IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 12, 2021, accepted June 26, 2021, date of publication July 9, 2021, date of current version July 20, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3096062

On the Security of IIoT Deployments: An Investigation of Secure Provisioning Solutions for OPC UA

FLORIAN KOHNHÄUSER[1], DAVID MEIER[2], FLORIAN PATZER[2], AND SÖREN FINSTER[1]
[1]ABB Corporate Research Center, 68526 Ladenburg, Germany
[2]Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB, 76131 Karlsruhe, Germany

Corresponding author: Florian Kohnhäuser (florian.kohnhaeuser@de.abb.com)

This work was supported in part by the German Federal Ministry of Education and Research in the scope of the Basissystem Industrie 4.0 (BaSys) 4.2 Project under Grant 01IS19022B.

ABSTRACT A key technology for the communication in the Industrial Internet of Things (IIoT) is the Open Platform Communications Unified Architecture (OPC UA). OPC UA is a standard that enables interoperable, secure, and reliable communication between industrial devices. To defend against cyber attacks, OPC UA has built-in security mechanisms that protect the authenticity, integrity, and confidentiality of data in transit. Before communicating securely, it is essential that OPC UA devices are set up in a secure manner. This process is referred to as secure provisioning. An improper provisioning can lead to weak or insecure OPC UA deployments that enable adversaries to eavesdrop or even manipulate communication between industrial devices. Such insecure deployments can also be maliciously provoked by adversaries who tamper with insecure provisioning solutions. Despite secure provisioning is essential for OPC UA security and usability, there exists no overview and systematic analysis on the patchwork of different solutions in industry and academia. This article presents the first investigation of secure device provisioning solutions for the OPC UA communication protocol. First, desired objectives and evaluation criteria for secure provisioning of OPC UA devices are defined. Next, existing and emerging OPC UA provisioning solutions are analyzed and compared based on the elaborated objectives and criteria. Additionally, an outlook into the future of OPC UA provisioning is given, based on solutions from the IoT domain. Finally, the analyzed OPC UA secure provisioning solutions are compared, recommendations are given, and research gaps are identified. It is shown that contemporary provisioning solutions offer an insufficient level of security. Emerging and future solutions provide much higher security guarantees but impose a tradeoff between usability and requirements on devices and infrastructures.

INDEX TERMS Communication system security, device provisioning, Industrial Internet of Things (IIoT), industry 4.0, network security, OPC UA, secure provisioning.

I. INTRODUCTION

In the course of the emerging Industry 4.0 [1], components for industrial automation systems, such as controllers, sensors, and actuators are becoming increasingly interconnected. This increase in connectivity facilitates data collection, data analysis, and automation, which eventually improves the productivity and efficiency of industrial facilities. A key technology for enabling the communication between industrial systems is the *Open Platform Communications Unified Architecture (OPC UA)* [2]. OPC UA is a common standard that allows seamless data exchange and interoperability between devices from different manufacturers. However, increasing connectivity between industrial components also raises their risk of being target of cyber attacks. To defend against attacks, OPC UA defines multiple security modes that enable devices to protect the authenticity, integrity, and confidentiality of data in transit.

Before devices are able to protect their communication using the OPC UA security modes, they must be securely provisioned. During secure provisioning, devices are transformed from their manufacturing state to a configured state that enables using the devices in a functional and secure manner. In case of OPC UA, secure device provisioning

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

TABLE 1. Overview of investigated products, standards, and academic works.

Category	Section	Subjects
OPC UA PLCs	IV-B	B&R X20 System [6], WAGO PFC200 [7], Phoenix Contact AXC F 2152 [8], Siemens SIMACTIC S7-1500 [9]
OPC UA Gateways	IV-B	Unified Automation UaGateway [10], Softing uaGate SI [11], Turck RFID-Gateway [12]
IoT Platforms	VI-A	Microsoft Azure IoT Hub [13], Amazon AWS [14], Google Cloud IoT Core [15]
OPC UA Standards	IV-A	OPC 10000-12 [16]
	V-B	OPC 10000-21†
IoT Standards	VI-B	RFC8366 [17], RFC8572 [18], BRSKI [19], 6tisch [20], DPP [21]
Academic Works	V-A	[22], [23], [24], [25], [26], [27], [28]

† No reference available, since OPC 10000-21 is not yet published.

includes supplying devices with the necessary configuration, user credentials, device credentials, and trust relationships for secure communication. Since every device needs provisioning, the effort for device provisioning scales at least linearly with the number of devices. Large installations of devices are therefore highly sensitive to delays introduced by provisioning. Especially manual tasks are time-consuming and thus impractical for large installations. For this reason, it is desired to automate device provisioning as much as possible, up to a degree where no manual intervention is required, which is referred to as zero touch provisioning.

Automated provisioning solutions not only provide the means to scale deployments from single to dozens of devices, they also have an essential impact on the security of OPC UA deployments. An inappropriately executed provisioning can lead to insecure OPC UA configurations that enable adversaries to eavesdrop or even manipulate communication between industrial devices. By performing such attacks, adversaries can steal confidential data or tamper with industrial processes, causing economical damage, physical damage, and human harm [3]. Despite the availability of partly automated mechanisms for secure provisioning of OPC UA and guidelines on how to apply them [4], correct application of these mechanisms is still a challenge. In fact, it has recently been shown that 92% of all Internet-facing OPC UA devices are configured deficiently, e.g., due to missing access control, disabled security modes, use of deprecated cryptographic primitives, or certificate reuse [5]. In addition to properly applying a provisioning solution, it is also vital that the solution itself provides a sufficient level of security. Insecure provisioning solutions enable adversaries to manipulate the provisioning process. This allows adversaries to maliciously configure and corrupt OPC UA deployments.

Although secure provisioning is essential for OPC UA security and usability, it is so far inappropriately addressed both in industry and academia. Today, a mixture of proprietary, standardized, and academic provisioning solutions are prevalent and there exists no work that has systematically analyzed and compared these solutions. This makes it hard for users and operators to reason about the requirements, degree of automation, applicability, and security properties of OPC UA secure provisioning solutions.

This article presents the first investigation of secure device provisioning solutions that are available for OPC UA or could be applied to OPC UA. The investigation includes existing provisioning solutions for OPC UA from industry and academia, as well as related provisioning solutions from the *Internet of Things (IoT)* domain. Table 1 provides an overview of the investigated products, standards, and academic works. To enable a fair and systematic comparison, fourteen evaluation criteria are defined and applied to the regarded provisioning solutions. Finally, the results of the comparison are discussed and recommendations are given.

A. CONTRIBUTIONS

This investigation of secure device provisioning solutions for OPC UA provides the following contributions:

- Definition of objectives and evaluation criteria that are used for subsequent evaluations of OPC UA secure device provisioning solutions (Section III)
- Analysis and evaluation of
  - state-of-the-art secure device provisioning solutions for OPC UA, collected from existing standards and industrial products (Section IV)
  - emerging secure device provisioning solutions for OPC UA, gathered from academic works and current standardization efforts (Section V)
  - potential future secure device provisioning solutions for OPC UA, based on solutions from the IoT domain (Section VI)
- Identification of research gaps and recommendations on secure provisioning for OPC UA (Section VII)

II. BACKGROUND

This section covers an introduction to OPC UA security mechanisms and the objectives of secure device provisioning for OPC UA.

A. OPC UA SECURITY

The OPC UA specification [29] defines three security modes for secure communication: None, Sign, and SignAndEncrypt. These modes offer unprotected communication (None), authenticated communication (Sign), and authenticated as well as confidential communication (SignAndEncrypt). In addition, multiple security policies are defined that specify the cryptographic algorithms and their parameters to realize the different security modes. All security modes and policies have in common that an OPC UA client and an OPC UA server first need to perform a security handshake to establish an OPC UA communication channel. Figure 1 illustrates this security handshake. During the security handshake, client and server authenticate themselves

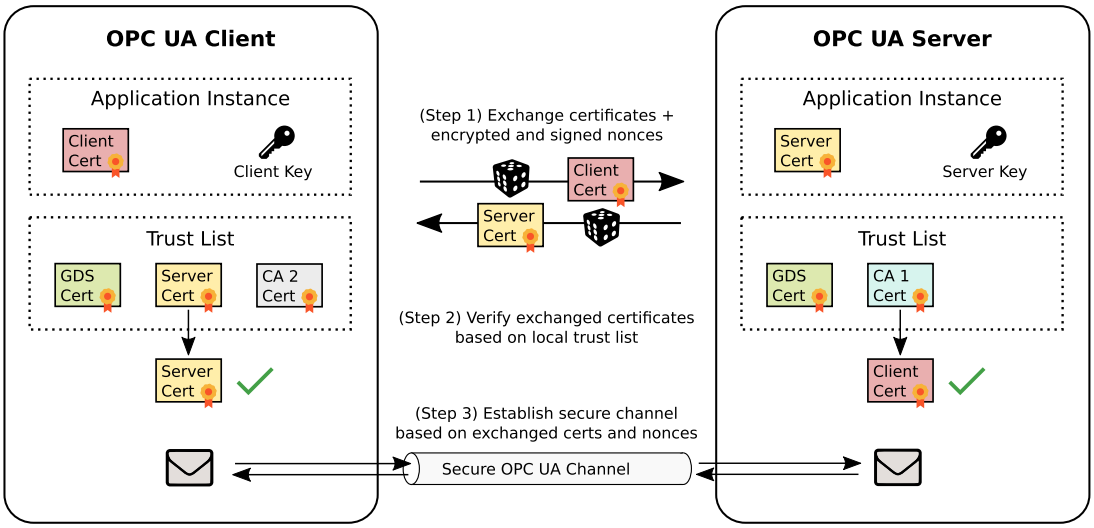

FIGURE 1. Illustration of establishing a secure OPC UA connection between an OPC UA client and OPC UA server.

using their OPC UA *Application* Instance Certificate, which are X.509 compliant digital certificates. These certificates are mutually exchanged and verified between devices when initiating a secure communication session. To verify received certificates, each device maintains a *Trust List*. This list consists of certificates that are trusted by the device. A received certificate is considered valid, if it is either in the trust list itself (see client in Figure 1) or part of a certificate chain that has an anchor in the trust list (see server in Figure 1). Devices store their certificates in a so-called *Certificate Store*, which contains a separate location for own certificates (Application Instance Certificates) and trusted certificates (Trust List).

In OPC UA, the management of certificates and trust lists can be carried out by a *Global Discovery Server (GDS)*, as defined in part 12 of the OPC UA specification (OPC 10000-12) [16]. The GDS can automatically discover devices on the network and provide them with the necessary certificates and trust lists. The GDS can also serve as a link to an existing *Public Key Infrastructure (PKI)* that can be used to issue certificates for OPC UA devices and applications. Unfortunately, the GDS does not solve secure provisioning. This is because OPC UA does not specify how a GDS and a device initially establish trust in each other. For this reason, *trust on first use (TOFU)* is a common approach that is followed when the GDS and a device communicate for the first time. In the TOFU model, it is assumed that the certificate received from an endpoint on the first connection attempt is genuine. Devices store this certificate in their trust list and use it to verify all subsequent connections to the respective endpoint. TOFU relies on the assumption that an adversary does not tamper with the first connection to the device. An adversary who can tamper with the first connection can impersonate a GDS towards a benign device or impersonate a device towards a benign GDS. Thus, TOFU allows the adversary to corrupt the provisioning of devices and undermine the security guarantees provided by OPC UA. For more information about the GDS, please refer to Section IV-A, which provides a detailed analysis of the OPC 10000-12 specification.

In 2016, the German Federal Office for Information Security (BSI) published a security study of OPC UA [4]. In the study, the OPC UA specification in version 1.02 was analyzed and the ANSI C implementation of the OPC Foundation was tested. The security study concludes that OPC UA provides a high level of security and that no systematic errors were detected. Unfortunately, provisioning was not studied in detail, mainly because part 12 of the OPC UA specification (OPC 10000-12) was still in an early state.

Please note that part 14 of the OPC UA specification defines an additional publish-subscribe mode (OPC UA PubSub) [30]. OPC UA PubSub is outside the scope of this article, as it uses different security mechanisms. Instead of relying on a mutual authentication with certificates, it makes use of a central key server that distributes symmetric keys for protected communication.

B. SECURE DEVICE PROVISIONING OBJECTIVES

Device provisioning is the task of transforming a device from its manufacturing state into an operational state. Secure device provisioning adds security as a requirement to this process. The end result of secure provisioning is a functional state of the device that complies with the security objectives of the operator. A secure functional state depends on the particular application setting. For OPC UA, it includes: (i) configuration of security modes and policies, (ii) configuration of user credentials (e.g., users, groups, and passwords), (iii) configuration of credentials on the device itself as well as on related devices (e.g., Global Discovery Servers, Aggregation Servers), and (iv) establishment of trust relationships (e.g., configuring the appropriate certificates in the respective trust lists). For a secure provisioning, the transformation of the device from the manufacturing state to the operational state must happen in a secure manner. A secure transformation includes:

- *Device Identification and Authentication:* Only legitimate devices can connect to an operator's network.
- *Network Identification and Authentication:* From a list of available networks, the device only connects to legitimate ones. This is more important in wireless use cases.
- *Configuration:* Transfer the necessary information for the device to operate in a secure way. This may, in addition to necessary configuration for functional operation, also contain information to connect to another network (i.e., when the first network is used only temporarily as a provisioning network).

The aforementioned secure provisioning objectives are referred to in the next section, when defining the comparison criteria.

III. SECURE OPC UA DEVICE PROVISIONING COMPARISON CRITERIA

This section presents criteria that enable analyzing, comparing, and assessing device provisioning solutions for OPC UA. In general, secure device provisioning solutions impose different requirements on involved parties and vary in their assumptions and goals. For a thorough comparison between secure device provisioning solutions, their differing properties need to be considered. In specific, the following comparison criteria are regarded:

1) DEVICE REQUIREMENTS

Secure provisioning solutions impose different requirements on devices, since they require devices to possess certain hardware properties to operate or to achieve their security goals. In particular, the following device requirements were identified:

- *Hardware Resources:* Necessary computational resources to implement and execute the respective provisioning solution, e.g., enough computing power to perform expensive cryptographic operations.
- *Secure Hardware:* Hardware features to enable support for specific security operations, e.g., secure boot, remote attestation, or secure storage for cryptographic material.
- *Out-of-band Communication Channel:* An additional way to get information from the device, e.g., via NFC, Bluetooth, USB dongle, or QR code printed on device casing.

The fewer device requirements a secure provisioning solution has, the more devices are suited to implement the solution. Thus, low device hardware requirements are desirable.

2) INFRASTRUCTURE REQUIREMENTS

Secure device provisioning solutions impose different requirements on the infrastructure of actors that are involved in the supply chain, e.g., manufacturers, vendors, integrators, or operators. Thus, actors must initially set up the necessary infrastructure before a specific secure provisioning solution can be put into operation. The following infrastructure requirements were identified:

- *Unique Device Credentials:* Manufacturers must have the necessary production infrastructure to equip each manufactured device with unique cryptographic credentials.
- *Offline Manufacturing CA:* Manufacturers must maintain a Certificate Authority (CA) that signs electronic information during the manufacturing of devices, e.g., IEEE 802.1AR DevID certificates [31].
- *Online Manufacturing CA:* Manufacturers must maintain a CA that is remotely accessible and signs electronic information for devices during the provisioning phase.
- *Global Discovery Server:* Operators must maintain a Global Discovery Server (GDS) that automates the OPC UA certificate management according to OPC 10000-12 [16] (see Section IV-A).
- *Provisioning Device:* Operators must maintain a specific device that automates the secure provisioning of devices.

Low infrastructure requirements are desirable to minimize the cost, complexity, and effort for setting up the necessary secure provisioning infrastructure.

3) DEGREE OF AUTOMATION

Secure provisioning solutions impose a varying degree of manual effort on actors. The degree of automation largely impacts the scalability of a provisioning solution, as manual effort constitutes the bottleneck for provisioning. Three degrees of automation are considered in this work:

- *Low Automation:* Solutions in this category need significant manual effort per device. This is, for instance, the case, if a device certificate needs to be manually generated and copied to each device.
- *Medium Automation:* Solutions in this category provide semi-automated provisioning with low manual effort per device. For example, when each device must be connected to an automated provisioning tool.
- *High Automation:* This category contains solutions that provide automated provisioning with practically no manual effort per device. For example, when after a one-time manual configuration of the provisioning system, devices only need to be connected to the network.

To reduce manual effort and increase scalability, provisioning solutions should strive for a high degree of automation.

4) INDUSTRIAL APPLICABILITY

The applicability of a secure provisioning solution is another important evaluation criteria. It determines the effort required to apply a solution to the industrial (IoT) environment. The specific properties of the industrial environment, which may differ from typical IoT environments, need to be considered. These properties include, for example, limited or non-existing connectivity to other networks (incl. Internet access), unavailability of a directory service (like Active Directory), or real-time demands. The following categories are regarded:

- *Low Applicability:* Substantial efforts are needed to apply the provisioning solution in industrial environments.
- *Medium Applicability:* The solution can be applied to most industrial use cases, but may require workarounds.
- *High Applicability:* The solution can be used in industrial environments without requiring modifications.

A high industrial applicability is most desirable, as it leads to improved adoption chances of a provisioning solution.

5) APPLICABILITY TO OPC UA

This evaluation criteria assesses the effort required to apply the solution to OPC UA. The following categories are considered:

- *Low Effort:* The solution can easily be integrated into OPC UA, for example by mapping the necessary steps of the solution to methods already available in OPC UA and using services already defined in OPC UA.
- *Medium Effort:* To apply the solution to OPC UA, new concepts have to be introduced to OPC UA that are not yet defined in the OPC UA specification.
- *High Effort:* The solution would need fundamental changes in at least some OPC UA concepts or introduce new dependencies outside of OPC UA and would hardly be compatible with existing OPC UA implementations.

The lower the effort, the fewer changes need to be made to the provision solution or an OPC UA implementation, which is most desirable.

6) SECURITY PROPERTIES

Secure device provisioning solutions aim at different security goals and thus offer varying levels of security. To analyze security, it is crucial to define the attacker's capabilities. This work assumes the well-known Dolev-Yao adversary model [32] that includes the characteristic capabilities of an internal attacker. Hence, the adversary can eavesdrop, modify, delete, or insert any message between devices in the network. To fulfill the secure provisioning objectives, a secure provisioning solution must accomplish the following security properties against a Dolev-Yao adversary:

- *Authentication of Devices:* Devices authenticate themselves towards the operator's network during provisioning. Note that this may require the exchange of credentials, e.g., certificates, before provisioning.
- *Authentication of Operator:* The operator's network authenticates itself towards the device during provisioning. Note that this may require the exchange of credentials, e.g., certificates, before provisioning.
- *Protected Communication Channel:* The integrity and confidentiality of data that is exchanged during provisioning is protected, including, for instance, secret information, such as private keys.

In general, secure provisioning solutions that offer a high security level are favorable. However, depending on the particular deployment scenario, certain security goals may not be required, e.g., authentication of the network towards devices.

IV. OPC UA SECURE PROVISIONING: STATUS QUO

In this chapter, the state-of-the-art in OPC UA device provisioning is investigated by first summarizing the existing OPC UA specification in Section IV-A and then analyzing OPC UA products on the market in Section IV-B.

A. CURRENT STANDARD: OPC 10000-12

Secure device provisioning is mainly addressed in part 12, Discovery and Global Services, of the OPC UA specification [16]. OPC 10000-12 describes the various functions of the *Global Discovery Server (GDS)*. Most relevant for secure provisioning are the *Certificate Management* functions of the GDS. They are used to manage and distribute certificates and trust lists to OPC UA applications. This is performed by using either the *Pull* or *Push Management* operation mode. In Pull Management, a client calls functions on the GDS to request a new application certificate or retrieve a list of trusted certificates. In Push Management, the GDS is remotely calling functions on an OPC UA application to trigger the creation of a certificate request, install new certificates, or update the certificate trust list of the application. The specification envisions Pull Management to be used by OPC UA clients and Push Management to be used for OPC UA servers.

Concerning the initial provisioning, the specification gives only few references on how to handle it securely. The GDS shall operate in a mode where any OPC UA application can connect with an arbitrary certificate and use administrative credentials to authenticate against the GDS. For OPC UA servers, the specification proposes a provisioning state. OPC UA servers in provisioning state allow clients, e.g., a GDS, to connect with an untrusted certificate, authenticate with administrative credentials, and perform the certificate provisioning using Push Management. After this provisioning, only clients with trusted certificates shall be able to connect and perform certificate management actions. The specification also describes the possibility for an application vendor to use an out-of-band channel for the initial certificate provisioning.

1) EVALUATION

While the usage of secure storage is encouraged by the OPC UA specification, it is not mandatory. All provisioning steps can be achieved using the OPC UA protocol itself without relying on out-of-band channels. OPC 10000-12 does not pose any infrastructure requirements other than having a GDS available and enables a highly automated provisioning. It is fully applicable to the industrial environment, as only local infrastructure is needed. As it is part of the OPC UA specification, its implementation requires only low effort and all communication is protected by OPC UA secure channels. However, TOFU is proposed for the first connection between devices and the GDS. Thus, there is no authentication of neither the operator nor the device itself during

provisioning. This allows an adversary to perform so-called *Man-in-the-Middle (MITM)* attacks, in which the adversary impersonates a device towards a benign GDS and vice versa. This way, the adversary can first sniff the administrative credentials from a connection with the benign GDS and then use those credentials to authenticate itself towards the benign device. Thus, the adversary has complete control over the provisioning and can deploy forged certificates to undermine the security of all subsequent OPC UA connections with the device.

B. OPC UA COMMERCIAL PRODUCTS

This section investigates OPC UA *Programmable Logic Controllers (PLCs)* and OPC UA gateways on the market to analyze the state-of-the-art in secure device provisioning for OPC UA. The analysis focuses on the most popular OPC UA products coming from manufacturers with the largest market shares. More specifically, four PLCs, the B&R X20 System [6], WAGO PFC200 [7], Phoenix Contact AXC F 2152 [8], and Siemens SIMACTIC S7-1500 [9], as well as three OPC UA gateways, the Unified Automation UaGateway [10], Softing uaGate SI [11], and Turck RFID-Gateway [12], are regarded. It is refrained from mapping findings to specific manufacturers and products. Instead, the amount of products fulfilling a particular criterion is quantified and compared to the rest of products. This way, anonymity is guaranteed to some degree without distorting conclusions that can be drawn from the analysis. The findings are based on information from product manuals as well as investigations of the provided product tools and testing of the actual products. Table 2 summarizes the main results, which are elaborated in the following paragraphs.

There are two options provided by the investigated products to manage OPC UA certificates: (i) a *Web-based Management (WBM)* interface, and (ii) an engineering software. Three out of the seven devices implement a WBM. Thus, they run a Web server that can be accessed with a Web browser to configure the OPC UA server. In addition, five of the seven devices enable OPC UA management over their engineering tool. Engineering tools are commonly used by technicians to configure and program the PLC or gateway device. One device offers OPC UA management over both WBM and engineering tool. In this case, the engineering tool provides more configuration options than the WBM.

On all devices, the first connection with the web browser or engineering tool to the device is either unprotected or based on a not yet established trust relationship. In case of WBM interfaces, two devices make use of default administrator credentials over a plain (unprotected) HTTP connection. After connecting to the device, the default credentials can be changed, but protection via HTTPS cannot be activated. Thus, an adversary who eavesdrops on the communication between an administrator and the devices, can sniff the administrator credentials and then impersonate the administrator to control the device. On the third device that offers a WBM, the security level is slightly higher, since the device is shipped with a self-signed TLS certificate and unique administrator credentials printed on the device casing. However, as long as the WBM certificate is not in the administrator's trusted root certification authorities store, an adversary can perform MITM attacks and likewise sniff credentials during login. This is in particular the case for the initial connection, where the administrator receives the device's WBM certificate for the first time.

TABLE 2. Summary of OPC UA product analysis.

Category	Property	Products (Total: 7)
Tool Type	Web-based Management	3
	Engineering Software	5
Tool Setup	Trust On First Use	7
	Protected Communication*	7
Tool Features	Generate Own Certificate	6
	Manage Trust List	7
	Manage CA Certificates	1
OPC 10000-12	GDS Push Model Support	2
	GDS Pull Model Support	1
Trust Anchor	DevID Certificates	1†

* Protected communication only possible after provisioning.
† Note that product does not make use of IDevID certificate.

A similar level of security is provided by the investigated engineering tools. Initially, all devices communicate with their engineering tools over an unprotected communication channel using default administrator credentials. After login, the devices allow administrators to change credentials and to protect the communication channel by deploying certificates and using TLS. Nevertheless, this mechanism likewise allows an adversary to perform MITM attacks on the initial connection, sniff credential, and bypass any security measures established thereafter. For this reason, none of the analyzed products fulfill any of the defined security goals (see Section III-6) on a secure provisioning solution.

After login to the WBM or engineering tool, OPC UA certificates can be managed as follows: All investigated products initially come with a self-signed OPC UA certificate. This certificate can be regenerated ad hoc in all but one product. The particular product only allows to upload a new OPC UA certificate that has been generated externally (e.g., by openssl). Using the WBM interface or engineering tool, all products allow retrieving the current OPC UA certificate, which is needed to install the certificate in the trust list of OPC UA communication partners. WBMs and engineering tools also support the other way around, as they allow managing the products' trust list by adding and removing OPC UA certificates from other parties.

To ease the deployment and management of OPC UA certificates, the tool of one device is additionally capable of managing CA certificates. It allows generating a CA certificate and corresponding OPC UA certificates that are signed by the CA. This is useful when provisioning multiple devices, as only a single CA certificate needs to be provided to communication partners, instead of multiple self-signed certificates.

When it comes to guidelines and best practices on secure provisioning for OPC UA, there exists a specific certificate deployment strategy only for one product. In all other cases, OPC UA certificate deployment and management is left unspecified. Furthermore, only two devices support a Global Discovery Server (GDS) for managing certificates after provisioning, of which only a single device supports both push and pull model (see Section IV-A). The rest of the devices only allow certificate deployment and management over their respective provisioning tool. In addition, one device is equipped with an IEEE 802.1AR IDevID certificate [31] by the manufacturer. IDevID certificates can be used to bootstrap trust by verifying the authenticity of the device. They are typically used by more advanced device provisioning solutions (see Sections V-B and VI-B). However, the particular device provides no mechanisms to make use of the equipped IDevID certificate and the certificate is also not mentioned in the documentation.

1) EVALUATION

All analyzed OPC UA products offer tools for deploying and managing OPC UA certificates, either in form of a WBM interface, an engineering software, or both. The device requirements are medium, since devices have no secure storage or special hardware features, but need to run a dedicated service for communicating with a web browser or engineering software. Demands on the infrastructure of manufacturers and operators are typically very low. This is because only one manufacturer equipped his product with unique credentials and a certificate signed by the manufacturer CA. Furthermore, since provisioning is performed via WBM and/or engineering tool, operators are not required to maintain a GDS or a particular provisioning device. Due to the low device and infrastructure requirements, applicability in the industrial context is high. On the other hand, the degree of automation is low, since the provisioning is a highly manual process for the operator who needs to manually connect to involved devices and modify their trust list and their OPC UA application certificates. In addition, a lack of guidance and best practices make the secure provisioning of the analyzed products an obscure process that requires external knowledge in OPC UA as well as security. Furthermore, most products still do not support OPC 10000-12, despite the standard being released for more than five years. Thus, certificate management after provisioning requires manual effort, leading to scalability issues in large networks. Most concerning, however, is the weak security level of the provisioning process in the analyzed OPC UA products. In all cases, the initial connection phases are prone to attacks that enable undermining any security measures provided by OPC UA. The root cause for this security issue is the absence concepts and mechanisms to bootstrap trust before initiating secure connections. For instance, trust can be bootstrapped from certificates that are deployed via a secure out-of-band channel, as described in the following chapters.

V. OPC UA SECURE PROVISIONING: EMERGENCE

In this chapter, emerging OPC UA device provisioning solutions are investigated. Section V-A summarizes academic solutions related to secure provisioning for OPC UA. Section V-B describes the OPC 10000-21 draft, an upcoming standard for secure provisioning of OPC UA devices.

A. ACADEMIC WORK

The different possible trust models in OPC UA, based on certificates, have been discussed and evaluated by Fernbach and Kastner [22]. Furthermore, Karthikeyan and Heiss [23] studied the usage of certificates in OPC UA and the challenges thereof. Both publications come to the conclusion that the certificate management process of OPC UA requires further study, but both works do not inspect the actual provisioning of devices, thus, the bootstrapping of trust.

More relevant investigations for the bootstrapping of trust in OPC UA have been conducted by Birnstill *et al.* [24] and Bienhaus *et al.* [25]. They researched the utilization of *trusted platform modules* (*TPMs*) for integrity attestation and key management in OPC UA applications. This approach can guarantee that credentials or specific application states are unmodified (e.g., using remote attestation). Nevertheless, both works do not solve the provisioning, since they assume credentials or anchors of trust to be distributed beforehand.

In order to solve the provisioning issue, two major types of approaches have been proposed so far. The first approach makes use of certificates or keys that are stored on mobile USB dongles or SD cards [26], [27]. Blume *et al.* [27] described an approach for industrial environments, where a USB dongle is connected to a network participant, which has to run a specific software. This software is used to enable the OPC UA-based communication between the dongle and a Licence Central (LC). This way, the certificates stored on the USB dongle can be used by applications on the device and are manageable by the LC. An alternative has been presented by Meier *et al.* [28] based on an approach that establishes initial trust using a portable provisioning device. By using a direct wire connection between the provisioning device and a target device, the target device is provided with secure onboarding credentials. This mechanism has been applied to OPC UA based on standardized GDS functions. For this purpose, the provisioning device hosts a GDS that provisions the target device using TOFU. Security is ensured by the direct connection between provisioning device and target device, which prevents the presence of a Dolev-Yao attacker.

1) EVALUATION

Both approaches, solving the provisioning issue, leverage additional hardware and out-of-band communication [27], [28]. However, whereas the first approach [27] requires a proprietary software on each network participant and the dongles must remain connected to the devices in order for them to remain functional, the second approach [28] requires just one provisioning device that is only used for the short period of

provisioning and does not demand any additional software on the provisioning target. Furthermore, both techniques require certain (local) PKI- and/or GDS-functionality and offer a moderate level of automation. Due to the high hardware- and software-requirements, the industrial applicability of the dongle solution is in the low to medium range. The provisioning device solution, however, is highly applicable to OPC UA and in industrial systems due to its compliance to the OPC UA standard. Since the dongle approach is basically an external certificate store, it should also not require much effort to use it with OPC UA. Security of the provisioning device approach has been successfully validated against a Dolev-Yao attacker. Yet, security has not been analyzed for the dongle approach. However, since both approaches exclude the actual network from the provisioning, the security criteria from section III can be argued to be met by both of them.

B. UPCOMING STANDARD: OPC 10000-21 DRAFT

OPC 10000-21 is an emerging standard for the secure and automated provisioning of OPC UA devices [33]. While most device provisioning use cases focus on the relationship and security functions between device, manufacturer, and operator, OPC 10000-21 also takes intermediate steps of the device lifecycle into account. It assumes that a device is handled by a multitude of parties before it is put into operation: manufactured by a vendor, assembled into a compound device by a machine builder, sold by a distributor, installed, and configured by an integrator, and finally operated, maintained, and decommissioned by an operator. OPC 10000-21 sets the goal of not only providing a secure device identification and authentication but also a secure log of all stages. Figure V-B illustrates the security measures of OPC 10000-21 during the device lifecycle, which are also described in the following.

To bootstrap security, manufacturers equip devices with an asymmetric key pair and an IEEE 802.1AR Initial Device Identity (IDevID) certificate [31] for this keypair. The IDevID is used as trust anchor for the device to prove its identity. Using the IDevID, manufacturers produce a so-called *DeviceIdentityTicket*, a separate digital document that describes a device (e.g., device name, revision, and serial number), and that is signed by the manufacturer with a separate PKI. Through the IDevID, a DeviceIdentityTicket is bound to the specific device. Once trust in a DeviceIdentityTicket is established by checking the manufacturers signature, trust in the source and genuineness of the device can be established by checking if it has the private key to the corresponding IDevID.

Since OPC 10000-21 takes the whole lifecycle of a device into consideration, it also provides mechanisms for a *MachineIdentityTicket*, which may include a number of DeviceIdentityTickets. A MachineIdentityTicket is produced and signed by a machine builder. Trust into a machine includes recursive trust into all included devices, which can be checked via the included DeviceIdentityTickets. With these mechanisms, each actor in the lifecycle can use the tickets provided by the previous actor to establish trust in devices. This works even for large composite devices. Eventually, the last actor in the chain, the operator, verifies the authenticity of the (composite) device before deployment. After verification, operators rely on OPC UA discovery (using the OPC 10000-12 standard) to securely provide new devices with necessary configurations and certificates for the network. This way, in combination with OPC 10000-12, protection against malicious devices and eavesdropping as well as manipulation of the used communication channels is provided.

However, the current draft does not include mechanisms to protect the device itself from being provisioned by a malicious actor, since there are no mechanisms for identification and authentication of the operator and its network. This functionality can be added with standard OPC UA user authentication, though, if manufacturer and operator can agree on the necessary credentials.

1) EVALUATION

Although a strong trust anchor is needed for devices, there are no additional device requirements, given that the selected cryptographic primitives can be used for OPC UA secure channels as well (e.g., using an RSA keypair). If the IDevID keypair uses cryptographic primitives that are not used in OPC UA, then the implementation of these cryptographic primitives are an additional requirement for the device. Cryptographic keys are recommended to be stored in a hardware-protected memory. All communication with the device is done via OPC UA. Devices do not need an additional out-of-band communication channel, which saves complexity and cost in manufacturing. Having no additional communication channel also saves security hardening effort in operation. Infrastructure requirements, however, are significant. For unique device credentials, manufacturers must adjust manufacturing processes to generate these credentials and sign them via an (offline) manufacturing CA. Additionally, the ticket system needs another, potentially unrelated PKI, which - in theory - could be realized as a separate offline PKI but is more likely to be implemented by using the already existing web PKI. Provisioning infrastructure needs not only compliant provisioning servers, but also an OPC 10000-12 GDS implementation. These applications could potentially reside on the same physical server. How tickets are managed and distributed is out of scope for the standard. It is likely, though, that automated solutions for this will emerge. Given that ticket management can be automated, OPC 10000-21 allows for a high automation degree, is fully applicable to the industrial use case, and - being an OPC UA standard - realizable in OPC UA with low effort. While security properties include full authentication of devices towards the operator's network, support for authentication of the operator's network towards devices is not implemented in the current draft. Otherwise, a high level of security is achieved.

VI. OPC UA SECURE PROVISIONING: OUTLOOK BASED ON THE INTERNET OF THINGS DOMAIN

Secure provisioning is also an important process in the lifecycle of IoT devices [34]. Since OPC UA is a communication

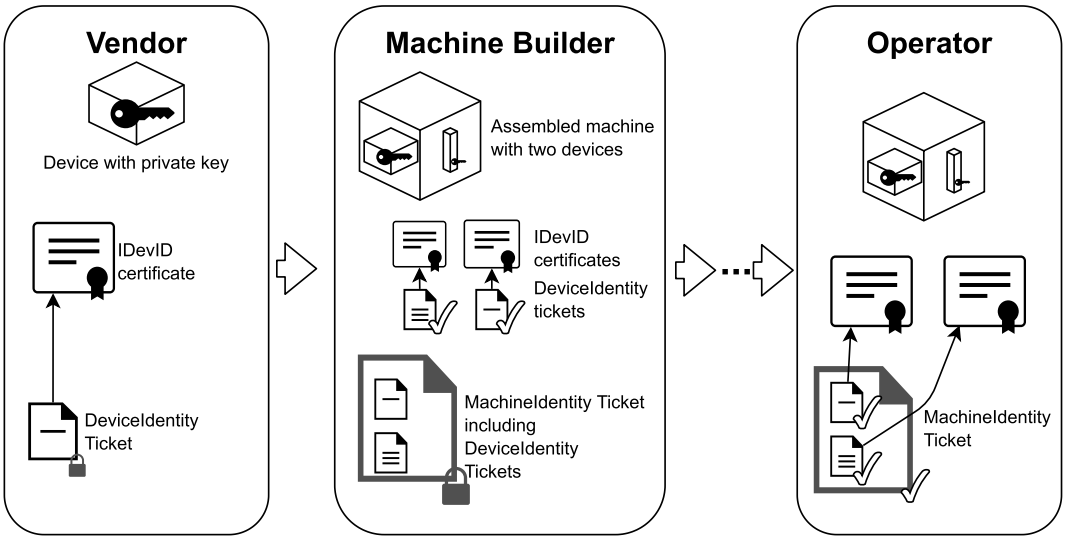

FIGURE 2. Illustration of OPC 10000-21 security measures during device lifecycle. A vendor manufactures a device with a public/private keypair, an IDevID certificate and a signed DeviceIdentityTicket that includes a reference to the certificate. A machine builder assembles a machine using two devices and checks the corresponding DeviceIdentityTickets in the process. For the new machine, a MachineIdentityTicket is generated and signed. An operator finally checks the MachineIdentityTicket, the included DeviceIdentityTickets, and uses the certificates for establishing secure OPC UA channels with the devices.

protocol for Industrial IoT (IIoT), there are similarities and synergies between secure provisioning for IoT, IIoT, and OPC UA. Therefore, in this chapter, an outlook into the potential future of secure provisioning for OPC UA is given based on solutions from the IoT domain. Section VI-A takes a deeper look into the provisioning of contemporary IoT products. Section VI-B then outlines current and future standards for the provisioning of IoT devices.

A. IoT COMMERCIAL PRODUCTS

To investigate the secure provisioning in commercial IoT products, this analysis focuses on the most wide-spread and mature IoT platforms. In specific, three platforms are investigated: (i) Microsoft Azure's IoT Hub Device Provisioning Service (IOT-DPS) [13], (ii) Amazon's AWS Device Provisioning [14], and (iii) Google's Cloud IoT Core [15]. It is again refrained from naming specific products in the findings.

In general, device provisioning follows the same principles on all three platforms. They all require devices to be initially equipped with specific credentials that enable them to securely connect to the provisioning service of the respective platform. Over this secure connection, the provisioning service configures the device, such that the device then securely connects to the actual service on the platform. However, there are differences between platforms in the way devices initially authenticate themselves towards the provisioning service.

All three platforms allow authentication based on a self-signed or CA-signed X.509 device certificate that belongs to a unique device key. In this case, a CA certificate or self-signed certificate that enables verifying the device certificate must be provided to the respective service before provisioning. In addition, there are three more authentication mechanisms, each offered by one platform. The first is authentication based on a symmetric key that is shared between device and service. The second is authentication based on the device's TPM, where the TPM endorsement key is used to bootstrap security. In this case, the device's attestation key and registration ID must be provided to the platform before provisioning. The third mechanism is bootstrapping security from a trusted user. It allows a trusted party, e.g., a mobile app, to obtain a temporary provisioning certificate for the device that should be provisioned. Using the temporary provisioning certificate, the device securely retrieves a permanent certificate and configuration for regular operations from the provisioning service.

1) EVALUATION

On average, device requirements are comparable to the previously described OPC 10000-21 draft (see Section V-B) and, thus, medium. Exceptions are (i) TPM-based authentication, which demands secure hardware features, and (ii) authentication based on a symmetric key, which relies on less computationally intensive cryptography and thus demands less hardware resources. Requirements on the infrastructure are dependent on the particular party. Manufacturers must equip each of their devices with unique credentials, which may also imply maintaining a (local) CA to issue X.509 certificates. In addition, demands on the platforms are high. They must realize a provisioning service that is always accessible via Internet, which includes maintaining one or multiple CAs, databases, APIs, etc. Since the heavy-lifting is done by manufacturers and platforms, on the operator's side, infrastructure requirements are low and the degree of automation is high. In specific, operators must only ensure Internet access for devices to realize a fully automated provisioning. Nevertheless, for security or infrastructure reasons, industrial devices

are often disconnected from the Internet. Therefore, applicability in the industrial context is problematic. In addition, significant effort is needed to port the platform-based provisioning approaches to an OPC UA use case. Finally, the provided security properties are high. With the offered authentication mechanisms and the platform-centric approach, mutual authentication is realized during provisioning. This is not achieved with any OPC UA-related provisioning solution. The only exception is authentication based on a trusted user, which provides less security guarantees, but enables an alternative approach for cases where devices cannot be equipped with credentials beforehand.

B. EMERGING IoT SECURE PROVISIONING STANDARDS

The basis for several IoT provisioning protocols is RFC8366 [17], which is well-known in the domain of IoT provisioning standards. RFC8366 specifies a *voucher artifact*, which is a YANG-defined [35] JSON document that can be used to securely integrate a new device, called pledge, into a network. It expects the device to be equipped with an IDevID. The details of the secure provisioning process using this artifact are defined by subsequent specifications based on RFC8366. The voucher artifact is signed by the manufacturer of the pledge, who implements a *Manufacturer Authorized Signing Authority (MASA)*. It holds a defined set of provisioning information, including an expiry date, the device's serial number, assertions, and a nonce. Most importantly, the voucher can convey a *pinned-domain-cert* that the device can use to authenticate the owner. The validity of vouchers can vary, ranging from one-time use to time-bound usages.

The IETF draft BRSKI [19] is one of the specifications that are based on RFC8366 vouchers. It extends the *Enrollment over Secure Transport (EST)* [36] standard to establish secure provisioning, enabling the pledge to also establish trust in the network using the owner's registrar. It defines the respective steps for the secure provisioning: Discover, Identify, Request Join, Imprint, Enroll and Enrolled. In BRSKI, devices are exclusively identified using their serial-number. It also extends the RFC8366 voucher definition, adding a *prior-signed-voucher-request*, as well as a *proximity-registrar-cert*. This enables a proximity confirmation of the pledge and registrar. BRSKI also extends EST with multiple well-known addresses to perform the voucher handling. Being based on RF8366, BRSKI depends on an initial device identity certificate that needs to be installed by the vendor. In BRSKI, the trust establishment with the manufacturer is out-of-scope.

RFC8572 [18], also referred to as *Secure Zero Touch Provisioning (SZTP)*, is another voucher-based secure provisioning specification. It aims at enabling non-technical workers to securely bring-up new devices in remote networks, without the need for prior configuration. Its functionalities include updating the boot image, adding an initial configuration, and executing arbitrary scripts. After executing SZTP, a device should be able to establish secure connections with the network. In SZTP, the sources for these information can be removable media, a DNS server, a RESTCONF bootstrap server, or DHCP. SZTP requires devices to follow a specific boot sequence. Because SZTP is relying on RESTCONF [37] and NETCONF [38], it allows to enroll certificates at any later time, in contrast to BRSKI.

The *6tisch Zero-Touch Secure Join protocol* [20] is a profile of the *Constrained-Voucher Draft* [39], which adjusts RF8366 to low-end devices. It aims to enable the secure introduction of new nodes to a 6tisch network without requiring direct modifications of the node. Compared with BRSKI, the following protocol details are changed or replaced: HTTP is replaced with CoAP, TLS is replaced with EDHOC/OSCOAP and CoAP, Anchor certificate is replaced with Raw Public Key, PKCS7 signed JSON is replaced with COSE signatures, GRASP discovery mechanism is replaced with beacon announcement, and proxies as well as expiry dates are not used. The main purpose of these changes are to improve the applicability to resource constrained devices.

In contrast to the previously described standards, the Device Provisioning Protocol (DPP) [21] does not build upon RFC8366. Instead, it was developed by the WiFi Alliance as a successor of the WiFi Protected Setup (WPS), and is part of WPA3. Its goal is to easily integrate clients into a WiFi network while eliminating the weak points of WPS. With DPP, devices can join a network without the need to enter login details, like an SSID and a key. Thus, DPP is, in particular, suited for IoT devices that have no user interface. For secure provisioning, an access point and an IoT device make use of an out-of-band communication channel, e.g., established via QR code or NFC tag. The operator uses a smartphone with a provisioning app to initiate a secure channel between access point and IoT device based on the out-of-band channel. Finally, the access point configures the device over the secure channel.

1) EVALUATION

In general, device requirements are medium. RFC8366 recommends the usage of HSMs to protect cryptographic material, while DPP prescribes the implementation of an out-of-band communication channel. The infrastructure needs depend on the standard. Voucher-based solutions have very high infrastructure requirements. They demand that manufacturers equip their devices with unique IDevID certificates and maintain a provisioning service that issues certificates during the provisioning process. Furthermore, they require operators to set up a provisioning device or service, for example, the registrar in BRSKI. DPP, on the other side, only needs a smartphone with an app and a compliant access point as infrastructure. RFC8366-based solutions can be highly automated and the provisioning steps are clearly defined. Manual steps mainly concern security checks of logged information. In DPP, manual effort is needed to obtain information from the out-of-band channel with the smartphone, which is why the degree of automation and industrial applicability is medium. Because RFC8366-based solutions implement an online process for voucher request and creation, these solutions are not fully applicable to typical industrial use cases.

TABLE 3. Summary of evaluation for secure device provisioning solutions

	Device Requirements			Infrastructure Requirements					Usability			Security Properties		
	Hardware Resources	Secure Hardware	Out-of-band Channel	Unique Device Credentials	Offline Manufacturing CA	Online Manufacturing CA	Global Discovery Server	Provisioning Device	Degree of Automation	Industrial Applicability	OPC UA Applicability	Device Authentication	Operator Authentication	Protected Communication
OPC 10000-12	low	no	no	no	no	no	yes	no	high	high	high	no	no	yes
OPC UA Products														
No GDS support	low	no	no	no	no	no	no	no	low	high	high	no	no	yes*
With GDS support	low	no	no	no	no	no	yes	no	medium	high	high	no	no	yes*
Academic Works														
Blume et al. [27]	high	no	yes	no	yes	no	no	yes	medium	low	medium	implicit	yes	yes
Meier et al. [28]	low	no	yes	no	no	no	yes	yes	medium	high	high	implicit	yes	yes
OPC 10000-21 Draft	medium	yes	no	yes	yes	no	yes	yes	high	high	high	yes	no	yes
IoT Products														
X.509 Auth.	medium	no†	no	yes	yes	no	-	no	high	low	low	yes	yes	yes
Shared Key Auth.	low	no†	no	yes	no	no	-	no	high	low	low	yes	yes	yes
TPM Auth.	medium	yes	no	yes	no	no	-	no	high	low	low	yes	yes	yes
Trusted User Auth.	medium	no†	no	no	no	no	-	yes	medium	low	low	implicit	no	yes
IoT Standards														
RFC8366-based	medium	no†	no	yes	yes	yes	-	yes	high	low	medium	yes	yes	yes
DPP	medium	no	yes	no	no	no	-	yes	medium	medium	medium	implicit	no	yes

* Limited protection level due to missing authentication.
† No, but feature is recommended.

All described standards can be applied to OPC UA, albeit requiring significant amendments. RFC8366-voucher solutions provide sound security, as communication is protected and devices as well as networks can be authenticated. DPP provides a lower level of security. This is because devices cannot verify the authenticity of the network. In addition, security heavily relies on the integrity of the smartphone and the proximity between devices and the smartphone to ensure exclusive access on the out-of-band channel.

VII. DISCUSSION

This chapter provides a final evaluation of the investigated provisioning solutions from Section IV, V, and VI. The considered products, standards, and academic works are discussed, compared, and assessed. Table 3 provides an overview by mapping the investigated solutions against the comparison criteria from Section III. Finally, this chapter ends with an identification of research gaps and proposal for future work.

A. COMPARISON & ASSESSMENT

Section IV has shown that there is a clear need for improving the state-of-the-art in secure provisioning for OPC UA. This is because the current OPC UA standard only provides a solution for securely managing keys and certificates during operation, but leaves establishing initial trust relationships between OPC UA endpoints undefined. In the best case, this leads to manufacturers recommending a TOFU provisioning for their OPC UA products. TOFU, however, relies on the strong assumption that an adversary cannot interfere with the provisioning process. If this assumption is unmet, an adversary is able to manipulate the provisioning and bypass any security mechanisms provided by OPC UA. Yet, potentially more severe, the rest of the manufactures leave the provisioning process undefined, instead of proposing TOFU. This leads to engineers being overwhelmed by securely provisioning and configuring their OPC UA devices. Thus, it is no surprise that 92% of Internet-facing OPC UA devices have shown to be configured deficiently, e.g., with disabled security functionality, deprecated cryptographic primitives, or certificate reuse [5].

Fortunately, Section V demonstrated that there are emerging solutions both from standardization efforts as well as academia that aim to improve the status quo. They offer not only a well-defined provisioning process, but also an authentication of OPC UA devices during provisioning. However, this increase in security comes with additional requirements on devices and infrastructures. In fact, the solutions demand secure hardware or an out-of-band communication channel on devices, as well as a PKI, a GDS, or a provisioning device as additional infrastructure. On the upside, the usability, in particular, the manual effort and degree of automation, does not suffer. This is surprising, as security solutions typically entail a tradeoff between usability and the level of protection. Nevertheless, there is still room for improvement regarding the provided security properties. This is because the emerging solutions either provide only an implicit device authentication or no operator authentication.

Section V showed that solutions offering a mutual authentication during provisioning can be found in IoT standards and products. To this end, the vendor, e.g., a manufacturer, maintains a dedicated provisioning service that authenticates and authorizes the user, e.g., an operator, for provisioning an IoT device. Since IoT solutions were not designed with

OPC UA and industrial applicability in mind, much effort is needed to use them for provisioning OPC UA devices. First, an IoT provisioning solution would have to be adapted to the provisioning objectives of OPC UA (see Section II-B). In addition, IoT solutions demand Internet access during the provisioning process, which prevents their usage in many industrial applications. This is because safety-critical systems have for security and infrastructure reasons typically no access to the Internet.

B. RESEARCH GAPS

Examining the currently emerging technologies, a trend towards higher security guarantees can be observed. This is achieved using cryptographic trust anchors, i.e., keys and certificates, that are exchanged between involved parties prior to the actual provisioning process. This means that the trust establishment is chronologically separated from the actual highly automated provisioning of devices. How exactly the required trust between manufacturers, vendors, integrators, and operators is arranged upfront is outside the scope of existing solutions. Thus, additional research is needed to define methods and processes for establishing and managing trust between the different actors in the supply chain, prior to applying the technical provisioning solution.

Another topic for future research are transitional solutions towards the presented emerging provisioning standards. Due to high requirements on devices and infrastructures, the buy-in for implementing an emerging provisioning standard is considerable. This barrier of adopting novel provisioning standards could be diminished by transitional solutions that enable current installations to be incrementally upgraded to the extensive new requirements. Such incremental solutions would propel the market to quickly adopt emerging provisioning standards, which is urgently needed from a security but also usability perspective.

VIII. CONCLUSION

This article presented an investigation of contemporary and emerging secure device provisioning solutions for OPC UA, including an outlook into the future, based on solutions from the IoT domain. The provisioning solutions were evaluated based on a set of fourteen criteria, covering device requirements, infrastructure requirements, usability, and security properties. The evaluation shows that state-of-the-art secure provisioning solutions for OPC UA offer an insufficient level of security that allow a network adversary to tamper with the provisioning and undermine the security guarantees of OPC UA. In addition, provisioning is often overly complex and vaguely described in product manuals, which leads to OPC UA deployments with disabled security functionality or security misconfigurations. Yet, emerging and future provisioning solutions offer more clarity and a higher level of security. On the other hand, they also involve higher requirements on devices and infrastructures. For manufacturers, vendors, integrators, and operators this entails additional effort and costs, as they become more and more involved in the provisioning process. Therefore, especially transitional solutions towards upcoming OPC UA secure provisioning standards are an important direction for future research. Such transitional solutions could foster the adoption of emerging provisioning solutions that offer the urgently needed level of security and usability.

REFERENCES

[1] H. Lasi, P. Fettke, H. G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Bus. Inf. Syst. Eng.*, vol. 6, no. 4, pp. 239–242, 2014.
[2] *International Electrotechnical Commission (IEC)*, document IEC TR 62541,2016.
[3] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems security—A survey," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1802–1831, Dec. 2017.
[4] *OPC UA Security Analysis*, Federal Office for Information Security (BSI), Bonn, Germany, Apr. 2017.
[5] M. Dahlmanns, J. Lohmöller, I. B. Fink, J. Pennekamp, K. Wehrle, and M. Henze, "Easing the conscience with OPC UA: An internet-wide study on insecure deployments," in *Proc. ACM Internet Meas. Conf.* New York, NY, USA: ACM, 2020, pp. 101–110.
[6] B&R Industrial Automation GmbH. (2020). *X20 System—User's Manual 3.60*. [Online]. Available: https://www.br-automation.com/en/products/plc-systems/x20-system/documentation/max20-eng/
[7] WAGO Kontakttechnik GmbH & Co. KG. (2020). *WAGO I/O System 750-8202 PFC200 2ETH RS—Version 3.8.0*. [Online]. Available: https://www.wago.com/global/plcs-%E2%80%93-controllers/controller-pfc200/p/750-8202
[8] PHOENIX CONTACT GmbH & Co. KG. (2019). *PLCnext Technology—User Manual 2020.0 LTS*. [Online]. Available: Available: https://www.phoenixcontact.com/online/portal/us?uri=pxc-oc-itemdetail:pid=2404267
[9] Siemens AG—Digital Industries. (2019). *SIMATIC S7-1500, ET200MP, ET200SP, ET200AL, ET200pro Communication Function Manual*. [Online]. Available: https://support.industry.siemens.com/cs/document/109742691/documentation-for-the-automation-system-s7-1500-and-the-et-200mp-distributed-i-o-system?dti=0&lc=en-WW
[10] Unified Automation GmbH. (2021). *UaGateway Documentation—V1.5.6*. Accessed: May 20, 2021. [Online]. Available: https://documentation.unified-automation.com/uagateway/1.5.6/html/index.html
[11] Softing Industrial Automation GmbH. (2019). *User Manual dataFEED Gateway—Version: EN_171_190601*. [Online]. Available: https://industrial.softing.com/products/gateways/gateways-for-access-of-controller-data/uagate-si.html
[12] Hans Turck GmbH & Co. KG. 2019. *TBEN-L.-4RFID-8DXP-OPC-UA Compact RFID Interface—Instructions for Use V01.00*. [Online]. Available: https://www.turck.us/en/product/6814126
[13] Microsoft. *Azure IoT Hub Device Provisioning (DPS) Documentation*. Accessed: Jul. 7, 2021. [Online]. Available: https://docs.microsoft.com/en-us/azure/iot-dps/
[14] Amazon. *Device Provisioning—AWS IoT Core*. Accessed: Jul. 7, 2021. [Online]. Available: https://docs.aws.amazon.com/iot/latest/developerguide/iot-provision.html
[15] Google. *Verifying Device Credentials | Cloud IoT Core Documentation*. Accessed: Jul. 7, 2021. https://cloud.google.com/iot/docs/how-tos/credentials/verifying-credentials
[16] *OPC Unified Architecture Specification Part 12: Discovery and Global Services*, OPC Found., Scottsdale, AZ, USA, 2018.
[17] K. Watsen, M. Richardson, M. Pritikin, and T. Eckert, "A voucher artifact for bootstrapping protocols," Internet Requests for Comments, RFC Editor, IETF, Wilmington, DE, USA, Tech. Rep. 8366, May 2018.
[18] K. Watsen, I. Farrer, and M. Abrahamsson, "Secure zero touch provisioning (SZTP)," Internet Requests for Comments, RFC Editor, IETF, Wilmington, DE, USA, Tech. Rep. 8572, Apr. 2019.
[19] M. Pritikin, M. Richardson, T. Eckert, M. Behringer, and K. Watsen, "Bootstrapping remote secure key infrastructures (BRSKI)," Internet Requests for Comments, RFC Editor, Tech. Rep. 8995, May 2021.
[20] M. Richardson, "6tisch zero-touch secure join protocol," Internet Requests for Comments, RFC Editor, Tech. Rep. draft-ietf-6tisch-dtsecurity-zerotouch-join-04, May 2021.
[21] *Device Provisioning Protocol Specification*, Wi-Fi Alliance, Austin, TX, USA, Version 1.1, 2018.

[22] A. Fernbach and W. Kastner, "Certificate management in opc ua applications: An evaluation of different trust models," in *Proc. ETFA*. Piscataway, NJ, USA: IEEE, Sep. 2012, pp. 1–6.
[23] G. Karthikeyan and S. Heiss, "PKI and user access rights management for OPC UA based applications," in *Proc. IEEE 23rd Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2018, pp. 251–257.
[24] P. Birnstill, C. Haas, D. Hassler, and J. Beyerer, "Introducing remote attestation and hardware-based cryptography to OPC UA," in *Proc. ETFA*. New York, NY, USA: IEEE, Sep. 2017, pp. 1–8.
[25] D. Bienhaus, L. Jäger, R. Rieke, and C. Krauß, "Gateway for industrial cyber-physical systems with hardware-based trust anchors," in *Intelligent Distributed Computing XIII*, (Studies in Computational Intelligence), vol. 868, I. Kotenko, C. Badica, V. Desnitsky, D. El Baz, and M. Ivanović, Eds. Cham, Switzerland: Springer, 2020, pp. 521–528.
[26] S. Nepal, J. Zic, D. Liu, and J. Jang, "A mobile and portable trusted computing platform," *EURASIP J. Wireless Commun. Netw.*, vol. 2011, no. 1, pp. 1–19, Dec. 2011.
[27] M. Blume, N. Koch, J. Imtiaz, H. Flatt, J. Jasperneite, M. Schleipen, O. Sauer, and S. Dosch, "An OPC UA based approach for dynamic-configuration of security credentials and integrating a vendor independent digital product memory," in *Proc. Jahreskolloquium Kommunikation der Automation (KommA)*, Lemgo, Germany, 2014, pp. 10–20.
[28] D. Meier, F. Patzer, M. Drexler, and J. Beyerer, "Portable trust anchor for opc ua using auto-configuration," in *Proc. 25th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, vol. 1, Sep. 2020, pp. 270–277.
[29] *OPC Unified Architecture Specification Part 2: Security Model*, OPC Found., Scottsdale, AZ, USA, 2015.
[30] *OPC Unified Architecture Specification Part 14: PubSub*, OPC Found., Scottsdale, AZ, USA, 2018.
[31] *IEEE 802.1AR: Secure Device Identity*, Standard IEEE Std 802.1AR-2018, Institute of Electrical and Electronics Engineers (IEEE), 2018.
[32] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. IT-29, no. 2, pp. 198–208, Mar. 1983.
[33] *OPC Unified Architecture Specification Part 21: Device Provisioning*, OPC Found., Scottsdale, AZ, USA, 2020.
[34] S. Symington, W. Polk, and M. Souppaya, "Trusted Internet of Things (IoT) device network-layer onboarding and lifecycle management (draft)," NIST, Gaithersburg, MD, USA, Tech. Rep., Sep. 2020, doi: 10.6028/NIST.CSWP.09082020-draft.
[35] M. Bjorklund, "The yang 1.1 data modeling language," Internet Requests for Comments, RFC Editor, IETF, Wilmington, DE, USA, Tech. Rep. 7950, Aug. 2016.
[36] M. Pritikin, P. Yee, and D. Harkins, "Enrollment over secure transport," Internet Requests for Comments, RFC Editor, IETF, Wilmington, DE, USA, Tech. Rep. 7030, Oct. 2013.
[37] A. Bierman, M. Bjorklund, and K. Watsen, "RestConf protocol," Internet Requests for Comments, RFC Editor, IETF, Wilmington, DE, USA, Tech. Rep. 8040, Jan. 2017.
[38] R. Enns, M. Bjorklund, J. Schoenwaelder, and A. Bierman, "Network configuration protocol (NetConf)," Internet Requests for Comments, RFC Editor, IETF, Wilmington, DE, USA, Tech. Rep. 6241, Jun. 2011. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6241.txt. http://www.rfc-editor.org/rfc/rfc6241.txt
[39] M. Richardson, P. van der Stok, P. Kampanakis, and E. Dijk, "Constrained voucher artifacts for bootstrapping protocols," Internet Requests for Comments, RFC Editor, Tech. Rep. draft-ietf-anima-constrained-voucher-11, Jun. 2021.

FLORIAN KOHNHÄUSER received the B.Sc. degree in computer science from the Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, in 2012, and the M.Sc. degree in IT security and the Ph.D. degree in computer science from the Technical University of Darmstadt (TU Darmstadt), Darmstadt, Germany, in 2014 and 2019, respectively. From 2015 to 2019, he worked as a Research Assistant and a Doctoral Candidate at the Security Engineering Group, TU Darmstadt, where he wrote his thesis on cryptographic protocols that enable verifying the software integrity of embedded systems. Since 2019, he has been working as a Scientist with ABB Corporate Research, Ladenburg, Germany. His research interests include secure protocols, systems, and applications for industrial automation systems.

DAVID MEIER received the B.Sc. and M.Sc. degrees in information system technology from the Technische Universität Darmstadt (TU Darmstadt), Darmstadt, Germany, in 2011 and 2013, respectively. Since 2014, he has been a part of the Research Group on Securely Networked Systems, Fraunhofer IOSB, Karlsruhe, where he participated in multiple projects for the Federal Ministry of Education and Research (BMBF) and European Commission. He has been involved with multiple projects in the area of industrial IT security for industry partners and public authorities. His research interests include network security, wireless networks, and security for industrial systems.

FLORIAN PATZER received the B.Sc. and M.Sc. degrees in computer science from the Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, in 2014 and 2016, respectively, where he is currently pursuing the Ph.D. degree in computer science. Since 2016, he has been a Security Researcher with the Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB), Karlsruhe. His research focusses on network-related security for industrial control systems. Moreover, his research interests include automated incident response and knowledge-based system analysis for application in intrusion detection, configuration analysis, compliance analysis, and threat intelligence.

SÖREN FINSTER received the Diploma degree in computer science from the University of Karlsruhe (now the Karlsruhe Institute of Technology, KIT), Karlsruhe, Germany, in 2008, and the Ph.D. degree in computer science from KIT, in 2014. From 2008 to 2015, he worked first as a Research Assistant and later as a Postdoctoral Researcher with the Institute of Telematics, KIT, where his research topics include privacy-aware computing, peer-to-peer protocols, and network simulation. From 2015 to 2018, he worked as a Security Architect and a Lead of the Embedded Security Development Team, Wibu-Systems AG, Karlsruhe. Since 2018, he has been working as a Research Scientist with ABB Corporate Research, Ladenburg, Germany. He wrote his Ph.D. thesis on privacy-aware smart metering, and designing and implementing protocols that allow smart metering applications without sacrificing privacy. His research interests include security for resource constrained devices, privacy-aware protocols, and security for industrial automation systems.