Pre-print version of paper submitted to W-P2DS´18 - 1st Workshop on Privacy by Design in Distributed Systems (co-organized with Eurosys 2018)

Securing Smart Metering applications in Untrusted Clouds with the SecureCloud Platform

Rodrigo J. Riella, Luciana M. Iantorno, Laerte C. R. Junior, Dilmari Seidel
Institutos LACTEC, Av. Comendador Franco, 1341, Curitiba, Brazil
riella,luciana.iantorno,laerte,dilmari@lactec.org.br

Keiko V. O. Fonseca, Luiz Gomes-Jr, Marcelo O. Rosa
Universidade Tecnologica Federal do Parana, Av. Sete de Setembro, 3165, Curitiba, Brazil
keiko,mrosa,lcjunior@utfpr.edu.br

Abstract

Data security in smart metering applications is important not only to secure the customer privacy but also to protect the power utility against fraud attempts. Usual deployment of metering applications rely on the power utility infrastructure, assuming its Advanced Metering Infrastructure (AMI) as trustworthy. This paper describes the design and deployment of a smart metering system focusing on the security of the AMI (smart meters, data aggregator on the field, Metering Data Collection system and metering database) considering the data processing on untrusted clouds. We discuss one use case of the SecureCloud project, an ongoing project that investigates how security and privacy requirements of smart grid applications can be met with a secure cloud platform based on Intel SGX enclaves. The paper describes the components of the advanced metering system as well as the security approach adopted to meet its requirements. A smart metering application has been prototyped in the SecureCloud platform and the integration challenges are discussed from the perspectives of security, privacy and scalability.

Keywords Smart Metering, power distribution grids, cloud computing, Intel Software Guard eXtension (SGX)

1. Introduction

The increase of Smart Grid applications in energy distribution utilities poses new demands for data processing and storage as well as data privacy and security. The migration from old energy metering systems, based on manual meter reading, to the concept of Advanced Metering Infrastructure (AMI) sets new requirements for communication systems, data storage and near real time processing of Big Data. Smart metering data also have particular requirements of data security and privacy depending on the stakeholder goal [13]. For example, privacy requirements of applications like billing differ from those of energy demand monitoring and requirements of applications for controlling customer energy consumption differ from those of fraud detection, at the same time all of them use the same metering measure databases.

Assuming cloud computing can assure high scalable data storage and processing, Smart Grid applications can benefit from cloud computing solutions as long as the provided security and privacy meet the requirements of the application. In particular, this paper focuses on applications for energy distribution utilities, mainly for those with a large number of customers, handling large volume of data from applications such as Metering Data Collection (MDC), billing and Metering Data Management (MDM).

AMI is part of the Smart Grid critical infrastructure of an energy distribution utility: from the communication system of a meter in the field until the metering data storage and processing, strict requirements of privacy and security should be met as discussed by several studies [5, 14, 22, 24] and guidelines [16, 23]. In special, for the Brazilian energy metering market, security is a concern due to the high level of fraud attempts (classified as non technical losses) in energy metering systems [10]. According to the Brazilian Association of Electrical Energy Distributors, in 2015, 5.75% of total delivered energy (27 TWh) was lost due to energy theft and fraud in the metering systems [9, 12]. Currently, attacks are mostly focused in metering tampering and in irregular connections but it can migrate to other components, such as communication, concentrators, Metering Data Collection (MDC) and billing systems with the increase in use of AMI.

Nowadays, in the Brazilian energy market, most of the industrial and commercial consumers are using AMI systems. This group represents only 7.7% of total

customers but accounts for more than 55% of total energy consumption [12]. Assuming the increase of AMI usage, the adoption of cloud

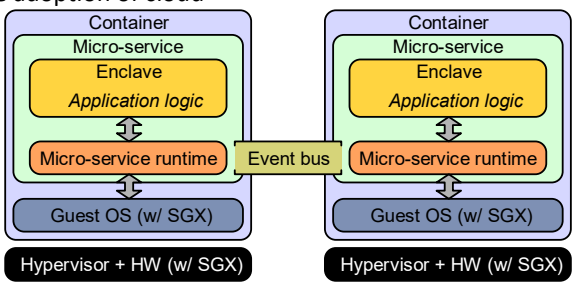

Figure 1. SecureCloud applications

computing applications for utilities can be economically feasible if data security and privacy requirements of Smart Grid applications could be met.

The SecureCloud platform aims at providing solutions to critical infrastructures, allowing secure processing of sensitive data within untrusted cloud environments[18]. These solutions can be designed to meet the requirements of privacy and security of Authentication and MDC applications, as well as secure storage schemes for metering databases, for example. The platform uses a new cryptographic hardware from modern Intel CPUs, named Software Guard Extensions (SGX) [2, 8] to build enclaves (protected memory areas of code execution). SGX aims to provide integrity and confidentiality guarantees to secure sensitive computation even when all the privileged software (kernel, hypervisor, etc.) are potentially malicious[8].

One use case of the SecureCloud Project refers to a secure smart metering application migration to untrusted clouds making use of the SecureCloud platform. We are particularly interested on assessing the challenges faced by software developers in this migration process.

This paper reports our achievements as follows: Section 2 describes to the SecureCloud infrastructure, its architecture and basic concepts; Section 3 describes the Secure AMI application scenario, its requirements, components and challenges; the details of the deployment are described at Section 5 and the integration to the SecureCloud platform at Section 6. The final section discusses the deployment details and the approaches used to overcome the challenges faced so far.

2. SecureCloud Infrastructure

Secure remote computation refers to executing software on a remote computer owned and maintained by an untrusted party, with integrity and confidentiality guarantees. In general, secure remote computation is still an open problem.

Figure 1 shows the baseline infrastructure of the SecureCloud Platform. An application consists of a set of microservices [15] connected via an event bus [18]. The application logic of each micro-service lives within an enclave. The micro-service runtime exists outside of the enclave. These runtime functions only access encrypted data. Encryption
and decryption of this data is performed automatically and transparently within the enclave. This approach limits the amount of code added to the Trusted Code Base (TCB).

To deploy a micro-service, the SecureCloud platform offers secure containers on top of the untrusted stack of the cloud provider in order to add confidentiality and integrity to Docker containers [1]. This enables system administrators to build secure container images within a trusted environment and to run them in an untrusted cloud. The creation of secure containers for the SecureCloud Platform is simplified by a Secure Linux Container Environment, called SCONE [3] that secures existing applications with SGX.

SCONE provides the micro-service with an interface based on external system calls, which are shielded from attacks. To protect itself from user space attacks, SCONE performs sanity checks and copies all memory-based return values to the inside of the enclave before passing the arguments to the micro-service. SCONE further (i) transparently encrypts and authenticates data that is processed via file descriptors, and (ii) provides acceptable performance by implementing tailored threading and an asynchronous system call interface[18]. SCONE integrates with existing Docker environments, and ensures that secure containers are compatible with standard containers.

SecureCloud also proposes SGX-based secure data communication between containers that protects messages exchanged by encrypting them inside of enclaves via an event bus. Encryption and decryption are automatically performed within enclaves. Decryption keys are stored securely within enclaves only. Authentication is also provided, as well as additional security properties of the bus, such as ensuring freshness of the delivered messages (ensured by the event bus stubs that run inside an enclave) [6]. Micro-services communicate with external clients and services through secure communication channels [4].

[1] https://www.docker.com/what-docker

3. Secure AMI Application

The smart metering application system consists of a set of smart meters connected to an aggregator, which is responsible for providing a backhaul connection to the MDC and the metering database. We explore the deployment of a Secure AMI application based on untrusted clouds. Figure 2 presents such application showing the MDC and the database as cloud applications.

In the Secure AMI application, the MDC remotely communicates with the smart meter (SM) through the aggregator (used to bridge the backhaul network into the wireless mesh network [1]), to convert the cryptography scheme to connect to the SMs through its protocol[7], and to insert this information as payload, so it can be sent to the meter. The MDC Application periodically sends requests to the SM in order to collect information stored in the database. Additionally, the aggregator is responsible for the wireless network management.

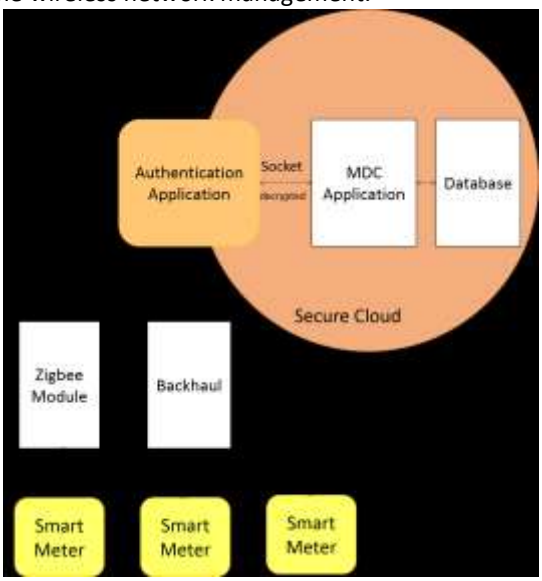

Figure 2. System architecture including cryptography and authentication scheme

3.1 Security Requirements

Based on the scenario and the use case outlined above the following security requirements can be derived:

- Confidentiality (including privacy)
  - Detailed load signature must be kept confidential. Only the affected consumers should be able to see their consumption data in full detail.
  - Different utility's computing systems should be allowed to access the data in different aggregated manner, only to reinforce user's privacy (different computing systems should have different data access privileges).
- Integrity (including non-repudiation)
  - The integrity as well as the accountability (including non-repudiation) must be ensured. This is critical due to the billing and load management.
- Availability
  - Some availability needs to be ensured but there is no warranty for real-time access to the necessary data.

4. AMI Implementation

4.1 Communication

The meter connects to the aggregator to synchronize, authenticate, and exchange data. The exchanged information is forwarded from the aggregator to the Authentication Application which informs the MDC about the available meters and sets the information needed to fill the phasor report. The phasor report is periodically sent (every 15 minutes) with all meter information required by the energy utility company. The communication between the meter and the MDC follows ABNT (Brazilian Association of Technical Standards) NBR 14522 [11] protocol, which is packed inside a proprietary protocol [7] defined for electricity metering system.

The current implementation uses this proprietary protocol to manage communication between the aggregator and with multiple devices in large networks: a preliminary negotiation between the aggregator and a meter defines the intent for data requests, and a limited number of "tokens" at a time (less than the number of devices) is used to coordinates the reception of answers from meters. The meter response is validated by the Authentication Application and the requested data is loaded into the database by the MDC Application.

4.2 Smart Meter

The smart meters were developed to enhance the security via an embedded Trusted Platform Module (TPM) and also to add advanced features. The embedded TPM implements both symmetric cryptography (AES) and digital signature based on asymmetric algorithms (RSA), following the recommendations of the Guidelines for Smart Grid Cyber Security from NIST [23] and following the PCB (Public Key Infrastructure) for authentication. They were deployed as single and polyphase energy meters, following the Brazilian standards for energy meters to residential and small commercial consumers, supporting AMI structure.

The main characteristics of this smart meter are its use of wireless communication through Mesh Networks, the internal RF antenna, its capability of energy measurement in four phase quadrants, and its ability of remote energy cut and reclosing operations. For remote communication, a RF module compliant with the IEEE 802.15.4 standard [2] implementing a ZigBee stack was also included.

4.2.1 Communication Interface

All smart meters manage their wireless communication with the aggregator (ZigBee stack mesh network) by acting as a router to the aggregator using a proprietary protocol [7]. Additionally they have their own digital certificates used on the authentication process. After the authentication process, all ABNT commands include a digital signature that has to be validated by both sides to assure a successful command execution. The digital signature configuration is hard coded in the meters and follows [17, 20]

After receiving the digital certificate, the smart meter is able to be installed in the field and to start the communication with the MDC. Figure 3 shows the temporal structure of communication between the meter and server.

4.3 Aggregator

The aggregator is the interface between the Authentication Application, the MDC and the smart meters. It is responsible for communication protocol conversion and packet routing, acting as a gateway between the Smart Meters and the MDC system. Its main components are a ZigBee coordinator communication module and a *Backhaul* application.

[2]https://standards.ieee.org/findstds/standard/802.15.4-2015.html

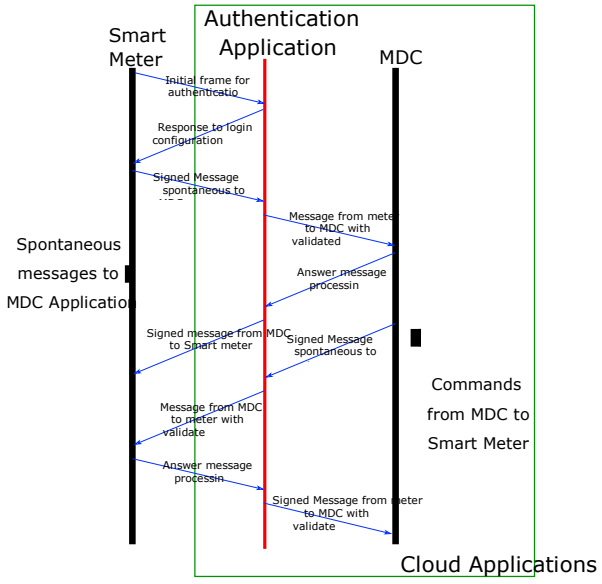

Figure 3. Temporal structure of communication between meter and MDC

The (the *Backhaul* software) is responsible for sending routing information from the meter to the Cloud Application and vice-versa, converting protocols. It also manages the list of available meters and synchronizes the messages between the MDC application and the smart meters.

5. Secure AMI Structure

The data security and authenticity is guaranteed by two layers: (1) an authentication scheme between the smart meters and the Authentication Application, which is encapsulated inside a container based on SecureCloud platform, and (2) a symmetric cryptography layer, which connects smart meters, the aggregator and the Authentication application, and is based on both Advanced Encryption Standard (AES)[19] and the authentication processes.

5.1 Authentication

The Authentication Application should assure that only authenticated meters can communicate with the MDC. Requests from non-authenticated devices should be ignored.

When a meter connects to the network, after receiving the synchronism response, it sends an Authentication Request with its digital certificate to the Authentication Application. The Authentication Application verifies the received certificate with the meter's public key and sends an authentication response. After the authentication, all data requests and responses exchanged between the meter and the application are signed.

5.2 Encryption

By default, ZigBee communication is not encrypted, so any device can connect to the network. A transport key message sent to the device has to be filled with a null key to indicate that the device should use a key saved in its memory. After that, all the message is encrypted.

The data exchanged in the network is encrypted using the symmetric cryptography AES-128 encryption algorithm (limited by Zigbee module). The meter TPM and the ZigBee module are able to process this type of cryptography. After decrypting the Zigbee messages, the aggregator communication module forwards the received information

to the *Backhaul* where the data is encrypted again with the AES256 encryption algorithm.

The Authentication Application decrypts, adds a digital signature to the received data and sends it to the MDC Application. Assuming an Authentication Application located in a secure platform, the data can be sent to the MDC Application without encryption. All data originated inside the MDC Application are also encrypted by the Authentication Application towards the aggregator.

6. Cloud Applications

The Cloud Applications consist of an Authentication Application, a MDC Application and a Database. These applications are divided into micro-services, executed on SGX containers and connected to each other via an event bus as illustrated in Figure 1.

To securely execute these applications inside SGX enclaves, three main issues must be addressed: (1) the secure data processing within container, (2) the secure data communication between containers and (3) the secure data communication to the outside [6]. Such issues are transparently addressed by SecureCloud platform since it leverages the SCONE secure container mechanism [3] to run each application inside a secure environment, use a REST communication protocol via HTTP between aggregator and the Authentication Application, and use enclave-terminated TLS connections [4].

6.1 Authentication Application

The authentication Application connects the aggregator and the MDC ensuring the authenticity of each smart meter connected to the MDC. Its first task is to establish a secure connection between each Smart meter and the MDC, using the aggregator as a bridge. Then, it verifies the certificate sent in the authentication request, validates it and answers with an authentication response. After the authentication process is successfully completed, the authentication application connects by socket to the MDC application.

Once the connections are complete, the authentication application keeps both connections opened and becomes the bridge to encrypt and decrypt messages but also the verifier through the validation of the messages from its signature verification. It receives encrypted messages from aggregator and send them decrypted to MDC; or it receives decrypted data messages from MDC and sends encrypted data messages to the aggregator.

There is a main process to receive the connections from the aggregator. For each connection a new thread to control the connected meter is created.

6.2 MDC Application

The MDC application consists of a TCP/IP server that opens communication channels with the authentication application to extract data from meters connected to the aggregator.

Every 15 minutes, the MDC application sends a request to the aggregator in order to collect data readings from the meters. The main function of the MDC application waits for new connections. For each new connection, a new reading thread is created and starts requesting data. A timer is used to control when the messages should be sent. When a message request is sent, the MDC waits for the answer of this command before sending the next message request. When a response is not received (timeout), the MDC should send a request retry. Upon a message reception, after the extraction and verification of the necessary data, the data is stored into the Database ”phasor” table. The data comprises information about voltages, currents, power and voltage angles or demands and totalizators from the meters.

6.3 DATABASE and CASCA/DB

After the authentication process, the MDC periodically extracts the ”phasor” data from the smart meter and input these data in a database. This databased is managed by our secure data management solution, CASCA/DB, described next.

The use case described in this paper has strict security requirements for communication and data storage. In order to meet them while also simplifying the application development, we implemented and employed an application framework that encapsulates secure communication (CASCA micro-service) and data management functionalities (CASCA/DB micro-service).

The framework named Customizable Adapter for Secure Cloud Applications (or CASCA) was conceived in order to allow quick development of secure micro-services. It provides a pool of threads working to execute configurable tasks, a logging system, and secure sockets (SSL or TLS) for TCP/IP communications. All these functionalities are present and requested by several system servers. The framework is also implemented over SCONE (in order to easily support code attestation) and TaLoS (in order to have all TCP/IP communications ending inside CASCA enclaves).

The data management solution (CASCA/DB) that completes the framework is based on the CASCA communication services and is itself implemented as a micro-service. CASCA/DB offers an SQL-based querying system that stores data on a separate secure key-object distributed service. The secure key-object service is called

ChocolateCloud, [21], and is also part of the SecureCloud project. CASCA/DB is responsible for translating SQL queries from the applications into key-value API calls to ChocolateCloud. The stored data is made available for use by other applications like Metering Data Management and billing systems.

For the smart meter cloud application, only CREATE TABLE, SELECT, and INSERT statements were supported by CASCA/DB. For obvious security reasons, the translating engine runs inside SGX and all statements are transmitted over TLS connections. Also the communication between CASCA/DB and the key-object distributed server uses TLS connectors.

7. RESULTS AND CONCLUSIONS

The use case specification was a joint work (secure cloud provider, application developer's team, micro-service developer's team, energy utility managers). Despite the expertise of the team, the deployment process has been facing several challenges. For example, application developers were often always unaware of limitations of non supported libraries. Also, performance issues related to the database population (network delays to populate a database in Europe with metering data collected in Brazil and code updates of the database storage server) had also impacted on the deployment. Such challenges are expected when integrating next-generation technologies. The initial issues are being addressed and the processes streamlined. The described applications of the use case are now fully operational and under performance tests. We are also addressing the SecureCloud platform impact on deploying and executing privacy algorithms chosen according to previously defined role access criteria mapped to the application (for example, billing or energy demand monitoring) or user.

This document described the implementation process of the monitoring and control applications developed as use case of an AMI system for the validation of a SecureCloud platform including secure data communication between real smart meters and the MDC and metering database which are implemented as cloud applications. The smart metering application was implemented and integrated with the SecureCloud platform to evaluate its performance in real applications. The cloud platform and the application were independently developed. Its integration are currently in a validation phase.

The use of the CASCA/BD framework played an important role in the development process. SGX is a new technology that introduces many hardware-level aspects that make the coding of secure application more complex. CASCA/DB abstracts most of these complexities, streamlining the development by providing efficient communication and data management services.

Acknowledgements: EU-BR SecureCloud project has been receiving funds granted from the 3rd EU-BR Coordinated Call (Brazilian Ministry of Science, Technology and Innovation, MCTIC/RNP, BR grant agreements # 2550, 2549, 2553, 2552 and 2568) and European Unions Horizon 2020 research and innovation programme - EU grant agreement # 690111). The project is also supported by the Swiss State Secretariat for Education, Research and Innovation (SERI).

References

[1] ZigBee Alliance. 2006. Zigbee specification. (2006).

[2] Ittai Anati, Shay Gueron, Simon Johnson, and Vincent Scarlata. 2013. Innovative technology for CPU based attestation and sealing. In *Proc. 2nd Intl. Workshop on Hardware and Architectural Support for Security and Privacy*.

[3] S. Arnautov, B. Trach, F. Gregor, T. Knauth, A. Martin, C. Priebe, J. Lind, D. Muthukumaran, D. O'Keeffe, M. Stillwell, D. Goltzsche, D. Eyers, R. Kapitza, P. Pietzuch, and C. Fetzer. 2016. SCONE: Secure Linux Containers with Intel SGX. In *OSDI*.

[4] Pierre-Louis Aublin, Florian Kelbert, Dan O'Keeffe, Divya Muthukumaran, Christian Priebe, Joshua Lind, Robert Krahn, Christof Fetzer, David Eyers, and Peter Pietzuch. 2017. *TaLoS: Secure and Transparent TLS Termination inside SGX Enclaves*. Technical Report. Imperial College London.

[5] Todd Baumeister. 2010. *Literature Review on Smart Grid Cyber Security*. Technical Report CSDL-10-11. Department of Information and Computer Sciences, University of Hawaii, Honolulu, Hawaii 96822. http://csdl.ics.hawaii.edu/techreports/2010/10-11/10-11.pdf

[6] F. Campanile, L. Coppolino, S. DAntonio, L. Lev, G. Mazzeo, L. Romano, L. Sgaglione, and F. Tessitore. 2017. Cloudifying Critical Applications: a Use Case from the Power Grid Domain. *Comput. Surveys* (2017).

[7] A. Canestraro, A. A. Barbiero, G. B. Wolaniuk, L. M. Iantorno, R. J. Riella, and D. Ribera. 2015. Proposal of a new protocol for smart metering. In *2015 IEEE PES Innovative Smart Grid Technologies Latin America (ISGT LATAM)*. 774–779. DOI:http://dx.doi.org/10.1109/ISGT-LA.2015. 7381253

[8] Victor Costan and Srinivas Devadas. 2016. *Intel SGX Explained*. Technical Report.

[9] Associac¸ao Brasileira de Distribuidores de Energia Eletrica. 2017.´ Furto e Fraude de energia. http: //www.abradee.com.br/setor-de-distribuicao/ perdas/furto-e-fraude-de-energia. (2017). Accessed: 2017-03-30.

[10] Rubens Alexandre De Faria, Keiko V Ono Fonseca, Bertoldo Schneider, and Sing Kiong Nguang. 2014. Collusion and

fraud detection on electronic energy meters-a use case of forensics investigation procedures. In *Security and Privacy Workshops (SPW), 2014 IEEE*. IEEE, 65–68.

[11] Associacao Brasileira de Normas Tecnicas. 2007. NBR 14522: Data Exchange for electricity metering systems. (2007).

[12] Empresa de Pesquisa Energtica. 2016. *2016 Statistical Yearbook of electricity, 2015 baseline year*. Technical Report. Accessed: 2017-03-30.

[13] Benjamin Fabian, Seda Gurses, Maritta Heisel, Thomas San-¨ ten, and Holger Schmidt. 2010. A comparison of security requirements engineering methods. *Requirements engineering* 15, 1 (2010), 7–40.

[14] M. A. Faisal, Z. Aung, J. R. Williams, and A. Sanchez. 2015. Data-Stream-Based Intrusion Detection System for Advanced Metering Infrastructure in Smart Grid: A Feasibility Study. *IEEE Systems Journal* 9, 1 (March 2015), 31–44. DOI:http: //dx.doi.org/10.1109/JSYST.2013.2294120

[15] Christof Fetzer. 2016. Building critical applications using microservices. *IEEE Security & Privacy* 14, 6 (2016), 86–89.

[16] M. Harvey, D. Long, and K. Reinhard. 2014. Visualizing NISTIR 7628, Guidelines for Smart Grid Cyber Security. In *2014 Power and Energy Conference at Illinois (PECI)*. 1–8. DOI: http://dx.doi.org/10.1109/PECI.2014.6804566

[17] Burt Kaliski. 1998. PKCS# 7: Cryptographic message syntax version 1.5. (1998).

[18] Florian Kelbert, F. Gregory, R. Pires, Stefan Kopsell, Marcelo¨ Pasin, A. Havet, V Schiavoni, Pascal Felber, Christof Fetzer, and Peter Pietzuch. 2017. SecureCloud: Secure Big Data Processing in Untrusted Clouds. In *DATE 17, Design, Automation and Test in Europe*.

[19] Frederic P. Miller, Agnes F. Vandome, and John McBrewster. 2009. *Advanced Encryption Standard*. Alpha Press.

[20] Magnus Nystrom and Burt Kaliski. 2000. *PKCS# 10: Certification request syntax specification version 1.7*. Technical Report.

[21] M Sipos, Patrik Jnos Braun, Daniel E. Lucani, Frank H. P. Fitzek, and Hassan Charaf. 2017. On the Effectiveness of Recoding-based Repair in Network Coded Distributed Storage. *Periodica Polytechnica.Electrical Engineering and Computer Science* 61, 1 (2017), 12–21. https://search.proquest.com/docview/1875386511?accountid=26636 Copyright - Copyright Periodica Polytechnica, Budapest University of Technology and Economics 2017; Last updated - 2017-03-09.

[22] F. M. Tabrizi and K. Pattabiraman. 2014. A Model-Based Intrusion Detection System for Smart Meters. In *2014 IEEE 15th International Symposium on High-Assurance Systems Engineering*. 17–24. DOI:http://dx.doi.org/10.1109/HASE.2014.12

[23] T. L. Brewer V. Y. Pillitteri. 2014. *Guidelines for Smart Grid Cybersecurity Rev1*. Technical Report. NIST Interagency/Internal Report (NIS-TIR). https://www.nist.gov/publications/ guidelines-smart-grid-cybersecurity

[24] Y. Yan, R. Q. Hu, S. K. Das, H. Sharif, and Y. Qian. 2013. An efficient security protocol for advanced metering infrastructure in smart grid. *IEEE Network* 27, 4 (July 2013), 64–71. DOI:http://dx.doi.org/10.1109/MNET.2013. 6574667