International Journal of Energy Research

INTERNATIONAL JOURNAL OF ENERGY RESEARCH
Int. J. Energy Res. 2016; **40**:313–331
Published online 14 July 2015 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/er.3355

An efficient and privacy-preserving scheme for P2P energy exchange among smart microgrids

Yuan Hong[1,*,†], Sanjay Goel[1] and Wen Ming Liu[2]

[1]University at Albany, State University of New York, Albany, NY, USA
[2]Concordia University, Montreal, Canada

SUMMARY

To date, increasing number of entities on the smart grid begin to establish their local energy generator for ensuring reliability and resilience of power supply. These ‘microgrids’ can either connect to the power grid or isolate themselves from the grid by consuming their locally generated or stored energy. In reality, some microgrids may have excessive energy while the others may have to request extra energy from the main grid. To better balance the demand and supply of the distributed smart microgrids, it is desired to develop peer-to-peer (P2P) energy exchange models that enable microgrids to interactively exchange their local energy instead of consuming energy delivered from the main grid. However, in this scenario, all the microgrids have to disclose their private information (e.g., demand load and energy storage amount) to each other in the exchange. To tackle these issues, in this paper, we first formulate several novel energy exchange optimization problems that minimize the global energy loss during the exchange in different scenarios, and then develop an efficient and privacy-preserving scheme to solve the energy exchange optimization problems without private information disclosure. We also extend the privacy-preserving scheme to a collusion-resistant scheme in which all the microgrids cannot learn any additional information through colluding with each other. The performance of our proposed approaches is experimentally validated on real microgrid data. Copyright © 2015 John Wiley & Sons, Ltd.

KEY WORDS
Smart Grid; Microgrid; Privacy; Security; Optimization

Correspondence
*Yuan Hong, University at Albany, State University of New York, 1400 Washington Ave.,·Albany, NY 12222, USA.
†E-mail: hong@albany.edu

Received 16 September 2014; Revised 18 March 2015; Accepted 3 May 2015

1. INTRODUCTION

The power grid today has low resilience, and large segments of the grid are running to capacity with little margin for error. The goal of the power grid is to balance supply and demand within a tight margin. If supply exceeds demand, there is a voltage spike; when the supply lags demand, the voltage sags. Both of these situations are detrimental to power grid operations and performance of consumer appliances. The smart grid techniques can address the problems of brittleness and instability in the grid.

1.1. Smart microgrid

Smart grid is an overlay of a communication network on top of the power grid [1]. The communication network is pervasive across the entire grid that connects generators, utilities, substations, consumers, and sensors into a monolithic network with millions of entities. The smart grid has four primary objectives: (1) improving resilience through the use of sensors and devices to isolate and correct faults autonomously; (2) improving grid stability through the use of sensors that collect state information from the grid; (3) facilitating distributed generation of electricity through integration of renewable micro energy sources on the grid; and (4) supporting demand response for households allowing for a fine-grained control of energy usage (at appliance level) to manage variability in supply and demand more economically. The smart grid assumes that the entities on the grid can produce or consume power and facilitate the metering to make distributed generation feasible. Given the provision of distributed generation of electricity, the concept of creating self-sufficient islands on the grid is appealing. Such segments of the grid can be isolated into self-sufficient islands where supply and demand are balanced. These ‘microgrids’ can operate completely isolated or in conjunction with the larger grid helping to balance any discrepancy between supply and demand. Figure 1 shows an example of smart grid energy distribution network that includes entities such as substation and

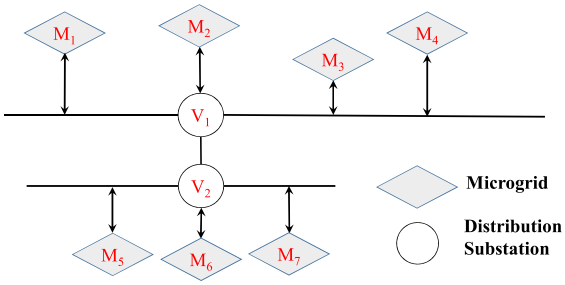


Figure 1. Microgrids in power distribution network.

microgrids. Note that some other devices (e.g., smart meter and transformers) in smart grid are not shown in Figure 1.

1.2. Peer-to-peer microgrid energy exchange

With distributed energy generation, some microgrids may have excessive power while the others may have to request extra power from the main grid. To better balance the demand and supply of distributed microgrids, it is desired to develop peer-to-peer (P2P) energy exchange models that enable microgrids to interactively exchange their local energy instead of consuming energy from the main grid [2]. There are many advantages of this new P2P energy exchange model; for instance, the energy delivery loss could be greatly reduced because local power could be delivered to adjacent entities and then energy request from the grid might be avoided. Moreover, such scheme contributes further to the autonomy of the microgrid system by decreasing the demand and reliance on the grid and gives timely response to the local demand.

Generally speaking, any microgrid can transmit energy to another microgrids on the grid. As shown in Figure 1, two adjacent microgrids (e.g., M_3 and M_4) can transmit electricity to each other via the main power distribution network. Moreover, a microgrid can also transmit excessive electricity to other non-adjacent microgrids by passing through one or more substations in the main power distribution network (e.g., M_4 to M_7 through V_1 and V_2).

1.3. Privacy in P2P energy exchange

Privacy issues in smart grid (especially meter readings) have received wide research interests recently [3]. For instance, smart meters provide real-time readings of household energy usage that can be used to make accurate demand prediction and managing the demand-response. Consumers fear that fine-grained usage information, if exposed, could lead to disclosure of their lifestyle making them vulnerable to discrimination, embarrassment, predatory marketing, and even federal and state investigations.

Different from the aforementioned generic privacy concern occurred in smart meters, we focus on *the problem of privacy among microgrids – microcosms of the entire grid, in the P2P energy exchange*. More specifically, while implementing P2P energy exchange (e.g., [2]), each microgrid's demand load and amount of stored electricity at different times as well as the amount of transmitted energy should be revealed to all the participating entities. Such thorough information disclosure has resulted in severe privacy leakage. First, every microgrid's demand load in the energy exchange explicitly reveals their private information [4]: if disclosed, other microgrids would know the amount of energy required at a particular time. Second, individual microgrid (e.g., a school or a hospital's) energy storage information at a specific time or within any time interval is its proprietary information: revealing it to untrusted parties would lead to significant security and privacy threat (e.g., power station hackers and energy thieves). Third, during the energy exchange, the amount of energy transmitted from a microgrid (e.g., M_3) to another microgrid (e.g., M_4) should be kept confidential to the other microgrids. Otherwise, the others can infer information based on it (inferring M_3's energy storage information and M_4's demand load). In summary, completely sharing the aforementioned three types of data in P2P microgrid energy exchange would directly or indirectly compromise microgrids' privacy and then pose severe security risks to the organizations[‡].

To the best of our knowledge, there is little work studying the privacy issues among multiple microgrids in the energy exchange. To address this deficiency, in this paper, we propose a novel privacy-preserving scheme to optimize the energy exchange among multiple microgrids without information disclosure in different scenarios. The main contributions of this paper are summarized in the following subsection.

1.4. Summary of contributions

1. We formulate a series of novel P2P Microgrid Energy Exchange opTimization (MEET) problems, each of which minimizes the global energy loss while feeding every individual microgrid's load demand for a practical scenario in smart grid. Efficient solvers for the optimization problems are available.
2. More importantly, we build an efficient and privacy-preserving scheme for the microgrids to exchange the energy without disclosing private information (e.g., demand load and energy storage amount) to other microgrids.
3. We also extend the privacy-preserving scheme to a collusion-resistant scheme, such that any subset of the microgrids cannot learn any additional information in the energy exchange by colluding with each other.
4. We experimentally validate the performance of the proposed scheme on real data.

The remainder of this paper is organized as follows. We first briefly review the relevant work in Section 2. Then we formulate three novel P2P MEET problems in Section 3. In

[‡] A more detailed analysis for microgrid's private information in P2P energy exchange is given in Section 1.

the same section, we also investigate the privacy concerns in the P2P energy exchange. To tackle such concerns, we present a privacy-preserving scheme (viz a secure communication protocol) for all the microgrids to solve the optimization problem without disclosing private information in Section 4. We extend the privacy-preserving scheme to a collusion-resistant scheme in Section 5. Finally, we demonstrate the effectiveness and efficiency of our proposed work using experimental results in Section 6 and conclude this paper in Section 7.

2. RELATED WORK

The privacy issues in smart grid were initially investigated in metering data, for example, households' fine-grained consumption information [3]. Then, a wide variety of techniques were developed to tackle the smart metering privacy concerns. For instance, Ács and Castelluccia [5] proposed a differentially private smart metering scheme that allows power supplier to periodically collect data from smart meters and compute aggregated statistics with strong and provable privacy guarantees. Wang *et al.* proposed a randomized response model to ensure privacy protection in smart metering [6], where the individual meters can report the true electricity consumption readings with a predetermined probability. He *et al.* [7] proposed a distortion-based privacy-preserving metering scheme by introducing tolerable noise to distort the consumption data. Rottondi *et al.* [8] presented a privacy-preserving infrastructure along with a multiparty communication protocol (based on cryptography) that allows utilities and data consumers to collect measurement data by securely aggregating smart meters. They proposed another security architecture for distributed aggregation of additive data in [9]. Renewable energy source like battery can be also utilized to hide the load/metering information of individual households that are studied in [10–13]. Energy harvesting is also an effective solution to increase smart meter privacy [12] via diversifying the energy source.

As the newer generation of energy resource, microgrid architecture typically includes the energy generator, storage devices, and inverter controller [14,15]. Some recent work studied the energy exchange problem [2,16] to share energy among homes because of the mismatch between the energy (generation) harvesting and the consumption time in microgrid. Zhu *et al.* [16] developed an energy sharing approach to determine which homes should share energy and when to minimize system-wide efficiency loss. Zhu *et al.* also studied the energy routing problem in energy exchange [17] and proposed a secure energy routing mechanism to share renewable energy in smart microgrids against security attacks such as spoofed routing signaling and fabricated routing messages. For different forms of the energy exchange models, some recent work resolved the self-interested behavior of the participating microgrids using game theoretical models (assuming rational players) [18–22,2]. Regarding privacy protection in microgrids, Wang *et al.* presented two privacy-preserving management schemes for the energy exchange [23] and scheduling [24], respectively. The energy management strategies are proposed for a single microgrid system, consisting of a control center and several cells (each includes a local controller, a distributed renewable energy generator, and some energy-consuming customers). This is different from our work that tackles the privacy issues among multiple microgrids.

Finally, because our P2P energy exchange scheme is implemented based on collaborative optimization [25], some theoretical work on privacy-preserving collaborative optimization [25] can be leveraged to develop privacy-preserving solvers for energy exchange optimization. Specifically, Li and Atallah [26] and Vaidya [27] proposed privacy-preserving techniques for solving two-party linear programming (LP) problems. Hong and Vaidya [28,29] and Mangasarian [30,31] protected the privacy while solving different forms of multiparty LP problems and their corresponding applications [32]. Besides LP, some other classic optimization problems can be also securely solved with limited disclosure in literature (e.g., [33–35]).

3. PROBLEM FORMULATION

In this section, we formally define the problems for P2P MEET in different scenarios, and then study their privacy concerns.

3.1. Peer-to-peer microgrid energy exchange optimization

Given time $t = 0, 1, 2, \ldots$, we denote the demand load and energy storage of microgrid M_i at time t as $D_i(t)$ and $S_i(t)$, respectively. Effective microgrid energy exchange generally seeks the optimal assignment for the amount of electricity transmitted between each pair of microgrids, by attaining an objective such as minimizing the global energy loss in the exchange [2,36]. As such, we name this category of optimization problems as '*Microgrid Energy Exchange opTimization* (*MEET*)'. Some representative MEET problems are outlined as follows.

3.1.1. Basic Microgrid Energy Exchange opTimization problem

Problem 1. (*Basic MEET*) Given n microgrids $i \in [1, n]$, M_i's demand load and energy storage amount (at time t): $D_i(t)$ and $S_i(t)$, basic '*Microgrid Energy Exchange opTimization* (*MEET*)' is to seek the optimal energy assignment among microgrids at time t to feed the current demand load of all microgrids, where the overall energy loss in the exchange is minimized.

Given the amount of energy to be transmitted from M_i to M_j at time t as variable x_{ij}, basic 'MEET' can be considered as a resource allocation optimization problem. We let $\theta_{ij} \in [0, 1]$ be the 'Energy Loss Rate' for power transmission from M_i to M_j in the P2P energy exchange. Thus, while transmitting energy with the amount x_{ij}, the delivery loss is given as $\theta_{ij}x_{ij}$. Notice that the energy loss rate θ_{ij} (from M_i to M_j) is determined by many factors, such as the distance between M_i and M_j in the power distribution network, types of high-voltage transmission line and voltage transformer, as well as the power transmission voltage (high, medium and low).[§]

Clearly, basic MEET problem has the objective function as follows: minimizing the overall energy loss in the exchange $\sum_{i=1}^{n} \sum_{j=1}^{n} \theta_{ij}x_{ij}$. On the other hand, the basic MEET problem essentially subjects to two groups of constraints: (1) each microgrid M_i's net energy should be no less than its demand load after the energy exchange at time t and (2) each microgrid M_i's overall outgoing amount of energy should be no greater than its storage amount $S_i(t)$. Therefore, we can formulate a LP problem to solve the basic MEET problem:

$$\min : \sum_{i=1}^{n} \sum_{j=1}^{n} \theta_{ij}x_{ij}$$
$$s.t.\begin{cases} \forall i\in[1,n], \sum_{j=1}^{n} \left(1-\theta_{ji}\right)x_{ji} - \sum_{j=1}^{n} x_{ij} + S_i(t) \geq D_i(t) \\ \forall i\in[1,n], \sum_{j=1}^{n} x_{ij} \leq \mathrm{S}_i(t) \\ \forall i\in[1,n], \forall j\in[1,n], x_{ij} \geq 0. \end{cases} \tag{1}$$

The optimal solution of the aforementioned problem $\forall i,j\in[1,n], x_{ij}^{*}$ makes the overall energy delivery loss minimized.

3.1.2. Relaxed Microgrid Energy Exchange opTimization problem

Basic MEET problem seeks the minimum energy loss, however in some special scenarios, the optimal solution does not exist because all the constraints cannot be satisfied together. For instance, if the overall amount of energy before exchange is less than the overall demand load, all the constraints in the first group (shown in Equation (1)) cannot be satisfied simultaneously. Then, the basic MEET problem (optimization) turns into infeasible. To tackle this infeasibility, a relaxed version of MEET problem can be proposed by utilizing *minimum energy* from the main grid to fill the electricity demand gap for all n microgrids. We define it as 'Relaxed MEET':

Problem 2. *Relaxed MEET* Given n microgrids $i\in[1,n]$, M_i's demand load and energy storage amount (at time t): $D_i(t)$ and $S_i(t)$, '*Relaxed Microgrid Energy Exchange opTimization* (*Relaxed MEET*)' is to seek the optimal energy assignment among microgrids at time t to feed the current demand load of all microgrids, where the overall energy loss in the exchange is minimized. Note that every microgrid is able to receive energy from the main grid in case that the overall energy storage amount is less than the overall demand load.

Letting $\in_i$ be the amount of electricity to be delivered from the main grid to microgrid M_i in the energy exchange; similarly, we denote the energy loss rate between the main grid and M_i as θ_i (which also depends on many factors, such as the distance between the corresponding substation and the microgrid in the energy distribution network, and the power transmission voltage). Then, relaxed MEET can be mathematically formulated as follows:

$$\min : \sum_{i=1}^{n} \sum_{j=1}^{n} \theta_{ij}x_{ij} + \sum_{i=1}^{n} \theta_i\varepsilon_i$$
$$s.t.\begin{cases} \forall i\in[1,n], \sum_{j=1}^{n} \left(1-\theta_{ji}\right)x_{ji} - \sum_{j=1}^{n} x_{ij} + S_i(t) \\ \qquad\qquad + (1-\theta_i)\in_i \geq D_i(t) \\ \forall i\in[1,n], \sum_{j=1}^{n} x_{ij} \leq \mathrm{S_i(t)} \\ \forall i\in[1,n], \forall j\in[1,n], x_{ij} \geq 0, \in_i \geq 0 \end{cases} \tag{2}$$

Because any amount of energy gap can be filled by the main grid, we can always find the feasible optimal solution for the relaxed MEET problem, where the overall energy delivery loss is minimized.

3.1.3. Predictive Microgrid Energy Exchange opTimization problem

In real world, microgrids usually store their energy in the batteries or as other formats for future usage (i.e., feeding a future peak demand load) and thus enhance the reliability of the power supply [1]. Essentially, peak demand of different microgrids may occur at different times, and we denote microgrid M_i's peak demand time as k_i where $k_i = 0, 1, 2, \ldots$.. Note that if $k_i = 0$, M_i's peak demand occurs at t, and energy received from the exchange will be consumed by M_i immediately. Thus, in this scenario, for each microgrid M_i, the energy exchange at time t (among all n microgrids) is implemented to feed its demand load at time $t + k_i$, where $k_i = 0, 1, 2, \ldots$.. To meet this requirement, we can define a 'Predictive MEET' problem as follows:

Problem 3. *Predictive MEET* Given n microgrids $i \in [1, n]$, M_i's energy storage amount (at time t) and demand load (at time t and M_i's peak demand time $t + k_i$): $S_i(t)$, $D_i(t)$, and $D_i(t + k_i)$, 'xtitPredictive Microgrid Energy Exchange opTimization (Predictive MEET)' is to seek the optimal energy assignment among microgrids at time t to feed the demand load of all microgrids at their corresponding peak demand time $t + k_i$, where the overall energy loss in the exchange is minimized.

Under this circumstance, each microgrid $\forall\, i \in [1, n], M_i$ is supposed to locally generate electricity between t and $t + k_i$. We let $\forall\, i \in [1, n], g_i(t, t + k_i)$ represent the difference between M_i's (estimated) newly generated and consumed energy between t and $t + k_i$, which is only known to M_i before the energy exchange. Thus, $g_i(t, t + k_i)$ is M_i's private

[§] Since energy loss can be considered as the cost of energy exchange, we can also use MEET to minimize other kinds of energy exchange cost. For instance, we can let θ_{ij} represent the unit cost of energy delivery from M_i to M_j on transmission line maintenance, labor cost, etc. Note that $\forall\, i \in [1, n], \theta_{ii} = 0$ and $x_{ii} = 0$ since every M_i does not transmit electricity to itself.

input (as the proprietary information in their operations plan) in the predictive MEET problem.

In predictive MEET problem, three groups of constraints need to be satisfied: (1) at time t, each microgrid M_i's net energy cannot be less than its demand load $D_i(t)$. Net energy of M_i at time t is regarded as its energy storage minus its outgoing energy; (2) at time $t+k_i$, each microgrid M_i's net energy cannot be less than its demand load $D_i(t+k_i)$. Net energy of M_i at time $t+k_i$ is regarded as its net energy at time t plus its (estimated) newly generated power and minus its (estimated) energy consumption between t and $t+k_i$ (3) at time t (exchange time), each microgrid M_i's overall outgoing amount of energy should be no greater than its storage amount $S_i(t)$. Then, we can mathematically formulate the predictive MEET problem as follows:

$$\min : \sum_{i=1}^{n}\sum_{j=1}^{n} \theta_{ij} x_{ij}$$

$$s.t. \begin{cases} \forall i \in [1,n], \sum_{j=1}^{n} \left(1-\theta_{ji}\right) x_{ji} - \sum_{j=1}^{n} x_{ij} + S_i(t) \geq D_i(t) \\ \forall i \in [1,n], \sum_{j=1}^{n} \left(1-\theta_{ji}\right) x_{ji} - \sum_{j=1}^{n} x_{ij} + S_i(t) \\ \qquad\qquad\qquad\qquad + g_i(t, t+k_i) \geq D_i(t+k_i) \\ \forall i \in [1,n], \sum_{j=1}^{n} x_{ij} \leq S_i(t) \\ \forall i, \forall j, x_{ij} \geq 0 \end{cases} \tag{3}$$

Two points regarding predictive MEET problem are worth noting: 1) if $\forall i \in [1,n], k_i = 0$, the peak demand time is the energy exchange time t; thus, we have $g_i(t, t+k_i) = 0$ and $D_i(t) = D_i(t+k_i)$. In this case, the predictive MEET problem turns into the basic MEET problem. (2) In case that the predictive MEET problem cannot find the feasible solution (e.g., Microgrids' capability of generating energy between the exchange time and the peak demand time is low; or the initial overall energy storage is too low at t), similar to basic MEET problem, the predictive MEET problem can be also extended to its relaxed version by additionally utilizing the same set of variables $\forall i \in [1,n], \in_i$ (by enabling energy feeding from the main grid).

In summary, the optimal solution of predictive MEET problem $\forall i,j \in [1,n], x_{ij}^*$ minimizes the overall energy loss in the exchange while sufficiently feeding each microgrid $(\forall i \in [1,n], M_i)$'s demand load at a future time $(t+k_i)$.

3.2. Privacy-preserving Microgrid Energy Exchange opTimization

In order to formulate and solve any of the MEET problems in Section 1, traditionally, all the microgrids have to share considerable amount of their private information: their energy storage amount at time t, their load demand at time t, and/or $t+k_i$, and so on. Instead, microgrids' privacy concerns can be tackled, by securely formulate and solve the MEET problems with limited disclosure, namely 'privacy-preserving MEET' (or PP-MEET).

Problem 4. *PP-MEET Given the private inputs for a MEET problem held by n microgrids $M_1, \ldots, M_n$, all microgrids jointly implement a secure communication protocol to solve the MEET problem, where the participating microgrids cannot learn any private information from each other.*

In this paper, we assume *semi-honest adversarial model*, where all the microgrids (the potential adversaries) are honest to follow the secure communication protocol but curious to derive private information of their peers. In the following, we discuss each microgrid's private share of input and output in the P2P MEET.

3.2.1. Private shares of input/output in privacy-preserving MEET against semi-honest adversaries

Section 1 has briefly described the potential privacy leakage in the microgrid energy exchange, for example, the demand load and energy storage amount, and the amount of outgoing and incoming electricity. More rigorously, each microgrid's any share of the MEET problem should be considered as the private/sensitive information, consisting of its share in both input and output. Specifically,

- In basic MEET problem (Problem 1), microgrid M_i has a private share of the input: constants $S_i(t)$, $D_i(t)$, and $\theta_{i1}, \ldots, \theta_{in}$, and variables $x_{i1}, \ldots, x_{in}$, and a private share of the output (optimal solution) $x_{i1}^*, \ldots, x_{in}^*$.
- In relaxed MEET problem (Problem 2), besides the aforementioned private share of the input/output, microgrid M_i privately holds a constant θ_i and variable $\in_i$ (as the share of the input).$\in_i$ represents M_i's amount of electricity fed by the main grid after the energy exchange, and its optimal value $\in_i^*$ should be kept confidential to other microgrids.
- In predictive MEET problem (Problem 3), besides the private share of the input/output in basic MEET problem, each microgrid M_i privately holds two additional constants $D_i(t+k_i)$ and $g_i(t, t+k_i)$ that are only known to M_i itself (confidential). Finally, if relaxation (energy feed from the main grid in case of insufficiency) is desirable for predictive MEET, constant θ_i and optimal value $\in_i^*$ should be protected as well.

Table I summarizes each microgrid's private share in the input and output of all three formulated MEET problems. We present a secure communication protocol (PP-MEET) in Section 4 to solve the MEET problems, where no microgrid can learn any other peer microgrids' any share of the input and output (shown in Table I) – strictly protecting all the private information. In case that energy loss rate between M_i and M_j, θ_{ij} is publicly known or jointly computed by M_i and M_j in practical energy transmission, we can simply tailor the protocol based on PP-MEET by disclosing the corresponding information.

3.2.2. Energy (information) flow in privacy-preserving MEET

Figure 2 presents an example of the energy (information) flow of microgrid M_4 in the exchange. We further illustrate the privacy protection using the example as follows.

Table I. Microgrid M_i's private share in the input/output of MEET problems ($i \in [1, n]$).

	Private inputs	Private output
Basic MEET problem	Constants: $S_i(t)$, $D_i(t)$, $\theta_{i1}, \ldots, \theta_{in}$ Variables: $x_{i1}, \ldots, x_{in}$	Optimal solution: $x^*_{i1}, \ldots, x^*_{in}$
Relaxed MEET problem	Constants: $S_i(t)$, $D_i(t)$, $\theta_{i1}, \ldots, \theta_{in}, \theta_i$ Variables: $x_{i1}, \ldots, x_{in}$, and $\in_i$	Optimal solution: $x^*_{i1}, \ldots, x^*_{in}$ and $\in^*_i$
Predictive MEET problem	Constants: $S_i(t)$, $D_i(t)$, $\theta_{i1}, \ldots, \theta_{in}$, $D_i(t+k_i)$, $g_i(t, t+k_i)$, and θ_i (if necessary) Variables: $x_{i1}, \ldots, x_{in}$ and $\in_i$ (if necessary)	Optimal solution: $x^*_{i1}, \ldots, x^*_{in}$ and $\in^*_i$ (if necessary)

MEET, Microgrid Energy Exchange opTimization.

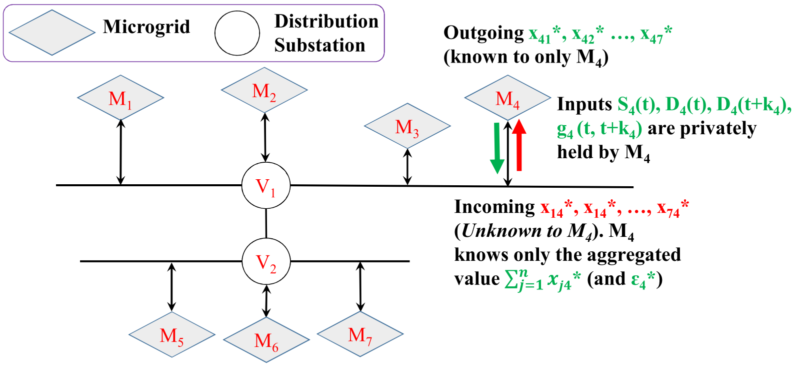

Microgrid
Distribution Substation
Outgoing x_{41}*, x_{42}* …, x_{47}* (known to only M_4)
M_1
M_2
M_3
M_4
Inputs S_4(t), D_4(t), D_4(t+k_4), g_4 (t, t+k_4) are privately held by M_4
V_1
Incoming x_{14}*, x_{14}*, …, x_{74}* (*Unknown to M_4*). M_4 knows only the aggregated value $\sum_{j=1}^{n} x_{j4}$* (and ε_4*)
V_2
M_5
M_6
M_7

Figure 2. Energy (information) flow in privacy-preserving MEET.

- While collaboratively formulating and solving the LP-based MEET problems, each microgrid only knows its share of the input: for instance, M_4 only knows $S_4(t), D_4(t), D_4(t+k_i), g_4(t, t+k_i)$, and so on.
- After jointly deriving the optimal solution with our secure communication protocol (presented in Section 4), every optimal value $\forall i,j \in [1, n], x^*_{ij}$ (the amount of transmitted electricity from M_i to M_j) is only known to the original microgrid M_i in the exchange: for instance, only M_4 knows $x^*_{41}, \ldots, x^*_{47}$.

That is to say, each original energy resource M_i and the corresponding optimal amount of transmitted energy $\forall j \in [1, n], x^*_{ij}$ would be *unknown to the energy recipient* M_j. For instance, $x^*_{14}, \ldots, x^*_{74}$ can be kept private to M_4, and we can let M_4 only know the aggregated amount of all the incoming electricity $x^*_{14} + x^*_{24} + \cdots + x^*_{74}$ by securely computing the above sum (e.g., using the technique in [37]).

- If relaxation is necessary in the MEET problem, the required electricity from the main grid ε^*_i is only revealed to M_i.

To implement energy exchange in grid distribution network (*after each microgrid knows its share of the optimal solution for MEET problem*), each microgrid can simply put the overall amount of all the outgoing electricity at time t on the grid [1,38]. Then, the electricity can flow to other microgrids automatically in the distribution network. Simultaneously, it can utilize its battery to receive incoming electricity from all other peer microgrids – terminating the external energy feed as soon as the incoming electricity amount reaches the aggregated incoming amount in the optimal solution. After the exchange, the microgrid obtains the aggregated amount of incoming electricity to feed its current or future demand load.

4. ALGORITHM AND ANALYSIS

Recall that energy flow in the P2P exchange is introduced in Section 2: each microgrid transmits the electricity based on the private shares of the optimal solution. Therefore, in this section, we focus on how to securely compute the shares of the optimal solution for the microgrids.

4.1. Microgrid Energy Exchange opTimization problem overview

All the MEET problems are collaborative LP problems [25,29] with private shares of input/output from participating microgrids (as discussed in Section 3.2.1). More specifically,

- Each microgrid $\forall i \in [1, n], M_i$ holds a *disjoint set of variables* in three MEET problems (Equations (1), (2), and (3)). For example, in basic MEET problem, M_i holds n variables – the amount of energy transmitted to n microgrids $\{x_{i1}, x_{i2}, \ldots, x_{in}\}$ (including $x_{ii}=0$). In relaxed MEET, M_i holds one more variable ε_i.

- In all three MEET problems, the global objective function – the overall energy delivery loss, is jointly obtained by all n microgrids (note that relaxed MEET problem additionally includes the energy delivery loss from the main grid to each microgrid). Then, each microgrid M_i holds a disjoint share of the objective function with respect to (w.r.t.) its privately held variables $x_{i1}, x_{i2}, \ldots, x_{in}$ and possibly $\in_i$.
- n global constraints – every microgrid's net energy (including the overall incoming energy) should be no less than the demand load at any time, are jointly obtained by all n microgrids (note that predictive MEET problem has *2n* global constraints for both time t and $t+k_i$). Then, each microgrid M_i holds a disjoint share of such constraints w.r.t. its privately held variables $x_{i1}, x_{i2}, \ldots, x_{in}$ and possibly ε_i.
- Each microgrid M_i privately holds some local constraints – its overall outgoing amount of energy should be no greater than its storage amount.

Thus, all three MEET problems can be considered as arbitrarily partitioned LP problems [29,25] (mixed with both horizontally partitioned and vertically partitioned constraints). An example for the basic MEET problem is presented in Table II. Notice that, the LP problem formulations (shares of variables, constraints, and objective function) of relaxed and predictive MEET problems are similar to that of the basic MEET problem (Table II). Thus, we first show how to securely solve the basic MEET problem in Section 2, analyze the security in Section 3, and then discuss the extension to relaxed and predictive MEET problems in Section 4.

4.2. Secure communication protocol for solving the MEET problems

We first briefly outline our privacy-preserving scheme for solving the MEET problems, which is a secure communication protocol for all n microgrids. Specifically, every microgrid M_i first *locally transforms* all its private share of the collaborative LP problem: hiding the right-hand side values of the constraints (Section 2), converting inequality constraints into equality constraints (Section 3), and encrypting its share of the constraint matrix and objective function (Section 4). Then, M_i *sends the transformed/encrypted shares* to the problem solver (which can be any of the n microgrids, or an external party; w.l.o.g., we let M_1 be the problem solver). After receiving the shares, the problem solver *formulates a transformed/encrypted LP problem* and *solves it* to obtain an optimal solution for the new problem. Subsequently, the problem solver *sends the shares of the new LP problem back* to the corresponding microgrids. Finally, each microgrid *decrypts* its share of the optimal solution for the original MEET problem.

In the following subsections, we detail the procedures in our secure communication protocol.

4.2.1. Standard form of Microgrid Energy Exchange opTimization problems (linear programming)

We denote $\forall i \in [1, n], \vec{x_i} = (x_{i1}, x_{i2} \ldots, x_{in}), \vec{c_i} = (\theta_{i1}, \theta_{i2} \ldots, \theta_{in})$, and $\vec{b_0}$ and $\vec{h}$ as

$$\vec{b_0} = \begin{pmatrix} S_1(t)-D_1(t) \\ S_2(t)-D_2(t) \\ \vdots \\ S_n(t)-D_n(t) \end{pmatrix}, \vec{h} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \tag{4}$$

Then, we can represent the standard form of basic MEET (LP) problem as follows:

$$\begin{array}{c} \min \quad \vec{c_1}^T \vec{x_1} + \vec{c_2}^T \vec{x_2} + \cdots + \vec{c_n}^T \vec{x_n} \\ s.t. \begin{cases} A_1 \vec{x_1} + \ \ldots \ + A_n \vec{x_n} & \leq & \vec{b_0} & (n \text{ Global Constraints}) \\ \vec{h}^T \vec{x_1} & \leq & S_1(t) & (\text{Local Constraint}) \\ \qquad \ddots & \vdots & \vdots & \vdots \\ \qquad\qquad \vec{h}^T \vec{x_n} & \leq & S_n(t) & (\text{Local Constraint}) \\ \vec{x_1}, \quad \ldots \quad , \vec{x_n} & \geq & 0 & \end{cases} \end{array} \tag{5}$$

where each of $A_1, \ldots A_n$ is an $n \times n$ matrix (please refer the detail values of them in Table II), each of length-n variable

Table II. Input data in MEET problems: for example, basic MEET.

	M_1 holds	M_2 holds	…	M_n holds	$S_i(t)$, $D_i(t)$ held by M_i
Objective function	$\theta_{11}x_{11} + \cdots + \theta_{1n}\theta_{1n} +$	$\theta_{21}x_{21} + \cdots + \theta_{2n}x_{2n}$	$+ \cdots +$	$\theta_{n1}x_{n1} + \cdots + \theta_{nn}x_{nn}$	
First global constraint	$x_{11} + \cdots + x_{1n}$	$+ (\theta_{21} - 1)x_{21}$	$+ \cdots +$	$(\theta_{n1} - 1)x_{n1} \leq$	$S_1(t) - D_1(t)$
Second global constraint	$(\theta_{12} - 1)x_{12} +$	$x_{21} + \cdots + x_{2n} +$	$+ \cdots +$	$(\theta_{n2} - 1)x_{n2} \leq$	$S_2(t) - D_2(t)$
⋮	⋮	⋮	⋮	⋮	⋮
nth global constraint	$(\theta_{1n} - 1)x_{1n} +$	$(\theta_{2n} - 1)x_{2n}$	$+ \cdots +$	$x_{n1} + \cdots + x_{nn} \leq$	$S_n(t) - D_n(t)$
M_1's local constraint	$x_{11} + x_{12} + \cdots + x_{1n}$				$\leq S_1(t)$
M_2's local constraint		$x_{21} + x_{22} + \cdots + x_{2n}$			$\leq S_2(t)$
⋮			⋱		⋮
M $_n$'s local constraint				$x_{n1} + x_{n2} + \cdots + x_{nn}$	$\leq S_n(t)$

MEET, Microgrid Energy Exchange opTimization.

vectors $\vec{x_1}, \ldots, \vec{x_n}$ includes n variables (for instance, $\vec{x_1} = (x_{11}, x_{12} \ldots, x_{1n})$). Indeed, the relaxed and predictive MEET problems can be also represented by the aforementioned stand form (Equation (5)) by simply adding more variables, constants intro the vector/matrix, and/or constraints into the problem.

4.2.2. Hiding $\vec{b}_0, S_1(t), \ldots, S_n(t)$

The first privacy-preserving step is to hide the right-hand side values of the constraints in the LP problem $\vec{b_0}$, $S_1(t)$,…, $S_n(t)$ (in Equation (5)) by the data holding microgrid.

Evidently, M_i privately holds a majority share of the ith global constraint:[¶]

$$\begin{aligned} & x_{i1} + x_{i2} + \cdots + x_{in} + \\ & \quad + (\theta_{1i} - 1)x_{1i} + (\theta_{2i} - 1)x_{2i} + \cdots + (\theta_{ni} - 1)x_{ni} \\ & \quad \leq S_i(t) - D_i(t) \end{aligned} \tag{6}$$

and a local constraint:

$$x_{i1} + x_{i2} + \cdots + x_{in} \leq S_i(t) \tag{7}$$

For the global constraint, M_i can create a pair of *artificial variable* and *equality constraint* to hide the right-hand side value $S_i(t) - D_i(t)$ or $S_i(t)$. For instance, Equation (6) is equivalent to the following two constraints:

$$\begin{aligned} (1)\quad & x_{i1} + x_{i2} + \cdots + x_{in} + y_{i1} + \\ & + (\theta_{1i} - 1)x_{1i} + (\theta_{2i} - 1)x_{2i} + \cdots + (\theta_{ni} - 1)x_{ni} \\ & \leq S_i(t) - D_i(t) + r_{i1}. \end{aligned}$$

(2) $r_{i2}y_{i1} = r_{i2}r_{i1}$, a new constraint created and privately held by M_i random value r_{i2} is created for hiding r_{i1} in the new constraint,

(8)

where the artificial variable y_{i1} (is fixed to a random number r_{i1}, known only to M_i) and its value r_{i1} are added to both sides of the global constraint, respectively. For instance, the equality constraint $r_{i2}y_{i1} = r_{i2}r_{i1}$ ensures that Equation (8) is equivalent to Equation (6); therefore, $S_i(t) - D_i(t)$ can be replaced by a random number $S_i(t) - D_i(t) + r_{i1}$ in the new constraint.

Similarly, the right-hand side value of every microgrid's local constraint (i.e., Equation (7)) can be hidden in the same manner – creating the equality constraint $r_{i4}y_{i2} = r_{i4}r_{i3}$ with random number r_{i3} for y_{i2} and add y_{i2} and r_{i3} to both sides of the local constraint, respectively:

$$\begin{aligned} (1)\quad & x_{i1} + x_{i2} + \cdots + x_{in} + y_{i2} \leq S_i(t) + r_{i3} \\ (2)\quad & r_{i4}y_{i2} = r_{i4}r_{i3}. \end{aligned} \tag{9}$$

Notice that newly created local constraints $r_{i2}y_{i1} = r_{i2}r_{i1}$ and $r_{i4}y_{i2} = r_{i4}r_{i3}$ are privately held by M_i. If they are requested for disclosure in the following steps, encryption will be applied before sending them out.

4.2.3. Preventing leakage from operator '≤' and '='

After hiding the right-hand side values in the MEET problems (e.g., $S_i(t)$ - $D_i(t)$ and $S_i(t)$) with the artificial variables and equality constraints, there are two kinds of constraints in the LP problem: *inequality* and *equality* constraints. If leaving '≤' and '=' in the constraints, adversaries can easily identify the corresponding constraints if they know the structure of the MEET problem (by considering the inequality constraint as the original local constraints and equality constraints as the newly created constraints). To prevent such attack, we convert all the constraints into equalities using slack variables (a standard preprocess of solving LP problems). For instance, adding a non-negative slack variable y_{i3} to the left hand side of Equation (9), it is equivalent to the following:

$$\begin{aligned} (1)\quad & x_{i1} + x_{i2} + \cdots + x_{in} + y_{i2} + y_{i3} = S_i(t) + r_{i3} \\ & y_{i3} \geq 0 \\ (2)\quad & r_{i4}y_{i2} = r_{i4}r_{i3} \end{aligned} \tag{10}$$

To further improve security, more artificial/slack variables can be created for every constraint [29]: for example, creating $y_{i4}, y_{i5}, \ldots$ and generating the equality constraint $r_{i5}(y_{i4} + y_{i5} + \ldots) = r_{i5}r_{i6}$ with random inputs, and then adding $y_{i4} + y_{i5} + \ldots$ and r_{i6} to two sides of the local constraint, respectively (this approach also applies to global constraints). In this case, more artificial/slack variables with random inputs are generated, then the original variables (and columns in the constraint matrix) can be effectively hidden and permuted. We will analyze the security in Section 3.

At this moment, different shares in the LP problem would be changed. Considering microgrid $\forall i \in [1, n], M_i$'s shares:

- n_i artificial/slack variables are added to each variable vector: $\vec{x_i} = (x_{i1}, \ldots, x_{in}) \Rightarrow \vec{x_i}' = (x_{i1}, \ldots, x_{in}, y_{i1}, y_{i2}, \ldots)$.
- Objective vector length is increased: $\vec{c_i}^T = (\theta_{i1} \ldots, \theta_{in}) \Rightarrow \vec{c_i}^T = (\theta_{i1}, \ldots, \theta_{in}, 0, 0, \ldots)$.
- The share of the global constraints is expanded by integrating artificial/slack variables and random values: $A_i\vec{x_i} \Rightarrow A_i^{'}\vec{x_i}\,'$, where A' has identical number of rows but more columns than A (because of artificial/slack variables), $\vec{b_0} \Rightarrow \vec{b_0}$ (length is identical but a random number is added to each entry of $\vec{b_0}$ by the corresponding microgrid).
- More local equality constraints are created: $x_{i1} + x_{i2} + \cdots + x_{in} \leq S_i(t) \;\Rightarrow\; B_i\vec{x_i}' = \vec{b_i}$. Specifically, now local constraints include a new equality constraint $x_{i1} + x_{i2} + \cdots + x_{in} + y_{i2} + y_{i3} + \cdots = S_i(t) + r_{i3}$ that is equivalent to the original local inequality constraint, and some other equality constraints w.r.t. artificial variables (for ensuring equivalence of the original and the new transformed constraints), such as

[¶] In Equation (12), Line 1 and 3 are held by M_i, while $(\theta_{1i} - 1)x_{1i}$, $(\theta_{2i} - 1)x_{2i}$, …, in Line 2 are held by other microgrids M_1, M_2, … respectively.

$r_{i2}y_{i1}=r_{i2}r_{i1}$ and $r_{i4}y_{i2}=r_{i4}r_{i3}$. All of M_i's local constraints are represented as $B_i\vec{x_i}'=\vec{b_i}$, with the constraint matrix B_i (which has identical number of columns as A_i'), variable vector $\vec{x_i}'$, operator '=', and random constant vector $\vec{b_i}$.

- Note that the original variables and slack variables are non-negative, yet artificial variables can be any real number. All the constraints are equalities.

Then, all the microgrids co-hold the following LP problem:

$$\min \quad \vec{c_1}'^T\vec{x_1}'+\vec{c_2}'^T\vec{x_2}'+\cdots+\vec{c_n}'^T\vec{x_n}'$$
$$s.t.\begin{cases} A_1'\vec{x_1}'+ & \dots & +A_n'\vec{x_n}' & = & \vec{b_0}' \\ B_1\vec{x_1}' & & & = & \vec{b_1} \\ & \ddots & & \vdots & \vdots \\ & & B_n\vec{x_n}' & = & \vec{b_n} \\ \vec{x_1}, & \dots & ,\vec{x_n}, \text{slack variables} & \geq & 0 \end{cases} \tag{11}$$

4.2.4. 'Encrypting' shares in Microgrid Energy Exchange opTimization problems (LP)

Besides the right-hand side values and operators, the private information in the constraint matrix and objective function of Equation (11) (e.g., θ_{ij} – the energy loss rate between M_i and M_j) should be protected against semi-honest adversaries. To this end, we present a secure mechanism to 'encrypt' all microgrid M_i's remaining unprotected shares in Equation (11), solve the new problem and 'decrypt' the optimal solution of the new problem to get the original optimal solution.

To securely solve collaborative LP problems, Du [39] and Vaidya [27] proposed a transformation approach for solving a special case of two-party LP problems: transforming an $m\times n$ constraint matrix M (and the objective vector c^T) to another $m\times n$ matrix $M\Rightarrow MQ$ (and $c^T\Rightarrow c^TQ$) by post-multiplying an $n\times n$ matrix Q, solving the transformed problem and reconstructing the original optimal solution. Bednarz *et al.* [40] showed that to ensure correctness, the transformation matrix Q must be monomial. Recently, Hong *et al.* [28,29] presented a privacy-preserving solver for (constraints) arbitrarily partitioned LP problem by extending the aforementioned matrix transformation to the multiparty case.

Following them, we let each microgrid $\forall i\in[1,n], M_i$ locally generate a 'Private Key' – a random monomial matrix Q_i, in which every row/column has exactly one non-zero random entry. Then, each microgrid encrypts its shares in the global objective vector $\vec{c_i}'^T$ and the global constraint matrix A_i', as well as its local constraint matrix B_i by post-multiplying its privately held monomial matrix Q_i (private key). As a consequence, a Shares-encrypted Form of the basic MEET problem can be jointly formulated by all microgrids as follows (privacy analysis is given in Section 3):

$$\min \quad \vec{c_1}'^TQ_1\vec{z_1}+\vec{c_2}'^TQ_2\vec{z_2}+\cdots+\vec{c_n}'^TQ_n\vec{z_n}$$
$$s.t.\begin{cases} A_1'Q_1z_1+ & \dots & +A_n'Q_n\vec{z_n} & = & \vec{b_0}' \\ B_1Q_1\vec{z_1} & & & = & \vec{b_1} \\ & \ddots & & \vdots & \vdots \\ & & B_nQ_n\vec{z_n} & = & \vec{b_n} \end{cases} \tag{12}$$

We represent the new variables in the Shares-encrypted Form as $\vec{z}=(\vec{z_1},\vec{z_2}\dots,\vec{z_n})$, which has exactly identical length as the variable vector in Equation (11): $\vec{x}'=(\vec{x_1}',\vec{x_2}',\dots,\vec{x_n}')$. Because both $\vec{z_i}'$ and x_i have $(n+n_i)$ variables,[‖] private key Q_i is generated as an $(n+n_i)\times(n+n_i)$ monomial matrix with random non-zero entries by M_i. Note that $\vec{z}=(\vec{z_1},\vec{z_2}\dots,\vec{z_n})$ can be any real number because the non-zero entries in $Q_1,\dots,Q_n$ might be negative.

4.2.5. Optimal solution reconstruction with 'Decryption'

In our model, any microgrid or external party could solve the Shares-encrypted Form (Equation (12)) without learning the private inputs from microgrids. After solving such new LP problem and obtaining an optimal solution (denoted as $\vec{z}^*$), each microgrid M_i can individually derive its share of the optimal solution for the MEET problem shown in Equation (11): $\vec{x_i}'^*$ (including $x_{i1}^*,x_{i2}^*,\dots,x_{in}^*$ and values for n_i artificial/slack variables).

Theorem 1. If the optimal solution of the Shares-encrypted Form (Equation (12)) is $\vec{z}^*=(\vec{z_1}^*,\dots,\vec{z_n}^*)$, then $\vec{x}'^*=(Q_1\vec{z_1}^*,\dots,Q_n\vec{z_n}^*)$ is the optimal solution of the MEET problem in Equation (11) (including values for the artificial/slack variables).[**]

Per Theorem 5, $\forall i\in[1,n], Q_i$ can be also considered as M_i's private key to 'decrypt' its share of the optimal solution for the Shares-encrypted Form. Specifically, the new optimal solution $\vec{z}^*=(\vec{z_1}^*,\dots,\vec{z_n}^*)$ is split into n shares, each of which belongs to the corresponding microgrid in $M_1,\dots,M_n$. Without loss of generality, we assume M_1 as the problem solver. Although M_1 knows $\vec{z}^*=(\vec{z_1}^*,\dots,\vec{z_n}^*)$, it cannot learn other microgrids' shares in the original optimal solution $\vec{x}'^*_2,\dots,\vec{x_n}'^*$ because M_1 does not know other microgrids' private keys $Q_2,\dots,Q_n$. Then, after solving the Shares-encrypted Form, M_1 distributes the shares of $\vec{z}^*=(\vec{z_1}^*,\dots,\vec{z_n}^*)$ to the corresponding microgrids $M_2,\dots,M_n$ respectively.

Finally, each microgrid $\forall i\in[1,n], M_i$ reconstructs its share in the original optimal solution using its private key Q_i: $\vec{x_i}^*=Q_i\vec{z_i}^*$. After removing all the artificial/slack variables in $\vec{x_i}'^*$, M_i can eventually obtain its share of the optimal solution for the original MEET problem $\vec{x_i}^*$, which is only known to itself.

[‖] $\vec{x_i}'$ includes n_i original variables $x_{i1}, x_{i2}, \dots, x_{in}$ and n_i artificial/slack variables.

[**] Theorem 1 is proven in the Appendix.

DOI: 10.1002/er

In summary, the private input and output of each microgrid can be kept private in the process of computing the optimal energy exchange solution at time t using the proposed scheme, as detailed in Algorithm 1.

4.3. Security analysis

In this section, we give the security analysis of Algorithm 1 through examining every microgrid's probability of inferring other microgrids' private information with its known inputs.

On the right-hand side of the constraints in the Shares-encrypted Form, $\forall i \in [1, n], S_i(t) - D_i(t)$ and $S_i(t)$ are converted into random values in $\vec{b_0}\,'$, $\vec{b_1}$, …, $\vec{b_n}$. Also, the random values added to $S_i(t) - D_i(t)$ and $S_i(t)$ (e.g., r_{i1} for y_{i1} and r_{i3} for y_{i2}) are also hidden in the product of random values (e.g., $r_{i2}r_{i1}$ and $r_{i4}r_{i3}$). Because all the aforementioned work is done locally by each microgrid, the problem solver (M_1 here) cannot infer $\forall i \in [1, n], S_i(t) - D_i(t)$ and $S_i(t)$ from the Shares-encrypted Form (Equation (12)).

Moreover, in the constraint matrix of the Shares-encrypted Form, each microgrid $\forall i \in [1, n], M_i$'s private constants $\forall j \in [1, n], \theta_{ij}$ and θ_i are also converted to random values in the encryption based on the private key (post-multiplying the share of the matrix by a random monomial matrix). For example, given A' and an $(n + n_i) \times (n + n_i)$ monomial matrix Q_i as follows:

$$A_i^{'} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1[n+n_i]} \\ a_{21} & a_{22} & \dots & a_{2[n+n_i]} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{n[n+n_i]} \end{pmatrix}, \tag{13}$$

$$Q_i = \begin{pmatrix} 0 & 0 & \dots & 0 & f_{n+n_i} \\ f_1 & 0 & \dots & 0 & 0 \\ 0 & f_2 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & f_{n+n_i\text{-}1} & 0 \end{pmatrix}$$

We thus have the following $A_i^{'}Q_i$ in the Shares-encrypted Form:

$$A_i^{'}Q_i = \begin{pmatrix} a_{12}f_1 & a_{13}f_2 & \dots & a_{1[n+n_i]}f_{n+n_i} \\ a_{22}f_1 & a_{23}f_2 & \dots & a_{2[n+n_i]}f_{n+n_i} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n2}f_1 & a_{n3}f_2 & \dots & a_{n[n+n_i]}f_{n+n_i} \end{pmatrix} \tag{14}$$

As shown previously, all the values in the matrix $A_i^{'}$ have been randomized in $A_i^{'}Q_i$, where the randomization is based on the random values $f_1, f_2, \dots, f_{n+n_i}$ and the corresponding positions in Q_i. However, the problem solver does not know the entries in the random monomial matrix Q_i (the random values $f_1, f_2, \dots, f_{n+n_i}$ and their corresponding positions in Q_i). Thus, it is impossible to infer the entries in $A_i^{'}$ without knowing the private key by the problem solver [29]. Similarly, M_i's B_i and $\vec{c_i}\,'^T$ in the MEET problem are encrypted by Q_i in the same fashion. Then, although the problem solver can obtain B_iQ_i and $\vec{c}\,'^{\,T}_{\,i}Q_i$ to formulate the Shares-encrypted Form, it cannot learn the values in B_i and $\vec{c_i}\,'^T$ without knowing Q_i.

Consequently, the problem solver (e.g., M_1) solves the Shares-encrypted Form (an LP problem with random inputs) to obtain the optimal solution $\vec{z_1}^* = \left(\vec{z_1}^*, \dots, \vec{z_n}^*\right)$. At this time, M_1 knows $\forall i \in [1, n], M_i$'s encrypted share z_i^* in the optimal solution but does not know the original true one $\vec{x_i}\,'^*$. For example,

$$\vec{z_i}^* = \begin{pmatrix} k_1 \\ k_2 \\ \vdots \\ k_{n+n_i} \end{pmatrix}, \tag{15}$$

$$\vec{x_i}\,'^* = Q_i\vec{z_i}^* = \begin{pmatrix} k_{n+n_i}f_{n+n_i} \\ k_1f_1 \\ \vdots \\ k_{n+n_i\text{-}1}f_{n+n_i\text{-}1} \end{pmatrix}.$$

Because the problem solver M_1 does not know Q_i (the random values $f_1, f_2, \dots, f_{n+n_i}$ and their corresponding

Algorithm 1 Privacy-preserving P2P MEET

```
Input: At time $t$, Microgrids $\forall i \in [1, n]$, $M_i$'s electricity storage $S_i(t)$, demand load $D_i(t)$; $\forall i, j \in [1, n]$, energy loss rate $\theta_{ij}$ (determined by distance and practical electricity transmission setup)
Output: The optimal amount of electricity to be delivered among microgrids in the energy exchange at time $t$: $x^*$
1: Each microgrid $\forall i \in [1, n]$, $M_i$ locally hides $S_i(t) - D_i(t)$ and $S_i(t)$ using $n_i$ local artificial variables $y_{i1}, y_{i2}, \dots$ and the corresponding random values $r_{i1}, r_{i2}, \dots$
Now $M_i$'s share in the global objective function and constraints matrix $\vec{c_i}\,'^T, A_i^{'}$ are expanded to $\vec{c_i}\,'^T, A_i^{'}$ by involving $n_i$ artificial/slack variables, and the right-hand side value in the global constraint is converted into a random number; $M_i$'s local constraints additionally include the new equality constraints w.r.t. $n_i$ artificial/slack variables, where constraint matrix is $B_i$, $\vec{x}\,'_{\,i}$ includes $(n + n_i)$ variables, and right-hand side vector is $b_i$
2: Each microgrid $\forall i \in [1, n]$, $M_i$ locally generates the private key – a random monomial matrix $Q_i$ with size $(n + n_i) \times (n + n_i)$, and encrypts its shares in the LP problem with $Q_i$: $\vec{c_i}\,'^T \Rightarrow \vec{c_i}\,'^T Q_i$, $A_i^{'} \Rightarrow A_i^{'}Q_i$ and $B_i \Rightarrow B_iQ_i$
Any arbitrary microgrid can be the problem solver; W.l.o.g., letting $M_1$ be the problem solver
3: Each microgrid $\forall i \in [2, n]$, $M_i$ sends the random right-hand side value of the $i$th global constraint (i.e. $S_i(t) - D_i(t) + r_{i1}$), the random right-hand side vector of its local constraints $\vec{b_i}$, encrypted shares $\vec{c_i}\,'^T Q_i, A_i^{'}Q_i, B_iQ_i$ to $M_1$
4: $M_1$ formulates the Shares-encrypted form, solves the new LP problem to obtain the optimal solution: $\vec{z}^* = \left(\vec{z_1}^*, \dots, \vec{z_n}^*\right)$, and sends $\forall i \in [2, n], \vec{z_1}^*$ to microgrid $M_i$
5: Each microgrid $\forall i \in [1, n]$, $M_i$ locally decrypts its share of the optimal solution with $Q_i$ to obtain $\vec{x_i}\,'^* = Q_1\vec{z_1}^*$
6: All the microgrids locally remove their artificial/slack variables in the optimal solution to obtain their shares in $x_i^*$ respectively
7: Each microgrid $\forall i \in [1, n]$, $M_i$ transmits the amount of $\sum_{j=1}^{n} x_{ij}^*$ electricity to the grid, and receives the amount of $\sum_{j=1}^{n} x_{ji}^*$ electricity from other microgrids
```

positions in Q_i), it is impossible for it to infer $\vec{x}_i^*$ (including values for the artificial/slack variables).

In summary, all the microgrids cannot learn any private information from each other in the secure communication protocol (Algorithm 1). Note that the aforementioned security analysis analyzes every microgrid's probability of inferring other microgrids' private information with its known inputs. It is slightly different from the security proof under the definition of secure multiparty computation (SMC) in which a polynomial machine is built to simulate all the messages received in all microgrids' views [41]. Although the analysis of inferring data does not claim that nothing can be leaked in the algorithm under SMC, it proves that the possibility of inferring private information from each other is ignorable. In the meanwhile, the efficiency of Algorithm 1 significantly outperforms an SMC protocol.

4.4. Applicability to relaxed and predictive Microgrid Energy Exchange opTimization problems

As described earlier, similar to basic MEET problem (see the standard form Equation (5) and Table II), each microgrid $\forall i \in [1, n], M_i$ in the relaxed and predictive MEET problems also has its disjoint set of variables: (1) in relaxed MEET problem, besides $\vec{x} = (x_{i1}, \ldots, x_{in})$, M_i has one more variable ε_i, and (2) in predictive MEET problem, M_i has exactly identical set of variables as the basic MEET problem.

As shown in Equations (2) and (3), each microgrid possesses a disjoint share of the global constraints and objective function w.r.t. their own set of variables in both relaxed and predictive MEET problems, and holds its own local constraint(s) w.r.t. its set of variables as well.

Therefore, three MEET problems have exactly identical structure for microgrids $M_1, \ldots, M_n$, and Algorithm 1 can be equally utilized to solve the relaxed and predictive MEET problems by simply replacing the constant inputs (note that only relaxed MEET problem requires an additional variable ε_i for each microgrid $\forall i \in [1, n], M_i$).

5. HANDLING COLLUSION

Microgrids cannot learn private information from each other if they follow the secure communication protocol (Algorithm 1). If some microgrids become malicious and collude with each other, they can learn additional information from the energy exchange (see the attack scenario in Section 1). We handle such potential collusion issue in energy exchange by extending the privacy-preserving scheme in this section.

5.1. Attack scenario

Suppose M_1 is the problem solver, if all microgrids follow the protocol, M_2 encrypts its own shares in the MEET problem and sends the encrypted shares to M_1, and finally, M_1 sends back M_2's share of the optimal solution for the Shares-encrypted Form. However, if M_1 colludes with M_2, M_2 can disclose its private key Q_2 to M_1. At this time, M_1 can easily derive M_2's share of the MEET problem and solutions (including the optimal solution and various non-optimal solutions while solving the LP problem) while implementing Algorithm 1. At the other end, because M_1 and M_2 may share everything in the collusion, M_1 can reveal all these to M_2.

Therefore, at least three side effects of compromising security/privacy occur in this scenario. First, extra information is leaked through the collusion between M_1 and M_2. Although M_2 receives its own information and it may not mind that M_1 learns its information, from the perspective of protocol security, additional information is leaked to M_1 and M_2 (this should be bounded in a provable secure protocol [42]). In other words, although the leaked information is not from other microgrids, such collusion behavior breaks the protocol (Algorithm 1) with additional information disclosure. Second, M_2 can receive its shares of all the non-optimal solutions from M_1 through collusion. Such additional information leakage can easily let M_2 have some self-interested intention to tamper its data (especially when its shares of the non-optimal solutions are not good) [29,2]. Third, M_1 and M_2 can learn more from the protocol than other microgrids, which leads to the violation of fairness.

5.2. Collusion-resistant Microgrid Energy Exchange opTimization problems

To handle the collusion issue in addition to privacy protection, we extend the secure communication protocol for PP-MEET (Algorithm 1) to achieve the following criterion.

Problem 5. Collusion-resistant MEET (*CR-MEET*) Given the private inputs for a MEET problem held by n microgrids $M_1, \ldots, M_n$, all microgrids jointly implement a secure communication protocol to solve the MEET problem, where any subset of the n microgrids cannot learn any additional information by colluding with each other.

In the following subsections, we extend the problem encryption and decryption in Algorithm 1 to a new version, where each single encryption/decryption involves all n microgrids. Then, any subset of the n microgrids cannot learn any additional information by colluding with each other.

5.3. Extended Microgrid Energy Exchange opTimization problem encryption

In public key infrastructure (a cipher was encrypted using a public key), threshold cryptosystem (e.g., [43]) requires that the number of parties to decrypt the cipher should exceed a threshold T and the private key is shared among those parties. If the number of parties participating the decryption of the cipher (out of n parties) is less than T, no useful information will be obtained.

In order to resolve the collusion among any subset of n microgrids in the MEET problems, inspired by the threshold cryptosystem, we extend our encryption/decryption for the shares of the LP problem (given in Sections 4 and 5) to a *threshold based scheme*. Specifically, all n microgrids locally generate shares of the key for encryption/decryption, and collaboratively encrypt and decrypt each share – setting the threshold is n. Missing any microgrid would result in failure of decryption (viz optimal solution reconstruction). That is to say, such threshold-based scheme is collusion-resistant.

After each microgrid locally hides the right-hand side values of the constraints with artificial/slack variables (identical to Algorithm 1), how to encrypt each microgrid $\forall i \in [1,n], M_i$'s shares in Equation (11) is detailed as follows (w.l.o.g., we let M_1 be the problem solver):

1. M_i first follows the encryption in Algorithm 1:
$$\begin{aligned} \forall i \in [1,n], \quad \vec{c_i}\,'^T &\Rightarrow \vec{c_i}\,'^T Q_i \\ A_i' &\Rightarrow A_i' Q_i \\ B_i &\Rightarrow B_i Q_i \end{aligned} \tag{16}$$

2. Each of the other (n-2) microgrids (except the problem solver M_1), $M_j, j \in [2,n], j \neq i$ locally generates an $(n+n_i)\times(n+n_i)$ random monomial matrix Q_{ij} and post-multiply it to each of $\vec{c_i}\,'^T Q_i$, $A_i' Q_i$, and $B_i Q_i$ in *an arbitrary order*.
$$\begin{aligned} \forall i \in [1,n], \quad \vec{c_i}\,'^T Q_i &\Rightarrow \vec{c_i}^T Q_i \prod_{j=2, j\neq i}^{n} Q_{ij} \\ A_i' Q_i &\Rightarrow A_i' Q_i \prod_{j=2, j\neq i}^{n} Q_{ij} \\ B_i Q_i &\Rightarrow B_i Q_i \prod_{j=2, j\neq i}^{n} Q_{ij} \end{aligned} \tag{17}$$

3. The problem solver M_1 locally generates another $(n+n_i)\times(n+n_i)$ random monomial matrix Q_{i1} and post-multiplies each of the aforementioned shares by Q_{i1}.

$$\begin{aligned} \forall i \in [1,n], \quad \vec{c_i}\,'^T Q_i \prod_{j=2, j\neq i}^{n} Q_{ij} &\Rightarrow \vec{c_i}\,'^T Q_i \prod_{j=2, j\neq i}^{n} Q_{ij} Q_{i1} \\ A_i' Q_i \prod_{j=2, j\neq i}^{n} Q_{ij} &\Rightarrow A_i' Q_i \prod_{j=2, j\neq i}^{n} Q_{ij} Q_{i1} \\ B_i Q_i \prod_{j=2, j\neq i}^{n} Q_{ij} &\Rightarrow B_i Q_i \prod_{j=2, j\neq i}^{n} Q_{ij} Q_{i1} \end{aligned} \tag{18}$$

Note that for the problem solver M_1's shares $\vec{c_1}\,'^T, A_1$, and B_1, only the first two steps are required, and step 2 involves all the remaining (n-1) microgrids. Then, M_1's final encrypted shares are the following: $\vec{c_1}\,'^T Q_1 \prod_{j=2}^{n} Q_{1j}$, $A_1' Q_1 \prod_{j=2}^{n} Q_{1j}$, and $B_1 Q_1 \prod_{j=2}^{n} Q_{1j}$.

Finally, the problem solver M_1 utilizes the aforementioned encrypted shares to formulate and solve the Collusion-resistant Shares-encrypted form of the MEET problems, as shown in Equation (19).

$$\min \quad \vec{c_1}\,'^T Q_1 \prod_{j=2}^{n} Q_{1j} \vec{z_1} + \vec{c_2}\,'^T Q_2 \prod_{j=3}^{n} Q_{2j} Q_{21} \vec{z_2} + \cdots + \vec{c_n}\,'^T Q_n \prod_{j=2}^{n-1} Q_{nj} Q_{n1} \vec{z_n}$$
$$s.t. \begin{cases} A_1' Q_1 \prod_{j=2}^{n} Q_{1j} \vec{z_1} + & A_2' Q_2 \prod_{j=3}^{n} Q_{2j} Q_{21} \vec{z_2} + & \dots & + A_n' Q_n \prod_{=2}^{n-1} Q_{nj} Q_{n1} \vec{z_n} & = & \vec{b_0}\,' \\ B_1 Q_1 \prod_{j=2}^{n} Q_{1j} \vec{z_1} & & & & = & \vec{b_1} \\ & B_2 Q_2 \prod_{j=3}^{n} Q_{2j} Q_{21} \vec{z_2} & & & = & \vec{b_2} \\ & & \ddots & & \vdots & \vdots \\ & & & B_n Q_n \prod_{j=2}^{n-1} Q_{nj} Q_{n1} \vec{z_n} & = & \vec{b_n} \end{cases} \tag{19}$$

5.4. Extended optimal solution reconstruction

The private key in the aforementioned encryption consists of n^2 random monomial matrices, shared among n different microgrids: $\forall i \in [1,n], M_i$ privately holds Q_i and $\forall j \in [1,n]$, $j \neq i, Q_{ij}$. Using the aforementioned key, all the microgrids participate the encryption process as well as the decryption (optimal solution reconstruction).

To securely solve the MEET problem using Algorithm 1, the original optimal solution can be reconstructed per Theorem 5 by letting each microgrid M_i locally pre-multiply its random monomial matrix Q_i. Similarly, if $Q_i \prod_{j=2, j\neq i}^{n} Q_{ij} Q_{i1}$ is post-multiplied to $\vec{c_i}\,'^T$, A_i', and B_i in the MEET problem, then all microgrids can jointly pre-multiply $Q_i \prod_{j=2, j\neq i}^{n} Q_{ij} Q_{i1}$ to the share of the optimal solution $\vec{z_i}^*$ to obtain its share of original optimal solution (because of the associative property of monomial matrices).

Theorem 2. If the optimal solution of the Collusion-resistant Shares-encrypted Form (Equation (19)) is $\vec{z}^* = \left(\vec{z_1}^*, \dots, \vec{z_n}^*\right)$, then $\vec{x}\,'^* = \left(Q_1 \prod_{j=2}^{n} Q_{1j} \vec{z_1}^*,\ Q_2 \prod_{j=3}^{n} Q_{2j} Q_{21} \vec{z_2}^*, \dots, Q_n \prod_{j=2}^{n-1} Q_{nj} Q_{n1} \vec{z_n}^*\right)$ is the optimal solution of the MEET problem in Equation (11) (including values for the artificial/slack variables).[††]

Similar to Algorithm 1, each microgrid $\forall i \in [1,n], M_i$'s share of the optimal solution $\vec{z_i}^*$, should be pre-multiplied by matrix $Q_i \prod_{j=2, j\neq i}^{n} Q_{ij} Q_{i1}$ to derive the original optimal solution. Because Q_i and $\forall j \in [1,n], j \neq i, Q_{ij}$ are privately held by M_i and $\forall j \in [1,n], j \neq i, M_j$, respectively, the

[††] Theorem 2 is proven in the Appendix.

decryption process (*pre-multiplying M_i's share of the optimal solution vector by each microgrid's private matrix*) should proceed with an exactly *inverse order* as the encryption: problem solver M_1, $\forall j = n, \ldots, 2, j \neq i, M_j$, [‡‡] and data owner M_i. Finally, M_i pre-multiplies its received share of the vector by Q_i to derive its share of optimal solution for the MEET problem $\vec{x_i}'^*$, and remove the artificial/slack variables in $\vec{x_i}'^*$ to obtain x_i^*.

Complying with the encryption process, decrypting M_1's share is also a special case (because M_1 is the problem solver). Such process proceeds in the following order: $\forall j = n, \ldots, 2, M_j$ and data owner M_1.

In summary, the decryption for optimal solution reconstruction is shown to be threshold-based (with threshold n). All the microgrids $M_1, \ldots, M_n$ should participate the decryption, otherwise any microgrid's share in original optimal solution cannot be reconstructed. Thus, the proposed scheme is effective to resolve the collusion among any subset of n microgrids.

5.5. Discussion on the order of microgrids in encryption and decryption

As discussed previously, all n microgrids jointly implement encryption and decryption in strictly inverse order. In our scheme, we adopt two essential criteria of determining the order of microgrids in encyrption/decryption to resolve the privacy/collusion issues as well as improve the efficiency.

Consider microgrid $\forall i \in [1, n], M_i$'s private shares in the MEET problem A_i', B_i, and $\vec{c_i}'^T$ and its private share of the optimal solution $\vec{x_i}'^*$.

First, because A_i', B_i, and $\vec{c_i}'^T$ are privately held by microgrid M_i, these shares should be post-multiplied by its random monomial matrix Q_i before sending out to any other party for privacy concern. According to the analysis in Section 3, it is safe to disclose $A_i'Q_i$, B_iQ_i, and $\vec{c_i}'^TQ_i$ to other parties because Q_i is unknown to them. Thus, Q_i must be the first monomial matrix post-multiplied to M_i's shares in the MEET problem. Otherwise, private information in A_i', B_i, and $\vec{c_i}'^T$ might be leaked to another microgrid (say M_j where $j \neq i$) that first post-multiplies the shares by its generated random monomial matrix Q_{ij}.

By contrast, to reconstruct M_i's share of the optimal solution $\vec{x_i}'^*$, Q_i should be the last monomial matrix to be pre-multiplied to the encrypted share of the optimal solution. Otherwise, other microgrids may learn M_i's share of the original optimal solution: $\vec{x_i}'^*$.

Second, the problem solver (M_1 here) is recommended to be the last microgrid to post-multiply monomial matrix Q_{i1} in encryption, and the first microgrid to pre-multiply monomial matrix Q_{i1} in decryption. At this time, the problem solver is the last party to receive M_i's encrypted shares and the first party to send the encrypted share of the optimal solution to the remaining (*n-1*) microgrids to decrypt M_i 's share of the optimal solution. Otherwise, if M_1 is not the last one to encrypt, M_1 needs to send out the encrypted shares to the next microgrid (after post-multiplying the shares by Q_{i1}) and finally receive M_i's fully encrypted shares again. As such, the computation and communication overheads of the protocol are not optimal.

Indeed, if the threshold for decryption is n as previous, the only collusion case/attack occurs when all n microgrids collude with each other: everything is disclosed to everyone. At this moment, every microgrid trusts the others (which is absolutely secure). Therefore, the protocol is secure in any case.

6. EXPERIMENTAL RESULTS

We implemented PP-MEET and CR-MEET. In this section, we present the experimental results.

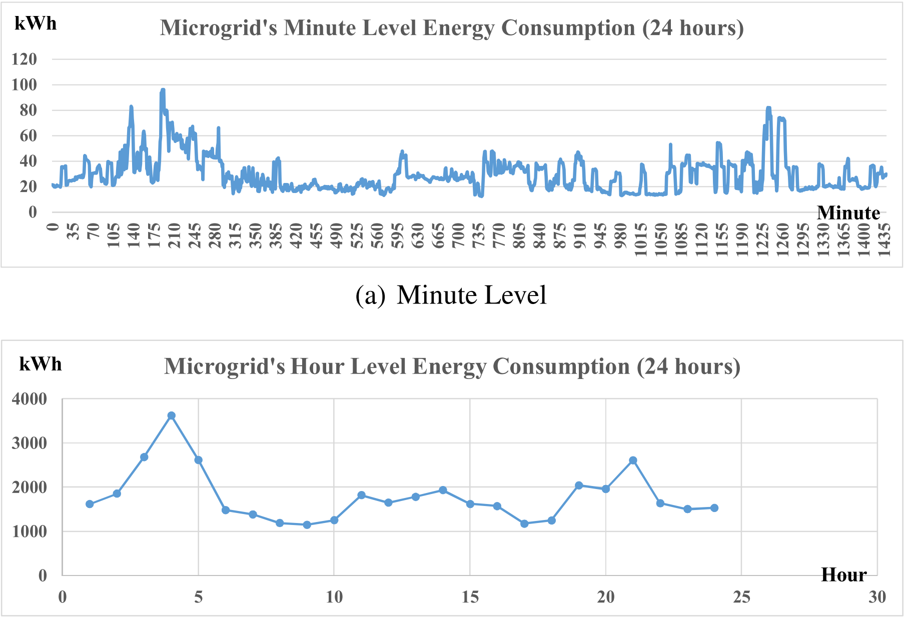
6.1. Experimental setup

We conduct the experiments using the real-world microgrid data (available at UMass Trace Repository)[§§], including 443 buildings' minute level consumption over a 24-h period. Figure 3(a) and (b) show a randomly selected building's minute and hour-level consumption, respectively.

We consider the hourly consumption as the demand load of each building at the beginning of that hour and randomly generate energy storage as well as energy generation capacity for microgrids (shown later on). For each group of experiments, we set the number of microgrids as 5, 10, 15, 20, and 25 (for testing the performance of energy exchange optimization among a varying number of microgrids). In order to simulate different sizes of microgrid in practices, we diversify the number of building each microgrid includes, by letting some microgrids (40%) involve 10 buildings each, some microgrids (40%) involve 20 buildings each, and some microgrids (20%) involve 25 buildings each. The demand loads of all these microgrids (including different number of buildings) are randomly selected from 443 buildings in the data collected by UMass Trace Repository. To simulate the energy loss between each pair of microgrids, we randomly generate the energy loss rate for pairs of microgrids from range [0.001, 0.02] and set the energy loss rate between each microgrid and the main grid (e.g., its corresponding substation) as $\theta_1 = \theta_2 = \cdots = \theta_n = 0.01$. Thus, the energy exchange

[‡‡] The order of the matrices $\forall j \in [2, n], j \neq i, Q_{ij}$ in the product in the encryption can be arbitrary; thus, the order of $\forall j = n, \ldots, 2, j \neq i, M_j$ in the decryption can be also arbitrary and only needs to be inverse to the order of $\forall j \in [2, n], j \neq i, Q_{ij}$. As long as the data owner is the first (last) and the problem solver is the last (first) in the encryption (decryption), the correctness can be ensured, as analyzed in Section 5.5.

[§§] http://traces.cs.umass.edu/index.php/Smart/Smart



(a) Minute Level

(b) Hour Level

Figure 3. An example of microgrid's energy consumption (a randomly selected building).

between some pairs of microgrids may exceed 0.01 (this matches the fact that those microgrids have to exchange energy by passing through the substations in the distribution network). We run every test for 10 times and average the experimental results. Table III summarizes the microgrid setup in our experiments.

6.2. Optimal P2P energy exchange

Considering the energy consumption in real data as the demand load at t, we simulate a reasonable energy storage amount based the given demand load to test the MEET problems.

6.2.1. Basic Microgrid Energy Exchange opTimization problem

For basic MEET problems, to increase the probability that the MEET problems are feasible, we generate each microgrid's energy storage amount at t as the sum of the demand load and a random number (between -25% of the demand load and 50% of the demand load). Then, some of microgrids' demand load is higher than their energy storage, while more microgrids are likely to have higher energy storage. Because the overall energy storage tends to be higher than the demand load, thus the basic MEET problems are very likely to be feasible (in our experiments, approximately 95% of the MEET problems are feasible).

Table III. Setup (microgrids' demand loads are captured from 443 real-world buildings).

Number of microgrids	5	10	15	20	25
Number of 10-building microgrids	2	4	6	8	10
Number of 20-building microgrids	2	4	6	8	10
Number of 25-building microgrids	1	2	3	4	5

$$\forall i \in [1, n], j \in [1, n], i \neq j, \theta_{ij} \in [0.001, 0.02]$$

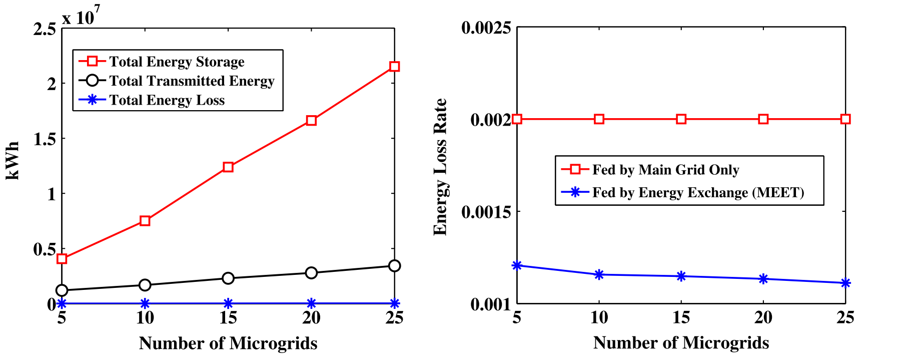
$$\forall i \in [1, n], \theta_i = 0.01$$

As shown in Figure 4(a), if we optimize the energy loss in delivery with the basic MEET problems, the total transmitted energy from all the microgrids incline to less than 1/4 of the total energy storage. Moreover, the total energy loss there is ignorable compared with the transmitted energy and the energy storage.

To justify the effectiveness of MEET problems, we compare the overall energy loss rate (viz total energy loss/total transmitted energy) in two cases: (1) no energy exchange (each microgrid's gap of the demand load and energy storage is fed by the main grid) and (2) energy exchange is implemented based on basic MEET. As shown in Figure 4(b), the overall energy loss rate of the basic MEET problem is significantly smaller than that of no energy exchange in any case.

6.2.2. Relaxed Microgrid Energy Exchange opTimization problem

To trigger the relaxed MEET problem, basic MEET problem should be infeasible. Thus, different from basic MEET problem, we generate each microgrid's energy storage amount at t as the sum of the demand load and a random number (between -50% of the demand load and 25% of the demand load). At this time, the overall energy storage amount tends to be lower than the demand load; thus, the basic MEET problems are very likely to be



(a) Energy in Basic MEET Problem

(b) Basic MEET vs. No Exchange (Overall Energy Loss Rate)

Figure 4. Basic MEET problems. MEET, Microgrid Energy Exchange opTimization.

infeasible. Once infeasibility occurs, relaxed MEET problem can be used to minimize the overall energy loss.

Figure 5 presents the total energy loss for no energy exchange and energy exchange with relaxed MEET problem, respectively. Clearly, the latter one is lower than the former one. The amount of energy fed by the main grid in relaxed MEET problem is very limited, as evident from the energy loss of the main grid portion in relaxed MEET problem (very small).

In the aforementioned randomly generated relaxed MEET problems, if the corresponding basic MEET problems are feasible, we have $\varepsilon_1^*, \ldots, \varepsilon_n^* = 0$ (unnecessary to obtain energy from the main grid).

6.2.3. Predictive Microgrid Energy Exchange opTimization problem

Predictive MEET problem requires each microgrid's demand load at both time t and $t+k_i$ as the input. Then, we utilize the energy consumption in the first hour and last hour in the real data as the demand load at t and $t+k_i$, respectively. Similar to basic and relaxed MEET problems, we generate each microgrid's energy storage amount at t as the sum of the demand load and a random number (between −50% of the demand load and 50% of the demand load). At this time, whether the overall energy is sufficient or not for feeding, all the microgrids by energy exchange is uncertain. Thus, if the generated problem is feasible, we solve the original version of the predictive MEET problem; otherwise, we solve the relaxed version of the predictive MEET problem (by allowing energy feeding from substations/main grid).

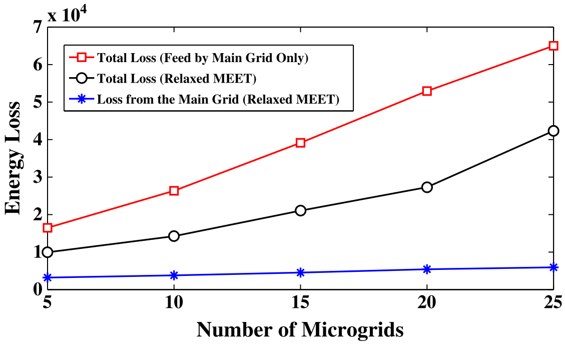


Figure 5. Energy in relaxed MEET problems. MEET, Microgrid Energy Exchange opTimization.

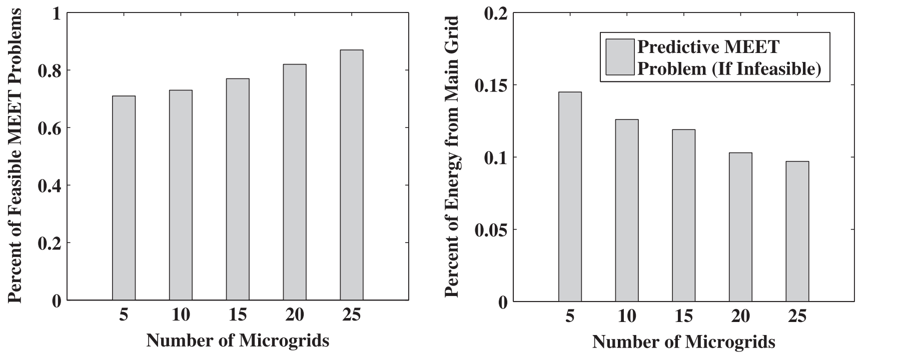
Because the performance of predictive MEET problems on energy loss is very close to the basic and relaxed MEET problems, we only demonstrate some other facts in addition to the previous two MEET problems. The percent of feasible predictive MEET problems is very high (Figure 6(a)). If the problem is infeasible, the percent of energy fed from the main grid is also limited (Figure 6(b)).

6.3. Privacy-preserving MEET and collusion-resistant MEET

We now look at the PP-MEET and the CR-MEET problems. According to the correctness proof for both encryption and decryption with the keys (monomial matrices), the quality of the outputs (e.g., minimized energy loss) in PP-MEET and CR-MEET is exactly identical to the original MEET problems. We thus test the efficiency and scalability of the PP-MEET and CR-MEET problems in the experiments.

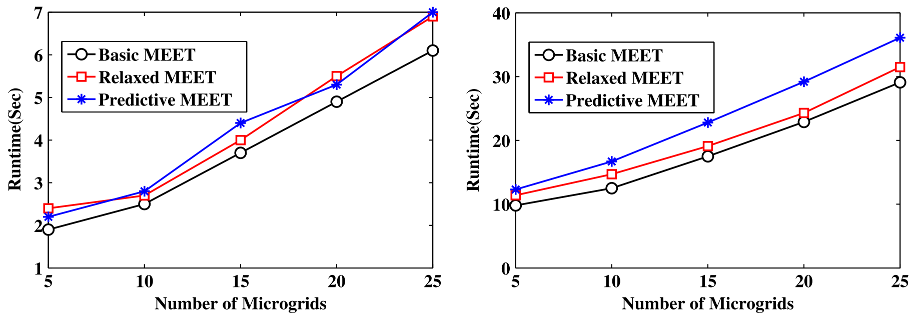
6.3.1. Efficiency

Compared with traditional secure communication protocol based on complex cryptographic primitives, Algorithm 1 (and the extended version for CR-MEET) implements encryption and decryption by post-multiplying random monomial matrices. Then, all three types of MEET problems can be securely solved with significantly boosted efficiency. More specifically, we plot the computational costs of PP-MEET and CR-MEET based on all three problems (basic, relaxed, and predictive) in Figure 7(a) and (b), respectively. Note that the largest LP problem, which is the relaxed version of the predictive MEET problem (25 microgrids), only includes approximately $25 * 25 + 25 = 650$ variables and $25 * 2 = 30$ constraints. It takes only a couple of seconds to implement PP-MEET and less than



(a) Percent of Feasible Predictive Problems (b) Percent of Energy from Main Grid (If Infeasible)

Figure 6. Predictive MEET problems. MEET, Microgrid Energy Exchange opTimization.



(a) Privacy-preserving MEET (b) Collusion-resistant MEET

Figure 7. Computation cost. MEET, Microgrid Energy Exchange opTimization.

40 s to implement CR-MEET (including encryption and decryption), because LP problems can be efficiently solved by standard solvers such as simplex or revised simplex algorithm [29]. Compared with PP-MEET, CR-MEET trades some efficiency for more rigorous security, and the computation cost is still not high. Thus, choosing PP-MEET or CR-MEET depends on the practical requirements on privacy/security and efficiency.

Moreover, the ciphertexts in both PP-MEET and CR-MEET (encrypted shares of the LP problems) have exactly identical size as the plaintext in MEET problems. Therefore, the communication overheads are ignorable – only some random matrices/vectors need to be sent to the problem solver before solving the problem. For instance, in the aforementioned largest MEET problem, each microgrid only sends a 26 × 51 (approximately) matrix to the problem solver, and finally receives a vector with 26 (approximately) entries from the problem solver. As shown in Table IV, the minor bandwidth consumptions (communication overheads) of the PP-MEET and CR-MEET will not cause any latency in most networking environment.

Table IV. Bandwidth consumption in the protocols (kilobytes).

Number of microgrids	5	10	15	20	25
Basic MEET (PP)	7.1	13.9	21.3	28.1	34.9
Basic MEET (CR)	38.2	127.4	342.8	567.9	914.3
Relaxed MEET (PP)	7.4	14.3	21.7	28.5	35.6
Relaxed MEET (CR)	41.1	132	357.5	585.3	969.7
Predictive MEET (PP)	7.6	14.2	22.1	28.9	35.9
Predictive MEET (CR)	42.6	130.9	362.8	579.2	947

MEET, Microgrid Energy Exchange opTimization; PP, privacy-preserving; CR, collusion-resistant.

6.3.2. Robustness

Figure 7 not only presents high efficiency of PP-MEET and CR-MEET (for three MEET problems) but also shows excellent scalability. The computation and communication overheads increase polynomially as the number of microgrids increases for both PP-MEET and CR-MEET. Furthermore, the network latency is ignorable while all microgrids jointly run the secure communication protocols. This makes the protocol easily implementable in any networking environment.

Because the protocol for PP-MEET and CR-MEET is highly efficient, the computation and communication capacity of current smart grid system can afford highly frequent energy exchange among all the microgrids. For instance, microgrids can implement the energy exchange (PP-MEET or CR-MEET) one time in every 5 min or even shorter without any delay. This accommodates excellent robustness to practical smart microgrid applications.

7. CONCLUSION AND FUTURE WORK

With the integration of distributed energy resources (e.g., solar panels and wind turbine) in contemporary smart grid, microgrids can choose different energy supply to feed its demand, rather than solely relying on the main grid. The newly proposed P2P microgrid energy exchange model (e.g., [2]) enhances the flexibility on the use of excessive energy from peer microgrids that bring system-wide benefits (e.g., reducing the global energy storage loss). However, privacy leakages may occur during the energy exchange – for instance, different microgrids' demand load and energy storage information might be disclosed to the public. Protecting microgrids' privacy should be integrated into the design of P2P energy exchange schemes. To this end, in this paper, we first formulated several novel energy exchange optimization problems that minimize the energy loss during the exchange in different scenarios, and then developed an efficient and privacy-preserving scheme for solving the energy exchange optimization problems without information disclosure. We also extended the privacy-preserving scheme to a collusion-resistant scheme to resolve a potential collusion-based attack. We demonstrated the efficiency and quality of our proposed approaches using experimental results.

The privacy issue in P2P microgrids energy exchange opens more opportunities for us to investigate the sensitive data protection in distributed energy resources environment. In the future, we will first work on the pure malicious model for energy exchange, by assuming that microgrid may behave arbitrarily (e.g., tampering its input data or the received encrypted data and intercepting messages among microgrids on the grid), as well as the key management problem [44] among multiple microgrids. We also intend to develop privacy-preserving energy routing [17] and scheduling [16] mechanisms in our novel P2P energy exchange optimization models.

APPENDIX

Proof of Theorem 1

Proof 1 *Suppose* $\vec{x}\,'^{*} = \left(\vec{x}_1^{\,*}, \dots, \vec{x}_n^{\,*}\right) = \left(Q_1\vec{z}_1^{\,*}, \dots, Q_n\vec{z}_n^{\,*}\right)$ *is not the optimal solution of the MEET problem with artificial/slack variables (Equation (11)). In this case, we have* $\exists\, \vec{v} = \left(\vec{v}_1, \vec{v}_2, \dots, \vec{v}_n\right)$ *such that* $\vec{c}\,'^{T}\vec{v} < \vec{c}\,'^{T}\vec{x}\,'^{*}$*(a better solution), then* $\vec{c}_1\,'^{T}\vec{v}_1 + \cdots + \vec{c}_n\,'^{T}\vec{v}_n < \vec{c}_1\,'^{T}\vec{x}\,'^{*} + \cdots + \vec{c}_n\,'^{T}\vec{x}_n'^{*}$.

Let another vector $\vec{z}\,' = \left(\vec{z}_1', \dots, \vec{z}_n\right) = \left(Q_1^{-1}\vec{v}_1, \dots, Q_n^{-1}\vec{v}_n\right)$; thus, we have $\vec{c}_1\,'^{T}\vec{v}_1 + \cdots + \vec{c}_n\,'^{T}\vec{v}_n = \vec{c}_1^{\,T} Q_1\, \vec{z}_1\,' + \cdots + \vec{c}_n\,'^{T} Q_n\, \vec{z}_n\,' < \vec{c}_1\,'^{T}\vec{x}_1\,'^{*} + \cdots + \vec{c}_n'^{T}\vec{x}_n\,'^{*} \Rightarrow \vec{c}_1\,'^{T} Q_1\, \vec{z}_1\,' + \cdots + \vec{c}_n\,'^{T} Q_n\, \vec{z}_n\,' < \vec{c}_1\,'^{T} Q_1\vec{z}_1^{\,*} + \cdots + \vec{c}_n\,'^{T} Q_n\vec{z}_n^{\,*}$ (since $Q_1^{-1}\vec{x}_1\,'^{*} = \vec{z}_1^{\,*}$, …, and $Q_n^{-1}\vec{x}_n\,'^{*} = \vec{z}_n^{\,*}$).

Hence, $\vec{z}\,'$ *is a better solution of the Shares-Encrypted Form than* $\vec{z}^{\,*}$*. This is a contradiction to that* $\vec{z}^{\,*}$ *is the optimal solution and thus completes the proof.*

Proof of Theorem 2

Proof 2 *It is straightforward to prove Theorem 7 based on Theorem 5 – simply replacing monomial matrices* $Q_1, Q_2, \dots, Q_n$ by $Q_1\prod_{j=2}^{n}Q_{1j}$, $Q_2\prod_{j=3}^{n} Q_{2j}Q_{21}$, $\dots$, $Q_n\prod_{j=2}^{n-1}Q_{nj}Q_{n1}$, *respectively, in the Proof of Theorem 5.*

REFERENCES

1. National institute of standards and technology. NIST framework and roadmap for smart grid interoperability standards, January, 2010.
2. Saad W, Han Z, Poor HV, Basar T. Game-theoretic methods for the smart grid: an overview of microgrid systems, demand-side management, and smart grid communications. *IEEE Signal Processing Magazine* 2012; **29**(5):86–105.
3. Gómez Mármol F, Sorge C, Ugus O, Martínez Pérez G. Do not snoop my habits: preserving privacy in the smart grid. *IEEE Communications Magazine* 2012; **50**(5):166–172.
4. Lisovich MA, Mulligan DK, Wicker SB. Inferring personal information from demand-response systems. *IEEE Security and Privacy* 2010; **8**(1):11–20.
5. Ács G, Castelluccia C. I have a dream! (differentially private smart metering). In *Information Hiding*, 2011; 118–132.
6. Wang S, Cui L, Que J, Choi D-H, Jiang X, Cheng S, Xie L. A randomized response model for privacy preserving smart metering. *IEEE Transactions on Smart Grid* 2012; **3**(3):1317–1324.
7. He X, Zhang X, Kuo C-CJ. A distortion-based approach to privacy-preserving metering in smart grids. *IEEE Practical Innovations*: *Open Solutions*, 1.
8. Rottondi C, Verticale G, Capone A. Privacy-preserving smart metering with multiple data consumers. *Computer Networks*. Elsevier: New York, NY, USA
9. Rottondi C, Verticale G, Krauss C. Distributed privacy-preserving aggregation of metering data in smart grids. *IEEE Journal on Selected Areas in Communications* 2013; **31**(7):1342–1354.
10. Kalogridis G, Cepeda R, Denic SZ, Lewis TA, Efthymiou C. Elecprivacy: evaluating the privacy protection of electricity management algorithms. *IEEE Transactions on Smart Grid* 2011; **2**(4):750–758.
11. McLaughlin SE, McDaniel P, Aiello W. Protecting consumer privacy from electric load monitoring. In

ACM Conference on Computer and Communications Security 2011; 87–98.

12. Tan O, Gündüz D, Poor HV. Increasing smart meter privacy through energy harvesting and storage devices. *IEEE Journal on Selected Areas in Communications* 2013; **31**(7):1331–1341.
13. Yang W, Li N, Qi Y, Qardaji WH, McLaughlin SE, McDaniel P. Minimizing private data disclosures in the smart grid. In *ACM Conference on Computer and Communications Security* 2012; 415–427.
14. Liang H, Choi BJ, Zhuang W, Shen X. Decentralized inverter control in microgrids based on power sharing information through wireless communications. In *Proceedings of Global Communications Conference (GLOBECOM)*. 2012; 5148–5153.
15. Liang H, Choi BJ, Zhuang W, Shen X. Stability enhancement of decentralized inverter control through wireless communications in microgrids. *IEEE Transactions on Smart Grid* 2013; **4**(1):321–331.
16. Zhu T, Huang Z, Sharma A, Su J, Irwin DE, Mishra AK, Menasché DS, Shenoy PJ. Sharing renewable energy in smart microgrids. In *ACM/IEEE 4th International Conference on Cyber-Physical Systems* (*with CPS Week 2013*), *ICCPS '13*, *Philadelphia*, *PA*, *USA*, *April 8-11*, 2013; 219–228.
17. Zhu T, Xiao S, Ping Y, Towsley D, Gong W. A secure energy routing mechanism for sharing renewable energy in smart microgrid. In *SmartGridComm*, pages 143–148, 2011.
18. Duan R, Deconinck G. Multi-agent coordination in market environment for future electricity infrastructure based on microgrids. In *SMC*. 2009; 3959–3964.
19. Maharjan S, Zhu Q, Zhang Y, Gjessing S, Basar T. Dependable demand response management in the smart grid: a stackelberg game approach. *IEEE Transactions on Smart Grid* 2013; **4**(1):120–132.
20. Maity I, Rao S. Simulation and pricing mechanism analysis of a solar-powered electrical microgrid. *IEEE Systems Journal* 2010; **4**(3):275–284.
21. Mohamed FA, Koivo HN. Multiobjective optimization using modified game theory for online management of microgrid. *European Transactions on Electrical Power* 2011; **21**(1):839–854.
22. Saad W, Han Z, Poor HV, Basar T. A noncooperative game for double auction-based energy trading between phevs and distribution grids. In *SmartGridComm*, 2011; 267–272.
23. Wang Z, Yang K, Wang X. Privacy constrained energy management in microgrid systems. In *SmartGridComm*, 2012; 670–674.
24. Wang Z, Yang K, Wang X. Privacy-preserving energy scheduling in microgrid systems. *IEEE Transactions on Smart Grid* 2013; **4**(4):1810–1820.
25. Hong Y. *Privacy-preserving Collaborative Optimization*. PhD thesis, Rutgers University: Newark, NJ, 2013.
26. Li J, Atallah MJ. Secure and private collaborative linear programming. In *Proceedings of the 2nd International Conference on Collaborative Computing*: *Networking*, *Applications and Worksharing*. November 17–20 2006; 1–8.
27. Vaidya J. Privacy-preserving linear programming. In *SAC*. 2009; 2002–2007.
28. Hong Y, Vaidya J, Lu H. Efficient distributed linear programming with limited disclosure. In *DBSec*. 2011; 170–185.
29. Hong Y, Vaidya J, Lu H. Secure and efficient distributed linear programming. *Journal of Computer Security* 2012; **20**(5):583–634.
30. Mangasarian OL. Privacy-preserving linear programming. *Optimization Letters* 2011; **5**(1):165–172.
31. Mangasarian OL. Privacy-preserving horizontally partitioned linear programs. *Optimization Letters* 2012; **6**(3):431–436.
32. Hong Y, Vaidya J, Lu H, Karras P, Goel S. Collaborative search log sanitization: toward differential privacy and boosted utility. In *Proceedings of Global Communications Conference (GLOBECOM)*. 2012; 5148–5153.
33. Hong Y, Vaidya J, Lu H, Shafiq B. Privacy-preserving tabu search for distributed graph coloring. In *SocialCom/PASSAT*. 2011; 951–958.
34. Hong Y, Vaidya J, Lu H, Wang L. Collaboratively solving the traveling salesman problem with limited disclosure. In *Data and Applications Security and Privacy XXVIII - 28th Annual IFIP WG 11.3 Working Conference*, *DBSec 2014*, *Vienna*, *Austria*, *July 14-16*, 2014; 179–194.
35. Sakuma J, Kobayashi S. A genetic algorithm for privacy preserving combinatorial optimization. In *GECCO*, 2007; 1372–1379.
36. Dall'Anese E, Zhu H, Giannakis GB. Distributed optimal power flow for smart microgrids. *IEEE Transactions on Smart Grid* 2013; **4**(3):1464–1475.
37. Hong Y, Vaidya J. An inference-proof approach to privacy-preserving horizontally partitioned linear programs. *Optimization Letters* 2014; **8**(1):267–277.
38. Sobe A, Elmenreich W. Smart microgrids: overview and outlook. *CoRR*, abs/1304.3944, 2013.
39. Du W. A study of several specific secure two-party computation problems. PhD thesis, Purdue University, West Lafayette, Indiana, 2001.
40. Bednarz A, Bean N, Roughan M. Hiccups on the road to privacy-preserving linear programming. In *Proceedings of the 8th ACM workshop on Privacy in the electronic society*, WPES '09. ACM: New York, NY, USA. 2009; 117–120.

41. Goldreich O. *The Foundations of Cryptography*. volume 2, chapter Encryption Schemes. Cambridge University Press, Elsevier: New York, NY, USA, 2004.
42. Yao AC. How to generate and exchange secrets. In *Proceedings of the 27th IEEE Symposium on Foundations of Computer Science*, pages 162–167, Los Alamitos, CA, USA, 1986. IEEE, IEEE Computer Society.
43. Damgård I, Jurik M. A length-flexible threshold cryptosystem with applications. In *ACISP*, 2003; 350–364.
44. Long X, Tipper D, Qian Y. An advanced key management scheme for secure smart grid communications. In *IEEE Fourth International Conference on Smart Grid Communications, SmartGridComm 2013, Vancouver, BC, Canada, October 21-24*, 2013; 504–509.