A Secure and Privacy-preserving Protocol for Smart Metering Operational Data Collection

Mustafa A. Mustafa, Sara Cleemput, Abdelrahaman Aly, and Aysajan Abidin

***Abstract*—Smart grid allows fine-grained smart metering data collection which can improve the efficiency and reliability of the grid. Unfortunately, this vast collection of data also impose risks to users' privacy. In this paper, we propose a novel protocol that allows suppliers and grid operators to collect users' aggregate metering data in a secure and privacy-preserving manner. We use secure multiparty computation to ensure privacy protection. In addition, we propose three different data aggregation algorithms that offer different balances between privacy-protection and performance. Our protocol is designed for a realistic scenario in which the data need to be sent to different parties, such as grid operators and suppliers. Furthermore, it facilitates an accurate calculation of transmission, distribution and grid balancing fees in a privacy-preserving manner. We also present a security analysis and a performance evaluation of our protocol based on existing multiparty computation algorithms.**

***Index Terms*—Secure Multiparty Computation, Smart Grid, Smart Metering, Renewable Energy Source, Security, Privacy.**

I. Introduction

SMART Grid (SG) is the electrical grid of the future, adding a communication network to the traditional electrical grid infrastructure. This allows bidirectional communication between the different entities and components of the grid, facilitating automated grid management. The overall aim is to make the electrical grid more reliable and efficient [1]. This is achieved by automatically collecting fine-grained metering data from Smart Meters (SMs), which replace the traditional electricity meters. These metering data include electricity consumption and production measurements. Electricity production takes place if households own a Distributed Energy Resource (DER), e.g., solar panels. All these data are sent to the grid operators and suppliers at regular intervals, e.g., 15 minutes.

Access to fine-grained metering data gives entities two main advantages. Firstly, these data allow suppliers to predict their customers' electricity consumption and production more accurately. These patterns are essential to allow the supplier to predict the amount of electricity it needs to buy on the wholesale market for every trading period. Since suppliers pay heavy imbalance fees for every deviation of the actual consumption compared to their prediction, it is crucial for them to obtain accurate consumption and production patterns. Secondly, fine-grained metering data also allow accurate settling of all the fees after each trading period, which is essential to realise local electricity trading markets [2], [3]. Currently, the imbalance fees for the suppliers are calculated proportional to their number of customers in each neighbourhood - i.e., households connected to the same feeder. The current imbalance fee is an estimate. With SM data, accurate settling of fees becomes possible. The same is true for the distribution and transmission fees which suppliers pay to the Distribution Network Operator (DNO) and Transmission System Operator (TSO).

Unfortunately, fine-grained metering data also have disadvantages: they pose a privacy threat to users. Any entity who has access to individual users' fine-grained metering data can use non-intrusive load monitoring techniques [4] to analyse consumption patterns and infer user activities [5]. As a simple example of how such a privacy invasion can lead to adverse consequences, consider an insurance company which increases the insurance fee if they observe from the consumption pattern that their customers do not have a healthy life style. The Netherlands have even abandoned their planned mandatory roll-out of SMs because of the privacy issues [6].

The UK has privacy protection as a requirement for their smart metering architecture [7]. However, their proposed architecture contains a centralised entity, the Data Communications Company (DCC), that collects all metering data and provides a privacy-friendly version of it to authorised entities. Although this might ensure privacy protection against these entities (if the privacy-friendly version is properly generated), it does not protect against the DCC which has access to all users' data.

There are two main approaches for user privacy protection: anonymisation and data aggregation. Proper anonymisation is difficult to achieve, as de-anonymisation is almost always possible [8]. Aggregation is a better approach, but the current proposals [9]–[11] still have shortcomings: (i) they are designed for system models in which data are sent to only one entity, thus they are not applicable to current electricity markets, (ii) they do not consider electricity generated by residential DERs and injected to the grid, and (iii) they do not support transmission, distribution and balancing fee calculation. The real challenge is to design solutions that protect users' privacy not only from external entities but also from internal ones, and that are efficient, fault-tolerant and applicable to real-world smart metering architecture.

In this paper we propose a secure and privacy-preserving protocol for collecting metering data. This work extends our previous research [12] by improving the data aggregation algorithm. Our main contribution is twofold:

- We design a secure and privacy-preserving protocol for collecting operational metering data which is required for calculating distribution, transmission and imbalance fees.

M.A. Mustafa is with the School of Computer Science, The University of Manchester, UK. e-mail: (mustafa.mustafa@manchester.ac.uk).

S. Cleemput, A. Aly and A. Abidin are with the imec-COSIC research group, KU Leuven, Belgium. e-mail: ({sara.cleemput, aysajan.abidin, abdelrahaman.aly}@esat.kuleuven.be).

Manuscript received Xxx xx, 2017; revised Xxx xx, 201x.

Our protocol uses Multiparty Computations (MPC) as the underlying cryptographic primitive and supports three different privacy-friendly data aggregation algorithms. Additionally, it supports realistic system models (with multiple data recipients of aggregates of various subsets of users' metering data); it is fault-tolerant; it is applicable to existing liberalised market models, and it also supports electricity production data generated by users.

- We analyse the computational complexity and communication cost of our protocol in a realistic setting based on the UK's smart metering architectue [7].

The remainder of the paper is organised as follows: Section II discusses the related work, Section III gives the necessary preliminaries, Section IV proposes a protocol (and three aggregation algorithms) for secure and privacy-preserving operational metering data collection. Sections V and VI analyse its security and privacy properties, and evaluate its performance, respectively. Finally, Section VII concludes the paper.

II. Related Work

Security and privacy concerns in SG have been raised [5] and various protocols have already been proposed [9]–[17]. To protect users' privacy, these protocols usually take two approaches: anonymisation or aggregation. Efthymiou and Kalogridis [13] proposed that each SM also has an anonymous ID for reporting only operational metering data. However, Cleemput et al. [8] showed that de-anonymisation is possible.

To achieve privacy-friendly aggregation, Li et al. [10] proposed to use homomorphic encryption. However, their protocol does not protect against active attackers nor facilitate current electricity markets. Mustafa et al. [14], [15] addressed these limitations by using digital signatures and a selective data aggregation and delivery method. Garcia and Jacobs [11] combined homomorphic encryption with data sharing to allow the data recipient to aggregate the data. The use of homomorphic encryption can protect users' privacy, but it also introduces high computational costs to SMs. To overcome this limitation, Kursawe et al. [9] proposed a lightweight aggregation scheme which requires SMs to mask their data with noise that cancels out when added together. The scheme is computationally efficient, but it requires a complex reinitialization process when adding SMs and does not support flexible aggregation groups. Rahman et al. [18] proposed a distributed aggregation of metering data where an initialisation SM adds random value to its metering data before sending the sum to the next SM in the group. In turns, each SM adds their data to the accumulated sum before the value is returned to the initial SM that extracts the beforehand added random value to obtain the aggregate data of all the SMs. However, the data of a SM could easily be learnt if two of the neighbouring SMs in the ring collaborate.

Gope and Sikdar [19] proposed a privacy-preserving aggregation scheme for billing and demand response management, named EDAS, that uses random values to mask the individual measurements of SMs. However, their scheme only partially protects users' privacy, i.e., the service provider is fully trusted, thus it learns all the users metering data, and the aggregator also learns the aggregate data of users located in the same region. Liu et al. [20] proposed a data aggregation scheme, named 3PDA, that uses a virtual aggregation area to mask the metering data of individual users. Although 3PDA does not rely on a trusted party to aggregate the data, its system model is simplified (it has only a service provider, aggregator and SMs), thus not practical to deploy in existing grid architecture. Lyu et al. [21] proposed a fog-enabled scheme, named PPFA. PPFA uses nodes that are closer to SMs to perform an initial aggregation of the data, and then the service provider uses the aggregate data provided by the fog nodes to extract statistics about the users' metering data. Although the obtained results protect users' privacy, i.e., they are differentially private, such results can not be used for billing and settlements as they are just (close) estimations of the real consumption.

He et al. [22] proposed a privacy-preserving aggregation scheme that also protects against internal attackers. However, their scheme uses bilinear mapping, which is computationally expensive for devices with limited resources. He et al. [23], [24] proposed modified schemes that outperform [22] in terms of computation and communication costs. Knirsch et al. [25] combined the schemes proposed in [26] and [27] to construct a scheme for privacy-friendly fault-tolerant aggregation over multiple sets of SMs. However, the scheme is not efficient, as it uses computationally expensive homomorphic techniques as well as a full round between SMs and the service provider. Li et al. [28] proposed a privacy-preserving multi-subset data aggregation scheme, named PPMA. Their scheme combines two super-increasing sequences (similar to EPPA [29]) and Paillier cryptosystem. However, the SMs of each subset are restricted to the ones whose consumption data is within a predefined range, limiting the usefulness of the aggregate data.

Knirsch et al. [30] use masking to achieve error-resilient data aggregation. Although their scheme supports aggregation over multiple sets of SMs, it adds additional communication costs as SMs communicate between each other to exchange their masking values. Abdallah and Shen [31], [32] proposed a lattice-based homomorphic data aggregation scheme, whereas Shen et al. [33] proposed a cube-data aggregation scheme that allows aggregation of multi-dimensional data to obtain sums of each dimension without revealing users' private information. However, these schemes do not support aggregation of multiple sets of SMs. Borges and Mühlhäuser [34] proposed a homomorphic encryption-based protocol, where each SM has two secret keys, and an authorised data recipient is aware of the sums of all the corresponding keys of all the SMs in the network. Each SM encrypts its metering data using its keys and sends the ciphertext directly to the data recipient. The data recipient aggregates all the ciphertexts and then using the sums of the keys, it can decrypt only the sum of the SMs' metering data. The protocol has lower computational costs at SMs (the costs of encryption), compared to the original Paillier scheme. However, the decryption cost at the data recipient is high. Also, the communication overheads are high, as each SM sends its data to the data recipient, i.e., the aggregation is done at the data recipient (the destination of the data), not in the network. Tonyali et al. [35] proposed mechanisms to reduce communication overheads when somewhat homomorphic encryption is used to perform in-network data aggregation.

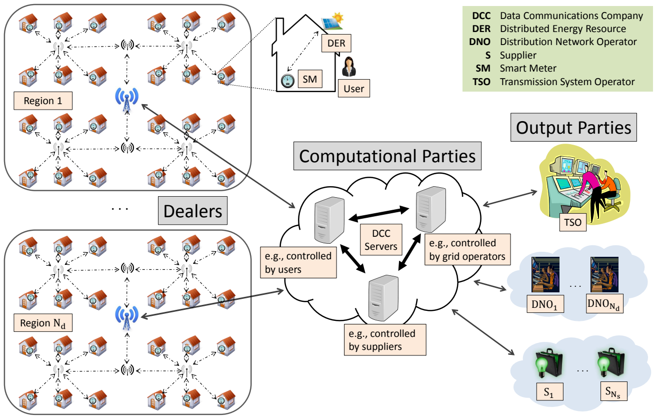

DCC Data Communications Company
DER Distributed Energy Resource
DNO Distribution Network Operator
S Supplier
SM Smart Meter
TSO Transmission System Operator
DER
SM
User
Region 1
Region N$_d$
Dealers
Computational Parties
Output Parties
DCC Servers
e.g., controlled by users
e.g., controlled by grid operators
e.g., controlled by suppliers
TSO
DNO$_1$
DNO$_{N_d}$
S$_1$
S$_{N_s}$

Fig. 1. System model.

Another approach to aggregate data in a privacy-preserving (and efficient) manner is MPC. Danezis et al. [16] proposed protocols using secret-sharing based MPC to detect fraud and to extract advanced grid statistics. Rottondi et al. [17] proposed a novel security architecture for aggregation of metering data. However, their architecture requires additional nodes in the system, i.e., gateways placed at the users' households.

Unlike the aforementioned work, our proposed MPC-based privacy-preserving protocol for operational metering data collection (i) is based on a real smart metering architecture, (ii) is readily applicable to a liberalised electricity market with various stakeholders, (iii) takes into account not only the electricity consumption data, but also electricity injected into the grid by households, and (iv) allows the TSO, DNOs and suppliers to calculate the exact distribution, transmission and balancing fees based on real data rather than on estimates. Furthermore, our protocol is fault-tolerant and it protects users' privacy against (colluding) internal adversaries.

III. Preliminaries

A. System Model

As shown in Fig. 1, our system model follows the smart metering architecture used in the UK [36], and consists of the following entities. *Users* consume electricity and are billed for this by their contracted supplier. *Distributed Energy Resources* (*DERs*) are mini generators (e.g., solar panels) located on users' premises. Most of the electricity they generate is consumed by their owners. However, surplus electricity may be injected into the grid. *Smart Meters* (*SMs*) are advanced metering devices that measure the electricity flowing from the grid to the house and vice versa per time slot, t_k. SMs regularly communicate with other authorised SG entities. *Suppliers* are responsible for supplying electricity to all users. They buy this electricity from generators on the wholesale market, and sell it to users. They are also obliged to buy any electricity their customers inject into the grid. If a supplier buys an incorrect amount of electricity, it will be punished with imbalance fees. *Distribution Network Operators* (*DNOs*) are responsible for managing and maintaining the electricity distribution lines (i.e., the low/middle voltage lines) in their respective regions. They charge suppliers distribution fees based on the electricity consumption of the suppliers' customers in each time slot. The suppliers then charge their customers this fee in turn. *Transmission System Operator* (*TSO*) is responsible for managing and maintaining the electricity transmission lines (i.e., the high voltage lines) in the grid as well as balancing the whole grid at any point in time. It charges suppliers transmission and balancing fees based on the consumption of their customers in each time slot. Similarly, suppliers pass this cost to users. *Data Communications Company* (*DCC*) is a centralised entity that consists of servers run by different parties. It collects and delivers users' metering data to the TSO, DNOs and suppliers.

We also classify some of these entities into three groups: *dealers* (i.e., SMs) who provide the input data, *computational parties* (i.e., DCC servers) who perform computations on the input data, and *output parties* (i.e., TSO, DNOs, and suppliers) who receive the results of the computations. The SMs generate and provide the DCC servers with input data including the electricity consumption and generation data measured per time slot. The DCC servers obtain input data from the SMs, jointly perform the necessary calculations and provide the TSO, DNOs and suppliers with the results of these calculations.

B. Threat Model and Assumptions

All external and internal entities (i.e., users, DCC servers, TSO, DNOs and suppliers) can act maliciously. External entities may eavesdrop or modify data in transit trying to gain access to confidential data or to disrupt the SG. Any user may try to modify metering data sent by their SMs in an attempt to gain financial advantage or learn other users' data. The TSO, DNOs and suppliers may manipulate users' metering data in an attempt to gain financial advantage, i.e., to manipulate the transmission/distribution fees and imbalance fines calculations. They may also try to learn individual users' consumption data or the aggregate consumption of any group of users located in different regions or contracted by competitors. Any DCC server may manipulate the metering data it receives/computes.

As our main concern is privacy protection of users, we make the following realistic assumptions. The communication channels among all entities are encrypted and authenticated. At least one DCC server is honest and trustworthy, i.e., it follows the protocol specifications. SMs are tamper-proof and sealed, thus no one can tamper with them without being detected. This is a common assumption in the community. If an SM is faulty or compromised (sending false data), there are standard techniques (e.g., use control meters) to trace back and identify this meter. However, this is out of the scope of this paper.

C. Notations

We denote the SM of household i as $\mathrm{SM}_i \in \mathbb{SM}$, where $\mathbb{SM}$ is the set of all the SMs in the grid of a country, and the amount of electricity taken from the grid (i.e., imported electricity) and the amount fed back to the grid (i.e., exported electricity) by household i during the kth time slot, t_k, as $\mathrm{E}_i^{\mathrm{imp},t_k} \in \mathbb{E}^{\mathrm{imp},t_k}$ and $\mathrm{E}_i^{\mathrm{exp},t_k} \in \mathbb{E}^{\mathrm{exp},t_k}$, respectively. $\mathbb{E}^{\mathrm{exp},t_k}$ and $\mathbb{E}^{\mathrm{exp},t_k}$ are the aggregate of consumption data and electricity fed back data, respectively, of all the households during t_k in the grid. More notations are given in Table I. Also, $\sum(X)$ denotes the aggregate (sum) value of all elements in set X.

TABLE I
NOTATIONS

Symbol	Meaning
t_k	kth time slot, $k = 1, \ldots, \mathrm{N_t}$
d_j	the DNO operating in region j, $j = 1, \ldots, \mathrm{N_d}$
s_u	uth supplier, $u = 1, \ldots, \mathrm{N_s}$
SM_i	the SM belonging to household i
$\mathbb{SM}$	set of all the SMs in a specific country
$\mathbb{SM}_{\mathrm{d}_j}$	set of all the SMs operated by DNO d_j
$\mathbb{SM}^{\mathrm{imp}}_{\mathrm{s}_u}$	set of all the SMs whose users buy electricity from s_u
$\mathbb{SM}^{\mathrm{exp}}_{\mathrm{s}_u}$	set of all the SMs whose users sell electricity to s_u
$\mathbb{SM}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}$	set of all the SMs operated by d_j and whose users buy electricity from s_u
$\mathbb{SM}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u}$	set of all the SMs operated by d_j whose users sell electricity to s_u
$\mathrm{E}^{\mathrm{imp},t_k}_i$	amount of electricity imported by household i during t_k
$\mathrm{E}^{\mathrm{exp},t_k}_i$	amount of electricity exported by household i during t_k
$\mathbb{E}^{\mathrm{imp},t_k}$	aggregate data of all $\mathrm{E}^{\mathrm{imp},t_k}_i$ for $\mathrm{SM}_i \in \mathbb{SM}$
$\mathbb{E}^{\mathrm{exp},t_k}$	aggregate data of all $\mathrm{E}^{\mathrm{exp},t_k}_i$ for $\mathrm{SM}_i \in \mathbb{SM}$
$\mathbb{E}^{\mathrm{imp},t_k}_{\mathrm{d}_j}$	aggregate data of all $\mathrm{E}^{\mathrm{imp},t_k}_i$ for $\mathrm{SM}_i \in \mathbb{SM}_{\mathrm{d}_j}$
$\mathbb{E}^{\mathrm{exp},t_k}_{\mathrm{d}_j}$	aggregate data of all $\mathrm{E}^{\mathrm{exp},t_k}_i$ for $\mathrm{SM}_i \in \mathbb{SM}_{\mathrm{d}_j}$
$\mathbb{E}^{\mathrm{imp},t_k}_{\mathrm{s}_u}$	aggregate data of all $\mathrm{E}^{\mathrm{imp},t_k}_i$ for $\mathrm{SM}_i \in \mathbb{SM}^{\mathrm{imp}}_{\mathrm{s}_u}$
$\mathbb{E}^{\mathrm{exp},t_k}_{\mathrm{s}_u}$	aggregate data of all $\mathrm{E}^{\mathrm{exp},t_k}_i$ for $\mathrm{SM}_i \in \mathbb{SM}^{\mathrm{exp}}_{\mathrm{s}_u}$
$\mathbb{E}^{\mathrm{imp},t_k}_{\mathrm{d}_j,\mathrm{s}_u}$	aggregate data of all $\mathrm{E}^{\mathrm{imp},t_k}_i$ for $\mathrm{SM}_i \in \mathbb{SM}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}$
$\mathbb{E}^{\mathrm{exp},t_k}_{\mathrm{d}_j,\mathrm{s}_u}$	aggregate data of all $\mathrm{E}^{\mathrm{exp},t_k}_i$ for $\mathrm{SM}_i \in \mathbb{SM}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u}$

D. *Design Requirements*

1) *Functional Requirements:*

(F1) For each time period t_k, each DNO d_j should access:

a) $\mathbb{E}^{\mathrm{imp},t_k}_{\mathrm{d}_j}$ and $\mathbb{E}^{\mathrm{exp},t_k}_{\mathrm{d}_j}$, in order to better manage the distribution network in its region,

b) $\mathbb{E}^{\mathrm{imp},t_k}_{\mathrm{d}_j,\mathrm{s}_u}$ and $\mathbb{E}^{\mathrm{exp},t_k}_{\mathrm{d}_j,\mathrm{s}_u}$, for $u = 1, \ldots, \mathrm{N_s}$, in order to split the distribution fees fairly among the suppliers.

(F2) For each time period t_k, each supplier s_u should access:

a) $\mathbb{E}^{\mathrm{imp},t_k}_{\mathrm{s}_u}$ and $\mathbb{E}^{\mathrm{exp},t_k}_{\mathrm{s}_u}$, in order to predict its customers' electricity consumption and production accurately, so that it can avoid receiving imbalance fines,

b) $\mathbb{E}^{\mathrm{imp},t_k}_{\mathrm{d}_j,\mathrm{s}_u}$ and $\mathbb{E}^{\mathrm{exp},t_k}_{\mathrm{d}_j,\mathrm{s}_u}$ for $j = 1, \ldots, \mathrm{N_d}$, so it can be assured that it pays the correct transmission and distribution network fees to the TSO and each DNO, respectively. Transmission network fees can also be made region-dependent to encourage suppliers to buy electricity from sources located closer to the demand.

(F3) For each time period t_k, the TSO should access:

a) $\mathbb{E}^{\mathrm{imp},t_k}_{\mathrm{d}_j,\mathrm{s}_u}$ and $\mathbb{E}^{\mathrm{exp},t_k}_{\mathrm{d}_j,\mathrm{s}_u}$, for $u = 1, \ldots, \mathrm{N_s}$, so it can split transmission network fees among suppliers,

b) $\mathbb{E}^{\mathrm{imp},t_k}_{\mathrm{s}_u}$ and $\mathbb{E}^{\mathrm{exp},t_k}_{\mathrm{s}_u}$, for $u = 1, \ldots, \mathrm{N_s}$, so it can calculate the imbalance fine for each supplier,

c) $\mathbb{E}^{\mathrm{imp},t_k}_{\mathrm{d}_j}$ and $\mathbb{E}^{\mathrm{exp},t_k}_{\mathrm{d}_j}$, for $j = 1, \ldots, \mathrm{N_d}$, to identify the regions that cause imbalance, thus to decide which measures to take to avoid the imbalance, and

d) $\mathbb{E}^{\mathrm{imp},t_k}$ and $\mathbb{E}^{\mathrm{exp},t_k}$, to balance the grid efficiently.

2) *Security and Privacy Requirements:*

(S1) Privacy of users: individual users' fine-grained metering data should not be revealed to any in(ex-)ternal SG entity.

(S2) Confidentiality of users' data: only the authorised entities, i.e., TSO, DNO and suppliers, should have access only to the aggregates of users' metering data.

(S3) Authorisation: SG entities should be allowed to access the aggregate data only of the users whom they provide services to. For the DNO this means only the users living in the region it operates, for the supplier this means only the users who have a contract with it.

(S4) Fault tolerance: (Partial) loss of metering data should still allow computation of aggregates of the data.

Algorithm 1: Generic Equality Test

```
Input: Secret share bit representation of $x$, $[x]_1, \ldots, [x]_\sigma$
       Bit representation $y_1, \ldots, y_\sigma$ of public scalar $y$ to which $x$ is compared
Output: A secret share of the output of the equality test [c]
$[c] \leftarrow 0$;
for $i \leftarrow 1$ to $\sigma$ do
    $[c'] \leftarrow [x]_i + y_i - 2 \cdot ([x]_i \cdot y_i)$;
    $[c] \leftarrow [c] + [c'] - [c] \cdot [c']$;
end
```

E. *Cryptographic Notation*

Security of MPC protocols is typically analysed in the Universally Composable (UC) framework [37], under which, the ideal functionality of MPC is modelled as an Arithmetic Black Box (ABB). ABB can be seen as a generic procedure for secure computation, where any party can send its private input to ABB and ask it to calculate any computable function. The functions are represented as arithmetic circuits comprising additions, multiplications, permutations, equality tests, etc. As long as the circuit components are UC secure, the UC framework guarantees that the circuit can be executed securely. Our protocol uses *equality test* and *permutation* as components.

Equality test can be implemented in oblivious fashion by using just multiplications and additions. Any existing test [38], [39] is suitable for use. To simplify it, SMs could share their ID in its bit representation. This way the bit-wise comparison would require only σ multiplications, where σ is the bit length of the suppliers' ID. Algorithm 1 illustrates this. It can be optimised by parallelising the computation of multiplications so that only $\log(\sigma)$ communication rounds are needed.

Oblivious permutation can be achieved by using an $n \times n$ Boolean permutation matrix, where n is the size of the input to be permuted. Under this approach each entry of the input is multiplied against a corresponding matrix column, and the results are aggregated. This method has $\mathcal{O}(n^2)$ complexity. Other approaches (e.g., use of sorting networks) can achieve better asymptotic complexity. However, using a pre-computed permutation network this can be achieved in (almost) $\mathcal{O}(n \log(n))$ complexity [40]. Such approaches have been adapted for practical use in electricity markets [41].

We assume that all secretly shared values are members of a field $\mathbb{Z}_p$ bounded by a sufficiently large prime p, such that no overflow occurs. If fixed point precision is needed, the entries can be multiplied with a large enough constant such that they can be shared as elements of $\mathbb{Z}_p$.

IV. PRIVACY-PRESERVING SMART METERING PROTOCOL

In this section we propose a privacy-preserving MPC-based protocol for operational metering data collection. We give an overview of the protocol, and then propose three aggregation algorithms that offer different privacy/performance trade-offs.

Algorithm 2: Naïve Aggregation Algorithm (NAA)

```
Input: Tuples from region $j$, $\{[\mathrm{s}_u^{\mathrm{imp}}], [\mathrm{s}_u^{\mathrm{exp}}], [\mathrm{E}_i^{\mathrm{imp}}], [\mathrm{E}_i^{\mathrm{exp}}]\}$ for $\mathrm{SM}_i \in \mathbb{SM}_{\mathrm{d}_j}$
Output: Shares of aggregate consumption data per supplier, $[\mathbb{E}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}]$
        Shares of aggregate production data per supplier, $[\mathbb{E}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u}]$
$[\mathbb{E}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}] \leftarrow \{0_1, ..., 0_{\mathrm{N_s}}\}$;
$[\mathbb{E}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u}] \leftarrow \{0_1, ..., 0_{\mathrm{N_s}}\}$;
for $i \leftarrow 1$ to $|\mathbb{SM}_{\mathrm{d}_j}|$ do
    for $u \leftarrow 1$ to $\mathrm{N_s}$ do
        $[c] \leftarrow [\mathrm{s}_u^{\mathrm{imp}}] \overset{?}{=} \mathrm{s}_u$;
        $[\mathbb{E}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}] \leftarrow [\mathbb{E}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}] + [c] * [\mathrm{E}_i^{\mathrm{imp}}]$;
    end
    for $u \leftarrow 1$ to $\mathrm{N_s}$ do
        $[c] \leftarrow [\mathrm{s}_u^{\mathrm{exp}}] \overset{?}{=} \mathrm{s}_u$;
        $[\mathbb{E}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u}] \leftarrow [\mathbb{E}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u}] + [c] * [\mathrm{E}_i^{\mathrm{exp}}]$;
    end
end
```

A. Overview of our Generic Protocol

The generic protocol consists of the following four steps.

1) Input data generation and distribution: Each SM generates three data tuples, each containing different shares of the user's contracted suppliers, consumption and generation data, and sends them to the corresponding computational parties.

2) Region-based data aggregation: Once the input data of all the SMs are received, the computational parties aggregate the consumption and generation data for each region using one of the three aggregation algorithms described below. The output is in shared form and represents the region-based aggregate consumption and generation data per supplier.

3) Grid-based data aggregation: The computational parties compute the shares of all the grid-based aggregate consumption and generation data by simply adding the corresponding shares of the region-based aggregate data.

4) Output data distribution: Following the functional requirements specified in Section III, the shares of the previously calculated aggregations are distributed to the TSO, DNOs and suppliers, accordingly. Finally, these entities reconstruct their required results by reconstructing the corresponding shares.

B. Region-based Data Aggregation Algorithms

In this section, we present our three region-based data aggregation algorithms that offer different trade-offs in terms of security, flexibility and performance.

1) Naïve Aggregation Algorithm (NAA): A naïve approach to perform data aggregation with perfect privacy would be to implement a basic circuit that uses equality tests to identify users' suppliers. As shown in Algorithm 2, SMs send their tuples $\{[\mathrm{s}_u^{\mathrm{imp}}], [\mathrm{s}_u^{\mathrm{exp}}], [\mathrm{E}_i^{\mathrm{imp}}], [\mathrm{E}_i^{\mathrm{exp}}]\}$ to the DCC servers, so that the servers can classify the inputs by using oblivious comparisons. Although the algorithm is fairly adaptive to a growing number of suppliers, denoted as $\mathrm{N_s}$, it is expensive in terms of performance as it still requires $\mathcal{O}(|\mathbb{SM}_{\mathrm{d}_j}| \cdot \mathrm{N_s})$ equality tests, where $|\mathbb{SM}_{\mathrm{d}_j}|$ is the number of SMs in a given region j.

2) No Comparison Aggregation Algorithm (NCAA): To improve the performance of the aggregation algorithm, some level of disclosure to the DCC servers can be allowed, in this case, the number of users linked to each supplier. As shown in Algorithm 3, the DCC servers permute the tuples corresponding to the same region and aggregate them in a non-interactive way afterwards. Considering that its complexity is dominated by the oblivious permutation calls, NCAA multiplication bound is $\mathcal{O}(|\mathbb{SM}_{\mathrm{d}_j}| \cdot \log(|\mathbb{SM}_{\mathrm{d}_j}|)$. Also, NCAA keeps its flexibility with respect to $\mathrm{N_s}$ at the cost of disclosing the number of SMs associated to each supplier.

Algorithm 3: No Comparison Aggr. Algorithm (NCAA)

```
Input: Tuples from region $j$, $\{[\mathrm{s}_u^{\mathrm{imp}}], [\mathrm{s}_u^{\mathrm{exp}}], [\mathrm{E}_i^{\mathrm{imp}}], [\mathrm{E}_i^{\mathrm{exp}}]\}$ for $\mathrm{SM}_i \in \mathbb{SM}_{\mathrm{d}_j}$
Output: Shares of aggregate consumption data per supplier, $[\mathbb{E}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}]$
        Shares of aggregate production data per supplier, $[\mathbb{E}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u}]$
$[\mathbb{E}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}] \leftarrow \{0_1, ..., 0_{\mathrm{N_s}}\}$;
$[\mathbb{E}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u}] \leftarrow \{0_1, ..., 0_{\mathrm{N_s}}\}$;
$[\mathbb{SM}'_{\mathrm{d}_j}] \leftarrow \mathtt{permute}([\mathbb{SM}_{\mathrm{d}_j}])$;
for $i \leftarrow 1$ to $|\mathbb{SM}'_{\mathrm{d}_j}|$ do
    $\mathrm{s}_u^{\mathrm{imp}} \leftarrow \mathtt{open}([\mathrm{s}_u^{\mathrm{imp}}])$;
    for $u \leftarrow 1$ to $\mathrm{N_s}$ do
        $c \leftarrow \mathrm{s}_u^{\mathrm{imp}} == \mathrm{s}_u$;
        $[\mathbb{E}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}] \leftarrow [\mathbb{E}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}] + c * [\mathrm{E}_i^{\mathrm{imp}}]$;
    end
end
$[\mathbb{SM}'_{\mathrm{d}_j}] \leftarrow \mathtt{permute}([\mathbb{SM}_{\mathrm{d}_j}])$;
for $i \leftarrow 1$ to $|\mathbb{SM}'_{\mathrm{d}_j}|$ do
    $\mathrm{s}_u^{\mathrm{exp}} \leftarrow \mathtt{open}([\mathrm{s}_u^{\mathrm{exp}}])$;
    for $u \leftarrow 1$ to $\mathrm{N_s}$ do
        $c \leftarrow \mathrm{s}_u^{\mathrm{exp}} == \mathrm{s}_u$;
        $[\mathbb{E}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u}] \leftarrow [\mathbb{E}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u}] + c * [\mathrm{E}_i^{\mathrm{exp}}]$;
    end
end
```

3) Non-Interactive Aggregation Algorithm (NIAA): To further improve the performance of the aggregation algorithm, the input data of SMs can be tweaked such that the aggregation could be done without the need of communication between the DCC servers. To achieve this, SMs have to encode their input data into vectors of all zeros but one unique non-zero entry. These vectors are of size $\mathrm{N_s}$ and the non-zero entries are their $\mathrm{E}^{\mathrm{imp}}$ and $\mathrm{E}^{\mathrm{exp}}$, respectively. This way the DCC servers only need to process the aggregation of the shares, which is non-interactive for any generalized Linear Secret Sharing Scheme (LSSS). By reducing the flexibility ($\mathrm{N_s}$ has to be fixed), NIAA, as shown in Algorithm 4, is implemented with neither comparison nor multiplication operations. To support the addition of a new supplier, SMs will have to use a vector with a sufficiently large pre-fixed size, providing 0 for the non-used slots, so that the system is flexible in accommodating a large number of suppliers. An easy alternative would be to

Algorithm 4: Non-Interactive Aggr. Algorithm (NIAA)

```
Input: Tuples from region $j$, $\{[\mathbf{E}_i^{\mathrm{imp}}], [\mathbf{E}_i^{\mathrm{exp}}]\}$ for $\mathrm{SM}_i \in \mathbb{SM}_{\mathrm{d}_j}$, where $\mathbf{E}_i^{\mathrm{imp}}$
       and $\mathbf{E}_i^{\mathrm{exp}}$ are vectors of size $\mathrm{N_s}$ with only one non-zero entry at position $u$
Output: Shares of aggregate consumption data per supplier, $[\mathbb{E}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}]$
        Shares of aggregate production data per supplier, $[\mathbb{E}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u}]$
$[\mathbb{E}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}] \leftarrow \{0_1, ..., 0_{\mathrm{N_s}}\}$;
$[\mathbb{E}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u}] \leftarrow \{0_1, ..., 0_{\mathrm{N_s}}\}$;
for $i \leftarrow 1$ to $|\mathbb{SM}_{\mathrm{d}_j}|$ do
    for $u \leftarrow 1$ to $\mathrm{N_s}$ do
        $[\mathbb{E}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}] \leftarrow [\mathbb{E}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}] + [\mathrm{E}^{\mathrm{imp}}_{i,u}]$;
        $[\mathbb{E}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u}] \leftarrow [\mathbb{E}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u}] + [\mathrm{E}^{\mathrm{exp}}_{i,u}]$;
    end
end
```

allow the system to feed (via an update) all the SMs with a parameter – the number of suppliers – so that SMs will encode their inputs as vectors of correct length. Moreover, the supplier ID position has to be agreed in advance. NIAA also produces no leakage, hence it achieves perfect security.

V. Security Analysis

We analyse the security of our protocol in the Universal Composability (UC) framework [37], in which the ideal functionality of MPC is modelled as Arithmetic Black Box (ABB) [42]. ABB is a generic procedure for MPC providing an abstraction of the details of MPC operations and of secret sharing. Players can send their private input to ABB to compute any computable function on their private inputs. The results from the computation are stored in the internal state of ABB to be used in the subsequent computations. Stored values are only made public if enough number of players agree. Formally, ABB is defined as follows.

Definition 1 (ABB Functionality $\mathcal{F}_{ABB}$)**.** *The ideal functionality $\mathcal{F}_{ABB}$ for MPC is defined as follows:*

- `input`*: Receive a value $\alpha \in \mathbb{Z}_M$ and store α.*
- `share`(α)*: Create a share $[\alpha]$ of α.*
- `product`$([\alpha],[\beta])$*: Compute $\gamma = \alpha \times \beta$ and store $[\gamma]$.*
- `compare`$([\alpha],[\beta])$*: Compare α and β, and return 0 if $\alpha < \beta$ and 1 otherwise.*
- `equal`$([\alpha],[\beta])$*: Check if $\alpha = \beta$; return 1 if $\alpha = \beta$, 0 otherwise.*
- `permute`$([\mathtt{X}])$*: Given an input $[X] \in \mathbb{Z}_M^n$ return a random permutation of it.*
- `open`$([\alpha])$*: Send the value α to all players.*

Addition and scalar multiplication are denoted by their corresponding conventional symbols $+$ and $\times$.

Definition 2 (UC-security [43])**.** *A real protocol π is UC-secure if, for any adversary $\mathcal{A}$, there exists a simulator $\mathcal{S}$ for which no environment $\mathcal{Z}$ can distinguish with a non-negligible probability if it is interacting with $\mathcal{A}$ and π or S and the ideal functionality $\mathcal{F}$.*

Definition 3 (Universal Composition [43])**.** *Let π and ρ be two protocols such that ρ ϵ_1-UC-emulates $\mathcal{G}$ and $\pi \circ \mathcal{G}$ ϵ_2-UC-emulates $\mathcal{F}$, when using $\mathcal{G}$ as a subroutine. Then $\pi \circ \rho$ $(\epsilon_1 + \epsilon_2)$-UC-emulates $\mathcal{F}$, when using ρ as a subroutine.*

We define the ideal functionality of the operation metering data collection protocol as follows.

Definition 4 (Ideal Functionality of Operational Metering Data Collection Protocol $\mathcal{F}_{OMDCP}$)**.** *Given a set of input tuples $\{\mathrm{s}_u^{\mathrm{imp}}, \mathrm{s}_u^{\mathrm{exp}}, \mathrm{E}_i^{\mathrm{imp}}, \mathrm{E}_i^{\mathrm{exp}}\}$ from SMs, the ideal functionality $\mathcal{F}_{OMDCP}$ computes the aggregate consumption data $\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{imp}}$ and aggregate production data $\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{exp}}$ per supplier, and returns the results to the relevant parties (i.e., suppliers, DNOs, and TSO).*

Next, we show that the three region-based data aggregation algorithms – NAA, NCAA, and NIAA – securely realise the ideal functionality $\mathcal{F}_{OMDCP}$. This follows from the fact that all three algorithms use MPC operations provided by ABB, which are both sequentially and concurrently universal composable.

Theorem 1. *Let π_{NAA} be the operational metering data collection protocol employing the NAA algorithm. Then π_{NAA} securely emulates $\mathcal{F}_{OMDCP}$.*

Proof. The proof is a straightforward application of the Universal Composition Theorem 3, since (i) each ABB operation is universally composable and (ii) the NAA algorithm comprises only ABB operations, namely, `equal`, addition, and scalar multiplication. □

Theorem 2. *Let π_{NCAA} be the operational metering data collection protocol employing the NAA algorithm. Then π_{NCAA} securely emulates $\mathcal{F}_{OMDCP}$.*

Proof. For the same reason as in the proof of the previous Theorem. In this case, the ABB operations that are used in NCAA are `permute`, `equal`, addition, scalar multiplication, and `open`. □

Theorem 3. *Let π_{NIAA} be the operational metering data collection protocol employing the NIAA algorithm. Then π_{NAA} securely emulates $\mathcal{F}_{OMDCP}$.*

Proof. In the case of NIAA, only addition is used, so the proof is again straightforward. □

Note that we build all our protocols over the idealised functionality given by $\mathcal{F}_{ABB}$, and no intermediate value is revealed on any of them. Security then trivially follows from the simulation of the protocols where the adversary is restricted to what already is allowed under the ideal functionality. This principle applies to all the results in this paper.

Fault tolerance and collusion-attacks resistance: Our solutions are also fault-tolerant. This property is guaranteed due to the fact that in all our solutions the SMs send shares of the consumption data to the respective entities. These shares are generated using Shamir secret sharing with a threshold, say, τ. Therefore, even if some shares of the metering data are lost in transit, as long as $\tau + 1$ shares are present, the DCC servers can still compute the aggregates correctly.

Regarding security against collusion attacks, security under MPC can be achieved under two possible settings: an honest majority, and a dishonest majority e.g., [44]–[47]. In the former, a majority of computational parties (i.e., at least two of the DCC servers in our case) need to remain honest in order to achieve perfect security. In the latter case, we can protect against any collusion attacks as long as there is at least one honest computational party, by choosing a protocol secure against dishonest majority to implement the underlying MPC functionality.

VI. Performance Evaluation

This section evaluates the performance of our protocol (and our proposed data aggregation algorithms) in terms of computational complexity and communication cost using parameters of the smart metering architecture in the UK. In addition, we compare our protocol to (i) the traditional protocol (denoted as TRAD) proposed by the UK government, and (ii) the DEP2SA protocol [15] which uses the same system model as this work. Note that TRAD does not provide sufficient privacy protection as the DCC access all metering data of all users.

TABLE II
THE COMPUTATIONAL COMPLEXITY OF OUR PROTOCOL WHEN A DIFFERENT DATA AGGREGATION ALGORITHM IS USED.

	SM	**DCC servers**	**TSO**	**DNO**	**Supplier**
TRAD protocol	-	-	-	-	-
DEP2SA protocol [15]	1 enc	-	-	$\mathrm{N_s}$ (dec + rnr)	$\mathrm{N_d}$ enc
Our protocol with NAA	1 share	$\lvert \mathrm{s}_u \rvert \times \lvert \mathbb{SM}_{\mathrm{d}_j} \rvert \times \mathrm{N_s} + \lvert \mathbb{SM}_{\mathrm{d}_j} \rvert \times \mathrm{N_s}$ multiplication	$\mathrm{N_d} \times \mathrm{N_s}$ open	$\mathrm{N_s}$ open	$\mathrm{N_d}$ open
Our protocol with NCAA	1 share	$2 \times (\lvert \mathbb{SM}_{\mathrm{d}_j} \rvert \times \log(\lvert \mathbb{SM}_{\mathrm{d}_j} \rvert) + \lvert \mathbb{SM}_{\mathrm{d}_j} \rvert$ multiplication	$\mathrm{N_d} \times \mathrm{N_s}$ open	$\mathrm{N_s}$ open	$\mathrm{N_d}$ open
Our protocol with NIAA	$\mathrm{N_s}$ share	0 multiplication	$\mathrm{N_d} \times \mathrm{N_s}$ open	$\mathrm{N_s}$ open	$\mathrm{N_d}$ open

TABLE III
SIMULATION PARAMETERS.

Parameter	**Value**
$\mathrm{N_d}$ - number of DNOs	14
$\mathrm{N_s}$ - number of suppliers	10
$\lvert \mathrm{s}_u \rvert$ - length of supplier ID	8
$\lvert \mathbb{SM}_{\mathrm{d}_j} \rvert$ - number of SMs in each DNO	$0.5M - 4M$
$\lvert x \rvert, \lvert r \rvert$ - length of data, random number	32
$\lvert [x] \rvert$ - length of share	63
$\lvert c \rvert$ - length of encrypted data with symmetric key	128
$\lvert C \rvert$ - length of encrypted data with public key	1024

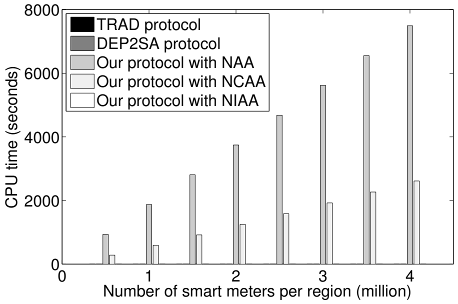


Fig. 2. Computational cost of the DCC servers.

A. Computational Complexity

The most computationally demanding step of our protocol is the *region-based aggregation* algorithm. Therefore, we focus on this step. Moreover, since the cost of a share, addition and open operations is negligible compared to the cost of a multiplication operation (in an MPC setting), we take into account only the number of multiplications in our calculation.

1) NAA complexity: This algorithm contains two loops which have the same number of multiplications. For each loop, NAA requires $\lvert \mathrm{s}_u \rvert \times \lvert \mathbb{SM}_{\mathrm{d}_j} \rvert \times \mathrm{N_s}$ multiplications to perform the equality tests needed, and $\lvert \mathbb{SM}_{\mathrm{d}_j} \rvert \times \mathrm{N_s}$ multiplications needed for the aggregation, where $\lvert \mathrm{s}_u \rvert$ is the bit length of the supplier ID, $\lvert \mathbb{SM}_{\mathrm{d}_j} \rvert$ is the number of SMs per region and $\mathrm{N_s}$ is the number of suppliers in the retail market. However, as both loops are parallelizible, the total number of multiplications in NAA is equal to $\lvert \mathrm{s}_u \rvert \times \lvert \mathbb{SM}_{\mathrm{d}_j} \rvert \times \mathrm{N_s} + \lvert \mathbb{SM}_{\mathrm{d}_j} \rvert \times \mathrm{N_s}$.

2) NCAA complexity: The number of multiplications used by the NCAA depends on the permutation network used. For instance, the Batcher oddeven merge sorting network requires $\lvert \mathbb{SM}_{\mathrm{d}_j} \rvert \times \log^2(\lvert \mathbb{SM}_{\mathrm{d}_j} \rvert)$ exchange gates. Each of these gates requires three multiplications per item being permuted, in this case the supplier ID and the respective electricity consumption or generation value. Also, the open operation performed by the DCC servers has the same computational cost as performing a multiplication. In total, this adds up to $2 \times (\lvert \mathbb{SM}_{\mathrm{d}_j} \rvert \times \log^2(\lvert \mathbb{SM}_{\mathrm{d}_j} \rvert) + \lvert \mathbb{SM}_{\mathrm{d}_j} \rvert$ multiplication-equivalent operations per loop. However, a permutation network can be built with only $\lvert \mathbb{SM}_{\mathrm{d}_j} \rvert \times \log(\lvert \mathbb{SM}_{\mathrm{d}_j} \rvert)$ exchange gates [40], reducing the total to $2 \times (\lvert \mathbb{SM}_{\mathrm{d}_j} \rvert \times \log(\lvert \mathbb{SM}_{\mathrm{d}_j} \rvert) + \lvert \mathbb{SM}_{\mathrm{d}_j} \rvert$.

3) NIAA complexity: NIAA does not perform any multiplications. As the cost of aggregation is negligible, given that it is just an arithmetic aggregation of shares, the total computational complexity of NIAA is negligible.

Table II summarises the computational complexity of our data aggregation algorithms on per entity base. The cost of the operations performed by each SM, TSO, DNO and supplier is negligible compared to the cost of the operations performed by the DCC servers. In terms of computational complexity, NIAA is the most efficient aggregation algorithm as it does not require any communication between the DCC servers.

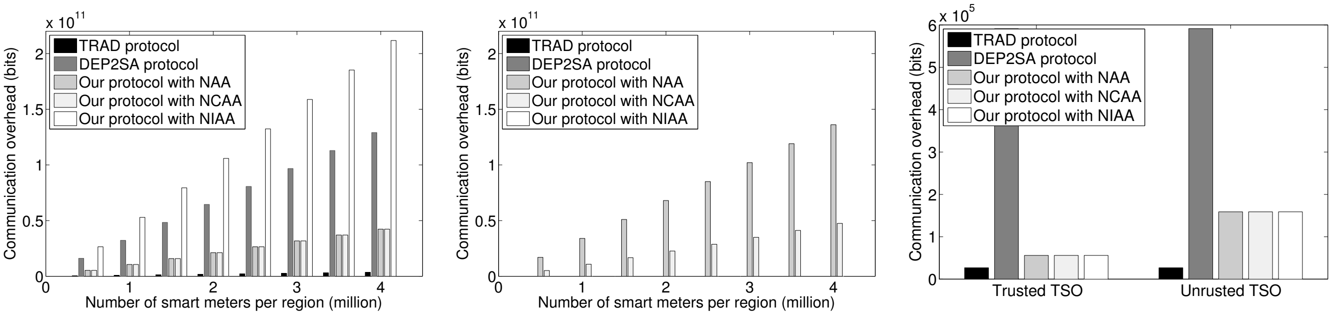
Note that TRAD does not introduce any additional cost at each entity as it does not provide privacy protection. In the case of DEP2SA, each SM encrypts its data using homomorphic encryption algorithm, each DNO performs $\mathrm{N_s}$ decryption as well as recovery of a random number from each of the ciphertexts it receives, and each supplier has to perform $\mathrm{N_d}$ encryptions. Similarly to the case of TRAD, DCC does not perform any computationally expensive operations.

We also conducted an experiment to test the performance of our algorithms. We used C++ and custom implementations of Shamir's SSS [48], its linear addition and improved BGW protocol from Gennaro et al. [49], all presented in [50]. We made use of the generalized equality test from Algorithm 1. We run the three computational parties on the same machine, a 64-bit 2*2*10-cores Intel Xeon E5-2687 server at 3.1GHz, thus our results do not consider network latency.

We first executed 2 million multiplications which, on average, resulted in 20.8×10^{-6} seconds per multiplication. We then calculated the CPU time needed by our algorithms for various settings. For our calculations we used the following parameters based on the UK's electrical grid [51] and smart metering architecture [7]: $\mathrm{N_d} = 14$, $\mathrm{N_s} = 10$, $\lvert \mathrm{s}_u \rvert = 8$, and $\lvert \mathbb{SM}_{\mathrm{d}_j} \rvert = \{0.5M, \ldots, 4M\}$. All of our simulation parameters are given in Table III. Note that the computational complexity does not depend on the metering data but on the smart metering architecture. Figure 2 depicts our experimental results. They indicate all the necessary CPU time required regardless of the number of processors. Considering that in each UK region there will be on average 2.2 million SMs, our protocol could be executed in less than ten minutes, even if NAA (our most computationally demanding algorithm) is used, by simply dividing the work between eight threads, thus making it practical for the UK smart metering architecture.

TABLE IV
THE COMMUNICATION OVERHEAD OF THE TRADITIONAL PROTOCOL AND OUR PROTOCOL.

	SMs-to-DCC	Between-DCC	DCC-to-TSO/DNOs/Suppliers
TRAD protocol	$2 \times \mathrm{N_d} \times \lvert\mathbb{SM}_{\mathrm{d}_j}\rvert \times \lvert x\rvert$	0	$6 \times \mathrm{N_d} \times \mathrm{N_s} \times \lvert x\rvert$
DEP2SA protocol [15]	$2 \times \mathrm{N_d} \times \lvert\mathbb{SM}_{\mathrm{d}_j}\rvert \times \lvert C\rvert$	0	$2 \times \mathrm{N_d} \times \mathrm{N_s} \times (2 \times \lvert C\rvert + \lvert x\rvert + \lvert r\rvert)$
Our protocol with NAA	$12 \times \mathrm{N_d} \times \lvert\mathbb{SM}_{\mathrm{d}_j}\rvert \times \lvert[x]\rvert$	$6 \times \lvert[x]\rvert \times (\lvert\mathrm{s}_u\rvert \times \lvert\mathbb{SM}_{\mathrm{d}_j}\rvert \times \mathrm{N_s} + \lvert\mathbb{SM}_{\mathrm{d}_j}\rvert \times \mathrm{N_s})$	$18 \times \mathrm{N_d} \times \mathrm{N_s} \times \lvert[x]\rvert$
Our protocol with NCAA	$12 \times \mathrm{N_d} \times \lvert\mathbb{SM}_{\mathrm{d}_j}\rvert \times \lvert[x]\rvert$	$6 \times \lvert[x]\rvert \times (2 \times (\lvert\mathbb{SM}_{\mathrm{d}_j}\rvert \times \log(\lvert\mathbb{SM}_{\mathrm{d}_j}\rvert) + \lvert\mathbb{SM}_{\mathrm{d}_j}\rvert)$	$18 \times \mathrm{N_d} \times \mathrm{N_s} \times \lvert[x]\rvert$
Our protocol with NIAA	$6 \times \mathrm{N_d} \times \lvert\mathbb{SM}_{\mathrm{d}_j}\rvert \times \mathrm{N_s} \times \lvert[x]\rvert$	0	$18 \times \mathrm{N_d} \times \mathrm{N_s} \times \lvert[x]\rvert$



(a) At the SMs-to-DCC part

(b) At the Between-DCC part

(c) At the DCC-to-TSO/DNOs/Suppliers part

Fig. 3. The communication overhead of our protocol at different parts of the grid.

B. Communication Cost

The communication cost of our protocol can be divided in three parts: SMs-to-DCC, Between-DCC and DCC-to-TSO/DNOs/Suppliers. For each part, we evaluate the communication cost of our protocol when a different aggregation algorithm is used and compare it to TRAD and DEP2SA.

1) SMs-to-DCC part: In each time slot each SM sends its tuple to each of the DCC servers. If our protocol uses NAA or NCAA, the format of the tuple is $\{[\mathrm{s}_u^{\mathrm{imp}}], [\mathrm{s}_u^{\mathrm{exp}}], [\mathrm{E}_i^{\mathrm{imp}}], [\mathrm{E}_i^{\mathrm{exp}}]\}$. Assuming there are three DCC servers, the communication cost is $3 \times \mathrm{N_d} \times \lvert\mathbb{SM}_{\mathrm{d}_j}\rvert \times ([\mathrm{s}_u^{\mathrm{imp}}] + [\mathrm{s}_u^{\mathrm{exp}}] + [\mathrm{E}_i^{\mathrm{imp}}] + [\mathrm{E}_i^{\mathrm{exp}}])$. If our protocol uses NIAA, the tuple's format is $\{[\mathbf{E}_i^{\mathrm{imp}}], [\mathbf{E}_i^{\mathrm{exp}}]\}$, where $\{[\mathbf{E}_i^{\mathrm{imp}}], [\mathbf{E}_i^{\mathrm{exp}}]\}$ are shares of vectors with size $\mathrm{N_s}$. This adds up to a cost of $3 \times \mathrm{N_d} \times \mathrm{N_s} \times \lvert\mathbb{SM}_{\mathrm{d}_j}\rvert \times ([\mathrm{E}_i^{\mathrm{imp}}] + [\mathrm{E}_i^{\mathrm{exp}}])$. If TRAD is used, each SM sends $\{\mathrm{E}_i^{\mathrm{imp}}, \mathrm{E}_i^{\mathrm{exp}}\}$ to the DCC which is a single entity in this case. This results in a communication cost of $\mathrm{N_d} \times \lvert\mathbb{SM}_{\mathrm{d}_j}\rvert \times (\mathrm{E}_i^{\mathrm{imp}} + \mathrm{E}_i^{\mathrm{exp}})$. If DEP2SA is used, the communication cost is $\mathrm{N_d} \times \lvert\mathbb{SM}_{\mathrm{d}_j}\rvert \times (\mathrm{C}^{\mathrm{imp}} + \mathrm{C}^{\mathrm{exp}})$, where $\mathrm{C}^{\mathrm{imp}}$ and $\mathrm{C}^{\mathrm{exp}}$ are the ciphertext of $\mathrm{E}_i^{\mathrm{imp}}$ and $\mathrm{E}_i^{\mathrm{exp}}$, respectively. Note that these ciphertexts are the results of the homomorphic encryption operation performed by the SMs.

2) Between-DCC part: In each time slot the DCC servers need to communicate between each other in order to preform the necessary computations for calculating the region-based aggregates per supplier. As each multiplication equals the transmission of a share from each of the DCC servers to the others, the communication cost for this part can be calculated by simply multiplying the total number of multiplications (given in Table II) with the total number of shares exchanged between the DCC servers per multiplication. In our case this is equal to $6 \times \lvert[x]\rvert$, where $\lvert[x]\rvert$ is the size of a share. TRAD and DEP2SA do not have any communication cost in this part.

3) DCC-to-TSO/DNOs/Suppliers part: In each time slot the DCC servers need to send the computed results to the TSO, DNOs and suppliers. As the output data of NAA, NCAA and NIAA is the same, the communication cost for this part is the same regardless of the aggregation algorithm. In detail, each DCC server has to send (i) $\mathrm{N_d} \times ([\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{imp}}] + [\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{exp}}])$ to each supplier, (ii) $\mathrm{N_s} \times ([\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{imp}}] + [\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{exp}}])$ to each DNO, and $\mathrm{N_d} \times \mathrm{N_s} \times ([\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{imp}}] + [\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{exp}}])$ to the TSO. This results in a total communication cost of $9 \times \mathrm{N_d} \times \mathrm{N_s} \times ([\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{imp}}] + [\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{exp}}])$. If the suppliers and DNOs trust the TSO (which is usually the case in practice), they could directly obtain the aggregation results from the TSO. In that case, the communication cost will be reduced to $3 \times \mathrm{N_d} \times \mathrm{N_s} \times ([\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{imp}}] + [\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{exp}}]) + (\mathrm{N_d} + \mathrm{N_s}) \times c_{\mathrm{d}_j,\mathrm{s}_u}$, where $c_{\mathrm{d}_j,\mathrm{s}_u}$ is an encrypted message containing the region-supplier based aggregate consumption and production data, i.e., $c_{\mathrm{d}_j,\mathrm{s}_u} = Enc_k(\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{imp}}, \mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{exp}})$. If TRAD (DEP2SA) is used, the DCC sends the respective (ciphertexts of the) aggregate consumption and generation data, $\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{imp}}, \mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{exp}}$ $(\mathbb{C}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{imp}}, \mathbb{C}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{exp}})$, to the output parties. This results in a communication cost of $3 \times \mathrm{N_d} \times \mathrm{N_s} \times (\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{imp}} + \mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{exp}})$ for TRAD. In the case of DEP2SA. each DNO also sends the respective aggregate of the consumption and generation data, as well as a random number extracted from each ciphertext, to each of the suppliers. This results in a communication cost of $\mathrm{N_d} \times \mathrm{N_s} \times (2 \times \mathbb{C}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{imp}} + 2 \times \mathbb{C}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{exp}} + \mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{imp}} + \mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{exp}} + \mathrm{r}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{imp}} + \mathrm{r}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{exp}})$.

Table IV summarises the communication cost of our protocol (with a different aggregation algorithm used), TRAD and DEP2SA, where $\lvert x\rvert$, $\lvert[x]\rvert$, $\lvert C\rvert$, and $\lvert r\rvert$, denote the length of a message, its share, a ciphertext and a random number, respectively. Furthermore, using the parameters from the previous section and setting $\lvert x\rvert = \lvert r\rvert = 32$, $\lvert[x]\rvert = 63$, $\lvert c\rvert = 128$ and $\lvert C\rvert = 1024$, we depict the communication cost of our protocol at each part and the entire smart metering architecture in Fig. 3 and Fig. 4, respectively. As expected, our protocol has higher communication cost than TRAD due to the privacy protection it offers. In comparison to DEP2SA, only

TABLE V
COMPARISON OF OUR PROTOCOL WITH EXISTING AGGREGATION PROTOCOLS.

Feature/Property	**Our**	**TRAD**	**[15]**	**[16]**	**[17]**	**[19]**	**[20]**	**[21]**	**[23]**	**[28]**	**[30]**	**[32]**	**[33]**	**[34]**	**[35]**
Real smart metering architecture	Yes	Yes	Yes	No	No	No	No	No	No	No	No	No	No	No	No
Multiple data recipients	Yes	Yes	Yes	Yes	Yes	No	No	No	No	No	No	No	No	No	No
Flexible subsets of SMs	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes	Yes	No	Yes	Yes	No
Consumption & production data	Yes	Yes	No	No	Yes	No	No	No	No	No	No	No	Yes	No	No
Multiple grid fees calculation	Yes	Yes	Yes	Yes	Yes	No	No	No	No	No	Yes	No	Yes	Yes	No
Easy supplier switch for users	Yes	Yes	Yes	No	No	No	No	No	No	No	No	No	No	No	No
Easy SM addition/removal	Yes	Yes	Yes	No	No	No	No	No	No	No	No	Yes	No	No	No
Confidentiality of user data	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Internal-attackers resistance	Yes	No	Yes	No	Yes	No	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes
Privacy of users	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Authorisation	Yes	No	Yes	No	Yes	Yes	Yes	No	No	No	No	No	No	No	No
Fault tolerant	Yes	No	No	No	Yes	No	Yes	Yes	No	No	Yes	Yes	No	No	No
Collusion-attacks resistance	Yes	No	Yes	Yes	Yes	No	Yes	No	No	Yes	Yes	No	Yes	No	No
Require secure channels*	Yes	Yes	No	Yes	Yes	No	No	No	No	No	Yes	No	No	Yes	No
Computational cost at SMs	L	L	H	H	L	L	H	L	M	H	L	M	H	H	H
Computational cost at recipients	L	L	M	L	L	L	L	L	L	H	L	L	L	H	H
Communication cost at SMs level	L	L	L	L	L	L	L	L	L	L	H	L	L	L	H
Overall communication cost	M	L	L	L	M	L	L	L	L	L	H	L	L	L	M
Scalable	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes

*This can be provided via the TLS/SSL protocol. L - Low, M - Medium, H - High

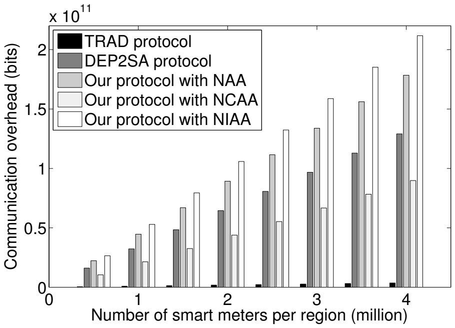


Fig. 4. The total communication overhead for our protocol.

our protocol with NCAA performs better. Regarding the choice of data aggregation algorithms, NCAA is the most efficient. However, this algorithm discloses towards the DCC servers the number of users linked to each supplier. In practice, such disclosure can be tolerated by users. If such disclosures are not accepted, NAA or NIAA should be used. Both algorithms have comparable communication costs, the difference being in the part of the smart metering architecture where the cost is concentrated. In the case of NAA, the main cost incurs at the Between-DCC part, whereas in the case of NIAA – at the SMs-to-DCC part.

C. Comparison with Existing Aggregation Protocols

Table V gives an overview of the comparison of our protocol with existing (privacy-preserving) smart metering data aggregation protocols in terms of features, functionalities, security and privacy properties as well as performance. Although most of the existing solutions provide a sufficient protection of users' privacy, they do not support functionalities needed for a real-world smart metering architecture. It is clear that our protocol is the only one that, apart from protecting users' privacy, is readily deployable in a real-world smart metering architecture due to the functionalities it supports as well as its low overhead and good scalability.

VII. CONCLUSIONS

We introduced an MPC-based protocol for aggregating electricity data, related to consumption and generation, in a secure and privacy-friendly manner. These data are required for operational purposes such as calculating the transmission, generation and balancing fees. Furthermore, we introduced three data aggregation algorithms that offer different security and performance trade-offs. We also analysed the associated computation and communication costs of all our protocols. Our results indicate the feasibility of our protocol for a setting based on a real smart metering architecture. As a future work, we will aim to have a testbed implementation of our protocol in order to test its performance in a real-world environment.

ACKNOWLEDGMENT

This work was supported in part by the Dame Kathleen Ollerenshaw Fellowship awarded by The University of Manchester, by the Research Council KU Leuven: C16/15/058, by the Flemish Government through FWO SBO project SNIPPET S007619, and by H2020-DS-2014-653497 PANORAMIX.

REFERENCES

[1] H. Farhangi, "The path of the smart grid," *IEEE Power and Energy Magazine*, vol. 8, no. 1, pp. 18–28, January 2010.

[2] A. Abidin, A. Aly, S. Cleemput, and M. A. Mustafa, "An MPC-based privacy-preserving protocol for a local electricity trading market," in *15th Int. Conf. on Cryptology and Network Security (CANS 2016)*, ser. LNCS, vol. 10052. Springer, 2016, pp. 615–625.

[3] ——, "Secure and privacy-friendly local electricity trading and billing in smart grid," *arXiv preprint arXiv:1801.08354*, 2018.

[4] G. W. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.

[5] G. Kalogridis, M. Sooriyabandara, Z. Fan, and M. A. Mustafa, "Toward unified security and privacy protection for smart meter networks," *IEEE Systems Journal*, vol. 8, no. 2, pp. 641–654, June 2014.

[6] W. Heck, "Smart energy meter will not be compulsory," *NRC Handelsblad*, April 2009, http://vorige.nrc.nl/international/article2207260.ece/Smart_energy_meter_will_not_be_compulsory.

[7] DECC, "Smart metering implementation programme data access and privacy," 2012, https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/43046/7225-gov-resp-sm-data-access-privacy.pdf.

[8] S. Cleemput, M. A. Mustafa, E. Marin, and B. Preneel, "Depseudonymization of smart metering data: Analysis and countermeasures," in *Global Internet of Things Summit (GIoTS)*, 2018, pp. 1–6.

[9] K. Kursawe, G. Danezis, and M. Kohlweiss, "Privacy-friendly aggregation for the smart-grid," in *11th Int. Symposium on Privacy Enhancing Technologies (PETs)*. Springer, 2011, pp. 175–191.

[10] F. Li, B. Luo, and P. Liu, "Secure information aggregation for smart grids using homomorphic encryption," in *IEEE 1st Int. Conf. on Smart Grid Communications (SmartGridComm)*, Oct 2010, pp. 327–332.

[11] F. D. Garcia and B. Jacobs, "Privacy-friendly energy-metering via homomorphic encryption," in *6th Int. Workshop on Security and Trust Management (STM)*. Springer, 2011, pp. 226–238.

[12] M. A. Mustafa, S. Cleemput, A. Aly, and A. Abidin, "An MPC-based protocol for secure and privacy-preserving smart metering," in *IEEE Innovative Smart Grid Technologies (ISGT-Europe)*, 2017, pp. 1–6.

[13] C. Efthymiou and G. Kalogridis, "Smart grid privacy via anonymization of smart metering data," in *IEEE 1st International Conference on Smart Grid Communications (SmartGridComm)*, October 2010, pp. 238–243.

[14] M. A. Mustafa, N. Zhang, G. Kalogridis, and Z. Fan, "MUSP: Multiservice, user self-controllable and privacy-preserving system for smart metering," in *Int. Conf. on Communications (ICC)*, 2015, pp. 788–794.

[15] ——, "DEP2SA: A decentralized efficient privacy-preserving and selective aggregation scheme in advanced metering infrastructure," *IEEE Access*, vol. 3, pp. 2828–2846, 2015.

[16] G. Danezis, C. Fournet, M. Kohlweiss, and S. Zanella-Béguelin, "Smart meter aggregation via secret-sharing," in *1st ACM Workshop on Smart Energy Grid Security (SEGS 2013)*. ACM, 2013, pp. 75–80.

[17] C. Rottondi, G. Verticale, and C. Krauss, "Distributed privacy-preserving aggregation of metering data in smart grids," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 7, pp. 1342–1354, July 2013.

[18] M. A. Rahman, M. H. Manshaei, E. Al-Shaer, and M. Shehab, "Secure and private data aggregation for energy consumption scheduling in smart grids," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 2, pp. 221–234, March 2017.

[19] P. Gope and B. Sikdar, "An efficient data aggregation scheme for privacy-friendly dynamic pricing-based billing and demand-response management in smart grids," *IEEE Internet of Things*, pp. 1–10, 2018.

[20] Y. Liu, W. Guo, C. I. Fan, L. Chang, and C. Cheng, "A practical privacy-preserving data aggregation (3PDA) scheme for smart grid," *IEEE Transactions on Industrial Informatics*, pp. 1–8, 2018.

[21] L. Lyu, K. Nandakumar, B. Rubinstein, J. Jin, J. Bedo, and M. Palaniswami, "PPFA: privacy preserving fog-enabled aggregation in smart grid," *IEEE Trans. on Industrial Informatics*, pp. 1–11, 2018.

[22] D. He, N. Kumar, and J.-H. Lee, "Privacy-preserving data aggregation scheme against internal attackers in smart grids," *Wireless Networks*, vol. 22, no. 2, pp. 491–502, Feb 2016.

[23] D. He, N. Kumar, S. Zeadally, A. Vinel, and L. T. Yang, "Efficient and privacy-preserving data aggregation scheme for smart grid against internal adversaries," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2411–2419, 2017.

[24] D. He, S. Zeadally, H. Wang, and Q. Liu, "Lightweight data aggregation scheme against internal attackers in smart grid using elliptic curve cryptography," *Wireless Communications and Mobile Computing*, vol. 22, no. 2, pp. 1–11, 2017.

[25] F. Knirsch, D. Engel, and Z. Erkin, "A fault-tolerant and efficient scheme for data aggregation over groups in the smart grid," in *IEEE Workshop on Information Forensics and Security (WIFS)*, Dec 2017, pp. 1–6.

[26] Z. Erkin, "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)*, Nov 2015, pp. 1–6.

[27] S. Rane, J. Freudiger, A. E. Brito, and E. Uzun, "Privacy, efficiency fault tolerance in aggregate computations on massive star networks," in *IEEE Workshop on Information Forensics and Security*, 2015, pp. 1–6.

[28] S. Li, K. Xue, Q. Yang, and P. Hong, "PPMA: Privacy-preserving multisubset data aggregation in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 462–471, Feb 2018.

[29] R. Lu, X. Liang, X. Li, X. Lin, and X. Shen, "EPPA: an efficient and privacy-preserving aggregation scheme for secure smart grid communications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 9, pp. 1621–1631, 2012.

[30] F. Knirsch, G. Eibl, and D. Engel, "Error-resilient masking approaches for privacy preserving data aggregation," *IEEE Transactions on Smart Grid*, pp. 1–12, 2015.

[31] A. Abdallah and X. Shen, "Lightweight security and privacy preserving scheme for smart grid customer-side networks," *IEEE Transactions on Smart Grid*, vol. 8, no. 3, pp. 1064–1074, May 2017.

[32] A. Abdallah and X. S. Shen, "A lightweight lattice-based homomorphic privacy-preserving data aggregation scheme for smart grid," *IEEE Transactions on Smart Grid*, vol. 9, no. 1, pp. 396–405, Jan 2018.

[33] H. Shen, M. Zhang, and J. Shen, "Efficient privacy-preserving cube-data aggregation scheme for smart grids," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 6, pp. 1369–1381, June 2017.

[34] F. Borges and M. Mühlhäuser, "EPPP4SMS: efficient privacy-preserving protocol for smart metering systems and its simulation using real-world data," *IEEE Trans. on Smart Grid*, vol. 5, no. 6, pp. 2701–2708, 2014.

[35] S. Tonyali, K. Akkaya, N. Saputro, A. S. Uluagac, and M. Nojoumian, "Privacy-preserving protocols for secure and reliable data aggregation in IoT-enabled smart metering systems," *Future Generation Computer Systems*, vol. 78, pp. 547–557, 2018.

[36] DECC, "The smart metering system," 2014, https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/426135/Smart_Metering_System_leaflet.pdf.

[37] R. Canetti, "Security and composition of multiparty cryptographic protocols," *Journal of Cryptology*, vol. 13, no. 1, pp. 143–202, 2000.

[38] I. Damgård, M. Fitzi, E. Kiltz, J. B. Nielsen, and T. Toft, "Unconditionally secure constant-rounds multi-party computation for equality, comparison, bits and exponentiation," in *Theory of Cryptography*, ser. LNCS, vol. 3876. Springer, 2006, pp. 285–304.

[39] H. Lipmaa and T. Toft, "Secure equality and greater-than tests with sublinear online complexity," in *ICALP (2)*, 2013, pp. 645–656.

[40] A. Czumaj, P. Kanarek, M. Kutylowski, and K. Lorys, "Delayed path coupling and generating random permutations via distributed stochastic processes," in *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 1999, pp. 271–280.

[41] A. Aly and M. Van Vyve, "Practically efficient secure single-commodity multi-market auctions," in *Financial Cryptography*, ser. LNCS. Springer, 2016.

[42] I. Damgård and J. B. Nielsen, "Universally composable efficient multiparty computation from threshold homomorphic encryption," in *CRYPTO*, vol. 2729. Springer, 2003, pp. 247–264.

[43] R. Canetti, "Security and composition of multiparty cryptographic protocols," *Journal of Cryptology*, vol. 13, no. 1, pp. 143–202, 2000.

[44] D. Chaum, C. Crépeau, and I. Damgård, "Multiparty unconditionally secure protocols," in *STOC*. ACM, 1988, pp. 11–19.

[45] M. Ben-Or, S. Goldwasser, and A. Wigderson, "Completeness theorems for non-cryptographic fault-tolerant distributed computation," in *STOC*. ACM, 1988, pp. 1–10.

[46] I. Damgård, V. Pastro, N. P. Smart, and S. Zakarias, "Multiparty computation from somewhat homomorphic encryption," in *CRYPTO '12*, ser. LNCS, vol. 7417. Springer, 2012, pp. 643–662.

[47] I. Damgård, M. Keller, E. Larraia, V. Pastro, P. Scholl, and N. P. Smart, "Practical covertly secure MPC for dishonest majority or: Breaking the SPDZ limits," in *ESORICS*. Springer, 2013, vol. 8134, pp. 1–18.

[48] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[49] R. Gennaro, M. O. Rabin, and T. Rabin, "Simplified VSS and fast-track multiparty computations with applications to threshold cryptography," in *Proceedings of the 17th Annual ACM Symposium on Principles of Distributed Computing*. ACM, 1998, pp. 101–111.

[50] A. Aly, "Network flow problems with secure multiparty computation," Ph.D. dissertation, Universté catholique de Louvain, IMMAQ, 2015.

[51] ELEXON, "The electricity trading arrangements a beginners guide," Nov. 2015, available online: https://www.elexon.co.uk/wp-content/uploads/2015/10/beginners_guide_to_trading_arrangements_v5.0.pdf.