An Attribute Certificate Management System for Attribute-based Access Control

Christoph Ruland
Faculty of Science and Technology
University of Siegen
Siegen/Germany
Christoph.Ruland@uni-siegen.de

Jochen Sassmannshausen
Faculty of Science and Technology
University of Siegen
Siegen/Germany
Jochen.Sassmannshausen@uni-siegen.de

Jyoti Pragyan Satpathy
Faculty of Science and Technology
University of Siegen
Siegen/Germany
Jyoti.Satpathy@student.uni-siegen.de

***Abstract*—This paper focuses on attribute-based access control (ABAC) in distributed automation and control systems. ABAC policies execute authorization decisions based on user information, object information and environment conditions. The proposed security system uses attribute certificates to represent both subject and object attributes and an LDAP server to store and distribute attribute certificates. This approach adapts concepts of credential management for subjects and uses the same mechanism for both subject and object management. An attribute management system provides an interface to edit and view both subject and object information. Certificate Revocation Lists (CRLs) exist for both types of attribute certificates. The entities of the distributed access control system implement synchronization mechanisms to keep local information up-to-date.**

Keywords: ABAC, Attribute Certificates, Access Control, Smart Grids, LDAP

I. Introduction

Systems like Industrial Automation and Control Systems (IACS) or Intelligent Energy Systems (Smart Grids) are evolving into highly interconnected and ”smart” systems. The increased degree of interconnection between different systems changes the ”traditional” topology of IACS and Smart Grids that had a clear separation between IT- and OT-Networks (Information Technology and Operation Technology). The increased connectivity enables a growing number of entities (users and automated systems) to have access to OT equipment. Communication security such as encryption and authentication of data origin is mandatory and protects against external attacks such as eavesdropping or data manipulation. Access control has to rely on authentication and proof of origin, supported on the application layer (more secure) or communication security, if the same level of trust is given, and examines, if the access rights are sufficient, ie. ”Who is allowed to perform which actions with which values?”. Threat reports such as [1] and [2] show that internal attackers are one of the major threads to IT and OT systems. Fine-grained Access Control policies aim to restrict privileges of single entities to the absolute minimum required (Need-to-know-principle), which reduces potential impact of internal attacks. An important part of all access control system are management systems that provide possibilities to adjust policies and security information as well as distribution of these information to the components of the system that are responsible for policy evaluation and enforcement of authorization decisions. This paper focuses on attribute-based access control and the management of subject and object attributes. A proof-of-concept of the proposed system is implemented for automation and control systems that use the IEC 61850 standard for communication and data modeling. IEC 61850 uses a tree-structured data model which fits very well to the hierarchical data organization of LDAP. The rest of this paper is organized as follows: Section II gives an overview about ABAC and other used techniques like attribute certificates and XACML. Section III presents the proposed system, section IV explains implementation details and section V will conclude the paper.

II. Background

A. Attribute-based Access Control

One of the most widely used access control schemes is Role-based Access Control which was proposed in the early 90s [3]. RBAC uses roles that are associated with users and thereby grant permissions to the holders of a role. Attribute-based access control (ABAC) is newer than RBAC and uses arbitrary attributes associated with subjects, resources and the environment state to perform access control decisions [4]. ABAC is described in [5]. The main difference to RBAC is the higher flexibility due to a richer set of information available for the policy design. However, both RBAC and ABAC have their individual advantages and disadvantages. The process of role engineering is the cumbersome part of RBAC, since the set of roles is limited and it has to be carefully decided, which roles should be implemented and which permissions are granted by which role. The advantage of RBAC is its simple structure once the roles are designed. ABAC does not have a role engineering process, but policy design is more complicated and it is harder to determine the maximum rights of a subject from its set of attributes. There are approaches that aim to combine RBAC and ABAC to overcome the lack of flexibility of RBAC with additional attribute-based policies [6] [7]. RBAC and ABAC have some similarities. From an ABAC point of view, the role can be seen as special subject attribute. From a RBAC point of view, a static combination of different subject attributes can be seen as equivalent of a role in an ABAC environment.

B. Access Control in Automation and Control Systems

This section aims to give an overview of related work on access control in distributed automation and control systems with similar approaches regarding target systems, access control techniques or attribute-distribution mechanisms. [8] gives an overview of different security standards and IEC 62351 in particular, as it covers role-based access control. IEC 62351-8 describes different approaches of how user credentials can be represented and contains a profile which uses attribute certificates to store additional user information, such as the role [9]. Other solutions like the one presented in [10] also use attribute certificates to store user information, but define an own format for attribute certificates which is different from the one specified in RFC 5755 [11]. They use the push model to provide credentials to the access control system, which is also supported by IEC 62351-8. Lee et. al [12] focus on RBAC for IEC 61850 and use XACML for policy description, but do not focus on methods to distribute user credentials to the access control system. Other aspects like object attributes are not considered since the focus is on RRBAC only. Other approaches on attribute-based access control are presented in [13] and [14], where the authors of [14] use Attribute-based access control in industry systems and XACML as policy description language. They use an SQL-server to represent object attributes, which cannot provide the same security properties as given by attribute certificates. [13] focuses more on the system architecture and policy distribution in distributed control systems. They distinguish between two possibilities to introduce access control: Either directly integrated into the end systems or as bump-in-the-wire device to protect legacy devices. An implmentation of a access control system for Smart Grids which can operate either as firewall or as integrated module is demonstrated in [15]. The later presented security solution also supports these two scenarios. [16] describes additional access control aspects such as situation awareness, which can be seen as environment conditions in the context of ABAC.

C. XACML

XACML stands for *eXtensible Access Control Markup Language* and is specified as version 3 by OASIS [17]. XACML specifies a XML-based format for the description of access control policies. The structure of these policies is hierarchical, there are policy sets, policies and rules. The rules are part of policies which are accumulated into a policy set. A rule contains an element ”target” and optionally an element ”condition”. Both target and condition contain a boolean/logic expression that evaluates to true when the rule is applicable. A rule can evaluate to *permit*, *deny*, *not applicable* (target and condition do not evaluate to true) and *indeterminate* (missing attributes). Rule combining algorithms and policy combining algorithms determine how to calculate the overall access control decision, since the results of single rules may be different, but for each access request there has to be an overall decision which is either *permit* or *deny*. The expressions used in the policies can use all attributes which are provided within an XACML request. There can be different types of attributes, such as integers, real numbers, strings or sets of attributes.

D. Attribute Certificates

The standard X.509 defines a structure for certificates that belong to an identity and contains, inter alia, a public key, a period of validity, and are signed by a certificate authority. A X.509 certificate confirms that a public key belongs to an entity that claims to be in possession of the corresponding private key. These certificates typically have a lifetime of over 2 years. RFC 5755 [11] defines a profile for attribute certificates (AC) that are similar to X.509 certificates, but do not contain a public key. Instead, an AC contains information about its holder and additional attributes of the holder. Figure 1 shows the difference between X.509 certificates and attribute certificates. The advantage of attribute certificates is the independence of public key certificates, which allows the issuing of new attribute certificates without the need to change the identity certificate. Attribute certificates have a much shorter lifetime as public key certificates (PKCs), because the content of an PKC is static and the content of an AC is more dynamic and subject to frequent changes.

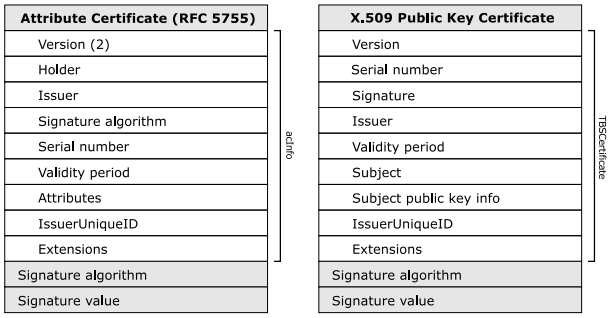

Fig. 1. Structure of Attribute Certificates according to RFC 5755 and X.509 identity certificates

E. IEC 61850

IEC 61850 is a comprehensive standard that covers data modelling communication within substations of the electrical power grid. There are several extensions that extend the original scope of IEC 61850 to cover additional fields of application such as distributed energy systems, battery systems or hydro-power plants. The flexibility of IEC 61850 makes this standard to an important standard of the future electrical grid. Important parts of IEC 61850 are the parts 7-1, 7-2, 7-3, 7-4 as they specify the data model and communication services. Extensions like 7-410 and 7-420 introduce data model extensions for domain-specific applications such as hydro-power plants or distributed energy resources [18],[19]. IEC 61850 defines a hierarchical data model with predefined ”building blocks” that are used to compose the data model of a physical device. Each physical device is represented by a virtual ”logical device” (LD), which contains ”logical nodes” (LN). LNs are defined in IEC 61850-7-4 [20] and represent a certain functionality. For example, the LN *TTMP*

represents a temperature sensor. Part 7-3 defines data objects (DO) which are part of LNs [21]. DOs are composed of Data Attributes (DA), which can also be composed of one or more data attributes. The definition of the data model results in a tree-structured data model, where each element can be identified by a path of the format *(LD Name)/(LN Name).(DO Name).(DA Name).[FC]*. The element FC is called functional constraint and can be seen as the type of an attribute. For example, the FC "MX" refers to a measured value which cannot be modified by an accessing client. IEC 61850-7-2 [22] defines the so-called *Abstract Communication Service Interface (ACSI)*, which comprises the service elements used between client and server for command and data exchange. The ACSI is protocol-independent and must be mapped to a specific communication protocol in order to enable client-server-communication. A mapping to the widely used MMS-Protocol (ISO 9506) is specified in IEC 61850-8-1 [23]. The mapping includes a conversation of ACSI-specific object identifiers to MMS-specific identifiers.

III. Proposed Solution

A. Architecture

1) Components: Figure 2 shows the overall architecture of the proposed access control solution. The left part shows the Attribute Certificate Management System, which issues, deletes and revokes attribute certificates stored on the LDAP server. The management system is the only entity that is allowed to modify the content of the LDAP server, all other entities only have reading access to the server. The IEC 61850 Client (e.g. a SCADA system or a human-machine-interface) is the entity that tries to access the Server which is protected by the access control system. The Access control system consists of three basic parts, the Policy Enforcement Point (PEP), which processes incoming requests, the Policy Decision Point (PDP), which evaluates the authorization request and the Policy Information Point (PIP), which retrieves subject and object attributes stored in attribute certificates. The Environment Conditions are observed by a module that delivers additional environment attributes for request evaluation. The Access Control Components can either be implemented as part of the endsystem, or as a firewall, which protects legacy devices from unauthorized access. The terms PEP, PDP, PIP are common components in access control models, they are used in the XACML information flow model [17], but originate back to the AAA authorization framework published as RFC 2904 of the year 2000 [24].

2) User Credentials: The policy enforcement point performs user authentication and a verification of provided credentials. There are two possibilities of how the access control system can obtain the user information (also see [9]).

a) The Push Model: The user obtains its attribute certificate (e.g. from the LDAP server) and sends it to the access control system during authentication. The policy enforcement point verifies the provided credentials. This requires up-to-date certification revocation lists (CRLs) that contain information about revoked certificates. The Security solutions maintains the CRL on the LDAP-server, but the access control systems also manage local copies of the CRLs.

b) The Pull Model: The user authenticates using the public key certificate and the policy information point obtains the corresponding certificate from the LDAP server or a local copy hold in the cache. This approach is simpler for the user as it does not have to obtain the attribute certificate first.

3) Policy Information Point: The PIP connects to the LDAP server in order to retrieve attribute certificates and certificate revocation lists. The PIP also holds a local copy of attribute certificates so that the information is available when the LDAP server is not available. The local copy of attributes is also used to enable fast request processing without additional PIP-LDAP communication.

B. LDAP-Server

The information stored in an AC is often redundant. It is retrieved or read far more times than it is updated or written as compared to a database, which is consistently altered. Furthermore, directories implement a hierarchical tree structure for data storage, which perfectly suits the data model of IEC 61850. The Lightweight Directory Access Protocol (LDAP) is a message-based client server protocol used to access directory services. The LDAP Server holds information to be accessed using the LDAP protocol over TCP/IP. One or more LDAP servers jointly host the data in the form of a Directory Information Tree (DIT). Each unit of the DIT is called an entry. An entry can have multiple child entries but only one parent entry. Entries are defined by attributes and their values. Attributes represent information about an entry. The type of attributes an entry can have is determined by the object class which is assigned when the entry is created. According to the object class, there are *MUST* attributes, which are mandatory to be assigned during creation of the entry and there are *MAY* attributes which can be used if required. The object classes are listed in different Schemas, which defines rules and policies about the type of information the server can hold. RFC 4512 [25] gives a comprehensive description of the directory information model. One or more attribute values form the RDN (Relative Distinguished Name) of an entry, which is unique with respect to its siblings. The DN (Distinguished Name) is formed by concatenating the RDN of an entry with the DN of its parent. The DN of an entry is the unique reference to it inside the DIT. The naming of the base of the DIT is based on Domain Name System (DNS). It is mostly derived from the organization whose data is being stored on the LDAP Server.

C. Attribute Certificate Management

The certificate management system connects to the LDAP Server as well as to the IEC 61850 server to display the data model and the information organized on the directory. At the time the administrator requests to edit security attributes of a particular IEC 61850 server, the management system connects to the server and retrieves the data model using ACSI services like *GetLogicalDevices* or *GetLogicalNodeDirectory*.

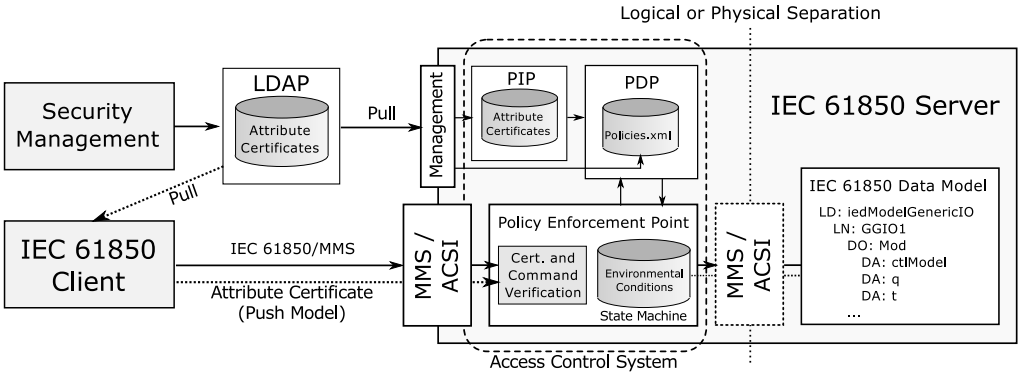

Fig. 2. The proposed overall system architecture

It is important to mention that the connection to the IEC 61850 server involves the access control system. The rights are restricted to only view the elements of the data model, but reading/writing to contents of the elements is not allowed. Further access restrictions can originate from the user that is using the UI. The management system enables adding attributes and issuing/deleting certificates for the objects and the subjects. It also retrieves attributes and other information on the AC (if present) and displays them in separate sections. The own as well as inherited attributes can be clearly distinguished on the UI. In addition, a notifications section logs all the actions performed on the UI and also displays error information.

IV. Implementation

A. Design of the LDAP DIT

Designing the LDAP DIT encompasses organizing the information available in an efficient way as well as considering the search algorithm and parameters. The ACs need to be stored as an attribute of the entries on the DIT. The most favorable choice for an AC is the userCertificate attribute, supported by the *inetorgperson* object class. However, this attribute can only be used with binary Distinguished Encoding Rules (DER) encoded digital certificates. The OpenSSL-based implementation of Attribute Certificates used in this project exports ACs in PEM format, which is easier to handle compared to the DER format. Thus, a basic string data type attribute is ascertained as the ideal choice for storing the ACs on the LDAP Server. The search algorithm is designed to examine the entire directory with the base DN as the starting point. In this case, the *commonName* or *cn* only cannot be used as the search filter as many elements might have the same name. However, the path from the root of the data model to a certain element can be an unambiguous reference. Hence, along with the name and AC, a third attribute is required for each element of the data model on the LDAP server, which stores the aforesaid path. The alternative way of searching could be to limit the scope of search to a specific DN of any entry, in which case, *cn* of an entry can be used as the parameter. Figure 3 shows the DIT of the LDAP server. The root of the DIT is branched into three broad categories, viz objects (the IEC 61850 Data model), subjects (list of users) and the CRL, with the object class *organizationalUnit* or *ou*. Based on the requirements stated above, the object class *device* is used for the objects. While *cn*, being the mandatory attribute for this object class, is used to store the name of the object, *serial number* and *description* attributes are used to store the path of the object in the tree and the AC respectively. Both these attributes permit storing the values in the string format. The object class most commonly used for the human entities in an LDAP server is *inetOrgPerson*. The mandatory attributes for this object class are *cn* and surname (*sn*). To maintain consistency, the *description* attribute is used to store the AC for a subject. It is assumed that no two subjects have the same common name as well as surname. The *cn* and *sn* are concatenated to form *displayName*, another string value storing attribute, which acts as the unique identifier in the search filter. The organization of the objects exactly emulates the tree structure of the IEC 61850 data model. Subjects are perceived as a list unlike the objects and are entered manually through the LDAP Data Interchange Format (LDIF), which is a text file that stores information about an entry and is used to add, modify or delete attributes of the entry on the DIT. The CRL is also implemented as a list, with each element either being a subject or an object. In case of objects, the *serial Number* is used as the *cn* in the CRL whereas in case of the subjects the *display Name* is used as the *cn*. The revoked certificates for each element are then numerically added to it as its child entities. The numerical value refers to the serial number of the attribute certificate (also see figure 1). The combination of a certification authority and the serial number of an attribute certificate is assumed to be unique within the system. The LDAP Server is set up on a device independent of the management system.

B. The User Interface

Figure 4 shows a part of the User interface of the management system. The left side shows the data model that was obtained from a IEC 61850 server, each element is shown with its ACSI specific identifier. Functional constrains are not considered as objects are only identified by their path. The right side shows a list of attributes that exist for the selected element. It is possible that an object does not have a certificate on the LDAP server, in which case the both information fields

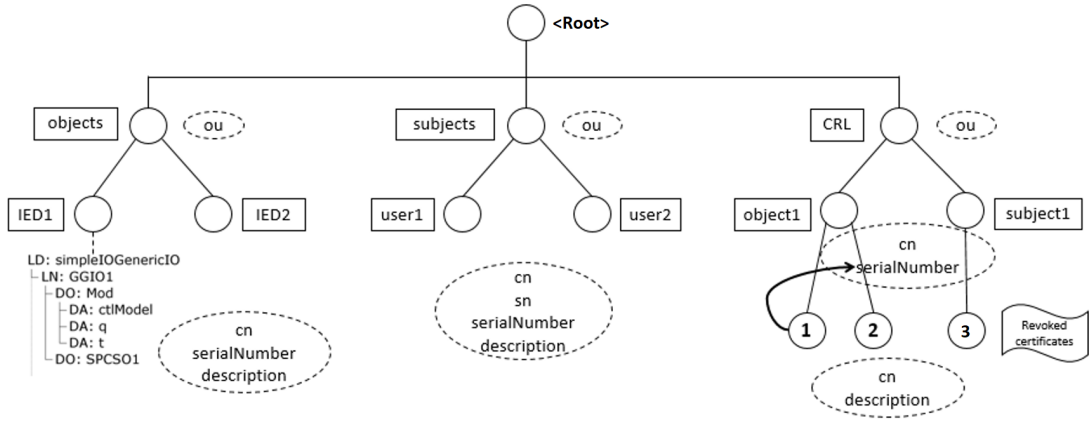

Fig. 3. The Directory Information Tree of the LDAP server.

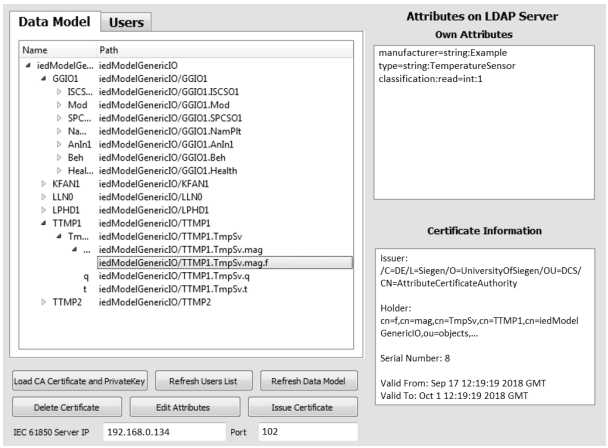

Fig. 4. Part of the UI of the Attribute Certificate Management System

would remain blank. The user interface allows to edit, add or delete attributes and the issuing of a new attribute certificate. This operation will put the old attribute certificate onto the certificate revocation list. The information field on the lower right side shows information about the attribute certificate. The element ”Issuer” refers to the distinguished name of the Attribute Certificate Authority and the element ”Holder” refers to the element of the IEC 61850 data model. The certificate stores the same distinguished name as the path of the particular object in the LDAP DIT. Other solutions are also possible, for example, the whole ACSI specific identifier could be stored in the ”Holder”-element as common name.

C. Synchronization between LDAP server and PIP

An important point regards the synchronization between PIP and LDAP server. When the Management System changes certificates, these changes have to be synchronized between PIP and LDAP-server without long delays. There are three possible synchronization mechanisms as LDAP does not directly support the PUSH-model:

a) Polling: The PIP can poll each required certificate on the LDAP server and check for changes in defined time intervals. This mechanism has the disadvantage as there may be delays and unnecessary communication between LDAP server and PIP.

b) Pull at demand: Every time the client accesses an object, the attribute certificate of the accessed objects will be obtained from the LDAP server. This solution also causes communication overhead as attribute certificates may not have changed in meantime. In addition, the request evaluation time gets longer as the PIP has to wait for the response of the LDAP-server.

c) Continuous search: The implementation uses an OpenLDAP server which supports persistent search according to RFC 4533, where changes within the search scope are detected automatically and pushed to the client. The implemented solution uses this option. The PIP updates the local certificate repository and initializes a continuous search on the whole DIT afterwards. This enables a fast synchronization between LDAP-server and PIP.

D. Request evaluation and Performance

Every time the PEP receives a request it authenticates the origin of this particular request and obtains additional information about the user from the PIP. If the user also sends its attribute certificate, the PIP checks the certificate revocation lists. In the next step, the PEP evaluates the PDU to obtain information about the accessed elements. The following example assumes that the client requests to read the value of a particular temperature sensor. The representation of object identifiers depends on the protocol and there must be a mapping between different identifiers used. Figure 5 shows different identifiers of the same object. The PEP maps the MMS-specific identifier on the ACSI identifier and passes it to the PIP. The PIP maps the ACSI-specific identifier to the LDAP-specific object identifier and obtains the attributes from the corresponding AC. Implementation details of the PIP are transparent to the PEP, which only operates with ACSI specific identifiers. At last, the PEP builds a XACML request containing all required attributes and sends it to the PDP for evaluation. The response of the PDP is either *Permit* or

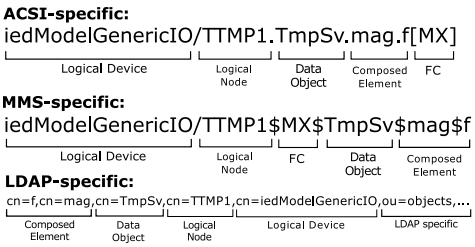

Fig. 5. Different object identifiers of the same IEC 61850 data element

Deny. Table I shows the impact of the number of attributes

TABLE I
EVALUATION TIME OF PDP FOR DIFFERENT NUMBER (N) OF BOTH SUBJECT AND OBJECT ATTRIBUTE SETS.

Platform	n=10	n=100
700 MHz ARM11, single core, 256 MB RAM	27 ms	45 ms
900 MHz ARM Cortex-A7, 4 cores, 1 GB RAM	5.3 ms	7.2 ms
1.2 GHz ARM Cortex-A53, 4 cores, 1 GB RAM	3.1 ms	3.9 ms

on policy evaluation performance and n refers to the number of both subject and object attributes provided for evaluation. All attributes have to be provided to the PDP within a request context. The PDP implements a slim policy set which enforces the *need-to-know-principle*. The assignment of subject rights on objects is performed by subject and object attributes. Every subject and object has its set of attributes and the performance test was executed with set sizes of 10 and 100 attributes. It can be seen, that large attribute sets have a notable impact on policy evaluation performance, especially on platforms with limited resources. This has to be taken into account by the attribute management system.

V. CONCLUSION

The presented security solution uses attribute certificates for both subject and object attributes, which is a difference to the existing approaches that use attribute certificates for user information only (For example, see [10] or the "Profile B" from IEC 62351-8 [9], where RBAC access tokens are presented by attribute certificates). The use of attribute certificates can be integrated using existing infrastructure used for user credential management. The management system provides a convenient interface to add, delete and edit attributes. Future work will include different profiles of pre-defined attributes that can be uploaded to the server.

REFERENCES

[1] Industrial Control System Security - Top 10 Bedrohungen und Gegenmanahmen 2016. BSI-Verffentlichungen zur Cybersicherheit, 2016. https://www.allianz-fuer-cybersicherheit.de/ACS/DE/_/downloads/BSI-CS_005.pdf?__blob=publicationFile&v=4#download=1.

[2] Vormetric Insider Threat Report, 2015. http://enterprise-encryption.vormetric.com/rs/vormetric/images/CW_GlobalReport_2015_Insider_threat_Vormetric_Single_Pages_010915.pdf.

[3] David Ferraiolo and Richard Kuhn. Role-based access control. In *In 15th NIST-NCSC National Computer Security Conference*, pages 554–563, 1992.

[4] V. C. Hu, D. R. Kuhn, D. F. Ferraiolo, and J. Voas. Attribute-based access control. *Computer*, 48(2):85–88, Feb 2015.

[5] Vincent C. Hu, David Ferraiolo, Rick Kuhn, Adam Schnitzer, Kenneth Sandlin, Robert Miller, and Karen Scarfone. NIST Special Publication 800-162: Guide to Attribute Based Access Control (ABAC) Definition and Considerations, January 2014.

[6] Edward J. Coyne Richard Kuhn and Timothy R. Weil. Adding Attributes to Role-Based Access Control. In *IEEE Computer, vol. 43, no. 6*, pages 79–81, June 2010.

[7] Ed Coyne and Timothy R. Weil. ABAC and RBAC: Scalable, Flexible, and Auditable Access Management. *IT Professional, Volume 15, Issue 3*, pages 14–16, May-June 2013.

[8] Steffen Fries, Rainer Falk, and Chaitanya Bisale. Handling Role-based Access Control in the Digital Grid. ENERGY 2017: The Seventh International Conference on Smart Grids, Green Communications and IT Energy-aware Technologies, May 2017.

[9] IEC 62351-8: Power systems management and associated information exchange - Data and Communication Security - Part 8: Role-based Access control, August 2007.

[10] Daniel Servos and Sylvia L. Osborn. Hgaa: An architecture to support hierarchical group and attribute-based access control. In *Proceedings of the Third ACM Workshop on Attribute-Based Access Control*, ABAC'18, pages 1–12, New York, NY, USA, 2018. ACM.

[11] S. Farrell, R. Housley, and S. Turner. An internet attribute certificate profile for authorization. RFC 5755, RFC Editor, January 2010.

[12] Byunghun Lee, Dae-Kyoo Kim, Hyosik Yang, and Hyuksoo Jang. Role-based access control for substation automation systems using xacml. *Information Systems*, 53:237 – 249, 2015.

[13] J. H. Huh, R. B. Bobba, T. Markham, D. M. Nicol, J. Hull, A. Chernoguzov, H. Khurana, K. Staggs, and J. Huang. Next-Generation Access Control for Distributed Control Systems. *IEEE Internet Computing, Vol. 20, Issue 5*, September 2016.

[14] Erkan Yalcinkaya, Antonio Maffei, and Mauro Onori. Application of Attribute Based Access Control Model for Industrial Control Systems. *International Journal of Computer Network and Information Security(IJCNIS), Vol.9, No.2*, pages 12–21, 2017. DOI: 10.5815/ijcnis.2017.02.02.

[15] C. Ruland and J. Sassmannshausen. Firewall for attribute-based access control in smart grids. In *IEEE International Conference on Smart Energy Grid Engineering (SEGE 2018)*, August 2018.

[16] Marc Hüffmeyer, Pascal Hirmer, Bernhard Mitschang, Ulf Schreier, and Matthias Wieland. Situation-aware access control for industrie 4.0. In Paolo Mori, Steven Furnell, and Olivier Camp, editors, *Information Systems Security and Privacy*, pages 59–83, Cham, 2018. Springer International Publishing.

[17] The eXtensible Access Control Markup Language (XACML), Version 3.0. OASIS Standard, January 2013. http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.pdf.

[18] IEC 61850-7-410: Communication networks and systems in substations - Part 7-410: Basic communication structure - Hydroelectric power plants - Communication for monitoring and control, 2012.

[19] IEC 61850-7-420: Communication networks and systems in substations - Part 7-420: Basic communication structure - Distributed energy resources logical nodes, 2009.

[20] IEC 61850-7-4: Communication networks and systems for power utility automation - Part 7-4: Basic communication structure - Compatible logical node classes and data object classes, November 2010.

[21] IEC 61850-7-3: Communication networks and systems for power utility automation - Part 7-3: Basic communication structure - Common data classes, 2010.

[22] IEC 61850-7-2: Communication networks and systems for power utility automation - Part 7-2: Basic information and communication structure - Abstract communication service interface (ACSI), April 2011.

[23] IEC 61850-8-1: Communication networks and systems for power utility automation - part 8-1: Specific communication service mapping (SCSM) - Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3, February 2012.

[24] J. Vollbrecht, P. Calhoun, S. Farrell, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege, and D. Spence. Aaa authorization framework. RFC 2904, RFC Editor, August 2000. http://www.rfc-editor.org/rfc/rfc2904.txt.

[25] K. Zeilenga. Lightweight directory access protocol (ldap): Directory information models. RFC 4512, RFC Editor, June 2006. http://www.rfc-editor.org/rfc/rfc4512.txt.