Addressing Security in OCPP: Protection Against Man-in-the-Middle Attacks

Juan E. Rubio, Javier Lopez, Cristina Alcaraz
Department of Computer Science, University of Malaga,
Campus de Teatinos s/n, 29071, Malaga, Spain
{rubio, alcaraz, jlm}@lcc.uma.es

Abstract

The Open Charge Point Protocol (OCPP) is a communication standard for the exchange of data between a Charge Point (CP) and the Central Server (CS) in the electric vehicle domain. This protocol is envisioned to offer interoperability between the different manufacturers of charging points, network systems and IT back-end vendors. However, the current version of the specification is quite vague in terms of handling security and privacy, which results in a set of non-addressed threats, which we look at in this paper. Specifically, this paper focuses on Man-in-the-Middle attacks between the CP and the CS that may expose sensitive data of special interest to the various stake-holders involved in this context. As a counter-measure, we present a feasible solution and assess its behaviour in a simulator. The inclusion of additional security mechanisms is also studied, in compliance with the IEC 62351 standard.
Keywords: Smart Grid, OCPP, Security, Privacy, Control.

1 Introduction

The Smart Grid technologies offer multiple improvements over the traditional grid model, due to the real-time communication between smart meters on the customer’s premises and the supervision systems of the central station [1]. On the one hand, it achieves a better management of congestions and a fine grained billing for operators. On the other hand, users receive an accurate measurement of consumption that allows them to regulate their power usage according to the price fluctuation and hence save money. Electric Vehicles (EVs) play an important role here, as they can be considered as geographically dispersed electrical appliances since they consume a lot of electricity, but also represent a power source if their batteries are discharged (so their users are possibly rewarded). It therefore becomes a challenge to balance electricity supply and demand in EV charging as it increases the complexity of the ICT infrastructure behind the grid because of the multiple information flows.

For this reason, it is essential to introduce security measures to keep the data safe when it is transmitted between the different parties involved, who may be interested in such information to gain a business advantage [2]. For instance, car manufacturers, the energy service provider or the charge point operators. Regardless of the business model, the physical procedure of charging a vehicle is basically the same: the customer plugs the vehicle into one of the sockets provided by the Charge Point (CP), which also contains an electricity meter. Before the charging starts, he/she is commonly identified with an RFID card, which is also required at the end, to be able to remove the cable. The details of the charging session are sent to the Central System (CS) of the charge point operator through OCPP (Open Charge Point Protocol), an internationally standardised protocol [3] designed to accommodate any type of charging technique and improve interoperability between different vendors of charge points and back-end systems. OCPP is an application protocol that uses a client/server architecture where the requests can be initiated by both the client and server (i.e. the CP and the CS), although certain operations are invoked only by one of them.

OCPP does not specify any underlying communication technology (e.g., Power Line Communication) and does not define strong security services [4]. This means it relies on the security provided by other protocols at lower levels. In the current version of the specification [3], version 1.6, released in 2015 (version 2.0 is under development), OCPP suggests the use of Transport Layer Security (TLS) at the transport layer to provide confidentiality for communication links. However, this protocol has been demonstrated to be insecure against Man-in-the-Middle (MITM) attacks in certain scenarios [5], especially in the EV context where it is difficult to guarantee true end-to-end security [6]. This is because, on the one hand, TLS is vulnerable with respect to certificates validation which can lead to communication hijacking [7]. On the other hand, because OCPP can also be used to send information from the CP to other entities for billing and monitoring purposes, and that data traverses various optional nodes within the EV architecture. This means that those intermediate points have to be trusted not to change or disclosure this data, so an additional security mechanism has to be introduced. Moreover, manufacturers may not consider the recommendation of implementing TLS as it introduces overhead and the use of cellular networks is charged by transmitted byte.

Data leakage from MITM attacks can have severe consequences for the EV market in financial terms and with regard to privacy [8], as we show in the next section. For this reason, it is necessary to introduce further services to OCPP to address a minimum set of security requirements that otherwise may slow down the adoption of EV technology. In this sense, the IEC 62351 standard [9] is a reference in the industry and is useful to help comply with these security requirements.

In this paper we are concerned with finding techniques to protect against MITM between the CP and the CS and reduce their impact. Additionally, we study possible security mechanisms to add to the current EV infrastructure for future upgrades or versions of the OCPP specification, considering the IEC

62351 standard. The paper is organised as follows. In Section 2 the MITM attack model and its consequences in the context of EV charging are presented. In Section 3 a potential solution to overcome this issue is designed, which is simulated in Section 4 to prove it is effective. Section 5 concludes the paper by analysing the potential implementation of other security mechanisms and discussing the future work.

2 Motivation: attack model

As presented, this paper strives to protect the EV infrastructure against Man-in-the-Middle attacks in OCPP. In this context, a potential attacker intercepts the communication in the interface between the charging point and the central system, secretly relaying and possibly altering the information exchanged between these two parties.

To perform such an attack, the adversary taps the communication link between the CP and the CS, which can be based on a wireless technology. Either way, OCPP is based on the transmission of data through that bidirectional link, by means of HTTP or Websockets, using SOAP or JSON, respectively. Independently of the format, that information is sent in clear text, which allows the attacker to disclose, distort or disrupt the data. Moreover, even when TLS is used and information is encrypted, it is possible to access the data due to vulnerabilities in the protocol [10]. Here, the attackers that we are dealing with are insiders (operators of the charge points) or malicious users that have access to the space where the CP is installed.

There is a taxonomy of potential threats [11] in the event an adversary succeeds in his/her attack, which could harm the privacy and reliability of the EV infrastructure and hence the trust placed in it by all the stakeholders:

Exposure of sensitive data: privacy is at risk when the location of the charges and the amount of energy are traced and analysed (e.g. when the EV is charged at home or in public places), which helps to build a personal profile of the user and predict his/her future behaviour (for instance, to determine the distance of his/her trips based on the battery load). This turns out to be valuable information for third parties (e.g. car manufacturers, and insurance companies).

Modification of charging parameters: apart from the private data described, there are other control parameters exchanged between the CP and CS that could motivate a MITM attack, such as pricing information, notifications or configuration data (e.g., firmware updates, access control policies).

The consequences of these threats for the stability of the system range from impairing the service quality (e.g. slow charging or congestion) to a Denial of Service (DoS). An especially negative situation occurs when the attacker tampers with pricing signals to modify the fees: if the price of energy is reduced to a low level a large number of users could try to charge their EVs at the same time, which could make the grid unstable. For the sake of clarity, we concentrate on the protection of sensitive data. Namely, the information provided by the smart meter embedded in the CP that indicates the quantity of energy supplied in a charging session, which notably affects user privacy. However, the same protection described in the following paragraphs can be applied in the scenario of communication between both peers to transmit charge parameters and other control data.

To illustrate the part of the information that we intend to protect, we must first look at the charge operation defined by the OCPP specification [3], which we described briefly in the introduction and is shown in Fig. 1. According to the sequence diagram in this type of operation, the user needs to be authenticated before the charge starts and before it ends. When the charge is provided, the charge point sends a *StopTransaction.req* message that informs the central system of the quantity of energy charged through the *meterStop* value (specifically, computing the difference between that value and the *meterStart* value received at the beginning of the transaction). Although more details about the transaction can be provided to the CS with the optional *transactionData* element within the message, we accept that *meterStop* is the one to be protected, hence its value has to be kept hidden from attackers.

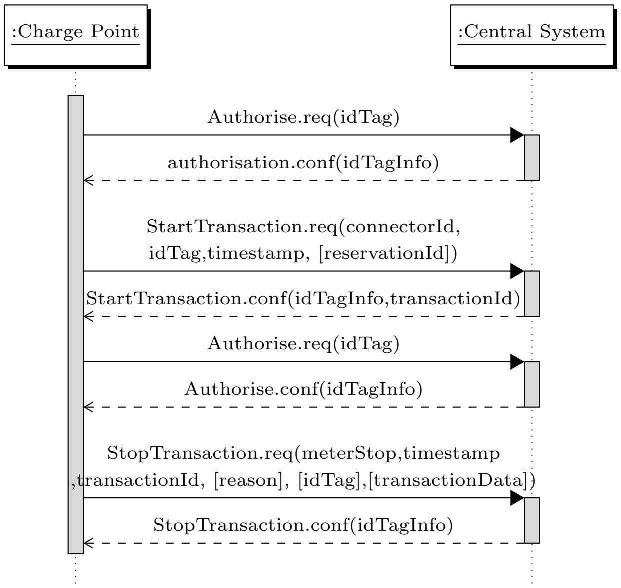
:Charge Point
:Central System
Authorise.req(idTag)
authorisation.conf(idTagInfo)
StartTransaction.req(connectorId,
idTag,timestamp, [reservationId])
StartTransaction.conf(idTagInfo,transactionId)
Authorise.req(idTag)
Authorise.conf(idTagInfo)
StopTransaction.req(meterStop,timestamp
,transactionId, [reason], [idTag],[transactionData])
StopTransaction.conf(idTagInfo)

Figure 1: Transaction message flow between CP and CS [3]

Protection against these attacks is essential for the rapid and successful adoption of these technologies by both users and operators. Therefore, technical measures should be deployed to ensure the correct functioning of all components within the EV architecture. In the next section, we propose a solution in this context, describing a possible way to circumvent MITM attacks, specifically with regards to the confidentiality and privacy of the energy data generated and exchanged between the CP and CS.

3 Proposed Solution

Having reviewed the consequences of MITM attacks for the EV infrastructure, we now look at the design of a technique that might work as countermeasure to reduce the impact of these threats. The objective is to avoid the interception of data between the CP and the CS in OCPP. In this study, we describe a technique that enables the transmission of the energy values securely, so only CP and CS know the actual energy usage data, proposing a solution for future versions of the standard. In order to provide confidentiality to this communication, we use the *Data Transfer* operation defined in OCPP, that allows both the CP and CS to send generic information to each other, so additional functions not defined in OCPP can be flexibly implemented. For our particular purpose, we have designed a light modification of the original charge operation to include an additional step after the *StopTransaction* request, where the *meterStop* is sent securely by applying different cryptographic mechanisms. This solution is based on a secret sharing scheme, which provides a robust measure against MITM attacks, and is simulated and analysed in Section 4. In this kind of algorithm [12], a secret (i.e. the meter value in this case) is divided into chunks (called shares) that are sent separately to the CS, in such a way that it cannot reconstruct the original data unless it collects a defined number of shares at least. Thus, for instance, if the secret is split into five shares with a threshold of three to reconstruct it, a potential attacker who intercepted a message would not be able to access the actual energy value, and he would have to intercept at least three of those messages. Conversely, if he managed to modify the content of a packet, the CS would still be able to reconstruct the secret (specifically, up to two packets could be altered, since only three would be required to rebuild the share).

When focusing on the OCPP scenario, this algorithm becomes even more useful. Let us assume a situation in which EV users perform several charges on the CP over given a time period. Each transaction consequently generates a meter value that must be dispatched to the CS to carry out billing and demand response. In accordance with the proposed scheme, that data is sent in the form of shares over multiple, separate transmissions. However, what is interesting in terms of privacy is that these values can be delivered at irregular intervals of time, so when one transaction ends another one can begin without having to reconstruct the information from the first one in the CS. By this means, for the same CP and CS, secrets belonging to different transactions can interleave and arrive at the CS in random order. In terms of implementation, it requires the CS to hold a buffer of shares, which are tagged with the CP identifier and its corresponding transaction, so secrets are reconstructed with the correct set of shares. It is worthy of note that these IDS must be protected: for instance, using encryption or protection at network level (e.g., with IPSec). An alternative consists in an algorithm agreed by CP and CS that permits the correlation of shares which are sent in a pre-established order.

This communication protocol makes an interception attack infeasible in practice. In order for an attacker to access the meter value when eavesdropping the channel between a CP and a CS, he/she would have to not only collect the minimum number of shares, but also determine which shares belong to the same transaction, which is only known by the CS through the associated ID. With an increase in the number of transactions performed by many users, the probability of finding a valid combination of shares decreases dramatically. Moreover, this technique is compatible with additional cryptographic mechanisms to provide a higher level of authentication, confidentiality and integrity. This prevents, for instance, the generation of fake shares by the attacker, through a digital signature.

4 Simulation and Analysis

This section presents the simulation results of the approach proposed in Section 3, which models the interaction between the CP and the CS when sending the meter data, to avoid MITM attacks. The test cases defined here have been designed and executed using *ocppjs* [13], a simulator of the OCPP protocol[1]. It allows the communication between a charge point and a central system through a command-line interface, implementing the operations defined in the OCPP standard. The *ocppjs* simulator uses SRPC (Symmetric Remote Procedure Call, whose messages are expressed in JSON) over Websockets as the transport protocol, and requires Node.js runtime [14] to be executed. Once we run it, we can start a central system simulator listening for requests on a specific port and then start a charge point simulator that connects to the CS whose IP and port is passed as argument. From this moment on, a prompt is given to enter commands on both CS and CP simulators. In addition to this functionality, where we can simulate simple OCPP operations (e.g. authorisation, transactions), the software also allows developers to define the simulator behaviour with plugins. Basically, they consist of Javascript files which detail custom message flows on both the CS and CP, through callback functions that are called when a request or response is received.

In our case, we have designed a plugin for a test case with the following parameters: the CP charges the vehicle for ten seconds. Then, after stopping the transaction, it sends the meter value to the CS through independent shares: specifically, we divide the secret into three, with a threshold of two to reconstruct the original message. These three shares are sent after a random time, that ranges from one to 300 seconds. After that, as we want to show the shares interleaving effect over time, the CP begins a new transaction of energy and proceeds in the same way, until it completes ten consecutive transactions. The

[1] The simulator is actually compliant with OCPP-v1.5. For our objective, version 1.5 is good enough since we only use the main charge operation and data transfer defined in the core of the specification, and OCPP-v1.6 does not provide any improvement over v1.5 in terms of security.

set of steps for the charge point are described in Algorithm 1. As shown there, $SendShareToCS$ is the function that sends each created share after the transaction of energy is complete, called from $OnStartTransactionConfirm$. After the three shares have been sent, the transaction ID value is incremented by 1 to control how many of them are performed (ten in this case).

Algorithm 1 Secret sharing approach: CP plugin

```
$transactionstarted \leftarrow false$; $transactionId \leftarrow 1$;
$numoftransactions \leftarrow 10$;
SendBootNotification;

function OnBootNotificationConfirm
    SendAuthorizeRequest;
end function

function OnAuthorizeConfirm
    if $transactionstarted$ then
        StopTransaction;
    else
        $transactionstarted \leftarrow true$; StartTransaction;
    end if
end function

function OnStartTransactionConfirm
    Wait 10 seconds;
    SendAuthorizeRequest;
end function

function SendShareToCS(share)
    $t \leftarrow random\ from\ 1\ to\ 300$; Wait $t$ seconds;
    DataTransfer($share$,$transactionId$);
end function

function OnStopTransactionConfirm
    $shares[1..3] \leftarrow SplitIntoShares(meterValue, 3, 2)$;
    for $i \in \{1, ..., 3\}$ do
        SendShareToCS($shares[i]$);
    end for
    $transactionId \leftarrow transactionId + 1$;
    if $transactionId \leq numoftransactions$ then
        $transactionstarted \leftarrow false$; SendAuthorizeRequest;
    end if
end function
```

It is worth commenting that the $SendShareToCS$ function must be run concurrently, so the program flow can continue with new transactions at the same time. In addition, because the CS has to know the transaction identifier of each share to ultimately reconstruct the secret message, the $transactionId$ value is included on the data transferred to the CS together with the share. Thus, the CS only has to buffer all the incoming shares in accordance with the CP and transaction ID, and then reconstruct the meter value once it has collected a determined number (two in this test case). This is shown in Algorithm 2.

From this point on, we can run both plugins on the CP and CS, so messages are generated automatically in sequence and their Websocket packets are sent to each other following the defined behaviour. In specific, we want to analyse how shares from different transactions interleave over time. Based on Algorithm 1, we run a CP simulator that performs ten consecutive transactions (of ten

Algorithm 2 Secret sharing approach: CS plugin

```
function OnDataTransfer($share$,$transactionId$)
    Save $share$ tagged with $transactionId$;
    if number of shares for $transactionId$ =2 then
        CombineShares;
    end if
end function
```

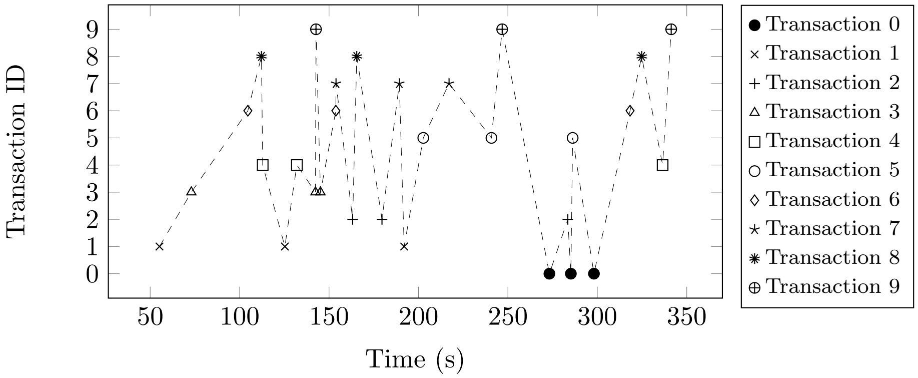

Figure 2: Time and order of transactions when sending shares

seconds), whose meter values are sent through three different shares that can take up to five minutes to reach the CS. After recording the transaction ID of each message the CS receives, we have generated the plot in Fig. 2, which shows the order and time in which the thirty shares arrive at the CS. As we can see, for the same transaction ID there are three independent shares which are disorderly transferred with the ones from other transactions. Therefore, the probability for an attacker to find two linked shares decreases with respect to a hypothetical scenario where the shares are sent sequentially and at regular time intervals.

5 Discussion and future work

We have designed an approach that preserves the privacy when sending smart meters data in OCPP, which has been validated by means of a test case in a simulator, as described in Section 4. This is a feasible countermeasure to MITM attacks and represents an alternative to traditional approaches based on encryption. The division of secrets into shares imposes a harder challenge on attackers, since they find themselves in the situation of having to gather correct shares, which may take an unknown length of time to be sent through the compromised channel. In addition, it avoids the destruction of data, as the information can be reconstructed even though the attacker disrupts some packets (which depends on the threshold of the scheme). Furthermore, the security level can be enhanced as we increase the values of all parameters involved in this particular protocol. Firstly, the number of transactions taking place in a CP, which boosts the shares' interleaving effect; secondly, the number of shares in which a secret is split, as the attacker needs to gather more correct shares; thirdly, the maximum time a share can take to be sent. However, the latter variable must be consistent with operator policies, which may impose time restrictions on performing near real-time monitoring operations that require a rapid response from the smart meters.

It is also interesting to consider some potential issues that this approach might still have to face. Currently, OCPP defines an offline functionality in its standard in case the charge point cannot communicate with the central system, holding the consumption data in a local cache. However, this behaviour could be impaired in the case of the secret sharing technique, because it requires more space to save all the secret packets in the CP, and a potential saturation of the storage system could lead to share losses, which may be seized by the attacker to perform charging sessions that are finally not recorded. For this reason, our ongoing work is currently revolving around the introduction of supporting mechanisms that replicate data or balance the load when receiving the shares, in order to avoid losses and decrease the overhead in the communications. Future work will also involve a statistical analysis of this scheme to accurately determine the optimum value for all the parameters involved (i.e., number of secrets and maximum time for sending shares) to ultimately achieve the best security level that also ensures the greatest level of reliability.

There have been previous approaches that concentrate on identifying the security shortcomings of OCPP ([15],[16]), but, to the best of the authors' knowledge, this is the first study that proposes a practical solution besides the recommendation of high level security services to be contemplated for future versions of the standard. However, this solution can be considered as a starting point when addressing security in OCPP. There are other security protocols that could be introduced in compliance with accepted security standards. With respect to industrial automation and automatic meter reading protocols, IEC 62351 is a reference. It provides guidelines for introducing suitable techniques that provide authenticity, confidentiality and integrity, as well as control access mechanisms. The standard comprises eleven parts that describe multiple security tools and recommended architectures. Related to the problem addressed in this paper, IEC 62351-3 suggests specific TLS profiles for TCP/IP communications, which can be taken into consideration for OCPP [17]. Namely, key exchange algorithms like Diffie-Hellman (DH) or RSA, which can be used for digital signature together with DSS (Digital Signature Standard). Encryption algorithms such as RCA-128, 3DES (Triple-Data Encryption Standard) or AES with 128 or 256 bits of key size can be used and lastly, SHA as the secure hash algorithm.

However, these mechanisms may not be enough to provide security in the communication channels with TLS, as argued in this paper. It is necessary to establish VPNs between all the peers in the EV infrastructure and configure firewalls and intrusion detection systems, with the ultimate goal of providing a trusted environment for both users and operators that boosts the adoption of these technologies, as stated in IEC 62351-7. Whereas parts 3 to 6 focus on the security of communications and applications, part 7 addresses the security of end devices (e.g., intelligent electronic devices, remote terminal units, gateways, data concentrators, etc.)[18]. Lastly, IEC 62351-8 applies role-based access control (RBAC) to power systems, which restricts the access to system resources to authorised users, according to their roles and associated permissions. Altogether, this provides flexibility to design a variety of security policies, resulting in a reduction of the risk of suffering an insider attack.

6 Acknowledgements

This work has been partially supported by the Spanish Ministry of Economy, Industry and Competitiveness through the research project SADCIP (RTC-2016-4847-8) and the research project SMOG (TIN2016-79095-C2-1-R). The first author is supported by the Spanish Ministry of Education through the National F.P.U. Program under Grant Agreement No. FPU15/03213.

References

[1] R. R. Mohassel, A. Fung, F. Mohammadi, and K. Raahemifar, "A survey on advanced metering infrastructure," *International Journal of Electrical Power & Energy Systems*, vol. 63, pp. 473–484, 2014.

[2] F. Van Den Broek, E. Poll, and B. Vieira, "Securing the information infrastructure for ev charging," in *International Conference on Wireless and Satellite Systems*. Springer, 2015, pp. 61–74.

[3] O. C. Alliance, "Open charge point protocol 1.6, 2015," http://www.openchargealliance.org/, last retrieved in July 2016.

[4] J. Schmutzler, C. A. Andersen, and C. Wietfeld, "Evaluation of ocpp and iec 61850 for smart charging electric vehicles," in *Electric Vehicle Symposium and Exhibition, 2013 World*. IEEE, 2013, pp. 1–12.

[5] Y. Sheffer, R. Holz, and P. Saint-Andre, "RFC7457: Summarizing Known Attacks on Transport Layer Security and Datagram TLS," Tech. Rep., 2015.

[6] C. Alcaraz, J. Lopez, and S. Wolthusen, "OCPP protocol: Security threats and challenges," *IEEE Transactions on Smart Grid*, 2017.

[7] J. JIA and Z. XUE, "The principle and prevention of ssl man-in-the-middle attack [j]," *China Information Security*, vol. 4, 2007.

[8] J. E. Rubio, C. Alcaraz, and J. Lopez, "Selecting privacy solutions to prioritise control in smart metering systems," in *11th International Conference on Critical Information Infrastructures Security*, 2016.

[9] W. of IEC TC57, "IEC 62351," retrieved: 2016-07-13. [Online]. Available: http://www.iec.ch/smartgrid/standards/

[10] F. Callegati, W. Cerroni, and M. Ramilli, "Man-in-the-middle attack to the https protocol," *IEEE Security and Privacy*, vol. 7, no. 1, pp. 78–81, 2009.

[11] C. Jouvray, G. Pellischek, and M. Tiguercha, "Impact of a smart grid to the electric vehicle ecosystem from a privacy and security perspective," in *Electric Vehicle Symposium and Exhibition (EVS27), 2013 World.* IEEE, 2013, pp. 1–10.

[12] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[13] GIR (Giaume Industrie & Recherche), "ocppjs: An experimental ocpp simulator," http://www.gir.fr/ocppjs/, last retrieved in October 2016.

[14] "Node.js," https://nodejs.org/, last retrieved in October 2016.

[15] M. A. Mustafa, N. Zhang, G. Kalogridis, and Z. Fan, "Smart electric vehicle charging: Security analysis," in *Innovative Smart Grid Technologies (ISGT), 2013 IEEE PES.* IEEE, 2013, pp. 1–6.

[16] R. Falk and S. Fries, "Securely connecting electric vehicles to the smart grid," *Int. Journal on Advances in Internet Technology*, vol. 6, no. 1, 2013.

[17] C. Alcaraz, J. Lopez, and S. Wolthusen, "Policy enforcement system for secure interoperable control in distributed smart grid systems," *Journal of Network and Computer Applications*, vol. 59, pp. 301–314, 2016.

[18] R. Schlegel, S. Obermeier, and J. Schneider, "Assessing the security of iec 62351," in *Proceedings of the 3rd International Symposium for ICS & SCADA Cyber Security Research.* British Computer Society, 2015, pp. 11–19.