Multi-Party Metering: An Architecture for Privacy-Preserving Profiling Schemes

Cettina Barcellona[1], Pietro Cassará[3], Giuseppe Di Bella[1], Jovan Golić[2], Ilenia Tinnirello[1]
[1]Universitá degli Studi di Palermo, Italy
[2]Telecom Italia, Italy
[3]ISTI CNR of Pisa, Italy.

***Abstract*—Several privacy concerns about the massive deployment of smart meters have been arisen recently. Namely, it has been shown that the fine-grained temporal traces generated by these meters can be correlated with different users behaviors. A new architecture, called multi-party metering, for enabling privacy-preserving analysis of high-frequency metering data without requiring additional complexity at the smart meter side is here proposed. The idea is to allow multiple entities to get a share of the high-frequency metering data rather than the real data, where this share does not reveal any information about the real data. By aggregating the shares provided by different users and publishing the results, these entities can statistically analyze the consumption data, without disclosing sensitive information of the users. In particular, it is proposed how to implement a user profiling clustering mechanism in this architecture. The envisaged solution is tested on synthetic electricity consumption data and real gas consumption data.**

I. Introduction

Several public utility systems (such as the electricity, the gas, and the water distribution systems) are recently deploying smart meters for improving the management of the distribution network and offering better services to the users. Smart meters differ from conventional meters in that they can provide user consumption data to authorized parties (e.g. utility providers) not only in cumulative terms, but also in fine-grained temporal traces which can bring benefits to different actors.

Consider for example the electricity case. The service providers, i.e., the companies that purchase and sell electricity to consumers, can offer more advanced billing schemes (e.g., for reducing the user demand when the electricity cost is higher) and reduce operational costs due to manual readings. The operators of the transmission and distribution systems can reduce the energy wastes by knowing the exact load demand in different location areas, which itself facilitates the carbon reduction. Finally, end-users can become more aware of their energy consumption (thus developing new energy-saving habits) and can benefit from new billing schemes by exploiting local energy production from renewable sources.

However, despite of these benefits, several concerns about privacy issues have emerged recently. In [1], [2], it is discussed how the load signatures techniques can be applied to energy demand traces of users for revealing different users behaviors, such as the time intervals in which they are not at home, the usual awaking time, the frequency of the shower usage, the time spent in front of the television, etc.. This information can be used for different purposes, from target advertising, to detecting if users are not at home during a sick leave, and even to understanding the user religion (e.g., if the user minimizes the load demand on a specific day of the week).

In order to respond to these concerns, different solutions have been proposed for avoiding leakage of personal data in smart metering. These privacy issues especially arise for high-frequency meter readings (typically, with a granularity of a reading every 5-15 minutes), while a cumulative reading in a month or longer intervals reveals much less information. In [3], the authors distinguish between high-frequency and low-frequency data and propose a new architecture for smart meters, based on the use of pseudonyms for reading the high-frequency data in a given location area, without binding the data to a specific user. A different approach, based on the addition of a random noise [4] or exploitation of storage/production energy [5] systems have been proposed for changing the user-created load demand, thus hiding the load behavior to the load signature schemes. Other solutions exploit homomorphic cryptography for allowing some operations on the user data without disclosing the information related to a given user. In this case, sophisticated, time-consuming protocols between the meters and the providers can be required for certifying that the final bill of the user is evaluated according to the desired tariff policy, such as protocols for supporting zero-knowledge proof as proposed in [6].

In this paper, we propose to exploit the existence of multiple providers (dealing with the same utility or with different utilities) and the availability of data networks independent from the utility distribution network for processing the user data in an aggregated form, without accessing the fine-grained data of single users. In particular, due to its market importance, we focus on the user profiling operations, i.e., on the possibility of classifying the users according to the utilization habits for a given utility, while also discussing how the framework can easily support other operations (such as aggregation of the total demand in a given location area). The approach is based on

978-3-901882-56-2

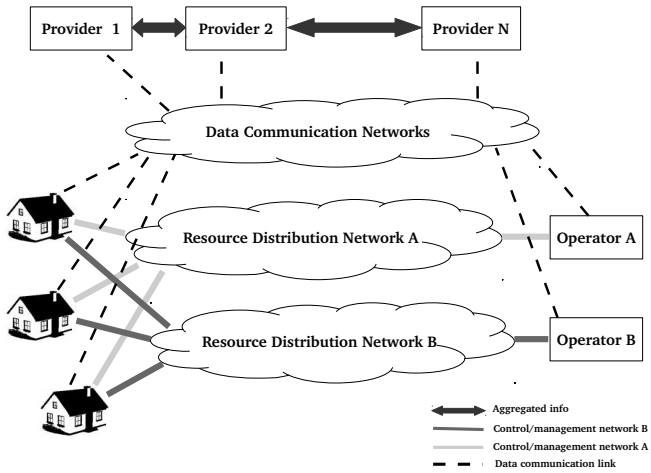


Fig. 1. Our reference scenario: smart meters are connected to the utility distribution network and to an independent data network.

a multi-party implementation of known clustering schemes working on secret shares of the user data [7], [8] in a privacy-preserving way. The scheme can be supported by the proposed architecture without requiring significant additional complexity at the smart meter side. Finally, we also show some profiling results that have been obtained by applying the proposed framework to simulated electricity data and actual gas metering data.

II. Multi-Party Metering Architecture

Although meters are under the complete control of utility providers and no tampering operations are allowed to the users, we envision a scenario in which the meters can support *by design* new privacy-preserving functionalities. We assume that the usual metering operations, such as the low-frequency readings of the cumulative consumption data, are not sensitive information for the users and can be signaled to the provider by usual means, e.g., by manual readings or remote readings through a data network. We also assume that, despite the fact that the network for reading the data in some cases can be physically built on top of the utility distribution network, as in the case of power line transmissions in the electric grid, each smart meter can access an independent public data network (i.e. the Internet), either because it is integrated with a DSL modem or because there is a local network between the meter and the modem.

Figure 1 summarizes the envisioned scenario and the involved actors: users are generally connected to multiple utility networks (energy, gas, water), and for each network multiple providers offer differentiated services by exploiting a distribution system under the control of one or multiple operators. (For simplicity, the figure indicates two operators.) All the actors (users, providers, and operators) are also connected to an independent data network. A different smart meter is usually required for each user utility, as in current deployments, but with the additional capability to be connected to the data network. While low-frequency data still belong to a single *party*, i.e., the provider to which the user has contractually subscribed, high-frequency data are sampled by the meter, but are not directly accessible to the service provider. Since these data are important to all providers (e.g., for offering better tariff policies) and operators (e.g., for minimizing the distribution network inefficiencies), the smart meter can send *to each of them* a share of the reading value rather than the value itself. For high-frequency measurements, the smart meter thus behaves as a multi-party meter, i.e., a meter available to multiple independent entities by providing shares of the measurements. We assume that independent providers and operators, who are in principle competitors, are motivated to cooperate for computing the desired aggregate public statistics, as linear functions of the shares. In other words, the entities reading the high-frequency shares of metering data are assumed to behave as in the *honest but curious* model, i.e., they work as prescribed by the multi-party profiling protocol described in the following section, but they can try to extract information from their data.

Different use cases of exploiting the fine-grained information provided by the high-frequency meter shares can be enabled in this scenario. For example, by simply summing all the shares read by each entity in a given location area and by publishing the results, it is possible to estimate the temporal behavior of the total utility demand in that area. Moreover, more sophisticated aggregations allow one to profile the user behaviors by clustering, for offering new service policies, for integrating the services offered by different utility networks, or for facilitating the entrance of new providers in the market.

III. User Profiling

In this section, we show how to exploit our architecture for profiling the users in a privacy-preserving manner. User profiling is based on data mining techniques known as classification and clustering, for which several algorithms are available in the literature. They are able of grouping a set of elements according to the similarity of their *features*, which is measured by using a distance metrics. Ideally, the distance metrics should be minimized within the same profile and maximized between different profiles.

Both clustering and classification schemes can work in multiple iterations. For example, hierarchical schemes attempt to organize data into a hierarchical structure, merging the clusters (profiles) found in previous steps, while the partitioning schemes work by splitting the profiles in successive iterations. Clustering schemes can be divided into *hard* and *soft* schemes: the hard schemes impose that an element belongs to one cluster only, while the soft schemes allow an element to belong to all clusters with different membership values.

Although our framework can support different clustering and classification solutions where the aggregate statistics

to be computed are linear, we here show how to implement the multi-party clustering operations by referring to a specific well-known clustering scheme called Fuzzy C-Means (FCM), which is an evolution of the basic K-means algorithm. We will define the clustering operations and multi-party implementation by representing user data as a vector, with integer-valued individual components representing different data features. The user profiles are then represented as vectors, whose size is equal to the data size, specifying the centroids of each cluster.

A. Fuzzy C-Means Algorithm

Let n be the number of m-dimensional data vectors to be profiled into c different clusters. If $\mathbf{d}_i \in \mathbb{R}^{1\times m}$ is the i-th data vector and $\mathbf{c}_j \in \mathbb{R}^{1\times m}$ is the j-th profile/cluster centroid vector, then the classical metric to be minimized in a hard clustering scheme, such as the K-means scheme, is given as the sum of the L2-norm distances between each data vector and the centroid vector of the cluster it belongs to, i.e., as:

$$\sum_{j=1}^{c} \sum_{\mathbf{d}_i \in C_j} \|\mathbf{d}_i - \mathbf{c}_j\|^2, \tag{1}$$

where C_j is the j-th cluster.

FCM [9] is a soft partitioning algorithm according to which the belonging relationship of data to clusters is fuzzy rather than deterministic. Specifically, the scheme defines a membership matrix $\mathbf{U} \in [0,1]^{n\times c}$, whose generic element u_{ij} is membership degree that the i-th data vector belongs to the j-th cluster. The matrix $\mathbf{U}$ satisfies two conditions: i) $\sum_{i=1}^{n} u_{ij} > 0$, i.e., each cluster j includes at least one data element with non-zero membership degree, and ii) $\sum_{j=1}^{c} u_{ij} = 1$, i.e., each data element i is contained in one of the considered clusters with the maximal membership degree 1. The function to be minimized by clustering is given by:

$$\sum_{j=1}^{c}\sum_{i=1}^{n} u_{ij}^{f} \|\mathbf{d}_i - \mathbf{c}_j\|^2 \tag{2}$$

where f is the fuzzification parameter affecting the shape of the clusters in the range $]1,\infty[$, which is typically set to 2.

Starting from a random initialization of the cluster centroids $\mathbf{c}_j(0)$,the scheme works at each step t as follows:

1) Update of the membership matrix: each data element i is fuzzily assigned to each cluster j with the membership degree computed as:

$$u_{ij}(t) = \frac{1}{\sum_{l=1}^{c} \left(\frac{\|\mathbf{d}_i - \mathbf{c}_j(t-1)\|}{\|\mathbf{d}_i - \mathbf{c}_l(t-1)\|}\right)^{\frac{2}{f-1}}} \tag{3}$$

2) Update of the cluster centroids: for each cluster j, the cluster centroid is computed by minimizing the objective function (2) as:

$$\mathbf{c}_j(t) = \frac{\sum_{i=1}^{n} u_{ij}(t)^f \cdot \mathbf{d}_i}{\sum_{i=1}^{n} u_{ij}(t)^f} \tag{4}$$

3) Convergence verification: if $\|\mathbf{U}(t) - \mathbf{U}(t-1)\| < \epsilon$, then the scheme is stopped.

B. Multi-Party Implementation

The clustering algorithm described in the previous section is based on the knowledge of the private data vector $\mathbf{d}_i$ for each user $i \in [1,n]$, where the membership degrees are computed from these vectors and the public cluster centroid vectors. The clustering scheme can be implemented by using a trusted server which iteratively computes (3) and (4) from the private data vectors. The scheme can be implemented in a privacy-preserving way by using homomorphic encryption in order to compute the linear functions in (4) by the server, where (3) is computed by individual users and the values of $u_{ij}(t)^f$ and $u_{ij}(t)^f d_{il}$ are sent encrypted to the server, in each iteration. Here, d_{il} denotes the l-th coordinate of $\mathbf{d}_i$, $l \in [1,m]$.

However, the homomorphic encryption is computationally heavy. Instead, we here propose an implementation of the same clustering scheme by using multi-party computation based on linear threshold secret sharing. This is possible since the functions in the nominator and denominator of (4) are linear. The secrets are the private data elements $u_{ij}(t)^f$ and $u_{ij}(t)^f d_{il}$ used in (4). They are computed by individual users (in our case, smart meters) and each of them is split into shares and distributed to K entities called nodes (in our case, utility providers and operators and, eventually, an external profiling node, as indicated in Section II). For the proposed (K,K) threshold scheme, it suffices [10] to represent each secret s as the modular integer sum of K random shares $r_1, \cdots r_K$, where the modulus p is chosen to be larger than n times the maximum measurement value d_{il}. More precisely, shares are generated by choosing $K-1$ random values $r_1, \cdots, r_{K-1}$ from the range $[0, p-1]$ and by computing the last share as $r_K = s - r_1 - \cdots - r_{K-1} \bmod p$.

One of the nodes serves as a profiler. In each iteration, it collects the aggregated shares of the other nodes, computes (4) in a privacy-preserving way, and then distributes the cluster centroids to the smart meters. The user data correspond to high-frequency smart meter readings captured every x minutes in an interval of one day (i.e., the data vector dimension is $24 \cdot 60\ min/x$). Rather than sending the data vector at the end of the day, in our approach, the i-th meter sends K shares of the iteratively computed high-frequency data to different nodes S times, S being the number of iterations required to meet the convergence criterion in the clustering algorithm.

More precisely, the multi-party implementation of the clustering scheme works as follows. The profiler sends the initial random centroids $\mathbf{c}_j$ and the fuzzification parameter f to all the smart meters, which locally evaluate their vector $\mathbf{u}_i^f \in \mathbb{R}^{1\times c}$, with components u_{ij}^f, and their matrix $\mathbf{M}_i = \mathbf{d}_i^T \cdot \mathbf{u}_i^f \in \mathbb{R}^{m\times c}$, with components $u_{ij}^f d_{il}$. The iteration number t is here omitted for simplicity. The vector and the matrix are then randomized in K shares

$\mathbf{r}_k(\mathbf{u}_i^f) \in \mathbf{R}^{1\times c}$ and $\mathbf{r}_k(\mathbf{M}_i) \in \mathbf{R}^{m\times c}$, $k \in [1, K]$, to be sent to the K nodes. The k-th node, in turn, linearly aggregates the shares of all the smart meters in a vector $\bar{\mathbf{r}}_k = \sum_{i=1}^{n} \mathbf{r}_k(\mathbf{u}_i^f) \bmod p$ and a matrix $\bar{\mathbf{M}}_k = \sum_{i=1}^{n} \mathbf{r}_k(\mathbf{M}_i) \bmod p$, for each $k \in [1, K]$.

The profiler receives all the aggregated shares and exploits the linearity to evaluate the two sums in (4). Let $\bar{m}_k(j,l) = \sum_{i=1}^{n} r_k(d_{il}u_{ij}^f) \bmod p$ denote a component of the aggregated matrix $\bar{\mathbf{M}}_k$ and let $\bar{r}_k(j) = \sum_{i=1}^{n} r_k(u_{ij}^f) \bmod p$ denote a component of the aggregated vector $\bar{\mathbf{r}}_k$. Then, by summing up modulo p all the aggregated shares $\bar{m}_k(j,l)$, the profiler recovers $\sum_{i=1}^{n} d_{il}u_{ij}^f \bmod p$. Similarly, by summing up modulo p all k aggregated shares $\bar{r}_k(j)$, it recovers $\sum_{i=1}^{n} u_{ij}^f \bmod p$. Now, under the condition that p is larger than n times the maximum value of d_{il}, each of the two modular sums reduces to the desired integer sum in (4). Namely, the profiler computes the l-th component of the j-th cluster centroid as:

$$\mathbf{c}_{jl} = \frac{\sum_{k=1}^{K} \bar{m}_k(j,l) \mod p}{\sum_{k=1}^{K} \bar{r}_k(j) \mod p}. \quad (5)$$

The new centroids are then sent to all the smart meters in order to update their membership degree vector $\mathbf{u}_i$ and the matrix $\mathbf{M}_i$ and repeat the whole process in a new iteration.

The multi-party implementation of the scheme has some overheads, both in terms of additional complexity and in terms of additional bandwidth required to transmit the data. However, the additional complexity is practically negligible, being limited to the evaluation of the shares to be performed by each meter, i.e., to the extraction of $K-1$ random variables and a computation of a modular sum for $mc + c$ different secret components of $\mathbf{M}_i$ and $\mathbf{u}_i$. The bandwidth overhead for a single smart meter is proportional to $SK(mc + c)$, where S is the number of iterations of the clustering algorithm.

A private data vector $\mathbf{d}_i$ is represented by a public cluster centroid $\mathbf{c}_j$ that maximizes the membership degree u_{ij}. This cluster centroid may then be used for user profiling by the utility providers. The main point of the whole approach is that the user profile can be computed by the smart meter without revealing the concrete private data vector to other entities. This ensures the privacy, provided that the number of clusters is reasonably small with respect to the granularity of measurement data.

IV. Performance Evaluation

In order to assess the effectiveness of the proposed approach, we simulated the proposed architecture in MATLAB, by implementing the FCM clustering algorithm at the profiler node, for both the scheme working on actual data and the scheme working on data shares generated by the smart meters. In both the cases, we obtained the same profiling results. We want to remark that our main goal is not providing numerical results for quantifying user profiles, but rather demonstrating the operations and the accuracy of clustering schemes working on data shares.

TABLE I
Probability distributions of centroids alteration

	≤ 0.0015	≤ 0.0045	≤ 0.0154	≤ 1	≤ 5	≤ 20	≤ 56
K-Means	0	0	0	0.4	0.7	0.8	1
FCM	0.5	0.7	1	1	1	1	1

For generating the user data, we used both simulated data emulating domestic electricity consumption and real data quantifying industrial gas consumption. The system simulator has been used for better understanding of the effects of different parameters to be tuned in the clustering operations.

A. Tuning of the Algorithm Parameters

Different parameters have to be configured for running the clustering scheme: the total number of clusters c, the fuzzification parameter f, and the initial centroid $\mathbf{c}_j(0)$ of each cluster j. The scheme can be applied multiple times to the same data, in order to identify the number of clusters that better reflects different users behaviors. Different indicators have been proposed in the literature for comparing alternative clustering solutions applied on the same data [11]. In particular, we chose the Davies-Bouldin Validity index [12], because its definition can be easily supported in our architecture and does not depend on the clustering algorithm. It is important to emphasize that in practice this index can be computed in a privacy-preserving way, without revealing individual data vectors, by a technique similar to the one explained in this paper.

We ran the clustering scheme for different c values in the interval $[2, 10]$ and for different fuzziness values in the set $[1.5, 2, 2.5, 3]$. The best clustering results have been generally identified as the ones minimizing the clustering validity index. In some cases, when data are natively clustered in distant groups, we applied a hierarchical optimization by finding the optimal number of sub-clusters per each group.

We actually chose the FCM algorithm due to its low sensitivity to the initial conditions. Such a feature has also been verified in the numerical results by comparing the variability of the final clustering results under different initial centroid values. We also considered the sensitivity results of the simpler non-fuzzy version of the scheme (i.e., K-means) for identifying the best complexity/accuracy trade-off. Table I shows the cumulative probability distribution of the distance between the centroids of the same cluster obtained with different initial conditions for both the FCM and the K-means scheme. The distance is always lower than 1 KWh for the FCM scheme, while for the K-means scheme can be higher than 20 kWh in the 20% of the cases. These considerations justify the choice of the FCM scheme with randomly chosen initial centroid values.

B. Numerical Results with Synthetic Data

The synthetic data used for testing our framework have been obtained by using a simulator of electrical domestic loads. For each smart meter, the simulator allows one to

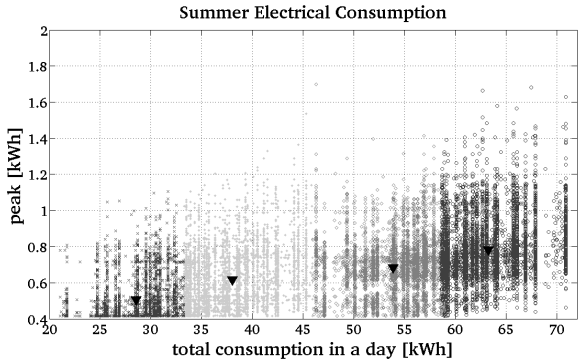


Fig. 2. Clulstering Electrical Summer Data

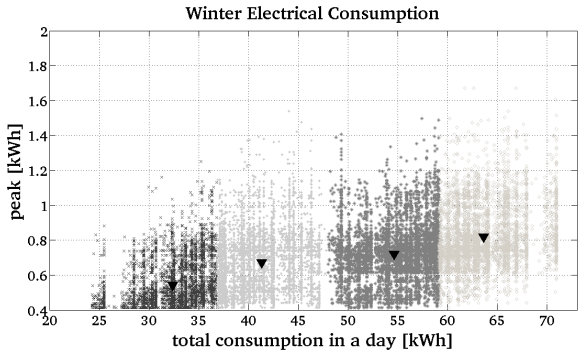


Fig. 3. Clulstering Electrical Winter Data

specify the energy class of the building and air conditioning system, the presence or absence of water heaters and photovoltaic systems, and the number of people living in the same unit. The switching-on probability of the domestic appliances and the intervals in which the appliances are kept active are modeled according to actual load traces and differentiated between summer and winter seasons.

We considered a total number of 800 users, generating a random daily load demand in 30 different days according to the configuration parameters of the relevant domestic unit. Each load demand curve is sampled into high-frequency meter readings, quantifying the energy consumed in intervals of 15 minutes. From these readings, we extracted two types of data to be clustered: a vector of 2 components, obtained from the previous one by considering the total daily consumption and the peak consumption, and a vector of 24 components corresponding to the meter cumulative readings in intervals of one hour.

For the simple case in which the data vector has two components only Figures 2 and 3 show the data collected, respectively, in thirty different summer and winter days, as well as the profiling results for $c = 4$, which was the number of profiles chosen by the simulator. We can see that two profiles are affected by seasonal variations (namely, the profiles corresponding to the lowest total and peak consumptions), while the two other profiles are almost unaffected. The profiles given by the coordinates of the clusters centroids have been obtained by considering the readings of the same meter on different days as different users. According to this approach, the profiler has to store the daily values of the aggregated shares sent by each utility provider for collecting all the $30K$ shares. We also implemented slightly different solutions (e.g., averaging the centroids obtained every day in a window of 30 days) and obtained similar results.

Figure 4a shows the user profiles obtained for the case in which the data vector includes 24 different components. Each profile is given by 24 different values of energy demand in different intervals of the day. There are profiles corresponding to a more uniform distribution of energy demand along the day as well as profiles requiring energy peaks of different durations. Figure 4b also shows the components of four different data vectors classified under the four clusters, respectively.

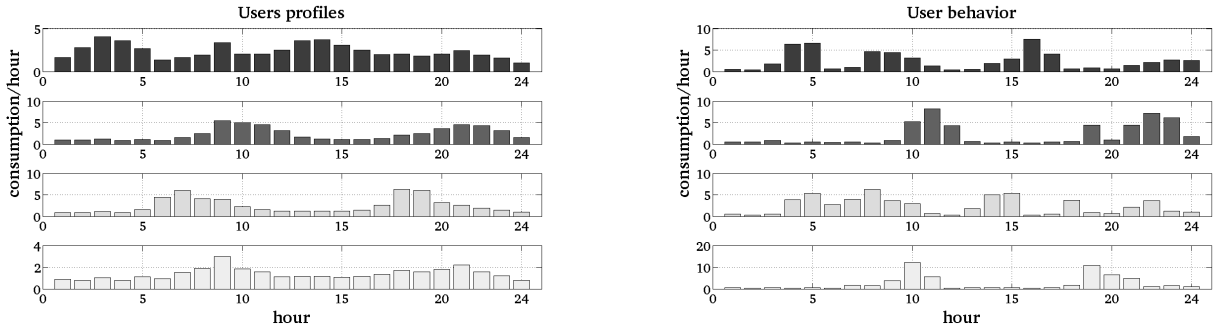
C. Numerical Results with Actual Data

The real data used for our tests are based on gas consumption data. In this case, the data do not refer to domestic users, but to important consumers such as industries, companies or schools. Apart from the consumption readings, the data also include additional information, such as the temperature and pressure readings and the longitude and latitude values of the smart meters.

As in the previous case, for plotting the profiles in a bidimensional coordinate system, we considered the total and peak value of the consumptions measured during the day and night and different clustering criteria for bidimensional data (e.g., day-time consumption and its relative peak, day-time and night-time consumptions, minimum pressure/temperature and maximum pressure/temperature, longitude and latitude, etc.). The data vectors were obtained from 200 users for the first 100 days of 2012. Since the data were real and no initial indication about the cluster numbers was available, we identified the best clustering solution by running multiple clustering schemes, under different c and f values. In many cases, we also considered the hierarchical optimization of the number of clusters, if the original data were clustered in distant sets. Figure 5 shows a profiling example referring to the day-time total and peak consumption. Data points belonging to different clusters have been colored with different gray scales. The figure has been plotted in a logarithmic scale to demonstrate that only a few data points have a peak consumption lower than $10m^3$ or higher than $100m^3$. In this case, the optimization of the validity index of the clustering solution corresponds to two clusters only. However, for further differentiating the behaviors of the majority of users, we applied a hierarchical optimization of the most populated cluster, thus providing a final number of nine different profiles.

V. Conclusion

Privacy concerns are very important for the future deployment of smart metering, especially in view of the fact that the amount of data collected in the future will



(a) (b)

Fig. 4. User profiles defined on the basis of per-hour energy consumption values and examples of metering data for users belonging to different profiles.

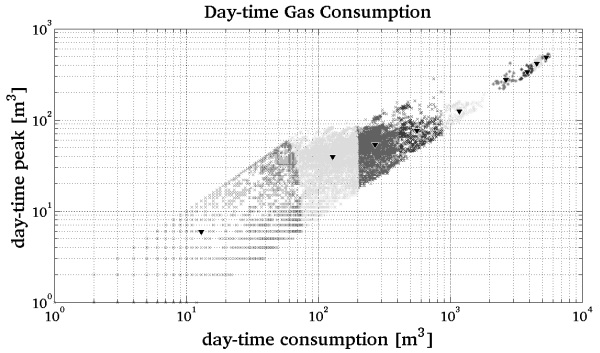


Fig. 5. Clustering day-time gas consumption vs. day-time peak.

be orders of magnitude more than the data collected from current meters. This data can easily be mined for estimating different user behaviors, as widely demonstrated by the application of load signature techniques.

In this paper, we address the smart metering privacy issue by proposing a new architecture for reading the high-frequency metering data. Rather than providing the data to one service provider only, the idea is sharing the readings among multiple independent parties that are motivated to cooperate for analyzing the aggregated data. In particular, we describe the multi-party implementation of a user profiling scheme based on the FCM clustering algorithm, which is able to work on random data shares, due to the linearity of functions evaluated in the algorithm. It enables the computation of data profiles without revealing the concrete metering data. Only if all the involved parties sharing the data collude together, the original data can be compromised. The scheme does not add significant computational overheads, while the bandwidth required for transmitting the data is proportional to the number of involved parties and iterations of the FCM algorithm.

We are currently investigating simple extensions of the scheme for optimizing different storage/complexity tradeoffs, by running the profiling operations with incremental data, i.e., with multiple data samples generated by the same users in different temporal windows. Further application scenarios, based on more general utilities such as the queries of DNS servers are also under investigation.

Acknowledgement

We would like to thank Prof. Maria Luisa Merani for her kind availability in sharing real metering data, Prof. Eleonora Riva Sanseverino for the fruitful discussions about the exploitation use cases of the framework, and Dr. Gaetano Zizzo for the generation of the synthetic data.

References

[1] "Smart meter data: Balancing consumer privacy concerns with legitimate applications," *Energy Policy*, vol. 41, no. 0, pp. 807 – 814, 2012.

[2] A. Molina-Markham, P. Shenoy, K. Fu, E. Cecchet, and D. Irwin, "Private memoirs of a smart meter," in *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*, ser. BuildSys '10. New York, NY, USA: ACM, 2010, pp. 61–66. [Online]. Available: http://doi.acm.org/10.1145/1878431.1878446

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

[4] J.-M. Bohli, C. Sorge, and O. Ugus, "A privacy model for smart metering," in *Communications Workshops (ICC), 2010 IEEE International Conference on*, 2010, pp. 1–5.

[5] G. Kalogridis, C. Efthymiou, S. Denic, T. Lewis, and R. Cepeda, "Privacy for smart meters: Towards undetectable appliance load signatures," in *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, 2010, pp. 232–237.

[6] A. Rial and G. Danezis, "Privacy-preserving smart metering," in *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society*, ser. WPES '11. New York, NY, USA: ACM, 2011, pp. 49–60. [Online]. Available: http://doi.acm.org/10.1145/2046556.2046564

[7] A. Shamir, "How to share a secret," in *Communications of the ACM*, vol. 22, November 1979, pp. 612–613.

[8] G. R. Blakley, "Safeguarding cryptographic keys," in *AFIPS Conf. Proc.*, vol. 48, 1979, pp. 313–317.

[9] W. Pedrycz, *Knowledge-Based Clustering: From Data to Information Granules*. Hoboken, NJ, USA: Wiley, 2005.

[10] M. Ito, A. Saito, T. Nishizeki, "Secret sharing schemes realizing general access structures," in *Proc. of the IEEE Global Telecommunication Conf., Globecom 87*, 1987, pp. 99–102.

[11] F. Kovacs, C. Legany, A. Babos, "Cluster Validity Measurement Techniques," in *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, Budapest, Hungary, November 2005.

[12] D. L. Davies and D. W. Bouldin, "A cluster seperation measure," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-1, no. 2, pp. 224–227, April 1979.