http://dx.doi.org/10.14236/ewic/icscsr19.17

Authentication of GOOSE Messages under Timing Constraints in IEC 61850 Substations

Ghada Elbez, Hubert B. Keller, Veit Hagenmeyer
Institute of Automation and Applied Informatics (IAI)
Karlsruhe Institute of Technology (KIT)
Hermann-von-Helmholtz-Platz 1,
76344 Eggenstein-Leopoldshafen
Germany
ghada.elbez@kit.edu, *hubert.keller@kit.edu*,
veit.hagenmeyer@kit.edu

For the future generation of energy systems, secure communication is a key component in ensuring a reliable and stable operation. The actual respective standard to define the communication network architectures for substation automation is the IEC 61850. In order to address the shortcomings of IEC 61850 w.r.t. communication security, IEC 62351-6 introduces respective recommendations. However, a thorough analysis of these recommendations shows that the authenticity and integrity of time-critical protocols such as Generic Object Oriented Substation Event (GOOSE) messages are not entirely covered by the proposed security measures. Therefore, in the present contribution, implementation of the RSASSA-PSS and HMAC-SHA256 authentication are investigated for the given context. Comparison with previous works is provided and obtained results show that the HMAC scheme has a better computational time than the recommended RSASSA-PSS. Thus, adjustment of the IEC 62351-6 considering the authentication of GOOSE messages shall be considered in the next edition of the standard.

Cyber-physical security; authentication; RSA; HMAC; electrical substations; IEC 61850; IEC 62351; GOOSE.

1. INTRODUCTION

The use of Information and Communication Technologies (ICT) is becoming more spread in energy systems as it offers real-time control, monitoring and maintenance features. The integration of the respective IEC 61850 standard International Electrotechnical Commission (IEC) (2007) in the development of the future energy systems provides several advantages – among interoperability and faster communication over Ethernet.

Despite all the advantages offered by IEC 61850, the standard was created without security being a primary goal. Unfortunately, there has been little focus on the threats that might affect the network communication security of electrical substations. Part 6 of IEC 62351 in International Electrotechnical Commission (IEC) (2010) was thereafter introduced in order to extend IEC 61850 with security measures. [More details about these security recommendations can be found in Section 2.2].

An accurate analysis of the proposed IEC 62351 security measures for the GOOSE protocol in IEC 61850 (which is used to broadcast event data over the electrical substation network) shows that the latter remains largely untouched by security premises in IEC 62351 (see Section 2).

In particular, authenticating the GOOSE messages to prevent replay and tampering attacks, was only partially addressed in IEC 62351-6. Ensuring the integrity and authenticity of GOOSE messages is critical for the optimal operation of the electrical substation. Thus, it is a relevant topic that has been considered within the research community in the last few years. Hohlbaum et al. (2010) analyze practical considerations of security recommendations in IEC 62351 by checking implementation of digital signatures on different platforms. The performance assessment of the authentication schemes was studied considering limitations of Intelligent Electronic Devices (IEDs) and real-time requirements of the GOOSE protocol. It is worth noting that the computational time of Rivest-Shamir-Adleman (RSA) signature as presented in Hohlbaum et al. (2010), includes only the signing of the hash value omitting the verification of the digital signature. In Ishchenko

and Nuqui (2018), implementation of RSA and Hash-Based Message Authentication (HMAC) for GOOSE authentication as bump-in-the-wire was presented. Computational time of RSA 1024-bit, HMAC and Galoi Message Authentication Code (GMAC) was compared with the previously mentioned work. A delay simulation of the network implementation of GOOSE authentication based on HMAC was presented in Fang et al. (2018). A software simulation based on OPNET was considered, but no details about the GOOSE authentication scheme were provided.

In the previously mentioned works, the exact authentication scheme recommended in IEC 62351-6, namely RSA Probabilistic Signature Scheme with Appendix (RSASSA-PSS), was not evaluated. Works considering the implementation of the exact signature scheme proposed in IEC 62351-6 are scarce. Farooq et al. (2019) analyzed the implementation of probabilistic signature schemes as RSASSA-PSS signature with 1024 and 2048-bit keys for the authentication of GOOSE messages and compared obtained results with previous works. Achieved results show that probabilistic signature schemes cannot meet the hard real-time requirements of GOOSE messages. However, no comparison with symmetric cryptographic authentication methods was undertaken.

Therefore, in the present paper, shortcomings of the authentication scheme proposed in the IEC 62351-6 standard are first presented and analyzed. In this light, the recommended signature-based RSASSA-PSS authentication is thus implemented in order to secure GOOSE messages and results show incompatibilities with real-time requirements of GOOSE protocol. An alternative authentication method based on HMAC is thus, proposed to satisfy the real-time constraints. Both authentication schemes are compared to assess their performance in order to propose necessary adjustments of the IEC 62351-6 in the next edition of the standard.

The organization of this paper is as follows. In Section 2, the basics of the GOOSE protocol are first introduced. Then, a summary of the security recommendations in IEC 62351-6 followed by an analysis of the different shortcomings of the security measures of the GOOSE protocol is given. To address the unveiled security flaws, authentication schemes for GOOSE messages are explained in Section 3.1 and further tested and implemented in Section 4. Finally, in Section 5 conclusions and future work are presented.

2. SHORTCOMINGS OF SECURITY RECOMMENDATIONS IN IEC 62351 FOR GOOSE PROTOCOL

When first proposed, the IEC 61850 International Electrotechnical Commission (IEC) (2007) was not particularly focused on the security aspects of energy systems. However, different attack scenarios against IEC 61850 electrical substations were presented in existing works such as in Kush et al. (2014) and Hoyos et al. (2012). Consequently, the IEC 62351 standard introduced recommendations to guarantee the information security of power systems using, for instance, authentication mechanisms. The IEC 62351 standard was developed by the Technical Committee 57 within the Working Group 15 (TC57 WG15) and the 1st edition was published back in 2007. The standard is split into eleven parts concerned with the end-to-end security of the communication in power systems. Part 6 of the IEC 62351 standard, in particular, proposes security measures for the electrical substations based on IEC 61850. Necessary knowledge about the GOOSE protocol will be introduced in the following. For ease of use, the IEC 61850 and IEC 62351 will be referred to as 61850 and 62351, respectively.

2.1. GOOSE protocol

The data object model defined in the 61850 standard is mapped to different protocols. The GOOSE protocol in the International Electrotechnical Commission (IEC) (2011) is a multicast publisher/subscriber data transfer method mapped directly over Ethernet. GOOSE messages are exchanged between process and bay levels as well as between IEDs in the bay level.

A specific transfer mechanism is used to ensure the reliability of GOOSE messages without the conventional acknowledgment procedure. When an event occurs, a GOOSE message is generated and repeated first at high frequency, then at a slower one until reaching a predefined frequency in stable conditions.

Even though security is not the core of the 61850 standard, there are some mechanisms to comply with the strict real-time requirements. A first approach consists in mapping GOOSE messages directly to the link-layer to reduce processing time. Another mechanism to ensure efficient processing and transmission is to select a high priority tag to avoid slowing down the transmission of GOOSE packets.

The GOOSE message frame has a datagram complying with ISO/IEC 8802.3. More details can be found in 61850-8.1 International Electrotechnical Commission (IEC) (2011). Within the frame structure,

there is a field called "Security" that is reserved for digital signature according to the recommendations in 62351 in International Electrotechnical Commission (IEC) (2010). However, no further details are provided in 61850-8.1. In the following, some of the main suggestions, presented in IEC 62351-6, for the security of the GOOSE protocol are reported.

2.2. Main security recommendations in IEC 62351

Different security measures for GOOSE messages were described in 62351-6 International Electrotechnical Commission (IEC) (2010). Firstly, use of the fields Reserved 1 for the number of the extension octets and Reserved 2 for a 16-bit cyclic redundancy check (CRC) is recommended. Secondly, the 62351-6 standard International Electrotechnical Commission (IEC) (2010) recommends the authentication of the GOOSE messages with a digital signature. Indeed, a security extension of the Application Protocol Data Unit (APDU) with an RSA-based signature with Appendix-Probabilistic Signature Scheme (RSASSA-PSS) is proposed (see Bellare and Rogaway (1996)). It is however, worth noticing that the RSA signature explicitly excludes the Ethernet header. Authentication and integrity of GOOSE messages supporting the digital signature is thus guaranteed.

Another security recommendation in 62351-6 introduces an extension in the Substation Configuration Language (SCL) files to allow the use of different certificates for GOOSE messages. An additional security algorithm against replay attacks was set in 62351-6. It is based on a check of the freshness of the messages with skew filtering and messages timestamps. However, for hard real-time applications such as for GOOSE messages with 3 ms response time, no encryption scheme is suggested. It is indeed, clearly stated in International Electrotechnical Commission (IEC) (2010) that "for applications [...] requiring 3 ms response times, multicast configurations and low CPU overhead, encryption is not recommended."

2.3. Security flaws in IEC 62351

Whereas the 62351-6 standard enhances the security of GOOSE messages considerably, it needs further improvements as there are still some security flaws. Implementation of security solutions for peer-to-peer communications might lead to an undesirable latency as concluded by Hoyos et al. (2012). Firstly, in 62351-6, it is clearly stated that encryption is not recommended for GOOSE messages in International Electrotechnical Commission (IEC) (2010). Thus, the risk of Denial Of Service (DoS) attacks is still present. The security of GOOSE messages is limited to adding a digital signature to their APDU. However, the performance of the authentication mechanisms based on digital signatures recommended for Sampled Values (SV) and GOOSE communications is still an open question. In fact, strict real-time requirements for certain messages cannot be respected yet as it was shown by Hohlbaum et al. (2010). This might be a shortcoming when considering specific type of messages such as 1A GOOSE messages that should have a maximum of 3 ms response time as specified in 61850 International Electrotechnical Commission (IEC) (2007). Therefore, one of the current challenges is to reconcile the needs for security and low latency as suggested by Hoyos et al. (2012) in electrical substations. Most manufacturers do not yet consider practical implementation of the encryption and authentication due to the remaining ambiguity. Thus, the acceptance of 62351 will largely depend on its impact on interoperability, performance, and manageability as presented by Hohlbaum et al. (2010). Works such as Hohlbaum et al. (2010) and Farooq et al. (2019) have been carried out w.r.t. testing the authenticity and integrity of GOOSE messages according to 62351-6. However, only one of them i.e. Farooq et al. (2019) implemented the exact RSAASS-PSS authentication scheme as reported in Section 1. As there is still little clarity on how to implement security for fast GOOSE messages without degrading the actual performance of Intelligent Electronic Devices (IEDs) Hoyos et al. (2012), the present paper will shed a light on possible authentication schemes that shall be considered for the security of GOOSE messages. Different authentication schemes that might be used to ensure the integrity and authenticity of GOOSE messages are presented in the next section.

3. AUTHENTICATION SCHEMES TO SECURE GOOSE MESSAGES

Authentication of the communication within modern electrical substations covers two main security goals which are integrity and authenticity, respectively. The integrity of a message is ensured provided that the data is accurate and consistent without any unauthorized tampering. Digital signatures or Message Authentication Codes (MAC) combined with cryptographic hashing techniques such as HMAC are commonly used to ensure integrity and authenticity.

3.1. Digital Signature Authentication

Digital signatures are mathematical algorithms used to ensure the authenticity and integrity of GOOSE messages. A hash value of the message is first calculated using a private key and attached to the original message by the publisher. The signed message is then sent to the subscriber. Using a public key, the signature of the received GOOSE message is decrypted. Next, the hash of the original

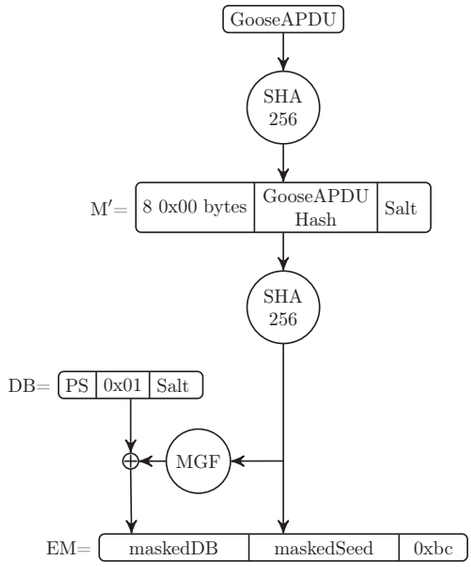

Figure 1: *Signing step of RSASSA_PSS*

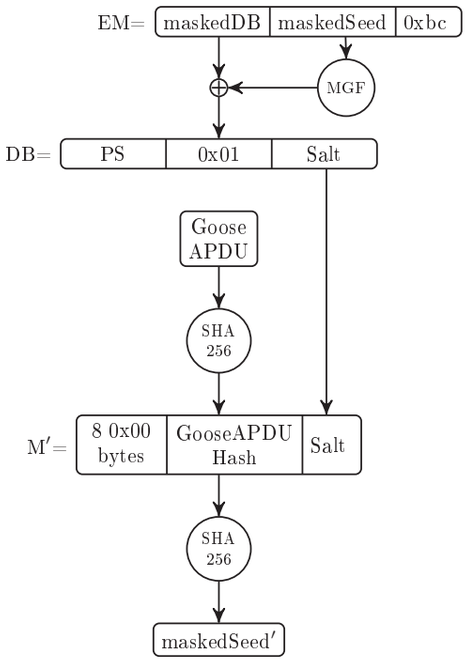

Figure 2: *Verification step of RSASSA_PSS*

message is computed. To check the authenticity of the GOOSE message, the hash computed from the received message and the one decrypted, are compared. Usage of digital signatures ensures that the message is truly sent from an authorized source. Integrity is also provided, since modifying a message after being signed would invalidate the signature.

RSA signatures are one of the most common digital signatures schemes that were first published in Rivest et al. (1978). Using probabilistic digital signatures is recommended in IEC 62351-6 as it offers better security International Electrotechnical Commission (IEC) (2010). Thereby the so-called Probabilistic Signature Scheme (PSS) takes the message as well as a random value as the inputs to the hash function.

The RSA-Probabilistic Signature Scheme with Appendix (RSASSA-PSS) algorithm was chosen in 62351-6 to authenticate GOOSE messages in electrical substations. A newer version of the signature scheme that uses the PKCS-v1_5 encoding operation was proposed in the PKCS standard Böck (2011). There are two main steps in the signing of a GOOSE message according to RSASSA-PSS procedure which are applying the signature and the validation steps.

Figure 1 describes the different operations of the probabilistic signature RSASSA-PSS signing stage. The first step when encoding a GOOSE message is to hash with Secure Hash Algorithm (SHA) 256 algorithm the GOOSE APDU. A masked hash (M') is obtained from the encoded message to which a salt and padded eight zeros are added. This value is further hashed using a SHA256 algorithm which results in a maskedSeed. Then, a Mask Generation Function (MGF) is applied to the result of the previous step. An XOR operation between the DB – that is a concatenation of zero padding PS, 0x01 value as well as a salt – and the MGF allows obtaining a maskedDB. The encoded message (EM) is a result of the concatenation of the maskedDB, the maskedSeed and a compatibility value (0xbc).

The second step of the process namely the RSASSA-PSS verification is presented in Figure 2. The reverse steps are performed: From the received encoded message, the DB is obtained from an XOR operation between the MGF of the maskedSeed and the maskedDB. Then, the hash value maskedSeed' is computed from the recovered salt. This value is finally compared with the one received in the message to check the authenticity of the GOOSE packet.

3.2. Message Authentication Code based on keyed hash (HMAC)

Instead of the asymmetric RSA algorithm suggested in 62351-6, the HMAC algorithm with the SHA256 hash calculation can be used (see Krawczyk et al. (1997)). Choosing a symmetric cryptography method such as HMAC for the integrity and authenticity of the GOOSE protocol offers different advantages. For instance, a reduced computational time is expected with the use of private keys. As the calculation of the digest with HMAC is based on a combination of the message content and the key at the same time, a faster computation is expected. The HMAC algorithm

is mainly based on calculating a code derived from the GOOSE message and a private key value. The HMAC algorithm has as inputs a key extracted from a corresponding certificate as well as the GOOSE APDU. The output of the algorithm, namely digest, is appended to the original message and sent by the publisher. The subscriber performs the same algorithm with the received message and compares both, the received and calculated HMAC values to check the authenticity and integrity of the message (see Krawczyk et al. (1997)).

Usage of the Group Domain of Interpretation (GDOI) scheme is recommended in 61850-90-5 for the key management, but no further specification are provided in the first edition of 62351 (International Electrotechnical Commission (IEC) (2010)). For comparison reasons with a 1024-bit RSA digital signature authentication scheme, a key length of 128-bit is used for the HMAC algorithm. The algorithm 3.1 describes the different steps to obtain a HMAC code.

The GOOSE publisher calculates the HMAC value and appends it to the message that is then sent to the subscriber. The subscriber recalculates a new value of the HMAC code based on the received message and the secret key. This second value is compared with the one appended to the received GOOSE message to verify the authenticity and the identity of the sending entity.

Algorithm 3.1: HMAC_SHA256 $(GooseAPDU, key)$

Vars: $\begin{cases} blockSize \text{ :integer}/\!/512 \text{ bits} \\ outputSize \text{ :integer}/\!/256 \text{ bits} \\ \text{SHA256 :hash function} \end{cases}$

if $length(key) > blockSize$
  then $key \leftarrow \text{SHA256}(key)$
if $length(key) < blockSize$
  then $key \leftarrow pad(key, blockSize)$
$o\_key\_pad = key\text{xOR}[0x5c * blockSize]$
$i\_key\_pad = key\text{xOR}[0x36 * blockSize]$
return $\begin{cases} \text{SHA256}(o\_key\_pad \parallel \text{SHA256} \\ (i\_key\_pad \parallel GooseAPDU)) \end{cases}$

4. IMPLEMENTATION AND RESULTS OF AUTHENTICATION SCHEMES OF GOOSE MESSAGES

The previously presented authentication mechanisms are implemented using the OpenSSL by Andrew Young and Hudson (1998) library version 1.0.2 on an Intel i7-8550U CPU @ 1.80 GHz with 16 GB RAM system. The simulation of the GOOSE messages is carried out as suggested in Elbez et al. (2018). Considerations of key management should take into account the large networks structure of electrical substations and the strict time requirements which is out of the scope of the present paper. Thus, private key management is not considered in the remainder, but it will be dealt with in future work.

To compare our results with the ones reported in the literature, table 1 is established in order to summarize the different characteristics of the used platforms to simulate GOOSE authentication methods.

As shown in table 2, most of the asymmetric cryptographic solutions based on RSA are unable to meet the real-time requirements for GOOSE messages, except when using a Xeon Server CPU (3,1) where the computation of an RSA 1024-bit requires 0.3 ms Ishchenko and Nuqui (2018). However, hardaware used in (3,1) does not reflect the one used in electrical substations (Ishchenko and Nuqui (2018)): When using the whole signature scheme RSASSA-PSS as specified in 62351-6, improved results are obtained, however, still without complying with the 3 ms requirement.

Choosing the most suitable key could be also challenging, since, according to a report of NIST back in 2011, RSA 1024-bit keys might be used. However, in a 2013 NIST report, the use of a 2048-bit key was recommended instead, cf. Barker and Dang (2015). In the present paper, RSA-1024 bit are implemented as use of 2048-bit would imply an even larger computational time.

As expected, when considering HMAC, the total authentication time is of the order of micro-seconds independently from the used platform as shown in table 3. In fact, HMAC algorithms have a better computational time as the digest calculation is directly computed from the message and the key together. The recommendations suggested in 62351-6 regarding the authentication and the integrity of GOOSE messages using RSASSA-PSS digital signature should be reconsidered and probably replaced by a symmetric cryptographic scheme such as HMAC-SHA256.

5. CONCLUSION AND FUTURE WORK

In this paper, a review of the the different security recommendations suggested in IEC 62351-6 is presented. The different vulnerabilities of those recommendations concerning GOOSE security are analyzed. Based on those shortcomings, authentication schemes to secure GOOSE messages are introduced and performance results concluded. Authentication using digital signatures with probabilistic

Ref.	Index	Technical details
The present work	(1,1)	Intel i7-8550U CPU @ 1.80 GHz
Hohlbaum et al. (2010)	(2,1)	Pentium M 1.7 GHz / 1GB RAM
	(2,2)	Intel Core 2 Duo @ 2.2 GHz / 2GB RAM
	(2,3)	FPGA (100 MHz)
	(2,4)	FPGA (200 MHz)
Ishchenko and Nuqui (2018)	(3,1)	Xeon X3440 server 2.53 GHz quad-core
	(3,2)	BeagleBone Black (TIAM3359 ARM Cortex A8 CPU @ 1 GHz)
	(3,3)	RPi2 Raspberry Pi 2 (Broadcom BCM2836 quad-core ARM Cortex A7 overclocked at 1 GHz
Farooq et al. (2019)	(4,1)	Intel i5-3210M CPU @ 2.50 GHz

Table 1: *Technical details in implementation of authentication of GOOSE messages*

	Digital Signature	
Index	RSA with 1024-bit key in ms	RSASSA-PSS with 1024-bit key in ms
(1,1)	9.2	4.3
(2,1)	6.8	-
(2,2)	4	-
(2,3)	3.748	-
(2,4)	1.917	-
(3,1)	0.3	-
(3,2)	$5 < t < 7$	-
(3,3)	$10 < t < 12$	-
(4,1)	10	5.45

Table 2: *Digital signature computational time in ms*

	Message Authentication Code	
Index	HMAC-SHA256 (32 Bytes key) in μs	HMAC (16 Bytes key) in μs
(1,1)	16	-
(3,1)	-	4
(3,2)	-	23
(3,3)	-	53

Table 3: *HMAC calculation computational time in μs*

signature scheme (RSASSA-PSS), as suggested in 62351-6, shows unsatisfactory results for hard real-time GOOSE messages that require 3 ms response time. When compared with digital signatures, authentication methods based on symmetric authentication mechanisms offers better computational time. Implementation of HMAC-SHA256 on a platform comparable to a modern Intelligent Electronic Device (IED) largely satisfies the GOOSE messages' strict time requirements. A respective adjustment of the IEC 62351-6 considering the authentication of GOOSE messages shall be considered in the next edition of the standard.

As future work, this authentication scheme will be further tested as a security filter or bump-in-the-wire implementation on a different hardware platforms in order to analyze its operation with legacy IEDs and thus its backward compatibility. As presented in Section 2, one of remaining challenges when addressing GOOSE security, is to defend against DoS attacks. Future work will include a better analysis of this shortcoming as well as possible solutions such as Intrusion Detection Systems (IDS) to ensure availability within the electrical substation networks.

ACKNOWLEDGEMENT

This research was partially supported by the Federal Ministry of Education and Research (BMBF) within the framework of the project "Neue EnergieNetzStruktURen für die Energiewende" ENSURE (FKZ 03SFK1N0) and the Competence Center for Applied Security Technology KASTEL_SKI project (16KIS0843).

REFERENCES

E. Andrew Young and T. Hudson. OpenSSL library, 1998. URL `https://www.openssl.org/`.

E. Barker and Q. Dang. NIST Special Publication 800-57, Part 3: Application-Specific Key Management Guidance, 2015.

M. Bellare and P. Rogaway. The exact security of digital signatures-how to sign with rsa and rabin. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 399–416. Springer, 1996.

J. Böck. RSA-PSS–Provable secure RSA Signatures and their Implementation. *Accessed: Jan*, 31:2019, 2011.

G. Elbez, H. B. Keller, and V. Hagenmeyer. A Cost-efficient Software Testbed for Cyber-Physical Security in IEC 61850-based Substations. In *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 1–6, Oct 2018. doi: 10.1109/SmartGridComm.2018.8587456.

F. Fang, D.-h. WANG, X.-h. XUAN, J.-h. XIE, and C. Qing. Application research of hmac in intelligent substation communication security. *DEStech Transactions on Engineering and Technology Research*, (ecar), 2018.

S. M. Farooq, S. S. Hussain, and T. S. Ustun. Performance evaluation and analysis of IEC 62351-6 probabilistic signature scheme for securing GOOSE messages. *IEEE Access*, 7:32343–32351, 2019.

F. Hohlbaum, M. Braendle, and F. Alvarez. Cyber Security Practical considerations for implementing IEC 62351. In *PAC World Conference*, 2010.

J. Hoyos, M. Dehus, and T. X. Brown. Exploiting the GOOSE protocol: A practical attack on cyber-infrastructure. In *Globecom Workshops (GC Wkshps), 2012 IEEE*, pages 1508–1513. IEEE, 2012.

International Electrotechnical Commission (IEC). IEC 61850: Power Utility Automation (TC57), 2007.

International Electrotechnical Commission (IEC). IEC 62351: Power systems management and associated information exchange - data and communications security - part 6: Security for IEC 62351, 2010.

International Electrotechnical Commission (IEC). IEC 61850: Communication networks and systems for power utility automation in substation - part 8.1, 2011.

D. Ishchenko and R. Nuqui. Secure communication of intelligent electronic devices in digital substations. In *2018 IEEE/PES Transmission and Distribution Conference and Exposition (T&D)*, pages 1–5. IEEE, 2018.

H. Krawczyk, M. Bellare, and R. Canetti. HMAC: Keyed-hashing for message authentication. 1997.

N. Kush, E. Ahmed, M. Branagan, and E. Foo. Poisoned GOOSE: exploiting the GOOSE protocol. In *Proceedings of the Twelfth Australasian Information Security Conference-Volume 149*, pages 17–22. Australian Computer Society, Inc., 2014.

R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21 (2):120–126, 1978.