Secure Cloud Processing for Smart Meters Using Intel SGX

Marcos V. M. Araújo[1], Charles B do Prado[2],
Luiz F. Rust C. Carmo[1,2], Alvaro E. R. Rincón[1], Claudio M. Farias[1]
Programa de Pós-Graduação em Informática – Federal University of Rio de Janeiro
CCMN/NCE,– Rio de Janeiro – RJ – Brazil[1]
National Institute of Metrology, Quality and Technology (INMETRO)
Xerem, Duque de Caxias – RJ – Brazil[2]
{marcos.araujo,alvaro.robles}@labnet.nce.ufrj.br, {rust,claudiofarias}@nce.ufrj.br, cbprado@inmetro.gov.br

Abstract. *Smart Grids are power grids that use Information and Communication Technologies to improve the power generation, transmission and distribution. The main characteristic on Smart Grids is the ability to process and transmit a huge data information in real-time. However, that rise a new security challenge on confidentiality and data integrity, due to Smart Grids can be part of hostile environments. For example, using a cloud computing to realize billing and load balance process. Thus, to ensure confidentiality and data integrity of Smart Grids in cloud environments, this paper proposes an application for the protection of billing and load balancing data using the Intel Intel Software Guard extension SGX platform. SGX allow to instantiate protected memory areas called enclaves, preventing subprocesses or other machines within the cloud from accessing the data stored in them.*

1. Introduction

The term Smart Grid (SG) refers to systems that combine the traditional electric power distribution with information technology and telecommunication to compose an efficient system in the distribution, supervision and generation of electric energy. Using these technologies it is possible to monitor, analyze, control the electricity distribution network and communicate with consumers [Barbosa et al. 2016]. Among the devices that make up the SGs, there is an important component to be highlighted, the Smart Meter (SM) that acts in the measurement of consumption data to be processed by the system [Rabieh et al. 2017].

SMs are key components in an SG as they provide mechanisms for data exchange, decision making and action [Mwansa et al. 2018]. They are designed to collect and monitor electrical power consumption data from one point of the SG in order to allow the system to coordinate the distribution and consumption of electric power.

The collection of data made by the SMs generates a large volume, since there are numerous devices connected to the network collecting data from the system. This large volume carries some concerns about how to handle this data load. In addition to the challenge of processing a large volume of data, there is concern about the sensitive data of each customer's electricity consumption. This data is in the possession of the concessionaires, this could imply in possible privacy problems.

Faced with the concerns of collecting and aggregating a large volume of SM data protecting client privacy and ensuring data integrity, this paper presents a proposal for using cloud computing. Through cloud computing it is possible to bypass the difficulties of processing large volumes of data. In the scenario of the SGs, the cloud is used due to

its characteristics of: (*i*) dynamic provisioning of resources; (*ii*) fast elasticity; and (*iii*) constant monitoring.

The use of cloud computing adds to SGs greater stability to process data and provide a more efficient response to the system. Nevertheless, the services provided by the cloud involve some security threats, such as: improper handling of data, untrusted applications, and features in different user-shared environments.

1.1. Proposal

In response to data security concerns and applications, this paper presents an application proposal for secure data processing of SMs, ensuring the following requirements: (*i*) extending the security of SMs from data measurement up to billing; (*ii*) prevent power utilities from being able to know the details of a customer's consumption; (*iii*) allow the load balancing function. Due to the requirements, for safe data processing to occur in a cloud environment the following procedures are proposed: (*i*) sign and encode all SM data; (*ii*) to anonymize the data securely by a trusted party; (*iii*) to use Software Guard Extensions (SGX) technology as it allows to ensure application and data integrity and privacy within the cloud at the processor level.

To ensure the safe processing of large volumes of data efficiently, SGX technology is investigated as it enables applications with selected codes and data to be developed against unauthorized disclosure or modification. Intel SGX makes this protection possible through the use of enclaves, which are protected areas of memory execution. The code of an application can be placed inside an enclave by special instructions of these processors.

Through this work it is possible to improve the security of data from SMs. In this way, it is possible to increase the privacy of customer data that benefit from SGs. In addition, the proposal ensures that the customer's billing process occurs safely and is confirmed by a trusted party. This reliable part ensures that customer billing has occurred correctly, from the process of sensing the data to the aggregation of the same by the utility.

To validate the proposed application, experiments were designed to observe the impact of the processing time on the application caused by the use of the protection of the SGX. The purpose of analyzing the processing time of applications for SGs is essential, since they need to generate information of large volumes data as soon as possible, so that it can make decisions on the balancing of the electric power distribution [Youssef 2017] . Experiments were first developed comparing the execution of functions using and not using the Intel SGX platform to verify the impact of this technology in the execution time of applications. Subsequently, the experiments were constructed using a proposed application in monolithic programming architecture compared to microservices and microservices executed with the protection of the SGX enclaves.

2. Secure cloud processing application for smart meters using Intel SGX

In secure processing application, the data generated by the SMs are sent over the network to the cloud computing environment, where this data is stored and processed. In the proposal of this work, the client will only be able to access in the cloud to his consumption data or billing. The utility will only be able to access anonymized data by the cloud to calculate the load balance of a SG specific area.

Therefore, in order to protect data within the cloud, the secure application approach for data processing of SMs uses Intel SGX technology. This technology assists in data protection as it is possible to reserve private code and data regions, ensuring confidentiality and data integrity while running in a cloud computing.

SGX has the feature of preserving applications and data by: (*i*) allowing developers to partition their applications and place sensitive code and data within an enclave; (*ii*) provide security against software with execution privileges; (*iii*) encrypt the memory used by the enclaves; (*iv*) reduce the attack surface to the hardware. However, Intel SGX presents some limitations presented by the manufacturer, in which they impact on the construction of applications with enclaves like: (*i*) to prevent system calls syscalls; (*ii*) restrict the enclave memory to 128 MB.

Given the Intel SGX limitation, there are some difficulties related to scalability in cloud environments that will receive a large load of SM data, as is the case with the SG scenario. Due to the difficulty, it is proposed to build the application based on microservices running in a computational cloud. The microservice architecture allows to develop applications as set of small services, where each one runs in its own process and communicates through lightweight protocols, as socket, http and https [Aderaldo et al. 2017].

The use of the SGX technology made it possible to add a trustworthy part, in this case INMETRO, to validate the billing process of SMs, since it is the responsibility of metrology authority, INMETRO, to establish what software elements, hardware or data from the measurement system are legally relevant. For INMETRO, all the elements participating in the measurement chain, directly involved, or that in any way interfere in the process of capture, processing and publication of the final result of the client are recognized as legally relevant.

Therefore, the behavior of the proposed secure processing application of this work is presented in Figure 1, in which it is divided into three main components based on the life cycle of sensitive SG data. These components are: (1) data generation; (2) data acquisition; and (3) aggregation of data. The generation of data is given by the SMs, which collect client consumption measurements and send them to the cloud through the data network. The data is received by the data acquisition component, in which it is responsible for checking the integrity of the data and storing in the cloud.

2.1. Data Generation

The data of the SGs are usually originated by collections of consumption of the SMs and sent to the SG. The 1 in data generation, shows the flow of functions that SMs must attend to ensure the confidentiality and integrity of secure data generation.

Display: Responsible for displaying information about accumulated energy consumption. The meters display has the role of serving the client; **Measure:** This function is the main one of SM, in which it collects the client's consumption data to be sent to SG; **Sign:** Each SM has a private key, which is used to sign the data packets. The signature of the package serves to ensure verification of the integrity and authenticity of the measurement data generated of the SM; **encode:** In this operation the input of the encoding function is given by the components of a data packet that will be encoded are: unique identifier, measurement, timestamp, and data signature. The packet is encoded using a cloud public key. In this way, all the data generated of the SMs are signed and encoded;

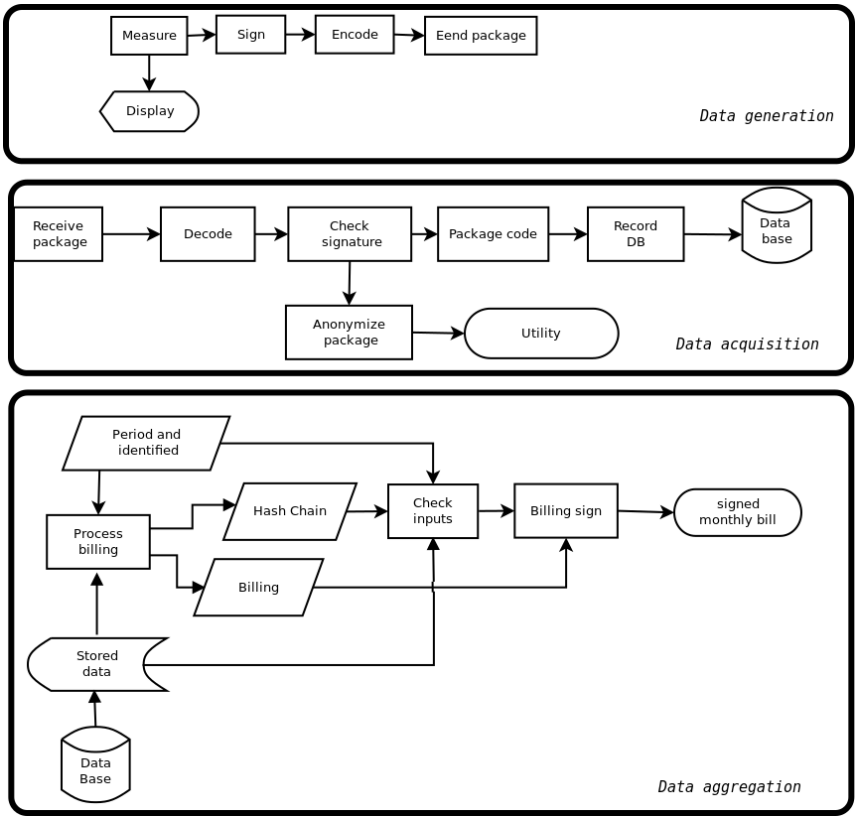


Figure 1. Secure cloud processing for smart meters

Send packet. This service opens a connection to the cloud to send a data packet. It is not necessary the channel is encoded, since the data when they leave the SM are already signed and codified.

2.2. Data Acquisition

The part of the acquisition of SM data is shown in Figure 1, describes the components that are the microservices proposed for a secure acquisition of data by the application in the cloud. Some of these microservices will run within SGX enclaves to ensure privacy and data integrity within the cloud.

Each proposed microservice is designed to ensure safe data processing in the cloud when data is collected by the cloud. The microservices are presented in the following items.

Receive packet: This microservice is responsible for making connections to the SMs and getting the packets into the cloud; **Decode packet:** This microservice is responsible for removing the encoding of the packet generated in the meter. This process occurs within a SGX enclave; **Check signature:** This microservice is responsible for verifying the signature of the package made by SM. This process verifies the integrity and authenticity of the data coming from SMs. This process occurs within the SGX enclave; **Anonymize package:** This microservice is responsible for replacing the unique identifier of each packet arriving in the cloud. As an example of the process, the substitution of the unique identifier, when a packet with identifier ”X” arrives at the microservice, it knows the predefined region that the SM belongs and replaces the unique identifier of the SM by

the region identifier. This operation anonymizes SM and therefore the aggregation is only of the established regions.

2.3. Data aggregation

For the third component of the application, Figure 1, presents the microservices that authenticate the data in process. Some of these microservices will also run within SGX enclaves to ensure privacy and data integrity within the cloud.

The description of each microservice is presented below.

Process billing: This microservice is responsible for calculating the billing collected by the SM and generating a string of hashes of all the packets that have entered the enclave within the billing time interval. This is the algorithm for calculating monthly customer billing. The hash string is to ensure the traceability of the processed packets; **Check inputs:** This microservice is responsible for comparing the hashed string generated by the billing aggregation microservice by doing the same hash chain process with the stored data in order to check if all entries were actually aggregated and no wrong package was entered in the operation; **Billing sign:** This microservice is responsible for signing the invoice after checking all packages and verifying that there were no disagreements in the process. This signature process is the trusted part that validates the aggregation of the correct SM data.

3. Experiments

To analyze the secure cloud processing, two experiments were run. The first was constructed to verify the possible impacts and viabilities of the protection of the enclaves. The second experiment aims to validate the application of secure processing, comparing 3 different programming architectures.

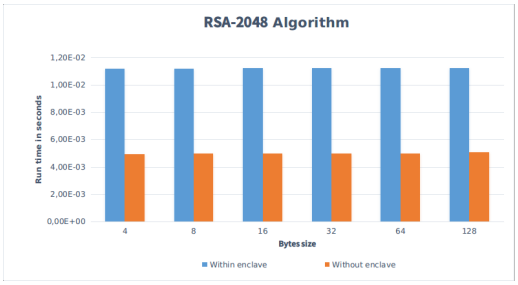
3.1. Enclave validation experiment

This first experiment aims to identify the overhead added by the Intel SGX, since the protection of the SGX creates areas of memory protected by enclaves. For the experiments, the RSA-2048, SHA-256 and SHA-512 algorithms were selected inside and outside a SGX enclave to compare the execution time of these algorithms using the SGX. The environment used in the experiments of the selected algorithms RSA-2048, SHA-256 and SHA-512 are as follows: Operational System: Linux Ubuntu 14.04.5 LTS; Kernel: 4.4.0-36; CPU: Intel Core I7 - 6700 and RAM: 8GB.

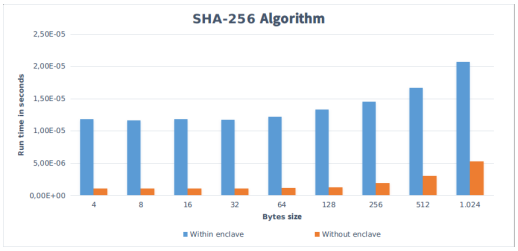
To observe the overhead of the SGX to applications, the algorithms RSA 2048, SHA 256 and SHA 512 were run within enclave and without enclave. The results of the experiments with the algorithms generated the following Figures 2(a), 2(b) and 2(c).

Considering the results presented in Figure 2(a), it can be inferred that for this algorithm the use of the SGX enclaves has a overhead approximately 2.25 times greater at runtime for RSA 2048. It is possible to note that regardless of the amount of Bytes that will be encoded and decoded, the algorithm has a linear execution time.

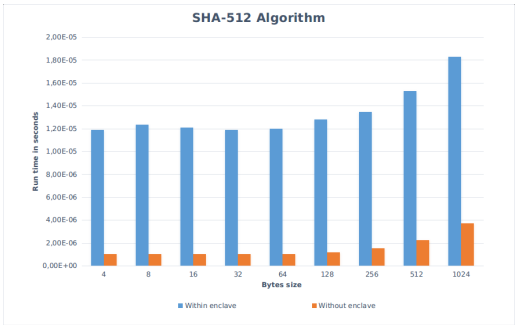
In the case of the algorithm, it is possible to infer that for these hash algorithms the use of the SGX enclaves has a cost approximately 11.6 times higher in the execution time of the algorithms.



(a) RSA 2048



(b) SHA 256



(c) SHA 512

3.2. Case study of the secure processing application in a cloud environment

In the application experiment, the execution time of the application of this work was verified by means of 3 different programming architectures: (*i*) based on monolithic programming, (*ii*) based on microservices and (*iii*) based on microservices running in the SGX enclaves. These architectures were used to compare the use of monolithic applications against microservice applications and the cost of SGX security to microservice applications in the secure cloud environment. The data collected from the applications were the averages of execution times in the data acquisition and data aggregation parts of the application. The environment in which all the experiments with monolithic, microservice and microservice applications with SGX were done, used the following resources: Operacional System: Linux Ubuntu 14.04.5 LTS; Kernel: 4.4.0-87; CPU: Xeon CPU E3-1225 v5 3.30GHz and RAM: 4GB; Database: PostgresSQL 9.6.4.

In the execution of the experiments performed in the 3 applications of safe processing for SMs the following procedures were used to cycle the sensitive data in the 3 applications for an Scenario of the SG.

(*i*) the generation of the SM measurement was given by a random integer; (*ii*) the SM sends a packet as the data in the interval of 1 second; (*iii*) each SM sends a total of 2880 packets. The 2880 packets were used because if the packets were sent every 15 minutes in a month there would be 2880 packets of consumption data; (*iv*) the aggregation of data is done by the application using 2880 packets. In this context, the 2880 packages represent the billing of a customer for a month and the aggregation of billing is given by the sum of the measurement of each package; (*v*) all SMs simulated in the tests used the same asymmetric private key RSA 1024 to sign the data and the same asymmetric public key RSA 4096 to encode the packet data; (*vi*) the execution time of the data acquisition is obtained by the instant that a package is received by the application until it is written to the database; (*vii*) the execution time of the aggregation is obtained by the moment

the query begins the database until the amount of data is calculated, checked, signed and recorded the billing in the database;

The experiments done with secure application were running in two parts: (*i*) data acquisition and (*ii*) data aggregation. The calculation of the average data acquisition execution time will occur upon receipt of the package, processing it and saving it in the database. After running the experiments with the 3 applications of data acquisition was obtained Figure 2

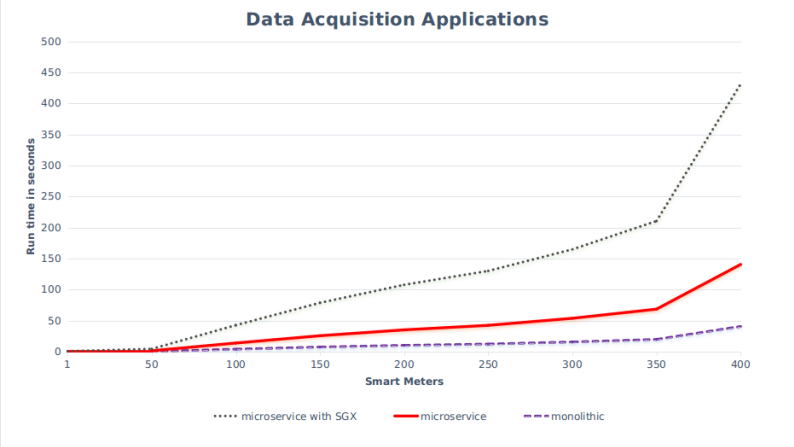


Figure 2. Data acquisition application

In the results presented by Figure 2 it is possible to observe that there was a time of execution of the monolithic application with the application of microservice. This is due to the fact that microservices need to communicate to pass information.

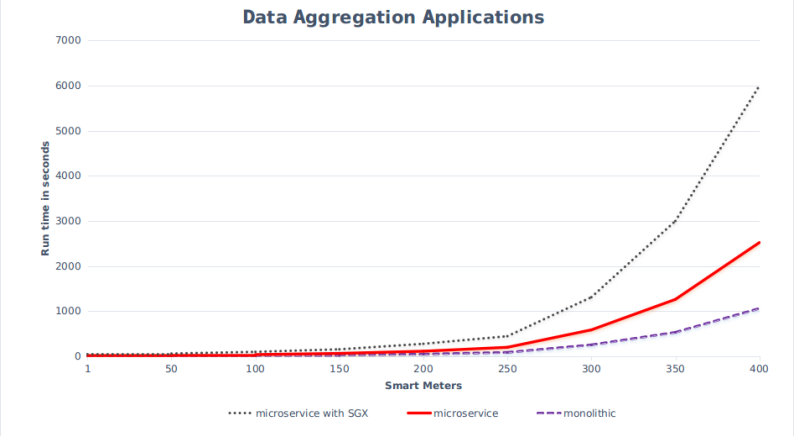


Figure 3. Data aggregation application

4. Conclusion

In order to give answers to these types of applications and data, in particular to the scenario of SG, this work presented a proposal of safe application for processing of data of SM. This application has met the following requirements (*i*) extend the security of SMs from data measurement to billing; (*ii*) to prevent power utilities from being able to know the details of a client's consumption; (*iii*) allow the load balancing function. In

this context, the proposed secure processing application was built with some fundamental principles of information security. As an example in the items below: (*i*) authenticity and irretractability: In order to guarantee these principles, all data coming from SMs are signed by their unique private key, which guarantees the authorship of the data that is identified by SM. In this way, it is feasible to trace the data source, since each SM has its private key; (*ii*) integrity: In addition to verifying the signature of the data by the data acquisition part of the application, in the aggregation the hashed string of the selected packets is calculated. This string is used by the trusted party, INMETRO, to check the integrity of the packets and the correct selected sequence; (*iii*) confidentiality: To ensure the confidentiality of data during transport over the network and also while in the cloud in regions that are not considered secure, the RSA asymmetric cryptography system is used.

Data security is critical when applications in the cloud manipulate sensitive data such as power consumption. In this work, a proposal for safe data processing of SMs was described to protect data of electric power consumption. This proposal enables the strategy of aggregating private data by load balancing and billing functionality in the cloud reliably.

Another contribution of this work is the proposal to add a trustworthy part in the process of acquiring and authenticating data originating from SMs. However, the this is part guarantees the completeness and completeness of the measurement data. Nevertheless, since the use of the SGX also allows to remotely attest the integrity of the application, INMETRO, the country's legal metrology authority, can verify the integrity of the application that performs the aggregation operation of SMs consumption data.

5. Acknowledgment

This work was partially funded by the EU-BRA Secure Cloud project (MCTI / RNP 3º called coordinate) and Capes. It also used features provided by Inmetro, Kontron, Neopath and Inovax for scientific research

References

Aderaldo, C. M., Mendonça, N. C., Pahl, C., and Jamshidi, P. (2017). Benchmark requirements for microservices architecture research. In *Proceedings of the 1st International Workshop on Establishing the Community-Wide Infrastructure for Architecture-Based Software Engineering*, pages 8–13. IEEE Press.

Barbosa, P., Brito, A., and Almeida, H. (2016). A Technique to provide differential privacy for appliance usage in smart metering. *Information Sciences*, 370-371:355–367.

Mwansa, M., Hurst, W., Chalmers, C., Shen, Y., and Boddy, A. (2018). A study into smart grid consumer-user profiling for security applications. *CLOUD COMPUTING 2018*, page 17.

Rabieh, K., Mahmoud, M. M., Akkaya, K., and Tonyali, S. (2017). Scalable certificate revocation schemes for smart grid ami networks using bloom filters. *IEEE Transactions on Dependable and Secure Computing*, 14(4):420–432.

Youssef, K. H. (2017). Power quality constrained optimal management of unbalanced smart microgrids during scheduled multiple transitions between grid-connected and islanded modes. *IEEE Transactions on Smart Grid*, 8(1):457–464.