A New Crypto-classifier Service for Energy Efficiency in Smart Cities

Oana Stan[1], Mohamed-Haykel Zayani[2], Renaud Sirdey[1], Amira Ben Hamida[2],
Alessandro Ferreira Leite[2] and Mallek Mziou-Sallami[2]
[1]*CEA, LIST, Point Courrier 172, 91191 Gif-sur-Yvette Cedex, France*
[2]*IRT SystemX, France*

Keywords: Smart City, Secure Classification, Data Privacy, Homomorphic Encryption.

Abstract: Smart Cities draw a nice picture of a connected city where useful services and data are ubiquitous, energy is properly used and urban infrastructures are well orchestrated. Fulfilling this vision in our cities implies unveiling citizens data and assets. Thus, security and data privacy appear as crucial issues to consider. In this paper, we study a way of offering a secured energy management service for diagnosis and classification of buildings in a district upon their energy consumption. Our remote service can be beneficial both for local authorities and householders without revealing private data. Our framework is designed such that the private data is permanently encrypted and that the server performing the classification algorithm has no information about the sensitive data and no capability to decrypt it. The underlying cryptographic technology used is homomorphic encryption, allowing to perform calculations directly on encrypted data. We present here the prototype of a crypto-classification service for energy consumption profiles involving different actors of a smart city community, as well as the associated performances results. We assess our proposal atop of real data taken from an Irish residential district and we show that our service can achieve acceptable performances in terms of security, execution times and memory requirements.

1 INTRODUCTION

The smart city concept is intrinsically related to the one of energy infrastructure and of smart grids. Despite their advantages, the smart metering systems and in general the monitoring of smart devices and the services for the energy domain cause serious security and privacy concerns. A reporting too fine-grained on a user consumption can reveal behavioral patterns and thus may infringe on his/her privacy. For example, the daily measurements can reveal whether a house is inhabited or not or when the inhabitants are away. In the same manner, the 15-minute or even the hourly reports can reveal a person timetable and habits making him/her vulnerable.

Therefore, in some countries and regions, privacy concerns are the main barriers for the large scale adoption of smart grid infrastructures (based on smart meters) and of the associated services one could benefit from. These problems of privacy and more generally data protection and computer security are making the object of various reports and recommendation documents. For example, (NIST, 2010) is advanced by the NIST (National Institute of Standards and Technology) in the USA as well as (Cavoukian et al., 2010) by the Canadian authorities. In France, the CNIL (Commission Nationale de l'Informatique et des Libertés) has published in November 2012 a compliancy kit for the communication meters (CNIL, 2012), conceived as a guide of best practices for the innovation process in the electrical industry by integrating data protection directly in the definition of new services, i.e. the "privacy by design".

It is in this context, of a real concern of citizens about their energy data privacy and the need of innovative services, making use of modern cryptographic primitives such as homomorphic cryptosystems, that our work arose.

The aim of this work is to propose a new privacy preserving classification service architecture, using homomorphic cryptography, in which the privacy of the energy data is assured by design. The system prototype proposed in this paper is a service for a smart district made of residential buildinds, performing classification and labelling remotely, without having access to sensitive data such as energy consumption or households characteristics (surface, number of inhabitants, etc.). Since one of the main issues with

Stan, O., Zayani, M., Sirdey, R., Ben Hamida, A., Ferreira Leite, A. and Mziou-Sallami, M.
A New Crypto-classifier Service for Energy Efficiency in Smart Cities.
DOI: 10.5220/0006697500780088
In *Proceedings of the 7th International Conference on Smart Cities and Green ICT Systems (SMARTGREENS 2018)*, pages 78-88
ISBN: 978-989-758-292-9

homomorphic cryptosystems are their costs in terms of applicability and performances, we are focusing here more on the required effort needed to implement a privacy preserving service based on this kind of technology. As such, we analyse the requirements in terms of protocol as well as the performances when using an additive homomorphic cryptosystem ((Paillier, 1999)), more easy to use but allowing only additions on encrypted data, and also a leveled homomorphic scheme (e.g. (Brakerski et al., 2011)). The latter is more recent and more complex but allows to execute, beside additions on the encrypted domain, a predefined number of consecutive multiplications. Let us also insist on the fact that, in this paper, we focus only on the privacy-preserving of the second step of a classification process, i.e. the operational phase of predicting the label for a given secured-through-encryption data, based on an already acquired model.

To summarize, the main contributions in this paper are: (a) the proposal of a new type of services for the smart city and energy field which are secured-by-design; (b) a Gaussian classifier for predicting the class of encrypted data (here, readings of smart meters); (c) the use of two appropiate homomorphic encryption schemes to protect energy data and, finally, (d) a description of a working demonstrator and a detailed analysis of its performances.

This work is organized as follows. In Section 2, we present some prior work related to the data privacy in the smart energy field with a focus on the privacy preserving data mining approaches. Section 3 describes the overall architecture of the classifier as well as the underlying mechanisms for performing the required operations using homomorphic properties. The implementation of the prototype, the dataset description and an evaluation of the associated performance results are given in Section 5. Lastly, the final section gives some insight about future works and perspectives.

2 RELATED WORK

Most of existing studies for energy data privacy concentrate on aggregation as the main application of homomorphic cryptography for smart grids (e.g. (Li et al., 2010), (Zirm and Niedermeier, 2012), (Vetter et al., 2012)). Data aggregation is one of the core functionalities often implemented in smart grids architectures at various scales (neighborhood, region, cities, etc.) and at different frequencies (every 15 minutes, daily, monthly, etc.). This comes with the purpose of monitoring and predicting power consumption, billing, reducing network load and traffic, efficiently administrate power generation, etc.

Moreover, a large category of existing services exploits the smart meters readings based on data mining techniques but without any guarantee on data privacy while performing this type of algorithms. We can also note a variety of approaches dedicated to the analysis and data mining of individual electricity consumption data, such as the HERS-Home Energy Rating System, NILM - Non-Intrusive Load Monitoring approaches (e.g., (Kim et al., 2010)) or other related literature on pattern identification (e.g., (De Silva et al., 2011), (Verdu et al., 2006)). However, to our knowledge, none of the existing work for the energy consumption classification has addressed the problem of assigning a class label for a given household while protecting its private data. For respecting the "privacy by design" principle one has to imagine new secure services using, for example, cryptographic techniques such as homomorphic encryption.

We consider that the application of efficient data mining techniques with underlying homomorphic schemes for building more privacy-friendly decision making tools in the context of smart energy could be beneficial for all the actors of this domain: the energy providers, the transmission system operators, the end-users, etc.

According to the mechanisms they rely on, we can distinguish between three main categories of privacy preserving data mining approaches:

- Data perturbation methods (e.g. (Agrawal and Srikant, 2000), (Bayardo and Agrawal, 2005))

  Before outsourcing the data to an external service which performs the data mining, the data is perturbed by adding random noise such that the final distribution seems different from the one of the actual data. Due to the addition of noise, data mining results may be significantly less accurate. Additionally, the data perturbation techniques do not offer strong cryptographic security properties such as the semantic security (Goldwasser and Micali, 1982).

- Data distribution or partitioning methods (e.g. (Lindell and Pinkas, 2000), (Kantarcioglu and Clifton, 2004))

  The main disadvantage of this type of methods is that they rely on heavy cryptographic mechanisms with high computational and communication overheads.

- Other cryptography-based techniques.

  In this category, we include the studies which are also using homomorphic encryption schemes beside other cryptographic techniques (e.g. (Bost et al., 2014), (Graepel and Naehrig, 2012),

(Samanthula et al., 2014)). Since there are only a few existing studies and this is the particular setting we are interested in, we will insist on this family of studies and provide more details for describing the classification algorithms belonging to this class.

Most of existing studies are dedicated to preserving privacy during the training phase in which a model is learned (using or not homomorphic encryption machinery, e.g. (Graepel and Naehrig, 2012),(Samanthula et al., 2014)) and only a few address the prediction step. Since, as stated previously, our approach addresses data privacy in the operational step of the classification process, we will describe further on the existing studies using homomorphic encryption during the labeling process.

The authors in (Graepel and Naehrig, 2012) propose a machine learning confidential protocol based on homomorphic encryption in which both training and prediction occur on encrypted data for a Linear Means classifier and a Fisher's Linear Discriminant Classifier. The results on the Wisconsin Breast Cancer dataset (UCI, 2016) show a slow down of 6-7 orders of magnitude when performing on encrypted data instead on plaintexts. For example, using a Linear Means Classifier, the classification of a test vector with 30 attributes takes roughly 6 seconds.

In (Bost et al., 2014), only the privacy for the classification process is addressed, by ignoring how the model has been build. As such, they are first conceiving a library with three building blocks which is further used as support for implementing classifiers such as hyperplane decision based, Naive Bayesian, decision trees and AdaBoost. The performances vary in function of the complexity of the classifier and the number of classes of the model (e.g. $>$ 1600 ms for a Naive Bayes Classifier on 5 classes and 9 features) and with an important overhead due to the communication (more than 70% of the total execution).

Here, we present a Gaussian classifier with the prediction performed on encrypted data, tailored to be applied for the protection of smart energy data readings.

3 SERVICE ARCHITECTURE

In this section, we present different global views for employing the classification service while assuring data privacy. The difference between these architectures lies in the manner the service is exploited by the actors in a smart city. In the scenario of a residential district, between the main users of such privacy-preserving classification energy service, one can enumerate:

- Owners of residential buildings, providers of energy data and other household characteristics (surface, number of inhabitants, revenues, etc.)
- A district management entity. One of his main concerns is how to ensure the district energy cost effectiveness based on district energy data. Estimating the energy efficiency of the district buildings is an interesting way towards the identification of greedy consumers. It is also a valuable feedback for future strategic decisions. However, for different reasons (legal or ethics, lack of resources or experience in data mining, focus on his core business) he does not perform the classification service by his own.
- A qualified remote third-party. This service provider designates a major actor of the use case and is able to process data, perform energy classification and assign ratings or labels. It can be perceived as an energy stakeholder having a valuable experience with classification or an energy consulting service supported by an energy provider/governmental programs (Ene, 2016). Typically, these services are built over conceived metrics and defined ratings (Nikolaou et al., 2015). They are extracted from users' feedback, investigations, surveys and simulations about residential electricity consumption and householders information. Thereby, sharing such data with the energy rating service is obviously necessary. Nevertheless, if sharing plain data threatens the privacy of inhabitants, this would compromise the rating process. The use of the homomorphic cryptography, in this situation, is a credible solution to overcome this constraint.

Attributing labels or ratings to the buildings, the same way it is done for some home appliances, could be helpful in providing synthetic indications about the energy efficiency throughout the district. Many architectures can be proposed in this sense but we will describe here two kinds of architectures. On the one hand, a three-tier architecture that involves the energy rating service, the district buildings and a district management entity. The latter role can be assumed by a district manager, (an) energy program administrator(s) or state/local authorities. According to the entity right access to district energy data, two subtypes of this architecture variant can be defined. On the other hand, if the process only concerns buildings and the energy rating service, a two-tier architecture fulfills the use case requirements.

We address the following questions: Concretely,

how do a district management entity or district buildings communicate with the energy rating service? What are the requirements that allow the energy rating service to process data and guarantee the exclusive access to the plain results to the district manager?

3.1 First Variant of the Three-Tier Architecture

As shown in Fig. 1, this architecture supposes that the district management entity collects the data and leads the encrypted data exchange with the energy rating service. This requires that the district management entity securely collects the energy data throughout the district (for example using standard cryptography techniques such as symmetric encryption). Then, when the district management entity needs to rate the energy efficiency of residential buildings, he launches the encryption of the data with her homomorphic public key.The encrypted data is sent to the third-party service to be processed and to determine encrypted ratings expressing the energy efficiency. In order to assign a rating to residential building in a secure way, an encrypted distance is computed between its energy efficiency metric and the one of each reference rating. The reference metrics are also encrypted with the public key of the district management entity. The sharing of the homomorphic public key can be realized by sending it from the district manager to the third-party service or by the recourse to a public key infrastructure (PKI). Finally, after the secure rating process, the district management entity collects the encrypted classification results. Her private key ensures that he has the exclusive right to decrypt the outputs of the energy rating service and obtain the energy efficiency for all the district he administrates.

3.2 Second Variant of the Three-Tier Architecture

Fig. 2 depicts a second variant of the three-tier architecture. In this case, the district buildings send the encrypted data to the energy rating service to perform the secure classification. As for the district management entity, he collects the encrypted ratings. The three tiers share the same public key, meanwhile, the district management entity possesses the private key which enable her to decrypt the service output in order to access the labeling results. Here, the architecture offers a credible solution when it is preferred that the district management entity has no visibility on plain data.

3.3 Two-Tier Architecture

When the process does not involve a district management entity, we head for a two-tier architecture as depicted in Fig. 3. A private key is possessed by each one of the buildings in the district. In this case, a householder who wants to obtain an energy efficiency evaluation launches the sending of his own encrypted data. Then, the energy rating service processes this data and returns the encrypted rating to the householder. Finally, he decrypts the service answer with his private key to access his evaluation. In this case, we assume either the existence of a PKI or that each district building has previously sent the public key to the energy rating service.

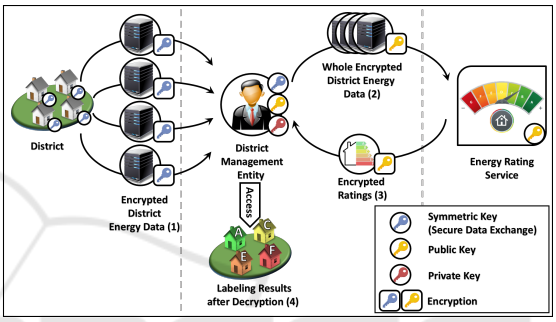

Figure 1: First Variant of the Three-Tier Architecture.

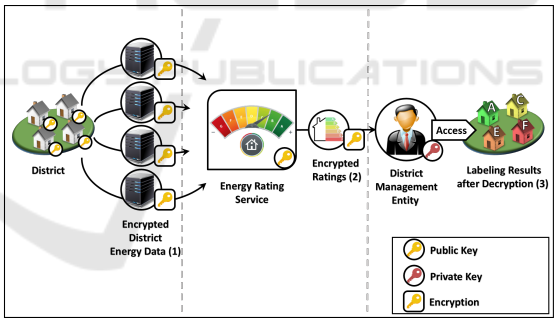

Figure 2: Second Variant of the Three-Tier Architecture.

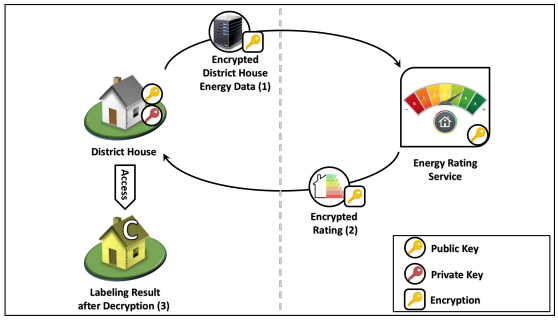

Figure 3: Two-Tier Architecture.

4 PRIVATE CLASSIFICATION ALGORITHM

4.1 General Description

For showing the feasibility of our approach, we have chosen a basic Gaussian classifier which was adapted in order to execute the prediction step on encrypted data. As such, given an encrypted attribute vector x, the purpose is to predict its class label based on the learning model acquired during the training step. Remember that we focus here only on the labeling step using private data and we suppose that the model building was realized previously in the clear domain.

In the case of a Gaussian Classifier, each class C_j from the m classes defined during the training phase is assumed characterized by a Gaussian distribution with a mean μ_j and a covariance matrix Σ_j.

The mean of a class C_j is the vector $\mu_j \in \mathbb{R}^n$: $\mu_j^\intercal = [\mu_{j_0}, \mu_{j_1}, \ldots, \mu_{j_n}]$ with μ_{j_i} the mean for the components i of the examples vectors x belonging to class C_j (i.e. $\mu_{j_i} = \frac{\sum_i^n x(i)}{n}$).

For vectors with n features, the covariance matrix of a class C_j is a semi-positive $n \times n$ matrix computed as: $\Sigma_j = \{c(a,b)\}$ with $a,b \in \{1,\ldots,n\}$ and $c(a,b)$ the covariance between the features a and b, measuring their tendency to vary together.

A feature vector x from the training set T is thus classified by measuring a Mahalanobis distance from x to each of the classes and by selecting the minimal norm. The main steps of the prediction phase of the Gaussian classification algorithm are described in Algorithm 1, Steps 4-6. The training phase realized on T_0, the set of training vectors x_0, has been realized before, resulting in a model with m classes. After computing the mean and the covariance of each class C_j (Steps 1-3), a class label is predicted for each testing vector $x \in T$.

As you can see, the prediction algorithm consists mainly on the computation of distances between the attribute vector x and the classes. Let us now explain how this distance can be evaluated on homomorphic encrypted data.

Algorithm 1: Gaussian classifier - prediction step.

Require: $T_0 = \{x_0 \in \mathbb{R}^n\}$; $T = \{x \in \mathbb{R}^n\}$; m classes C_j

1: **for** $\forall C_j$, $j \in \{1,\ldots,m\}$ **do**
2: compute μ_j and Σ_j using x_0
3: **end for**
4: **for** $x \in T$ **do**
5: compute $d_M(x,C_j)$, $\forall j \in \{1,\ldots,m\}$
6: $C(x) \leftarrow argmin(d_M(x,C_j))$
7: **end for**

Ensure: $C(x)$, $\forall x \in T$

4.2 Homomorphic Distance Evaluation

Given a vector x and a class C_j, the Mahanalobis distance from x to class C_j is defined as:

$$d_M^2(x,C_j) = (x-\mu_j)^\intercal \Sigma_j^{-1} (x-\mu_j).$$

Note that in the particular case where the features are uncorrelated or of a unidimensional feature vector the Mahalanobis distance is equivalent to the Euclidean distance.

For simplification reasons, let us note Σ_j^{-1} as S, μ_j simply as μ and $r = d_M^2(x,C_j)$. Thus, the distance metric becomes:

$$r = (x-\mu)^\intercal S(x-\mu) = x^\intercal Sx - 2\mu^\intercal Sx + \mu^\intercal S\mu.$$

Let suppose that we want to protect x and that μ and S were previously computed on plaintexts. Thus, the computation of the distance has to be realized using the encrypted form of x leading to an encrypted r. The last term is only using plaintext data so it is easy to calculate it and to add it to the other terms using the properties of the underlying homomorphic scheme. Also, since $\mu^\intercal$ is a vector with the same dimension as x, the term $\mu^\intercal Sx$ is linear (the scalar tensor between a plaintext vector and an encrypted one) and can be computed with an additive homomorphic scheme. The first term $\alpha = x^\intercal Sx$ is a little bit more complicated. If $\sum_{j=1}^n s_{ij}x_j = y$ then:

$$\alpha = \sum_{i=1}^n x_i y_i = \sum_{i=1}^n \sum_{j=1}^n s_{ij}x_i x_j = \sum_{i=1}^n \sum_{j=1}^n s_{ij} z_{ij} \qquad (1)$$

with $z_{ij} = x_i x_j$. So, if we have at our disposal the n ciphertexts x_i and the $n(n-1)/2$ ciphertexts z_{ij}, we can use an additive homomorphic cryptosystem. If we want to avoid the transfer of the quadratic terms from the client to the server, we can also make appeal to a homomorphic cryptosystem with a multiplicative depth of at least 1. An example is the encryption scheme from (Catalano and Fiore, 2014), extension of Paillier cryptosystem, allowing to perform one multiplication over encrypted data. One can also use the so-called leveled homomorphic cryptosystems, such as the Ring-LWE-based (e.g. (Brakerski et al., 2011)), more complex but with more computing capabilities and quantum-safe. Let us remember that the multiplicative depth, notion related to the number of consecutive multiplications one can execute on ciphertexts, is a important characteristic of these leveled homomorphic schemes, defining the maximum allowable level of noise for a given set of parameters (the

multiplication inducing a much bigger noise than the addition).

In the following, for simplicity sake, we explain the case of a classical architecture, in which the client has some private data, a single attribute vector, and makes appeal to a classification service in order to obtain a label for this data only. Of course, the protocols described below still can be applied if, instead of a single instance, we have to label k such instances, as is the case with the district manager in the scenarios described in Section 3.

4.3 Prediction Step on Homomorphic Encrypted Data

Additive Homomorphic Encrypted Data. Let us suppose that the system relies on an additive homomorphic cryptosystem and analyze more in details the overall protocol required by the classification service.

The client having the sensitive data to be protected in the form of an attribute vector x of size n, has to encrypt, using his public key, for each vector each component x_i as well as the products $x_i x_k$, with $i,k \in \{1 \dots n\}$ and $i \leq k$. He send these encrypted data to the service provider which has the model build on clear data, i.e. for the Gaussian classifier, the m classes mean μ_j and the inverse of the covariance matrix Σ_j for $j \in \{1, \dots, m\}$. When receiving an encrypted vector to be labeled, the server computes the distances as described in the previous section and sends these encrypted m ciphertexts to the client. The client decrypts the distances using his secret key, then performs the sorting and selects the minimal distance, corresponding to the classes his data belongs. Moreover, the access to all clear distances can give the client an idea not only of the class he belongs to now but also how far is it from the other classes and, in some way, the needed effort in order to target a better label. The main drawback is the communication overhead induced by sending the quadratic terms from the client to the server.

Leveled Homomorphic Encrypted Data. If the underlying homomorphic scheme allows to realize at least one multiplication on the ciphertext, there are some slight modifications in the protocol for the labeling step. This time, the client having the attribute vector x of size n will encrypt only the components $\{x_i\}$ and send them to the service provider. As such, the quantity of upload information is in this case linear in the number of attributes. The server computes the m encrypted distances as previously, with the mention that now it is able to compute alone the first term α since one multiplication is allowed. The remaining of the protocol is similar to the previous one, on the client side.

Moreover, this protocol can be improved due to the batching technique (Smart and Vercauteren, 2014), allowing to pack and encode data for processing it in parallel in a SIMD (Single Instruction Multiple Data) fashion without any extra cost. As such, all the m distances can be embedded in slots and encrypted with a single ciphertext, computed in parallel by the service provider, and send in a single shot to the client (and thus reducing the download communication cost).

Threat Model. For both protocols, one could argue that the client could infer the properties model of the server from the information he has access to, through repeated requests and statistical inference. We focus in this paper more on protecting the client data from a honest-but-curious or not secured enough service provider and one could easily imagine protection mechanisms also for the server side (e.g. no more than a given number of requests). Note also that the security model we present here is not designed to assure the model nondisclosure in the case of collusion between several clients. As for the underlying homomorphic cryptosystems, they assure semantic security (Goldwasser and Micali, 1982).

Let us now give some details about the homomorphic schemes we used for testing the feasibility of our approach, with the remark that the general principles remain the same if other additive or leveled homomorphic schemes are deployed.

4.4 Homomorphic Cryptosystems

Paillier Additive Encryption Scheme. As described previously, it appears that an additively homomorphic system is enough in order to execute the classification algorithm. We have chosen Paillier cryptosystem (Paillier, 1999) a well-known and popular additively homomorphic cryptosystem. Let us recall here some of its main characteristics allowing the distance computation in the encrypted domain.

Let p and q denote two large primes and $n = pq$. Then, the cleartext domain of the Paillier system is $\mathbb{Z}_n$ and the ciphertext domain is $\mathbb{Z}_{n^2}$. Additionally, let $\lambda = \text{lcm}(p-1, q-1)$ and $g < n^2$ be randomly chosen such that

$$\gcd(L(g^{\lambda} \mod n^2), n) = 1,$$

with $L(u) = \frac{u-1}{n}$. The public (encryption) key is provided by n and g whereas the private (decryption) key in given by p and q or, equivalently, λ. Then, encryption is done by computing

$$c = \text{enc}(m) = g^m r^n \mod n^2, \qquad (2)$$

where $m < n$ is the message and r is uniformly chosen in $\mathbb{Z}_n$. Letting $D = L(g^{\lambda} \mod n^2)$ and D^{-1} its multiplicative inverse in $\mathbb{Z}_n$, decryption is then performed by evaluating

$$m = \text{dec}(c) = L(c^{\lambda} \mod n^2) \times D^{-1} \mod n.$$

More importantly for the present purpose, this cryptosystem has the following homomorphic properties:

1. $\text{dec}(\text{enc}(m_1)\text{enc}(m_2)) \mod n^2 = m_1 + m_2 \mod n$ (addition of two encrypted messages).
2. $\text{dec}(\text{enc}(m)g^k) \mod n^2 = m + k \mod n$, for all $k \in Z_n$ (addition of an encrypted message to a clear integer).
3. $\text{dec}(\text{enc}(m)^k) \mod n^2 = km \mod n$, for all $k \in Z_n$ (multiplication of an encrypted message by a clear integer).

BGV Encryption Scheme. This leveled homomorphic encryption scheme, based on the Ring-Learning with Errors problem, uses a series of different integer modulus for ciphertexts evaluation, allowing the modulus switching between these modulus in order to reduce the noise. Let $\mathbb{A} = \mathbb{Z}[x]/\Phi_m(x)$ be the ring of integers modulo the m-th cyclotomic polynomial. The ciphertext space is composed of vectors over the polynomial ring $\mathbb{A}_q = \mathbb{A}/q$ where q is an odd modulus evolving during homomorphic evaluation. The cleartext domain is the ring of polynomials $\mathbb{A}_t$, with the native plaintext being $t = 2$ but larger modulus allowing to execute operations on integers modulo t are also possible.

In its batching version, the plaintext space ring $\mathbb{A}_t$ is factored into sub-rings (through CRT-factorization of $\Phi_m(x)$ modulo t) such that the operations of addition and multiplication can be applied on each sub-ring independently. As such, it is possible to pack several several messages into the slots of a single ciphertext and execute the homomorphic operations in parallel on all messages at once.

Let us enumerate some of the important homomorphic operations supported by this cryptosystem:

1. $Keygen(1^{\lambda})$ with λ security parameter (generation of the secret key sk, the public key pk and of an additional set of evaluation keys evk for key-switching in homomorphic operations).
2. $\text{enc}_{pk}(m)$ (encryption of the plaintext message $m \in \mathbb{A}_t$ using pk).
3. $\text{dec}_{sk}(ct)$ (decryption of the ciphertext ct using sk).
4. $\text{dec}_{sk}(Add(ct_1 ct_2)) = \text{dec}_{sk}(ct_1) + \text{dec}_{sk}(ct_2)$ (addition of two encrypted messages).
5. $\text{dec}_{sk}(Mult(ct_1, ct_2, evk)) = \text{dec}_{sk}(ct_1) * dec_{sk}(ct_2)$ (multiplication of two encrypted messages using if needed key-switching for reducing the noise).

More details are to be found in the original paper (Brakerski et al., 2011).

5 SYSTEM PROTOTYPE

5.1 Load Profiles Dataset and Energy Efficiency Rating

In order to reproduce the scenario of the residential district, we have used the CER Smart Metering Project dataset[1]. It represents a comprehensive data source as it encompasses several residential load profiles (4225 load profiles in total) with related environmental information. On top of proposing electricity consumption information, the dataset provides specific indications about the householders owners. Subsequently, we chose 40 residential load profiles among the available ones to create our district. The selected profiles have to fulfill the following conditions:

- No missing data between January 1st, 2010 and December 31st, 2010.
- Householders of the retained load profiles must have completed the information about the surface of the residential building and the number of occupants.
- Electric heating systems are installed in these buildings.

These conditions are defined so to enable us to express the energy efficiency of a residential building.

Regarding the ratings, we proposed to create clusters from the metrics for each load profile, each cluster defined by a label (ranged from 'A', heavy consumers, to 'F', light consumers) and an average metric (expressed in kWh/(year.m^2.occupant)). For this purpose, we simply applied a k-means (MacQueen, 1967) and we set the number of cluster at 6. This choice proposed the maximum number of rating levels where no cluster has a size of one. To determine the category of a residential building, a distance is computed between its metric and the ones of each cluster using homomorphic encryption.

5.2 Performances Results

All the experiments were realized using a standard workstation, with a processor Intel Core I7 at 2.6

[1] www.ucd.ie/issda/data/commissionforenergyregulationcer/

GHz, with 16 GB of RAM memory and Ubuntu 16.04 as operating system (on 64 bits). The performance tests use a home made C++ implementation of Paillier additive cryptosystem (based on GMP library) as well as HElib (Halevi, 2013), the open-source library from IBM, implementing the BGV cryptosystem. For the version based on Paillier, the code has parallelized sections for the encryption part and the distance evaluation (using pragma omp instructions). For the HElib-based tests, we implemented two basic solutions both using batching but one of them being more optimized.

Results for Paillier-based Prototype. Our experiments showed that, as expected, the size of the upload data increases with the number of instances we want to label and, also, with the number of attributes for each instance and their quadratic products. The download data is proportional with the number of households' instances to classify, the dimension of the attribute vector for each household (one dimension for our demonstrator) and the number of classes the model presents.

Table 1 shows the size, in bytes, of one ciphertext for different security levels (i.e. the modulus size) as well as the latency in seconds for uploading the encrypted data (column "UP") and downloading the 6 distances (column "DW") for all of the 40 instances, when considering a network with a throughput of 10 Mbps.

We have also been interested in measuring the processing times of the encryption, distance computation and decryption steps according to the size of the key. For this evaluation we defined a scenario by a couple of parameters: the key size and the step to execute. We considered 2 key sizes for this purpose, by analyzing the execution times when using 1024- and 2048-bit keys for the encryption of the 40 residential profiles, the computation of the distances for these instances with regards to the 6 classes and the decryption of the results for all the households. We collected valuable information after executing each scenario 40 times. Table 2 summarizes these execution times for each step and each key size. As expected, the larger the key size, the longer the execution times are, for each of the steps. This is particularly remarkable for the distance computation step, taking the most important part of the overall computation. Besides its dependency on the key size, the labeling is also proportional with the number of instances to classify (here 40), their dimension (here 1) and the number of the reference classes (here 6). The execution times for the encryption depend on the number of instances to classify and their dimension while the decryption processing times depend on the dimension and the number of references.

Table 1: Data communication for different key sizes.

Bits	Size/ciphertext (bytes)	Latency (sec)	
		UP	DW
1024	617	0.03	0.12
2048	1233	0.08	0.23

Table 2: Execution times (sec) of labeling steps for different keys.

Bits	Step	Avg.	Min.	Max.
1024	Encryption	0.76	0.42	2.25
	Labeling	3.97	3.23	5.73
	Decryption	0.49	0.34	1.91
2048	Encryption	3.96	3.19	5.45
	Labeling	19.84	17.97	22.85
	Decryption	2.69	2.47	4.34

Results for HElib-based Prototype. For the first solution based on HElib tests (named "SOL 1") and for a given attribute vector x with n elements, each of the attributes x_i, with $i \in \{1,\ldots,n\}$ is embedded in a different plaintext slot in the form of an integer modulo p^r where p is an arbitrary prime (which does not divide m) and r a small positive integer. This allows to encrypt all the attributes of x in the same ciphertext. The references, i.e. the means of the classes, are represented as m vectors of dimension n. As such, for one instance to label, we obtain m ciphertexts corresponding to the encrypted distances to each class. When such a ciphertext is decrypted, the sum on the slots for the obtained plaintext gives the clear distance to the associated class (modulo p^r).

In the second solution, named "SOL 2", we take advantage of the free plaintexts slots (usually the number of slots is much larger than the number of attributes) and, for a single instance x of dimension n to label with regards to m classes, we replicate it m times and embedded into the slots of a plaintext, by padding with 0 the remaining space. In this configuration, the means are expressed as a single array of dimension $m \times n$ and we can compute all the distances in the same time using a single ciphertext. Once received and decrypted, one can obtain the clear distances by making the sum on sub-sets of successive slots. The necessary condition for this approach is that the number of slots has to be higher or equal to $m \times n$.

Table 3 shows two configuration of parameters for HElib testing we chosen in order to have s, the right number of slots, (sufficient but not too large) and a security level of at least 80.

Table 3: Parameters for HElib tests.

Test	m	p	r	L	s	Security
TEST 1	6679	2	8	3	42	180.46
TEST 2	8253	2	8	4	12	92.17

Tables 4 - 5 highlight the data size for both solutions when using the first and respectively the second set of HElib parameters (TEST 1 and respectively TEST 2). As previously, the latency is computed in seconds for uploading the encrypted data (column "UP") and downloading the 6 distances (column "DW") for all of the 40 instances, when considering a network with a throughput of 10 Mbps. This time, the size of a ciphertext is much larger than the one for a Paillier encrypted data (several thousands of kbytes versus thousands of bytes), due to the complexity of BGV cryptosystem. We also note that the second solution allows to decrease the download latency. In fact, instead of sending m ciphertexts, only one is sent back to the client for decryption.

Table 4: Data communication for different key sizes (TEST 1).

SOL	Size/ciphertext (kbytes)	Latency (sec)	
		UP	DW
1)	290.98	9.31	55.87
2)	290.98	9.31	23.14

Table 5: Data communication for different key sizes (TEST 2).

SOL	Size/ciphertext (kbytes)	Latency (sec)	
		UP	DW
1)	204.61	6.54	98.30
2)	204.61	6.54	16.38

Tables 6 - 7 summarize the execution times obtained for the first and respectively second solution, when using the above configurations of parameters, for labeling the 40 households relying on 6 references.

Table 6: Execution times (sec) of labeling steps for different key sizes (TEST 1).

SOL	Step	Avg.	Min.	Max.	Context Reading
1)	Enc.	0.67	0.60	0.79	7
	Label	9.56	9.02	10.51	
	Dec.	27.49	25.64	31.44	
2)	Enc.	0.71	0.64	0.91	7.15
	Label	1.65	1.48	2.10	
	Dec.	4.36	4.05	5.02	

Table 7: Execution times (sec) of labeling steps for different key sizes (TEST 2).

SOL	Step	Avg.	Min.	Max.	Context Reading
1)	Enc.	0.83	0.80	0.98	4.06
	Label	14.57	13.55	15.32	
	Dec.	9.80	9.25	10.55	
2)	Enc.	0.90	0.81	1.31	4.31
	Label	2.45	2.31	3.27	
	Dec.	1.58	1.38	2.57	

We consider that the context reading (the parameters and the keys reading) is realized once for all the 40 instances and, as the results indicate, depends on the set of initial HElib parameters. The results of execution times of the second optimized solution (SOL 2) are of better quality than for the first solution for the labeling and decrypting step, which looks right since we are executing the homomorphic evaluation and the decryption on a single ciphertext. Also, we obtain that for the second set of parameters (TEST 2), aiming a smaller security level, the processing times are smaller that the ones for the first set (TEST 1) which seems quite normal.

Finally, when comparing the second optimized solution with Paillier-based prototype on 2048 modulus, we remark that in general the execution times for encryption and labeling steps are faster but the decryption takes longer.

Of course, these are just some preliminary tests using HElib and a more thoughtful analysis of the parameters setting is necessary. Moreover, we can imagine several solutions for improving the performances. One of the problems we have in the current form is that most of the time passes in the context reading. Also, for now, the 40 instances are executed sequential and in the future this treatment could be also parallelized.

6 CONCLUSION AND PERSPECTIVES

This paper presents a demonstrator of a practical implementation of a secure energy data classifier to be deployed in a Smart City. The system was tested with a homomorphically additive cryptosystem and a leveled homomorphic scheme and achieves performances acceptable in a real-world setting. The results obtained attested the effectiveness of our proposal and the ability of our solutions to perform processes on data while guaranteeing privacy. This is just a first proposal of a secure rating energy service

using homomorphic encryption and thus many improvements can be imagined. First at all, we plan to implement and test the classification algorithm using other homomorphic cryptosystems (e.g., more recent third generation homomorphic schemes such as (Gentry et al., 2013)). At the same time, we will focus on the scalability of such an application and the subsequent impacts on processing performance. Secondly, one could imagine a more complex classification algorithm, less naive than the Gaussian one along with a more thorough evaluation process of the accuracy of the proposed service. Last but not least, one has to think of the way the labeling provided by this outsourced service could be usefully exploited by other tools, such as optimization scenarios, in order to endow the Program Administrator with a cost efficient overall solution.

REFERENCES

(2016). Energy rating. http://www.energyrating.gov.au/.

(2016). Uci: Machine learning repository.

Agrawal, R. and Srikant, R. (2000). Privacy-preserving data mining. *SIGMOD Rec.*, 29(2):439–450.

Bayardo, R. and Agrawal, R. (2005). Data privacy through optimal k-anonymization. In *Proceedings 21st International Conference on Data Engineering, 2005. ICDE 2005*, pages 217–228.

Bost, R., Popa, R., Tu, S., and Goldwasser, S. (2014). Machine learning classification over encrypted data. Cryptology ePrint Archive, Report 2014/331. http://eprint.iacr.org/.

Brakerski, Z., Gentry, C., and Vaikuntanathan, V. (2011). Fully homomorphic encryption without bootstrapping. Cryptology ePrint Archive, Report 2011/277. http://eprint.iacr.org/.

Catalano, D. and Fiore, D. (2014). Boosting linearly-homomorphic encryption to evaluate degree-2 functions on encrypted data. Cryptology ePrint Archive, Report 2014/813. http://eprint.iacr.org/2014/813.

Cavoukian, A., Polonetsky, J., and Wolf, C. (2010). Smartprivacy for the smart grid: embedding privacy into the design of electricity conservation. *Identity in the Information Society*, 3(2):275–294.

CNIL (2012). Pack de conformite sur les compteurs communicants. Technical report.

De Silva, D., Yu, X., Alahakoon, D., and Holmes, G. (2011). A data mining framework for electricity consumption analysis from meter data. *IEEE Transactions on Industrial Informatics*, 7(3):399–407.

Gentry, G., Sahai, A., and Waters, B. (2013). Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *CRYPTO*, pages 75–92. Springer.

Goldwasser, S. and Micali, S. (1982). Probabilistic encryption and how to play mental poker keeping secret all partial information. In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, STOC '82, pages 365–377, New York, NY, USA. ACM.

Graepel, T.and Lauter, K. and Naehrig, M. (2012). Ml confidential: Machine learning on encrypted data. *IACR Cryptology ePrint Archive*, 2012:323.

Halevi, S. (2013). Helib - an implementation of homomorphic encryption. https://github.com/shaih/HElib.

Kantarcioglu, M. and Clifton, C. (2004). Privately computing a distributed k-nn classifier. In *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases*, PKDD '04, pages 279–290. Springer-Verlag New York, Inc.

Kim, H., Marwah, M., Arlitt, M., Lyon, G., and Han, J. (2010). Unsupervised disaggregation of low frequency power measurements. In *In Proceedings of SIAM Interational Conference on Data Mining*, pages 747–758.

Li, F., Luo, B., and Liu, P. (2010). Secure information aggregation for smart grids using homomorphic encryption. In *2010 First IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 327–332.

Lindell, Y. and Pinkas, B. (2000). Privacy preserving data mining. In *JOURNAL OF CRYPTOLOGY*, pages 36–54. Springer-Verlag.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In Cam, L. M. L. and Neyman, J., editors, *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press.

Nikolaou, T., Kolokotsa, D., Stavrakakis, G., Apostolou, A., and Munteanu, C. (2015). Review and state of the art on methodologies of buildings' energy-efficiency classification. In *Managing Indoor Environments and Energy in Buildings with Integrated Intelligent Systems*, pages 13–31. Springer.

NIST (2010). Guidelines for smar grid cyber security. Technical report.

Paillier, P. (1999). Public-key cryptosystems based on composite degree residuosity classes. In Stern, J., editor, *Advances in Cryptology - EUROCRYPT 99*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238. Springer Berlin Heidelberg.

Samanthula, B., Elmehdwi, Y., and Jiang, W. (2014). k-nearest neighbor classification over semantically secure encrypted relational data. *CoRR*.

Smart, N. P. and Vercauteren, F. (2014). Fully homomorphic simd operations. *Des. Codes Cryptography*, 71(1):57–81.

Verdu, S. V., Garcia, M., C., S., Marin, A. G., and Franco, F. J. G. (2006). Classification, filtering, and identification of electrical customer load patterns through the use of self-organizing maps. *IEEE Transactions on Power Systems*.

Vetter, B., Ugus, O., Westhoff, D., and Sorge, C. (2012). Homomorphic primitives for a privacy - friendly smart metering architecture. In *SECRYPT*, pages 102–112.

Zirm, M. and Niedermeier, M. (2012). The future of homomorphic cryptography in smart grid applications. In *Procedeedings of the 3rd IEEE Germany Student Conference Passau 2012*.