Privacy-Preserving Techniques in Smart Metering: An Overview

Pedro Barbosa[1], **Lucas Freitas**[1],
Andrey Brito[1], **Leandro Silva**[1]

[1]Computer and Systems Department, Federal University of Campina Grande, Brazil

pedroyossis@copin.ufcg.edu.br, jose.freitas@ccc.ufcg.edu.br,

andrey@dsc.ufcg.edu.br, leandro@ufcg.edu.br

Abstract. *Smart energy meters grant to power providers the capability to perform many advanced services, and several countries already started to deploy them. However, data analysis can raise privacy issues by inferring daily activities and appliance usages of consumers. Hence, there is a crucial need to deal with the problem of consumers' privacy in smart metering. Several approaches offer ways to provide privacy and preserve some of the benefits. In this paper, we list, experiment and evaluate five of these approaches based on three orthogonal technologies: noise addition, rechargeable batteries, and homomorphic encryption. We evaluate them based on the main needed attributes, such as complexity and accuracy, and conclude that there are many tradeoffs to be dealt with.*

1. Introduction

Governments and power providers across the world have recognized that the traditional grid, which has not significantly changed in the last decades, must be replaced by more efficient, flexible and intelligent energy-distribution networks, called Smart Grids. These are digitally monitored, self-healing energy systems that deliver electricity from generation sources, including distributed renewable sources, to points of consumption. They optimize power delivery and enable end-user energy management, minimizing power disruptions and transporting only the required amount of power. The result is a lower cost to the power provider and to the consumer, and a more reliable power generation, transmission and distribution.

Smart meters are devices that measure electricity consumption in real time and transmit this data to remote servers. They may represent a turning point in the energy industry and foster the development of new services and improvement of existing ones. In a typical smart metering architecture, the analysis of the collected data can help power providers to learn how to better manage the areas within their networks. Thus, helping to understand the business benefits of investing in Smart Grids. Figure 1 presents a smart metering system architecture.

Despite the benefits, smart meters raise concerns about the privacy of consumers. Electricity data may contain private sensitive information, such as which appliances are being used, if the house is empty, when people take a shower or shut down the television. Using advanced power signature analysis tools, such as the Non-Intrusive Appliance Load Monitoring (NIALM), it is possible to find out private information about

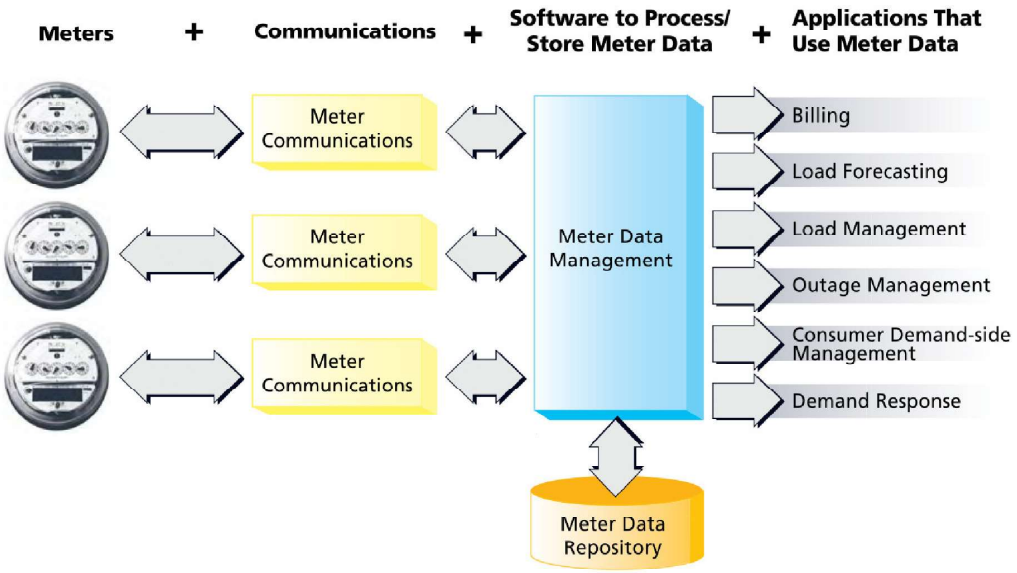

Figure 1. Smart metering system architecture and data usage applications [Waters 2006].

consumer's lifestyle. [Batra et al. 2014] designed some methodologies to identify the use of appliances from load profiles. If the load monitoring algorithm is running remotely, the consumers may not know that their behaviors are being monitored.

The information about a lifestyle of a person may be interesting to many businesses that can take advantages of it and offer their goods and services. In order to protect consumers from misuse of their data, and to prevent smart meters to become a new type of Big Brother, rules and regulations are needed [Boccuzzi 2010]. Sadly, such laws may take decades to be created and applied. While this still needs to happen, smart meters are already operational.

Taking these problems into account, there are a handful of solutions to protect privacy in smart metering. Many of them provide levels of privacy and still preserve several of the benefits of the smart metering deployment. Regarding some performance aspects such as computational complexity, some solutions are fast and lightweight, while others require heavy processing due to complex computations.

The solutions can be classified based on their approach to solve the privacy issues. There are techniques based on homomorphic encryption, the techniques that use rechargeable batteries, and the ones that make use of noise addition. In this paper, we review these techniques, evaluate and discuss their performance, according to a few relevant performance aspects. Besides computational complexity, we also evaluate other aspects, such as scalability, meters' independence, cost, environmental impact, and accuracy.

The rest of the paper is organized as follows. We summarize the problem statement in Section 2 and present the privacy techniques in Section 3. In Section 4 we present the conducted experiments and in Section 5 we discuss the obtained results. Finally, we conclude the paper in Section 6.

2. Problem Statement

There are many benefits of using a smart metering system. Some of them are: identify technical and non-technical losses (*e.g.*, power thefts), monitor energy quality scores, and optimize load forecasting. However, information of power consumption may be used for reasons not related to energy management, making the collection and distribution of such information dangerous to the privacy of consumers. The privacy issues are some of the main reasons why smart meters were still not deployed in many countries [Koehle 2012].

With the considerable amount of privacy-preserving techniques that can be adopted to ensure privacy in smart metering, the need to better understand and compare these solutions arises. To enable power providers to process smart meter measurements, while preventing them to access private data, cryptographic tools like homomorphic encryption may be used. With these approaches, before sending its measurement, each smart meter runs a cryptographic routine. The utility provider receives encrypted measurements, but can still perform useful computations and output the correct aggregate values, like the total consumption of a consumer during a billing period or the total consumption in the region in an instant of time. Thus, some benefits of using smart metering are still provided, while the consumer's privacy is maintained.

Techniques based on the usage of rechargeable batteries consist in using a battery between the smart metter and home appliances. Therefore, the disclosed information is the battery load profile, but not the daily activities and appliance usages of consumers.

Through noise addition techniques, individual measurements are masked by adding random numbers. This masking happens in a way that does not affect the outcome of the aggregating operations but hides the individual measurements.

In this paper, we address the following business problem: which privacy preserving solution in smart metering is more suited for a specific scenario? It is important for the development of smart grids that all involved parties understand the solutions, knowing what are their strengths and weaknesses. Some solutions might be simple for only one end (*e.g.*, in the smart meters), but not for the other (*e.g.*, power providers). Having an outline of solutions, descriptions, advantages and disadvantages is useful to assist the choice of techniques that are best fit for specific situations.

3. Privacy Techniques

Below we present some of the proposals for preserving privacy in smart metering. Since the main privacy issues are derived from individual data (like the consumption in a instant of time), these privacy preserving approaches tend to reveal aggregate data (like the total consumption through a billing period or the total consumption in a region) and hide individual data. The presented approaches follow a *Trust No One* philosophy, that is, the privacy techniques are applied in the smart meter, and trusted third parties are not required.

3.1. Noise Addition

Noise addition is a privacy preserving technique to mask the data. [Wang et al. 2012] propose an approach to mask the data adding random numbers from a GMM (Gaussian Mixture Models), whereas [Bohli et al. 2010] and [He et al. 2013] propose approaches to

mask the data using Gaussian noise. Noise addition is a promising and efficient technique, however, these mentioned aproaches do not have formal models to calculate the amount of noise that should be added to guarantee desired privacy and utility levels. In fact, [He et al. 2013] argue that for real world system design, a proper trade-off between privacy protection and accuracy should be considered.

[Barbosa et al. 2014] propose that for every measurement, the smart meter reads the consumption and adds a random number. Thus, after an aggregating operation (such as the calculation of the total consumption in a region or the total consumption of a consumer in the end of a billing period), the result will be:

$$\sum_{i=1}^{N} c_i \;\approx\; \sum_{i=1}^{N} (c_i \;+\; x_i)$$

where N is the total number of measurements, x_i is a random number generated from a probabilistic distribution and c_i is an individual consumption measurement.

The previous formalization can also be rewritten as follows:

$$\sum_{i=1}^{N} c_i \;=\; \sum_{i=1}^{N} (c_i \;+\; x_i) \;-\; e_o$$

where e_o is the obtained error by the addition of random numbers. Therefore, e_o is the sum of all added random values:

$$e_o \;=\; \sum_{i=1}^{N} x_i \;.$$

[Barbosa et al. 2015] developed many analytical models using probability theory for different distributions. Here we will consider the Laplace distribution. Let x_i be a random variable generated from this distribution. Its variance is $\sigma_x^2 \;=\; 2b^2$, where b is a scale parameter. Now, for a large N, the central limit theorem ensures that the obtained error for billing purpose follows a normal distribution with mean $\mu_{e_o} = 0$ and variance:

$$\sigma_{e_o}^2 \;=\; N^2(\sigma_x^2 \;/\; N) \;=\; N\sigma_x^2 \;= 2Nb^2 \;. \tag{1}$$

In other words, to have an obtained error between two accepted values (with high probability), we can use the following normal distribution:

$$e_o \;\sim\; N\left(0,\; 2Nb^2\right) \;.$$

As an example, Figure 2 shows a daily profile of a residential consumer.[1] There are 16 appliances in this consumption profile. However, the appliance with highest wattage and easier to identify is the laundry dryer.

[1]Combining appliance signatures we can generate arbitrary large populations and measurement frequency. Several databases of appliance signatures are available online (*e.g.*, Tracebase [Reinhardt et al. 2012]).

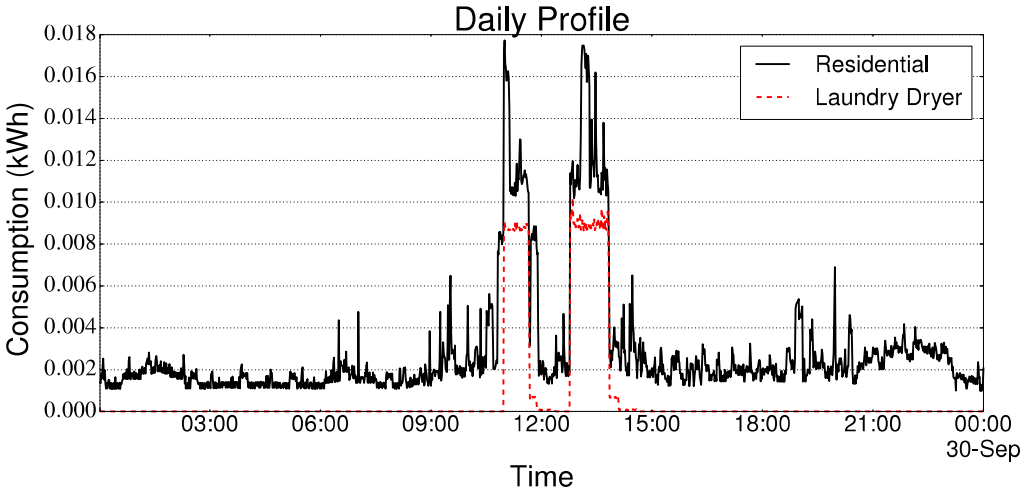

Figure 2. Residential (black solid) and Laundry Dryer (red dashed) daily profiles with measurements at each 1 minute.

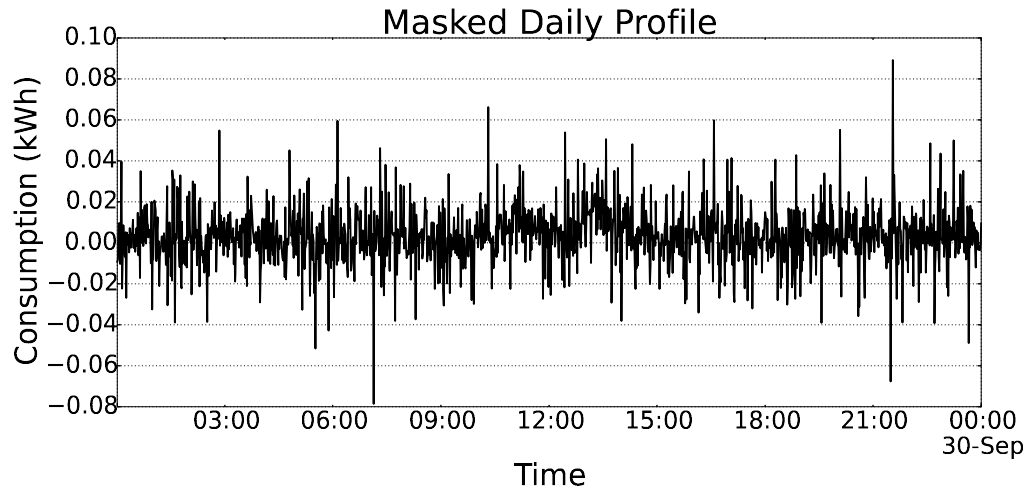

Figure 3. Residential masked daily profile with measurements at each 1 minute.

Assuming the billing period as one month, the total consumption of this consumer during the billing period (one month of 31 days) is 131.978 kWh. Considering a maximal allowed error of 5% for billing purpose, we have 6.5989 kWh. Thus, the variance for a high probability (*e.g.*, 0.98) of not exceeding this value is $\sigma^2_{e_o} = 8.04626$. Isolating the scale parameter b from Equation 1, we have (for measurements at each 1 minute, $N = 44,640$):

$$b = \sqrt{\sigma^2_{e_o} / (2N)} = 0.0094934\,.$$

Figure 3 presents the daily masked profile using this Laplacian noise. Considering that at the end of the month the power provider sums the informed masked values by the consumer, it obtains a value of 133.7977 kWh. The real value is 131.978 kWh. The difference between these values is an error of 1.3788%, less than the maximum allowed error (5%). This error may be less if the consumer does not mask the measurements all the time.

Using this noise addition approach, it is possible to provide differential privacy [Dwork 2006] guarantees for appliance usages, making them indistinguishable in a consumption profile [Barbosa et al. 2016].

3.2. Rechargeable Batteries

Rechargeable batteries between appliances and smart meters can help to reduce the privacy issues as the appliance signatures are no longer legible [Backes and Meiser 2014, Kalogridis et al. 2010, McLaughlin et al. 2011, Zhao et al. 2014].

[McLaughlin et al. 2011] propose an approach called Non-Intrusive Load Leveling (NILL). The goal of a NILL system is to level the load profile to a constant *target load*, thus removing appliance signatures. When an appliance turns ON, it will exert a load beyond the target load. Thus, NILL will discharge the battery to partially supply the load created by the appliance, maintaining the target load. Similarly, if an appliance enters the OFF state, the load profile will decrease below the target load. These opportunities are used to charge the battery while restoring the target load.

The NILL system consists of two parts: a battery and a control system that regulates the battery's charge and discharge based on the present load and battery state. The controller attempts to maintain a steady state target load K_{SS}, but will go into one of two special states K_L or K_H if the battery needs to recover from a low or high state of charge.

The essence of NILL is described by the equation, $u(t) = d(t) + b(t)$, where $b(t)$ is the battery's rate of charge overtime, $d(t)$ is the actual load profile of the residence and $u(t)$ is the load under the influence of NILL as perceived by the smart meter and what is disclosed to the power provider. If $b(t) > 0$, the battery is charging, otherwise $b(t) < 0$ and the battery is discharging. Finally, $c(t)$ is used to represent the battery's state of charge, thus:

$$c(t) = \int_{t_0}^{t} b(t)dt + c(t_0) \ .$$

Therefore, $c(t)$ is monitored. If $c(t) < L$, where L is the lower safe limit on the battery's state of charge, then the battery needs to be recharged and the system goes to the K_L state. Similarly, if $c(t) > H$, the system goes to the K_H state and the battery is discharged.

3.3. Using a Modified ElGamal Encryption

The first homomorphic encryption solution that we consider is based on a modification in the ElGamal encryption, a cryptographic system that relies on the discrete logarithm problem. The ElGamal cryptosystem proceeds as follows:

- *Set up*: two large primes p and q such that $q \mid p - 1$ are chosen. Next, a generator g of the order q multiplicative subgroup G of $\mathbb{Z}_p^*$ is selected. Afterwards, g, p and q are published.
- *Key generation*: a secret key x is generated by setting its value as a random number $x \in_R \mathbb{Z}_q^*$. The corresponding public key is computed as $y = g^x$.
- *Encryption*: a message $m \in G$ is encrypted under public key y by taking a random number $r \in_R \mathbb{Z}_q^*$ and computing $c = g^r$ and $d = m \cdot y^r$. The ElGamal encryption of m under public key y, $E_y(m)$, is the tuple (c, d).

- *Decryption*: a ciphertext $E_y(m)$ is decrypted using the private key x by computing $m = d \cdot c^{-x}$.

Given messages m_1 and m_2, we can obtain an encryption of $m_1 \cdot m_2$ by computing:

$$\begin{aligned} E_y(m_1) \cdot E_y(m_2) &= (c_1 \cdot c_2, d_1 \cdot d_2) \\ &= (g^{r_1+r_2}, m_1 \cdot m_2 \cdot y^{r_1+r_2}) \\ &= E_y(m_1 \cdot m_2)\ . \end{aligned}$$

Hence, ElGamal is a multiplicative homomorphic cryptosystem.

To calculate the total consumption in a region, [Busom et al. 2015] propose a protocol which uses an additive ElGamal cryptosystem. Given $E_y(g^{m_1})$ and $E_y(g^{m_2})$, then, $E_y(g^{m_1}) \cdot E_y(g^{m_2}) = E_y(g^{m_1} \cdot g^{m_2}) = E_y(g^{m_1+m_2})$.

Initially, each smart meter possess the following values: a big prime number q and its generator g; a secret key x_i; a public key $y_i = g^{x_i}$. To encrypt the measurements, it is necessary a global public key $y = \prod_{i=1}^{N} y_i$.

Let m_i denote the measurement of a smart meter. To calculate the total consumption in the region, the following protocol is executed:

1. Each meter generates a random noise value $z_i \in \mathbb{Z}_q^*$ and computes a ciphertext as $C_i = E_y(g^{m_i+z_i}) = (c_i, d_i)$ which is sent to the aggregator (which can be the power provider).
2. The aggregator combines all the messages as $C = (\prod_{i=1}^{N} c_i, \prod_{i=1}^{N} d_i) = (c, d)$ and sends c to each meter.
3. Each meter computes $T_i = c^{x_i} \cdot g^{z_i}$ and sends the result to the aggregator. After that, each meter removes z_i from its memory.
4. Finally, the aggregator computes $D = d \cdot (\prod_{i=1}^{N} T_i)^{-1}$ and $log_g D = M = \sum_{i=1}^{N} m_i$, where M is the total consumption in the region.

Notice that, since M is a relatively small number, the discrete logarithm problem in step 4 can be solved in a short time. In step 2, the aggregator computes:

$$C = (\prod_{i=1}^{N} g^{r_i}, \prod_{i=1}^{N} g^{m_i+z_i} \cdot y^{r_i}) = (g^r, g^{M+z} \cdot y^r) = (c, d)\ ,$$

and in step 3, each meter computes:

$$T_i = c^{x_i} \cdot g^{z_i} = g^{r \cdot x_i} \cdot g^{z_i} = g^{x_i \cdot r} \cdot g^{z_i} = y_i^r \cdot g^{z_i}\ .$$

Therefore, the protocol works because in step 4 the aggregator computes:

$$D = d \cdot (\prod_{i=1}^{N} T_i)^{-1} = \frac{g^{M+z} \cdot y^r}{\prod_{i=1}^{N} (y_i^r \cdot g^{z_i})} = \frac{g^{M+z} \cdot y^r}{(\prod_{i=1}^{N} y_i^r) \cdot g^z} = \frac{g^{M+z} \cdot y^r}{g^z \cdot y^r} = g^M\ .$$

3.4. Using Paillier Encryption and Secret Sharing

A protocol based on Paillier encryption and *secret sharing* was proposed by [Garcia and Jacobs 2010]. The Paillier cryptosystem proceeds as follows:

- *Set up*: two large primes p and q are chosen, $n = p \cdot q$, and $\lambda = lcm(p-1, q-1)$. A random number $g \in_R \mathbb{Z}^*_{n^2}$ is chosen in such a way that $gcd(b, n) = 1$, where $b = L(g^\lambda \ mod \ n^2)$ and $L(u) = \frac{(u-1)}{n}$.
- *Key generation*: let μ be the modular multiplicative inverse of b modulo n, *i.e.*, $\mu = b^{-1} \ mod \ n$. Thus, the public key is $P_k = (n, g)$ and the private key is $S_k = (n, \lambda, \mu)$.
- *Encryption*: a message m is encrypted under public key P_k by taking a random number $r \in_R \mathbb{Z}^*_{n-1}$ and computing $E_{P_k}(m) = g^m \cdot r^n \ mod \ n^2$.
- *Decryption*: a ciphertext $c = E_{P_k}(m)$ is decrypted using the private key S_k by computing $m = L(c^\lambda \ mod \ n^2) \cdot \mu \ mod \ n$.

Given messages m_1 and m_2, we can obtain an encryption of $m_1 + m_2$ by computing:

$$\begin{aligned} E_{P_k}(m_1) \cdot E_{P_k}(m_2) &= g^{m_1} \cdot r_1^n \cdot g^{m_2} \cdot r_2^n \ mod \ n^2 \\ &= g^{m_1+m_2} \cdot (r_1 \cdot r_2)^n \ mod \ n^2 \\ &= E_{P_k}(m_1 + m_2) \ . \end{aligned}$$

Hence, Paillier is an additive homomorphic cryptosystem.

[Garcia and Jacobs 2010] propose that each smart meter possess a public key P_{ki} and a private key S_{ki}. Let m_i denote the measurement of the meter. To calculate the total consumption in the region, the following protocol is executed:

1. Each meter sends its public key to the aggregator.
2. The aggregator receives all public keys and shares them with all meters. Thus, each meter stays with its private key S_{ki} and all the public keys $\{P_{k1}, P_{k2}, \ldots, P_{kn}\}$.
3. Each meter calculates N secret sharings for its measurement m_i, in such a way that $m_i = \sum_{j=1}^{N} s_{ij}$. Then, the meter keeps s_{ii} privately and sends to the aggregator all the other secret sharings encrypted with the public keys of the other $N-1$ meters, *i.e.*, it sends $E_{P_{kj}}(s_{ij})$ for $j = 1, \ldots, i-1, i+1, \ldots, N$.
4. After receiving all the encrypted secret sharings, the aggregator multiplies the ones encrypted with the same public key. Due to the Pailier homomorphic property, for each meter i, it has $E_{P_{ki}}(m'_i) = \prod_{j \neq i}^{N} E_{P_{ki}}(s_{ji}) = E_{P_{ki}}(\sum_{j \neq i}^{N} s_{ji})$. Then, the aggregator sends $E_{P_{ki}}(m'_i)$ for each meter i.
5. Using its private key S_{ki}, each meter decrypts $E_{P_{ki}}(m'_i)$ and adds its s_{ii}, obtaining $\sum_{j=1}^{N} s_{ji}$. The meter then sends this value to the aggregator.
6. Finally, the aggregator can sum all the received values, obtaining the total consumption in the region $M = \sum_{i=1}^{N} \sum_{j=1}^{N} s_{ji}$.

In this approach, the total consumption in the region is computed and at the same time, neither the aggregator nor any other consumer has access to any real measurement from a consumer, for they can only access random shares. Since in step 6 the aggregator simply sum all the secret shares, the proof that the protocol works is straightforward.

3.5. Using a Modified Paillier Encryption

[Erkin and Tsudik 2012] propose a protocol based on a modification in the Paillier encryption. Starting, there is a single pair of Paillier keys (P_k and S_k) shared with all N meters. Let m_i denote the measurement of the meter. To calculate the total consumption in the region, the following protocol is executed:

1. Each meter generates $N-1$ random numbers, one for each other meter, and sends them using secure communication (*e.g.*, RSA encryption between meters). Thus, there is a total of $N \cdot (N-1)$ message exchanges in this step.
2. After receiving all the random numbers generated by the other meters, the meter computes $R_i = n + \sum_{j \neq i}^{N} r_{(i \to j)} - \sum_{j \neq i}^{N} r_{(j \to i)}$, where n is the Paillier modulo and $r_{(i \to j)}$ is the random number generated by the meter i for the meter j.
3. Following, the meter computes a hash h_t using the timestamp of the current measurement m_i. This hash must be coprime with the Paillier modulo n, *i.e.*, $gcd(h_t, n) = 1$. Since the timestamp is synchronized, the obtained hash is the same for all meters.
4. After computing R_i and h_t, the meter encrypts m_i using the following modified scheme of Paillier: $E_{P_k}(m_i) = g^{m_i} \cdot h_t^{R_i}$. Then, this encrypted measurement is disclosed to all other $N-1$ meters.
5. Finally, after receiving all the encrypted measurements of the other meters, the meter calculates $E_{P_k}(M) = \prod_{i=1}^{N} E_{P_k}(m_i) = E_{P_k}(\sum_{i=1}^{N} m_i)$. This is true due the homomorphic property.

Possessing $E_{P_k}(M)$, the meter can decrypt this value and then send the total consumption in the region M to the aggregator. This way, the total consumption is computed and privacy is preserved, for the meter does not have access to the other measurements in plaintext.

The protocol works because in step 5, the meter computes:

$$E_{P_k}(M) = g^{m_1+m_2+\cdots+m_N} \cdot h_t^{(\sum_{i=1}^{N} n)+(\sum_{i=1}^{N}\sum_{j\neq i}^{N} r_{(i\to j)})-(\sum_{i=1}^{N}\sum_{j\neq i}^{N} r_{(j\to i)})},$$

and

$$E_{P_k}(M) = g^{M} \cdot h_t^{N \cdot n}.$$

Considering that $r = h_t^N$, this configuration represents the original paillier cryptosystem.

4. Our Experiments

In order to analyze and compare solutions, we describe the performance aspects that are considered in our studies. It is important because, for example, solutions might excel at

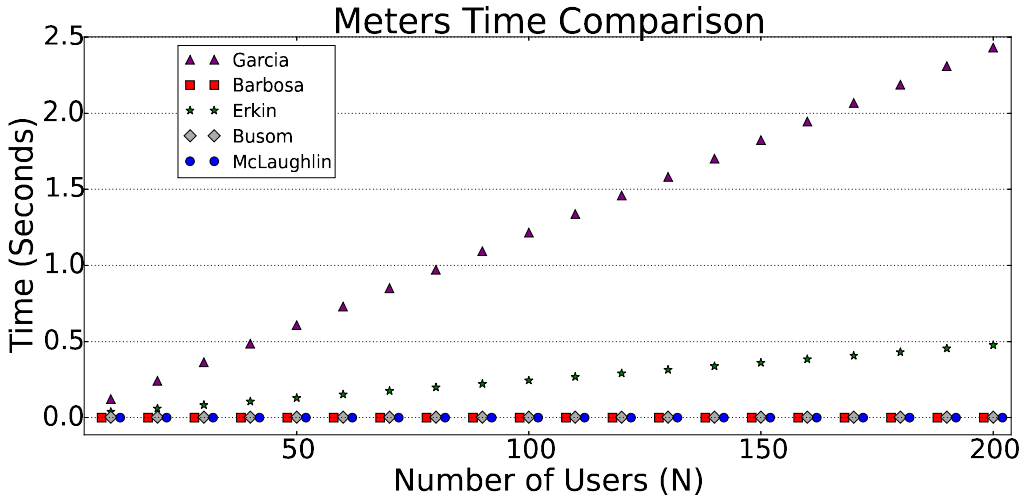

Figure 4. Processing time of smart meters in 5 different privacy preserving approaches.

computational complexity but their installation and usage has high costs and harms the environment. Thus, the need to consider different aspects of performance. In Section 5, we discuss the following aspects: computational complexity, scalability, meters' independence, cost, environmental impact and accuracy. Here, we present an experiment that aims to compare the response time (which is also related to computational complexity) of the approaches.

We implemented simulators[2] in *C* programming language. These simulators make use of a few functions in the *libgmp*[3], *libpaillier*[4] and *libcrypto*[5] libraries to implement algorithms that mimic the protocols described in this paper. The simulators were executed in a machine with 1.6 GHz Intel Core i5 processor, 6 GB of RAM memory and the Ubuntu 14.04 operating system.

In our simulations, using different configuration scenarios (number of meters, ranging from 1 to 200) to calculate the total consumption in the region, we measured the processing time of each meter (Figure 4) and the aggregator (Figure 5).

Each scenario was executed 10 times and the average values are being considered. This amount of repetitions were enough to get precise average values. Due to the very low obtained variations, the confidence intervals are being omitted here, except for the aggregation in the approach proposed by [Busom et al. 2015], which needs a trial and error mechanism to solve the discrete logarithm problem. The confidence intervals in these cases are of 95%.

From these measurements, we conclude that the noise addition approach and the one that uses rechargeable batteries presented very low response times, whereas the homomorphic encryption approaches presented considerable delays. It is also important to

[2]The source codes can be found at our GitHub repository (`https://git.lsd.ufcg.edu.br/pedroysb/privacy-performance-smart-metering/tree/master`).

[3]*libgmp*: `https://gmplib.org`

[4]*libpaillier*: `http://acsc.cs.utexas.edu/libpaillier`

[5]*libcrypto*: `https://www.openssl.org/docs/manmaster/crypto/crypto.html`

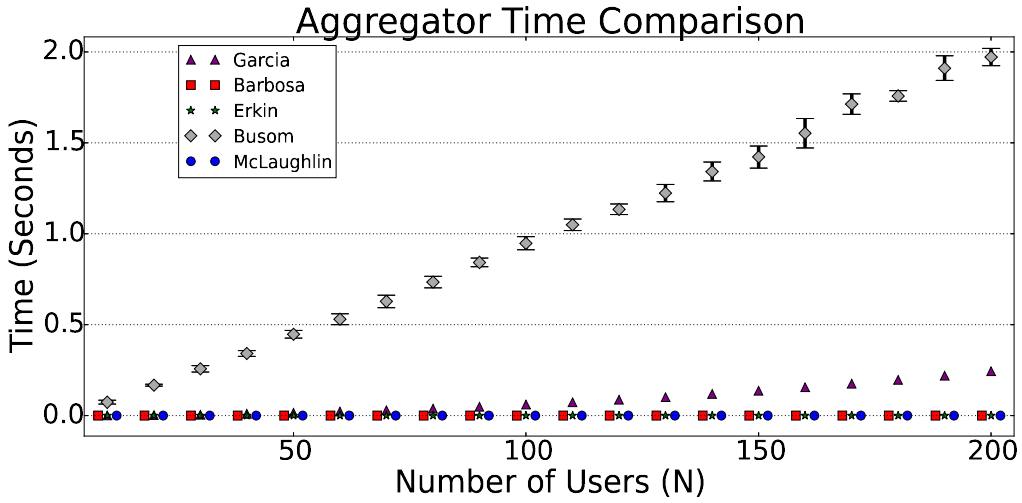

Figure 5. Processing time of the aggregator in 5 different privacy preserving approaches.

note that there are many message exchanges in the homomorphic encryption approaches, but we are not considering possible network delays in our experiments.

5. Discussions

We have presented several examples of privacy-preserving solutions to the computation of aggregate values. Now, we discuss the solutions regarding each of the performance aspects and taking into consideration the experimental results presented in Section 4. Table 1 presents a summary and comparison of approaches discussed.

Table 1. Comparison of noise addition (NA), rechargeable batteries (RB), and homomorphic encryption (HE) approaches.

	NA	RB	HE
Low complexity	✓	✓	
Scalability	✓	✓	
Meters' independence	✓	✓	
Low cost	✓		✓
Low environmental impact	✓		✓
Accuracy		✓	✓

5.1. Cost and Environmental Impact

As mentioned, usage of rechargeable batteries can help to diminish many privacy issues. Nevertheless, it is hard to ignore the environmental effects and the costs of using batteries. [McLaughlin et al. 2011] stipulate that a lead-acid battery of 50 Ah which operates at 12 V may cost approximately $100, and to achieve a typical residential nominal voltage of 120 V it is required 10 of such batteries (aprox. $1,000). The lifetime of each battery is approximately two years. Therefore, these solutions can not be considered low cost and cause a high environmental impact. Noise addition and homomorphic encryption approaches do not have these limitations.

5.2. Meters' Independence and Scalability

In order to exchange randomly generated numbers, keys and secret shares, the homomorphic encryption protocols require communication between meters and/or the power provider. Some solutions even require a large key distribution and certification. For this reason, in these approaches, the meters are not independent. This overhead can be the bottleneck of the smart metering system. Hence, actually optimizing communication at the meters is a hard task that has not been fully addressed in homomorphic encryption approaches. Additionally, if one meter fails in any message exchange, the aggregation becomes impossible because it requires computations using all distributed keys or secret shares. Therefore, these solutions may raise scalability issues when used in an area with a large number of meters. Noise addition and rechargeable batteries approaches do not have these limitations.

5.3. Accuracy

As mentioned, noise addition masks the data and introduces some error in an aggregation operation. This error is controlled and usually smaller than an acceptable value. Also, allowing the enabling or disabling of masking would make errors smaller in both dimensions: for billing, since a consumer would not mask all the time, and for load monitoring in a region, since not all consumers would mask in an instant of time. However, the noise addition privacy preserving approach is still not considered one hundred percent accurate. In the case of [Barbosa et al. 2015], the use of noise will introduce errors in the billing reports. These errors tend to be cancelled over time, but there is a probability that a higher value can occur. Rechargeable batteries and homomorphic encryption approaches do not have this limitation.

5.4. Computational Complexity

Low complexity is desired mainly because most of the deployed smart meters are low-cost microcontrollers and with limited computational resources. Through the analysis of the solutions' running time growth order, we have the complexities presented in Table 2. We consider that, regarding computational complexity, noise addition and rechargeable batteries stand in relation to the homomorphic encryption approaches.

Although estimation through asymptotic complexity is a good way to estimate computational complexity, we claim that experimental analysis is essential to present concrete results when comparing different proposals. As explored in Section 4, the experiments we conducted have shown that the noise addition and the rechargeable batteries approaches have considerably better performance regarding processing time.

6. Conclusions

Through the collection of energy consumption data, smart meters foster the development of new utilites and services. However, this same data can bring privacy issues for consumers, thus the need to study and develop privacy protection solutions. In this paper, we reviewed five solutions, one based on the use of noise addition, other based on the use of rechargeable batteries and three others based on homomorphic encryption schemes. We evaluated these approaches considering the main needed performance aspects and conclude that each solution has its advantages and disadvantages. As an example, by using rechargeable batteries the desired low levels of computational complexity is achieved, but on the other hand, they have a high cost and cause damage to the environment.

Table 2. Complexity analysis for different privacy preserving approaches in smart metering.

	NA		RB		MEE		PESS		MPE	
Operation	SM	AG	SM	AG	SM	AG	SM	AG	SM	AG
Encryption	-	-	-	-	$O(1)$	-	$O(N)$	-	$O(1)$	-
Decryption	-	-	-	-	-	$O(M)$	$O(1)$	-	$O(1)$	-
Transmission	$O(1)$	$O(N)$	$O(1)$	$O(N)$	$O(1)$	$O(N)$	$O(N)$	$O(N^2)$	$O(N)$	-
Sum	$O(1)$	$O(N)$	$O(1)$	$O(N)$	-	-	$O(1)$	$O(N)$	$O(N)$	-
Product	-	-	-	-	-	$O(N)$	-	$O(N^2)$	$O(N)$	-

Legend:
- **NA**: Noise Addition [Barbosa et al. 2015];
- **RB**: Rechargeable Batteries [McLaughlin et al. 2011];
- **MEE**: Modified ElGamal Encryption [Busom et al. 2015];
- **PESS**: Paillier Encryption and Secret Sharing [Garcia and Jacobs 2010];
- **MPE**: Modified Paillier Encryption [Erkin and Tsudik 2012];
- **SM**: Smart Meter;
- **AG**: Aggregator;
- **N**: Number of consumption measurements;
- **M**: Total (aggregate) consumption value.

Acknowledgement

This work was partially funded by the EU-BRA SecureCloud project (MCTI/RNP 3rd Coordinated Call) and by the Conselho Nacional de Desenvolvimento Científico e Tecnólogico (CNPq).

References

Backes, M. and Meiser, S. (2014). Differentially private smart metering with battery recharging. In *Data Privacy Management and Autonomous Spontaneous Security*, pages 194–212.

Barbosa, P., Brito, A., and Almeida, H. (2015). Defending against load monitoring in smart metering data through noise addition. In *Proc. of the 30th Annual ACM Symposium on Applied Computing (SAC)*, pages 2218–2224, Salamanca, Spain.

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

Barbosa, P., Brito, A., Almeida, H., and Clauß, S. (2014). Lightweight privacy for smart metering data by adding noise. In *Proc. of the 29th Annual ACM Symposium on Applied Computing (SAC)*, pages 531–538, Gyeongju, South Korea.

Batra, N., Kelly, J., Parson, O., Dutta, H., Knottenbelt, W., Rogers, A., Singh, A., and Srivastava, M. (2014). Nilmtk: An open source toolkit for non-intrusive load monitoring. In *5th International Conf. on Future Energy Systems (ACM e-Energy)*, Cambridge, UK.

Boccuzzi, C. (2010). Smart grid e o big brother energético. *Metering International América Latina*, 3:82–83.

Bohli, J., Sorge, C., and Ugus, O. (2010). A privacy model for smart metering. In *Proc. IEEE International Conf. Communications Workshops (ICC)*, pages 1–5, Cape Town, South Africa.

Busom, N., Petrlic, R., Sebé, F., Sorge, C., and Valls, M. (2015). Efficient smart metering based on homomorphic encryption. *Computer Communications*, pages 95–101.

Dwork, C. (2006). Differential privacy. In *Proc. of the 33rd International Colloquium on Automata, Languages and Programming, part II (ICALP)*, pages 1–12, Venice, Italy.

Erkin, Z. and Tsudik, G. (2012). Private computation of spatial and temporal power consumption with smart meters. In *Proc. of the 10th Int. Conf. on Applied Cryptography and Network Security (ACNS)*, pages 561–577.

Garcia, F. D. and Jacobs, B. (2010). Privacy-friendly energy-metering via homomorphic encryption. In *Security and Trust Management*, volume 6710, pages 226–238.

He, X., Zhang, X., and Kuo, C. C. J. (2013). A distortion-based approach to privacy-preserving metering in smart grids. *IEEE Access*, 1:67–78.

Kalogridis, G., Efthymiou, C., Denic, S. Z., Lewis, T. A., and Cepeda, R. (2010). Privacy for smart meters: towards undetectable appliance load signatures. In *IEEE 1st International Conf. on Smart Grid Communications (SmartGridComm)*, pages 232–237, Gaithersburg, USA.

Koehle, O. (2012). *Just say no to big brother's smart meters. The latest in bio-hazard technology*. ARC Reproductions.

McLaughlin, S., McDaniel, P., and Aiello, W. (2011). Protecting consumer privacy from electric load monitoring. In *Proc. of the 18th ACM Conf. on Computer and Communications Security (CCS)*, pages 87–98, Illinois, USA.

Reinhardt, A., Baumann, P., Burgstahler, D., Hollick, M., Chonov, H., Werner, M., and Steinmetz, R. (2012). On the accuracy of appliance identification based on distributed load metering data. In *Proc. of the 2nd IFIP Conf. on Sust. Internet and ICT for Sustainability*, pages 1–9.

Wang, S., Cui, L., Que, J., Choi, D.-H., Jiang, X., and Xie, L. (2012). A randomized response model for privacy preserving smart metering. *IEEE Trans. on Smart Grid*, 3:1317–1324.

Waters, G. (2006). Conquering advanced metering cost and risk. *Electric Energy T&D Magazine*, 10:22–25.

Zhao, J., Jung, T., Wang, Y., and Li, X. (2014). Achieving differential privacy of data disclosure in the smart grid. In *Proc. of IEEE INFOCOM*, pages 504–512.

