# Distributed Multi-Unit Privacy Assured Bidding (PAB) for Smart Grid Demand Response Programs

Muhammed Fatih Balli, Suleyman Uludag, *Member, IEEE,* Ali Aydin Selcuk, and Bulent Tavli *Member, IEEE,*

***Abstract*—The stringent requirement of the demand-supply equilibrium for delivering electricity has traditionally been dealt with a supply-side perspective, assuming that the demand is not alterable. With the promises of the Smart Grid, demand-side management techniques are increasingly becoming more feasible. A demand-side management technique, called Demand Response, aims at inducing changes in electricity load in response to financial incentives, some of which involve bidding as the underlying facilitator. It is well-established that the effectiveness of the DR is proportional to the number of participants. Yet, many of the DR programs, including those involving bidding, may suffer due to consumer privacy concerns. Within this context, in this paper, we propose a distributed and multi-unit privacy guaranteeing bidding mechanism as part of a DR program without relying on any third party, trusted or not, to protect the participants' bidding information, except obviously for the winning price and the winner exposed to the utility. To the best of our knowledge, this is the first such approach for the DR bidding programs. We provide a security analysis of our approach under the honest-but-curious and active adversary assumptions and prove the privacy assuring property.**

***Index Terms*—Smart Grid Privacy, Demand Response, Demand Response Bidding Privacy, Bidding Privacy without Trusted Third Party.**

## I. Introduction

A major paradigm shift in the generation, transmission, and distribution of electricity has been gaining more momentum than ever under the umbrella term of Smart Grid (SG) [1]–[3]. Electricity service has a distinctive characteristic that requires the maintenance of the supply and demand equilibrium at all times. The loss of this equilibrium may result in regulatory intervention, cost increases, and/or frequency instabilities. With the infeasibility of its storage, the generated power must be consumed rapidly to avoid any complications in the infrastructure. The conventional mechanism to cope with this intrinsically required equilibrium has been through adjusting the supply side since demand has been assumed to be non-manipulatable. Demand-side techniques have been gaining more attention with advances in the computing and communications technologies [4], [5]. These approaches, generally referred to as Demand Response (DR) programs, are at the same time a key facilitator of the SG to induce change/shift in electricity consumption to restore the demand-supply equilibrium under perilous conditions or during imbalance periods.

Wild fluctuations of the real demand for power are either suppressed or distended to achieve a smoother and more desirable *effective demand*, which in turn has dramatic effects on the price of the electricity produced. The aggregate values of effective and real demand may be identical, if only demand shifting mechanisms are utilized, or different, if demand suppression mechanisms are employed. Thus, DR may reduce new capital expenditures in generation, transmission, and distribution [6]. See [7], [8] for more detailed discussion of financial and performance benefits.

DR programs include incentives, tariffs, and programs (among other mechanisms). Many DR programs include some form or shape of bidding in their implementations. For example, some expose Real Time Pricing (RTP) [7], [9] to end users and some are based on customers bidding for energy usage, such as demand bidding [8], [10] where customers bid for incentives to alter load. Upcoming emissions trading [11], ancillary services market program [8], distribution automation load management, electric vehicle charging/discharging, retail power electricity market [11], [12] are all expected to include some bidding mechanism.

A very basic *sine qua non* of any DR is the generation, transmission, storage, maintenance, and analysis of unprecedented amount of data through smart meters in the Advanced Metering Infrastructure (AMI). An inevitable consequence of such abundance of data is the ease of extraction of Personally Identifiable Information (PII) for potential abuse or misuse, such as behavioral inferences, deduction of individual habits or activities [11], [13]–[17]. Bidding as part of DR programs in the SG is thus subject to privacy concerns.

In this paper, we propose a novel system to provide a distributed, privacy-guaranteeing bidding protocol involving only a service provider and bidding customers without the need for the involvement of any other entities such as a trusted third party. At the end of our proposed bidding process, only the winning bidder is disclosed to the service provider while neither the bidders nor the service provider can learn the other customers' private bidding prices to infer any private information. To the best of our knowledge, ours is the first privacy-preserving protocol proposed in the SG energy bidding process where no third party entity is involved to minimize the exposure of the private information.

The rest of the paper is organized as follows: Section II summarizes the related work. A synopsis of our proposed approach in terms of the linear expressions is provided in Section III

M. F. Balli is with Computer and Communication Sciences, Ecole Polytechnique Federale de Lausanne, Switzerland, E-mail: fatih.balli@epfl.ch

S. Uludag is with the Department of Computer Science, Engineering and Physics, University of Michigan - Flint, MI, USA. E-mail: uludag@umich.edu

A. A. Selcuk is with the Department of Computer Engineering, TOBB University of Economics and Technology, Ankara, Turkey, E-mail: aselcuk@etu.edu.tr

B. Tavli is with the Department of Electrical and Electronics Engineering, TOBB University of Economics and Technology, Ankara, Turkey. E-mail: btavli@etu.edu.tr

with an illustrative toy example. Section IV provides the full cryptographic details built on Elgamal encryption. A security analysis with Zero-Knowledge Proofs (ZKPs) describe the privacy assuring features of our approach in Section V. Section VI concludes the paper.

## II. Related Work

The awareness and sensitivity of the public have been increasing on privacy issues due partly to such recent news as the European Court of Justice's invalidation of Safe Harbor Law, Wikileaks, US NSA leaks by Edward Snowden, Facebook's recent disclosure of Emotion Experiment, and EU's recent ruling on "right to be forgotten." It is within the same line of interest that privacy dimension of the DR initiatives needs to be addressed.

A succinct definition of privacy may stated as the "the right to be left alone" [18]. In more general terms, our work falls into the domain of privacy-enhancing technologies (PET) as coined by Chaum in 1995 [19] and defined in [20]:

> PET stands for a coherent system of Information and communications technology (ICT) measures that protects privacy by eliminating or reducing personal data or by preventing unnecessary and/or undesired processing of personal data, all without losing the functionality of the information system.

The main focus of SG related privacy studies has been placed on the smart meter data collection for monitoring and billing [21] in terms of perturbing, anonymizing, minimizing, and/or obfuscating the transmission. Privacy of behavior, action, lifestyle, presence/absence, and number of persons may be derived from the smart meter data. The inviolability of family life and homes is in danger. A report by Dutch Consumers' Association concluded that smart meters would violate article 8 of the the European Convention of Human Rights [17]. In the DR context, Customer Energy Usage Data (CEUD) as well as any other personally identifiable data from collection, transmission, aggregation, dissemination, and analysis should also be included as part of the privacy studies. Similar to potential inferences that can be drawn from CEUD, bidding information may reveal consumer's behavior, which may be personally objectionable or outright unlawful in some countries [17]. Yet, DR privacy, and bidding in particular, cannot make use of the existing privacy techniques [21] directly as accuracy is crucially essential for acceptable operation and has not received much attention.

OpenADR (Automated Demand Response) [22], [23] is an open standard communications specification to relay DR signals back and forth among the participating entities. OpenADR version 1 specifies a Demand Response Automation Server (DRAS) as entity that manages notification, participation as well as final rewarding of customers whenever DR-incentive is requested by the utility provider. With that provision, Paverd *et al.* suggest in [24], [25] a form of a trusted third party relaying entity, called Trusted Remote Entity (TRE), that is mainly responsible for anonymization of customers in DR-bidding to prevent utility provider having direct access to customer private bidding information. TRE is also responsible for billing of customers and consumption data in real time. In fact, privacy is established in the presence of mutual distrust between the utility and customers by means of a trusted platform module (TPM).

Karwe *et al.* [26] similarly focus on interactive DR-demands, stating that a semi-honest DRAS may easily compromise the privacy of customers. They propose enhanced functionality for DRAS, the trusted third party, to hide consumption profiles of customers. DRAS is responsible for pseudonymization of real identities to protect customer privacy from utility provider, whereas utility and customers use attribute based encryption (ABE) to hinder DRAS from looking into private data.

Gong *et al.* [27] propose a solution in which identity of customers are linked with pseudonyms at the proxy and incentive based DR is implemented on these anonymous accounts. They incorporate identity-committable signatures, ZKPs and partially blind signatures to prevent malicious activity with these anonymous parties. However, they also rely on a trusted third party, called demand response provider, who is capable of tracking the bidding history of any participant for future potential privacy violations. In our scheme, bidding history of the participants cannot be tracked and no profiling may take place by any third-party, except obviously for the winner that must be known to the utility.

Similarly, Rahman *et al.* [28] propose a privacy solution for incentive based DR. While the authors claim to establish a bidding process without a trusted third party, the Bidding Manager (BM) in their setup acts as an identity escrow agency, implying a level of trust and hence a notion of trusted third party in the scheme. Further, any coalition between BM and Relaying Manager (RM) leaves the customer privacy exposed. That is, BM and RM are capable of keeping track of bidding history of the participants for future potential privacy violations. Finally, it is unclear whether their bidding scenario works for single-winner or multi-winner and how winners are chosen in a verifiable manner.

In the aforementioned studies, the deployment of an intermediary or a trusted third party[1], be it TRE, proxy, BM or DRAS, proliferates the entities involved and potential attack vectors, and increases the uneasiness of, at the very least, the privacy-conscious customers, if not a larger population (*i.e.*, the possibility of existence of a weak chain in privacy protection system deters some customers from taking place in bidding processes). Please note that there is nothing required in our scheme to eliminate intermediaries. A hierarchical system would just be fine with these intermediaries to continue providing their useful function. We are just hiding irrelevant information from the utility or utility-like entities (such as

[1]It is a generally known fact that companies are after consumer data to be able to market more products and bombard them with more other advertisements. The consumers are profiled from these extracted data. The tracked information includes habits, patterns, behavior, location, demographics, etc. Bidding history is a useful piece of information to these profiling activities. This is what we are shielding from the utility company in the bidding process, of course except for the winner. Thus, it is perfectly fine with our scheme to have intermediaries (or aggregators) to act like a utility in a hierarchical bidding system. What we are providing is the privacy protection to the customers against these intermediaries without eliminating them.

intermediaries) from accessing, and hence collecting, more information than necessary in line with the generally accepted security best practices of *the principle of the least privileged*.

To the best of our knowledge, the only other privacy-guaranteeing DR bidding scheme proposed in the literature that truly does not rely on a third party in any shape or form is reported in our earlier work [29], which has a high computational complexity in a single-winner based auction system. In this paper, we present our novel privacy-assured bidding (PAB) solution for the SG DR by augmenting our aforementioned solution with a multi-unit, multi-winner auction algorithm for both the customers and the utility company. As part of the security analysis, we provide Zero-Knowledge Proofs to show that our algorithms can guarantee the privacy under different threat models.

Such an approach without a trusted third party is likely to be a key argument in the post-Snowden era in allaying customers' fears about privacy violations and/or in recruiting more customers into DR programs, which is critical in long-term success and sustainability of such initiatives.

## III. BASIC ALGORITHM FOR PRIVACY ASSURED BIDDING

Our auction can be classified as a multi-unit, multi-winner, and single-price auction. That is, in our scheme, the utility starts the bidding process by announcing the relevant parameters, namely the price vector and the total number of units being auctioned. Interested customers enter into the biding by specifying the units demanded and the bidding price. It is always possible for customers to stay out of the bidding process and to take advantage of the standard tariff. At the end of the bidding, winners and losers[2] together with the winning price is decided. Winners get the exact quantity they originally sought at the price they specified or lower.

In this section, we describe fundamental linear operations of our approach without emphasizing the cryptographic dimension in order to simplify the notation and the narrative. The motivation is to build a linear system, that fits the auction description above, in which participants can submit their bids, distribute the available number of units from highest to lowest bid offer, and determine their individual outcomes through a given linear function. By restricting our system to linear operations, we will be able to make use of the homomorphic properties of Elgamal encryption, as described in Section IV.

### A. Mathematical Formulation

The notation table for our approach is given in Table I. We use indices $i$ or $a$ to imply that an element is related to customer $i$ or $a$, respectively, and similarly $j$ to imply that particular element corresponds to the $j$-th price.

[2]DR-bidding mechanism envisions shifting the peak consumption to relaxed hours. The energy auctioned is not the total energy delivered by the utility companies. The bidding is a method in the overall demand response mechanism where we would like to have some demand-side attempts to maintain the load-supply equilibrium, especially during periods of higher-than-normal risk. As such, that is not the main means of delivering energy, nor it is the sole methodology. Thus, losing an auction is just missing the opportunity to take advantage of the incentives or financial compensation offered in exchange for a change or shift in demand; it does not mean the loser will be left without energy.

TABLE I
NOTATIONS USED IN OUR APPROACH.

| | |
|---|---|
| $n$ | Number of customers participating in bidding |
| $k$ | Number of acceptable price values in bidding |
| $i$ | Generic index ranging from 1 to $n$ |
| $j$ | Generic index ranging from 1 to $k$ |
| $a$ | Secondary index ranging from 1 to $n$ |
| $\boldsymbol{\pi}$ | The descending price vector with $k$ discrete values |
| $M$ | Pre-announced available number of units |
| $V$ | Maximum number of units each customer can demand |
| $\mathbf{b}_a$ | The bid vector for customer $a$ |
| $B$ | The matrix to represent all bid vectors together |
| $\mathbf{d}$ | Cumulative demand vector |
| $\mathbf{c}$ | Winning price indicator created by the utility |
| $bid_a$ | The index of bidding price submitted by customer $a$ |
| $unit_a$ | The number of units demanded by customer $a$ |
| $x_a$ | The private key of customer $a$ |
| $y_a$ | The public key of customer $a$ |
| y | The master public key jointly generated by all customers |

We base the principal protocol design of our approach on the idea described in [30], [31]. $M$ is the total number of units being sold, e.g. the amount of energy units, $V$ corresponds to the maximum number of units each customer can demand, e.g. 50KWh, and $n$ is the number of customers participating, i.e. electric consumers. Also, $\boldsymbol{\pi}$ is a price vector of discrete values sorted in descending order, e.g. $[60, 50, 40, 30]$, defined by the utility, and $k$ is the size of $\boldsymbol{\pi}$, i.e. $k = |\boldsymbol{\pi}|$:

$$\boldsymbol{\pi} = [\pi_1\ \pi_2\ \pi_3\ \cdots\ \pi_k]$$

Let $L_\ell$, $U_\ell$, $I_\ell$ denote the $\ell \times \ell$ lower triangular, upper triangular and identity matrices, respectively. Furthermore, let $R^*_\ell$ denote an $\ell \times \ell$ "randomization matrix",

$$R^*_\ell = \begin{bmatrix} * & 0 & \cdots & 0 \\ 0 & * & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & * \end{bmatrix}$$

whose diagonal entries $*$ are jointly-generated random numbers unknown to any single customer. This random matrix is used upon the termination of the protocol to guarantee that no private information is leaked, as explained later.

Whenever a new DR bidding is initiated by the utility, along with the announcement of the parameters $(\boldsymbol{\pi}, M, V)$, each customer $i$ chooses an element with index $bid_i$ from the price vector $\boldsymbol{\pi}$ and the number of units $unit_i$ to buy at $\boldsymbol{\pi}_{bid_i}$ price, such that $1 \leq unit_i \leq V$ and $1 \leq bid_i \leq k$. Then, each customer creates a bid vector denoted by $\mathbf{b}_i$ such that $|\mathbf{b}_i| = k = |\boldsymbol{\pi}|$, which consists of $(k-1)$ 0s, and $unit_i$ at index $bid_i$, as given below:

$$\mathbf{b}_i = \begin{bmatrix} 0 & \cdots & 0 & unit_i & 0 & \cdots & 0 \end{bmatrix}^T$$

which states that the customer is bidding to purchase $unit_i$ many units of energy at up to a price of $\boldsymbol{\pi}_{bid_i}$. In order to simplify the notation, we define $B$ as a matrix whose columns correspond to the bid vectors of customers, i.e. $B = [\mathbf{b}_1|\mathbf{b}_2|\cdots|\mathbf{b}_n]$. Now that any customer $i$ has his own

bid vector placed in $i$-th column of matrix $B$, the cumulative demand vector is defined as $\mathbf{d}$:

$$\mathbf{d} = \sum_{i=1}^{n} \mathbf{b}_i \tag{1}$$

The first goal of the utility is to find the maximum index $t$ that satisfies the inequalities for the pre-announced available amount $M$:

$$\sum_{j=1}^{t} d_j \leq M < \sum_{j=1}^{t+1} d_j \tag{2}$$

After finding $t$, the utility creates a vector, denoted by $\mathbf{c}$, that consists of $(k-1)$ 1s, and one 0 at the $t$-th entry only.

$$\mathbf{c} = \begin{bmatrix} 1 & \cdots & 1 & 0 & 1 & \cdots & 1 \end{bmatrix}^T \tag{3}$$

The final outcome function for each customer $i$ is given below, with which each customer can infer whether he won, and if so, at what price. It is important to randomize non-zero elements of the outcome function so that a customer can only infer his win/loss status.

$$f_i(B) = R_k^* \left( \mathbf{c} + (U_k - I_k)\, \mathbf{b}_i \right) \tag{4}$$

By marking the indices except $t$ with ones in $\mathbf{c}$, we are masking the other bid values, so that any winner concludes this price as the final. Finally, each customer $i$ checks the result of $f_i(B)$ to find whether there is a 0 in it or not. The latter simply means the customer lost the bidding, whereas the former indicates that he won and can determine the price by using the position of 0 in $f_i(B)$ that corresponds to the actual price in $\boldsymbol{\pi}$. Note that in this multi-unit multi-winner auction system, the available units are distributed among the bidders who offered one of the highest $t$ prices in the price vector $\boldsymbol{\pi}$. This implies that there might be some unsold units from available pool at the end.

Our focus in this paper is in the privacy-protection of the bidding process. There are various different auction methodologies in the literature (*e.g.*, English Auction, Dutch Auction, Vickery Auction). Our goal is not to optimize the bidding process nor to pick one of the aforementioned auction mechanisms. The distribution of units algorithm in our protocol is such that the available number of units are distributed among the highest $t$ price bidders depending on the available units $M$. We thereby assume a simple and fair auctioning mechanism as the underlying bidding in order to provide a cryptographically secure and private protocol. We do not claim any optimality in that sense, but we assure that bidder privacy is protected. Nevertheless, our protocol can be modified fairly easily to accommodate a wide range of bidding protocols in the removal of dependency on trusted third parties to achieve privacy preservation.

### *B. Toy Example*

As an example, below are the three bid vectors for three customers; for the price vector $\boldsymbol{\pi} = [60\ 50\ 40\ 30]$, where $M$ is defined as 6 units. From $n = 3$ customers in total, customer 1 wants to buy 3 units at a price of up to \$60, customer 2 wants to buy 2 units at a price of up to \$50, and customer 3 wants 4 units up to \$40. They create their $\mathbf{b}_i$ vectors as follows:

$$\mathbf{b}_1 = \begin{bmatrix} 3 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \mathbf{b}_2 = \begin{bmatrix} 0 \\ 2 \\ 0 \\ 0 \end{bmatrix}, \mathbf{b}_3 = \begin{bmatrix} 0 \\ 0 \\ 4 \\ 0 \end{bmatrix}$$

Then, customers should jointly calculate $\mathbf{d}$:

$$\mathbf{d} = \sum_{i=1}^{n} \mathbf{b}_i = \begin{bmatrix} 3 \\ 2 \\ 4 \\ 0 \end{bmatrix}$$

Now the utility finds $t = 2$ as stated in Equation (2) and creates the $\mathbf{c}$ vector.

$$\mathbf{c} = \begin{bmatrix} 1 & 0 & 1 & 1 \end{bmatrix}^T$$

From this we can easily compute the outcome function $f_i$ for each customer as shown below:

$$f_1(B) = R_{k_1}^* \left( \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 3 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right) = \begin{bmatrix} * \\ 0 \\ * \\ * \end{bmatrix}$$

$$f_2(B) = R_{k_2}^* \left( \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \\ 0 \\ 0 \end{bmatrix} \right) = \begin{bmatrix} * \\ 0 \\ * \\ * \end{bmatrix}$$

$$f_3(B) = R_{k_3}^* \left( \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 4 \\ 0 \end{bmatrix} \right) = \begin{bmatrix} * \\ * \\ * \\ * \end{bmatrix}$$

Note that each * denotes a uniform non-zero random value independent of others, and is used to mask non-zero values so a customer cannot infer anything about his rivals. At the end, customer 1 and customer 2 concludes that they won the bidding with the price of \$50 with the quantities they submitted. Customer 3 only realizes that he lost.

The operations mentioned so far actually consist of three phases:

i. Customers jointly calculate $\mathbf{d}$ from the $\mathbf{b}_i$ vectors.
ii. The utility determines $t$ from $\mathbf{d}$, and creates $\mathbf{c}$.
iii. Customers calculate their outcome $f_i(B)$ by using $\mathbf{c}$.

Note that the first and third operations are linear and may be performed in the exponent as part of the Elgamal encryption to be described in the next section by exploiting the homomorphic property. They are performed as separate rounds in the multiparty computation to be detailed below. However, the second operation requires a full knowledge and disclosure of the complete vector $\mathbf{d}$. Thus, we assume that the utility will have an access to that information in order to carry out that phase of the operations.

Please note that the matrix representations we used so far are chosen to simplify the formal description and imply linearity

of operations in our protocol, but the real implementation of the protocol replaces the matrix operations with the cryptographic operations described in the following section.

## IV. Privacy Assured Bidding Protocol

In this section, we enhance the algorithm presented in the previous section with cryptographic operations using Elgamal encryption, thereby define our protocol.

### A. Preliminaries

Elgamal encryption is one of the best-known public key cryptosystems after RSA. It has a simple and elegant mathematical structure that allows such operations as distributed key generation and homomorphic encryption. In a prelude to the explanation of our approach, we below summarize the Elgamal encryption algorithm and its various relevant features. All the arithmetic is carried out in $\mathbb{Z}_p$ unless otherwise stated.

**Elgamal Cryptosystem:** Let $p$ be a large prime, and $g$ be an element of order $q$ in $\mathbb{Z}_p^*$, for some large prime $q\,|\,(p-1)$. Alice chooses a random $x \in \{1,2,\ldots,q-1\}$ as her private key, and $y = g^x \bmod p$ is her public key. To encrypt a message $\mu$ for Alice, Bob chooses a random $r \in \{1,2,\ldots,q-1\}$ as a one-time secret, and computes the ciphertext $(\alpha,\beta)$ as $\alpha = \mu y^r \bmod p$ and $\beta = g^r \bmod p$ (i.e., the message is masked by $g^{xr} \bmod p$). Alice decrypts the message by recomputing the masking factor by $g^{xr} = \beta^x$ and removing it from the message: $\mu = \alpha(\beta^x)^{-1}$.

**Homomorphic Encryption:** Elgamal encryption is homomorphic according to multiplication: Given $E(\mu_1) = (\alpha_1,\beta_1)$ and $E(\mu_2) = (\alpha_2,\beta_2)$, we can compute the encryption of $\mu_1\mu_2$ by $E(\mu_1\mu_2) = (\alpha_1\alpha_2, \beta_1\beta_2)$.

**Distributed Key Generation:** A common Elgamal public/private key pair can be generated by a group of participants by each participant generating a part of the key: Each party $i$ generates his partial private and public key $(x_i, y_i = g^{x_i})$ and broadcasts the public key $y_i$ to the group. The master public key is $y = \prod_i y_i$. The private key $x = \sum_i x_i \bmod q$ is held in a distributed fashion by the group where party $i$ has share $x_i$.

**Distributed Decryption:** Let $(\alpha,\beta)$ be a ciphertext encrypted under a public key $y = g^x$, where the private key $x = \sum_i x_i \bmod q$ is distributed among a group of participants with user $i$ having $x_i$. The message can be decrypted collectively, without any participant revealing his secret share: User $i$ computes and broadcasts $\phi_i = \beta^{x_i}$. One participant combines these partial results and decrypts the message as $\mu = \alpha(\prod_i \phi_i)^{-1}$.

**Distributed Randomization:** The homomorphic property of Elgamal encryption enables a group of users to randomize an encrypted message while the randomization factor is unknown to any user: For a given ciphertext $(\alpha,\beta)$, each party picks his randomization parameter $m_i$, calculates and broadcasts $(\alpha_i = \alpha^{m_i}, \beta_i = \beta^{m_i})$. The randomized ciphertext is calculated as $(\alpha^m = \prod_i \alpha_i, \beta^m = \prod_i \beta_i)$, for $m = \sum_i m_i$.

### B. Bidding Privacy Using Multiparty Computation

Secure multi-party computation (MPC) is used to compute a function collectively by a number of participants such that, in the end, no participant can learn anything except its own input and the result [32]. In this section, we give an MPC protocol that calculates the outcome of the auction described in Section III privately.

We first give a high-level description of our protocol and then the details:

1) Each customer $a$ generates a public-private key pair $(x_a, y_a)$ and broadcasts the $y_a$ value.
2) The common group public key $y = \prod_{i=1}^n y_i \bmod p$ is calculated by all, and the group private key $x = \sum_{i=1}^n x_i \bmod q$ is composed of partial private keys held by the customers, none of which may construct the full group private key.
3) Each customer $a$ generates his bid vector $\mathbf{b}_a$, encrypts it by the group public key, and broadcasts.
4) Each customer calculates a partial decryption factor using his share of the private key, and sends it to the utility.
5) The utility decrypts the cumulative demand vector $\mathbf{d}$ using the partial results computed by the customers. Then the utility calculates the $\mathbf{c}$ vector, encrypts it by the group public key, and broadcasts.
6) Each customer calculates his auction function in Equation (4) in the exponent and broadcasts the result.
7) Each customer calculates a set of partial decryption factors over the values broadcast in the previous step, and sends them to the utility.
8) The utility shares these partial results selectively such that each customer can calculate the auction result privately, without learning anything extra.

We now describe the protocol in detail. The calculations are in $\mathbb{Z}_p^*$:

**Round 1:** Customer $a$ generates a private key $x_a$, its public key $y_a = g^{x_a}$, a random vector $\mathbf{r}_a$, and a matrix of random values $m_{ij}^{(a)}, 1 \le i \le n, 1 \le j \le k$. Then, he broadcasts his partial public key $y_a$ and calculates the common public key $y$, by using the partial public keys of the others:

$$y = \prod_{i=1}^{n} y_i$$

**Round 2:** Customer $a$ encrypts his bid vector $\mathbf{b}_a$,

$$\alpha_{aj} = g^{b_{aj}} \cdot y^{r_{aj}} \qquad\qquad \beta_{aj} = g^{r_{aj}}$$

for $1 \le j \le k$, and broadcasts it.

**Round 3:** Each customer calculates, for $1 \le j \le k$,

$$\alpha'_j = \prod_{i=1}^{n} \alpha_{ij} \ , \qquad\qquad \beta'_j = \prod_{i=1}^{n} \beta_{ij} \ .$$

**Round 4:** Customer $a$ calculates and sends his partial decryption factor $\phi'_{aj}$, for $1 \le j \le k$, to the utility privately (over Transport Layer Security –TLS–) so that the utility can decrypt $\mathbf{d}$:

$$\phi'_{aj} = \beta'^{x_a}_j$$

**Round 5:** The utility computes, for $1 \le j \le k$,

$$g^{d_j} = \frac{\alpha'_j}{\prod_{i=1}^{n} \phi'_{ij}} \ .$$

In order to obtain **d**, the utility needs to extract each $d_j$ by solving a discrete logarithm with an upper bound of $nV$. This problem can be solved practically by "square-root" methods such as Shank's baby-step-giant-step [33] and Pollard's kangaroo [34]. For cases where $k \gg nV$, the algorithm described in [35] can be preferred.

After calculating **d**, the utility determines the index $t$ and creates the **c** vector as described in Equations (2) and (3).

**Round 6:** The utility encrypts **c** using random $r_{uj}$ to calculate $\alpha_{uj}$, for $1 \leq j \leq k$, where $u$ denotes the utility:

$$\alpha_{uj} = g^{c_j} \cdot y^{r_{uj}} \qquad \beta_{uj} = g^{r_{uj}}$$

**Round 7:** Customer $a$ executes the linear operation given in Equation (4) in the exponent, randomized by $m_{ij}^{(a)}$,

$$\gamma_{ij}^{(a)} = \left(\alpha_{uj} \cdot \prod_{q=j+1}^{k} \alpha_{iq}\right)^{m_{ij}^{(a)}} \quad \sigma_{ij}^{(a)} = \left(\beta_{uj} \cdot \prod_{q=j+1}^{k} \beta_{iq}\right)^{m_{ij}^{(a)}}$$

for $1 \leq i \leq n, 1 \leq j \leq k$, and broadcasts the result.

**Round 8:** Customer $a$ calculates and sends his decryption factors $\phi_{ij}^{(a)}$, for $1 \leq i \leq n, 1 \leq j \leq k$, to the utility privately (over TLS):

$$\phi_{ij}^{(a)} = \prod_{q=1}^{n} \left(\sigma_{ij}^{(q)}\right)^{x_a}$$

The utility broadcasts all $\phi_{ij}^{(a)}$ parameters for $1 \leq i \leq n, 1 \leq j \leq k, 1 \leq a \leq n$, except $a = i$. By doing so, the utility guarantees that only the customer $a$ himself can compute the function $f_a(B)$.

**Round 9:** Finally, customer $a$ does the following component-wise computation as the final operation to compute his individual $f_a(B)$ to check his win/loss status:

$$g^{f_a(B)_j} = \frac{\prod_{i=1}^{n} \gamma_{aj}^{(i)}}{\prod_{i=1}^{n} \phi_{aj}^{(i)}}$$

At this point, customer $a$ checks the final result in the exponent to see whether the vector $g^{f_a(B)}$ contains a 1, which corresponds to having a 0 in $f_a(B)$. If there is a 1 in the result, the position of 1 is used to determine the unit price from the $\boldsymbol{\pi}$ price vector. Note that, the utility also computes $f_a(B)$ function for each customer $a$ to determine the winners.

In the end, the only extra information available to the utility is the cumulative demand vector **d**, which corresponds to total demand versus price information pertaining to current bidding process. Note that this vector does not contain any private information about any customer. As such, making the total demand vector **d** transparent does not compromise any privacy notion with respect to the customers. Further, it may facilitate a useful mechanism for the customers for their future bids as a historical data point. Utility's access to this information is also beneficial in the sense that it will know what the current market conditions are.

The most performance demanding aspect of our algorithm comes from the Round 7, in which randomization of elements are performed. This requires computations with matrices that have $n^2k$ elements. Thus our algorithm scales with $O(n^2k)$. We believe our algorithm can be easily implemented for bidding in low-cost embedded devices. Please also note that the bidding algorithm we hereby propose is not expected to run in a real-time scenario. The time-granularity is likely to be in a similar magnitude of real-time pricing updates from the utility, which is typically in 15 minutes or so. When the time scale is considered within these parameters, our algorithm is significantly well within the expected frequency of such mechanisms and in an acceptable time frame.

## V. Security Analysis

In the previous section, we have defined our basic protocol which is secure against *honest but curious (HbC)* adversaries, i.e., a party who acts honestly and performs the prescribed operation exactly, but may go beyond what is expected of him and tries to extract privacy-violating information by acting curiously. In this section, we enhance our protocol to be secure against fully malicious adversaries [36] who do not necessarily follow the protocol or create their inputs as specified. Later, we also discuss a hybrid version which is more efficient than the fully secure version.

We employ zero-knowledge proofs (ZKPs) to verify that expected operations are honestly performed. ZKPs enable us to verify the integrity of the protocol through validation without forcing parties to disclose their private information. In what follows, we first elaborate on four ZKPs. Building on top of these four ZKPs, we go on to provide the security analysis to prove that privacy of customer bids are guaranteed in our approach.

### A. Zero-Knowledge Proofs

In what follows, we drop *mod p* from the notations to reduce the clutter. As before, all the arithmetic is in mod $p$ (in $\mathbb{Z}_p^*$) unless otherwise stated.

`ZK1`: Alice has $y = g^x$ and wants to prove her knowledge of $x$ to Bob, without disclosing $x$, where $y$ and $g$ are publicly known [37].

i. Alice picks a random $z$, sends $g^z$ to Bob.
ii. Bob sends a random $c$ as a challenge to Alice.
iii. Alice sends $r = (z + xc) \bmod q$ to Bob.
iv. Bob checks if $g^r = g^z y^c$.

`ZK2`: Alice has $y_1 = g_1^x$ and $y_2 = g_2^x$ and wants to prove equality and knowledge of discrete logarithm $\log_{g_1} y_1 = \log_{g_2} y_2 = x$, without disclosing $x$, where $y_1$, $y_2$, $g_1$ and $g_2$ are public [38].

i. Alice picks $z$ randomly and sends $g_1^z$, $g_2^z$ to Bob.
ii. Bob sends a random $c$ as challenge to Alice.
iii. Alice sends $r = (z + xc) \bmod q$ to Bob.
iv. Bob verifies the equality of two discrete logarithm by checking both $g_1^r = g_1^z y_1^c$ and $g_2^r = g_2^z y_2^c$.

`ZK3`: Alice has $(\alpha, \beta) = (my^r, g^r)$ and wants to prove that either $m = z$ or $m = 1$ without disclosing which one it is. [39]

i. If $m = 1$, Alice chooses $r_1$, $d_1$, $w$ at random and sends $(\alpha, \beta)$, $a_1 = g^{r_1} \beta^{d_1}$, $b_1 = y^{r_1} (\alpha / z)^{d_1}$ and $a_2 = g^w$, $b_2 = y^w$ to Bob.

If $m = z$, Alice chooses $r_2$, $d_2$, $w$ at random and sends $(\alpha, \beta)$, $a_1 = g^w$, $b_1 = y^w$, $a_2 = g^{r_2}\beta^{d_2}$, $b_2 = y^{r_2}\alpha^{d_2}$ to Bob.

ii. Bob sends a random $c$ as challenge to Alice.

iii. If $m = 1$, Alice sends $d_1$, $d_2 = c - d_1 \bmod q$, $r_1$ and $r_2 = w - rd_2 \bmod q$ to Bob.
If $m = z$, Alice sends $d_1 = c - d_2 \bmod q$, $d_2$, $r_1 = w - rd_1 \bmod q$, and $r_2$ to Bob.

iv. Bob verifies that encrypted value $(\alpha, \beta)$ is either 1 or $z$ by checking $c = d_1 + d_2 \bmod q$, $a_1 = g^{r_1}\beta^{d_1}$, $b_1 = y^{r_1}(\alpha/z)^{d_1}$, $a_2 = g^{r_2}\beta^{d_2}$ and $b_2 = y^{r_2}\alpha^{d_2}$.

`ZK4`: Alice wants to prove that her encrypted element $(\alpha, \beta) = (g^c y^r, g^r)$ decrypts to $g^c$, where $r$ is ephemeral key (one-time key) and $c \in [0, 2^l - 1]$, where $l$ is an arbitrary system parameter.

i. Alice writes down $c$ in binary form, e.g. $c = c_{l-1} \cdots c_1 c_0$ each $c_t$ being either 1 or 0 for $t \in [0, l-1]$. Then, she creates encrypted elements $(\alpha'_{l-1}, \beta'_{l-1})$, $(\alpha'_{l-2}, \beta'_{l-2})$, $\cdots$, $(\alpha'_0, \beta'_0)$ such that each element satisfies $(g^{2^t c_t} y^{r'_t}, g^{r'_t}) = (\alpha'_t, \beta'_t)$ as well as ephemeral keys satisfying $r = \sum r'_t$. Then, she sends new encrypted elements $(\alpha'_t, \beta'_t)$ to Bob.

ii. For each element, Alice proves each $(\alpha'_t, \beta'_t)$ decrypts to either 1 or $g^{2^t}$ with `ZK3`.

iii. Bob checks if $\prod \alpha'_t = \alpha$ and $\prod \beta'_t = \beta$, along with `ZK3` proofs for each $(\alpha'_t, \beta'_t)$ pair, and confirms that $(\alpha, \beta)$ indeed corresponds to $g^c$ where $c \in [0, 2^{l-1}]$.

### B. Active Adversaries

The protocol we described in Section IV so far assumes that each customer follows the procedure rounds given in Section III honestly, leaving possible vulnerabilities. Therefore, the protocol is required to be secured against active adversaries, for which one possible solution is enforcing the software with Trusted Computing (TC). However, in order to eliminate another trusted party, i.e. TC hardware/software platform, we are proposing an extension to our protocol with ZKPs.

We consider a full-adversary model. In this security scenario, all operations of the customers, and even the utility, must be verified. We design our protocol such that any participant should be able to verify others using the aforementioned ZKPs with their non-interactive Fiat-Shamir versions [40]. This strong privacy version requires two slight changes from the above protocol: each customer publicly announces his decrypting factor in Round 4 and receives others' as well as each calculates $r$ and $c$ such that every customer ends up having the same element via independent computations, implying the same ephemeral key for the encryption. It also implies that the contribution of the utility to the DR-bidding is limited to only determining the available number of units $M$, thereby eliminating any possible malicious attempt by the utility. Below, we describe the rounds as to how the ZKPs in Section V-A can be used:

R1: Each customer uses `ZK1` to prove knowledge of $x_a$ for his published partial public key $y_a$.

R2: Each customer uses `ZK4` for every $b_{ja}$, $1 \le j \le k$, to prove that his bid vectors satisfy the price vector interval condition to prevent any kind of price rigging or collusion.

R3: Any verifier does the same computation and checks equality of the results.

R4: Each customer uses `ZK2` to prove $\log_g y_a = \log_{\prod \sigma_{ij}} \phi_{aj}$, for $1 \le j \le k$.

R5-6: Each customer calculates $r$ individually, creates $c_j$ and encrypts with public key $y$ using the same ephemeral key.

R7: With `ZK2`, discrete logarithm equality is shown for each $(\gamma_{ij}^{(a)}, \sigma_{ij}^{(a)})$, $1 \le i \le n, 1 \le j \le k$ by customer $a$.

R8: With `ZK2`, discrete logarithm equality is shown for each $\log_g y_a = \log_{\prod \sigma_{ij}^{(a)}} \phi_{ij}^{(a)}$, $1 \le i \le n, 1 \le j \le k$ by customer $a$.

R9: Between the utility and each customer, the outcomes should be equal and can be verified via the direct equality check.

Remember that the protocol description in Section III assumes that the utility is an HbC. Obviously, under the full adversary model, we can no longer trust the utility to carry out the computations given in Equation (2). Thus, we resort to the distributed computation of these values by everyone using a form of MPC. In the former model when the utility is an HbC, it has the luxury of keeping the winning price secret without disclosing it to the losers of the bidding process. However, for the latter, since it is done in a distributed fashion, all parties will find out the settlement price, including the losers. While more parties are exposed to this previously secret settlement price, one might argue that this exposure might be beneficial from a game-theoretical perspective. However, this dimension is out of the scope of our paper.

A major computational overhead of the new protocol is the requirement that every party verify every ZKP created by every other party. A hybrid solution is possible to relieve this burden: The utility verifies the ZKPs and broadcasts the results. It is up to each customer whether to accept these results or to verify them on his own. We believe that having the possibility to invoke the full ZKP of each and every step of the bidding process is by itself a great deterrent to any misuse or abuse to compromise the integrity. So, in practice, the computational load of the ZKP for each and every bid can be avoided. A random invocation of the full ZKP with strict sanctioning of any misbehavior is likely to be a strong mechanism against any adversarial participation.

## VI. Conclusion

Demand response is a central component of the Smart Grid paradigm as it is the fundamental enabler of a significant portion of financial and operational benefits to all the parties involved. Indeed, it is reported by various interest groups that realization of a demand response ecosystem with full participation of market stakeholders has a potential to reduce the peak electricity demand as much as %20. Hence, increasing the participation in demand response programs is imperative for boosting the benefits for both individual consumers and the

system as a whole. Yet, efficient mechanisms for mitigating the privacy violation possibilities due to the participation in demand response programs are not readily available. Therefore, in this study, we present a distributed. multi-unit privacy assuring bidding (PAB) protocol for demand response.

The most important feature of our protocol is that it does not rely on any kind of trusted third party which is essential in eliminating a potential source for security breaches by adversaries. We also provide a thorough security analysis of PAB by means of four Zero-Knowledge Proofs to show that any potential privacy concerns of the customers are addressed while the integrity of the process from the utility's perspective is preserved. What replaces the trusted third party in our protocol is the computational hardness of discrete logarithm problem and efficient zero knowledge proofs.

## ACKNOWLEDGEMENT

Suleyman Uludag is partially supported by The Scientific and Technological Research Council of Turkey (TUBITAK) BIDEB 2221 Fellowship for Visiting Scientists Program 2015/12.

## REFERENCES

[1] National Institute of Standards and Technology Special Publication 1108r3, "NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 3.0," Smart Grid Interoperability Panel (SGIP), 2014.

[2] European Committee for Electrotechnical Standardization, "Final report of the CEN/CENELEC/ETSI Joint Working Group on Standards for Smart Grids," 2011.

[3] "IEEE Guide for Smart Grid Interoperability of Energy Technology and Information Technology Operation with the Electric Power System (EPS), End-Use Applications, and Loads," *IEEE Std 2030-2011*, pp. 1–126, 2011.

[4] Y. W. Law, T. Alpcan, V. C. Lee, A. Lo, S. Marusic, and M. Palaniswami, "Demand Response Architectures and Load Management Algorithms for Energy-Efficient Power Grids: A Survey," in *Proc. IEEE International Conference on Knowledge, Information and Creativity Support Systems*, 2012, pp. 134–141.

[5] M. Albadi and E. El-Saadany, "A summary of demand response in electricity markets," *Electric Power Systems Research*, vol. 78, no. 11, pp. 1989–1996, Nov. 2008.

[6] "White paper: Demand Response: A Multi-Purpose Resource For Utilities and Grid Operators," Energy Network Operations Center (EnerNOC), Tech. Rep., 2009.

[7] P. Siano, "Demand response and smart grids: A survey," *Renewable and Sustainable Energy Reviews*, vol. 30, pp. 461–478, Feb. 2014.

[8] "Benefits of Demand Response in Electricity Markets and Recommendations for Achieving Them - A Report to the United States Congress Pursant to Section 1252 of the Energy Policy Act of 2005, US Department of Energy," US Department of Energy, Tech. Rep., 2006.

[9] "NERC Demand Response Availability Data System (DADS): Phase I & II Final Report," The North American Electric Reliability Corporation (NERC), Tech. Rep., 2011.

[10] "The Power to choose: demand response in liberalised electricity markets," International Energy Development Agency, Organisation for Economic Co-operation and Development (OECD), Tech. Rep., 2003.

[11] "NISTIR 7628 Revision 1, Guidelines for Smart Grid Cybersecurity: Vol. 2, Privacy and the Smart Grid," Smart Grid Interoperability Panel (SGIP), Smart Grid Cybersecurity Committee, 2014.

[12] C.-L. Su and D. Kirschen, "Quantifying the Effect of Demand Response on Electricity Markets," *IEEE Trans. Power Syst.*, vol. 24, no. 3, pp. 1199–1207, Aug. 2009.

[13] Z. Wang and G. Zheng, "Residential Appliances Identification and Monitoring by a Nonintrusive Method," *IEEE Trans. Smart Grid*, vol. 3, no. 1, pp. 80–92, Mar. 2012.

[14] M. A. Lisovich, D. K. Mulligan, and S. B. Wicker, "Inferring Personal Information from Demand-Response Systems," *IEEE Security Privacy*, vol. 8, no. 1, pp. 11–20, Jan.-Feb. 2010.

[15] E. L. Quinn, "Privacy and the new energy infrastructure," Center for Energy and Environmental Security (CEES) Working Paper No. 09-001, 2008.

[16] M. Zeifman and K. Roth, "Nonintrusive appliance load monitoring: Review and outlook," *IEEE Trans. Consum. Electron.*, vol. 57, no. 1, pp. 76–84, Feb. 2011.

[17] C. Cuijpers and B.-J. Koops, *European Data Protection: Coming of Age*. Dordrecht: Springer Netherlands, 2013, ch. Smart Metering and Privacy in Europe: Lessons from the Dutch Case, pp. 269–293.

[18] S. D. Warren and L. D. Brandeis, "The Right to Privacy," *Harvard Law Review*, vol. 4, no. 5, pp. 193–220, Dec. 1890.

[19] H. van Rossum, H. Gardeniers, J. Borking, A. Cavoukian, J. Brans, N. Muttupulle, and N. Magistrale, *Privacy-Enhancing Technologies: The Path to Anonymity*. Den Haag: Information and Privacy Commissioner / Ontario, Canada & Registratiekamer, The Netherlands, 1995.

[20] G. V. Blarkom, J. Borking, and J. Olk, "Handbook of privacy and privacy-enhancing technologies," *Privacy Incorporated Software Agents*, pp. 42–50, 2003.

[21] S. Uludag, S. Zeadally, and M. Badra, "Techniques, Taxonomy, and Challenges of Privacy Protection in Smart Grid," in *Privacy in a Digital, Networked World : Technologies, Implications and Solutions*, S. Zeadally and M. Badra, Eds. Springer London, 2015, ch. 15.

[22] "White paper: The OpenADR Primer, An introduction to Automated Demand Response and the OpenADR Standard," OpenADR Alliance, Tech. Rep., 2011.

[23] M. A. Piette, G. Ghatikar, S. Kiliccote, E. Koch, D. Hennage, P. Palensky, and C. McParland, "Open Automated Demand Response Communications Specification (Version 1.0)," 2009.

[24] A. Paverd, A. Martin, and I. Brown, "Privacy-Enhanced Bi-Directional Communication in the Smart Grid using Trusted Computing," in *Proc. IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2014, pp. 872–877.

[25] ——, "Security and Privacy in Smart Grid Demand Response Systems," in *Smart Grid Security*, ser. Lecture Notes in Computer Science, J. Cuellar, Ed. Springer, 2014, vol. 8448, pp. 1–15.

[26] M. Karwe and J. Struker, "Maintaining Privacy in Data Rich Demand Response Applications," in *Smart Grid Security*, ser. Lecture Notes in Computer Science, J. Cuellar, Ed. Berlin, Heidelberg: Springer, 2013, vol. 7823, pp. 85–95.

[27] Y. Gong, Y. Cai, Y. Guo, and Y. Fang, "A Privacy-Preserving Scheme for Incentive-Based Demand Response in the Smart Grid," *IEEE Trans. Smart Grid*, vol. 7, no. 3, pp. 1304–1313, May 2015.

[28] M. S. Rahman, A. Basu, and S. Kiyomoto, "Privacy-friendly secure bidding scheme for demand response in smart grid," in *Proc. IEEE International Smart Cities Conference (ISC2)*, Oct. 2015, pp. 1–6.

[29] S. Uludag, M. F. Balli, A. A. Selcuk, and B. Tavli, "Privacy-Guaranteeing Bidding in Smart Grid Demand Response Programs," in *Proc. IEEE Globecom Workshop on SmartGrid Resilience (SGR) (GC'15 - Workshop - SGR)*, San Diego, USA, Dec. 2015, pp. 1–6.

[30] F. Brandt and T. Sandholm, *Financial Cryptography and Data Security: 9th International Conference (FC 2005)*. Springer, 2005, ch. Efficient Privacy-Preserving Protocols for Multi-unit Auctions, pp. 298–312.

[31] F. Brandt, "How to obtain full privacy in auctions," *International Journal of Information Security*, vol. 5, no. 4, pp. 201–216, Oct. 2006.

[32] A. C. Yao, "Protocols for secure computations," *Proc. Annual Symp. Foundations of Computer Science (SFCS 1982)*, pp. 160–164, Nov. 1982.

[33] D. Shanks, "Class number, a theory of factorization, and genera," in *Proc. Sympos. Pure Math.*, 1971, vol. XX, pp. 415–440.

[34] J. M. Pollard, "Kangaroos, Monopoly and Discrete Logarithms," *Journal of Cryptology*, vol. 13, no. 4, pp. 437–447, Sep. 2000.

[35] D. J. Bernstein and T. Lange, "Computing small discrete logarithms faster," in *Proc. Progress in Cryptology (INDOCRYPT 2012)*, S. Galbraith and M. Nandi, Eds. Springer, 2012, pp. 317–338.

[36] O. Goldreich, *Foundations of Cryptography: Volume 1*. New York, NY, USA: Cambridge University Press, 2006.

[37] C. P. Schnorr, "Efficient signature generation by smart cards," *Journal of Cryptology*, vol. 4, no. 3, pp. 161–174, Jan. 1991.

[38] D. Chaum and T. P. Pedersen, "Wallet databases with observers," in *Proc. Advances in Cryptology (CRYPTO'92)*, E. F. Brickell, Ed. Springer, 1993, pp. 89–105.

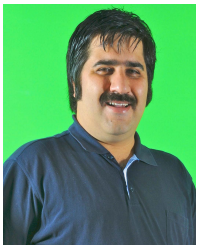
[39] R. Cramer, R. Gennaro, and B. Schoenmakers, "A secure and optimally efficient multi-authority election scheme," in *Proc. Annual International Conference on Theory and Application of Cryptographic Techniques (EUROCRYPT'97)*, 1997, pp. 103–118.

[40] A. Fiat and A. Shamir, *Proc. Advances in Cryptology (CRYPTO'86)*. Springer, 1987, ch. How To Prove Yourself: Practical Solutions to Identification and Signature Problems, pp. 186–194.

**Muhammed Fatih Balli** received his BS degrees in both Electrical and Electronics Engineering and Computer Engineering from TOBB University of Economics and Technology, Ankara, Turkey, in 2015. He is currently a PhD candidate at the Laboratory of Cryptography and Security (LASEC), at Ecole Polytechnique Federale de Lausanne. He is interested in applied cryptography, homomorphic encryption and biometric identity privacy.

**Suleyman Uludag** is an Associate Professor of Computer Science at the University of Michigan - Flint. His research interests have been around secure data collection, Smart Grid communications, Smart Grid privacy, Smart Grid optimization, demand response bidding privacy, Denial-of-Service in the Smart Grid, cybersecurity education and curriculum development, routing and channel assignment in Wireless Mesh Networks, Quality-of-Service (QoS) routing in wired and wireless networks, topology aggregation.

**Ali Aydin Selcuk** is a Professor at the Computer Engineering Department, TOBB University of Economics and Technology, Ankara, Turkey. He received his BS and MS degrees in Industrial Engineering from Middle East Technical University and Bilkent University, Ankara, Turkey, in 1993 and 1995, respectively. He received his PhD degree in Computer Science from University of Maryland Baltimore County, Maryland, USA in 2001. Prior to joining TOBB University, he worked at Bilkent University, Purdue University, Novell Networks, and RSA Laboratories. His research interests are in applied cryptography and network security.



**Bulent Tavli** (S'97–M'05) is a Professor at the Electrical and Electronics Engineering Department, TOBB University of Economics and Technology, Ankara, Turkey. He received his BS degree in Electrical and Electronics Engineering from the Middle East Technical University, Ankara, Turkey, in 1996. He received his MS and PhD degrees in Electrical and Computer Engineering from the University of Rochester, Rochester, NY, USA in 2002 and 2005, respectively. Wireless communications, networking, optimization, embedded systems, information security, and smart grid are his current research areas.