Secure Granular Interoperability with OPC UA

Venesa Watson[1], Jochen Sassmannshausen[2] and Karl Waedt[3]

Abstract: Open Platform Communications Unified Architecture (OPC UA) is the communication standard earmarked for future industrial automation, particularly for the Industry 4.0 (I4.0) infrastructure where it provides the key services for interoperability and built-in communication security. OPC UA defines several models for these services and has already been deployed by industrial partners in their efforts to achieve I4.0 market readiness and to provide more robust systems. Of particular interest is the security services offered by OPC UA, as they are expected to strengthen the security posture of industrial automation systems, which have so far suffered a number of sophisticated cyber-attacks. In general, cyber-attacks are more severe based on the level of access acquired by the attacker, for example, an attacker with unrestricted administrative level access can issue more powerful commands. It is safe to say then that a more stringent access control security concept can offer systems greater protection from unauthorized access. Several access control models exist, which are categorized under two headings discretionary (data owners/users set the access control rules) and non-discretionary (security administrators control the access granted to users). Here, a non-discretionary access control model, namely the attribute-based access control (ABAC) model is compared to the role-based access control (also non-discretionary) typically assumed with OPC UA, to ascertain how a more granular security structure with ABAC could provide additional security advantages for industry.

Keywords: OPC UA, I4.0, access control, cyber-attacks, ABAC

1 Introduction

Open Platform Communications Unified Architecture (OPC UA), as defined in IEC 62541, is a platform-independent communication standard that defines a communication model that can be mapped to diverse communication protocols to support interoperability between systems from different manufacturers. Initially, OPC UA specified a client-server architecture to describe the data exchanges, however, a later publisher-subscriber architecture was introduced to extend the OPC UA protocol [Ho18] [So16]. The client-server architecture was thought of as not ideal for supporting the requirements for implementing I4.0, particularly where the integration of Time Sensitive Networking (TSN) with OPC UA for the underlying time-critical transmission, was concerned. The OPC UA *PubSub* specification (IEC 62541-14) addressed these concerns, extending the use cases for OPC UA implementation [Ho18] [So16]. OPC UA is also described as provid-

[1] Framatome GmbH, Erlangen, Germany/University of Siegen, Faculty of Science and Engineering, Adolf-Reichwein-Straße 2, Siegen, 57068, venesa.watson@ framatome.com /venesa.watson@uni-siegen.de
[2] University of Siegen, Faculty of Science and Engineering, Adolf-Reichwein-Straße 2, Siegen, 57068, jochen.sassmannshausen@uni-siegen.de
[3]Framatome GmBH, Erlangen, Germany, karl.waedt@framatome.com

doi:10.18420/inf2019_ws34

ing robust, secure end-to-end communication, where the security objectives addressed include authentication, authorization, confidentiality, integrity, auditability and availability [IE16]. The OPC UA security mechanisms that fulfil these objectives are outlined in the standard document as follows:

- IEC 62451-2 specifies the security model of OPC UA – defines an architecture that ensures application and transport security;
- IEC 62541-4 specifies the data transfer security – defines mechanisms for OPC UA clients and servers to establish a secure channel for data transfer; and
- IEC 62541-7 specifies the security profiles – defines four security policies suitable for different levels of security to be implemented where viable.

In general, some aspects of the security mechanisms of OPC UA are presumed as too burdensome for wide-scale deployment in industrial environments [IE16b] [WH17]. The integration of OPC UA with TSN is expected to offset possible negative impacts; however, for highest safety and performance, some of the OPC UA security mechanisms may be neglected. For example, in most cases, the key security objectives for industry are integrity and availability, therefore controls for confidentiality could be excluded. Still, it is imperative that the remaining controls are sufficient to meet the level of security for the processes and data concerned. Stringent access control security is one measure by which a strong level of security can be attained without necessitating the use of cryptographic measures for confidentiality. Furthermore, it has the potential to significantly reduce the capabilities of attackers (both insiders and outsiders), addressing a critical need in industry. As observed with recent high-profile attacks such as Stuxnet and Black Energy, attackers were able to elevate their access resulting in serious losses for critical infrastructures. It is safe to say then that an insider with as much malicious intent could have surpassed the impacts of these attacks. Therefore, additional efforts to provide a more granular access control will work to protect systems from unauthorized access and increase security in critical environments.

This paper looks at how OPC UA handles access control, with the intention to address its limitation according to the advantages presented with attribute-based access control (ABAC). A general overview of access control is discussed in section 2, with a detailed description of access control in OPC UA provided in section 3. In section 4, an OPC UA-ABAC concept is presented, and section 5 summarizes the arguments put forward in this paper.

2 Access Control Overview

Threat reports such as [Vo15] and [BS16] list insider attacks are one of the major threats to industrial automation and control systems (IACS). Insiders already have access to critical infrastructure and have privileges to perform certain actions within a system. Communication security is a mandatory pre-requirement for access control as it prevents unauthorized access by external attackers and provides mechanisms to authenticate the

origin of exchanged data and commands. Access control relies on the correct authentication of origin of commands and evaluates (based on a defined set of access control policies) if the accessing subject is allowed to perform the requested action. The following sections will give an overview of the most important principles and access control models.

2.1 Access Control Principles

An important principle for access control system is the *Least Privilege Principle.* This principle means that no entities of the system have more privileges than required for the assigned tasks. This limits the potential damage to the system caused by compromised components. The least privilege principle can be enforced using different access control models. Another principle is the principle of *Separation of Duties.* This principle means that privileges of an entity must not cause conflicts of interests. For example, an entity should not be allowed to configure the security settings for a part of the system on which the same entity would act as an operator as this would allow the entity to exceed their privileges. There are two different kinds of separation of duty: Static and dynamic separation of duties. In a dynamic separation of duty scenario, access privileges can change dynamically depending on previous actions. The privilege assignment processes of the access control models must ensure the principle of Separation of Duty.

2.2 Access Control Models

The access control model covers the aspects of how access control policies are modelled, how access privileges are represented and how privileges can be assigned to entities. An overview of different access control models is given in [BU14]. Access control matrices are the simplest access control model, where each user and each object is represented as column resp. row and the assigned privileges are given in the corresponding cell. An object-centric approach supported by many file systems stores all privileges of users as property of the objects. Both systems are not very scalable as the size of the access control matrices grows very fast with increasing user/object number.

2.2.1 Role-based Access Control (RBAC)

A well-known and widely adapted access control model is role-based access control (RBAC), which was introduced by [FK92] in 1992. In RBAC systems, there is a set of roles with associated permissions, i.e. allowed operations on certain objects. Accessing entities (e.g. users, processes, etc.) can hold one or more roles and thereby the associated privileges. The least privilege principle and the separation of duties is ensured by the role design/role-subject assignment. It is possible that a subject can have several roles which cannot be active at the same time. The role-engineering process is the complex part of RBAC systems [RC10] [CW13]. RBAC is scalable, as the number of roles is independent of the number of subjects. However, RBAC is not very flexible and with a growing number of different tasks and the requirement of fine-grained access control,

RBAC might require a large number of roles in order to ensure the least-privilege principle. Another advantage of RBAC is the auditability: It is easy to determine the maximum privileges of subjects that hold a certain set of roles.

2.2.2 Attribute-based Access Control (ABAC)

Attribute-based access control (ABAC) is described in [HF14]. ABAC is an access control model where subjects and objects can hold different attributes, which can have different values. In addition, there can environment attributes that represent the current state of the system. The types and possible values of attributes are dependent on the application scenario. Access control decisions are made based on policies that evaluate all present attributes within a request context. ABAC is fine-grained and scalable, as there are no limitations on subject and object attribute assignment. An additional advantage of ABAC is that there is no cumbersome role-engineering process. Disadvantages of ABAC are policies that can grow complex with an increasing number of different attributes and a limited auditability: It can be difficult to determine the maximum number of permissions of an entity based on the set of assigned attributes and attribute values. Core parts of ABAC models are attribute assignment and policy evaluation. There are different technologies for policy definition and evaluation, a widely used system is the eXtensible Access Control Markup Language (XACML) [Oa04], an example for newer approaches is the so-called “Policy Machine” [FA11].

2.3 Hybrid RBAC-ABAC models

There are approaches that combine RBAC and ABAC in order to benefit from the advantages of both approaches and to overcome the disadvantages of both systems [RC10] [CW13]. This approach bases on RBAC with a fixed number of roles with associated permissions and additional attributes used to implement constraints. This way, the role-permission assignment can be coarse-grained with a limited set of roles and subject- and object-specific attributes restrict the privileges which were assigned through roles. For example, there can be a single role “Operator”, but the holder of this role might have additional attributes such as the “Area of Responsibility” or a clearance, which limits the access to only certain parts of the system.

2.4 Standards and Guidelines – Power Systems and IACS focus

Several standards and guidelines cover access control in IACS and Power Systems in particular. For smart energy grids, there are the NIST NISTIR 7628 Guidelines on Smart Grid Cyber Security [GR10] that categorize different application scenarios in the smart grid and give guidance on security requirements, goals and mechanisms. A similar guideline for IACS is the NIST Guideline for Industrial control Systems security [SP15]. Another detailed guideline is the NIST SP 800-53 Guideline for Security and Privacy in Federal Information Systems and Organizations [IN15]. Guidelines like the NIST Framework for improving critical infrastructure security [Se18] give a summary of re-

quirements and refer to guidelines like [IN15] or standards like IEC 62443 for additional information and explanations. IEC 62443 is an important standard for IACS security that covers a broad spectrum of security aspects, such as Information Security Management Systems (ISMS), security throughout the whole product lifecycle and security levels. IEC 62443-3-3 [IE13] introduces security measures that must be implemented in order to achieve certain security levels. Important security aspects of access control systems are account management, privilege assignment and enforcement, authentication, separation of duties, and the least privilege principle.

3 Security in OPC UA

The design of the security mechanisms of OPC UA followed the detailed analysis of a myriad of attacks as described in part 2 of the standard [IE16b]. The resulting security mechanisms provide security at three (3) levels, namely, user security, application security and transport security, and are realized through [IE16a] [IE16b] [OP18]:

- user and application authentication on the application layer
- authorization of access rights and permissions to information
- confidentiality of data exchanges via encryption on the transport layer
- authenticity and integrity of data exchanges with digital signatures
- auditability/accountability through the logging of security-related events
- availability through limiting message size and hiding associated error codes

Access control is defined at the user security level where authorization mechanisms are used to determine access rights and permissions that are defined by the OPC UA information model and/or by the user or role of the user..

3.1 OPC UA Access Control Security

Communication occurs between the OPC UA *Servers* and *Clients*, where the Servers are the software applications that implement and offer the OPC UA Services, and the Clients are the software applications that send messages to the Servers by using the OPC UA Services [IE16a]. The OPC UA standard specifies the *AddressSpace* model, which is the set of *Objects* and related information that the OPC UA Servers make available to the Clients [IE15]. Typically, the AddressSpace model is represented as a set of *Nodes* (Objects), described by *Attributes* and interconnected by *References* (explicit relationship between Nodes) [IE15] [SS16]. The *View* represents a subset of the AddressSpace of interest to OPC UA clients [SS16].

OPC UA defines a list of eight non-extensible *NodeClasses*, which are described in terms of the Attributes and References that are instantiated when a Node is defined. Each Node is an instance of these NodeClasses [IE15]. The Attributes are data elements that describe Nodes and include: an attribute id, a name, a description, a data type and an

indicator (mandatory/optional). Attribute values can be accessed by the Clients using Read, Write, Query and Subscription/Monitored Item operations. Attribute values are not included in the AddressSpace. Again, the References are used to indicate a relationship between Nodes and can be accessed using the browsing and querying operations. References are defined as instances of *ReferenceType* Nodes and are visible in the AddressSpace [IE15].

To summarize, the AddressSpace is comprised of Nodes that represent the infrastructure (systems and their interconnections) from which the Client request data and receive events [Kn13]. Considering the AddressSpace as a tree structure, the root Node is the single-entry point through which the entire AddressSpace can be accessed [Tu16]. For OPC UA, user access to the AddressSpace is defined by role-based/user-based access control. First, the Server authenticates the user and then authorizes the user requests to access Objects in the Server. However, OPC UA does not specify authorization mechanisms, which are application or system-specific [IE16a]. The standard does, however, recommends enforcing the need-to-know and least privilege principles, advocating for the implementation of coarse-grained rules (allow/deny all actions on all data) to fine-grained rules (allow/deny specific actions on specific data) [IE16b].

Figure 1 gives an example overview of how user access restrictions on the OPC UA AddressSpace can be visualized. In this scenario, *User A* can read only a part of the address space, *User B* can read/write and *User C* cannot browse. Considering the scope for User B in a real-world application, this user would have read-write privileges for the entire system or subsystem. With this role-based access control model, the need-to-know and least privilege principles could be distorted by the uncontrolled distribution of access rights and permission resulting from unmanaged user role assignment. However, as OPC UA Clients and Server applicants may determine what data and operations are allowed, then the current security posture of OPC UA can be extended to include an ABAC model.

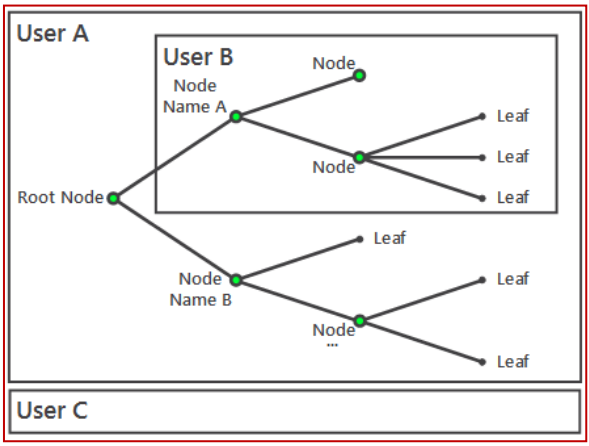

Figure 1. AddressSpace Security [In13]

4 Secure Granular Interoperability

A security analysis of the OPC UA protocol indicates that there are some weaknesses in the current user access control implementation, and further, that there are weaknesses that can be exploited after access has been granted to an authorized user, allowing an attacker a high level of control over the systems. For instance, it has been found that the security objective "Authorization" is not defined precisely enough. The recommendation is that the definition highlight that the rights must be granted according to the *need-to-know* principle [OP17]. Again, the need-to-know principle ensures access to a particular set of data is limited to those persons whose duties require such access. It is expected that the next update of part 2 of the standard should address this limitation. Further, the security analysis mentions the threat of *manipulation of access rights in the address space*. If an attacker is successful in this attack, then the attacker can read confidential data or manipulate data. It is indicated then that external countermeasures are necessary to protect against this threat. However, further refinement of user access can also limit the reach of an attacker to prevent privilege escalation. There is also the issue of the use of the "anonymous" identifier. The recommendation is that this identifier should only be used for accessing non-critical OPC UA servers (determined at the discretion of the administrator) as it does not provide any security protection. Furthermore, there is no adequate restriction of rights of the anonymous identifier, and its use prevents trustworthy auditing and accounting. The consequences are huge if an attacker can gain access with this identifier.

With ABAC, instead of considering pre-defined roles to assign rights and privileges to a subject (user or application), access is assigned considering a policy defined by a set of attributes (characteristics of subject, object and environment conditions) to provide context-aware, robust access control security. ABAC is especially advantageous for the OPC UA-supported Industrial Internet of Things (IIoT)/I4.0 infrastructures, in an effort to ensure a more secure interconnected industrial network.

4.1 OPC UA - ABAC

Figure 2 demonstrates a typical ABAC scenario, with OPC UA Concepts added. The **Subject** is typically described with attributes such as role title, affiliation (department or company), privilege (management level), user group memberships, certifications or competencies, user ID, and so forth [Ax13]. These can be retrieved from a user database, such as an LDAP server. OPC UA allows the administrator to determine and implement the preferred user database as a part of the authentication procedure. In an OPC UA environment, the subject is the OPC UA client. The **Object** is the system resource defined by attributes such as data type, security label, object ID, value and status. The **Operations** that can be executed in the Object, for example: read, write, edit, delete, copy, execute, and modify. In an OPC UA environment, the objects and operations are respectively defined in/by the AddressSpace and Services of the OPC UA Server.

Environment conditions are subject- and object-independent and concern the operational or situational context in which the subject requests access to the object. Examples can include date (time of day, day of the week, etc.), location of the subject, or the current threat level [Ax13]. Finally, the **Policy** represents the rules used to ascertain whether the request should be allowed or denied based on the attributes of the subject, object and environment conditions [Ax13]. An example ABAC rule would then be, *Operators of the Refueling Station in the Main Control Room can inspect the logs of refueling activities from the current year when the Refueling Pump is not in use*:

- Subject attributes → *Refueling Station Operators*
- Object attributes → *Refueling activities log from the current year*
- Environmental Conditions → *Main Control Room*, *Refueling Pump is not in use*
- Operations → *Read (inspect)*

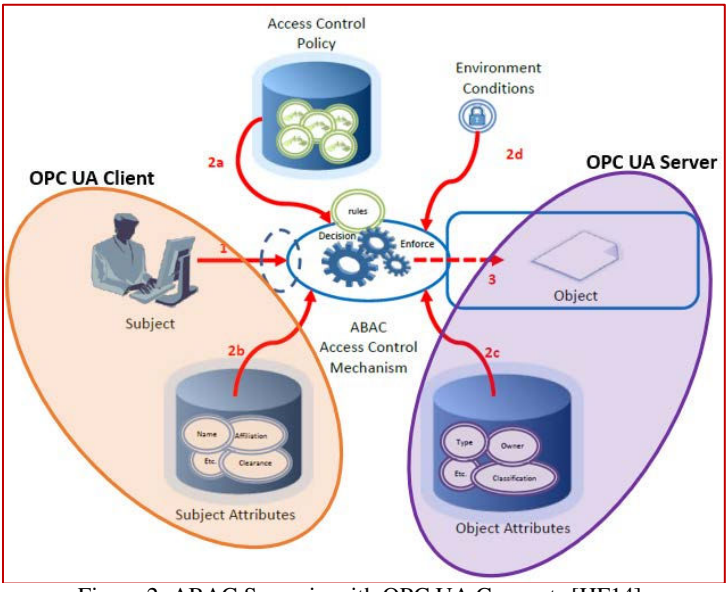

Figure 2: ABAC Scenario with OPC UA Concepts [HF14]

Within an OPC UA environment, the main concern is the object attributes – the characteristics available and how they can be applied in an ABAC rule to define access for the subjects. As aforementioned, OPC UA defines objects as Nodes within its AddressSpace. The NodeClasses are associated with OPC UA Attributes, which are data elements that describe the Nodes themselves and have values (Boolean, Byte, Double, Integer, etc.) that are accessed by the OPC UA Clients via the OPC UA Services. The standard lists 22 OPC UA Attributes that the NodeClasses use either in an optional or mandatory way [IE15]. An access control policy following the ABAC model can employ these Node Attributes in combination with the OPC UA Client attributes and environment conditions to decide on a user access request (operation). For instance, NodeClass

Attributes include *AccessLevel* (how a variable value may be accessed); *UserAccessLevel* (how a variable value may be accessed after taking a user's access rights into account); *WriteMask* (indicates which attributes are writeable) and *UserWriteMask* (indicates which attributes are writeable by the current user). An example ABAC rule could be:

- PERMIT user with attributes:
  - has role **Operator**
  - has **level 3 access**
  - has **write permission**
- To PERFORM actions with parameters:
  - **increase the value of critical turbine parameters**
  - **requires level 3 clearance**
- on DATA TYPE Turbine Speed:
  - **requires level 3 clearance**
  - has **metric cycles per second**
- if the CONDITIONS indicate that:
  - **an event notification recommends an increase in the turbine speed**

Considering the user access scope as shown in Figure 1, if the Operators (as described in the example ABAC rule above) are granted access equivalent to *User B* to the Turbine Object simply based on their role, then a malicious insider could change the critical Turbine parameters as he/she pleases. However, by employing the OPC UA NodeClasses with other attributes and conditions, the capabilities of the insider (and also that of an outsider who has assumed the privileges of such an Operator) and lessen the opportunity for these users to conduct malicious activities. As OPC UA provides a flexible infrastructure for administrators to implement the user authentication and authorization, then it is possible and recommended that a fine-grained user access control model as provided with an ABAC model be deployed. As aforementioned, even the OPC UA standard document endorses the use of fine-grained user control, as well as coarse grained control [IE16b]. Furthermore, Gartner [Tr15] indicates that "*by 2020, 70% of all businesses will use ABAC as the dominant mechanism to protect critical assets, up from 5% today (2014).*"

Caution must be taken when considering migrating to ABAC, as it is not a silver-bullet solution. ABAC can become quite complex and has some disadvantages in terms of audibility. Solutions to combine the advantages of both RBAC (auditability) and ABAC (fine-grained access control) should be explored. In this, RBAC can be used to define a fixed set of roles that grant privileges to users, whilst ABAC is used to restrict privileges of users depending on accessed resources, environment conditions and context-specific information.

5 Conclusion

OPC UA is the preferred standard for secure communication and interoperability for

I4.0. As it relates to security, the standard defines a number of models to address the following security objectives: authentication, authorization, confidentiality, integrity, auditability and availability. Particularly considering user access, which is one implementation to protect these objectives, a role-based access control model is typically assumed in OPC UA. However, RBAC does not provide the fine-grained level of security necessary to enforce a high level of security, for example, to disarm both insiders and outsiders and limit their reach for executing malicious actions. A further security analysis of the OPC UA protocol highlights vulnerabilities in its implementations that can be exploited to escalate user privileges. It is recommended that an Attribute-based Access Control (ABAC) model will provide the support to implement such a granular security for improved security, particularly for critical assets in industry. However, ABAC is not completely superior to RBAC, as it does have its own disadvantages. A combination of ABAC and RBAC to leverage their individual strengths should provide a more advantageous option for user access control security.

Acknowledgements

Some of the addressed topics are being elaborated as part of Framatome GmbH's participation in the WIPANO R&D (2018-2020) with Universität Siegen.

Bibliography

[Ax13] Axiomatics, Attribute Based Access Control (ABAC), https://www.axiomatics.com/attribute-based-access-control/, accessed: 26/04/2019.

[BS16] BSI-Veröffentlichungen zur Cybersicherheit: Industrial Control System Security - Top 10 Bedrohungen und Gegenmaßnahmen 2016, https://www.allianz-fuer-cybersicherheit.de/ACS/DE/ /downloads/BSI-CS 005.pdf?blob=publicationFile&v=4#download=1, accessed: 26/04/2019.

[BU14] Bokefode Jayant, D., Ubale Swapnaja, A., Apte Sulabha, S. and Modani Dattatray G.: Article: Analysis of DAC MAC RBAC Access Control based Models for Security, International Journal of Computer Applications, vol. 104, no. 5, pp. 6–13, October 2014, full text available.

[CA15] CAS: OPC UA Information Model Deployment, http://www.cas.internetdsl.pl/commserver/P_DowloadCenter/P_Publications/20140301EN_DeploymentInformationModel.pdf, accessed: 26/04/2019.

[CW13] Coyne, E., and Weil, T. R.: ABAC and RBAC: Scalable, Flexible, and Auditable Access Management, IT Professional, Volume 15, Issue 3, pp. 14–16, May-June 2013.

[FA11] Ferraiolo, D., Atluri, V. and Gavrila, S.: The policy machine: A novel architecture and framework for access control policy specification and enforcement, Journal of Systems Architecture, vol. 57, no. 4, pp. 412 – 424, 2011.

[Fr18] Friedl, S.: OPC Unified Architecture and its use in SeRoNet, https://static1.squarespace.com/static/51df34b1e4b08840dcfd2841/t/5b18d73903ce64e055dc1b86/1528354751257/ROSI-Spring18-UniStuttgart.pdf, accessed: 26/04/2019.

[FK92] D. Ferraiolo and R. Kuhn, "Role-based access control," In 15th NIST-NCSC National Computer Security Conference, 1992, pp. 554–563.

[GR10] T. S. G. I. P. C. S. W. Group: NISTIR 7628 Guidelines for Smart Grid Cyber Security, U.S. Department of Commerce, National Institute of Standards and Technologies, August 2010.

[Ho18] Hoppe, S.: OPC Foundation announces OPC UA PubSub release as important extension of OPC UA communication platform, https://opcfoundation.org/news/press-releases/opc-foundation-announces-opc-ua-pubsub-release-important-extension-opc-ua-communication-platform/, accessed: 26/04/2019.

[HF14] Hu, V. C., Ferraiolo, D., Kuhn, R., Schnitzer, A., Sandlin, K., Miller, R. and Scarfone, K.: NIST Special Publication 800-162: Guide to Attribute Based Access Control (ABAC) Definition and Considerations, January 2014.

[IE13] IEC: IEC 62443-3-3: Industrial communication networks - network and system security - Part 3-3: System security requirements and security levels, International Electrotechnical Commission - IEC, August 2013.

[IE15] IEC: IEC 62541-3: 2015 OPC unified architecture – Part 3: Address Space Model

[IE16a] IEC: IEC/TR 62541-1:2016 OPC unified architecture – Part 1: Overview and concepts

[IE16b] IEC: IEC/TR 62541-2:2016 OPC unified architecture – Part 2: Security Model

[In13] Inmation: Chapter 5. OPC Connectivity, https://inmation.com/wiki/index.php?title=Sysdoc/OPC_Connectivity, accessed: 26/04/2019.

[IN15] J. T. F. T. Initiative: Security and privacy controls for federal information systems and organizations, National Institute of Standards and Technology, NIST Special Publication 800-53r4, January 2015.

[Kn13] Knopp, W.: Industrial Automation via OPC UA, https://www.todaysoftmag.ro/article/567/automatizari-industriale-prin-opc-ua, accessed: 26/04/2019.

[Oa04] OASIS: eXtensible Access Control Markup Language (XACML) Version 3.0 Policy Distribution Protocol Use - cases and Requirements, OASIS Working Draft, October 2004, https://www.oasis-open.org/committees/download.php/9582/access control-xacml-3 0-distribution-rquirements-wd-01.pdf, accessed: 26/04/2019.

[OP12] OPC Foundation: OPC UA Information Modeling, https://www.automaatioseura.fi/site/assets/files/1442/opc_6_hunkar_informationmodel-4.pdf, accessed: 26/04/2019.

[OP17] OPC Foundation: OPC UA Security Analysis, https://opcfoundation.org/wp-content/uploads/2017/04/OPC_UA_security_analysis-OPC-F-Responses-2017_04_21.pdf, accessed: 26/04/2019.

[OP18] OPC Foundation: Practical Security Recommendations for building OPC UA Applications, https://opcfoundation.org/wp-content/uploads/2017/11/OPC-UA-Security-Advise-EN.pdf, accessed: 26/04/2019.

[RC10] Kuhn, D. R., Coyne, E. J. and Weil, T. R.: Adding Attributes to Role-Based Access Control, in IEEE Computer, vol. 43, no. 6, June 2010, pp. 79–81.

[Se18] Sedgewick, A.:Framework for improving critical infrastructure cybersecurity, version 1.1, National Institute of Standards and Technology, Tech. Rep., 2018.

[So16] Softing: Implementing Deterministic OPC UA Communication, industrial.softing.com/uploads/softing_downloads/OPCUAPublisherSubscriber_W_EN_160407_100.pdf, accessed: 26/04/2019.

[SP15] Stouffer, K., Pillitteri, V., Lightman, S., Abrams, M. and Hahn, A.: Guide to industrial control systems (ICS) security, NIST Special Publication 800-82r2, https://doi.org/10.6028/ nist.sp.800-82r2, accessed: 26/04/2019.

[SS16] Shin, I., Song, B. and Eom, D.: Auto-Mapping and Configuration Method of IEC 61850 Information Model Based on OPC UA, Energies, MDPI, Open Access Journal, vol. 9(11), pages 1-16, 2016.

[Tr15] TripWire: RBAC is Dead – Now What?, https://www.tripwire.com/state-of-security/security-data-protection/security-controls/rbac-is-dead-now-what/, accessed: 26/04/2019.

[Tu16] Tuovinen, T.: OPC UA Address Space Transformations, Master's Thesis, Aalto University School of Electrical Engineering, https://pdfs.semanticscholar.org/c0a2/2518584866291d9b9cf89914a18c251d36e7.pdf, accessed: 26/04/2019.

[Vo15] Vormetric: Vormetric Insider Threat Report, http:// enterprise-encryption.vormetric.com/rs/vormetric/images/CW_GlobalReport_2015_Insider threat Vormetric_Single_Pages_010915.pdf, accessed: 26/04/2019.

[WH17] Wei, H., Huang, Q., Li, S., Wang, P and Zhang, S.: The consideration of OPC UA security in constrained environments draft-wei-ace-opc-ua-security-00, accessed: 26/04/2019.