An Improved Privacy Solution for the Smart Grid

Mohamad Badra[1] and Sherali Zeadally[2]
(Corresponding author: Mohamad Badra)

College of Technological Innovation, Zayed University[1]
P.O. Box 144534, Abu Dhabi, U.A.E.
College of Communication and Information, University of Kentucky[2]
Lexington, KY 40506-0224, USA
(Email: mohamad.badra@zu.ac.ae)

(Received Dec. 20, 2013; revised and accepted Feb. 12 & Mar. 9, 2014)

Abstract

Recent advances in hardware, software, computing, and communication technologies have enabled the design and deployment of a smarter, interactive, dynamic 21st century electrical grid, also known as the smart grid. The bi-directional flow of information between the customer premise and the utility provider opens up several privacy challenges that must be addressed. We describe possible man-in-the-middle attacks against one (proposed by Marmol et al.) of the recently proposed privacy solutions for the smart grid environment. To address this vulnerability, we propose an improved privacy solution. We demonstrate the robustness and efficiency of our solution through a detailed security analysis.

Keywords: Advanced metering infrastructure, communication, man-in-the-Middle attack, privacy, protocol, security, smart grid

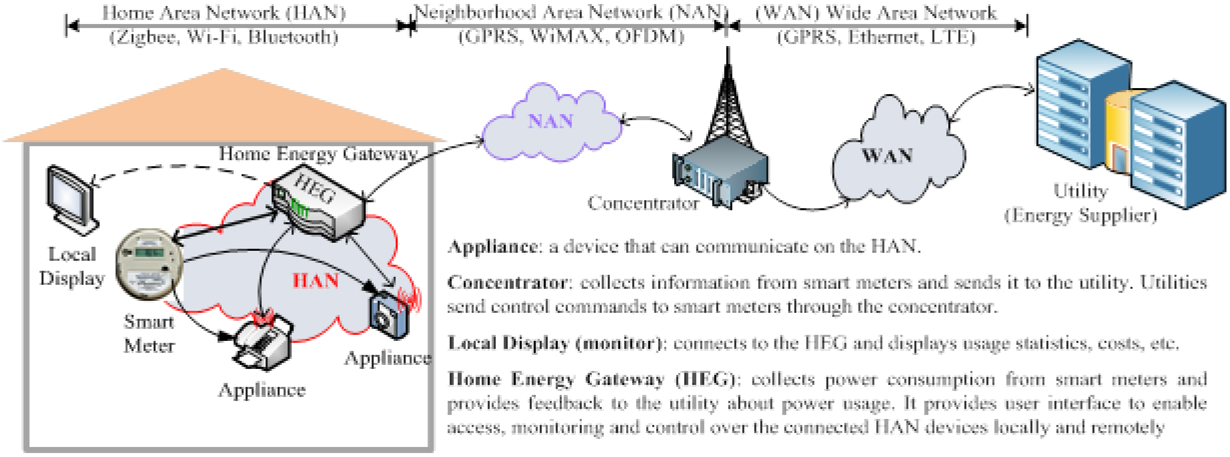
1 Introduction

In the last few years, we have witnessed a growing interest and increasing investments in smart grid technologies around the world. A smart grid is a complex infrastructure based on a set of seven domains [18]: bulk generation, energy distribution, power transmission, operation and control, market, service providers, and customers. Each domain comprises heterogeneous elements that include organizations, buildings, individuals, systems, system resources and other entities. The backhaul communication and the Internet are crucial for connecting the different entities involved such as customers and utility systems through an Advanced Metering Infrastructure (AMI) [6]. An AMI is an interface with the capability for managing and interacting with smart meters and utility business systems through a bi-directional communication. This communication replaces the traditional one-way Advanced Meter Reading (AMR) approach by enabling business utilities or providers to notify their customers of electricity pricing at any time, providing them with customizable services to manage their power consumption themselves in addition to controlling the demand in real time.

There are several technologies and applications that have been integrated into an AMI system [5] including (as shown in Figure 1): smart meters, wide-area communications infrastructure, Home (local) Area Networks (HANs) and operational gateways working as main collectors. The smart meter is an advanced meter that measures energy consumption in much more detail than a conventional meter does. Future smart meters are envisaged to communicate information back to the local utility company for monitoring voltage loads and for billing purposes. Among some of the tasks that a smart meter can do are [18]: time-based pricing, collecting consumption data for consumer and utility, net metering, or communications with other intelligent devices or appliance devices in the home. As a result, the smart meters make it possible to add some kind of "intelligence" to the network and individual features of each residential consumer.

One of the key characteristics of the smart grid is its support for bi-directional information flow between the customer's premise and the utility provider using Internet Protocol-based technologies [19]. The bi-directional nature of information flow has however opened up various security and privacy challenges which are being addressed by many recently proposed security solutions. Currently, there are several types of concerns related to the privacy and security of data associated with the smart grid. The most serious threats related to the privacy deterioration of smart grid consumers include:

1) Cyber-attack and intrusion: the use of communication capabilities and technologies for critical functions such as control and monitoring of smart meters makes smart grid more prone to cyber-attacks.



Appliance: a device that can communicate on the HAN.

Concentrator: collects information from smart meters and sends it to the utility. Utilities send control commands to smart meters through the concentrator.

Local Display (monitor): connects to the HEG and displays usage statistics, costs, etc.

Home Energy Gateway (HEG): collects power consumption from smart meters and provides feedback to the utility about power usage. It provides user interface to enable access, monitoring and control over the connected HAN devices locally and remotely

Figure 1: The AMI architecture

Some examples of cyber-attacks include denial-of-service attacks and the cyber vulnerabilities that are exploitable by malicious entities to disrupt smart grid operations on a large scale [1].

2) Identity theft: an attacker could manipulate, clone or steal the smart meter's identifier by tracking and observing the behavioral patterns of the consumers and the appliances being used, and could conduct real time spying and surveillance [5].

The two-way information flow between the consumer and the utility in the smart grid environment opens up several privacy issues. In this paper, we focus on the issue of privacy primarily related to the information on the consumer's energy usage. Some of the privacy concerns associated with smart grid consumers include [19]: the type of data collected from the consumer; the frequency of such collection; the future usage and disclosure of such data to other parties; what permissions will be needed to allow the collected data to be shared among other third parties; and any legal consequences related to any unauthorized disclosure or analysis of consumer information.

The rest of this paper is organized as follows. In Section 2, we review the privacy scheme of Marmol et al. [13] and we show its vulnerability to man-in-the-middle attacks. In Section 3, we propose an improved security solution that can mitigate these attacks. Section 4 presents a security analysis of our solution and Section 5 presents a second solution to mitigate man-in-the-middle attacks. Finally, our concluding remarks are presented in Section 6.

Privacy issues in the smart grid environment are being studied extensively at the moment [2, 3, 4, 8, 9, 12, 13, 16, 17]. In [19], we presented an analysis of most of the recently proposed smart grid privacy solutions and identify their strengths and weaknesses in terms of their implementation complexity, efficiency, robustness, and simplicity.

Recently, Marmol et al. proposed a Homomorphic encryption based solution to protect the privacy of smart grid customers. Homomorphic encryption [11] allows specific types of computations to be carried out on cipher text and obtains an encrypted result. It allows one to compute arbitrary functions over encrypted data without the decryption key. In [13], smart meters individually encrypt their requests with an encryption function that allows the energy supplier to decrypt their aggregation result with an aggregated key, but no one can decrypt them individually. An encryption mechanism with this property is known as additively Homomorphic encryption [13]. In this paper, we demonstrate that the solution proposed by Marmol et al. is not resilient against man-in-the-middle attacks and we extend Marmol's solution to counter these attacks.

The attacks described later cannot be avoided without establishing an authenticated and secure channel between the *ES* and each smart meter belonging to the group. In a recent publication [14], the authors propose establishing a Transport Layer Security (TLS) secure connection to authenticate the *ES* using digital certificates. After the *ES* has been authenticated and in order to avoid smart meters' profile creation, the authors opted to use anonymous credentials as a solution to ensure the privacy of smart meters. By using an anonymous credential scheme, the smart meters prove that they are entitled to send their requests. However, using TLS based certificate can affect the performance of memory-constrained systems. A key impediment to the adoption of TLS is the computational and memory constraints of smart meters. The authors do not take into consideration the memory overhead for the smart meters to execute TLS as well as the communication overhead related to the TLS negotiation. In contrast to the approach described in [14], our proposed solutions in this work can effectively prevent the attacks described later without introducing additional overheads when compared to [14].

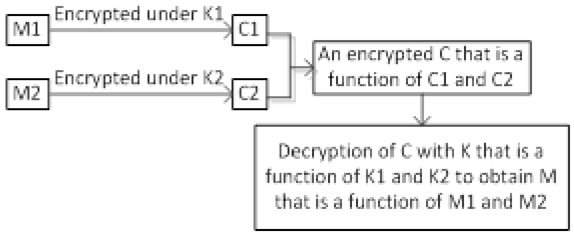

M1
Encrypted under K1
C1
M2
Encrypted under K2
C2
An encrypted C that is a function of C1 and C2
Decryption of C with K that is a function of K1 and K2 to obtain M that is a function of M1 and M2

Figure 2: An example of two messages encrypted using Homomorphic encryption

2 Man-in-the-middle Attacks on the Privacy Scheme

In this section, we review Marmol et al. [13] privacy scheme for the smart grid and we show that the scheme they claimed to be secure against the man-in-the-middle attack is vulnerable to this attack. In their scheme, they use an additive Homomorphic encryption which allows specific types of computations to be carried out on cipher texts to obtain an encrypted result.

2.1 Homomorphic Encryption

It is usually impossible for someone without the decryption key to manipulate the underlying data in any useful way [7]. However, some encryption schemes are Homomorphic; they are based on specific types of computations on encrypted data and allow the manipulation of the encrypted data, even without knowing the secret key [7] used to encrypt the data. By applying this scheme for securing data from a group's nodes, each node encrypts its request (e.g., M1 and M2 in Figure 2) with a different key (e.g., K1 and K2) and sends the encrypted request (e.g., C1 and C2) to an aggregator node. The aggregator node does not need to individually decrypt the date received from the nodes. It performs a transformation (e.g., addition or multiplication) on the received requests and decrypts the obtained result (e.g., C) with the aggregation key (e.g., K). The aggregation key is computed from the secret keys used by the group's nodes to encrypt their data.

Homomorphic encryption is being used by many practical applications where privacy is required. It was initially proposed in the context of Electronic Voting [15] in order to prevent the identification of users based on application-layer information.

2.2 Review of Marmol et al. Scheme

Marmol et al. proposed forming multiple groups of smart meters; each group consists of several smart meters belonging to the same building/street and is limited to one Energy Supplier (ES). One smart meter is randomly designated as a key aggregator to receive the group members' keys (as shown in Figure 3). Each member of the group encrypts its request using its current key and sends the encrypted request to the *ES* which performs an addition on the received requests and decrypts the obtained result with the aggregation key received from the key aggregator. However, if a smart meter sends its key to the key aggregator but does not send the corresponding encrypted content to the *ES* (or vice and versa), the *ES* cannot decrypt the aggregate value and causing the whole process to fail.

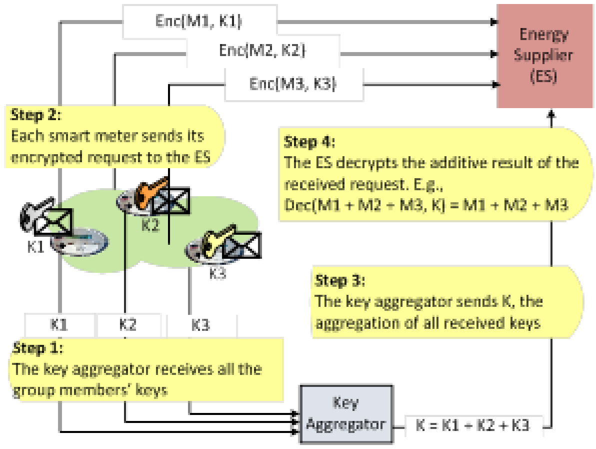

Enc(M1, K1)
Enc(M2, K2)
Enc(M3, K3)
Energy Supplier (ES)
Step 2:
Each smart meter sends its encrypted request to the ES
Step 4:
The ES decrypts the additive result of the received request. E.g.,
Dec(M1 + M2 + M3, K) = M1 + M2 + M3
K1
K2
K3
Step 3:
The key aggregator sends K, the aggregation of all received keys
K1
K2
K3
Step 1:
The key aggregator receives all the group members' keys
Key Aggregator
K = K1 + K2 + K3

Figure 3: Basic privacy solution of Marmol et al.

To address this problem, the authors propose an additional mechanism called "tokens solution" that works as follows (as shown in Figure 4). Before aggregating the individual keys received from smart meters, the key aggregator generates a token for each key and sends them back to the corresponding smart meters. Each smart meter reports its encrypted request together with the received token to the ES. The *ES* sends an acknowledgement message for each request received with a valid token to the key aggregator and the key aggregator will aggregate only the keys which are acknowledged by the *ES* [13]. Next, the ES performs a transformation (e.g., addition) on the received requests and decrypts the obtained result with the aggregation key received from the key aggregator. Since the key aggregator is elected periodically, the possibility to match the smart meter and its request is limited. However, it is always possible to establish this match if the key aggregator and the *ES* collaborate with each other.

2.3 Man-in-the-middle Attacks on Marmol et al. Scheme

As we mentioned previously, the solution described in [13] is not effective in thwarting man-in-the-middle attacks. We describe two scenarios where man-in-the-middle attacks are possible. We also describe an impersonation attack scenario.

Scenario A. In this scenario, the attacker replaces the encrypted re-quest being transmitted from a particular smart meter (victim) to the *ES* with another

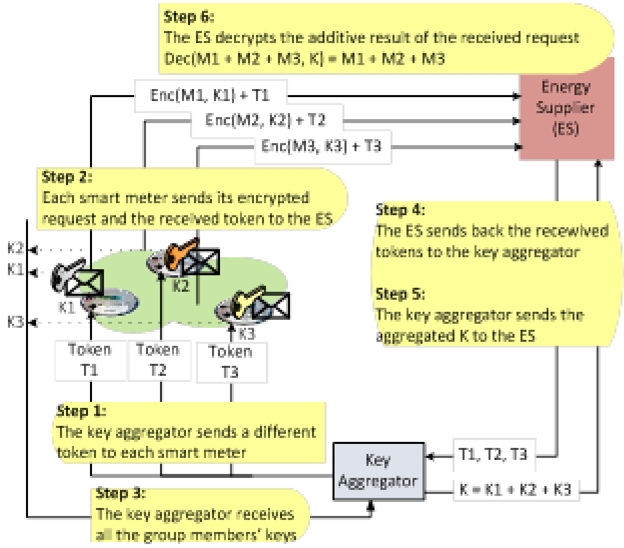

Step 6:
The ES decrypts the additive result of the received request
Dec(M1 + M2 + M3, K) = M1 + M2 + M3
Enc(M1, K1) + T1
Enc(M2, K2) + T2
Enc(M3, K3) + T3
Energy Supplier (ES)
Step 2:
Each smart meter sends its encrypted request and the received token to the ES
Step 4:
The ES sends back the recewived tokens to the key aggregator
Step 5:
The key aggregator sends the aggregated K to the ES
K2
K1
K3
K1
K2
K3
Token T1
Token T2
Token T3
Step 1:
The key aggregator sends a different token to each smart meter
Key Aggregator
T1, T2, T3
K = K1 + K2 + K3
Step 3:
The key aggregator receives all the group members' keys

Figure 4: Enhanced privacy architecture of Marmol et al.

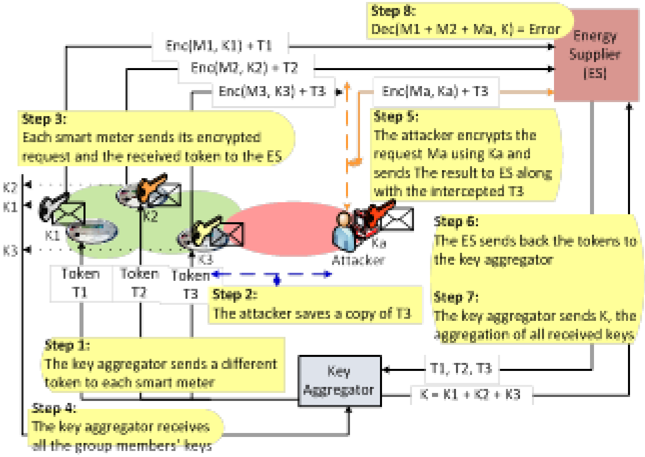

Step 8:
Dec(M1 + M2 + Ma, K) = Error
Energy Supplier (ES)
Enc(M1, K1) + T1
Enc(M2, K2) + T2
Enc(M3, K3) + T3
Enc(Ma, Ka) + T3
Step 3:
Each smart meter sends its encrypted request and the received token to the ES
Step 5:
The attacker encrypts the request Ma using Ka and sends The result to ES along with the intercepted T3
K2
K1
K3
K1
K2
K3
Ka
Attacker
Step 6:
The ES sends back the tokens to the key aggregator
Token T1
Token T2
Token T3
Step 2:
The attacker saves a copy of T3
Step 7:
The key aggregator sends K, the aggregation of all received keys
Step 1:
The key aggregator sends a different token to each smart meter
Key Aggregator
T1, T2, T3
K = K1 + K2 + K3
Step 4:
The key aggregator receives all the group members' keys

Figure 5: The man-in-the-middle attack on the privacy architecture proposed by Marmol et al.

request encrypted with a key that is never sent to the key aggregator (as shown in Figure 5). The attacker sends its encrypted request to the ES together with the token extracted from the request generated by the victim.

Next, the *ES* sends an acknowledgement message for the request received from the key aggregator. The key aggregator will consider the victim's key when aggregating the keys acknowledged by the ES. Hence, the *ES* will not be able to decrypt the additive result of the received requests and consequently, the entire process is compromised.

Scenario B. In this scenario, the man-in-the-middle attack intercepts the communication between the smart meter and the key aggregator and the communication between the same smart meter and the

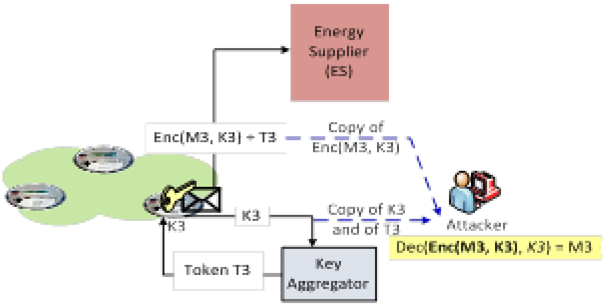


Figure 6: Another man-in-the-middle attack on Marmol et al.

ES (as shown in Figure 6). Since the smart the smart meter sends its key in clear text to the key aggregator, the attacker will be able to decrypt the encrypted request sent from the smart meter to the ES. Hence, the solution of [13] does not always guarantee the privacy of customers.

Impersonation Attack. An attacker spoofs the key aggregator and sends tickets to the group's members. By doing so, the attacker can let the group's members send their current keys to it. Since the current keys of the group's member are sent in clear text to the key aggregator, the attacker is able to compute the aggregation key and intercepts any future encrypted data being transmitted by the group's members to the *ES* or to the key aggregator.

3 Our Improved Privacy Solution

In this section, we describe our proposed solution to mitigate the possible attacks described above. Our solution uses a modified architecture of the solution proposed by Marmol et al. In our proposed privacy solution, the smart meters of the same group share the same key (key group) with the ES. The key group is generated by the *ES* and is installed on every smart meter of the same group (the installation could be done during the personalization phase of the smart meters).

3.1 Notations

The notations in Table 1 are used throughout this paper.

3.2 Overview of Our Proposed Solution

Before aggregating the individual keys received from smart meters, the key aggregator generates a token for each key and sends the tokens back to the corresponding smart meters. The token used by Marmol et al. is opaque or a string of data. In our architecture, we propose a semantic meaning for the token being used here. By semantic meaning we mean that the key aggregator

Table 1: Notations and definitions

Notation	Definition
$Dec(C, K)$	Decrypting the encrypted value C using the key K
$Enc(M, K)$	Encrypting the message M suing the key K
Entity	Smart meter, Energy Supplier, etc.
ES	Energy Supplier
$HMAC(SK, M)$	Calculating a message authentiction code (MAC) of the message M by using a hash function and a secret key SK
K	Aggregated key
KA	Key Aggragator
K_G	The group's key shared between all group's members
K_{sm}	A key generated by the smart meter sm
T_E	A token generated by the entity E
M_{sm}	A request generated by the smart meter
sm	sm smart meter
$\Vert$	Concatenation

will be authenticated which results in reducing the identity usurpation attacks. The token is generated as follows. The key aggregator generates a digest by applying the Keyed-Hashing for Message Authentication Code (HMAC) [10] on the token using the group key. Next, the key aggregator sends the digest along with the token to the smart meter. Upon receipt, the smart meter computes the HMAC and compares it with the received HMAC for equality (as shown in Figure 7).

Upon receipt of the token, each smart meter sm generates its key ksm, encrypts it along with the received token Tsm and sends the result back to the key aggregator (i.e., Enc(ksm —— Tsm, kG)). Next, the sm encrypts the Tsm and its request M_{sm} using the generated key ksm (i.e. Enc(M_{sm}, ksm)). Finally, the sm sends the encrypted value along with the digest value obtained by applying HMAC on the concatenation of the encrypted value and the token Tsm using the key group kG, as follows.

$$Enc(M_{sm}, k_{sm})||T_{sm}||HMAC(k_G, Enc(M_{sm}, k_{sm})||T_{sm}).$$

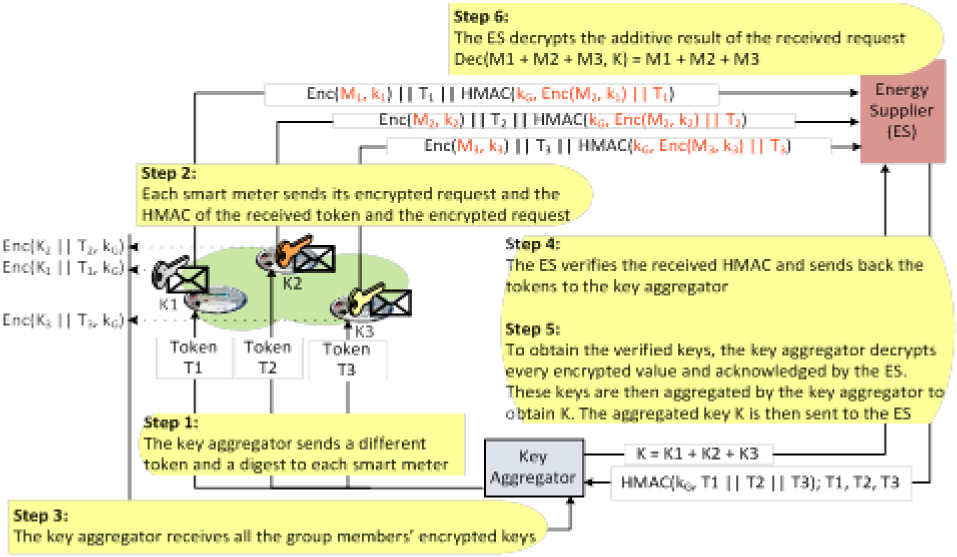
The HMAC value in the message being transmitted from the smart meter sm to the key aggregator authenticates the smart meter as a member of the group of authorized smart meters. Moreover, it detects falsified messages injected by man-in-the-middle attacks as we discuss later.

The ES computes the HMAC and then compares it with the received HMAC for equality (an attacker of scenario A and B described earlier cannot compute a valid HMAC without the key). Next, the ES collects the received tokens and sends them back to the key aggregator which then aggregates only the keys which are acknowledged by the ES to obtain the key K. During the key aggregation, the key aggregator decrypts only the encrypted values received from the smart meters and acknowledged by the ES. Finally, the key aggregator sends the aggregated key to the ES using a secure channel that could be established using the Transport Layer Security (TLS) or any other available security protocol.

3.3 Upgrading the Keys of Smart Meters

Marmol et al. define the bi-Homomorphic encryption as an encryption that is additive Homomorphic on both the plaintext and key spaces. For the key spaces, they define a mechanism based on the use of a ring to update their keys without changing the aggregated key (the aggregation of all the keys remains constant). To this end, each smart meter in the ring selects a random value and then subtracts this random value from its own key and at the same time sends that random value to its successor through a secure channel. The random value received from the predecessor is added to each smart meter's own key. Each random value added to one smart meter's key is therefore subtracted from another smart meter key's to keep the key K constant (the key K is updated every time one smart meter leaves/fails or enters/joins the group). However, the authors did not define the way to build the ring and did not specify the way a smart meter securely sends its random value to its successor. Moreover, it is not clear how the smart meters can be sure that the key update is successfully achieved.

Since each smart meter sends its updated key to the key aggregator, there is no need to use a ring and increase both the management and the computation overheads. We propose that every smart meter selects a random value and then subtracts this random value from its own key and at the same time sends update key and the selected random value to the key aggregator through a secure channel. Next, the key aggregator has the option of sending the sum of all received random values to the ES to update the aggregated key by subtracting the sum of the received random values from the current aggregated key K. If the objective of Marmol et al.'s proposal is to


Enc(M_1, k_1) || T_1 || HMAC(k_G, Enc(M_1, k_1) || T_1)
Enc(M_2, k_2) || T_2 || HMAC(k_G, Enc(M_2, k_2) || T_2)
Enc(M_3, k_3) || T_3 || HMAC(k_G, Enc(M_3, k_3) || T_3)
Energy Supplier (ES)
Enc(K_2 || T_2, k_G)
Enc(K_1 || T_1, k_G)
Enc(K_3 || T_3, k_G)
K1
K2
K3
Token T1
Token T2
Token T3
Key Aggregator
K = K1 + K2 + K3
HMAC(k_G, T1 || T2 || T3); T1, T2, T3

Step 6:
The ES decrypts the additive result of the received request
Dec(M1 + M2 + M3, K) = M1 + M2 + M3

Step 2:
Each smart meter sends its encrypted request and the HMAC of the received token and the encrypted request

Step 4:
The ES verifies the received HMAC and sends back the tokens to the key aggregator

Step 5:
To obtain the verified keys, the key aggregator decrypts every encrypted value and acknowledged by the ES. These keys are then aggregated by the key aggregator to obtain K. The aggregated key K is then sent to the ES

Step 1:
The key aggregator sends a different token and a digest to each smart meter

Step 3:
The key aggregator receives all the group members' encrypted keys

Figure 7: Our improved privacy solution

keep the unmodified key K constant, then the key aggregator selects one of the smart meters in the group and sends to it the sum of the received random values. The selected smart meter adds the received sum to its current key.

4 Security Analysis of Our Proposed Solution

In this section, we evaluate our proposed approach to demonstrate its effectiveness in maintaining data privacy and confidentiality. We also show that it is resilient against man-in-the-middle attacks and replay attacks.

4.1 Replay Attack

It may be possible for an attacker to read the data being transmitted from the group's members to the ES or to the key aggregator and save them for later use. However, it is useless for the attacker in both the scenarios A and B described earlier to use the token being transmitted from the key aggregator to a specific smart meter because the attacker needs to compute a correct HMAC operation that is applied on the fresh token and the secret key, an operation that is not possible if the attacker does not have the secret key.

4.2 Spoofing Attack

It is meaningless for the attacker to impersonate the key aggregator because it could learn nothing about the current keys of the group's members which are encrypted before being transmitted to the key aggregator. The attacker is unable to decrypt the current key of a smart meter without the group's key kG. Moreover, the group's members can verify the token's authenticity to avoid the impersonation attack described here.

4.3 Man-in-the-middle Attack

Our proposed solution mitigates both the man-in-the-middle scenarios A and B described earlier via the mutual authentication between smart meters and the ES and the key aggregator as well. Moreover, all the messages being exchanged between the entities are encrypted and HMACed using secret keys. Hence, it is not be possible for man-in-the-middle attacks to falsify the exchanged messages without being detected. If a man-in-the-middle-attack falsified the authenticated message being transmitted by a specific smart meter, only that specific smart meter would not be served, the ES would be able to serve the other group's members as long as the HMAC value in each of their requests is successfully verified. During the key aggregation phase, the key aggregator would omit the key of the victim and only those keys acknowledged by the ES would be considered. The proposal of Marmol et al. is not only ineffective in thwarting man-in-the-middle attacks but also fails when a single encrypted request is falsified by such attacks.

4.4 Analysis of Computational Costs and Comparison with Marmol et al. Scheme

We evaluate the performance of our improved privacy solution and compare it with the proposed approach of Marmol et al. Our solution introduces the use of HMAC to protect the encrypted requests and the tokens against man-in-the-middle attacks. Every smart meter performs one HMAC operation compared to Marmol et al. Moreover, every smart meter encrypts its key before being transmitted to the key aggregator. On the ES side, one HMAC operation is needed to protect the list of verified tokens from any modification. Table 2 summarizes the additional crypto-graphic operations needed by our solution when compared to Marmol et al. In Table!2, t_e denotes a time to encrypt or decrypt a message by using a symmetric cryptosystem; t_{HNAC} denotes a time to execute one HMAC operation.

Table 2: Additional computational costs needed by our solution when compared to Marmol et al.

	Marmol et al.	Our improved solution
Each sm	$1t_e$	$2t_e + 1t_{HMAC}$
ES	$2t_e$	$(n+1)t_{HMAC}$
KA	$1t_e$	$nt_e + 1t_{HMAC}$
Total	$(n+3)t_e$	$3nt_e + 2(n+1)t_{HMAC}$

As shown in Table 2, our solution introduces more crypto-graphic operations when compared to Marmol et al. Each smart meter authenticates the key aggregator by verifying the HMAC value (i.e., the token) received from the key aggregator. Moreover, each smart meter performs a HMAC operation to link its encrypted request and the token received from the key aggregator in order to mitigate the man-in-the-middle attacks described above and to authenticate itself as a member of the group of authorized smart meters.

By deploying our solution, it is easy for the ES to identify the sender of a falsified or badly formatted request and asks that sender to resend its request again. This operation costs four HMAC operations; two of them are executed by the victim and the other two are executed by the ES. In the case of Marmol et al., if a single request is falsified, the ES will not be able to detect it or to identify the sender of that falsified request. Hence, the entire process will fail and none of the group's member will be served by the ES. As a result, all of the cryptographic operations should be repeated by the involved entities (i.e., $O(n+3)$ encryption operations).

5 A Second Solution to Mitigate Man-in-the-middle Attacks

We assume that smart meters within the same group form a ring in which every smart meter in the ring has only two neighbors - a clockwise one (upstream neighbor) and an anticlockwise one (downstream neighbor). Each smart meter sm shares its initial key IK_{sm} with the ES so the ES is able to compute the aggregated key K.

Every smart meter sm will compute a second key K_{sm} as follows. Every time a smart meter is designated as a key aggregator or when a smart meter leaves/fails or enters/joins the ring, a key update process is initiated by the key aggregator. To this end, each smart meter sm in the ring will generate a random value RV_{sm} and then subtracts it from the random value received from its downstream neighbor. The result obtained is added to its initial key IK_{sm} to obtain its current key K_{sm}. At the end of the process, the keys of the ring's members are updated but the aggregated key K is always constant. The key K_{sm} is used by sm to encrypt its requests.

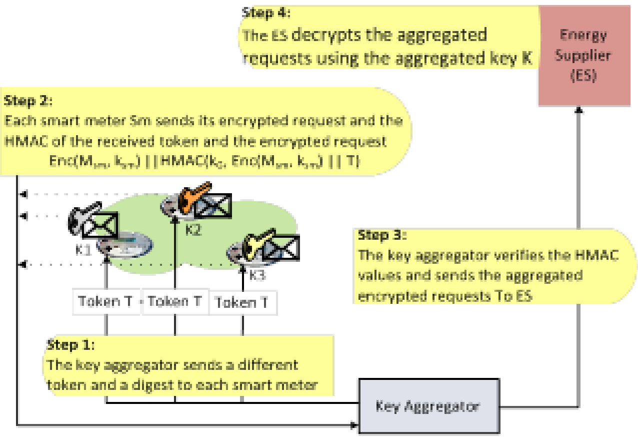

Step 4:
The ES decrypts the aggregated requests using the aggregated key K
Energy Supplier (ES)
Step 2:
Each smart meter Sm sends its encrypted request and the HMAC of the received token and the encrypted request
Enc(M_sm, k_sm) || HMAC(k_G, Enc(M_sm, k_sm) || T)
K1
K2
K3
Token T
Token T
Token T
Step 3:
The key aggregator verifies the HMAC values and sends the aggregated encrypted requests To ES
Step 1:
The key aggregator sends a different token and a digest to each smart meter
Key Aggregator

Figure 8: A second proposed solution

Figure 8 summarizes the required steps to send the requests of the ring's members as follows:

Step 1. The key aggregator multicasts to the group's members a token T that could be generated and authenticated as shown in Figure 7.

Step 2. Each smart meter encrypts its request using its current key K_{sm} and performs a HMAC on its encrypted request and on T using the group's key kG.

Step 3. The key aggregator receives the encrypted requests and the HMAC values from the group's members. It individually verifies the HMAC values and in case a specific HMAC value is not true, the key aggregator will inform the concerned smart meter to resend its request and to compute a valid HMAC value. Next, the key aggregator aggregates the received encrypted requests and sends the obtained value to the ES through a secure channel.

Step 4. The ES decrypts the aggregated requests using the aggregated key K.

By using the above scheme, the key aggregator can individually verify the HMAC value of each smart meter in the ring. As a result, it is not possible to have an unexpected decryption error caused by man-in-the-middle attacks. Since the keys of the ring's members are updated every time a smart meter is designated as a key aggregator, matching the smart meter and its request is not possible in case the key aggregator and the ES are collaborating with each other.

We evaluate the performance of our second solution and we compare it with our first solution described ear-

Table 3: Additional computational cost needed by our 1^{st} and 2^{nd} solutions when compared to Marmol et al.

	Our 1^{st} solution	Our 2^{nd} solution
Each sm	$2t_e + 1t_{HMAC}$	$1t_e + 1t_{HMAC}$
ES	$(n+1)t_{HMAC}$	-
KA	$nt_e + 1t_{HMAC}$	nt_{HMAC}
Total	$3nt_e + 2(n+1)t_{HMAC}$	$nt_e + 2nt_{HMAC}$

lier. Table 3 summarizes the additional cryptographic operations needed by our second solution when compared to the first solution. In our second solution, the smart meters do not send their keys to the key aggregator and hence we save 2n encryption/decryption operations.

However, the key aggregator verifies n HMAC values received from the smart meters. Table 4 summarizes the performance comparison in terms of the communication overhead for the two proposed solutions. The analysis shows that our second solution reduces the number of exchanged messages by (n+1) messages.

6 Conclusion

In this paper, we first analyze Marmol et al.'s privacy scheme for the smart grid and we show that it is easily broken by man-in-the-middle attacks. To address the weaknesses resulting from such attacks, we propose an improved privacy solution which extends the scheme of Marmol et al. We show that our proposed extension is secure against replay attacks, man-in-the-middle attacks and provides mutual authentication. It also provides the Energy Supplier the ability to identify falsified smart meters' requests without revealing those smart meters' identities and without dropping the requests of other smart meters. Compared to Marmol et al.'s scheme, our smart grid privacy solution requires more symmetric encryption and HMAC operations. However, these operations have no considerable performance impact given the security robustness our extension provides.

Acknowledgments

We would like to express our gratitude to the anonymous reviewers for their valuable feedback and comments which helped us to improve the quality and presentation of this paper.

References

[1] A. AlMajali, A. Viswanathan, and C. Neuman, "Analyzing resiliency of the smart grid communication architectures under cyber attack," in *Proceedings of the Fifth Workshop on Cyber Security Experimentation and Test*, pp. 4, 2012.

[2] M. Badra and S. Zeadally, "Design and Performance analysis of a virtual ring architecture for smart grid privacy," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 2, pp. 321–329, 2014.

[3] M. Chen, C. Yang, and M. Hwang, "Privacy protection data access control," *International Journal of Network Security*, vol. 15, no. 6, pp. 411–419, 2013.

[4] S. Das, K. Kant, and N, Zhang, *Security and Privacy in the Smart Grid, Handbook on Security Cyber-Physical Critical Infrastructure*, Morgan Kaufmann, Chapter 25, Feb. 2012.

[5] C. Efthymiou, G. Kalogridis, "Smart grid privacy via anonymization of smart metering data," in *Proceedings of the First IEEE International Conference on Smart Grid Communications*, pp. 238–243, 2010.

[6] Federal Energy Regulatory Commission, Assessment of Demand Response & Advanced Metering, Staff Report, 2008. (`http://www.ferc.gov/legal/staff-eports/demand-response.pdf`)

[7] C. Gentry, *Computing Arbitrary Functions of Encrypted Data*, 2008. (`http://crypto.stanford.edu/craig/easy-fhe.pdf`)

[8] Q. Jiang, J. Ma, G. Li, and L. Yang, "Robust two-factor authentication and key agreement preserving user privacy," *International Journal of Network Security*, vol. 16, no. 3, pp. 229–240, 2014.

[9] W. Juang and J. Wu, "Efficient user authentication and key agreement with user privacy protection," *International Journal of Network Security*, vol. 7, no. 1, pp. 120–129, 2008.

[10] H. Krawczyk, M. Bellare, and R. Canetti, *HMAC: Keyed-Hashing for Message Authentication*, RFC 2104, 1997.

[11] J. Liu, Y. Lu, and C. Koh, *Performance Analysis of Arithmetic Operations in Homomorphic Encryption*, 2010. (`http://docs.lib.purdue.edu`)

[12] R. Lu, X. Liang, X. Li, X. Lin, and X. Shen, "EPPA: An efficient and privacy-preserving aggregation scheme for secure smart grid communications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 9, pp. 1621–1631, Sep. 2012.

[13] F. Marmol, C. Sorge, O. Ugus, and G. Perez, "Do not snoop my habits: Preserving privacy in the smart grid," *IEEE Communications*, pp. 166–172, 2012.

[14] F. Marmol, C. Sorge, R. Petrlic, O. Ugus, D. Westhoff, and G. Perez, "Privacy-enhanced architecture for smart metering," *International Journal of Information Security*, vol. 12, no. 2, pp. 67–82, 2013.

[15] R. Rivest, *Lecture Notes 15: Voting, Homomorphic Encryption, Computer and Network Security*, 2002. (`http://web.mit.edu`)

[16] F. Siddiqui, S. Zeadally, C. Alcaraz, and S. Galvao, "Smart grid privacy: Issues and solutions," in *Proceedings of Second International Workshop on Privacy, Security, and Trust in Mobile and Wireless Systems (MobiPST 2012)*, (held in conjunction with IEEE ICCCN 2012), Munich, Germany, July 2012.

Table 4: Communication overhead of our 1^{st} and 2^{nd} solutions and of Marmol et al. (Messages exchanged between X and Y ($X \leftrightarrow Y$))

	$KA \leftrightarrow sm$	$KA \leftrightarrow ES$	$sm \leftrightarrow ES$	Total
Marmol et al. Scheme	$2n$	2	n	$3n + 2$
our 1^{st} solutions	$2n$	1	-	$2n + 1$
our 2^{nd} solutions	$2n$	2	n	$3n + 2$

[17] Y. Simmhan, A. Kumbhare, B. Cao, V. Prasanna, "An analysis of security and privacy issues in smart grid software architectures on clouds," in *Proceedings of IEEE International Conference on Cloud Computing*, pp. 582–589, 2011.

[18] U. S. Department of Energy, Advanced Metering Infrastructure, White paper by the NETL for U.S. DOE Office of Electricity Delivery and Energy Reliability, 2008.

[19] S. Zeadally, A. Pathan, C. Alcaraz, M. Badra, "Towards privacy protection in smart grid," *Wireless Personal Communications*, vol. 73, no. 1, pp. 23–50, 2013.

Mohamad Badra is an Assistant Professor at Zayed University, Abu Dhabi, UAE. He received a PhD degree in Networks and Computer Science from TLECOM ParisTECH. His research interests include key exchange, wireless/wired network security and privacy, public key infrastructures, smart cards, and wireless sensors networks. He is the author of several international standards on the security of exchanges and the co-author of many international conference and journal papers.

Sherali Zeadally is an Associate Professor at the College of Communication and Information, University of Kentucky, Lexington, KY, 40506, USA. He received his Bachelor's degree and Doctoral degree, both in Computer Science, from the University of Cambridge, England and the University of Buckingham, England respectively. He is a Fellow of the British Computer Society and a Fellow of the Institution of Engineering Technology, England.