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

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

Abstract

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 well known multiparty computation algorithms implemented in C++.

1 Introduction

The 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 several times per hour.

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 consumption and production 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. Currently, the imbalance fees for the suppliers are calculated proportional to their number of customers in each neighbourhood - i.e. a group of households connected to the same electrical substation. The current imbalance fee is only 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 serious privacy threat to users. Any entity having access to individual users' fine-grained metering data can use non-intrusive load monitoring techniques [2] to analyse consumption patterns and infer user activities [3]. 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 learn from the consumption pattern that their customers do not sleep the recommended eight hours per night. Further illustrating the importance of this issue, the Netherlands have abandoned their planned mandatory roll-out of SMs because of the privacy issues [4].

The UK has privacy protection as a requirement for their smart metering architecture [5]. 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 work was supported in part by the Research Council KU Leuven: C16/15/058. In addition, this work was supported by the European Commission through KU Leuven BOF OT/13/070, H2020-DS-2014-653497 PANORAMIX and H2020-ICT-2014-644371 WITDOM.

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

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 [6]. Aggregation is a better approach, but the current proposals [7–9] 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.

In this paper we propose a secure and privacy-preserving protocol for collecting metering data. This work extends our previous research [10] by improving the data aggregation algorithm. Our main contributions are twofold:

- We design a secure and privacy-preserving protocol for collecting operational metering data which is required for calculating distribution, transmission and imbalance fees. Our protocol uses Multiparty Computations (MPC) as the underlying cryptographic primitive and supports three different privacy-friendly data aggregation algorithms.
- We analyse the computational complexity and communication cost of our protocol in a realistic setting based on the UK's smart metering architectue [5].

The remainder of the paper is organised as follows: Section 2 discusses the related work, Section 3 gives the necessary preliminaries, Section 4 proposes a protocol (and three aggregation algorithms) for secure and privacy-preserving operational metering data collection. Sections 5 and 6 analyse its security and privacy properties, and evaluate its performance, respectively. Finally, Section 7 concludes the paper.

2 Related Work

Security and privacy concerns in SG have been raised [3] and various protocols have already been proposed [7–16]. To protect users' privacy, these protocols usually take two approaches: anonymisation or aggregation. Efthymiou and Kalogridis [11] proposed that each SM also have an anonymous ID for reporting only operational metering data. However, Tudor et al. [6] have shown that de-anonymisation is possible.

To achieve privacy-friendly aggregation, Li et al. [8] 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 [9] combined homomorphic encryption with a data sharing scheme to allow the data recipient to perform the aggregation. 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. [7] proposed a lightweight aggregation scheme which requires SMs to mask their data with noise that cancels out when added together. Their scheme is computationally efficient, but it requires a complex reinitialization process when adding SMs and does not support flexible aggregation groups.

Another approach to aggregate data in a privacy-preserving (and efficient) manner is MPC. Danezis et al. [12] proposed protocols using secret-sharing based MPC to detect fraud and to extract advanced grid statistics. Rottondi et al. [13] 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.

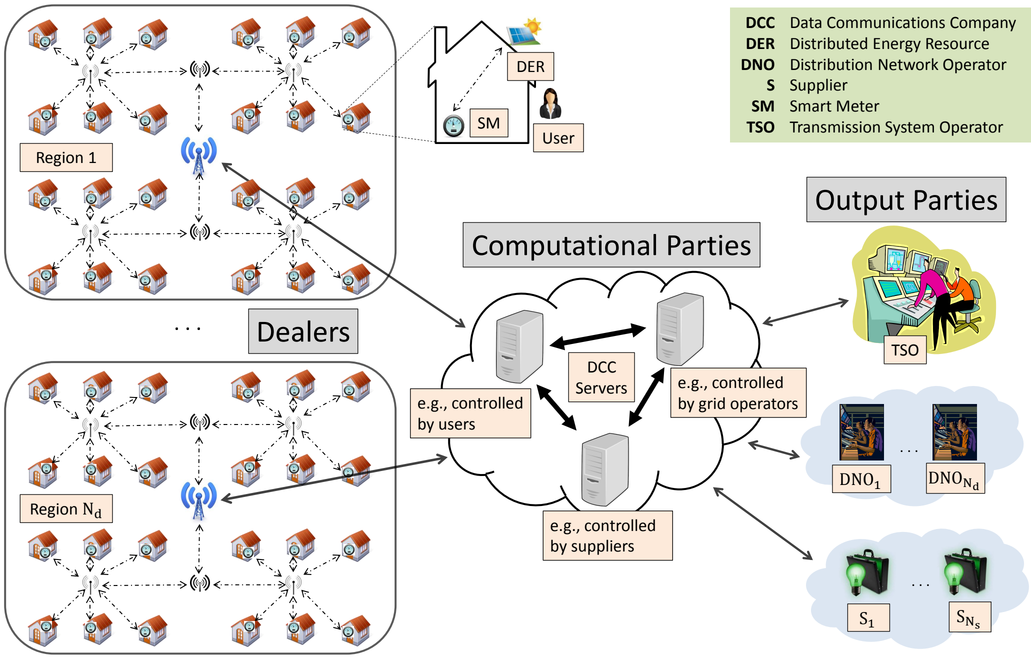


Figure 1: System model.

3 Preliminaries

3.1 System Model

As shown in Fig. 1, our system model consists of the following entities. *Users* consume electricity and are billed for this by their contracted supplier. *Distributed Energy Resources (DERs)* are mini electricity 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 electricity metering devices that measure the amount of electricity flowing from the grid to the house and vice versa per time slot, t_k. The SMs regularly communicate with other authorised SG entities. *Suppliers* are responsible for supplying electricity to all users including those whose DERs did not generate sufficient electricity for their needs. They buy this electricity from generators on the wholesale market, and sell it to the users. They are also obliged to buy any electricity their customers inject into the grid. If the supplier buys an incorrect amount of electricity on the wholesale market, 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. To this end, 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. For this, it charges the suppliers transmission and balancing fees based on the electricity consumption of their customers in each time slot. Similarly, the suppliers pass this cost to the users. *Data Communications Company (DCC)* is a centralised entity that consists of several servers run by different parties. It is responsible for collecting and delivering the metering data to the TSO, DNOs and suppliers.

We also classify some of these entities into three groups: *dealers* (i.e., the SMs) who provide the input data, *computational parties* (i.e., the DCC servers) who perform computations on the input data, and *output parties* (i.e., the 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 must be run by stakeholders with competing interest. We set the number of computational parties to three. They obtain input data from the SMs, jointly perform the necessary calculations and provide the TSO, DNOs and suppliers with the results.

3.2 Threat Model and Assumptions

For our protocol design we use the following threat model. Users are malicious. They 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 are also malicious. They may manipulate users' metering data in an attempt to gain financial advantage, i.e., to manipulate the transmission and distribution fees as well as 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 their competitors. The DCC, i.e., the computational servers, are honest but curious. They follow the protocol specifications, but they may try to learn the consumption data of individual users or aggregate data of any group of users. External entities are malicious. They may eavesdrop and/or modify data in transit trying to gain access to confidential data or to disrupt the SG.

We also make the following assumptions. Each entity in the system model has a unique identifier. SMs are tamper-proof and sealed, thus no one can tamper with them without being detected. All entities are time synchronised and the communication channels among them are encrypted and authenticated.

3.3 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 of electricity 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 electricity consumption data and electricity fed back data, respectively, of all the households during t_k in the grid. We denote the following subsets:

- $\mathbb{SM}_{\mathrm{d}_j} \subset \mathbb{SM}$ as the set of all the SMs operated by the jth DNO, d_j, (located in region j).
- $\mathbb{SM}_{\mathrm{s}_u}^{\mathrm{imp}} \subseteq \mathbb{SM}$ as the set of all the SMs whose users have a contract for buying electricity from the uth supplier, s_u.
- $\mathbb{SM}_{\mathrm{s}_u}^{\mathrm{exp}} \subseteq \mathbb{SM}$ as the set of all the SMs whose users have a contract for selling electricity to the uth supplier, s_u.
- $\mathbb{SM}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{imp}} \subseteq \mathbb{SM}_{\mathrm{d}_j}$ and $\subseteq \ \mathbb{SM}_{\mathrm{s}_u}^{\mathrm{imp}}$ as the set of all the SMs operated by d_j and whose users buy electricity from s_u.
- $\mathbb{SM}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{exp}} \subseteq \mathbb{SM}_{\mathrm{d}_j}$ and $\subseteq \ \mathbb{SM}_{\mathrm{s}_u}^{\mathrm{exp}}$ as the set of all the SMs operated by d_j and whose users sell electricity to s_u.
- $\mathbb{E}_{\mathrm{d}_j}^{\mathrm{imp},t_k}$ and $\mathbb{E}_{\mathrm{d}_j}^{\mathrm{exp},t_k}$ as the aggregate of imported and exported electricity data during t_k, respectively, measured by the SMs belonging to the set $\mathbb{SM}_{\mathrm{d}_j}$.
- $\mathbb{E}_{\mathrm{s}_u}^{\mathrm{imp},t_k}$ and $\mathbb{E}_{\mathrm{s}_u}^{\mathrm{exp},t_k}$ as the aggregate of imported and exported electricity data during t_k measured by the SMs belonging to the sets $\mathbb{SM}_{\mathrm{s}_u}^{\mathrm{imp}}$ and $\mathbb{SM}_{\mathrm{s}_u}^{\mathrm{exp}}$, respectively.
- $\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{imp},t_k}$ and $\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{exp},t_k}$ as the aggregate of imported and exported electricity data during t_k measured by the SMs belonging to the sets $\mathbb{SM}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{imp}}$ and $\mathbb{SM}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{exp}}$, respectively.

More notations are given in Table 1.

3.4 Design Requirements

The smart metering protocols should satisfy the following functional and security requirements.

3.4.1 Functional Requirements

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

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

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

Table 1: Notations

Symbol	Meaning
t_k	kth time slot, $k = 1, \dots, \mathrm{N_t}$
d_j	the DNO operating in region j, $j = 1, \dots, \mathrm{N_d}$
s_u	uth supplier, $u = 1, \dots, \mathrm{N_s}$
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 $\mathrm{SM}_i \in \mathbb{SM}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u}$

(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 $j = 1, \dots, \mathrm{N_d}$, so it can be assured that it pays the correct transmission and distribution network fees to the TSO and each DNO, respectively. Note that transmission network fees can also be made region dependent to encourage suppliers to buy electricity from sources located as close to the demand as possible.

(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 $u = 1, \dots, \mathrm{N_s}$, 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 $u = 1, \dots, \mathrm{N_s}$, 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 $j = 1, \dots, \mathrm{N_d}$, to identify the regions which are the source of the imbalance, thus to decide which measures from which sources to activate to avoid the imbalance, and

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

3.4.2 Security Requirements

(S1) Confidentiality of users' data: the aggregates (over several users) of users' consumption/production data should only be accessed by authorised entities.

(S2) User privacy preservation: individual users' fine-grained consumption/production data should not be revealed to any SG entity, apart from the users themselves.

(S3) Authorisation: SG entities should only be allowed to access the aggregate data 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.

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] ← 0;
for i ← 1 to σ do
    [c′] ← [x]_i + y_i − 2 · ([x]_i · y_i);
    [c] ← [c] + [c′] − [c] · [c′];
end
```

3.5 Cryptographic Notation

Security of MPC protocols is typically analysed in the Universally Composable (UC) framework, which was first introduced by Canetti [17]. Under this framework, the ideal functionality of MPC is modelled as an Arithmetic Black Box (ABB). ABB can be thought of 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, for example, additions, multiplications, equality tests, permutations, etc. As long as the arithmetic 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 which we describe below.

An *equality test* can be implemented in an oblivious fashion by using just multiplications and additions. Any existing test [18,19] is suitable for use in our protocol. To simplify the test, 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. Note that the algorithm can also be optimised by parallelising the computation of multiplications such that only $\log(\sigma)$ communication rounds are needed.

An oblivious permutation can be typically 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 aggregated. This method has a complexity of $\mathcal{O}(n^2)$. Other approaches, including the 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 [20]. Such oblivious permutation protocols have been also adapted for practical use in wholesale electricity markets [21].

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 $\mathbb{Z}_p$.

4 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.

4.1 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 3, 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.

4.2 Region-based Data Aggregation Algorithms

In this section, we present our three region-based data aggregation algorithms which offer different trade-offs in terms of security, flexibility and performance. The selection of the algorithm depends on the application requirements and available computational and communication resources.

4.2.1 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.

4.2.2 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.

4.2.3 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 $\mathrm{E^{imp}}$ and $\mathrm{E^{exp}}$, 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 allow the system to feed (via an update) all the SMs with a

---

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**
    $[c] \leftarrow [\mathrm{s}_u^{\mathrm{imp}}] \stackrel{?}{=} \mathrm{s}_u$;
    $[\mathbb{E}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}] \leftarrow [\mathbb{E}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}] + [c] * [\mathrm{E}_i^{\mathrm{imp}}]$;
  end
  for $u \leftarrow 1$ **to** $\mathrm{N_s}$ **do**
    $[c] \leftarrow [\mathrm{s}_u^{\mathrm{exp}}] \stackrel{?}{=} \mathrm{s}_u$;
    $[\mathbb{E}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u}] \leftarrow [\mathbb{E}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u}] + [c] * [\mathrm{E}_i^{\mathrm{exp}}]$;
  end
end

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, $[\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{imp}}]$ Shares of aggregate production data per supplier, $[\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{exp}}]$

$[\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{imp}}] \leftarrow \{0_1, \ldots, 0_{\mathrm{N_s}}\}$;
$[\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{exp}}] \leftarrow \{0_1, \ldots, 0_{\mathrm{N_s}}\}$;
$[\mathbb{SM}'_{\mathrm{d}_j}] \leftarrow \mathtt{permute}([\mathbb{SM}_{\mathrm{d}_j}])$;
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**
    $c \leftarrow \mathrm{s}_u^{\mathrm{imp}} == \mathrm{s}_u$;
    $[\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{imp}}] \leftarrow [\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{imp}}] + c * [\mathrm{E}_i^{\mathrm{imp}}]$;
  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**
    $c \leftarrow \mathrm{s}_u^{\mathrm{exp}} == \mathrm{s}_u$;
    $[\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{exp}}] \leftarrow [\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{exp}}] + c * [\mathrm{E}_i^{\mathrm{exp}}]$;
  end
end

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 $\mathbf{E}_i^{\mathrm{imp}}$ are vectors of size $\mathrm{N_s}$ with only one non-zero entry at position u

Output: Shares of aggregate consumption data per supplier, $[\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{imp}}]$ Shares of aggregate production data per supplier, $[\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{exp}}]$

$[\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{imp}}] \leftarrow \{0_1, \ldots, 0_{\mathrm{N_s}}\}$;
$[\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{exp}}] \leftarrow \{0_1, \ldots, 0_{\mathrm{N_s}}\}$;
for $i \leftarrow 1$ **to** $|\mathbb{SM}_{\mathrm{d}_j}|$ **do**
  for $u \leftarrow 1$ **to** $\mathrm{N_s}$ **do**
    $[\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{imp}}] \leftarrow [\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{imp}}] + [\mathrm{E}_{i,u}^{\mathrm{imp}}]$;
    $[\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{exp}}] \leftarrow [\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{exp}}] + [\mathrm{E}_{i,u}^{\mathrm{exp}}]$;
  end
end

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.

5 Security Analysis

To begin with, we note that the security assumptions listed in Section 3 are intended for protection against some of the threats in our threat model. For instance, the natural assumption that the SMs are tamper-proof and sealed is intended for protection against malicious users attempting to modify the metering data for financial advantage; and the assumption on communication channel being encrypted and authenticated, which can be achieved by using TLS, is for protection against adversaries attempting to eavesdrop or modify the data in transit. Hence, we focus on the security of our protocol against malicious DNOs/TSO/suppliers as well as semi-honest DCC.

We say that the protocol is secure against a (possibly malicious) entity if the privacy of the inputs to the protocol is preserved against such entity. Therefore, by security against malicious DNOs/TSO/suppliers, we mean that the SMs' inputs (i.e., the users' consumption or production data) to the protocol is preserved against these malicious entities. In our case, DNOs, TSO, and suppliers are the recipients of the MPC output. By definition, one cannot learn anything from MPC output other than what can already be learned from the output itself. Therefore, security against malicious DNOs, TSO, and suppliers is straightforward. Security against DCC is a different matter because it comprises several computation servers run by different parties. Therefore, security against DCC depends on the security of the MPC algorithm used in the protocol. As shown by Ben-Or et al. [22] and Chaum et al. [23], MPC allows to compute any function with perfect (information-theoretic) security when an honest majority is considered. Ben-Or et al. [22] further shows that perfect security against semi-honest adversaries can be achieved using the LSSS introduced by Shamir [24], as long as half of the parties remain honest. Moreover, security against malicious adversaries can be achieved using verifiable secret sharing techniques and, in this case, security can be achieved against collusion among up two thirds of the parties involved in the computation. See, for example, [25, 26] for various MPC protocols that offer security against malicious majorities.

As already mentioned, security of MPC protocols is typically analysed in the UC framework, using the ABB as the ideal functionality for MPC. As long as the arithmetic circuit components are UC secure, the UC framework guarantees that the circuit can be executed securely. In our case, the three algorithms (NAA, NCAA, and NIAA) use *multiplication*, *addition*, *equality test* and *permutation* operations as components, all of which are operations in ABB and thus can be realised securely against semi-honest or malicious adversary. Therefore, each of our three algorithms can be viewed as composition of operations provided by ABB, and thus the security of our protocol against semi-honest DCC is also straightforward.

6 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.

6.1 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.

6.1.1 NAA complexity

This algorithm contains two loops which have the same number of multiplications. For each loop, NAA requires $|\mathrm{s}_u| \times |\mathbb{SM}_{\mathrm{d}_j}| \times \mathrm{N_s}$ multiplications to perform the equality tests needed, and $|\mathbb{SM}_{\mathrm{d}_j}| \times \mathrm{N_s}$ multiplications needed for the aggregation, where $|\mathrm{s}_u|$ is the bit length of the supplier ID, $|\mathbb{SM}_{\mathrm{d}_j}|$ 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 $|\mathrm{s}_u| \times |\mathbb{SM}_{\mathrm{d}_j}| \times \mathrm{N_s} + |\mathbb{SM}_{\mathrm{d}_j}| \times \mathrm{N_s}$.

6.1.2 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 $|\mathbb{SM}_{\mathrm{d}_j}| \times \log^2(|\mathbb{SM}_{\mathrm{d}_j}|)$ 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 $2 \times (|\mathbb{SM}_{\mathrm{d}_j}| \times \log^2(|\mathbb{SM}_{\mathrm{d}_j}|) + |\mathbb{SM}_{\mathrm{d}_j}|$ multiplication-equivalent operations per loop. However, a permutation network can be built with only $|\mathbb{SM}_{\mathrm{d}_j}| \times \log(|\mathbb{SM}_{\mathrm{d}_j}|)$ exchange gates [20], reducing the total to $2 \times (|\mathbb{SM}_{\mathrm{d}_j}| \times \log(|\mathbb{SM}_{\mathrm{d}_j}|) + |\mathbb{SM}_{\mathrm{d}_j}|$.

6.1.3 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 2 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.

We also conducted an experiment to test the performance of our algorithms. We used C++ and custom implementations of Shamir's SSS [24], its linear addition and improved BGW protocol from Gennaro et al. [27], all presented in [28]. We made use of the generalized equality test from Algorithm 1.

Table 2: The computational complexity of our protocol when a different data aggregation algorithm is used.

Entities	SM	DCC servers	TSO	DNO	Supplier
Operations performed	share	multiplication	open	open	open
Our protocol with NAA	1	$\lvert \mathrm{s}_u \rvert \times \lvert \mathbb{SM}_{\mathrm{d}_j} \rvert \times \mathrm{N_s} + \lvert \mathbb{SM}_{\mathrm{d}_j} \rvert \times \mathrm{N_s}$	$\mathrm{N_d} \times \mathrm{N_s}$	$\mathrm{N_s}$	$\mathrm{N_d}$
Our protocol with NCAA	1	$2 \times (\lvert \mathbb{SM}_{\mathrm{d}_j} \rvert \times \log(\lvert \mathbb{SM}_{\mathrm{d}_j} \rvert) + \lvert \mathbb{SM}_{\mathrm{d}_j} \rvert$	$\mathrm{N_d} \times \mathrm{N_s}$	$\mathrm{N_s}$	$\mathrm{N_d}$
Our protocol with NIAA	$\mathrm{N_s}$	0	$\mathrm{N_d} \times \mathrm{N_s}$	$\mathrm{N_s}$	$\mathrm{N_d}$

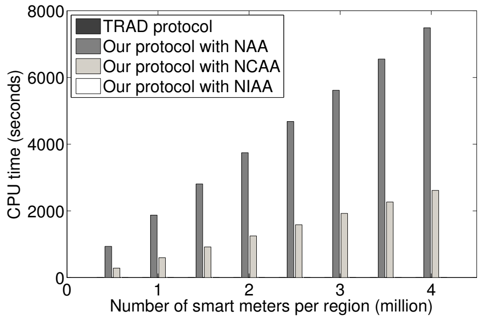


Figure 2: Computational cost of our protocol.

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 [29] and smart metering architecture [5]: $\mathrm{N_d} = 14$, $\mathrm{N_s} = 10$, $\lvert \mathrm{s}_u \rvert = 8$, and $\lvert \mathbb{SM}_{\mathrm{d}_j} \rvert = \{0.5M, \ldots, 4M\}$. 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.

6.2 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, as well as, we compare it to the traditional protocol (denoted as TRAD) proposed by the UK government. Note that TRAD does not provide sufficeint user privacy protection as the DCC access all metering data of all users.

6.2.1 SMs-to-DCC part

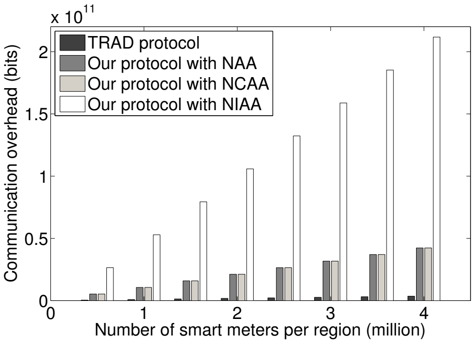
In each time slot each SM sends its tuple to each of the DCC servers. If our protocol uses NAA or NCAA, the format of the tuple is $\{[\mathrm{s}_u^{\mathrm{imp}}], [\mathrm{s}_u^{\mathrm{exp}}], [\mathrm{E}_i^{\mathrm{imp}}], [\mathrm{E}_i^{\mathrm{exp}}]\}$. Assuming there are three DCC servers, the 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 $\mathrm{N_d} \times \lvert \mathbb{SM}_{\mathrm{d}_j} \rvert \times (\mathrm{E}_i^{\mathrm{imp}} + \mathrm{E}_i^{\mathrm{exp}})$.

6.2.2 Between-DCC part

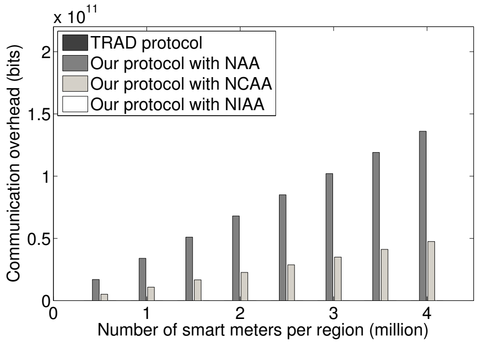
In each time slot the DCC servers need to communicate between each other in order to preform the necessary computations for calculating the region-based aggregates per supplier. As each multiplication equals the transmission of a share from each of the DCC servers to the others, the communication cost for this part can be calculated by simply multiplying the total number of multiplications (given in Table 2)

Table 3: The communication overhead of the traditional protocol and our protocol.

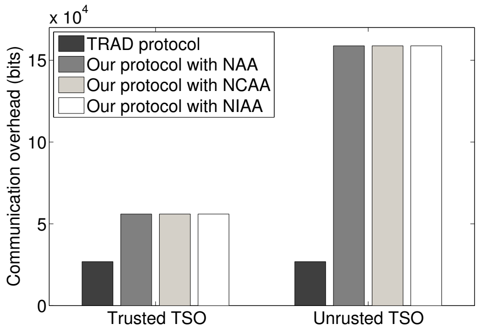
	SMs-to-DCC	Between-DCC	DCC-to-TSO/DNOs/S
TRAD protocol	$2 \times \mathrm{N_d} \times \lvert\mathbb{SM}_{\mathrm{d}_j}\rvert \times \lvert x\rvert$	0	$6 \times \mathrm{N_d} \times \mathrm{N_s} \times \lvert x\rvert$
Ours with NAA	$12 \times \mathrm{N_d} \times \lvert\mathbb{SM}_{\mathrm{d}_j}\rvert \times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

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

with the total number of shares exchanged between the DCC servers per multiplication. In our case this is equal to $6 \times |[x]|$, where $|[x]|$ is the size of a share. Note that TRAD does not have any communication cost in this part.

6.2.3 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) $\mathrm{N_d} \times ([\mathbb{E}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}] + [\mathbb{E}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u}])$ to each supplier, (ii) $\mathrm{N_s} \times ([\mathbb{E}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}] + [\mathbb{E}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u}])$ to each DNO, and $\mathrm{N_d} \times \mathrm{N_s} \times ([\mathbb{E}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}] + [\mathbb{E}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u}])$ to the TSO. This results in a total communication cost of $9 \times \mathrm{N_d} \times \mathrm{N_s} \times ([\mathbb{E}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}] + [\mathbb{E}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u}])$. 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 $3 \times \mathrm{N_d} \times \mathrm{N_s} \times ([\mathbb{E}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}] + [\mathbb{E}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u}]) + (\mathrm{N_d} + \mathrm{N_s}) \times C_{\mathrm{d}_j,\mathrm{s}_u}$, where $C_{\mathrm{d}_j,\mathrm{s}_u}$ is an encrypted message containing the region-supplier based aggregate consumption and production data, i.e., $C_{\mathrm{d}_j,\mathrm{s}_u} = Enc_k(\mathbb{E}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}, \mathbb{E}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u})$. If TRAD is used, the DCC sends the respective aggregate consumption and generation data, $(\mathbb{E}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}, \mathbb{E}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u})$, to the output parties. This results in a communication cost of $3 \times \mathrm{N_d} \times \mathrm{N_s} \times (\mathbb{E}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u} + \mathbb{E}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u})$.

Table 3 summarises the communication cost of our protocol (with a different aggregation algorithm used) and TRAD, where $|x|$ and $|[x]|$ denote the length of a message and of its share, respectively. Furthermore, using the parameters from the previous section and setting $|x| = 32$, $|[x]| = 63$ and $|C_{\mathrm{d}_j,\mathrm{s}_u}| = 128$, we depict the communication cost of our protocol at each part and the entire smart metering

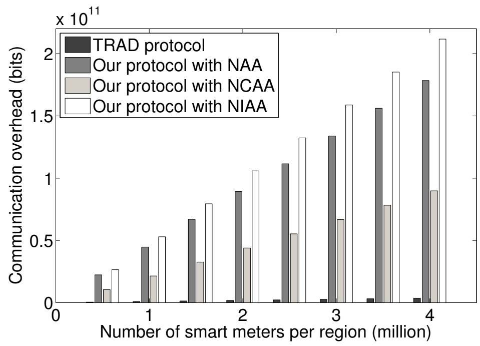


Figure 4: The total communication overhead for our protocol.

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. 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.

7 Conclusions

We introduced an MPC-based protocol for aggregating electricity consumption and generation data 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 proposed three data aggregation algorithms that offer different security and performance trade-offs. We also analysed the computational and communication cost of our protocol, including the data aggregation algorithms. Our results indicate the feasibility of our protocol for a setting based on a real smart metering architecture.

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

[3] 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, June 2014.

[4] 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.

[5] 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.

[6] V. Tudor, M. Almgren, and M. Papatriantafilou, "A study on data de-pseudonymization in the smart grid," in *ACM 8th European Workshop on System Security*, 2015, pp. 1–6.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] M. A. Mustafa, N. Zhang, G. Kalogridis, and Z. Fan, "MUSP: Multi-service, 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] 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.

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

[18] 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 *TCC 2006*, ser. LNCS, vol. 3876. Springer, 2006, pp. 285–304.

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

[20] A. Czumaj, P. Kanarek, M. Kutylowski, and K. Lorys, "Delayed path coupling and generating random permutations via distributed stochastic processes," in *SODA '99*. SIAM, 1999, pp. 271–280.

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

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

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

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

[25] 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.

[26] 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.

[27] R. Gennaro, M. O. Rabin, and T. Rabin, "Simplified VSS and fast-track multiparty computations with applications to threshold cryptography," in *PODC '98*. ACM, 1998, pp. 101–111.

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

[29] 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.