-Based Cryptography Standard (IBCS) #1: Supersingular Curve Implementations of the BF and BB1 Cryptosystems*, document IETF, RFC 5091, 2007.

[35] M. Groves, *Elliptic Curve-Based Certificateless Signatures for Identity-Based Encryption (ECCSI)*, document IETF, RFC 6507, 2012.

[36] M. Groves, *Sakai-Kasahara Key Encryption (SAKKE)*, document IETF, RFC 6508, 2012.

[37] M. Pritikin, P. Yee, and D. Harkins, *Enrollment Over Secure Transport*, document IETF, RFC 7030, 2013.

[38] J. Schaad and M. Myers, *Certificate Management Over CMS (CMC)*, document IETF, RFC 5272, 2008.

[39] C. Adams, S. Farrell, T. Kause, and T. Mononen, *Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)*, document IETF, RFC 4210, 2005.

[40] R. Barnes, J. Hoffman-Andrews, D. McCarney, and J. Kasten, *Automatic Certificate Management Environment (ACME)*, document IETF, RFC 8555, 2019.

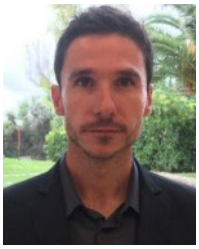
[41] *Internet Security Research Group*. Accessed: Jun. 29, 2021. [Online]. Available: https://www.abetterinternet.org

[42] G. Selander, S. Raza, M. Vucinic, M. Furuhed, and M. Richardson, "Enrollment with application layer security," IETF, Work in Progress, Fremont, CA, USA, Tech. Rep. draft-selander-ace-eals-01, 2017.

[43] G. Selander, J. Mattsson, F. Palombini, and L. Seitz, *Object Security for Constrained RESTful Environments (OSCORE)*, document IETF, RFC 8613, 2019.

[44] G. Selander, J. Mattsson, and F. Palombini, "Ephemeral Diffie-Hellman over COSE (EDHOC)," IETF, Work in Progress, Fremont, CA, USA, Tech. Rep. draft-ietf-lake-edhoc-01, 2020.
[45] J. Sasiain, A. Sanz, J. Astorga, and E. Jacob, "Towards flexible integration of 5G and IIoT technologies in industry 4.0: A practical use case," *Appl. Sci.*, vol. 10, no. 21, p. 7670, Oct. 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/21/7670
[46] *Open Source MANO*. Accessed: Jun. 29, 2021. [Online]. Available: https://osm.etsi.org
[47] *The OpenXPKI Project*. Accessed: Jun. 29, 2021. [Online]. Available: https://www.openxpki.org
[48] *CertNanny*. Accessed: Jun. 29, 2021. [Online]. Available: https://github.com/certnanny/CertNanny
[49] *Eclipse Californium (Cf) CoAP Framework*. Accessed: Jun. 29, 2021. [Online]. Available: https://projects.eclipse.org/projects/iot.californium
[50] *MIRACL Crypto SDK*. Accessed: Jun. 29, 2021. [Online]. Available: https://github.com/miracl/MIRACL

JASONE ASTORGA received the B.Sc. and M.Sc. degrees in telecommunication engineering and the Ph.D. degree from the University of the Basque Country (UPV/EHU), in 2004 and 2013, respectively. From 2004 to 2007, she worked with Nextel S.A., a Telecommunications Enterprise. She joined the UPV/EHU, as a Lecturer, in 2007, and as a Researcher with the I2T Research Laboratory. She is currently an Assistant Professor with the UPV/EHU. She has participated in several research projects at the local, national, and European levels. She has supervised two Ph.D. theses. Her research interests include software defined networking and network function virtualization for industrial communications, IP-enabled wireless sensor networks, and cybersecurity.



MARC BARCELO received the M.Sc. degree in electrical engineering (Hons.) and the Ph.D. degree *(cum laude)* from the Universitat Autonoma de Barcelona (UAB), in 2010 and 2015, respectively. From 2009 to 2015, he was with the UAB's Telecommunications and Systems Engineering Department participating in several international research and development projects. He was an International Research Intern with Nokia Bell Labs, Berkeley Heights, NJ, USA, in 2014. Since 2016, he has been with the Ikerlan Technology Research Centre, Basque Research and Technology Alliance (BRTA), where he is currently a Researcher with the Cybersecurity for Digital Platforms working group. He has published over 20 articles in recognized international journals and conferences. His research interest includes the design of secure communication architectures for the Internet of Things and Industry 4.0.

AITOR URBIETA received the Ph.D. degree in computer science from the University of Mondragon, Spain, in 2010. He studied informatics engineering at the University of Mondragon. Since 2007, he has been with the Ikerlan Technology Research Centre, Basque Research and Technology Alliance (BRTA). He is currently a Research Fellow and belongs to the Digital Platform Cybersecurity Team as the Team Leader. He is the author or coauthor of more than 20 peer-reviewed scientific publications in the field of the Internet of Things, service-oriented architectures, and semantic web. During the last ten years, he has been working on the development and validation of the several IoT platforms for various domains (transport, energy, and smart manufacturing), with emphasis on cybersecurity and safety issues. His current research interests include cybersecurity, the Internet of Things (IoT), machine to machine (M2M), industrial control systems, digital platforms, fog computing, edge computing, the IoT environments validation, the IoT environments testing, and middleware.

EDUARDO JACOB (Senior Member, IEEE) received the B.Sc. degree in industrial engineering and the M.Sc. degree in industrial engineering and industrial communications and electronics, in 1987 and 1991, respectively, and the Ph.D. degree in communications engineering from the University of the Basque Country (UPV/EHU), in 2001. During two years, he worked with a public telecommunications research and development enterprise (currently Tecnalia). He spent several years as the IT director in the private sector. Since 1994, he has been at full-time with the UPV/EHU, where he was elected as the Head of the Department of Communications Engineering, from 2012 to 2016. He is currently a Full Professor and leads the I2T (Engineering and Research on Telematics) Research Laboratory. He has also directed several Ph.D. theses and managed several research projects at the local, national, and European levels. His research interests include applying software-defined networks to industrial communications, cybersecurity in distributed systems, software-defined wireless sensor networks, and in-network processing.