Unterweger *et al. EURASIP Journal on Information Security* (2016) 2016:21
DOI 10.1186/s13635-016-0044-1

EURASIP Journal on
Information Security

RESEARCH Open Access

Privacy-preserving load profile matching for tariff decisions in smart grids

Andreas Unterweger[1]*, Fabian Knirsch[1,2], Günther Eibl[1] and Dominik Engel[1]

Abstract

In liberalized energy markets, matching consumption patterns to energy tariffs is desirable, but practically limited due to privacy concerns, both on the side of the consumer and on the side of the utilities. We propose a protocol through which a customer can obtain a better tariff with the help of their smart meter and a third party, based on privacy-preserving load profile matching. Our security analysis shows that the protocol preserves consumer privacy, i.e., neither the load profile nor the matching result are disclosed to the utility, unless the consumer later decides to actually purchase the tariff. In addition, the utility's load profiles used for matching remain private, allowing each utility to offer special tariffs without disclosing the associated load profiles to their competitors. Our approach is shown to have a smaller ciphertext size than homomorphic encryption in practically relevant configurations. However, matching is only possible with up to about 98 % accuracy in general and 93.5 % based on real-world load profiles, respectively. Depending on the practical requirements, two protocol parameters provide a tradeoff between matching accuracy and ciphertext size.

Keywords: Smart meter, Load profile, Tariff, Matching, Privacy, Smart grid

1 Introduction

A global trend towards the modernization of energy grids is ongoing: Information and communication technologies are integrated into the energy grid infrastructures, creating so-called smart grids. A multitude of new use cases become possible, including the balancing of power generation and consumption, electric mobility, renewable energy sources, and real-time pricing. The modernization on the technical side is accompanied by a move towards deregulation on the regulatory side [1], thereby vastly increasing the number of choices on the side of the end consumer. In liberalized energy markets, the paradigm for the customers will be changed: They will be able to change tariff, provider or both frequently and to adapt dynamically. There are a number of pilot regions that have shown the benefit of this deregulation, e.g., the German *eEnergy* projects, most notably the *MeRegio* project [1].

Customers have to choose one tariff out of those offered from their local utilities. In order to select the optimal tariff, i.e., the one fitting their electricity consumption at a given time, customers need to match their current usage habits to the offered pricing schemes. As an example, consider an energy provider which is interested in moving energy consumption away from times with high demand because this decreases the amount of immediately available but expensive energy resources. In order to encourage customers to use energy at off-peak times, the provider could offer better tariffs for load profiles that are dissimilar to normal consumption profiles by, e.g., lowering a fixed-price tariff from 23 cents/kWh to 17 cents/kWh. As another example, in order to stimulate people loading their cars during the night instead of during the day, a load profile with high values during night might be associated with another cheap tariff. A convenient way to perform the matching of current and desired consumption profiles is for the customer to provide a current load profile, i.e., a recent energy usage record, to different energy providers or to a third party providing a matchmaking service for tariffs offered by various energy providers.

*Correspondence: andreas.unterweger@en-trust.at
[1] Josef Ressel Center for User-Centric Smart Grid Privacy, Security and Control, Salzburg University of Applied Sciences, Urstein Süd 1, 5412 Puch bei Hallein, Austria
Full list of author information is available at the end of the article

However, while this is a convenient approach, there are privacy concerns: It has been shown that personal information, such as lifestyle, religion, habitual patterns, sleep-wake cycles, and activities can be extracted from load profiles (e.g., [2, 3]). On the side of the energy providers, there is also reluctance to disclose the load profiles that different tariffs are based on (e.g., typical load profiles of customer groups), as this is internal information that, in a liberated market, can make a difference in commercial success [1]. Therefore, while both, the consumer side and the provider side, are interested in facilitating optimal matching, there are strong reasons against disclosing full information for the matchmaking process.

In this paper, we propose a method for tariff selection that preserves both consumer privacy and the internal company data of energy providers. More precisely, the load profiles of neither party are disclosed, as motivated above. The method outperforms previously suggested methods—in particular, it limits data expansion, which is an issue when classical homomorphic encryption is used. The trade-off is matching accuracy, which in the proposed method is not 100 %. Depending on choice of parameters, higher accuracy can be traded in for increased data expansion. However, in real-word application scenarios, the accuracy of 93.5 %, which can be realized with the proposed approach at negligible data expansion, is sufficient as will be shown.

The possible use cases of the proposed protocol are manifold. An exemplary use case would be for a number of utilities to (continually) analyze and cluster the energy usage data of their customers. Based on this analysis, different tariff groups can be created. For each tariff group, the matching data is provided to a third party, a broker which provides a matchmaking service, in a privacy-preserving way, i.e., this third party cannot access the content of the data in any way, but only perform matchmaking. The third party—the broker—collects the matchmaking data from any number of utilities. This broker could, for example, be an institution created (and possibly operated) by the regulator (who, by design, is interested in creating competition and providing freedom of choice to energy consumers). Consumers who wish to select the tariff which is best suited to their current consumption habits can provide their load profiles to the broker, again in a privacy-preserving way, i.e., the broker cannot access the consumer data as it may reveal sensitive information about them [4, 5]. The tariff selection will usually be done automatically, e.g., by using the communication capabilities of a smart meter. Note that the broker need not be trusted (there are, however, some security considerations, which are discussed in the paper). The broker matches the consumer load profile to the information provided by the utilities in a privacy-preserving way and determines the closest match. It then sends sufficient information back to the consumer's smart meter to determine the best-matching tariff, again in a privacy-preserving manner. During the whole process, no load information is disclosed either by the consumer or by the utilities. Additionally, and in contrast to an aggregation protocol, neither the escrow service itself nor the utilities systematically collect load profiles of the smart meters. The smart meters individually decide on their tariff, which is neither revealed to the broker nor to the other utilities.

In summary, this paper contributes (i) a novel approach based on nearest neighbor embedding [6] and oblivious transfer [7] for finding the best-matching tariff without revealing any load profiles to any involved party; and (ii) it is shown that this approach can achieve a matching performance comparable to classical homomorphic encryption, but with less data expansion.

The rest of this paper is structured as follows: In Section 2, related work is reviewed and previously proposed approaches are discussed. In Section 3, we describe our three-phase protocol in detail, i.e., initialization, matching, and oblivious transfer. Section 4 discusses both the security and complexity of the proposed protocol and outlines its advantages over similar state-of-the-art privacy-preserving protocols. In Section 5, we present a prototypical implementation and evaluate the matching performance of our protocol for real-world use cases. Finally, in Section 6, this paper is summarized and an outlook to future work is given.

2 Related work

In this section, we provide an overview of related work. We distinguish between secure distance computation methods which are suitable to compare load profiles and other related work including oblivious transfer and smart-grid literature with tariff selection and profile matching. Comparisons to our approach described in Section 3 are provided.

2.1 Secure distance computation

The following related work describes secure distance computation algorithms. In our approach, we compare load profiles using Euclidean distance measures, which is a similar type of computation.

Mukherjee et al. [8] propose a method where a number of selected Fourier transform coefficients are permuted and communicated between the involved parties. The properties of the Fourier transform are used to provide limited guarantees on distance preservation based on the selection of transform coefficients. This makes their approach probabilistic with the number of coefficients retained providing a tradeoff between privacy and accuracy. Although our approach is probabilistic as well, the level of privacy is not influenced by the parameters which affect the accuracy of our approach.

Ravikumar et al. [9] describe an algorithm for secure Euclidean distance computation. Their approach is probabilistic as well, but requires a high number of vectors to be compared in order to come close to the actual distance values. In contrast, our approach finds the minimum distance in nearly all cases, allowing for near-perfect matching.

Wong et al. [10] introduce transformations of database points and query points that enable a ranking of the database points with respect to their nearness to the query points suitable for k-nearest neighbor determination. However, their transformations are not distance-preserving which enhances security against attackers with known plaintexts, but limits possible applications.

Rane and Boufounos [6] and Boufounos and Rane [11] propose the use of distance-preserving embeddings for privacy-preserving matching and nearest-neighbor searches in the context of image retrieval. We apply these distance-preserving embeddings as one step in our proposed protocol and make use of their privacy-preserving properties.

Homomorphic cryptosystems can be used for secure distance computation [12, 13], e.g., in the context of image retrieval [14], fingerprint matching [15], and face recognition [16]. Kolesnikov et al. [17] combines homomorphic encryption for computing distances with Garbled circuits for choosing the point having the minimum distance to the query point. We provide a detailed comparison between our approach and additive homomorphic cryptosystems in Section 4.

2.2 Other related work

In our proposed approach, we make use of oblivious transfer [7, 18, 19]. It allows one of two parties to query one of an arbitrary number of items from the second party without revealing (i) which item has been requested and (ii) any of the other items. A detailed description of the use within our protocol is provided in Section 3.

From a use case point of view, apart from oblivious transfer, related work includes literature on tariff decisions, load profile matching, and forecasting as well as demand-response and demand-side management. For the latter two, Palensky and Dietrich [20] provide an overview. In general, privacy concerns and communication overhead on the smart meter side are seldomly addressed. We focus on these aspects by providing some examples below.

Caron and Kesidis [21] describe an approach where customers share load profile information so that the utility can achieve a smoother aggregate load profile. This is also possible with the approach proposed in our paper. However, the load profiles are not revealed to the utility, thus preserving the privacy of the customers.

Similarly, Shao and Zhang [22] attempt to reduce peak loads by incentivizing customers to shift time of use of electrical devices. This behavior can also be triggered when applying our approach to provide suitable template load profiles, albeit not in real time. Again, the load profiles do not need to be shared with the utility as opposed to the approach by Shao and Zhang.

Ramchurn et al. [23] follow a game-theoretic approach with customer incentives for reducing peak loads. They use a decentralized protocol, as opposed to our as well as to Caron and Kesidis's [21] and Shao and Zhang's [22] approach. The approach by Ramchurn et al. defers certain loads with defined probability imposing constraints on the customer's choices. In contrast, our approach gives the customer full authority on tariff decisions.

Another game-theoretic approach for shifting energy consumption is proposed by Mohsenian-Rad et al. [24]. In this setting, the smart meters of the users interact in order to minimize overall energy consumption. Their approach requires peer-to-peer communication with transmission overhead depending on the number of smart meters involved. Conversely, our approach does not require any peer-to-peer communication whatsoever.

3 Profile matching protocol

In this section, we propose our protocol which allows one smart meter to find an optimal tariff based on its load profile forecast. A number of template load profiles corresponding to tariffs from different utilities are used for this search. Throughout the process, none of the involved parties has access to the others' data. The process is therefore privacy-preserving.

3.1 Protocol description

Figure 1 illustrates the different steps of our protocol which are described in detail below. The protocol can be split into three phases, named *initialization*, *matching*, and *oblivious transfer*.

3.1.1 Initialization

Let the set of participating utilities be denoted as $\mathcal{U}$. Each utility $\mathbf{U}_u$, $u \in \{1, \ldots, |\mathcal{U}|\}$ (cf. Fig. 1, right) has a list of tariffs $T_{u,l}$ with corresponding template load profiles $\mathbf{L}_{u,l}$ (denoted as set $\mathcal{L}_u$ in Fig. 1 where utilities can have different numbers of load profiles $l \in \{1, \ldots, |\mathcal{L}_u|\}$). For example, utility $\mathbf{U}_1$ has a standard tariff $T_{1,1}$ for day workers and a night-owl tariff $T_{1,2}$ for night workers. The corresponding template load profiles $L_{1,1}$ and $L_{1,2}$ show peak loads at different times of the day opposite to the respective working hours.

Template load profiles allow each utility to control demand and response in a fine-grained manner, e.g., by rewarding customers with atypical load profiles so that peak loads are avoided. Although the tariffs need to be known to the customers for billing and transparency, the template load profiles are considered to be private to the

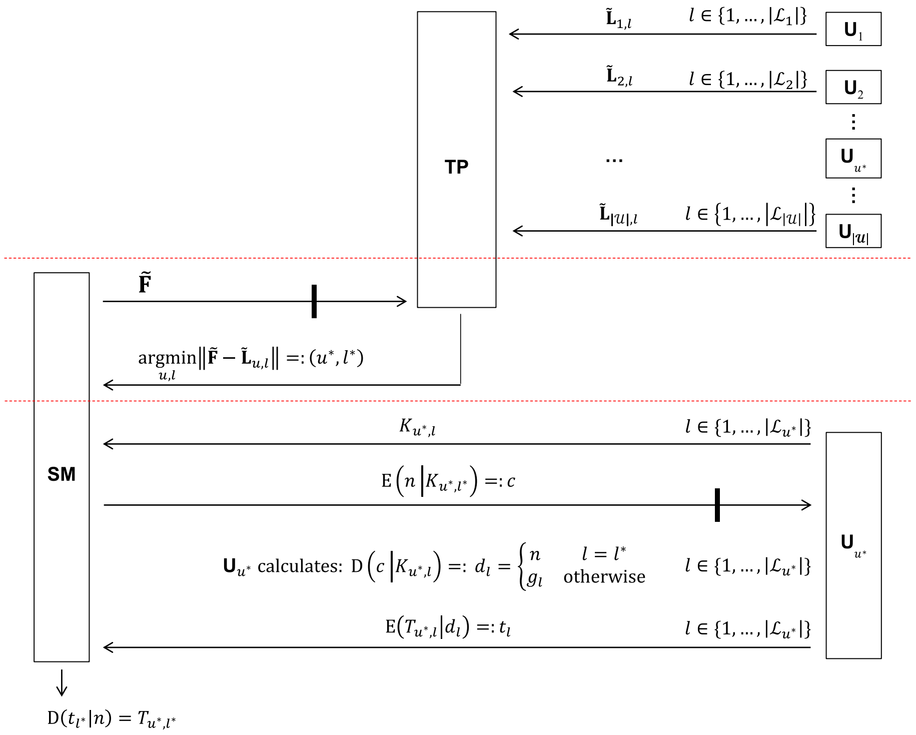
$\tilde{\mathbf{L}}_{1,l}$ $l \in \{1, \dots, |\mathcal{L}_1|\}$
$\mathbf{U}_1$
$\tilde{\mathbf{L}}_{2,l}$ $l \in \{1, \dots, |\mathcal{L}_2|\}$
$\mathbf{U}_2$
…
$\mathbf{U}_{u^*}$
$\tilde{\mathbf{L}}_{|\mathcal{U}|,l}$ $l \in \{1, \dots, |\mathcal{L}_{|\mathcal{U}|}|\}$
$\mathbf{U}_{|\mathcal{U}|}$
TP
$\tilde{\mathbf{F}}$
$\underset{u,l}{\operatorname{argmin}} \|\tilde{\mathbf{F}} - \tilde{\mathbf{L}}_{u,l}\| =: (u^*, l^*)$
SM
$K_{u^*,l}$ $l \in \{1, \dots, |\mathcal{L}_{u^*}|\}$
$\mathrm{E}\left(n \middle| K_{u^*,l^*}\right) =: c$
$\mathbf{U}_{u^*}$ calculates: $\mathrm{D}\left(c \middle| K_{u^*,l}\right) =: d_l = \begin{cases} n & l = l^* \\ g_l & \text{otherwise} \end{cases}$ $l \in \{1, \dots, |\mathcal{L}_{u^*}|\}$
$\mathbf{U}_{u^*}$
$\mathrm{E}(T_{u^*,l} | d_l) =: t_l$ $l \in \{1, \dots, |\mathcal{L}_{u^*}|\}$
$\mathrm{D}(t_{l^*} | n) = T_{u^*,l^*}$

Fig. 1 Overview of the proposed protocol: All participating utilities $\mathbf{U}_u \in \boldsymbol{\mathcal{U}}$ provide embedded template load profiles $\boldsymbol{\mathcal{L}}_u$ corresponding to their tariffs $T_{u,l}$ to a third party **TP**. A smart meter **SM** sends its embedded load profile forecast $\tilde{\mathbf{F}}$ to this third party which returns the best-matching utility index u* and its tariff index l*. Subsequently, the smart meter can request information about the best-matching tariff T_{u^*,l^*} from the best-matching utility $\mathbf{U}_{u^*}$ through oblivious transfer. *Black vertical bars* indicate rate limiters

respective utility. Therefore, no details are shared with competing utilities. In practice, template load profiles and tariffs may be significantly more complex than the example described above, with privacy on the utility side being a much more pressing issue.

In order to keep the template load profiles private, each utility calculates an embedding $\tilde{\mathbf{L}}_{u,l} \in \{0,1\}^m$ for each of its original template load profiles $\mathbf{L}_{u,l} \in \mathbb{R}^k$ as the first step of the *initialization* phase:

$$\tilde{\mathbf{L}}_{u,l} = \left\lceil \frac{\mathbf{A} \cdot \mathbf{L}_{u,l} + \mathbf{W}}{\Delta} \right\rceil \mod 2 \tag{1}$$

$\mathbf{A}$ is a random $m \times k$ matrix with i.i.d. Gaussian elements with mean 0 and variance σ^2, and $\mathbf{W}$ is a random m-dimensional vector with i.i.d. uniform elements in the range $[0, \Delta]$. Δ is both a quantization and a security parameter and described in detail in [6, 11]. The values of k and m are discussed in Section 5.

This embedding does not allow a potential attacker to reconstruct the original load profile, but preserves the distance between load profiles within a very small margin of error as described below. This enables comparisons of load profiles without the need to handle the respective original, private data.

The second step of the *initialization* phase of our protocol requires each utility to send all of its calculated embeddings $\tilde{\mathbf{L}}_{u,l}$ to a third party, denoted as **TP**. The need for this third party will become clear in the subsequent *matching* phase.

3.1.2 Matching

In this phase, a smart meter, denoted as **SM**, first creates a load profile forecast $\mathbf{F}$. It can either be based on past load profiles, e.g., of the current day or week, or on user input, e.g., prospective changes in work schedules. Similar to each utility in the preceding *initialization* phase, the smart meter first calculates an embedding of $\mathbf{F}$, denoted as $\tilde{\mathbf{F}}$. This way, the smart meter does not need to disclose its original load profile which may reveal sensitive information about the user.

As a second step, the embedding is sent to the third party, like the embeddings from the utilities in the previous phase. As a third step, the third party finds the best match for the load profile forecast out of the list of template load profiles from all utilities through the received embeddings. More precisely, it finds the template load profile with the smallest normalized Hamming distance to the forecast and outputs the template load profile index l* as well as the corresponding utility index u^*, i.e.,

$$(u^*, l^*) = \underset{u,l}{\operatorname{argmin}} ||\tilde{\mathbf{F}} - \tilde{\mathbf{L}}_{u,l}||_1. \tag{2}$$

This is possible due to the distance-preserving property of the embeddings (as described in more detail in [6]), where the Euclidean distance of the original data vectors is proportional to the normalized Hamming distance of the embedded vectors with a configurable small error ϵ, i.e.,

$$||\tilde{\mathbf{F}} - \tilde{\mathbf{L}}_{u,l}||_1 \sim ||\mathbf{F} - \mathbf{L}_{u,l}||_2 + \epsilon. \tag{3}$$

As a consequence, the probability that the best match in the original space and the best match in the embedded space coincide, is close to one:

$$\Pr\left[\underset{u,l}{\operatorname{argmin}} ||\tilde{\mathbf{F}} - \tilde{\mathbf{L}}_{u,l}||_1 = \underset{u,l}{\operatorname{argmin}} ||\mathbf{F} - \mathbf{L}_{u,l}||_2\right] = 1 - \delta. \tag{4}$$

This probability is referred to as matching accuracy. The smaller ϵ is, the higher the accuracy is.

The result (u^*, l^*) of the matching operation is a pair of indices identifying the utility $\mathbf{U}_{u^*}$ of the best match and its tariff T_{l^*}. However, no information about any load profile is revealed. The tuple (u^*, l^*) is transmitted to the smart meter as a fourth and final step, allowing the smart meter to fetch the tariff information from the utility $\mathbf{U}_{u^*}$ directly in the next phase in order not to disclose the actual tariff T_{l^*} associated with the index l^*.

The third party needs to be involved in the calculation above since neither the smart meter nor any of the utilities can be completely trusted to correctly perform calculations on the data. In addition, malicious parties are considered (see Section 4 below), i.e., they could manipulate their own input to bias the result in their favor or they could use multiple different inputs to derive additional information about the other parties' data. This is avoided by the use of an independent third party with a rate limiter (vertical black bars in Fig. 1) which prevents bulk-probing from the other parties. A detailed security analysis can be found in Section 4.

The third party can be thought of as a neutral party which performs only computations, e.g., a proxy of the Council of European Energy Regulators (CEER) which strives for a fair tariff market and competition. However, since any party may be distrusted, including the third party, the latter is not allowed to perform calculations on the actual data, but on embeddings only. This is why the latter need to be calculated by the other parties. This way, the third party is not able to access the original load profiles of either the smart meter or the utilities.

Note that the third party may collect statistics on the matching results (i.e., the indices (u^*, l^*)) of all smart meters. However, this can be rendered futile if the utilities regularly shuffle their template load profiles' indices in the *initialization* phase, e.g., each day. For example, $(u^*, l^*) = (1, 1)$ means a standard tariff and $(u^*, l^*) = (1, 2)$ a night-owl tariff, respectively, on one day, and the other way around, i.e., $(u^*, l^*) = (1, 1)$ means a night-owl tariff and $(u^*, l^*) = (1, 2)$ a standard tariff, respectively, on the next day.

Note that the third party could be omitted when using verifiable computing [25, 26]. However, this would induce substantial overhead which is critical on a device with limited capabilities, such as smart meters. In the subsequent *oblivious transfer* phase, the third party is not involved at all and hence never has access to the tariff T_{u^*,l^*} itself. Therefore, the third party does *not* need to be trusted.

3.1.3 Oblivious transfer

In the last phase of our proposed protocol, the smart meter sends a query to the utility $\mathbf{U}_{u^*}$ in order to obtain the best-matching tariff T_{l^*} based on its index l^*. The third party is not involved in this transaction and does therefore not obtain any information about the tariff itself apart from its index.

At this stage, the customer has not yet made the decision whether or not to switch to the best-matching tariff—they still need the tariff information for this. Thus, on the one hand, the smart meter must not disclose l^* to the utility since it would allow the utility to deduce information about the original load profile, even when the customer chooses not to switch to the matching tariff. On the other hand, the utility does not want to disclose all tariffs, some of which may exclusively be available to certain customers or groups. It only wants to disclose the tariff corresponding to the index l^*, but without being allowed to know this very index.

A solution for this is oblivious transfer [7, 18, 19]. It allows the smart meter to retrieve a tariff l^* from a vector of tariffs $T_{u^*,l}$, without the query (index) being known to the utility $\mathbf{U}_{u^*}$ and without any other tariffs being disclosed to the smart meter apart from T_{u,l^*}. In our use case, communication with $\mathbf{U}_{u^*}$ yields the tariff T_{u^*,l^*}.

The steps of the *oblivious transfer* phase can be summarized as follows: Initially, i.e., before any other communication, the utility $\mathbf{U}_{u^*}$ sends one key $K_{u^*,l}$ per template load profile $\mathbf{L}_{u^*,l}$ to the smart meter. The number of template load profiles is identical to the number of tariffs as described above. Thus, in total, $|\mathcal{L}_{u^*}|$ keys are sent.

Secondly, the smart meter generates a nonce n which is encrypted with the (l^*)th key. The encrypted nonce, c, is then sent to the utility. This step requires a rate limitation (e.g., one query per day) on the utility's side since the smart meter could otherwise query all available tariffs by iterating through the available indices. The rate limit has to be chosen such that the maximum number of allowed queries is lower than or equal to the average frequency at which utilities update their tariffs. If, for instance, utilities update their template load profiles daily, a rate limit of one query per day is sufficient.

Thirdly, the utility decrypts c with all of its keys, yielding decryptions d_l. For the key with the index sent by the smart meter, the decryption yields the nonce n, while, for all other keys, the decryption result is a garbage value g_l. For the utility, however, these are indistinguishable from the nonce. Thus, the utility cannot find out the index l^*.

Fourthly, the utility encrypts all tariffs $T_{u^*,l}$ with the decryption results d_l as keys, i.e., the nonce known to the smart meter for the index l^* and garbage values g_l for the others. The encrypted tariffs, t_l, are then sent to the smart meter which decrypts the (l^*)th encrypted tariff with the previously generated nonce. This yields the desired tariff T_{u^*,l^*}. The other tariffs cannot be decrypted since the encryption and decryption keys do not match, thus do not leak any information to the smart meter.

Encryption is performed with a symmetric cryptosystem. This allows for fast computation without ciphertext expansion. However, oblivious transfer requires the transmission of all tariffs in encrypted form at one point, which is analyzed in detail in Section 4 in terms of time and space complexity and compared to existing approaches.

4 Analysis

In this section, we review the protocol presented in this paper with respect to security, complexity, and in comparison to related work.

4.1 Privacy and security analysis

This protocol is privacy-preserving as none of the participating entities has access to the original load profiles of the others. Only in case of the third party and utility colluding, privacy may not be preserved fully as explained in detail below.

4.1.1 Assumptions

For the privacy and security analysis, we assume that the security parameters $(\mathbf{A}, \mathbf{W}, \Delta)$ are kept secret among the smart meters and the utilities. These parameters need to be distributed by a trusted party or the smart meters, and the utilities have to agree upon these parameters, e.g., through Diffie-Hellman key exchange [27]. Dealing with leaks of the security parameters to the remaining (third) party remains future work since our protocol is intended as a prototypical alternative to existing solutions.

We further assume that all parties involved in communication through our proposed protocol are authenticated, e.g., through X.509 certificates [28]. In addition, for the *initialization* and *matching* phases, we assume all communication links to be encrypted, e.g., by symmetric encryption such as AES [29]. Note that, in the *oblivious transfer* phase, no additional encryption is necessary. Nevertheless, the security properties of oblivious transfer are not discussed in this analysis, but can be found in literature, e.g., [7, 18, 19], since this analysis mainly focuses on the privacy impact of the information that is transferred.

4.1.2 Adversaries

Our protocol involves three distinct types of parties—smart meters, the third party, and utilities. As shown in Tables 1 and 2, a privacy breach occurs if (a) the load profile forecast of a smart meter is revealed to either the third party or any utility; (b) the template load profile of a utility is revealed to the third party, any smart meter or any other utility; or (c) the tariff that the smart meter receives at the end of the protocol is revealed to either the third party or the utility. Thus, all of the involved parties can be seen as a potential adversary, attempting to learn any information considered private (as described above) or manipulating the matching result for their own benefit.

For this analysis, we consider each type of party individually, explicitly distinguishing between a semi-honest and a malicious adversary. Additionally, collusions of different parties are analyzed.

4.1.3 Single semi-honest adversary

A *semi-honest smart meter* wants to learn any template load profile from any of the utilities. Since the smart meter only communicates with the third party, it does not get access to the template load profiles, but only the best-matching tariff index (u^*, l^*) from which the associated template load profile cannot be deduced. Furthermore, the rate limiter together with daily shuffling of the indices l prevents the smart meter from querying an arbitrary

Table 1 Comparison of potential privacy breaches and their feasibility for semi-honest adversaries: none of the attacks are feasible in our protocol without collusions

	Smart meter		Third party		Utility	
	Interested	Feasible	Interested	Feasible	Interested	Feasible
(a) Load profile forecast revealed			✓	–[a]	✓	–[a]
(b) Template load profiles revealed	✓	–[a]	✓	–[a]	✓[b]	–[a]
(c) Best matching tariff revealed			✓	–[a]	✓	–[a]

Int. stands for interested, *feas.* for feasible. Whenever attacks require collusions to be feasible, they are denoted by –[a]. A [b] denotes interest in revealing data from other entities of the same type, but not oneself, e.g., other utilities

Table 2 Comparison of potential privacy breaches and their feasibility for malicious adversaries: some of the attacks are feasible in our protocol

	Smart meter		Third party		Utility	
	Interested	Feasible	Interested	Feasible	Interested	Feasible
(a) Load profile forecast revealed			✓	–[a]	✓	–[a]
(b) Template load profiles revealed	✓	–[a]	✓	–[a]	✓[c]	–[a]
(c) Best matching tariff revealed			✓	–[c,b]	✓	–[c,b]

Int. stands for interested, *feas.* for feasible. Whenever attacks require collusions to be feasible, they are denoted by –[a]. If values are modified maliciously, denoted by –[b], information may be revealed. A [c] denotes interest in revealing data from other entities of the same type, but not oneself, e.g., other utilities

number of times with distinct load profile forecasts calculated using different forecasting algorithms.

A *semi-honest third party* wants to learn (i) the load profile forecast of the smart meter, (ii) any template load profile from any of the utilities, and (iii) the tariff the smart meter actually receives at the end of the protocol.

The semi-honest third party does not have access to the unembedded load profile forecast and template load profiles, since the embedding parameters $(\mathbf{A}, \mathbf{W}, \Delta)$ are kept secret among the smart meters and the utilities. The inability to deduce the original data relies on information-theoretic privacy results of [30]. The embedding map can be seen as minimizing the information needed to calculate nearest neighbors keeping only the distance information in a ball around the embedded profile. Theorem 3.1 in [30] states that given two profiles with distance d, the mutual information between the corresponding two embedded signals exponentially decays with $\left(\frac{d}{\Delta}\right)^2$. Therefore, the radius of the ball and thus privacy depends on the size of the security parameter Δ.

Since this paper focuses on the demonstration of the usefulness of the approach, the accuracy of the approach is studied in Section 5.2 using a rather large value of Δ. How Δ should be set to reach a good tradeoff between privacy and utility in practice is a topic for future research.

The best-matching tariff T_{u^*,l^*} is never revealed to the third party, since it does not take part in the *oblivious transfer* phase at all. Note that the third party may collect statistics on the matching results (i.e., the indices (u^*, l^*)) of all smart meters. However, if the utilities shuffle their template load profiles' indices frequently enough, the linkage between the matching results gets lost.

A *semi-honest utility* wants to learn (i) the load profile forecast of the smart meter, (ii) the tariff the smart meter actually receives at the end of the protocol, and (iii) the template load profile from any of the other utilities. Since the utility only communicates with the third party, it does not get access either to the load profile forecast or to template load profiles of other utilities. The tariff the smart meter receives at the end of the protocol is not revealed to the utility due to the properties of the oblivious transfer.

4.1.4 Collusion of semi-honest adversaries

In any of the collusion attacks including the trusted third party, it is assumed that the other colluding party does not share the embedding parameters because otherwise its own privacy could more easily be violated.

U-U: Similar to the single semi-honest utility case, colluding utilities also get no information about smart meters' or non-colluding utilities' load profiles because they get no information except in the *oblivious transfer* phase.

SM-SM: Semi-honest smart meters may collude in order to get information about the utilities' template load profiles by combining their load profile forecasts $\mathbf{F}$ and matching results (u^*, l^*). One possible attack is the attempt to construct a Voronoi diagram defined by the template load profiles as centers. While the borders of the Voronoi diagram do not change due to shuffling, only the labels of the partitions change from one shuffling period to another. However, it is not clear (i) how accurately the Voronoi diagram could be estimated in a high-dimensional space; (ii) how the estimation depends on the security parameter Δ; (iii) given the Voronoi-diagram, how well the centers (corresponding to the template load profiles) could be estimated; and (iv) how many smart meters would need to collude in order to get meaningful results.

These questions are left as open research questions for future research. It should be noted, however, that, due to the small information gain for the smart meters and the limited usefulness, their interest in revealing template load profiles can be considered small.

TP-U$_{u^*}$: A semi-honest third party and the best-matching utility can determine the tariff that any of the smart meters receives at the end of the protocol by simply combining (u^*, l^*) (known to **TP**) and T (known to **U**$_{u^*}$), revealing T_{u^*,l^*}.

TP-U/SM: A semi-honest third party may collude with either smart meters or utilities in order to determine load profiles of non-colluding parties. Here, we describe the collusion between the third party and smart meters—the other case can be treated analogously.

TP can provide the colluding smart meters with embedded template load profiles of utilities and computed distances to each embedded load profile forecast of the colluding smart meters. Based on this information, the template load profiles, each consisting of k real values, may be obtained by the smart meters as follows.

Since the smart meters know the embedding parameters, they also know the relation between the embedding distance and the Euclidean distance of the original data. When the smart meters have at least $p \geq k$ different load profile forecasts $F_i \in \mathbb{R}^k$ within the privacy ball of a given template load profile $L \in \mathbb{R}^k$, i.e., in the linear part of Fig. 5, they can simply read out the p original distances $d_i = d(L, F_i)$ of their known load profile forecasts to the given template load profile. The desired template load profile $L \in \mathbb{R}^k$ can then be determined from the distances d_i to the $p \geq k$ different points $F_i \in \mathbb{R}^k$ by minimization of the norm of the residual vector $R \in \mathbb{R}^p$, where $R_i = ||L - F_i||^2 - d_i^2$.

In the same way, a third party colluding with utilities could determine the load profile forecast of the smart meter. The smaller the security parameter Δ, the higher this information-theoretic privacy will be because it will be more difficult to find $p \geq k$ profiles in a smaller ball around the load profile forecast.

The necessity of the quantization operation in the embedding calculation can be demonstrated for this kind of collusion. If quantization is omitted, even the distance of the load profile forecast to a *single* template load profile leaking from **TP** to **U** may be enough information for **U** to reconstruct the load profile forecast. Without quantization, the embedding calculation is a linear transformation of the input profile. Especially for a high embedding dimension $m > k$, this then forms an overdetermined system of m equations that enables **U** to calculate the k values of the load profile forecast.

Therefore, it would be safer to choose a low embedding dimension $m < k$. However, this paper later shows that a high embedding dimension $m >> k$ is desired for reasonable accuracy (Fig. 7). Thus, without quantization, this constitutes a bad tradeoff between privacy and accuracy which highlights the importance of quantization in Eq. 1.

In summary, collusions involving the third party lead to a privacy breach. However, collusions like this are unlikely to happen, given regulatory restrictions.

4.1.5 Single malicious adversary

A malicious adversary may make up data or violate the protocol, e.g., replay and change messages, avoid or add communication steps. However, due to state-of-the-art encryption and authentication, as described at the beginning of this section, the interception of other connections is infeasible, making man-in-the-middle attacks difficult.

A *malicious smart meter* that wants to learn any template load profile from any of the utilities cannot intercept the communication of any $\tilde{\mathbf{L}}_{u,l}$ due to encryption as described above. In contrast to the semi-honest case, however, it may try to send many artificial load profile forecasts.

On the one hand, it may re-send any load profile forecast to determine the new index of the matching template load profile. However, the rate limiter in the *matching* phase prevents the smart meter from exploiting the knowledge about index shuffling because it does not allow to send another load profile forecast before the next index shuffling (e.g., on the next day).

On the other hand, the smart meter may send a variation of a previous load profile to find the border between the previous best-matching tariff and the new one. However, due to index shuffling, it will not know whether a changed index was indeed due to a different match or due to the index shuffling.

Note that the information about the index shuffling could be derived by associating the index with the tariff obtained in the *oblivious transfer* phase. Therefore, the second rate limiter in the *oblivious transfer* phase is used to prevent the smart meter from querying tariffs other than the best-matching one, including those of other utilities than u^*. The limit for this additional rate limiter in the *oblivious transfer* phase should be chosen to be equal to or even stricter than the limit for the rate limit in the *matching* phase.

In summary, the combination of the rate limiter in the *matching* phase, the rate limiter in the *oblivious transfer* phase and index shuffling prevents a single malicious smart meter interested in any template load profile from creating Voronoi diagrams defined by the template load profiles as centers.

A *malicious third party* wants to learn (i) the load profile forecast of any of the smart meters; (ii) any template load profile from any of the utilities; and (iii) the tariff that any of the smart meters actually receives at the end of the protocol. For the third party, it is infeasible to learn any load profiles or the tariff because the extension of the attacker's capabilities still does not include reversing the embedding without knowledge of the embedding parameters as in the semi-honest case. However, a malicious third party may modify the matching result (u^*, l^*). This way, the third party can force a tariff onto the smart meter. However, due to index shuffling, the third party will not know which tariff it forces onto the smart meter.

A *malicious utility* wants to learn (i) the load profile forecast of the smart meter, (ii) the tariff the smart meter actually receives at the end of the protocol, and (iii) the template load profile from any of the other utilities.

The utility cannot intercept the submission of template load profiles of any other utility (due to encryption as described above). In addition, the utility may, theoretically, provide one tariff for every possible template load profile. This would allow the utility to associate perfect matches and thereby reveal load profile forecasts. However, in a practical setting like in Section 5, with 96 values of 16 bits size, this would require the utility to provide a total of $2^{16^{96}} > 10^{10^{115}}$ different template load profiles, which is clearly infeasible.

A malicious utility may attempt to provide one single template load profile per day and change it on a daily basis. If a load profile forecast matches, the utility knows which tariff the smart meter receives in the *oblivious transfer* phase. Additionally, the matching template load profile can be seen as an approximation to the load profile forecast. In order to prevent this, the third party can enforce $|\mathcal{L}_u| > 1$ for all utilities u.

4.1.6 Collusion of malicious adversaries

Whenever a collusion of semi-honest adversaries allows for privacy breaches, the same is, of course, true for colluding malicious adversaries. Therefore, in the following, only the cases where a collusion of semi-honest adversaries does not lead to a privacy breach are discussed.

U-**U**: Multiple malicious utilities may choose their template load profiles in a coordinated way in order to infer load profile forecasts. This case is analogous to the case of a single malicious utility described above.

A *malicious smart meter* may collude with the third party in order to learn any template load profile from any of the utilities.

SM-**SM**: Multiple malicious smart meters may collude in order to learn the template load profile from the utilities. To do so, they make up a set of disjoint load profile forecasts. Since each smart meter can send its forecast separately, all smart meters combined bypass the rate limiter and may derive more information about the template load profiles.

This is a breach of privacy. However, this prevents every involved smart meter from obtaining a matching tariff for itself, likely keeping all of them at more unsuitable tariffs with higher costs for the smart meters and increasing the utility's revenue.

Finally, a *malicious third party* may collude with a single utility u' and return a matching result (u', l^*) from that utility, even if it is just slightly worse than a matching result (u'', l'^*) from another utility.

In summary, except for the case when two utilities collude, privacy is breached for colluding malicious adversaries.

4.2 Complexity analysis

In this section, we analyze both the time and space complexity of our proposed approach. For the sake of readability, in this section only, the number of utilities, formerly denoted as $|\mathcal{U}|$, will be denoted as M. Similarly, the number of tariffs per utility $\mathbf{U}_u$, formerly denoted as $|\mathcal{L}_u|$, will be denoted as N. For simplicity, N is considered to be equal for all utilities, yielding a total of MN tariffs for M utilities.

Note that by space complexity, we here refer to the number and size of messages with respect to M and N, split into inbound and outbound traffic. We do *not* refer to any memory and storage requirements, be they temporary or permanent. Similarly, the time complexity is specified with respect to M and N.

Tables 3, 4, and 5 summarize the time and space complexity per entity and phase, respectively. In the following sections, a detailed complexity analysis is given per phase with references to the two aforementioned tables.

4.2.1 Initialization

In the *initialization* phase, each utility calculates one embedding per template load profile. Since there is one template load profile per tariff, each utility calculates N embeddings, yielding a time complexity of $O(N)$. Since all embeddings have to be sent to the third party, the outbound space overhead per utility is also $O(N)$, whereas the inbound space complexity for the third party is $O(MN)$, since it receives all tariffs from all utilities.

However, neither the third party nor the smart meter perform any calculations or generate outbound traffic. Thus, no time and outbound space complexity is given for them in Tables 3 and 4, denoted by hyphens. Similarly, neither the smart meter nor the any of the utilities receive inbound messages in this phase.

4.2.2 Matching

Likewise, no utility is involved in the subsequent *matching* phase which is limited to interactions between the smart meter and the third party. The former calculates the embedding of its load profile forecast, which is independent of both, the number of utilities and the number of tariffs, yielding a time complexity of $O(1)$ and an equal outbound space complexity for the request, yielding an identical inbound space complexity for the third party.

The third party needs to calculate the differences between the load profile forecast and each template load

Table 3 Time complexity overview: list of the combined time complexity of our proposed approach, detailed per entity and phase

	Smart meter	Third party	Utility
Initialization	-	-	$O(N)$
Matching	$O(1)$	$O(MN)$	-
Oblivious transfer	$O(1)$	-	$O(N)$

Hyphens denote parties which do not perform any operations in the respective phases

Table 4 Outbound space complexity overview: list of the combined space complexity for outbound messages of our proposed approach, detailed per entity and phase

	Smart meter	Third party	Utility
Initialization	-	-	$O(N)$
Matching	$O(1)$	$O(1)$	-
Oblivious transfer	$O(1)$	-	$O(N)$

Hyphens denote parties which do not perform any operations in the respective phases

profile of each utility. Since there are M utilities with N template load profiles each, $O(MN)$ calculations have to be performed by the third party. However, only one result—the best match—has to be sent back to the smart meter, yielding an outbound space complexity of $O(1)$ and an identical inbound space complexity for the smart meter.

Note that the rate limiting function is not considered in the above complexity analysis. We consider this to be an implementation-specific issue which requires no additional time or outbound space complexity. It does, however, require additional memory (space complexity) to store at least one time stamp per smart meter which contacted the third party at least once. This is considered to be outside the scope of this paper.

4.2.3 Oblivious transfer

In the *oblivious transfer* phase, the best-matching utility sends N keys to the smart meter, yielding an initial outbound space complexity of $O(N)$ and an identical inbound space complexity for the smart meter. After receiving the encrypted nonce of the smart meter, the utility decrypts and re-encrypts it with N different keys, corresponding to a total time complexity of $O(N)$. Sending these N re-encrypted messages yields an outbound space complexity of $O(N)$ and an identical inbound space complexity for the smart meter. Together with the initial space complexity of $O(N)$, the total outbound space complexity for the utility and the inbound space for the smart meter are both $O(N)$.

In contrast to the third party's time complexity for calculating the different encryptions, the smart meter only has to encrypt one message—the nonce. Since this is independent of both,the number of utilities and the number of tariffs, the time complexity for this step is $O(1)$. Similarly, the outbound space complexity when sending this encrypted message is $O(1)$, as is the inbound space complexity for the utility. The final step, where one of the N encrypted tariffs received from the utility is decrypted, also yields a time complexity of $O(1)$. In total, this does not change the overall time complexity of $O(1)$ for the smart meter in this phase.

As in Section 4.2.2, the rate limiting function is not considered in the above complexity considerations. However, it does neither affect the time nor the outbound nor the inbound space complexity denoted in Tables 3, 4, and 5.

4.2.4 Summary

In summary, it is clear that our proposed approach is very efficient in terms of computational complexity. In particular, all operations on the smart meter—a device with very limited computational capabilities—can be performed in constant time with respect to the number of utilities and tariffs. In contrast, the matching complexity is outsourced to the third party and each utility which can determine the computational complexity by the number of tariffs it offers.

There are other approaches with similar total computational complexity considering only the number of utilities and tariffs as variables. However, for other variables, e.g., ciphertext size, the complexity of related work is inferior to ours as shown in the following section.

4.3 Comparison to related work

The core purpose of the presented approach is to find similarities, i.e., nearest neighbors, of time series by Euclidean distance computation while preserving the privacy of all actors. In [6, 14, 31], approaches based on additively homomorphic encryption, e.g., the Paillier cryptosystem [32], are discussed which also allow calculating Euclidean distances. Thus, they can be used as an alternative to nearest-neighbor embeddings as proposed in this paper and are therefore compared to the latter. Figure 2 shows a simplified version of both our proposed scheme and approaches using homomorphic encryption.

The smart meter communicates with each utility separately sending an additively homomorphic encrypted load profile forecast. Each utility responds with a partial result for each template load profile as shown below. From this, the smart meter can calculate the Euclidean distance in order to retrieve the index (u^*, l^*) of the best-matching template load profile. Finally, the smart meter and the utility u^* run the oblivious transfer protocol identically to our approach in order to retrieve the tariff T_{u^*,l^*}.

This homomorphic protocol comes without the need of a third party. By contrast, fetching the template load profiles as well as the distance computation itself is performed

Table 5 Inbound space complexity overview: list of the combined space complexity for inbound messages of our proposed approach, detailed per entity and phase

	Smart meter	Third party	Utility
Initialization	-	$O(MN)$	-
Matching	$O(1)$	$O(1)$	-
Oblivious transfer	$O(N)$	-	$O(1)$

Hyphens denote parties which do not perform any operations in the respective phases

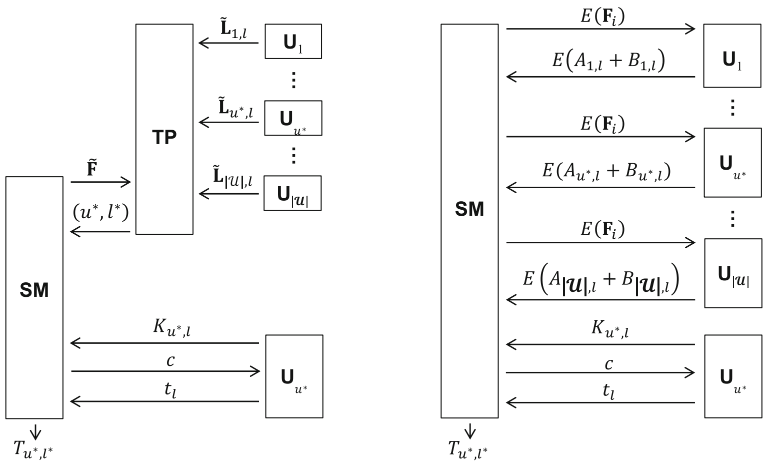

Fig. 2 Comparison of our proposed scheme (*left*) to approaches using homomorphic encryption (*right*). The latter comes without a third party needed for comparison, which is instead done by the smart meter

by the smart meter. All encryptions are performed with the public key, decryptions can only be performed by the smart meter, which is the only party owning the private key.

The Euclidean distance between the forecast load profile and each of the template load profiles of all utilities must be computed by using an additively homomorphic cryptosystem [6] in an identical manner, i.e., with the same modulus n of Paillier's cryptosystem. Therefore, for the sake of readability, $\mathbf{L}_i$ is written instead of $\mathbf{L}_{u,l,i}$ and A and B are written instead of $A_{u,l}$ and $B_{u,l}$. The goal of the protocol is to privately compute

$$||\mathbf{F}-\mathbf{L}||_2 = \sum_i \mathbf{L}_i^2 - \sum_i 2\mathbf{F}_i\mathbf{L}_i + \sum_i \mathbf{F}_i^2 =: A+B+C. \quad (5)$$

First, the smart meter submits its encrypted load profile forecast values $E(\mathbf{F}_i)$ directly to each utility. As a single load profile forecast value is much smaller than the modulus, data packing is used for better exploiting the input domain. Therefore, not a single value is encrypted, but all k values of the load profile forecast are packed, encrypted, and sent as one message. Data packing is achieved by shifting values to a certain bit range, such that for all operations, the value remains within that range [13], e.g., for k values and a range of b bits $v = v_1|v_2|\ldots|v_k = \sum_{i=1}^{k} v_i 2^{b(i-1)}$.

Using this encrypted load profile forecast and exploiting the homomorphic properties $E(x+y) = E(x)E(y)$ and $E(x)^c = E(cx)$, the utility can compute and send back the partial result

$$E(A+B) = E\left(\sum_i \mathbf{L}_i^2\right) \prod_i E(\mathbf{F}_i)^{-2\mathbf{L}_i} \quad (6)$$

to the smart meter. Exploiting the homomorphic property again, term C can be added and the smart meter gets the Euclidean distance by

$$||\mathbf{F}-\mathbf{L}||_2 = D\left(E\left(\sum_i \mathbf{F}_i^2\right) E(A+B)\right) \quad (7)$$

When following the above protocol, neither the utility knows the smart meter's load profile forecast, nor does the smart meter know the utility's template load profile. In addition, the inner product $\mathbf{F} \cdot \mathbf{L}$ is never revealed to any of them.

Now, the communication need is calculated and compared with this paper's solution. In the first step of the protocol, the smart meter needs to send k packed and homomorphically encrypted load values F_i to each of the $|U|$ utilities. By encrypting a single value with the Paillier cryptosystem, a plaintext $p \in \mathbb{Z}_n^*$ results in a ciphertext $E(p) \in \mathbb{Z}_{n^2}^*$ leading to an expansion of the bit size by a factor of 2.

The second message is the encryption of

$$A+B = \sum_i \mathbf{L}_i^2 - \sum_i 2\mathbf{F}_i\mathbf{L}_i \in \left[-\sum_i F_i^2, \sum_i L_i^2\right] \quad (8)$$

where the lower limit follows from the fact that

$$||\mathbf{F}-\mathbf{L}||_2 = A+B+C \geq 0 \Rightarrow A+B \geq -C. \quad (9)$$

Since squaring of a number leads to an expansion of 2, both the lower and upper limits need a maximum of $k \cdot 2s$ bits, where s is the bit of a single load value F_i. Therefore, $A + B$ needs $2 \cdot 2ks = 4ks$ bits. Because of subsequent data expansion by a factor of 2 due to encryption, finally, the ciphertext fits into $8ks$ bits, which would require a modulus of $8ks$ bits size for the Paillier cryptosystem with modulus n. If the modulus is smaller (see below for a practical example), the message can be split into multiple messages, the number of which is $\left\lceil \frac{8ks}{2n} \right\rceil = \left\lceil \frac{4ks}{n} \right\rceil$.

As a practical example, consider $k = 96$ values of a day profile arising from a 15-min measurement interval, a bit size of $s = 16$ and $|\mathcal{U}| = 5$ utilities, each having $|\mathcal{L}_u| = 20$ template load profiles. Therefore, a template load profile (as well as load profile forecast) is of size $4ks = 4 \cdot 96 \cdot 16 = 6144$ bits. According to latest NIST recommendations [33], a Paillier modulus of $n = 2048$ bits (expanding to 4096 bits) is chosen, which requires three messages of that size, since $\frac{8ks}{2n} = \frac{4ks}{n} = 3$.

For the homomorphic encryption approach, the smart meter sends its homomorphically encrypted load profile forecast to each of the $|\mathcal{U}|$ utilities. As described above, sending one load profile forecast requires three messages of 4096 bits size after encryption. Sending all load profile forecasts in this step therefore requires $3 \cdot 4096|\mathcal{U}|$ bits.

In addition, the *oblivious transfer* step at the end requires one message of 256 bits size (when using AES-256 [29]). Thus, the total sending need for a single smart meter is 7.53 KiB (rounded to two decimal places).

Each utility responds to the requesting smart meter with a partial result (see Eq. 6) for each of the $|\mathcal{L}_u|$ template load profiles, as described above. From this, the smart meter calculates the Euclidean distance. One template load profile requires $3 \cdot 4096$ bits, as described above. One utility therefore sends $3 \cdot 4096|\mathcal{L}_u|$ bits. Thus, $|\mathcal{U}|$ utilities send $3 \cdot 4096|\mathcal{L}_u||\mathcal{U}|$ bits, which are received by the smart meter.

In addition, the *oblivious transfer* step requires sending $|\mathcal{L}_u|$ messages of 256 bits size. The smart meter thus receives a total of $3 \cdot 4096|\mathcal{L}_u||\mathcal{U}| + 256|\mathcal{L}_u|$ bits $= 150.63$ KiB (rounded to two decimal places). The resulting communication need is shown in Fig. 3 (solid and dashed black lines).

With the embedding approach, the smart meter needs to send its load profile forecast consisting of m bits of data to the third party. All $|\mathcal{U}|$ utilities need to send $|\mathcal{L}_u|$ template load profiles of m bits each, totaling $|\mathcal{U}||\mathcal{L}_u|m$ bits. For the communication of the best-matching indices (u^*, l^*), $s = 16$ bits $= 2$ B should suffice.

Figure 3 shows the communication need of our proposed embedding approach (solid and dashed red lines, as well as dash-dotted blue line) for various embedding dimensions m. $m = 8192$ (dashed-dotted gray line) is a reasonable choice (cf. Fig. 7), resulting in an overhead of 100 KiB.

The sending and receiving need of smart meters is orders of magnitudes smaller for our proposed embedding protocol than for the protocol with homomorphic encryption used for comparison. This is important, since in practical scenarios, the bandwidth connecting smart meters with other parties is likely to be low [34], especially when using power-line communication (PLC). While the total communication amount for the embedding method can even be higher than for the homomorphic method, most of it is needed for the communication from utilities to the third party where a much better connection is likely.

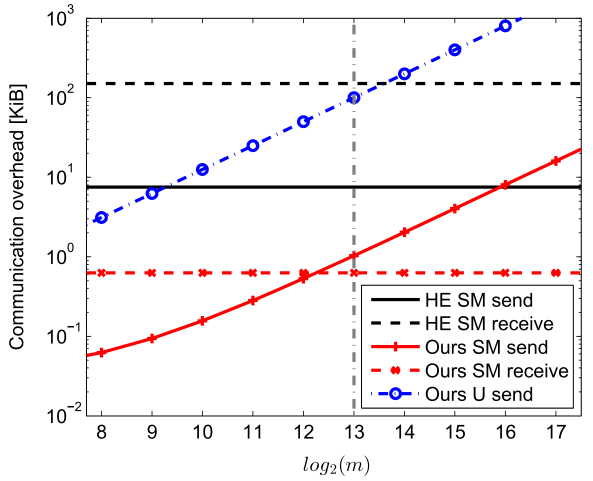

Fig. 3 Comparison of the communication need of our protocol and the one used for comparison with variable embedding dimension *m*: The communication need for the smart meter is considerably lower with our proposed embedding protocol (denoted *ours*) than with the protocol using homomorphic encryption (denoted *HE*). Utilities only need to send data to the **TP** using our proposed embedding protocol (*blue*, *dashed-dotted*)

The approach based on homomorphic encryption does not require a third party to perform the distance calculation, since either of the participants is involved exactly once in exchanging messages and can limit the rate of requests in order to prevent chosen-plaintext attacks to learn about load profiles. However, the smart meter—which is usually a device with only limited computational capabilities—has to perform the distance computation for every template load profile from every utility, which results in a total of $\sum_u |\mathcal{L}_u|$ distance computations. This is likely to be impractical for a device with low computational capabilities like a smart meter and thus another disadvantage compared to our approach.

However, the homomorphic approach calculates all distances exactly. This is not the case in the profile matching approach presented in this paper. In summary, the smaller overhead in data expansion comes at the cost of only near-perfect matching. However, the accuracy depends on *m* which can be chosen appropriately. Thus, an evaluation of this parameter is conducted in the following section.

5 Evaluation

In this section, we evaluate privacy-preserving load profile matching. We briefly describe the test data and assess the matching performance of our approach using that data. For the evaluation, we prototypically implemented the complete proposed matching protocol as described in Section 3 in *Java* and *Matlab*.

5.1 Data

The data set we use for our evaluation consists of load profiles of 40 residential households from a local energy provider. The evaluated load profiles cover a time span of about 1 year with 96 measurements per day, i.e., 15-min data granularity. The measured values are in Watts with 16 bits of precision. Thus, one day of one household can be represented by $96 \cdot 16 = 4096$ bits.

All load profiles are normalized by dividing every one of the 96 load values by the average load per day. We use the captured data as load profile forecasts $\mathbf{F}$ as described in Section 3. Note that forecasting can be done with historic data (e.g., [35, 36]) since usage patterns are unlikely to change very much over longer periods of time.

For matching, we consider the following template load profiles $\mathbf{L}_{u,l}$ depicted in Fig. 4:

- *Flat* (gray circles): The load does not change over time.
- *Standard* (blue plusses): There is a lower load during night hours.
- *Night owl* (dashed red line): There is a higher load during night hours and a lower load during the day.
- *H0 workday* (dash-dotted green line): The standardized [37] average load of a household during workdays with two small load peaks at noon and one larger peak in the evening. The type of households and the broader geographic region for the "H0" profile in the standard are the same as for our test data described above.
- *H0 Sunday* (solid black line): The standardized average load of a household on Sundays with one load peak at noon and one in the evening.

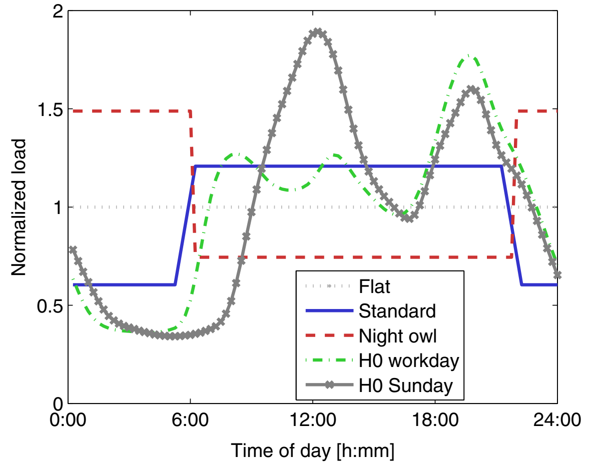

Fig. 4 Template load profiles: usage-specific load profiles are used as $\mathbf{L}_{u,l}$ for the *matching* phase

5.2 Matching performance

All load profile forecasts of all 40 households from the data set are matched with one of the five template profiles in Fig. 4. For each day's forecast, the template load profile with the lowest Euclidean distance is found. For comparison, the same operation is performed with the embedded load profiles and the normalized Hamming distance. The matching performance is then assessed by the matching accuracy (Eq. 4) which is estimated by the percentage of coinciding matches. The slightly different distance calculations (see Eq. 3) lead to a decrease in accuracy. Ideally, parameters are chosen such that the correlation between the distances in the original space $\mathbb{R}^k$ and the distances in the embedded space $\{0, 1\}^m$ is high. While the matching accuracy for a single smart meter is not affected by the number of participating smart meters, two parameters influence the matching accuracy in two different ways.

5.2.1 Effect of Δ

If the first security parameter, Δ, is chosen too large (at least as big as the maximum original distance), it only has a constant effect on all computed distances in the embedded space since the modulo operation in Eq. 1 does not change the final result. As already discussed in Section 4, embedding will not increase privacy, and Eq. 3 holds everywhere.

However, the security parameter has an essential influence on the embedding distance when it is small enough to be within the range of the original distance. This influence is described in the left panel of Fig. 5, where distances in the embedding space are computed with varying security parameter Δ. Two regions can then be distinguished: region R^- of original distances that are (roughly) smaller than Δ and region R^+ of original distances that are (roughly) larger than Δ. In region R^-, the embedding distance is linearly depending on the original distance and Eq. 3 holds. Due to this linear dependence, the *utility-region* R^- offers high utility, but no privacy.

In contrast, in region R^+, the embedding distance is near 0.5 and not depending on the original distance. Thus, the *privacy-region* R^+ exhibits privacy, but no utility.

As a consequence, if all template profiles are far enough away from the forecast such that the original distances are all in region R^+, the best match in the embedding distance will lead to no information on the best match in the original space. This behavior is confirmed by Fig. 6 which shows that a very small Δ has a devastating effect on the accuracy yielding values that are as bad as pure guessing. For $\Delta \geq 14$, which is near the maximum original distance values, distances are in the utility region R^- leading to a high accuracy above 90 %. Accuracy does not increase significantly with further increases in Δ (see also [11]).

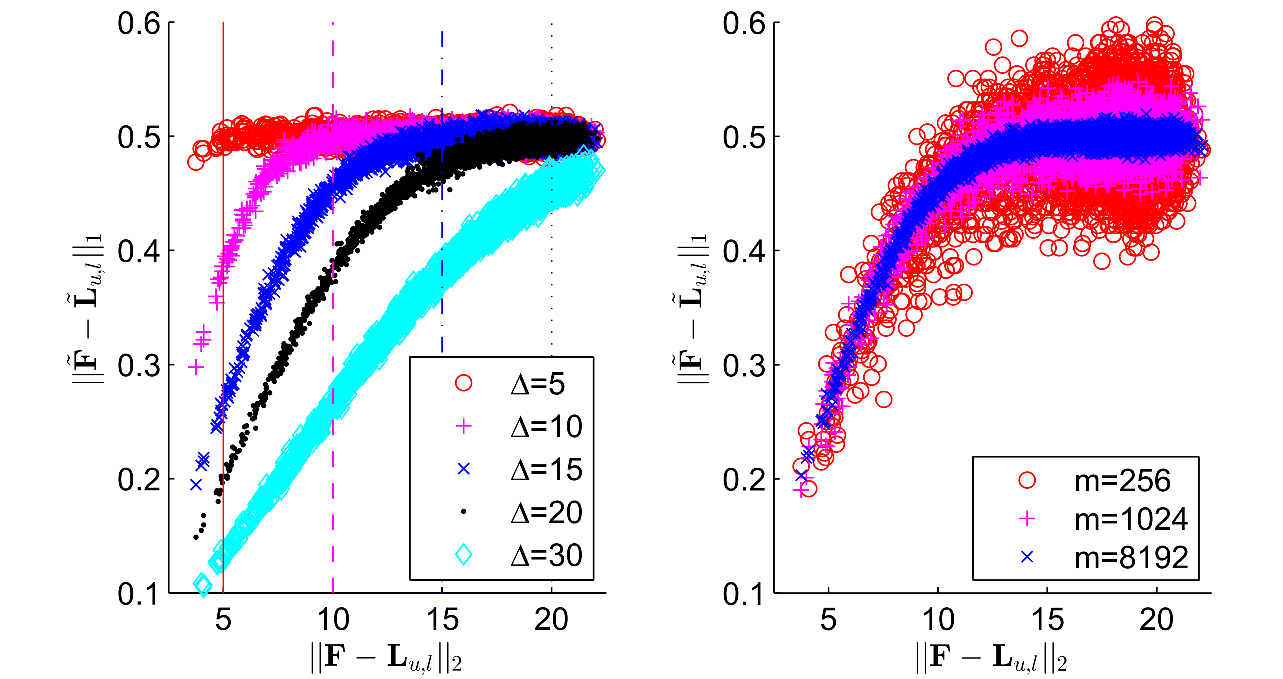

Fig. 5 Comparison of the Euclidean distance (*x* axis) of the original data and the normalized Hamming distance of the embedded data (*y* axis). *Left panel* The security parameter Δ influences the position of the "knee" ($m = 8192$). *Right panel* An increase in the embedding dimension m decreases scattering and thus the correlation between the original and the embedded distance in the linear part of the curve ($\Delta = 15$)

5.2.2 Effect of *m*

For our application, we are interested in finding template load profiles which have a small distance to the forecast. Therefore, the distance of the load profile to the best matching template load profile are supposed to be in the utility region R^{-}. In this part, the correlation between the original and the embedding distance depends on the second parameter of our matching protocol, the dimension m of the embedded space: the higher m is, the higher the correlation. This is an effect of the smaller amount of scatter in the right panel of Fig. 5 ($\Delta = 30$ was chosen high in order not to mix the effect of the two parameters).

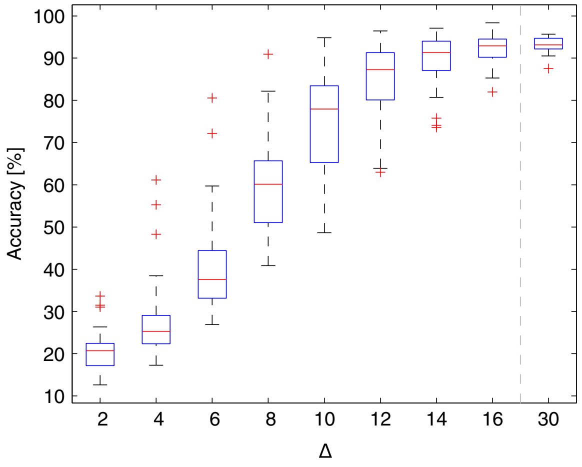

Fig. 6 Accuracy of the matching process with embedding dimension $m = 8192$ and variable Δ: the percentage of coinciding matches for distance calculations with and without embedding increases with the security parameter Δ. The *dotted gray line* symbolizes the gap between $\Delta = 16$ and $\Delta = 30$

As a consequence, the distance preservation property of the embedding transformation will have the effect that the best match in the original space $\mathbb{R}^k$ should also be the best match in the embedding space $\{0, 1\}^m$. It is experimentally confirmed for all 40 households that the accuracy increases with the embedding dimension m (Fig. 7). For small values of m, the median accuracy is impractically low (below 65 %), while it is 98 % for very large values of m. However, such values are not desired since they result in a large ciphertext size as already discussed in Section 4.

An embedding dimension of $m = 8192$ provides a reasonable tradeoff between ciphertext size and accuracy at 93.5 %. In practice, 93.5 % are sufficient since the selection of the (most likely) second-best-matching tariff only results in a minor deviation as shown below. Note that, as described above, the accuracy can be increased if necessary at the cost of data expansion. This is can be also be used to compensate for a small number of tariffs in borderline cases.

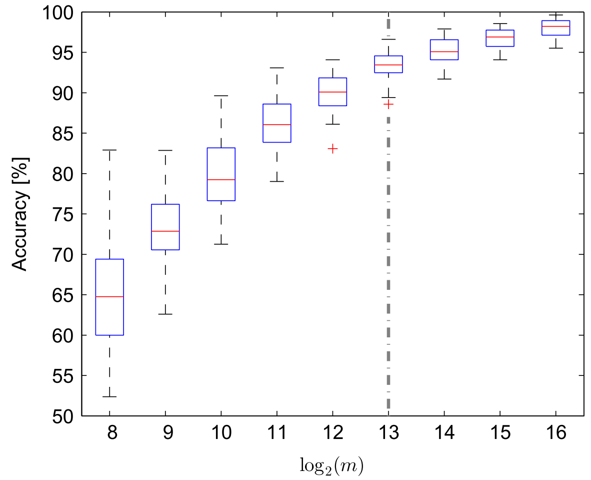

Fig. 7 Accuracy of the matching process with security parameter $\Delta = 30$ and variable embedding dimension *m*: The percentage of coinciding best matches based on distance calculations with and without embedding increases with *m*

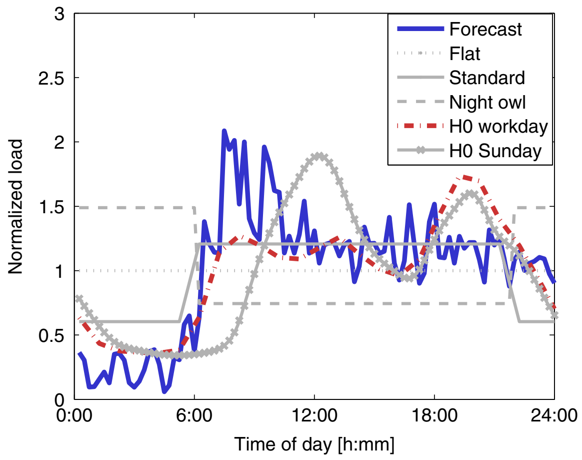

Fig. 8 Comparison of a load profile forecast and all template load profiles: the H0 workday template load profile (*dash-dotted red line*) matches the forecast (*solid blue line*) best. The inferior matches are depicted in *gray*

5.2.3 *Load profile examples*

We conclude this section by matching two example load profiles for the purpose of illustration of tariff matching. While the selection of a suitable parameter Δ is a topic for future research, for purpose of illustration, Δ was selected high as 30. Two cases are shown. The first shows a load profile forecast where matching in the embedded space returns the same template load profile as in the original space. The second depicts a forecast where the best match in the original space $\mathbb{R}^k$ does not coincide with the best match in the embedded space $\{0, 1\}^m$.

The case in Fig. 8 shows a load profile of household 3 and our template load profiles. Using this load profile (solid blue line) as load profile forecast **F**, we see that the H0 workday template load profile (dash-dotted red line) is the best match of all five template load profiles, i.e., the Euclidean distance is smallest.

In contrast, Fig. 9 shows the load profile of household 1 which is an example for a mismatch. It is the worst case from within all days of that household. The best match for the load profile forecast of this worst-case example in the original space $\mathbb{R}^k$ is the flat template load profile (dashed green line). However, the use of embedding slightly perturbs the result by yielding a smaller distance to the night owl template load profile (dash-dotted red line). The distances in the embedded space are 3.38 for the flat profile compared to 3.50 for the night owl profile. Further distances are 5.33 for the standard profile, 6.64 for the H0 workday profile, and 6.41 for the H0 Sunday profile which shows that the best match found in the embedded space $\{0, 1\}^m$ still yields a good result if compared to the other template load profiles.

6 Conclusions

We described a load profile matching protocol which enables tariff decisions in smart grids. We showed that the protocol design and the use of embeddings and oblivious transfer make our approach privacy-preserving for both the smart meter and the participating utilities when all parties are semi-honest and do not collude. In comparison to a protocol based on the additive homomorphic Paillier cryptosystem, the proposed protocol reduces the communication need for smart meters by several orders of magnitude, albeit at the cost of non-perfect matching. We achieve a matching accuracy of

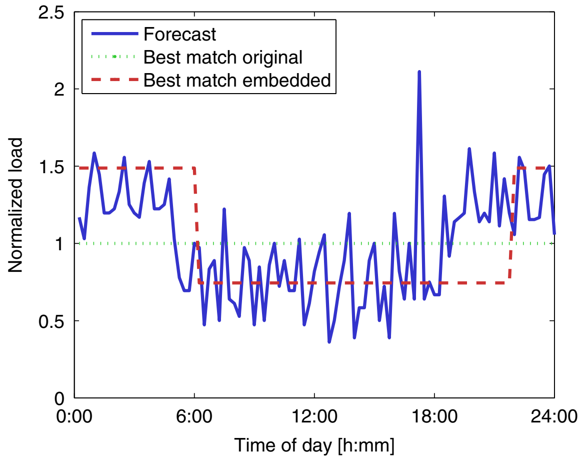

Fig. 9 Comparison of a worst-case load profile forecast: The load profile forecast (*blue line*) best matches the flat template load profile (*dashed green line*) in the original space, but the night owl profile (*dash-dotted red line*) in the embedded space

93.5 % with negligible outliers, which may be enough to make it a viable, more lightweight alternative to homomorphic encryption for load profile matching in smart grids.

Future work will focus on a broader set of use cases and sample data. Further analysis and evaluation of a prototypical implementation of the entire protocol, including all three phases, is planned. For this implementation, time measurements for comparing homomorphic encryption and our embedding approach are to be conducted.

Acknowledgements
The financial support by the Austrian Federal Ministry of Science, Research and Economy and the Austrian National Foundation for Research, Technology and Development is gratefully acknowledged. Funding by the Federal State of Salzburg is gratefully acknowledged. The authors would like to thank their partner Salzburg AG for providing real-world load data.

Authors' contributions
This paper was written by AU (30 %), FK (30 %), GE (30 %), and DE (10 %). The detailed contributions are as follows: The abstract was written by AU (80 %) and DE (20 %). Section 1 was written by DE (90 %) and AU (10 %). Section 2 was written by AU (70 %) and GE (30 %). Section 3 was written by AU (50 %) and FK (50 %). Section 4 was written by AU (40 %), FK (40 %), and GE (20 %). Section 5 was written by GE (70 %) and AU (30 %). Section 6 was written by AU (80 %) and GE (20 %). The protocol-related figures were created by FK (100 %). The evaluation-related figures were created by GE (100 %). A prototypical implementation in Java was created by FK (100 %). The evaluation was performed by GE (100 %). All authors read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]Josef Ressel Center for User-Centric Smart Grid Privacy, Security and Control, Salzburg University of Applied Sciences, Urstein Süd 1, 5412 Puch bei Hallein, Austria. [2]Department of Computer Sciences, University of Salzburg, Jakob-Haringer-Str. 2, 5020 Salzburg, Austria.

Received: 24 February 2016 Accepted: 26 August 2016
Published online: 26 September 2016

References

1. L Karg, K Kleine-Hegermann, M Wedler, C Jahn, E-Energy Abschlussbericht – Ergebnisse und Erkenntnisse aus der Evaluation der sechs Leuchtturmprojekte. Technical report, Bundesministerium für Wirtschaft und Technologie (German Federal Ministry for Economy and Technology). German (2014). http://www.digitale-technologien.de/DT/Redaktion/DE/Downloads/ab-gesamt-begleitforschung.pdf?_blob=publicationFile&v=4. Accessed 16 Aug 2016
2. M Lisovich, D Mulligan, S Wicker, Inferring personal information from demand-response systems. IEEE Secur. Priv. **8**(1), 11–20 (2010)
3. E McKenna, I Richardson, M Thomson, Smart meter data: balancing consumer privacy concerns with legitimate applications. Energy Policy. **41**, 807–814 (2012)
4. P McDaniel, S McLaughlin, Security and privacy challenges in the smart grid. IEEE Secur. Priv. Mag. **7**(3), 75–77 (2009)
5. G Eibl, D Engel, Influence of data granularity on smart meter privacy. IEEE Trans. Smart Grid. **6**(2), 930–939 (2015)
6. SD Rane, P Boufounos, Privacy-preserving nearest neighbor methods: comparing signals without revealing them. IEEE Signal Process. Mag. **30**(2), 18–28 (2013)
7. J Kilian, in *ACM Symposium on Theory of Computing*. Founding cryptography on oblivious transfer (ACM, Chicago, 1988), pp. 20–31
8. S Mukherjee, Z Chen, A Gangopadhyay, A privacy-preserving technique for Euclidean distance-based mining algorithms using Fourier-related transforms. VLDB J. **15**(4), 293–315 (2006)
9. P Ravikumar, WW Cohen, SE Fienberg, in *International Conference on Data Mining (ICDM)*. A secure protocol for computing string distance metrics, (2004), pp. 40–46
10. WK Wong, DW-L Cheung, B Kao, N Mamoulis, in *Proceedings of the 35th SIGMOD International Conference on Management of Data*. Secure kNN computation on encrypted databases categories and subject descriptors (ACM, 2009), pp. 139–152
11. PT Boufounos, S Rane, in *2013 Data Compression Conference (DCC)*. Efficient coding of signal distances using universal quantized embeddings (IEEE, 2013), pp. 251–260
12. JH Cheon, M Kim, K Lauter, Homomorphic computation of edit distance. Lect. Notes Comput. Sci. **8976**, 194–212 (2015)
13. Z Erkin, T Veugen, T Toft, RL Lagendijk, Generating private recommendations efficiently using homomorphic encryption and data packing. IEEE Trans. Inf. Forensics Secur. **7**(3), 1053–1066 (2012)
14. SD Rane, W Sun, A Vetro, in *2009 16th IEEE International Conference on Image Processing (ICIP)*. Secure distortion computation among untrusting parties using homomorphic encryption, (IEEE, 2009), pp. 1485–1488
15. M Barni, T Bianchi, D Catalano, M Di Raimondo, RD Labati, P Failla, D Fiore, R Lazzeretti, V Piuri, A Piva, F Scotti, in *IEEE 4th International Conference on Biometrics: Theory, Applications and Systems, BTAS 2010*. A privacy-compliant fingerprint recognition system based on homomorphic encryption and fingercode templates (IEEE, 2010), pp. 1–7
16. A-R Sadeghi, T Schneider, I Wehrenberg, in *Information, Security and Cryptology ICISC 2009. Lecture Notes in Computer Science*, ed. by D Lee, Hong S. Efficient privacy-preserving face recognition, vol. 5984 (Springer, Berlin, Heidelberg, 2010), pp. 229–244
17. V Kolesnikov, AR Sadeghi, T Schneider, Improved garbled circuit building blocks and applications to auctions and computing minima. Lect. Notes Comput. Sci. **5888**, 1–20 (2009)
18. AC-C Yao, in *27th Annual Symposium on Foundations of Computer Science*. How to generate and exchange secrets (IEEE Computer Society, Washington, DC, 1986), pp. 162–167
19. D Catalano, R Cramer, G DiCrescenzo, I Darmgard, T Takagi, D Pointcheval, *Provable Security for Public Key Schemes*. (Birkhäuser Verlag, Basel, 2005), pp. 133–190
20. P Palensky, D Dietrich, Demand side management: demand response, intelligent energy systems, and Smart loads. IEEE Trans. Ind. Inform. **7**(3), 381–388 (2011)
21. S Caron, G Kesidis, in *2010 First IEEE International Conference on Smart Grid Communications (SmartGridComm)*. Incentive-based energy consumption scheduling algorithms for the smart grid (IEEE, 2010), pp. 391–396
22. S Shao, T Zhang, M Pipattanasomporn, S Rahman, in *2010 IEEE PES Transmission and Distribution Conference and Exposition: Smart Solutions for a Changing World*. Impact of TOU rates on distribution load shapes in a smart grid with PHEV penetration (IEEE, 2010), pp. 1–6
23. S Ramchurn, P Vytelingum, A Rogers, N Jennings, in *The 10th International Conference on Autonomous Agents and Multiagent Systems. AAMAS '11*. Agent-based control for decentralised demand side management in the smart grid, vol. 1 (International Foundation for Autonomous Agents and Multiagent Systems, Taipei, 2011), pp. 5–12. http://eprints.soton.ac.uk/271985/
24. A-H Mohsenian-Rad, VWS Wong, J Jatskevich, R Schober, A Leon-Garcia, Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid. IEEE Trans. Smart Grid. **1**(3), 320–331 (2010)
25. R Gennaro, J Katz, H Krawczyk, T Rabin, in *Public Key Cryptography PKC 2010. Lecture Notes in Computer Science*, ed. by D Pointcheval, PQ Nguyen. Secure network coding over the integers, vol. 6056 (Springer, Berlin, Heidelberg, 2010), pp. 142–160
26. D Fiore, R Gennaro, V Pastro, in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. CCS '14*. Efficiently Verifiable Computation on encrypted data (ACM, Scottsdale, 2014), pp. 844–855
27. W Diffie, M Hellman, New directions in cryptography. IEEE Trans. Inf. Theory. **22**(6), 644–654 (1976)
28. ITU-T. Recommendation ITU-T X.509—Information technology—Open Systems Interconnection—The Directory: Public key and attribute certificate frameworks, (2012)
29. National Institute of Standards and Technology (NIST). Specification for the Advanced Encryption Standard (AES), (2001)

30. P Boufounos, S Rane, in *IEEE International Workshop on Information Forensics and Security*. Secure binary embeddings for privacy preserving nearest neighbors, (2011)
31. R Lagendijk, Z Erkin, M Barni, Encrypted signal processing for privacy protection. IEEE Signal Process. Mag. **30**, 82–105 (2013)
32. P Paillier, in *Proceedings of Eurocrypt '99, Advances in Cryptology. Lecture Notes in Computer Science*, ed. by J.˜Stern. Public-key cryptosystems based on composite degree residuosity classes, vol. 1592 (Springer, Prague, 1999), pp. 223–238
33. National Institute of Standards and Technology (NIST). NIST 800-57: computer security. NIST, (2012)
34. A Unterweger, D Engel, Resumable load data compression in smart grids. IEEE Trans. Smart Grid. **6**(2), 919–929 (2015)
35. Y Chen, PB Luh, C Guan, Y Zhao, LD Michel, MA Coolbeth, PB Friedland, SJ Rourke, Short-term load forecasting: similar day-based wavelet neural networks. IEEE Trans. Power Syst. **25**, 322–330 (2010)
36. C Guan, PB Luh, LD Michel, Y Wang, PB Friedland, Very short-term load forecasting: wavelet neural networks with data pre-filtering. IEEE Trans. Power Syst. **28**, 30–41 (2013)
37. E-Control, Sonstige Marktregeln Strom – Kapitel 6: Zählwerte, Datenformate und standardisierte Lastprofile. E-Control, Vienna, Austria (2011). https://www.e-control.at/recht/marktregeln/sonstige-marktregeln-strom. Accessed 16 Aug 2016