End-to-End Application Security over Intermediaries on the Example of Power System Communication

Steffen Fries, Rainer Falk
Corporate Technology
Siemens AG
Munich, Germany
e-mail: {steffen.fries|rainer.falk}@siemens.com

Abstract—**Connecting client and server applications directly via a transport connection allows the application of existing security protocols directly, as known from classical Web applications. Typically, Transport Layer Security (TLS) is applied to protect the communication link end-to-end. This approach is utilized in substation automation to protect the Transmission Control Protocol (TCP)-based communication between a substation controller and a protection relay applying mutual authentication of the end-points. If a direct communication link is not available, communication is realized over an intermediary system. Providing end-to-end security over multiple communication hops, including mutual endpoint authentication (client and a target application service) as well as integrity and confidentiality of communicated data deserves specific attention, even if the communication hops with the intermediary are protected hop-by-hop by security protocols like TLS. In power system automation, this kind of communication involving an intermediary is used with publish subscribe protocols, e.g., when integrating Decentralized Energy Resources (DER) or when integrating into the German Smart Meter Gateway architecture. This paper investigates existing solutions and specifically analyses the end-to-end security approach defined for power system automation within the International Electrotechnical Commission (IEC) and motivates broader application in session-based communication scenarios.**

Keywords—security; device authentication; end-to-end security; multi-hop security; IEC 62351 Publish/Subscribe.

I. INTRODUCTION

Security in power system communication is getting more momentum, as energy supply is part of the critical infrastructure. For critical infrastructures, the European Network and Information System (NIS) Directive [1] requires security measures to be supported by the system operator. This directive has been ratified by the European member states. Germany, for instance, has passed the Information technology (IT) Security Act already in 2015 [2], which required the definition of domain-specific security standards that have to be implemented by operators of critical infrastructures. For the power system infrastructure, the domain specific security standard is provided by ISO 27019 [3] in conjunction with the IT security catalog of the German BNetzA [4]. Both documents target communication security in terms of authentication of communicating entities in addition to integrity and confidentiality protection of the data exchange, but without specifying specific technical means in terms of protocols to be used. Security requirements for critical infrastructures are also defined outside Europe, for instance in requirements specified by NIST Cybersecurity framework [5] and specifically for the power system infrastructure by the North American Energy Reliability Council in the NERC Critical Infrastructure Protection (CIP) standards [6]. These documents pose similar requirements, which relate most often to the processes of an operator and partly to supporting technology. Common to all of the requirement documents is that additional standards/specifications are necessary to address the implementation of such requirements in components and systems, while ensuring interoperability between different vendor's products.

One standard defining specific technical requirements is provided by the framework IEC 62443 [7], describing specifically in two distinct parts technical requirements for different security levels, which relate to the strength of the considered attacker. They also refer to security of communicated data.

Besides these technical requirements, different standards and draft standards exist, addressing communication security covering standard requirements for entity authentication, integrity protection and confidentiality protection. One example for such a standard protecting specifically TCP based communication is provided by the Transport Layer Security Protocol (TLS 1.2 [8], TLS 1.3 [9]).

As analyzed in [10], the necessity to support communication over multiple hops between two entities in power system automation has been emphasized by the support of Decentralized Energy Resources (DER). Integrating DER into the current energy distribution network requires to monitor and control these DER to a similar level as centralized energy generation in power plants to keep the stability of the power network. To cope with the fact that DER are typically operated within a private operator network protected by a firewall, the standard IEC 61850-8-2 [11] defines a communication approach based on the eXtensible Messaging and Presence Protocol – XMPP [12]. Here, both sides, the DER controller, as well as the control center, connect to an intermediate server node, which facilitates the communication between both entities. In this specific case, the standard IEC 62351-4 [13] ensures that the

 ISBN: 978-1-61208-746-7

communication between the control center and the DER is secured in an end-to-end fashion. Meanwhile, this standard has been released and will be compared to other existing and meanwhile developed solutions.

The remaining part of the paper is structured as follows. Section II describes the communication overview and derives high level security requirements. These requirements are taken into consideration later in the description of the security approach taken for the integration of DER into the power system based on IEC 61850. Section III investigates a selection of existing approaches to provide end-to-end security (message-based and session-based methods). Section IV provides more insight into the actual design and application of the protocol defined in IEC 62351-4 to motivate broader application. Section V concludes the paper with an outlook.

II. COMMUNICATION ARCHITECTURE AND DERIVATION OF SECURITY REQUIREMENTS

A. Communication architecture

For the discussion of end-to-end communication, the integration of DER resources into a power system control network is taken as example, see Figure 1. The lower part of the figure shows the distributed generators (photovoltaic and wind power) that are managed by the control function shown in the upper part. All entities are connected via a communication network in which the intermediary XMPP server in the middle provides the connectivity between the control center and the DER controller. The control function may be located at a Distribution Network Operator, a virtual power plant operator, or a smart energy market operator.

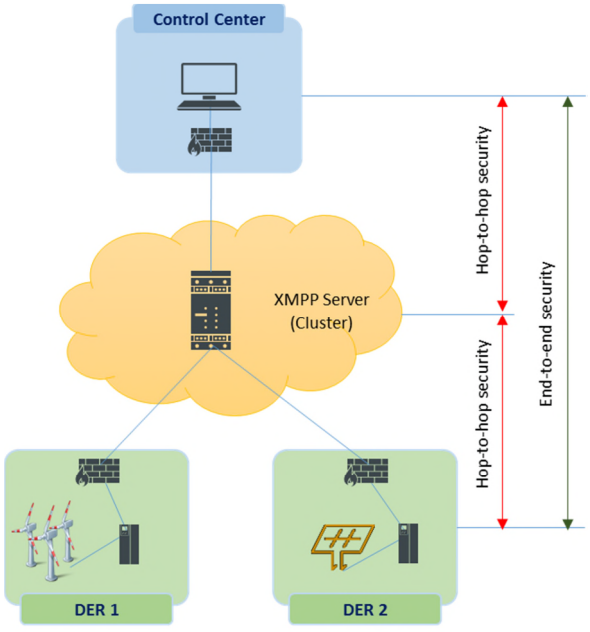

Figure 1. DER Integration based on IEC 61850 over XMPP

The data exchanged between the DER controller and the control center comprises different types of data:

- Customer data, which may be identification information, location data, consumption data or other information belonging to the DER owner.
- Control data, which may be either commands issued by the control center, or event and monitoring information from the DER controller.
- Market data, which may be tariff information provided from a marketplace via the control center or directly (not shown in Figure 1) to the DER controller.

In the context of utilizing IEC 61850 to connect DER to a control center, the communication between the DER controller and the XMPP server is secured using TLS as transport layer security protocol. The same holds for the connection between the control center and the XMPP server. Note that the XMPP server may belong to a different administrative domain and may therefore not be trusted to access the data exchanged between the DER controller and the control center. Hence, the communication relation between the DER controller and the control center is secured at application layer using IEC 62351-4, which will be analyzed in more detail in Section IV.

B. Derivation of Security Requirements

As stated in the introduction, there are different types of security requirements stemming, on one hand, from the obligation to comply with international and national regulations. On the other hand, security requirements are derived from the system architecture based on a risk-based approach. The international industrial security standard IEC 62443 [7] is a security requirements framework jointly developed by the International Electrotechnical Commission (IEC) and the International Society of Automation (ISA99) to address the need to design cybersecurity robustness and resilience into Industrial Automation and Control Systems (IACS). The standard covers both organizational and technical aspects of security over the life cycle of systems. It can be used in conjunction with ISO/IEC 27019 (the Information Security Management System (ISMS) profile for the energy domain based on ISO 27002) and with IEC 62351, providing specific security solutions. Here, the parts IEC 62443-3-3 (focus on system security requirements) and IEC 62443-4-2 (focus on component security requirements) can be used in the context of a risk-based approach, as they specify technical security requirements for four security levels, corresponding to different strengths of an attacker. For both views, system and component, foundational requirements groups have been defined. For each of the foundational requirements, several concrete technical Security Requirements (SR) and Requirement Enhancements (RE) to address a specific security level exist.

The overall approach applies to the systems and the communication connections are shown in Figure 1. In the context of this paper, the focus is placed on the communication relations only, to address the specific target of providing communication security over potentially untrusted nodes. The protection of the communication is addressed by different security requirements focusing on

 ISBN: 978-1-61208-746-7

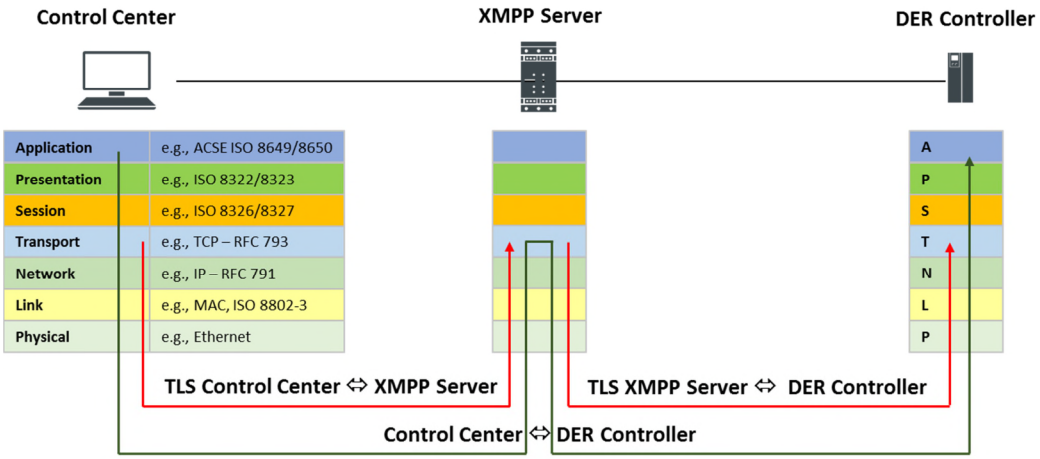

Figure 2. End-to-end-Security and hop-by-hop security according to IEC 62351-4

end-to-end security requirements and hop-to-hop security requirements. Note that the hop-to-hop security requirements contribute to the overall system security approach and may be used in conjunction with the end-to-end security. Figure 2 shows the data exchange between the control center and the DER controller via the XMPP server. The security requirements comprise specifically:

– End-to-end authentication between the DER controller and the control center to ensure identification and authentication of the communicating endpoints.

– End-to-end integrity protection to ensure that data in transit has not been tampered with (unauthorized modification) between the DER controller and the control center.

– End-to-end confidentiality protection to ensure that data in transit has not been accessed (read) in an unauthorized way by the XMPP server.

Hop-to-hop authentication between the XMPP client (DER controller, control center) and the XMPP server is used to identify and authenticate an intermediary system proxying the end-to-end communication between the DER controller and the control center.

III. SECURITY MEASURES ON APPLICATION LAYER

This section investigates a selection of existing end-to-end security approaches, which can be used to provide authentication, integrity, and confidentiality. Note that XMPP enhancements to achieve end-to-end security between the clients connected via the XMPP server have already been discussed as part of [10] and are not further discussed here. The IETF drafts discussed in this respect are already outdated and have not been updated in the last two years. Therefore, they are not considered further. In the following examples of existing standards or standards in development supporting end-to-end security on application layer, are summarized. They are distinguished into message-based approaches and session-based approaches. Message-based approaches are independent of the actual communication session and can be applied to single messages. Session-based approaches are relying on a communication connection, which comprises at least an initialization phase and a data exchange phase. Both approaches have their merits, but also certain drawbacks.

A. Message-based security

The following examples target the protection of single messages and do not rely on an established communication connection:

– IETF RFC 3923 [14] describes end-to-end signing and object encryption utilizing S/MIME to protect the messages exchanged over XMPP connections. This approach is similar to using secure email. It provides end-to-end authentication based on a digital signature and confidentiality protection based on symmetric encryption. As this approach targets message-based communication, without a communication session it will result in a higher per message overhead, as the messages are protected using symmetric encryption, while the key for the symmetric encryption is encrypted with the recipient's public key. This approach has two drawbacks. It is performance intensive due to the use of asymmetric operations and it is bound to RSA as asymmetric algorithm. Newer algorithms like ECDSA based on elliptic curves may not be used.

– W3C defined XML security may also be used to address a secure data exchange on application layer. There are two different standards available, which are already utilized to provide security: XML Signatures [15] and XML Encryption [16]. Both can be used in conjunction, ideally on XML encoded data in so-called XML elements and support the given security requirements. XML encryption allows the encryption of any type of data with symmetric and asymmetric methods. XML signature on the other side applies asymmetric methods to achieve integrity protection and

 ISBN: 978-1-61208-746-7

non-repudiation. Note that there exist adequate standards for the binary data representation.

- The IETF working group for JavaScript Object Signing and Encryption (JOSE) defined two further standards, which can be used to protect messages encoded in JavaScript Object Notation (JSON). IETF RFC 7515 [17] specifies JSON Web Signatures, while IETF RFC 7516 [18] defines JSON Web Encryption. The combination of both documents is similar to XML documents developed by W3C for specific JSON encoding.
- A further IETF standard is provided with RFC 8152 [19] defining authentication, integrity protection, and confidentiality protection for Concise Binary Object Representation (CBOR), which enhanced the data model of JSON with a binary representation. This approach allows for enveloping and encryption of arbitrary message blocks.

B. Session-based security

The following examples target the protection of communication sessions for application data exchanges. For this, it is assumed that a communication session is established between two entities during which both participants can authenticate and negotiate a set of session keys for protecting further communication. This approach has the advantage for consecutive communication to result in less overhead for the bulk data handling as part of the communication session. This is due to the fact that the combination of symmetric encryption and an additional integrity protection or the direct application of authenticated encryption has a much better performance instead of invoking asymmetric cryptography on a per packet base.

- IETF draft on Application Layer TLS [20] leverages the existence of a TLS implementation on the communicating entities. The approach utilizes the option of TLS stacks to create and process TLS records based on access to the byte buffer. Based on this, the TLS packets may be transmitted over arbitrary transport connections. This approach has the advantage that the application layer security immediately benefits from new cipher suites and cryptographic algorithm support by the underlying TLS stack. In addition, several TLS stacks allow key material export using the approach defined in IETF RFC 5705 [21] to leverage the TLS key agreement and to utilize the negotiated key in the context of other protocols.
- Signal [22] is a protocol used in messaging systems, which allows to establish a secure session based on an authenticated triple Diffie Hellman key agreement in which EdDSA signatures are employed for integrity protection during the key establishment phase. The negotiated key material is applied to protect the integrity and confidentiality of the established session based on the Double Ratchet algorithm. Note that peer authentication is not directly supported by signal.
- Off-the-Record (OTR [23]) is a further protocol used in messenger applications to ensure integrity and confidentiality. In versions 2 and 3 of the protocol, peer authentication is also supported. Here, shared keys are utilized to achieve the authentication.
- Application Layer Transport Security (ALTS [24]) has been developed by Google in 2017 and is utilized to secure Remote Procedure Calls (RPC). The protocol is defined in a similar way as TLS, consisting of a handshake protocol and a record protocol. It allows for mutual authentication and session integrity and confidentiality. Authentication is bound to an entity rather than an instance (e.g., hostname) as the approach targets mainly cloud environments. Note that there are tradeoffs to TLS described in the specification [24], which relate to privacy concerns for the handshake messages and perfect forward secrecy. Note that these properties are supported out of the box in TLS 1.3, but not in TLS 1.2 and below.

IV. END-TO-END SECURITY DESIGN IN IEC 62351-4

As described in Section II.B, the security requirements for providing an application layer end-to-end security supporting DER integration can be summarized as:

- Mutual peer authentication between the DER controller and the control center. As existing security measures described in the context of IEC 62351 always rely on authentication using X.509 certificates, being intended for authentication, too.
- Session key management between the communicating peers supporting initial key agreement providing perfect forward secrecy as well as key update.
- Integrity protection of exchanged data to ensure that data in transit has not been tampered with.
- Optionally, confidentiality protection to ensure that an intermediary cannot access the content of the data exchange.

Note that it should be possible to use either distinct algorithms for integrity and confidentiality or a combined approach (authenticated encryption).

The standard IEC 62351-4 was updated in 2018 and specifies a transport security profile and an application security profile. The application security targets the provisioning of end-to-end security, as outlined by the requirements above. The following description depicts the protocol.

A. Precondition

The involved endpoints are expected to possess a certificate and corresponding private key as well as a root certificate trusted by both sides (e.g., bound to the operator) and a common set of Diffie Hellman base parameter.

 ISBN: 978-1-61208-746-7

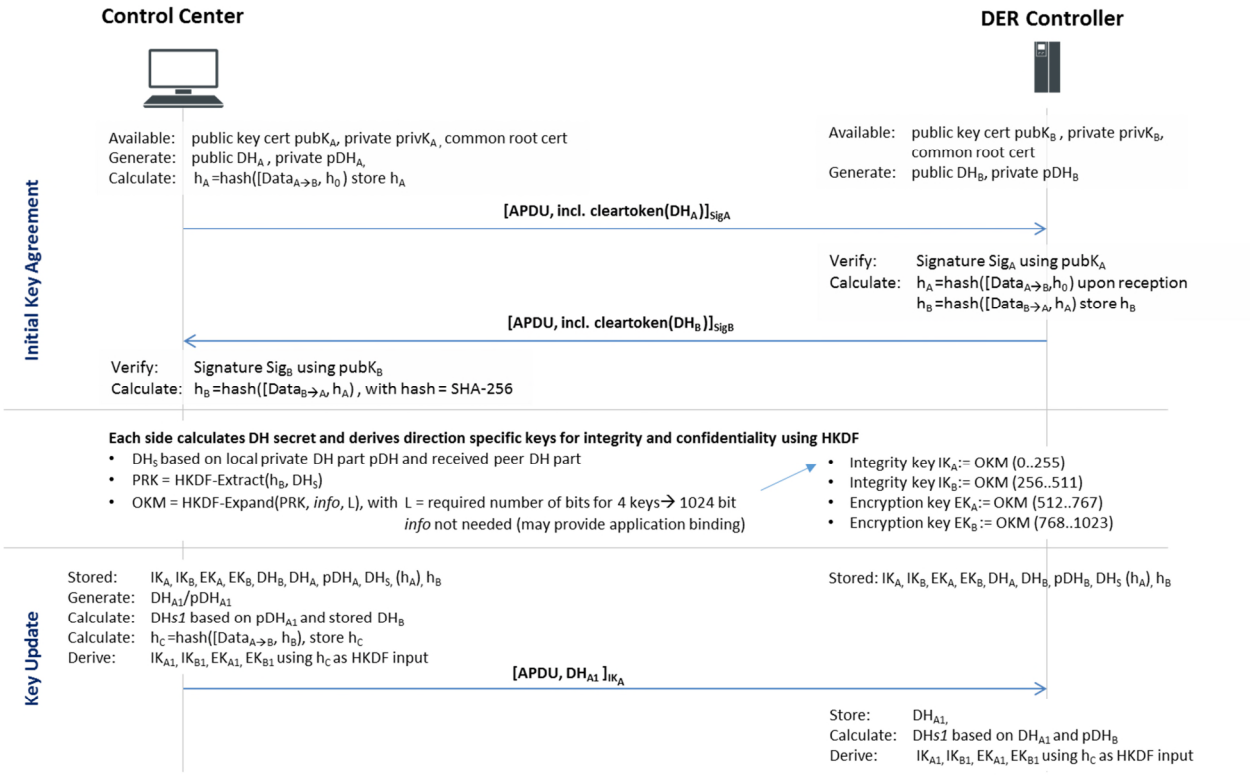

Figure 3. End-to-end-Security and hop-by-hop security according to IEC 62351-4

Additionally, a protocol is assumed that supports session initiation. In the specific example, this is provided by the Manufacturing Message Specification (MMS [25]) using the *MMS Initiate* and *MMS Initate Response* messages.

B. Session Handling

The session handling can be distinguished into the initial key agreement during the session initialization and a key update phase. Both sequences are shown in Figure 3. At the beginning of the session, both sides generate a Diffie Hellman key pair to be used in the key agreement resulting in an ephemeral Diffie-Hellman secret. All data necessary for the establishment of the security association between both peers are kept in a data structure called clear token (as the data s transmitted n clear, but integrity protected). From each of the handshake messages a fingerprint is taken using a hash function. The hash is calculated over the concatenation of the current message and the hash of the previous message (the first message uses “0” for the previous message). This fingerprint is used to ensure the right order of messages and to provide additional randomness to the messages. This “running” hash was inspired by the TLS handshake [8]. Upon reception of the initiation message, the receiver verifies the signature, calculates the fingerprint and generates the response message, from which again the fingerprint is taken. After providing the signed response to the imitator, both sides can calculate the Diffie-Hellman secret and utilize it together with the running hash over the response message as input for the hash based key derivation function HKDF. This will generate different keys per direction for integrity protection and confidentiality protection, resulting in four keys. The keys are applied according to the security association.

The key update can be done using a single message. Figure 3 shows the key update triggered by the control center. As in the initial step, the control center generates a fresh Diffie Hellman key pair and utilizes the already received and stored Diffie-Hellman key from the DER controller to immediately to calculate a new Diffie-Hellman secret and the resulting set of updated session keys. Once this message is received by the DER controller, it can calculate the updated set of keys.

C. Packet construction

Figure 4 shows the packet construction and how the different parts of the messages are protected. Note that during the initial handshake, the clear token is only integrity protected. As stated before, the clear token carries all cryptographic parameter necessary to establish the security association.

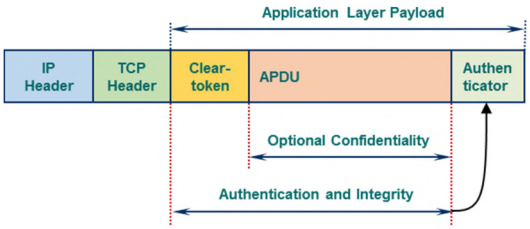

Figure 4. Application of IEC 62351-4 end-to-end security

V. CONCLUSIONS

The lean approach taken in IEC 62351-4 as described in Section IV establishes an end-to-end security session between two communicating peers with mutual entity authentication resulting in session keys being applied for end-to-end message integrity and confidentiality.

Two points should be obeyed when applying the discussed approach. First, the initial key agreement results in an ephemeral set of session keys, as both sides are expected to generate fresh Diffie Hellman parameters. The key update performed in a single message initiated by either peer results in a semi-static Diffie Hellman key agreement. Depending on the security requirements, the receiver may initiate another key update to ensure the freshness of his Diffie Hellman parameters. The second point relates to potential privacy requirements. The initial key agreement utilizes a clear-text token, which is only integrity protected. Thus, all information contained in the token is potentially readable by an intermediary. As the clear token also contains certificate information, it may allow to identify the communication end points.

This paper described an approach of handling end-to-end security over intermediate nodes from a system point of view, by investigating existing security requirements and existing solutions. The paper focused on the description of the end-to-end security approach defined in IEC 62351-4 from a general perspective protecting higher layer session-based communication in an end-to-end fashion, to motivate the re-use of this lean approach in other scenarios or protocol frameworks in industrial communication. As an outlook to this, it is intended to apply the described approach also to other publish-subscribe protocols utilized in automation scenarios like MQTT or AMQP.

REFERENCES

[1] European Commission, “The Directive on security of network and information systems (NIS Directive 2016/1148)”, https://eur-lex.europa.eu/eli/dir/2016/1148/oj, [retrieved: June 2019].

[2] German IT Security Act, official web site (German) https://www.bsi.bund.de/DE/Themen/Industrie_KRITIS/KRITIS/IT-SiG/it_sig_node.html, [retrieved: June 2019]

[3] ISO 27019: Information technology - Security techniques - Information security controls for the energy utility industry, https://www.iso.org/standard/68091.html, [retrieved: June 2019].

[4] IT Security Catalog, https://www.bundesnetzagentur.de/SharedDocs/Downloads/DE/Sachgebiete/Energie/Unternehmen_Institutionen/Versorgungssicherheit/IT_Sicherheit/IT_Sicherheitskatalog_2018.pdf, [retrieved: June 2019].

[5] NIST Framework for Improving Critical Infrastructure Cybersecurity, https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04162018.pdf , [retrieved: June 2019].

[6] NERC CIP Set of Standards, https://www.nerc.com/pa/Stand/pages/cipstandards.aspx, [retrieved: June 2019].

[7] IEC 62443, “Industrial Automation and Control System Security” (formerly ISA99), available from: https://www.isa.org/isa99/, [retrieved: June 2019].

[8] T. Dierks and E. Rescorla, “Transport Layer Security Protocol version 1.2”, RFC 5246, August 2008, https://tools.ietf.org/html/rfc5246, [retrieved: June 2019].

[9] E. Rescorla, “Transport Layer Security Protocol version 1.3”, RFC 8446, August 2018, https://tools.ietf.org/html/rfc8446, [retrieved: June 2019].

[10] S. Fries, R. Falk, H. Dawidczak, and T. Dufaure, “Decentralized Energy in the Smart Energy Grid and Smart Market – How to master reliable and secure control Secure Integration of DER into Smart Energy Grid and Smart Market,” International Journal of Advances in intelligent Systems, vol. 9 no 1&2, 2016, ISSN: 1942-2679, page 65-75, https://www.thinkmind.org/download.php?articleid=intsys_v9_n12_2016_6, [retrieved: June 2019].

[11] ISO 61850-x: Communication networks and systems for power utility automation, https://www.iec.ch/search/?q=61850, [retrieved: June 2019].

[12] P. Saint-Andre, ”Extensible Messaging and Presence Protocol (XMPP): Core,” RFC 6120, https://tools.ietf.org/html/rfc6120 [retrieved: June 2019].

[13] IEC 62351-x Power systems management and associated information exchange – Data and communication security, https://www.iec.ch/search/?q=62351 [retrieved: June 2019].

[14] P. Saint-Andre, “End-to-End Signing and Object Encryption for the Extensible Messaging and Presence Protocol (XMPP),” RFC 3923, https://tools.ietf.org/html/rfc3923 [retrieved: June 2019].

[15] W3C: XML Signature Syntax and Processing Version 2.0, June 2015, https://www.w3.org/TR/xmldsig-core2/, [retrieved: June 2019].

[16] W3C: XML Encryption Syntax and Processing Version 1.1, April 2013, https://www.w3.org/TR/xmlenc-core1/, [retrieved: June 2019].

[17] M. Jones, J. Bradley, and N. Sakimura, “JSON Web Signature (JWS),” RFC 7515, https://tools.ietf.org/html/rfc7515, [retrieved: June 2019].

[18] M. Jones and J. Hildebrand, “JSON Web Encryption (JWE),” RFC 7516, https://tools.ietf.org/html/rfc7516, [retrieved: June 2019].

[19] J. Schaad, “CBOR Object Signing and Encryption (COSE),” RFC 8152, https://tools.ietf.org/html/rfc8152, [retrieved: June 2019].

[20] O. Friel, R. Barnes, M. Pritikin, H. Tschofenig, and M. Baugher, “Application layer TLS,” IETF Draft, https://tools.ietf.org/html/draft-friel-tls-atls-02, [retrieved: June 2019].

[21] E. Rescorla, Key Material Exportes fro Transport Layer Security, “ RFC 5705, https://tools.ietf.org/html/rfc5705, [retrieved: June 2019].

[22] Signal protocol, https://signal.org/docs/, [retrieved: June 2019].

[23] Off-the-record Protocol Description version 3, https://otr.cypherpunks.ca/Protocol-v3-4.1.1.html, [retrieved: June 2019].

[24] Application Layer Transport Security, https://cloud.google.com/security/encryption-in-transit/application-layer-transport-security/, [retrieved: June 2019].

[25] Manufacturing Message Specification, ISO 9506, https://www.iso.org/standard/37080.html, [retrieved: June 2019].

 ISBN: 978-1-61208-746-7