Protecting the grid topology and user consumption patterns during state estimation in smart grids based on data obfuscation

Nandakumar, Lakshminarayanan; Tillem, Gamze; Erkin, Zekeriya; Keviczky, Tamas

DOI
10.1186/s42162-019-0078-y

Publication date
2019

Document Version
Final published version

Published in
Energy Informatics

Nandakumar *et al. Energy Informatics* (2019), 2(Suppl 1): 25
https://doi.org/10.1186/s42162-019-0078-y

Energy Informatics

RESEARCH Open Access

Protecting the grid topology and user consumption patterns during state estimation in smart grids based on data obfuscation

Lakshminarayanan Nandakumar[1*], Gamze Tillem[2], Zekeriya Erkin[2] and Tamas Keviczky[3]

From The 8th DACH+ Conference on Energy Informatics
Salzburg, Austria. 26-27 September, 2019

*Correspondence:
srinath0393@gmail.com
[1]CGI Nederland B.V, Rotterdam, The Netherlands
Full list of author information is available at the end of the article

Abstract

Smart grids promise a more reliable, efficient, economically viable, and environment-friendly electricity infrastructure for the future. State estimation in smart grids plays a pivotal role in system monitoring, reliable operation, automation, and grid stabilization. However, the power consumption data collected from the users during state estimation can be privacy-sensitive. Furthermore, the topology of the grid can be exploited by malicious entities during state estimation to launch attacks without getting detected. Motivated by the essence of a secure state estimation process, we consider a weighted-least-squares estimation carried out batch-wise at repeated intervals, where the resource-constrained clients utilize a malicious cloud for computation services. We propose a secure masking protocol based on data obfuscation that is computationally efficient and successfully verifiable in the presence of a malicious adversary. Simulation results show that the state estimates calculated from the original and obfuscated dataset are exactly the same while demonstrating a high level of obscurity between the original and the obfuscated dataset both in time and frequency domain.

Keywords: State estimation, Smart grids, Data obfuscation, Privacy

Introduction

Smart grids are widely regarded as a key ingredient to reduce the effects of growing energy consumption and emission levels (Commission 2014b). By 2020, the European Union (EU) aims to replace 80% of the existing electricity meters in households with smart meters (Commission 2014b). Currently, there are close to about 200 million smart meters accounting for 72% of the total European consumers (Commission 2014b). This smart metering and smart grid rollout can reduce emissions in the EU by up to 9% and annual household energy consumption by similar amounts (Commission 2014b). Despite the environment-friendly and the cost-cutting nature of the smart grid, deployment of smart meters at households actually raises serious data privacy and security concerns for the users. For example, with the advent of machine learning and data mining techniques, occupant activity patterns can be deduced from the power consumption

measurement data (Molina-Markham et al. 2010; Lisovich et al. 2010; Kursawe et al. 2011; Zeifman and Roth 2011). Additionally, the configuration of the power network/grid topology can be used by attackers to launch stealth attacks (Liu et al. 2011). Thus, despite the apparent benefits, without convincing privacy and security guarantees, users are likely to be reluctant to take risks and might prefer conventional meters to smart meters.

State estimation in smart grids enables the utility providers and Energy Management Systems (EMS) to perform various control and planning tasks such as optimizing power flow, establishing network models, and bad measurement detection analysis. State estimation is a process of estimating the unmeasured quantities of the grid such as the phase angle from the measurement data. The operating range of the state variables determines the current status of the network which enables the operator to perform any necessary action if required. The state of the system, the network topology, and impedance parameters of the grid can be used to characterize the entire power system (Huang et al. 2012). Traditionally, the centralized state estimation technique with the weighted-least-squares method yielded a very accurate result (Rahman and Venayagamoorthy 2017). However, now due to the increased complexity and the scale of the grid size, state estimation in a wide area grid network requires multiple smart meters from different localities to share data, some of which could be hosted by a third-party cloud infrastructure (Kim et al. 2011) due to coupling constraints, superior computational resources, greater flexibility, and cost-effectiveness.

The problem with the current cloud computation practice is that it operates mostly over plaintexts (Ren et al. 2012; Deng 2017); hence users reveal data and computation results to the commercial cloud (Ren et al. 2012). It becomes a huge problem when the user data contains sensitive information such as the power consumption patterns in smart meters. Moreover, there are strong financial incentives for the cloud service provider to return false results especially if the clients cannot verify or validate the results (Wang et al. 2011). For example, the cloud service provider could simply store the previously computed result and use it as the result for future computation problems to save computational costs. A recent breakthrough in fully homomorphic encryption (FHE) (Gentry and Boneh 2009) has shown that secure computation outsourcing is viable in theory. However, applying this mechanism to compute arbitrary operations and functions on encrypted data is still far from practice due to its high complexity and overhead (Wang et al. 2011). This problem leads researchers to alternative mechanisms for the design of efficient and verifiable secure cloud computation schemes.

Existing work and our contributions

Numerous privacy challenges related to smart grids are pointed out in the literature in different contexts. Amongst them, the most popular and widely studied is the privacy-preserving billing and data aggregation problem in smart grids (Molina-Markham et al. 2010; Kursawe et al. 2011; Erkin 2015; Ge et al. 2018; Knirsch et al. 2017; Emura 2017; Danezis et al. 2013). Our main objective is different from these work since we focus on the privacy concerns of state estimation in smart grids. Existing literature in smart grid state estimation problem focuses either on the problem of protecting the grid topology (Liu et al. 2011; Rahman and Venayagamoorthy 2017; Deng et al. 2017) or on preserving

the power consumption data of the users separately (Kim et al. 2011; Beussink et al. 2014; Tonyali et al. 2016). In Liu et al. (2011), the authors present a new class of attacks called false data injection attacks (FDI) against state estimation in smart grids and show that an attacker can exploit the configuration of a power network to successfully introduce arbitrary errors into the state variables while bypassing existing techniques for bad measurement detection. The authors in Deng et al. (2017) propose a design for a least-budget defense strategy to protect the power system from such FDI attacks. The authors in Rahman and Venayagamoorthy (2017) extends this problem to a non-linear state estimation and examines the possibilities of FDI attacks in an AC power network. To preserve the privacy of the user's daily activities, (Kim et al. 2011) exploits the kernel of the electric grid configuration matrix. In Beussink et al. (2014), a data obfuscation approach for an 802.11s-based mesh network is proposed to securely distribute obfuscated values along the routes available via 802.11s. The obfuscation approach in Tonyali et al. (2016) tackles this problem through advanced encryption standard (AES) scheme for hiding the power consumption data and uses elliptic-curve cryptography (ECC) for authenticating the obfuscation values that are distributed within the advanced metering infrastructure (AMI) network.

Contrary to the above work in smart grid state estimation, we focus on protecting *both* the power consumption data of the users and the grid topology. An open problem pointed out in Efthymiou and Kalogridis (2010); Li et al. (2010); Kim et al. (2011) is to provide a light-weight implementation of state estimation that can run in a smart meter platform. In this paper, we attempt to solve this problem by proposing *Obfuscate*(.), an efficient secure masking scheme based on randomization. Our scheme obfuscates the measurement data of a collection of smart meters installed in a particular locality and send it to the lead smart meter in their respective locality. These lead smart meters, in turn, gather these randomized data and send it to the cloud service provider to perform the required computations.

The major contributions of our paper are as follows:

- We propose *Obfuscate*(.), the first batch-wise state estimation scheme in smart grids with the goal of protecting *both* the power consumption data of the consumers and the grid topology. Our scheme is based on secure masking through obfuscated transformation and is proven to be efficient with no major computational overhead to the users.
- We evaluate the performance of *Obfuscate*(.) with real-time hourly power consumption dataset of different smart meters. We use the dataset under the assumption that these meters are connected to an IEEE-14 bus test grid system and a fully measured 5 bus power system. Furthermore, we evaluate the illegibility of the obfuscated dataset with respect to the original dataset.

In the rest of the paper, first we discuss the necessary prerequisites on state estimation in smart grids and the adversarial models in "Background information" section. In "Secure state estimation with *Obfuscate*(.)" section, we explain *Obfuscate*(.) in detail. In "Analyses of *Obfuscate*(.)" section, we present the correctness, privacy, verification and complexity analyses of our scheme. In "Simulation results" section, we present the simulation results and we conclude the paper in "Conclusions and future work" section.

Background information

Static state estimation in electric grids

The static state estimation (SSE) in smart grids is a well established problem with well-known techniques that rely on a set of measurement data to estimate the states at regular time intervals (Schweppe and Wildes 1970; Schweppe and Rom 1970; Schweppe 1970). The state vector $x = [x_1, x_2, \cdots x_n]^T \in \mathbb{R}^n$ represents the phase angles at each electric branch or system node, and the measurement data $z \in \mathbb{R}^m$ denotes the power readings of the smart meters. The state vector x and the measurement data z are related by a nonlinear mapping function h such that $z = h(x) + e$, where the sensor measurement noise e is a zero-mean Gaussian noise vector. Typically, for state estimation a linear approximation of this equation is used (Kim et al. 2011; Liu et al. 2011; Gera et al. 2017) as $z = \mathbf{H}x + e$, where $\mathbf{H} \in \mathbb{R}^{m \times n}$ is the full column rank ($m > n$) measurement Jacobian matrix determined by the grid structure and line parameters (Liang et al. 2017). The matrix $\mathbf{H}$ is known as the *grid configuration* or the *power network topology* matrix (Kim et al. 2011; Liang et al. 2017; Gera et al. 2017). In an electric grid $m \gg n$ (Zimmerman et al. 2009) and the best unbiased linear estimation of the state (Wood and Wollenberg 1996) is given by

$$\hat{x} = \left(\mathbf{H}^T W \mathbf{H}\right)^{-1} \mathbf{H}^T W z, \tag{1}$$

where $W^{-1} \in \mathbb{R}^{m \times m}$ represents the covariance matrix of the measurement noise. W^{-1} is taken to be a diagonal matrix $W^{-1} = \sigma^2 I$ (Wood and Wollenberg 1996), so Eq. 1 reduces to

$$\hat{x} = \left(\mathbf{H}^T \mathbf{H}\right)^{-1} \mathbf{H}^T z. \tag{2}$$

The SSE technique reduces the computational complexity of performing state estimation in smart grids, where the estimates are usually updated on a periodic basis (Huang et al. 2012). Measurement devices in current transmission systems are installed specifically catering to the needs of SSE (Krause and Lehnhoff 2012). The recent evolution of phasor measurement units (PMUs) are able to measure voltage and line current phasors with high accuracy and sampling rates. However, deployment of a large number of PMUs across the system requires significant investments since the average overall cost per PMU ranges from $40k to $180k (Department of Energy 2014). Hence SSE will remain an important technique to estimate the state variables at medium and low voltage levels (Cosovic and Vukobratovic 2017). Practically, state estimation is run only for every few minutes or only when a significant change occurs in the network (Cosovic and Vukobratovic 2017; Monticelli 2000).

Bad measurement detection (BMD)

Bad measurements may be introduced due to meter failures or malicious attacks. They may affect the outcome of state estimation and can mislead the grid control algorithms, possibly causing catastrophic consequences such as blackouts in large geographical areas. For example, a large portion of the Midwest and Northeast United States and Ontario, Canada, experienced an electric power blackout affecting a population of about 50 million (n.a. 2003). The power outage cost was about $80bn in the USA and usually, the utility operators amortize it by increasing the energy tariff, which is unfortunately transferred to consumer expenses (Salinas and Li 2016). Thus, BMD is vital to ensure smooth and reliable operations in the grid.

The most common technique to detect bad measurements is to calculate the L_2-norm $\|z - \mathbf{H}\hat{x}\|$, and if $\|z - \mathbf{H}\hat{x}\| > \tau$, where τ is the threshold limit, then the measurement z is considered to be bad. The reason is that, intuitively, normal sensor measurements yield estimates closer to their actual values, while abnormal ones deviate the estimated values away from their true values. This inconsistency check is used to differentiate the good and the bad measurements (Liu et al. 2011). However, this is not always the case, as exposing **H** could make the grid vulnerable to stealth attacks (Liu et al. 2011). Liu, Reiter and Ning proved that a malicious entity can exploit the row and column properties of **H** when exposed, and launch false data injection attacks without getting detected (Liu et al. 2011). The **H** matrix includes the arrangement of of loads or generators, transmission lines, transformers, and status of system devices and is an integral part of state estimation, security, and power market design (Gera et al. 2017). Thus, there is a strong need to protect not just the power consumption data but also the power network topology during state estimation.

Cryptographic preambles

To understand the privacy goals of our problem, we state the following definitions:

Obfuscation (Shoukry et al. 2016) is the procedure of transforming the data into masked data through randomization and performing the necessary operations on this masked obfuscated data. The obfuscated data can be unmasked by inverting the randomized transformation using the respective private keys.

Semi-honest Adversary (Lindell and Pinkas 2009) is an adversary who correctly follows the protocol specification but keeps track of all the information exchanged to possibly analyze it together with any other public information to leak sensitive data. It is also known as *honest-but-curious* or *passive* adversary.

Malicious Adversary (Lindell and Pinkas 2009) is an adversary who can arbitrarily deviate from the protocol specification. Here the attacks are no longer restricted to eavesdropping since the adversary might actually inject or tamper with the data provided. It is also known as *active* adversaries.

Secure state estimation with *Obfuscate*(.)

In this section, we explain our secure state estimation protocol *Obfuscate*(.) along with the setup and the threat model.

Setup

Let an area $\mathcal{A}$ consist of two localities[1] denoted by L_1 and L_2 as shown in Fig. 1. The symbol S_{ij} refers to the smart meter installed at the household j situated in locality L_i and $X_i \in \mathbb{R}^{n_i \times T}$ denotes the state sequences of all the smart meters installed in L_i for a given batch of time duration T. The electric grid configuration matrix of L_i is represented as $\mathbf{H}_i$ and the coupling matrices between L_i and L_j are denoted as $\mathbf{H}_{ij}$ and $\mathbf{H}_{ji}$ respectively. The symbol $[\cdot]$ denotes the obfuscation of a vector or matrix. For example, $[Z_i]$ represents the obfuscated value of the matrix $Z_i \in \mathbb{R}^{m_i \times T}$ where m_i is the number of smart meters in L_i. The participating entities in our design are as follows:

[1] For brevity, here we assume that the area consists of only two localities. The protocol presented in this paper can easily be extended to an area consisting of more than two localities.

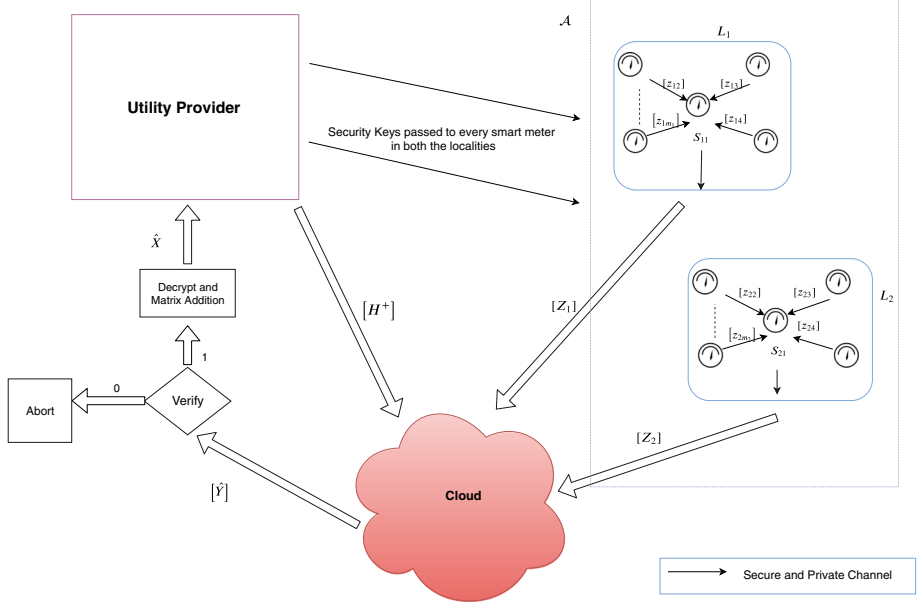

Fig. 1 Proposed solution framework

Utility Provider $\mathcal{U}$: provides utility services to $\mathcal{A}$ and has access to the grid configuration matrix **H**. $\mathcal{U}$ generates all the keys to initiate *Obfuscate*(.) and distributes a selected portion of these keys to the smart meters at each locality through a private channel to carry out obfuscation. $\mathcal{U}$ is a decision-making entity performing any necessary action after receiving the state variables at regular intervals.

Lead Smart Meter S_{i1} receives the randomized masked data from the other meters connected to it and obfuscates the dynamics of the power consumption pattern of all the meters in its locality. Then, sends it to the cloud for state estimation. The lead meter at every locality is assumed to be a trusted node in the local network. A similar entity was proposed in Kim et al. (2011) where the lead meter is connected to all the meters based on the mesh topology network. The lead meter, for instance, could be the local distributed system operator (DSO) of a particular locality.

Other Smart Meters S_{ij} ($\forall j \neq 1$) are all the other meters in L_i. They obfuscate their measurement data and send it to the lead meter S_{i1} to avoid leaking information about their respective consumptions to any potential eavesdropping.

Cloud $\mathcal{C}$ is computationally super efficient and hence provides computation services for $\mathcal{A}$ performing state estimation. As pointed out before, since most of the current cloud computations are performed in plaintext, modeling the cloud as a malicious entity is crucial in practice.

Threat model

The smart meters in L_i and L_j, where $j \neq i$, are considered to be semi-honest to each other i.e., clients living in different localities are *curious* about each other consumption data. This means that people who are situated geographically apart may try to learn information about people in other localities such as energy usage consumption pattern, pricing, etc. Also, households living in the same locality are modeled to be honest-but-curious.

Albeit, it is natural for people living in the same locality - next to each other to have at least some prior knowledge about each other's activity pattern, it is not acceptable if the neighbors can deduce the usage of a particular appliance at a given time-stamp applying techniques such as non-intrusive load monitoring (Zeifman and Roth 2011) to the original power consumption data. Thus, all the smart meters in a particular locality securely mask their consumption data before sending it to their respective lead meter.

Unlike the problem of protecting the user power consumption data from the utility provider for billing, data aggregation and other statistical purposes (Kursawe et al. 2011; Erkin 2015; Ge et al. 2018; Knirsch et al. 2017; Emura 2017; Danezis et al. 2013), here we study the problem of carrying out secure state estimation by outsourcing the data to an untrusted third party. These state variables with high accuracy are essential to the utility provider for effective decision-making and providing good quality services such as demand forecasting, optimal power flow, and contingency analysis. Hence $\mathcal{U}$ here is not considered to be an adversarial entity and is non-colluding in nature. The utility provider's main objective is to earn the consumer trust by protecting their privacy and encouraging more user participation to install smart meters for business and commercial purposes. Investment in smart metering technology is directly impacted by customer trust in the utility operators (Commission 2014a). To protect the privacy of consumers, utility providers make use of secure communication channels and databases with access control (Kim et al. 2011). In addition, with EU's newly devised General Data Protection Regulation (GDPR), energy companies are liable to pay large penalties up to €20m (Hunt 2017), if customer data are misused. One might argue about the need to apply a similar compliance factor to the cloud service provider. However, the major problem specific to cloud computation services is that, with the current technology, most of the computations in the cloud are performed in plaintext (Ren et al. 2012; Deng 2017). Arbitrary computations on encrypted data using FHE schemes are still under active research for effective implementation (Tebaa and Hajji 2014). Providing data in the clear makes the cloud vulnerable to both active and passive attacks. According to the latest Microsoft security intelligence report (Simos 2017), the number of attacks in the cloud environment has increased by 300% which further justifies considering the cloud as a malicious entity in our problem setup.

Obfuscate(.)

The aim of our scheme is to protect the privacy of the power consumption data of the consumers Z *and* the grid configuration matrix $\mathbf{H}$ during state estimation, while outsourcing these pieces of information to an untrusted malicious third party cloud. Our design goals are as follows:

Input/Output Privacy: Neither the input data sent nor the output data computed by the cloud should be inferred by the cloud.

Correctness: Any cloud server faithfully following the protocol must be able to compute an output that can be verified successfully.

Verification: If the cloud server acts maliciously, then it should not be able to pass the utility-side verification test with a high probability.

Efficiency: Computational overhead for the clients ($\mathcal{U}$ and S_{ij}) should be minimal.

Nevertheless it is important to note that local smart meters in the localities cannot estimate the states on their own due to the coupling constraints (See Eq. 3). The

efficiency criteria is mainly considered to exploit the nearly unlimited computational resources of the cloud. Furthermore, since the smart meters in different neighborhoods are semi-honest to each other, the designed protocol should also guarantee a very low probability of a particular neighbour inferring any sensitive information through eavesdropping and combining any other publicly available information of the localities.

Proposed scheme

Consider the proposed scheme depicted in Fig. 1. The equation $z = \mathbf{H}x + e$, can be rewritten as :

$$\begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} = \underbrace{\begin{bmatrix} H_1 & H_{12} \\ H_{21} & H_2 \end{bmatrix}}_{\mathbf{H}} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}, \tag{3}$$

where $H_1 \in \mathbb{R}^{m_1 \times n_1}$ and $H_2 \in \mathbb{R}^{m_2 \times n_2}$ are the grid configuration matrix of L_1 and L_2. The matrix $H_{12} \in \mathbb{R}^{m_1 \times n_2}$ and $H_{21} \in \mathbb{R}^{m_2 \times n_1}$ denote the coupling matrices. The measurement data and the states of Locality L_i are represented by $Z_i \in \mathbb{R}^{m_i \times T}$ and $X_i \in \mathbb{R}^{n_i \times T}$ respectively. The solution to Eq. 3 is given by Eq. 2.

In general, the configuration of the power network $\mathbf{H}$ is not time-varying during the state estimation process (Schweppe and Wildes 1970; Schweppe and Rom 1970; Schweppe 1970; Wood and Wollenberg 1996), and hence the matrix $\mathbf{H}^+ = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T$ can be pre-computed during the offline stage. Typically, this information is computed during the creation of the power network by the utility provider using a trusted party. Hence, the state estimation can be recast and reduced into $\hat{X} = \mathbf{H}^+ Z$, where $\hat{X} \in \mathbb{R}^{n \times T}$, $Z \in \mathbb{R}^{m \times T}$ and $\mathbf{H}^+ \in \mathbb{R}^{n \times m}$ with $m = m_1 + m_2$ and $n = n_1 + n_2$. Thus, our privacy-aware state estimation problem can be recast into solving a matrix multiplication securely. The matrix $\mathbf{H}^+$ can be rewritten block-wise as follows:

$$\mathbf{H}^+ = \left(\begin{bmatrix} H_1 & H_{12} \\ H_{21} & H_2 \end{bmatrix}^T \begin{bmatrix} H_1 & H_{12} \\ H_{21} & H_2 \end{bmatrix} \right)^{-1} \begin{bmatrix} H_1 & H_{12} \\ H_{21} & H_2 \end{bmatrix}^T = \begin{bmatrix} F_1 & F_{12} \\ F_{21} & F_2 \end{bmatrix}, \tag{4}$$

where $F_1 \in \mathbb{R}^{n_1 \times m_1}, F_2 \in \mathbb{R}^{n_2 \times m_2}, F_{12} \in \mathbb{R}^{n_1 \times m_2}$ and $F_{21} \in \mathbb{R}^{n_2 \times m_1}$. The exact expression of the F matrix is omitted here due to space constraints. Notice from $\hat{X} = \mathbf{H}^+ Z$ that it is not possible for the lead meter in each locality to carry out the estimation process locally due to the coupling constraints generated by the matrices H_{12} and H_{21}. Namely, the state estimate $\hat{X}_1$ also depends on the consumption data of the other locality Z_2 and vice versa. Thus, the lead meter collects all the obfuscated measurement data from the other meters in its locality and sends it to the cloud. The matrix $\mathbf{H}^+$ is obfuscated by the utility provider and sent to the cloud. However, it is important that the matrix $\mathbf{H}^+$ is not completely randomized using a single key but is randomized block-wise with different keys for different blocks (see Eq. 4). The estimation problem can be further broken down into

$$\begin{bmatrix} \hat{X}_1 \\ \hat{X}_2 \end{bmatrix} = \begin{bmatrix} F_1 Z_1 + F_{12} Z_2 \\ F_{21} Z_1 + F_2 Z_2 \end{bmatrix}. \tag{5}$$

Let us denote the matrix

$$Y = \begin{bmatrix} F_1 Z_1 & F_{12} Z_2 \\ F_{21} Z_1 & F_2 Z_2 \end{bmatrix} = \begin{bmatrix} Y_1 & Y_{12} \\ Y_{21} & Y_2 \end{bmatrix}. \quad (6)$$

Using Eq. 5 for estimating the states, we solve the matrix multiplication of each blocks in Eq. 6 privately and then perform matrix addition.

The matrix multiplication is a fundamental problem in cryptography and several solutions have been proposed to solve it (Atallah and Frikken 2010; Atallah et al. 2012; Fiore and Gennaro 2012; Zhang and Blanton 2014). However, these protocols are not designed for the cloud environment and hence do not consider the computational asymmetry of the cloud server and the client. Another drawback is that these protocols use advanced cryptography to encrypt the input and output dataset, which makes them unsuitable for the computation on the cloud with large datasets due to high overhead. Furthermore, the verification of the result, which is an essential requirement in a malicious cloud setting, is not considered in these protocols (Kumar et al. 2017). A secure multiparty computation (MPC) approach was considered in Dreier and Kerschbaum (2011); López-Alt et al. (2012), where the computation is divided among multiple parties without allowing any participating entity to access another individual's private information. However, this approach is not feasible for our problem setup since all the parties are required to have a comparable computing capability. Also, in MPC approach, the result verification is often troublesome since it requires expensive zero-knowledge proofs (Saia and Zamani 2015; Goldwasser et al. 2015).

Recently, a privacy-preserving, verifiable and efficient outsourcing algorithm for matrix multiplication to a malicious cloud was proposed in Kumar et al. (2017) utilizing linear transformation techniques. In our paper, we adopt a similar approach to the one prescribed in Kumar et al. (2017) to outsource the multiplication of block matrices in Eq. 6 securely to the cloud. However, *Obfuscate*(.) is not a straightforward application of the protocol in Kumar et al. (2017). Kumar et al. (2017) considers only a single client and a cloud setup, where the client performs the key generation, problem transformation, re-transformation and verification on his/her own. In our scheme, there are multiple smart meters installed in different neighborhoods. The keys cannot be generated locally by the individual households because the smart meters have access only to their respective consumption data which forms only a part of the information required for state estimation. Hence, besides the key generation we also propose *KeyDist* - a key distribution scheme as shown in Fig. 2 used by $\mathcal{U}$ to distribute keys to the smart meters. *Obfuscate*(.) comprises of eight subalgorithms which are explained in the rest of this section.

KeyGen$(1^{\lambda}, m_1, n_1)$ algorithm (Algorithm 1) takes as input the security parameter λ and generates a total of $n_1 + m_1$ non-zero random numbers each of bit size λ. These numbers are used to generate the key matrices of size $\mathbb{R}^{m_1}$ and $\mathbb{R}^{n_1}$. Table 1 shows the entire keys that are generated per batch.

After the *KeyDist*() (Algorithm 2), matrix transformation $\psi_K()$ is carried out by the respective entities using their respective keys K. For every new input matrix, $\psi_K()$ is invoked to securely mask the input through linear transformation in

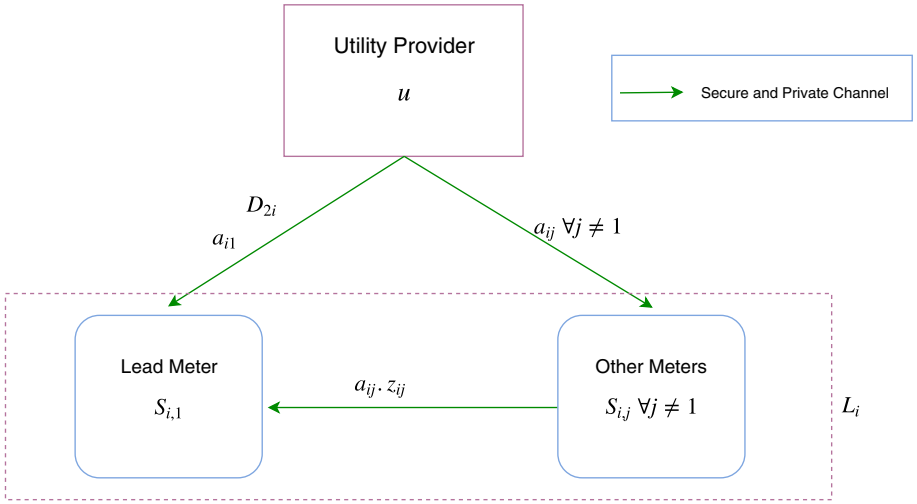

Fig. 2 A triangular key distribution scheme for a locality L_i

Algorithm 1 *KeyGen*

Input λ, m_1, n_1
$N_1 \rightarrow_R \{\alpha_1, \alpha_2 \cdots \alpha_{n_1}\}$
$M_1 \rightarrow_R \{\beta_{11}, \beta_{12} \cdots \beta_{1m_1}\}$
for $i = 1$ to n_1 **do**
    $d_i = \alpha_i.I_{(i,i)}$
end for
Set $D_1 = diag(d_1, d_2, \cdots d_{n_1})$
for $i = 1$ to m_1 **do**
    $b_i = \beta_{1i}.I_{(i,i)}$
end for
Set $A_1 = diag(b_1, b_2, \cdots b_{m_1})$.
Output: D_1 and A_1.
Repeat 1 to 10 per batch.

order to preserve the privacy. This operation dominates the client-side computation cost, but is not significant compared to the computations performed by the cloud. The matrix transformation for a given input matrix F_1 and Z_1 are given by Algorithm 3 and 4, respectively. Table 2 summarizes the complete matrix transformation protocol.

Table 1 Key generation protocol run by $\mathcal{U}$ per batch

Protocol	Output
$KeyGen(1^\lambda, n_1, m_1)$	D_1, A_1
$KeyGen(1^\lambda, n_2, m_2)$	D_5, A_2
$KeyGen(1^\lambda, n_2, A_1)$	D_3, A_1
$KeyGen(1^\lambda, n_1, A_2)$	D_6, A_2
$KeyGen(1^\lambda, T)$	D_2
$KeyGen(1^\lambda, T)$	D_4

Algorithm 2 *KeyDist*

Input: $A_i, D_{2.i}$
Set $a_{ij} = \frac{1}{\beta_{ij}}$
for $j = 2$ to m_i **do**
    Send a_{ij} to S_{ij} through private channel $\{i, j\}$
end for
Send a_{i1} and $D_{2.i}$ to S_{i1} through private channel $\{i, 1\}$
Repeat 1 to 6 per batch.

Algorithm 3 MatrixTrans $\psi_K(F_1)$

Input D_1, A_1
for $i = 1$ to n_1 **do**
    $[F_1(i,:)] = D_1(i,i).F_1(i,:)$
end for
for $i = 1$ to m_1 **do**
    $[F_1(:,i)] = F_1(:,i).A_1(i,i)$
end for
Output $[F_1]$

Algorithm 4 *MatrixTrans* $\psi_K(Z_1)$

Input D_2 ▷ Here i =1 considering first locality.
for $j = 2$ to m_1 **do**
    Send $z'_{1j} = a_{1j}.z_{1j}$ to S_{11}.
end for
S_{11} constructs $Z'_1 = A_1^{-1}.Z_1$. ▷ $a_{1j} = 1/\beta_{1j}$
for $i = 1$ to T **do**
    $[Z_1(:,i)] = Z'_1(:,i).D_2(i,i)$
end for
Output $[Z_1]$

Next, the obfuscated matrix $\mathbf{H}^+$ and the masked measurement matrix Z_i are sent by $\mathcal{U}$ and S_{i1}, respectively to $\mathcal{C}$ to perform $Compute_\psi([F_1], [Z_1])$ algorithm given in Algorithm 5. This algorithm performs the computation on the cloud server. It computes MM as $\psi([F_1], [Z_1]) = (D_1F_1A_1).\left(A_1^{-1}Z_1D_2\right)$. Table 3 shows the $Compute_\psi()$ protocol run by the cloud server for estimating the state samples.

Table 2 Matrix transformation protocol run per batch

Protocol	Keys	Run by	Output
$\psi_K(F_1)$	D_1, A_1	$\mathcal{U}$	$[F_1]$
$\psi_K(F_2)$	D_2, A_2	$\mathcal{U}$	$[F_2]$
$\psi_K(F_{12})$	D_6, A_2	$\mathcal{U}$	$[F_{12}]$
$\psi_K(F_{21})$	D_3, A_1	$\mathcal{U}$	$[F_{21}]$
$\psi_K(Z_1)$	D_2, A_1^{-1}	S_{11}	$[Z_1]$
$\psi_K(Z_2)$	D_4, A_2^{-1},	S_{21}	$[Z_2]$

Algorithm 5 $Compute_{\psi}$

Input $[F_1], [Z_1]$
$\mathcal{C}$ computes $[Y_1] = [F_1] . [Z_1]$
Output $[Y_1]$

Upon computing the Y matrix, the cloud sends the computed result to the utility provider $\mathcal{U}$ to execute the verification step. $Verify([Y], \gamma)$ algorithm computes $Q = ([F] \cdot ([Z] \cdot \gamma)) - ([Y] \cdot \gamma)$, where γ is a binary key matrix of size T i.e. $\gamma \in \{1, 0\}^T$. The algorithm introduces the binary column matrix key γ to minimize the complexity of computation since the matrix-vector multiplication only cost quadratic time. The verification protocol for L_i is given in Algorithm 6.

Algorithm 6 $Verify([Y_i], \gamma_i)$

Input: $[Y_i]$, $[F_i]$, $[Z_i]$
$\mathcal{U}$ generates $\gamma_i \in \{0, 1\}^T$ and sends it to S_{i1} through a private channel.
S_{i1} computes $Z_{\gamma_i} = [Z_i] . \gamma_i$ and sends it to $\mathcal{U}$.
$\mathcal{U}$ computes $Q_i = [Y_i] . \gamma - F_i . Z_{\gamma_i}$
if $(Q_i == \{0, 0, \cdots, 0\}^T)$ **then**
    return (1)
else
    return (0)
end if

It is important to note that the verification step serves as the BMD test in our setup and is run for all the four block matrices given by Eq. 6. Table 4 presents the verification protocol. The results are accepted only if the cloud server passes all the four verification tests. If the verification is positive, then it means that no false data has been injected into the measurements by the cloud which is conclusive to the absence of bad measurements in the network.

After positive verification, $Unmask(Y, K)$ algorithm (Algorithm 7) is run by $\mathcal{U}$. This algorithm returns the original values of the states $\hat{X}$ by de-randomizing Y using their respective keys K. Table 5 summarizes the $Unmask()$ protocol carried out for all the four block matrices. Once, all the four blocks of Y are unmasked, $\mathcal{U}$ carries out the protocol given in Algorithm 8 to reach the final state estimates.

Table 3 Computation protocol run by $\mathcal{C}$ per batch

Protocol	Output
$Compute_{\psi}([F_1], [Z_1])$	$[Y_1]$
$Compute_{\psi}([F_2], [Z_2])$	$[Y_2]$
$Compute_{\psi}([F_{21}], [Z_1])$	$[Y_{21}]$
$Compute_{\psi}([F_{12}], [Z_2])$	$[Y_{12}]$

Table 4 Verification Protocol run by $\mathcal{U}$ per batch

Protocol	Output
$Verify([Y_1], \gamma_1)$	Q_1
$Verify([Y_2], \gamma_2)$	Q_2
$Verify([Y_{12}], \gamma_2)$	Q_{12}
$Verify([Y_{21}], \gamma_1)$	Q_{21}

Algorithm 7 $Unmask([Y_1], K)$

1: **Input** $[Y_1]$ and the respective keys D_1 and D_2
2: Compute D_1^{-1} and D_2^{-1}
3: Compute $Y_1 = D_1^{-1}.[Y_1].D_2^{-1}$
4: **Output** Result Y_1

Algorithm 8 $MatrixAdd(Y)$

1: **Input** Y_1, Y_2, Y_{12}, Y_{21}
2: Compute $\hat{X}_1 = Y_1 + Y_{12}$
3: Compute $\hat{X}_2 = Y_{21} + Y_2$
4: **Output** Result $\hat{X}_1$, $\hat{X}_2$

Analyses of *Obfuscate*(.)

In this section, we show that *Obfuscate*(.) complies with the design goals stated in "Secure state estimation with *Obfuscate*(.)" section which are correctness, privacy, verifiability, and efficiency.

Correctness analysis

If the smart meters, utility provider, and the cloud correctly follow *Obfuscate*(.) as per the protocol, then *Obfuscate*(.) produces correct results for all the four matrix multiplications. This follows from a simple proof:

Proof $\mathcal{U}$ first transforms the matrix F_1 into $[F_1] = D_1 F_1 A_1$ and the lead smart meter in L_1 transforms the matrix $Z_1' = A^{-1} Z_1$ into $[Z_1] = A_1^{-1} Z_1 D_2$. The cloud server computes $[Y_1] = [F_1] \cdot [Z_1] = (D_1 F_1 A_1) \cdot \left(A_1^{-1} Z_1 D_2\right) = D_1 Y_1 D_2$. Then, in the de-randomization step, $\mathcal{U}$ computes Y_1, where $Y_1 = D_1^{-1} [Y_1] D_2^{-1} = F_1 \cdot Z_1$. □

The above analysis holds for all the $Compute_{\psi}(.)$ presented in Table 3, thereby proving the correctness of *Obfuscate*(.).

Table 5 Unmasking Protocol run by $\mathcal{U}$

Protocol	Keys	Output
$Unmask([Y_1], K)$	D_1, D_2	Y_1
$Unmask([Y_2], K)$	D_5, D_4	Y_2
$Unmask([Y_{21}], K)$	D_3, D_2	Y_{21}
$Unmask([Y_{12}], K)$	D_6, D_4	Y_{12}

Privacy analysis

Input Privacy: Since $\mathcal{C}$ has only access to the masked input matrices $[F]$ and $[Z]$, it cannot not retrieve the original input matrices F and Z. Furthermore, the keys generated as in Table 1 do not leak any information about the original input since the keys are completely random devoid of dependency on the topology and the power consumption data. This can be seen from the following proof:

Proof The key matrix A_1 and A_2 are diagonal matrices with each element being a random real number of λ bit long. There are $2^{m_i\lambda}$ possibilities of A_i matrix where $i \in \{1, 2\}$. For diagonal matrices D_1 and D_2, there are in total $2^{n_1\lambda+T\lambda}$ possibilities. Thus for a single block F_1 in Y, there are a total of $2^{(m_1+n_1+T)\lambda}$ possible choices of key matrices, which is an exponential bound quantity in terms of (m_1, n_1, T). □

For example, consider a practical scenario where a locality has $m_1 = 1000, n_1 = 600, T = 400$ for which we have $2^{2000\lambda}$ possibilities. Thus, with increase in m_1, n_1 and T, the cloud does not recover any meaningful information.

Output Privacy: Similar to the input privacy analysis, the output result is also protected. The resulting obfuscated matrix does not leak any information to $\mathcal{C}$, even if it records all the computed results. Besides, for every batch, $\mathcal{U}$ generates new keys given in Table 1 which makes our protocol resistant to any known-plain-text attack (KPA) or chosen-plain-text-attack (CPA) (Kumar et al. 2017).

Verification analysis

Since in a malicious threat model, the cloud server may deviate from the actual instructions of the given protocol, we equip *Obfuscate*(.) with a result verification algorithm to validate and verify the correctness of the result. The proof that a wrong or an invalid result never passes the verification step follows from the *total probability theorem* as followed in Kumar et al. (2017); Lei et al. (2013).

Proof If the cloud produces the correct result say Y_1, then $Q_1 = ([F_1] \cdot [Z_1] - [Y_1]) = [0, 0, \cdots 0]^T$. If the cloud produces the wrong result, then $Q_1 \cdot \gamma_1 \neq [F_1]\,[Z_1] \cdot \gamma - [Y_1]\,.\gamma$, i.e. there exists at least a row in Q_1 which is not equal to zero, $Q_1\gamma_1 = [q_1, \cdots q_{m_1}]^T$. Let the row $q_i \neq 0$, where

$$q_i = \sum_{j=1}^{T} Q_{1i,j} \cdot \gamma_j = Q_{1i,1} \cdot \gamma_1 + \cdots Q_{1i,k} \cdot \gamma_k + Q_{1i,T} \cdot \gamma_T.$$

There exists at least one element in this row which is not equal to zero. Let $Q_{1i,k} \neq 0$, $q_i = Q_{1i,k} \cdot \gamma_k + \Gamma$ where $\Gamma = \sum_{j=1}^{T} Q_{1i,j} \cdot \gamma_j - Q_{1i,k} \cdot \gamma_k$. Applying the total probability theorem yields,

$$\Pr(q_i = 0) = \Pr[\,(q_i = 0)|(\Gamma = 0)]\,\Pr[\,\Gamma = 0] + \Pr[\,(q_i = 0)|(\Gamma \neq 0)]\,\Pr[\,\Gamma \neq 0] \tag{7}$$

$$\Pr[\,(q_i = 0)|(\Gamma = 0)] = \Pr[\,\gamma_k = 0] = 1/2 \tag{8}$$

$$\Pr[\,(q_i = 0)|(\Gamma \neq 0)] \leq \Pr[\,\gamma_k = 1] = 1/2$$

Substituting (8) in (7), we derive

$$\begin{aligned} &\Pr[\,(q_i = 0)] \leq 1/2\Pr[\,\Gamma = 0] + 1/2\Pr[\,\Gamma \neq 0]\,, \\ &\Pr[\,(q_i = 0)] \leq 1/2(1 - \Pr[\,\Gamma \neq 0]\,) + 1/2\Pr[\,\Gamma \neq 0]\,, \\ &\Pr[\,(q_i = 0)] \leq 1/2. \end{aligned} \tag{9}$$

If the verification process is run p times, then $\Pr[\,(q_i = 0)] \leq 1/2^p$. □

The value p reveals the trade-off between computational efficiency and verifiability. Theoretically $p \geq 80$ is sufficient to ensure *negligible* probability for the cloud to pass the verification test despite producing wrong result. However, in practice, $p = 20$ is acceptable with $1/2^{20} \approx 1$ million (Kumar et al. 2017; Lei et al. 2013). The verification process fails to detect a wrong result one in a million times.

Efficiency analysis

In this section, we carry out the computation complexity analysis to prove the efficiency of *Obfuscate*(.). The computational cost of each step in *Obfuscate*(.) is analyzed in Table 6. *KeyDist*() protocol introduces an additional communication cost of $O(m)$ since $\mathcal{U}$ distributes the key a_{ij} to all the smart meters through a private channel for obfuscating their measurement data. In Table 6, it is clear that the computations performed by the client side are substantially lower than that of the cloud server. Due to the diagonal structure of the key matrices, the problem transformation step given by Algorithms 3 and 4 only costs $O(nm + mT)$. The asymptotic complexity of the client side computation is only $O(nm + mT + nT)$ (Kumar et al. 2017). Thus, outsourcing the computation yields a performance gain of $O\left(\frac{1}{n} + \frac{1}{m} + \frac{1}{T}\right)$. Clearly, when n, m, and T increases, the clients will achieve a higher performance gain. Especially, with the increase in the number of smart meters m by the year 2020 as aimed by the EU (Commission 2014b), *Obfuscate*(.) will significantly reduce the computational overhead of its clients in the long run.

Simulation results

In this section, we evaluate the degree of obscurity of *Obfuscate*(.) using two case studies: a fully measured 5-bus system and the IEEE 14-bus system with real-time power consumption data. We start with a fully measured 5-bus system and the structure of the **H** matrix for this system can be found in the Appendix. In this case, the total number of meters $m = 10$ and the state variables $n = 4$. We consider $m_1 = 4$, $m_2 = 6$ and $n_1 = n_2 = 2$ and the duration of every batch to be 13 hours. Note in practice, smart

Table 6 Computation complexity analysis of the protocol

Client side computations			Cloud computations
Utility provider $\mathcal{U}$	S_{i1}	S_{ij}	
KeyGen - $O(m + n + T)$	–	–	$Compute_{\psi}([F_1], [Z_1])$ - $O(n_1 m_1 T)$
MatrixTrans $\psi_K(.)$ - $O(n_1 m_1 + n_1 m_2 + n_2 m_1 + n_2 m_2) = O(nm)$	$O(m_i T)$	O(1)	$Compute_{\psi}([F_{12}], [Z_2])$ - $O(n_1 m_2 T)$
Verify - $O(n_1 T + n_2 T) = O(nT)$	$O(m_i T)$	–	$Compute_{\psi}([F_{12}], [Z_2])$ - $O(n_2 m_1 T)$
Unmask - $O(n_1 T + n_2 T) = O(nT)$	–	–	$Compute_{\psi}([F_2], [Z_2])$ - $O(n_2 m_2 T)$
MatrixAdd - $O(n_1 T + n_2 T) = O(nT)$	–	–	$Compute_{\psi}(\left[\mathbf{H}^+\right], [Z])$ -,$O(nmT)$
Total client-side computation cost $\approx O(nm + mT + nT)$			Total cloud computation cost = $O(nmT)$

meters can sample at much higher frequencies (Chen et al. 2011). Research on disaggregating electricity load has been conducted on smart meter readings with a fine granularity of frequency between 1 Hz to 1 MHz (Chen et al. 2011). The authors in Kim et al. (2011) collected real-time power consumption data of both residential and office spaces with a sampling rate of 1 Hz. Hence in practice the number of data points collected per batch T could be in order of tens of thousands. However, due to the unavailability of such high-frequency measurement data, we restrict the size of T. Since, we had access to only hourly power consumption data we restrict $T = 13$. Although the size of the matrix $Z \in \mathbb{R}^{m \times T}$ is smaller than in practice, the state estimation still cannot be performed locally due to the coupling constraints between the two localities. Upon inspecting the power consumption values of all the meters, we found these values are mostly 4 to 5 decimal digits long. To mask this data securely, we use a key size of length $\lambda = log_2(10^5) \approx 16 + 80 \approx 96$ bits. The additional 80 bits ensures that *Obfuscate*() follows the National Institute of Standards and Technology (NIST) recommendations[2] to securely mask the data. Based on the present computational capabilities, it is not possible to break our scheme, thereby proving it's robustness in terms of attack from a malicious adversary.

Figure 4 shows the illegibility of the *Obfuscate*(.) for a fully measured 5-bus power system. Illegibility measures the level of difficulty of interpreting and mining data to the malicious cloud server (Kim et al. 2011). In Fig. 4a, we can see the original power consumption data of a household (blue) is always positive, whereas, the obfuscated data (red) show negative power readings and behave more as random variables. The degree of obscurity becomes more clear when transforming these datasets into the frequency domain. Figure 4b plots the Fast Fourier Transform (FFT) coefficients against various frequencies and shows that the original data consists mostly of low-frequency components, whereas the obfuscated data exhibits high-frequency components. This can also be inferred from the power spectral density plot shown in Fig. 4c. Clearly, we can see that the original data (top) consists of a higher power in low-frequency regions, whereas the obfuscated dataset (bottom) behaves exactly the opposite consisting of a higher power in high-frequency regions. Nevertheless, as it can be seen from Fig. 4d, the estimated states from these obfuscated dataset are exactly the same as that of the original measurement data. Thus, *Obfuscate*(.) does not degrade the quality of the estimate of the state variables. Furthermore, to evaluate the resilience of *Obfuscate*(.), we estimate the Pearson's correlation coefficient. The Pearson's correlation coefficient gives us the measure of the degree of similarity between two signals. The correlation coefficient between two identical signals in phase is always 1 while two identical signals out of phase (phase difference $= 180\,^{\circ}$) is -1. Figure 3 depicts the plot showing the Pearson correlation coefficient of all the metering points of the 5-bus systems. It can be seen that the correlation between the original and the obfuscated datasets are mostly smaller than 0.2 for almost all the metering points. This implies that it is very hard for any pattern recognition and data mining algorithm to infer information about the original power consumption pattern of the smart meters from the obfuscated datasets (Kim et al. 2011).

Next, we evaluate the degree of obscurity for an IEEE 14 bus system. The **H** matrix for the 14 bus system is extracted from MATPOWER (Zimmerman et al. 2011), an open-source tool for solving steady-state power system simulation and optimization problems.

[2]https://www.keylength.com/en/

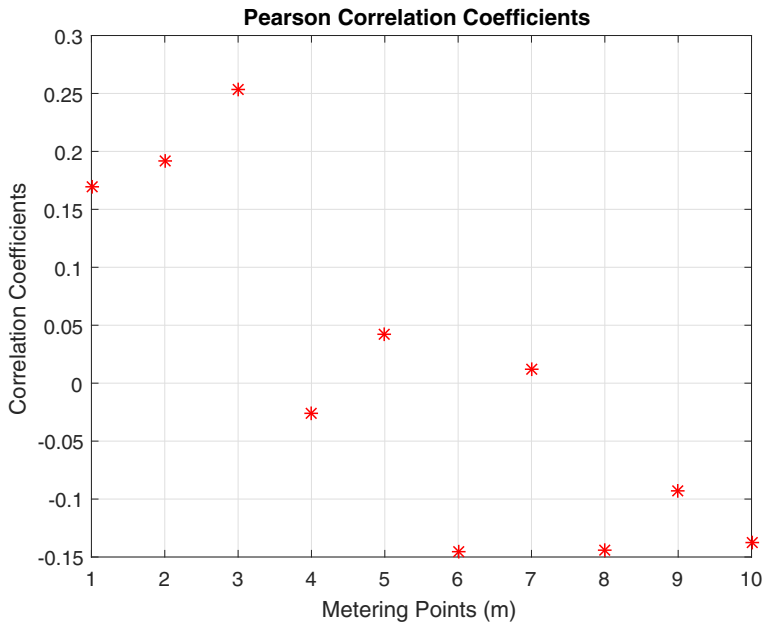

Fig. 3 Pearson correlation coefficients for all the metering points in a 5-bus power network

In this case, the number of metering points $m = 31$ and the number of state variables $n = 13$. We further partition the number of meters and state variables for L_1 and L_2 as $m_1 = 15, m_2 = 16$ and $n_1 = 6, n_2 = 7$. Figure 5 depicts the time domain, frequency domain data and the estimated states from the original and obfuscated measurement data. Comparing Figs. 4 and 5, we arrive at similar conclusions for a 14-bus system to that of a 5-bus system. Figure 6a shows the correlation coefficients of all the 31 metering points for $T = 13$ and it can be seen that the values are lesser than 0.3. Note from Fig. 6b that as expected when the number of measurement data samples is increased i.e., when the value of T was increased from 13 to 360, the correlation coefficient was found to be lesser than 0.2 which makes this scheme practically secure for estimation with fine granular high-frequency meter readings. Also, in this case, since each key size is 96 bits, a semi-honest neighbor trying to infer the power consumption of a household in the same locality has about $2^{96} = 7.92 \times 10^{28}$ possibilities for every batch. Naturally, when the time duration per batch drops down to every few minutes with high-frequency datasets, the task becomes almost impossible for a semi-honest adversary to deduce the appliance usage patterns of his/her neighbor living in the same locality.

However, *Obfuscate*(.) still has a shortcoming since it cannot preserve the privacy of zero elements. The power grid topology matrix $\mathbf{H}$ is, in general, a full column rank and a sparse matrix. However, $\mathbf{H}^+$ is less sparse than $\mathbf{H}$ and is likely to be dense. Upon inspecting the sparsity pattern of $\mathbf{H}^+$ for both the 5-bus and 14-bus power system, we found that $\mathbf{H}^+$ for the 14-bus was about 20% sparse, whereas $\mathbf{H}^+$ for the 5-bus power system was completely dense. Exposing the sparsity pattern of $\mathbf{H}^+$ to the cloud may, in turn, reveal some information about the structure of $\mathbf{H}$ which is undesirable. Thus, to confront such sparse attacks, we introduce the matrix $\mathbf{H}^+_\Delta = \mathbf{H}^+ + \Delta$, where the matrix Δ is 100% dense. The utility provider $\mathcal{U}$ sends $\mathbf{H}^+_\Delta$ instead of $\mathbf{H}^+$ to the cloud which com-

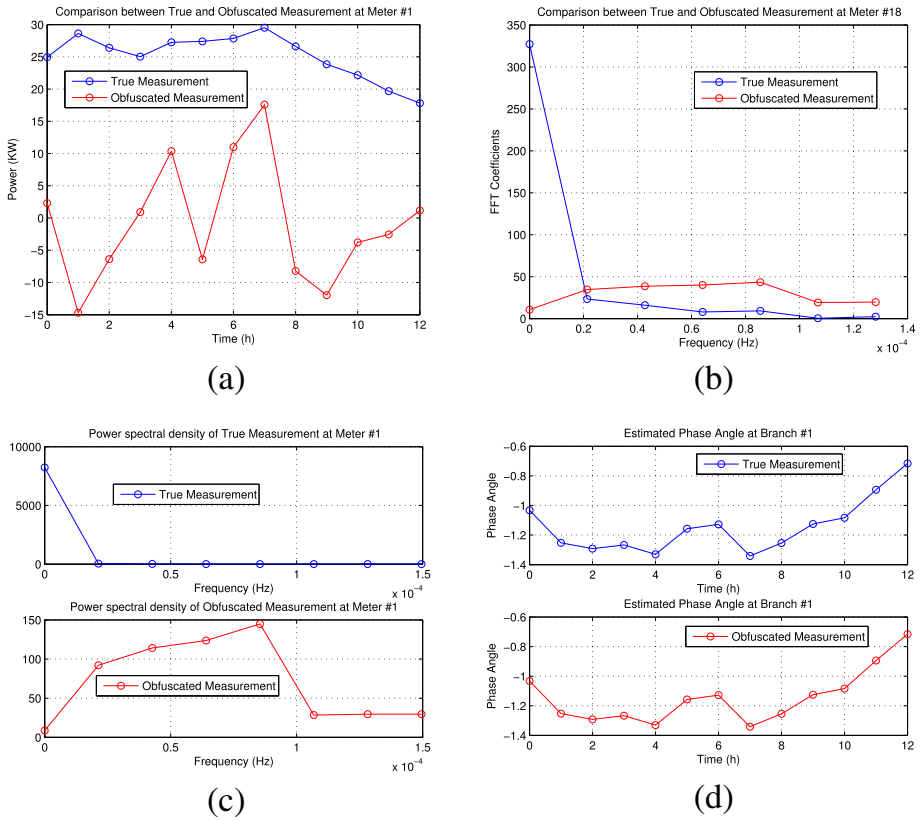

Fig. 4 Illustration of data Obfuscation in a 5-bus power network. **a** Original and Obfuscated Time Domain Data from Meter #1. **b** Original and Obfuscated Frequency Domain Data from Meter #1. **c** Power Spectral Density of True and Obfuscated Measurement Data. **d** Estimate State Value at Branch #1. Estimation error between true and obfuscated dataset = 0

putes $X_\Delta = (\mathbf{H}^+ + \Delta)Z$. Then, $\mathcal{U}$ computes the product ΔZ by invoking *Obfuscate*(.) again. Later, the original state estimates can be retrieved by $\mathcal{U}$ as $\hat{X} = X_\Delta - \Delta Z$. Note that this step does not incur any major computational overhead on the utility provider since it requires another simple invocation of *Obfuscate*(.).

Conclusions and future work

In this paper, we considered a privacy-aware batch-wise state estimation problem in power networks with the objective of protecting both the grid configuration and power consumption data of the smart meters. We formulated a weighted least-squares problem and reduced the state estimation problem of a power grid into a matrix multiplication problem of four block matrices. Our proposal, *Obfuscate*(.), exploits highly efficient and verifiable obfuscation-based cryptographic solutions. It supports error-free estimation between the original and obfuscated dataset without compromising the accuracy of the state variables essential to the utility provider and is proven to be correct and privacy-preserving. Complexity analysis shows the efficiency and the practical applicability of *Obfuscate*(.). We further evaluated the performance of *Obfuscate*(.) in terms of its illegibility and resilience with a real-time hourly power consumption data. Simulation results demonstrate a high level of obscurity making it hard for the malicious cloud server to infer any behavioral pattern from the obfuscated datasets. We also discussed the problem

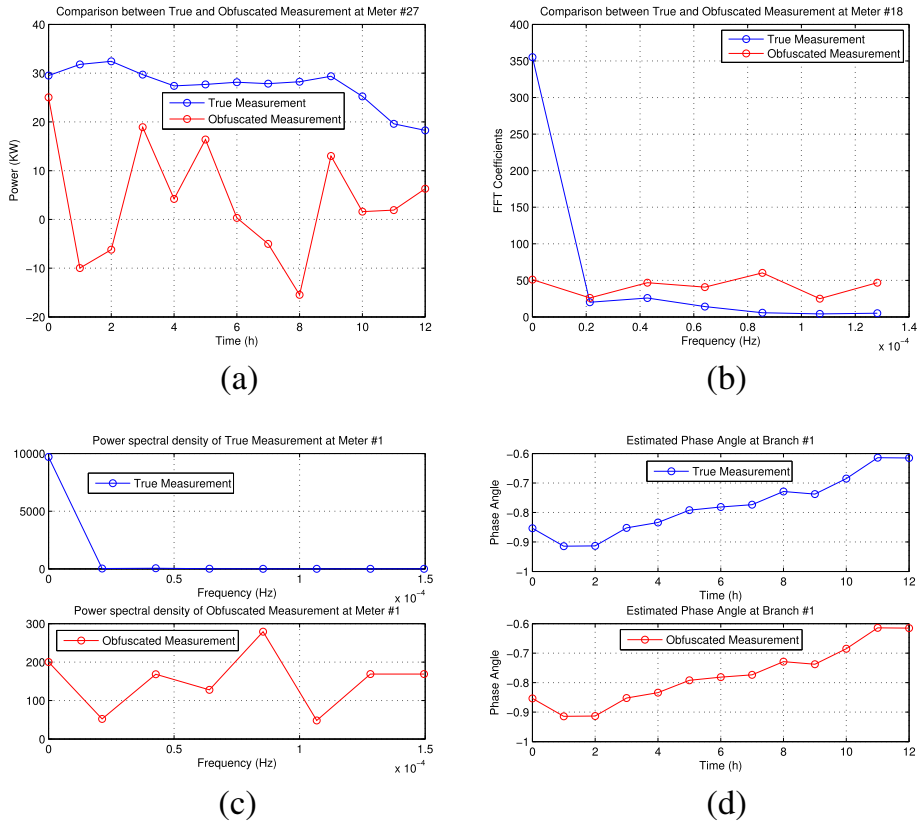

Fig. 5 Illustration of data obfuscation in IEEE 14-bus power network. **a** Original and Obfuscated Time Domain Data from Meter #27. **b** Original and Obfuscated Frequency Domain Data from Meter #27. **c** Power Spectral Density of True and Obfuscated Measurement Data. **d** Estimate State Value at Branch #7. Estimation error between true and obfuscated dataset = 0

of revealing the sparsity structure of the pseudo-inverse of network topology matrix and proposed a solution to resist such sparse attacks.

Currently, our scheme does not take into account that the grid configuration matrix **H**, although time invariant during the state estimation process may still be susceptible to changes all the time. For example, consider a person living in a particular locality is now motivated to install a smart meter at his home due to good security reasons or a person

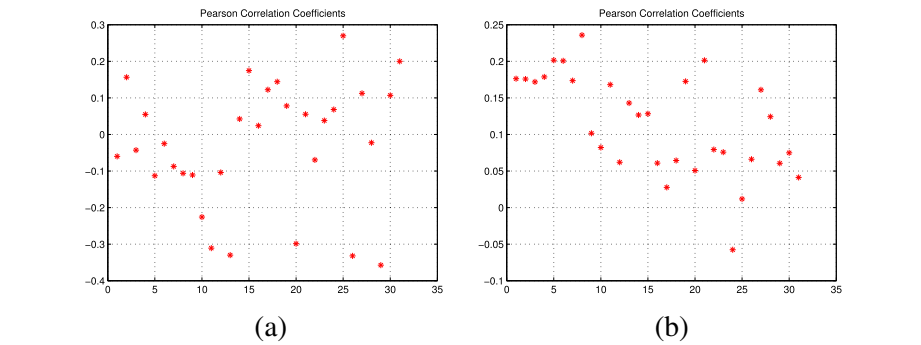

Fig. 6 Pearson Correlation Coefficients of all the metering points in IEEE 14 bus system. **a** $T = 13$. **b** $T = 360$

living in one locality is now moving to another locality. Such situations clearly result in an extra row addition or deletion of the existing **H** matrix, and assuming a pre-computation of $\mathbf{H}^{+}$ at every stage is not reasonable. Hence, to deal with such instances, we require a protocol computing the matrix $A = (\mathbf{H}^T\mathbf{H})^{-1}$ for secure outsourcing of large matrix inversion to the cloud ensuring the privacy of sparsity pattern of the matrix. It is also important to point out that the proposed solution can be applied only to those classes of state estimation which essentially boils down to solving a matrix multiplication problem batch-wise or recursively.

Although the behavioral pattern and the power dynamics of the other smart meters in every locality are hidden from the malicious cloud, the respective lead meter has access to this information. The lead meter can access to the scaled measurements $z' = a_{ij} \cdot z$ (Pearson coefficient = 1) whose dynamics are exactly the same as z. Hence, it was essential in our problem setup to consider a single non-collusive trusted node in every local network termed as the lead meter to initiate the obfuscation of the measurement data dynamics. Future work may involve developing privacy-aware protocols without any such assumptions. Another possible future work is developing a statistical measure to quantify the degree of obscurity introduced by these obfuscation schemes to understand how indistinguishable the obfuscated datasets are compared to the original measurement datasets.

Appendix

A fully measured 5-bus power system is shown in Fig. 7. The total number of meters m is 10 and the meter measurements are $z = [F_{12}, F_{23}, F_{24}, F_{35}, F_{45}, P_1, P_2, P_3, P_4, P_5]^T$ where F_{ij} represents the branch (i, j) active power flow and P_j represents bus j active power injection. The structure of the measurement matrix **H** is given in Eq. 10, where b_{ij} denotes the susceptance of the transmission line (i, j) (Deng et al. 2017). The susceptance is the imaginary part of admittance and the admittance matrix is obtained from (McCalley 2018). The $\mathbf{H}^{+}$ is pre-computed from **H** and the F blocks are partitioned according to their respective dimensions.

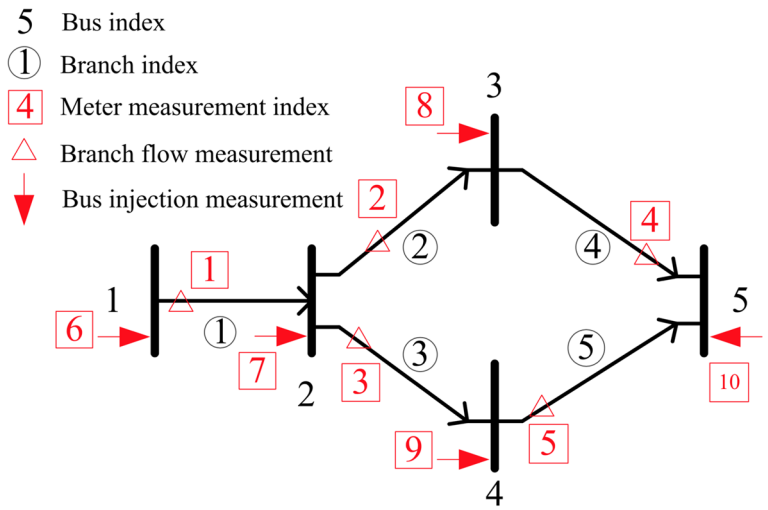

Fig. 7 A fully measured 5-bus power system. Taken from (Deng et al. 2017)

$$H = \begin{pmatrix} b_{12} & 0 & 0 & 0 \\ -b_{23} & b_{23} & 0 & 0 \\ -b_{24} & 0 & b_{24} & 0 \\ 0 & -b_{35} & 0 & b_{35} \\ 0 & 0 & -b_{45} & b_{45} \\ b_{12} & 0 & 0 & 0 \\ -b_{12} - b_{23} - b_{24} & b_{23} & b_{24} & 0 \\ b_{23} & -b_{23} - b_{35} & 0 & b_{35} \\ b_{24} & 0 & -b_{24} - b_{45} & b_{45} \\ 0 & -b_{35} & -b_{45} & b_{35} + b_{45} \end{pmatrix} \tag{10}$$

Acknowledgements
We would also like to thank Antans Sauhatas from Riga Technical University for sharing the real-time power consumption data of the smart meters.

About this supplement
This article has been published as part of?*Energy Informatics*?Volume 2 Supplement 1, 2019: Proceedings of the 8th DACH+ Conference on Energy Informatics. The full contents of the supplement are available online at?https://energyinformatics.springeropen.com/articles/supplements/volume-2-supplement-1.

Authors' contributions
Authors 1, 2, and 4 conceived and conceptualized the presented framework. Author 1 developed the theory, performed the simulations and analyses, and took the lead to write the manuscript. Author 2 helped in drafting the manuscript and in critical revision of the same. Authors 3 and 4 supervised the findings of this work and provided valuable feedback for the final version of the manuscript. All authors read and approved the final manuscript.

Funding
This work was supported by the TU Delft Safety and Security Institute under the DSyS Grant. Publication of this supplement was funded by Austrian Federal Ministry for Transport, Innovation and Technology.

Availability of data and materials
Not applicable.

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

Author details
[1]CGI Nederland B.V, Rotterdam, The Netherlands. [2]Cyber Security Group, Delft University of Technology, Delft, The Netherlands. [3]Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands.

Published: 23 September 2019

References

Atallah MJ, Frikken KB (2010) Securely outsourcing linear algebra computations. In: Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, ASIACCS 2010, Beijing, China, April 13-16, 2010. ACM, New York. pp 48–59

Atallah MJ, Frikken KB, Wang S (2012) Private outsourcing of matrix multiplication over closed semi-rings. In: SECRYPT 2012 - Proceedings of the International Conference on Security and Cryptography, Rome, Italy, 24-27 July, 2012, SECRYPT Is Part of ICETE - The International Joint Conference on e-Business and Telecommunications. SciTePress, Setúbal. pp 136–144

Beussink A, Akkaya K, Senturk IF, Mahmoud M. M. E. A. (2014) Preserving consumer privacy on IEEE 802.11s-based smart grid AMI networks using data obfuscation. In: 2014 Proceedings IEEE INFOCOM Workshops, Toronto, ON, Canada, April 27 - May 2, 2014. IEEE, New York. pp 658–663

Chen F, Dai J, Wang B, Sahu S, Naphade MR, Lu C (2011) Activity analysis based on low sample rate smart meters. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011. ACM, New York. pp 240–248

Commission E (2014a) Benchmarking Smart Metering Deployment in the EU-27 with a Focus on Electricity. https://ses.jrc.ec.europa.eu/publications/reports/benchmarking-smart-metering-deployment-eu-27-focus-electricity

Commission E (2014b) Energy. Smart Grids and Meters. https://ec.europa.eu/energy/en/topics/market-and-consumers/smart-grids-and-meters

Cosovic M, Vukobratovic D (2017) Fast real-time DC state estimation in electric power systems using belief propagation. In: 2017 IEEE International Conference on Smart Grid Communications, SmartGridComm 2017, Dresden, Germany, October 23-27, 2017. IEEE, New York. pp 207–212

Danezis G, Fournet C, Kohlweiss M, Béguelin SZ (2013) Smart meter aggregation via secret-sharing. In: SEGS'13, Proceedings of the 2013 ACM Workshop on Smart Energy Grid Security, Co-located with CCS 2013, November 8, 2013, Berlin, Germany. ACM, New York. pp 75–80

Deng R (2017) Why We Need to Improve Cloud Computing's Security? https://phys.org/news/2017-10-cloud.html
Deng R, Xiao G, Lu R (2017) Defending against false data injection attacks on power system state estimation. IEEE Trans Ind Inform 13(1):198–207
Department of Energy US (2014) Factors Affecting PMU Installation Costs. https://www.smartgrid.gov/files/PMU-cost-study-final-10162014_1.pdf
Dreier J, Kerschbaum F (2011) Practical privacy-preserving multiparty linear programming based on problem transformation. In: PASSAT/SocialCom 2011, Privacy, Security, Risk and Trust (PASSAT), 2011 IEEE Third International Conference on and 2011 IEEE Third International Conference on Social Computing (SocialCom), Boston, MA, USA, 9-11 Oct., 2011. IEEE, New York. pp 916–924
Efthymiou C, Kalogridis G (2010) Smart grid privacy via anonymization of smart metering data. In: 2010 First IEEE International Conference on Smart Grid Communications. IEEE, New York. pp 238–243
Emura K (2017) Privacy-preserving aggregation of time-series data with public verifiability from simple assumptions. In: Information Security and Privacy - 22nd Australasian Conference, ACISP 2017, Auckland, New Zealand, July 3-5, 2017, Proceedings, Part II. Springer, Cham. pp 193–213
Erkin Z (2015) Private data aggregation with groups for smart grids in a dynamic setting using CRT. In: 2015 IEEE International Workshop on Information Forensics and Security, WIFS 2015, Roma, Italy, November 16-19, 2015. IEEE, New York. pp 1–6
Fiore D, Gennaro R (2012) Publicly verifiable delegation of large polynomials and matrix computations, with applications. In: the ACM Conference on Computer and Communications Security, CCS'12, Raleigh, NC, USA, October 16-18, 2012. ACM, New York. pp 501–512
Ge S, Zeng P, Lu R, Choo KR (2018) FGDA: fine-grained data analysis in privacy-preserving smart grid communications. Peer-to-Peer Netw Appl 11(5):966–978
Gentry C, Boneh D (2009) A fully homomorphic encryption scheme. PhD thesis, Stanford University, Stanford vol. 20, no. 09
Gera I, Yakoby Y, Routtenberg T (2017) Blind estimation of states and topology (BEST) in power systems. In: 2017 IEEE Global Conference on Signal and Information Processing, GlobalSIP 2017, Montreal, QC, Canada, November 14-16, 2017. IEEE, New York. pp 1080–1084
Goldwasser S, Kalai YT, Rothblum GN (2015) Delegating computation: Interactive proofs for muggles. J ACM 62(4):27–12764
Huang Y, Werner S, Huang J, Kashyap N, Gupta V (2012) State estimation in electric power grids: Meeting new challenges presented by the requirements of the future grid. IEEE Signal Process Mag 29(5):33–43
Hunt G (2017) What Does GDPR Mean for Your Energy Business? https://www.siliconrepublic.com/enterprise/gdpr-energy-sector
Krause O, Lehnhoff S (2012) Generalized static-state estimation. In: 2012 22nd Australasian Universities Power Engineering Conference (AUPEC). IEEE, New York. pp 1–6
Kumar M, Meena J, Vardhan M (2017) Privacy preserving, verifiable and efficient outsourcing algorithm for matrix multiplication to a malicious cloud server. Cogent Eng 4(1)
Lei X, Liao X, Huang T, Li H, Hu C (2013) Outsourcing large matrix inversion computation to A public cloud. IEEE Trans Cloud Comput 1(1)
Li F, Luo B, Liu P (2010) Secure information aggregation for smart grids using homomorphic encryption. In: 2010 First IEEE International Conference on Smart Grid Communications. IEEE, New York. pp 327–332
Liang G, Zhao J, Luo F, Weller SR, Dong ZY (2017) A review of false data injection attacks against modern power systems. IEEE Trans Smart Grid 8(4):1630–1638
Lindell Y, Pinkas B (2009) Secure multi-party computation for privacy-preserving data mining. J Privacy Confidentiality 1(1):59–98
Lisovich MA, Mulligan DK, Wicker SB (2010) Inferring personal information from demand-response systems. IEEE Secur Priv 8(1):11–20
Liu Y, Ning P, Reiter MK (2011) False data injection attacks against state estimation in electric power grids. ACM Trans Inf Syst Secur 14(1):13–11333
López-Alt A, Tromer E, Vaikuntanathan V (2012) On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In: Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012. ACM, New York. pp 1219–1234
Kim Y, Ngai ECH, Srivastava MB (2011) Cooperative state estimation for preserving privacy of user behaviors in smart grid. In: IEEE Second International Conference on Smart Grid Communications, SmartGridComm 2011, Brussels, Belgium, October 17-20, 2011. IEEE, New York. pp 178–183
Knirsch F, Engel D, Erkin Z (2017) A fault-tolerant and efficient scheme for data aggregation over groups in the smart grid. In: 2017 IEEE Workshop on Information Forensics and Security, WIFS 2017, Rennes, France, December 4-7, 2017. IEEE, New York. pp 1–6
Kursawe K, Danezis G, Kohlweiss M (2011) Privacy-friendly aggregation for the smart-grid. In: Privacy Enhancing Technologies - 11th International Symposium, PETS 2011, Waterloo, ON, Canada, July 27-29, 2011. Proceedings. Springer, Heidelberg. pp 175–191
McCalley JD (2018) The Power Flow Problem. Technical report, Iowa State University. Iowa State University. https://home.engineering.iastate.edu/~jdm/ee553/PowerFlow.doc
Molina-Markham A, Shenoy PJ, Fu K, Cecchet E, Irwin DE (2010) Private memoirs of a smart meter. In: BuildSys'10, Proceedings of the 2nd ACM Workshop on Embedded SensingSystems for Energy-Efficiency in Buildings, Zurich, Switzerland, November 3-5, 2010. ACM, New York. pp 61–66
Monticelli A (2000) Electric power system state estimation. Proc IEEE 88(2):262–282
n.a. (2003) U.S.-Canada Power System Outage Task Force. https://digital.library.unt.edu/ark:/67531/metadc26005/
Rahman MA, Venayagamoorthy GK (2017) Distributed dynamic state estimation for smart grid transmission system. IFAC-PapersOnLine 50(2):98–103
Ren K, Wang C, Wang Q (2012) Security challenges for the public cloud. IEEE Internet Comput 16(1):69–73
Salinas SA, Li P (2016) Privacy-preserving energy theft detection in microgrids: A state estimation approach. IEEE Trans Power Syst 31(2):883–894

Saia J, Zamani M (2015) Recent results in scalable multi-party computation. In: SOFSEM 2015: Theory and Practice of Computer Science - 41st International Conference on Current Trends in Theory and Practice of Computer Science, Pec Pod Sněžkou, Czech Republic, January 24-29, 2015. Proceedings. Springer, Heidelberg. pp 24–44
Schweppe FC (1970) Power system static-state estimation, part III: Implementation. IEEE Trans Power Appar Syst PAS-89(1):130–135
Schweppe FC, Rom DB (1970) Power system static-state estimation, part II: Approximate model. IEEE Trans Power Appar Syst PAS-89(1):125–130
Schweppe FC, Wildes J (1970) Power system static-state estimation, part I: Exact model. IEEE Trans Power Appar Syst PAS-89(1):120–125
Shoukry Y, Gatsis K, Al-Anwar A, Pappas GJ, Seshia SA, Srivastava MB, Tabuada P (2016) Privacy-aware quadratic optimization using partially homomorphic encryption. In: 55th IEEE Conference on Decision and Control, CDC 2016, Las Vegas, NV, USA, December 12-14, 2016. IEEE, New York. pp 5053–5058
Simos M (2017) Microsoft Security Intelligence Report. https://www.microsoft.com/en-us/security/Intelligence-report
Tebaa M, Hajji SE (2014) Secure cloud computing through homomorphic encryption. CoRR abs/1409.0829
Tonyali S, Cakmak O, Akkaya K, Mahmoud MMEA, Güvenç I (2016) Secure data obfuscation scheme to enable privacy-preserving state estimation in smart grid AMI networks. IEEE Internet Things J 3(5):709–719
Wang C, Ren K, Wang J (2011) Secure and practical outsourcing of linear programming in cloud computing. In: INFOCOM 2011. 30th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, 10-15 April 2011, Shanghai, China. IEEE, New York. pp 820–828
Wood AJ, Wollenberg BF (1996) Power Generation, Operation, and Control. Wiley, Hoboken
Zhang Y, Blanton M (2014) Efficient secure and verifiable outsourcing of matrix multiplications. In: Information Security - 17th International Conference, ISC 2014, Hong Kong, China, October 12-14, 2014. Proceedings. Springer, Cham Heidelberg New York. pp 158–178
Zeifman M, Roth K (2011) Nonintrusive appliance load monitoring: Review and outlook. IEEE Trans Consum Electron 57(1):76–84
Zimmerman RD, Murillo-Sánchez CE, Thomas RJ (2009) Matpower's extensible optimal power flow architecture. In: 2009 IEEE Power Energy Society General Meeting. IEEE, New York. pp 1–7
Zimmerman RD, Murillo-Sánchez CE, Thomas RJ (2011) Matpower: Steady-state operations, planning, and analysis tools for power systems research and education. IEEE Trans Power Syst 26(1):12–19

Publisher's Note