A Lightweight Authentication Protocol for Advanced Metering Infrastructure in Smart Grid

Luis Fernando A. Roman[1], Paulo R. L. Gondim[1], Ana Paula G. Lopes[1]

Departamento de Engenharia Elétrica – Universidade de Brasília (UnB)
Campus Universitário Darcy Ribeiro – 70910-900 – Brasília – Brasil

lfroman@aluno.unb.br , pgondim@unb.br, anagolembiouski@aluno.unb.br

Abstract *Electric networks have evolved rapidly in recent years due to the integration of new technologies, such as Internet of Things (IoT), cloud computing and cyber-physical systems. This evolution brings optimization in the generation, distribution and consumption of energy, in addition to offering new services to customers, but these new services also bring new challenges in information security. This manuscript proposes a group authentication protocol for key management in an AMI infrastructure integrated with the cloud. The proposed protocol shows improvements in the security and performance index compared to other protocols.*

1. Introduction

The next generation of electrical networks is called Smart Grid (SG), whose Advanced Measurement Infrastructure (AMI) integrates advanced sensors, Smart Meters (SM), monitoring systems, and systems of administration of data. On the other hand, Cloud computation is one of the options that can aid the meeting of AMI requirements. The integration of cloud and AMI leads to the need for an efficient authentication and distribution key scheme that supports the particular characteristics of the AMI network for the protection of data of messages, since some threats (e.g., Denial of Service (DoS), Man in the Middle (MITM) and personification) can destabilize it. These attacks can cause a blackout in cities, altering the customer's billing information or changing the price information [Wan 2014]. Therefore, a protocol that guarantees the confidentiality, integrity and authentication of communication among AMI entities and a cloud infrastructure must be designed.

This article proposes a group authentication and key management protocol that considers an AMI architecture integrated to a private cloud. The protocol is based on groups and uses an anonymous key agreement protocol based on ECDH (Elliptic Curve Diffie-Hellman) for sharing secrets and bilinear pairing towards an efficient simultaneous authentication of a group of devices.

The article is organized as follows: Section 2 addresses some related work; Section 3 introduces the protocol; Section 4 analyses its security properties; Section 5 analyses its performance, comparing it with other protocols; finally, Section 6 summarizes the conclusions.

2. Related Work

This section reports on some relevant studies on the protection of the AMI against computer attacks and preservation of its privacy.

Wan et al. [Wan 2014] proposed a mixture of symmetric and asymmetric cryptography systems based on elliptic curve and bilinear pairing for the creation of a key management scheme, called Scalable Key Management (SKM). The first step is the generation of a session key for a secure point-to-point communication between each SM and the Meter Data Management System (MDMS). Through a tree key creation technique, a Group key that sends messages is broadcasted from MDMS to SM. The scheme does not perform well due to the high computational costs of bilinear pairing operations and high communication costs, since messages must be exchanged between MDMS and each SM for the generation of the session keys.

Nicanfar et al. [Nicanfar 2013] developed Smart Grid Key Management (SGKM) that ensures mutual authentication between SMs and the Security and Authentication Server (SAS) in the SG network using passwords and public key infrastructure (PKI). The authors use of an enhanced version of identity-based encryption (IBC) for a reduction in the key update overhead. The scheme showed the same weaknesses of that designed by Wan et al. [Wan 2014], i.e., high computational costs due to the use of exponentiation and calculation of a high number of Hash operations.

Other references can be found at [Genge 2014], [Ye. 2015] and [Bera 2015].

3. Proposed Protocol

We consider the architecture of an AMI network as shown in Figure 1, with a Cloud Service Provider (CSP) which implements a trustworthy private cloud, and devices, such as Smart Meters (SM) and Aggregators (AG). The channels of communication between SMs and AG and between AGs and CSP are considered **unsafe**. The proposed scheme considers the aggregation of devices into groups and their simultaneous authentication. CSP serves as a trustful authority and the ECDH key agreement protocol is used in the key agreement between AG/SM and the CSP. Secure channels are denoted by arrows and unsecure ones are denoted by dotted arrows in a graph for clarifying the characteristics of the channels through which messages are exchanged.

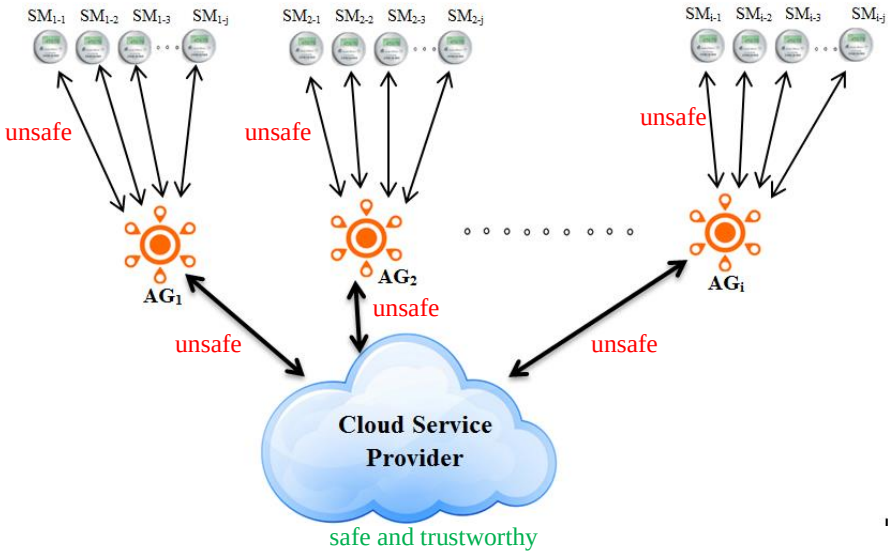

Figure 1. Proposed architecture of AMI in the Cloud

Figure 2 shows an overview of the protocol operation, described as follows:

1. A group of SMs is deployed in a specific area (neighborhood, buildings, etc.) and sends a connection request to the aggregator.
2. The aggregator groups the connection requests and sends them in a group, so that the CSP validates the identities.
3. Once the SM and AG identities have been authenticated, the CSP sends a Broadcast message to the device group (SM and AG). The message contains data for the calculation of the session keys and verification of the CSP authenticity.

The protocol has three phases: initialization, registration and authentication.

1st phase: Initialization
The *CSP* proceeds as follows:

i. SMs, AG and CSP are organized into a binary tree structure, where each of them is a leaf and has an associated SECy secret value derived from the secret values of the nodes above it, similarly to the organization adopted in [Choi 2015].

ii. CSP chooses a random k-bits prime number and generates two elliptic curve groups, G1 and G2 of order p, and a generator point P in G1.

iii. a random number $x_{csp} \in Z_p^*$ is chosen as a private key and the public key is calculated as $PK_{csp} = x_{csp} * P$ for the generation of the master keys of the system;

iv. the group key is calculated and generates a random number g ϵ Z_p^* e according to

$$GK_i = h_2(SEC_{i-1} \oplus SEC_{i-2} \oplus \ldots \oplus SEC_{i-j} \oplus (g * \mathrm{PK_{csp}}))$$

v. parameters {p, P, PK, G1, G2, e, h_1, h_2, h_L, h_R} are published (h_1(.) and h_2(.) are hash functions, h_L(.) and h_R(.) are hash function used for the creation of the secrets of the binary tree node, and e(-,-), is the bilinear pairing function).

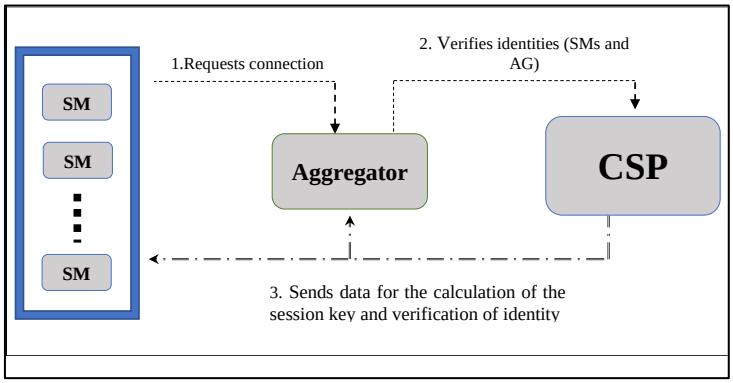

Figure 2. General scheme of the proposed protocol.

2nd phase: Registration
A secure channel is used for the registration of SMs and AGs, whose process is described below:

I. $\begin{matrix} \text{SM} \quad \{\gamma_{SM_{i-i}}, ID_{SM_{i-i}}\} \\ \text{AG} \quad \{\gamma_{AG_i}, ID_{AG_i}\} \end{matrix} \longrightarrow$ CSP

AG and SM choose a random number γ_{AG_i} and $\gamma_{SM_{i-j}}$ respectively, and each of them sends a message to the CSP with the random number and device identity: $\{\gamma_{AG_i}, ID_{AG_i}\}$ $\{\gamma_{SM_{i-j}}, ID_{SM_{i-j}}\}$.

II. SM $\{\psi_{i-j}, \psi_i, S_{i-j}, GK_i, Sec_{i-j}\}$ ← CSP

AG $\{\psi_i, S_i, GK_i, Sec_i\}$

After receiving the messages, the CSP chooses a random value $K_{CSP_i}$and Q_i per group and calculates the authentication variables shown in table 1.

Table 1. Authentication variables

SM	AG
$r_{CSP_i} = K_{CSP_i} * P$	
$R_{SM_{i-j}} = r_{CSP_i} * \gamma_{SM_{i-j}}$	$R_{AG_i} = r_{CSP_i} * \gamma_{AG_i}$
$\psi_{i-j} = h_1(R_{SM_{i-j}} + ID_{i-j})$	$\psi_i = h_1(R_{AG_i} + ID_i)$
$s_{i-j} = x_{CSP} * \psi_{i-j} * K_{CSP_i}$	$s_i = x_{CSP} * \psi_i * K_{CSP_i}$
$Sec_{i-j} = SEC_{i-a} \oplus SEC_{i-b} \oplus \ldots \oplus SEC_{i-z}$	$Sec_i = SEC_{i-a} \oplus SEC_{i-b} \oplus \ldots \oplus SEC_{i-z}$

Then, it generates and sends a message with such values $\{\psi_{i-j}, \psi_i, S_{i-j}, GK_i, Sec_{i-j}\}$ to each SM and $\{\psi_i, S_i, GK_i, Sec_i\}$ AG.

III. SM $\{y_{i-j}\}$ → CSP

AG $\{y_i\}$

After receiving the message, both MS and AG calculate public and private keys, as in table 2. Finally, the public keys of SMs (y_{i-j}) and AG (y_i) are sent back to the CSP.

Table 2. Private / Public keys

	SM	AG
Private Key	$x_{i-j} = s_{i-j} + \gamma_{SM_{i-j}}$	$x_i = s_i + \gamma_{AG_i}$
Public Key	$y_{i-j} = \hat{e}(x_{i-j}, P)$	$y_i = \hat{e}(x_i, P)$

Figure 3 summarizes the registration phase.

3rd phase: Authentication:

A group of SM_{i-j}that aims at authentication in an SG network through an AG proceeds as follows:

M1. SM $\{MC_{SM_{i-j}}\}$ AG

Each SM_{i-j} chooses a random number $\sigma_{SM_{i-j}} \in Z_p^*$ and computes:

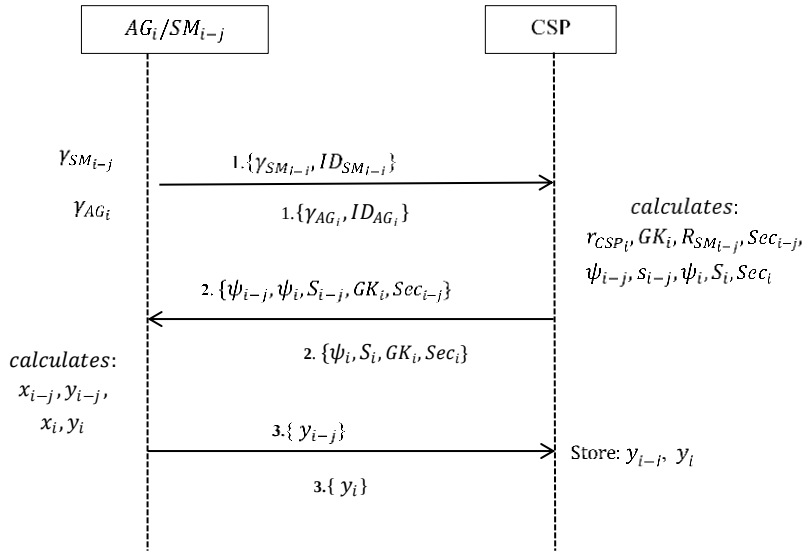

Figure 3. Registration phase

$$\lambda_{i-j} = h_1(\sigma_{SM_{i-j}} + \gamma_{SM_{i-j}})$$
$$MAC_{SM_{i-j}} = h_2\left(\psi_{i-j} \left|\left|\lambda_{i-j}\right|\right| LAI_{i-j}\right)$$
$$M_{SM_{i-j}} = \left(\psi_{i-j} \left|\left|\lambda_{i-j}\right|\right| LAI_{i-j} || MAC_{SM_{i-j}}\right)$$
$$MC_{SM_{i-j}} = M_{SM_{i-j}} \oplus (GK_i * \psi_i)$$

where LAI is the Location Area Identificator. Then, each SM sends $MC_{SM_{i-j}}$ to aggregator AG_i.

M2. AG $\{\text{AUTH}_{\text{Gi}}, \text{LAI}\}$ CSP

Upon receiving the message from the other devices $\{MC_{SM_{i-j}}\}$, AG_i performs an XOR operation with its temporary identity to obtain the data of the resized message:

$$M's_{SM_{i-j}} = MC_{SM_{i-j}} \oplus (GK_i * \psi_i) = (M_{SM_{i-1}} \left|\left|M_{SM_{i-2}}\right|\right| \ldots\ldots || M_{SM_{i-j}})$$

Since only the members of the group know the temporary group key, if the message is unreadable, an intruder is in the group and the aggregator initiates a process to search for the intruder and eliminate the connection.

Simultaneously, the aggregator chooses a number $\sigma_{AG_i} \in Z_p^*$ and, calculates:

$$\lambda_i = h_1(\sigma_{AG_i} + \gamma_{AG_i})$$

$$MAC_{AG_i} = h_2(\psi_i||\lambda_i||LAI_i)$$
$$M_{AG_i} = (\psi_i||\lambda_i||LAI_i||MAC_{AG_i})$$

Otherwise, it subtracts the $MAC_{SM_{i-j}}$ messages and calculates the message authentication of the group, MAC_{Gi}:

$$MAC_{Gi}: = \mathrm{h}_2\left(MAC_{AG_i} \oplus MAC_{SM_{i-1}} \oplus MAC_{SM_{i-2}} \oplus \ldots \oplus MAC_{SM_{i-j}}\right)$$

Then, the AG calculates a challenge L_h and generates an $AUTH_{Gi}$ message containing SM group information:

$$L_h = h_1(\mathrm{LAI}||ID_{Gi})$$
$$AUTH_{Gi} = \left(MAC_{Gi}||M_{AG_i}\left|\left|M_{SM_{i-1}}\right|\right|M_{SM_{i-2}}\left|\left|\ldots M_{SM_{i-j}}\right|\right|y_{AG_i}||L_h\right)$$

AG_i finally sends $AUTH_{Gi}$ and LAI to the CSP.

M3. AG/SM $\{AUTH_{CSP}\}$ CSP $\leftarrow$ - - - - - - - - - - - - - - - -

When the CSP receives the *AGi* message, it checks the LAI value declared by the devices, validates the message performing $L'_h = h_1(\mathrm{LAI}'||ID_{Gi})$ and compares $L'_h = L_h$. If the hashes do not match, the CSP sends a message to the whole failed group and terminates the authentication procedure. Otherwise, it calculates $MAC'_{AG_i} = h_2(\psi_i||\lambda_i||LAI_i)$ and all $MAC'_{SM_{i-j}} = h_2\left(\psi_{i-j}\left|\left|\lambda_{i-j}\right|\right|LAI_{i-j}\right)$ for generating the message authentication code of group $MAC'_{Gi} = \mathrm{h}_2\left(MAC_{AG_i} \oplus MAC_{SM_{i-1}} \oplus MAC_{SM_{i-2}} \oplus \ldots \oplus MAC_{SM_{i-j}}\right)$ and verifies if $MAC_{Gi} = MAC'_{Gi}$, . If the hashes do not match, CSP sends a MAC failure message to the group. Otherwise, it verifies the authenticity of the messages sent by SMs and AG through a bilinear pairing operation, shown in Table 3. The mathematical proof of the identity verification can be found in [Arias 2018].

If the verification of some SMs is not satisfactory, the CSP groups its connections into a quarantine list. The satisfactory SMs are grouped into a list of connections. If the AG verification is satisfactory, the CSP calculates the variables for the session key; otherwise, it sends an error message on the authentication to the group and closes connection.

Table 3. Verification of identity

SM	AG
$y_{SM_{i-j}} = \hat{\mathrm{e}}\left((x_{csp} * \psi_{i-j}), r_{CSP_i}\right)\hat{\mathrm{e}}(\lambda_{i-j}, P)$	$y_{AG_i} = \hat{\mathrm{e}}\left((x_{csp} * \psi_i), r_{CSP_i}\right)\hat{\mathrm{e}}(\lambda_i, P)$

After verifying the authentication data sent by all SMs through AG and the AG authentication data, CSP calculates a temporary group key and generates variables for the calculation of the session keys of each MS and AG.

a) CSP generates a random number r_{CSP1}, and calculates the temporary key for the group and a check value to authenticate it:

$$GTK_i = h_1(GK_i||r_{CSP1})$$

A new group´s temporary key is generated in each session.

b) the CSP chooses a random number $r_{CSP2} \in Z_p^*$ and generates variables to calculate session keys

$$F = r_{CSP2} * P$$
$$MAC_{CSP} = h_2(\mathrm{F}||GTK_i)$$
$$AUTH_{\mathrm{CSP}} = (\mathrm{F}||MAC_{CSP}||r_{CSP1})$$

It then **broadcasts** $AUTH_{\mathrm{CSP}}$ to all group members (AG_i / SM_{i-j}).

M4. SM/AG $\{Success/Failure\}$ CSP

c) when SM_{i-j} and AG_i receive the message, they compute

$$GTK_i = h_1(GK_i||r_{CSP1})$$
$$MAC'_{CSP} = h_2(\mathrm{F}||GTK_i)$$

Then, they check if $MAC_{CSP} = MAC'_{CSP}$. If the verification fails, they send a MAC failure message to the CSP; otherwise, the CSP is authenticated by the devices.

At the end of the authentication phase, the CSP is bound to the binary tree as a leaf and an SECy secret value is associated. Then, CSP and $\mathrm{AG_1}$/ $\mathrm{SM_{i-j}}$ compare the secret they know and find out what secrets they have in common. When the common secrets are identified between SM / AG and the CSP, the calculation of the session key is initiated (see Table 5).

Table 4. Session key generation

Session Key AG	$SK_{i-CSP} = ((SEC_a \oplus SEC_b \oplus \ldots \oplus SEC_z) * \lambda_i * F)$
Session Key SM	$SK_{i-j-CSP} = ((SEC_e \oplus SEC_f \oplus \ldots \oplus SEC_w) * \lambda_{i-j} * F)$

where SEC_a, $SEC_b \ldots SEC_z$ and SEC_e, $SEC_f \ldots SEC_w$ are the common secret values of AGi / SM_{i-j} and CSP , respectively. This model of session key is based on the session key presented by *Choi et al.*[11] and can be used for device-to-device communication (D2D) among SSM_{i-j}, AG_i and CSP_i. The entire Key Agreement and Key Distribution process is shown in Figure 4. Whenever a device is added or leaves the group, the group key must be updated to ensure backward secrecy and forward secrecy [Cremers 2015], [Saxena 2016]. If a new device wishes to join the group, it must be attached to the binary tree and, depending on the place, the member will have a secret SECi-y associated.

A new group key is then calculated with this secret:

$$GK'_i = h_3(GK_i \oplus SEC_{i-y})$$

If a device leaves the group, the new group key is calculated as follows:

$$GK''_i = GK_i \oplus SEC_{i-y}$$

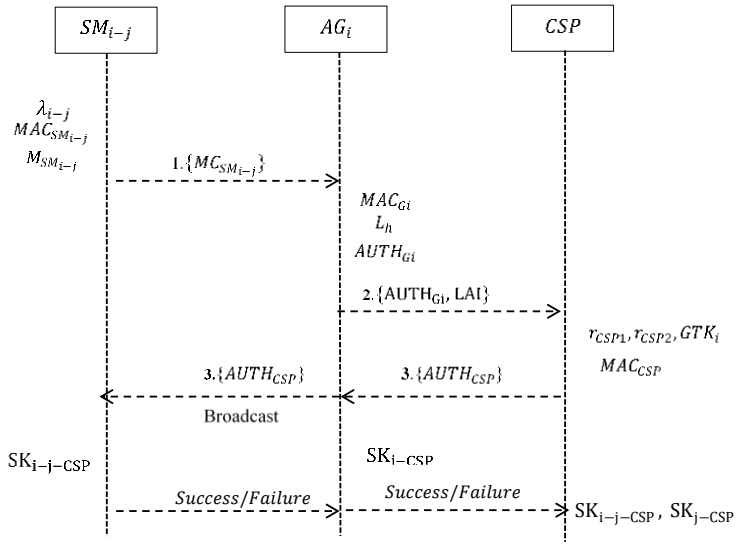

Figure 4. Authentication Phase

4. Security Analysis

This section reports on an analysis of the proposed protocol, in terms of security properties, as follows (other properties are analysed in [Arias 2018]).

- **Mutual Authentication:** In the first message, the SM sends data to be authenticated in the system to the AG. The data are encrypted with the initial group key (Gki) multiplied by the AG signature, so that only an authentic AG can un-message the message. On the other hand, if the AG can unmute the message, it confirms the SM is authenticated and part of the group. In the second message, the AG and SMs authentication data are sent to the CSP, which performs a bilinear pairing operation to check if the entities that sent the data are authentic. In the third message, the CSP sends data through Broadcast to the AG and the SM that are part of the group, and both SM and AG check if the CSP that sent the message is authentic.
- **Confidentiality and Integrity:** Messages exchanged among SM, AG, and CSP are protected by encryption with a session key, generated at the end of the authentication process, and combined with a hash function in each message, so that the receiver of the message can verify the integrity of the messages, thus guaranteeing its confidentiality and integrity.
- **Privacy (Anonymity):** Each SM and AG has a temporary identity (λ_{i-j}). Additionally, only the CSP can know their permanent identities (ID). If an attacker intercepts a message, it will obtain only its temporary identities, therefore, the privacy of the system is guaranteed.
- **DoS Attack:** Value L_h is very important for the verification of the authenticity of the devices and avoidance of DoS attacks, since the CSP checks the MAC of the group only if L_h is valid. DoS attacks are also mitigated with the implementation of a challenge in the protocol.
- **Man-in-the-Middle attack:** The session key cannot be calculated from information intercepted from the communication channel, because its calculation is based on binary tree secret values and ECDH encryption techniques. Group keys GK and GTK cannot be calculated either, because they are not exposed in any message.

5. Performance Evaluation

This section evaluates the protocol costs, for "n" devices per aggregator. The evaluation of communication costs is based on the total quantity of bits necessary for the operation of the protocol. The size of each parameter was taken from Saxena et al. [Saxena 2016]. Table 5 shows the computational costs of the three protocols. Saxena et al. [Saxena 2017] requires a very high communication cost in bits. Wan et al. [Wan 2014] requires smaller number of messages transmitted, if SM =< 8. Our protocol is better for groups with SM >8. However, Wan et al. [Wan 2014] was better for SM < 2, and for groups with SM >= 2, our protocol showed better performance, with smaller number of bits.

Table 5. Communication costs in bits per message and total.

	M1	M2	M3	M4	TOTAL
[Wan 2014]	192n	448n	64n	192n	864n
[Saxena 2017]	352n	640n	256n	-	1248n
Proposed	360n	360n +656	384	-	720n + 1040

The graph in Figure 5 shows a linear growth in the costs, related to an increase in the number of authenticated SMs. Our protocol uses aggregators in the AMI architecture for grouping authentication data of SMs and sending fewer messages to CSP for authenticating each member of the group, thus reducing the communication costs in the authentication phase.

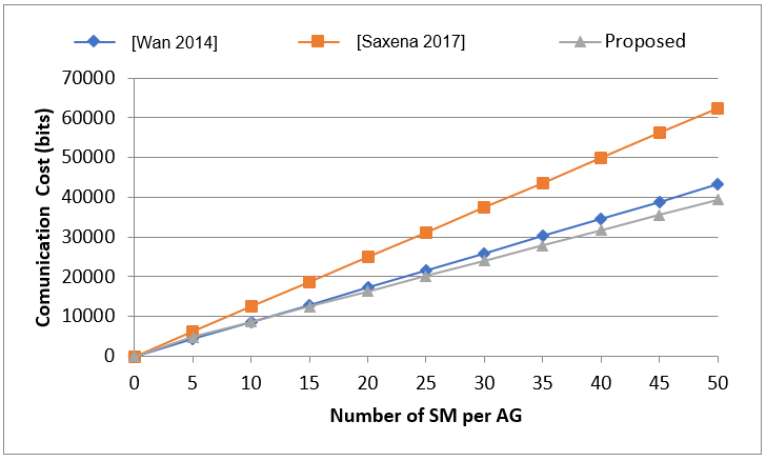

Figure 5. Communication Costs of the Protocols

About computational cost, its evaluation is based on an estimate of the time necessary for the execution of unitary operations, and the number of such operations. The running times were based on the same platform adopted by Wan et al. [Wan 2014], ensuring fairness in the comparison. Considering the mentioned running times, it is possible to obtain cost equations (similarly to Table 5) and to produce a graphical view of the computational costs (similarly to figure 5). As shown in [Arias 2018], our protocol shows the best computational cost when more than 4 SM's are connected to an aggregator. Regarding security, the aggregator checks the MACs of messages sent by SMs and also performs the authentication process, which guarantees its reliability in the system.

6. Conclusions

This article introduced a new group authentication protocol for the AMI network, integrated with the cloud and based on ECDH and bilinear pairing. It comprehends a simultaneous authentication scheme of a group of devices, and guarantees the integrity, confidentiality and privacy of users´ data. Our scheme shows better computational and communication costs than the protocol by Nicanfar et al. [Nicanfar 2013], and, compared with that of Wan et al. [Wan 2014], it showed the lowest number of messages exchanged for groups of SM> 2 and a smaller number of bits transmitted for groups of SM> 8. Moreover, it offers smaller computational cost in groups larger than 4 SM's.

The optimal performance of our protocol is due to several factors, such as use of an aggregator that groups the communications and enables the simultaneous verification of SM, and efficient application of the variables created for the authentication and execution of operations that require more processing in the entity with better computing resources. The scheme has proven an excellent solution to AMI authentication and authorization needs.

References

Arias, L. F. (2018) "Proposal and Evaluation of Authentication Protocols for Smart Grid Networks", Dissertação de Mestrado, ENE, UNB.

Bera, S., Misra, S., Rodrigues, J. J. P. C. (2015)"Cloud Computing Applications for Smart Grid: A Survey". IEEE Transactions on Parallel and Distributed Systems, v. 26, n. 5, pp. 1477-1494.

Choi, D., Hong, S. and Choi, H.K. (2015) "A group-based security protocol for Machine Type Communications in LTE-Advanced". Wireless Networks, v. 21, n. 2, p.405-419.

Cremers, C. J. F. and Feltz, M. Beyond C, (2015) "Perfect Forward Secrecy Under Actor Compromise and Ephemeral-key Reveal". Designs, Codes and Cryptography, v. 74, n. 1, pp. 183–218.

Genge, B., Beres, A., Haller, P. (2014), "A Survey On Cloud-Based Software Platforms To Implement Secure Smart Grids". 49th Universities Power Engineering Conference - UPEC2014.

Nicanfar, H. et al. (2013) "Efficient Authentication and Key Management Mechanisms for Smart Grid Communications". IEEE Systems Journal, v. 8, n 2, pp.629-640.

Saxena, N., Grijalva, S. (2017) "Dynamic Secrets and Secret Keys Based Scheme for Securing Last Mile Smart Grid Wireless Communication". IEEE Trans. Industrial Informatics, v. 13, n 3, pp. 1482-1491

Saxena, N., Choi, B., and Lu, B., (2016) "Authentication and Authorization Scheme for Various User Roles and Devices in Smart Grid". IEEE Trans. Information Forensics and Security. v.11, nr. 5, pp.907-921

Ye, F., Qian, Y., Hu, R.. (2015), "An Identity-Based Security Scheme for a Big Data Driven Cloud Computing Framework in Smart Grid". Global Communications Conference.

Wan, Z., Wang, G., Yang, Y., Shi, S (2014)"SKM: Scalable Key Management for Advanced Metering Infrastructure in Smart Grids". IEEE Trans. Ind. Electron., vol. 61, no. 12, pp. 7055-7066.