Sparse Attack Construction and State Estimation in the Smart Grid: Centralized and Distributed Models

Mete Ozay, Iñaki Esnaola, Fatos T. Yarman Vural, Sanjeev R. Kulkarni, and H. Vincent Poor

***Abstract*—New methods that exploit sparse structures arising in smart grid networks are proposed for the state estimation problem when data injection attacks are present. First, construction strategies for unobservable sparse data injection attacks on power grids are proposed for an attacker with access to all network information and nodes. Specifically, novel formulations for the optimization problem that provide a flexible design of the trade-off between performance and false alarm are proposed. In addition, the centralized case is extended to a distributed framework for both the estimation and attack problems. Different distributed scenarios are proposed depending on assumptions that lead to the spreading of the resources, network nodes and players. Consequently, for each of the presented frameworks a corresponding optimization problem is introduced jointly with an algorithm to solve it. The validity of the presented procedures in real settings is studied through extensive simulations in the IEEE test systems.**

***Index Terms*—Smart grid security, false data injection, distributed optimization, sparse models, attack detection.**

I. Introduction

POWER networks are complex systems consisting of generators and loads that are connected by transmission and distribution lines [1]. These systems can be modeled by complex networks, in which the generators and loads are represented by physically distributed nodes and power lines are represented by edges that connect the nodes. Because of the geographic and physical distribution of the nodes and the power transmission constraints [2], various structural properties of complex networks are observed in power networks [3]. For instance, the distribution of electrical distances of Eastern, Western and Texas interconnects in the North American power network obeys a power-law distribution, which leads to scale free and hierarchical network structures [3].

The aforementioned structural properties of power networks constrain the way in which both attack and defense schemes are designed for the smart grid. Several attack vector construction and detection methods have been introduced using either centralized [1], [4], [5], [6] or distributed [7], [8] models. Data sparsity properties have been analyzed for constructing unobservable sparse attack vectors by Liu et al. [1]. Kosut et al. [5] have introduced the relationship between attack detectability and network observability using a graph-theoretic model. Xie et al. [9] proposed a distributed wide-area state vector estimation algorithm which is also employed for bad data detection [7]. However, they do not exploit the sparsity and instead they define the state estimation problem as a weighted least squares (WLS) problem. Pasqualetti et al. [10] solved a similar WLS problem using a measurement distributed decomposition method for distributed state estimation and attack detection. Yang et al. [11] have proposed a hierarchical architecture to construct sparse attack vectors using combinatorial search methods. Vukovic et al. [12] have analyzed various mitigation schemes of data integrity attacks for state estimation. Recent advances for attack vector construction and state vector estimation methods in power systems have been reviewed in [13] and [14].

The centralized attack schemes proposed in this paper follow the undetectability criteria given in [1]. First, *sparse targeted false data injection attacks* are introduced which provide a strategy for tampering with the measurements from meters in order to build a specific data injection vector. In the second proposed method, called *strategic sparse attacks*, the sparse attack vector is constructed by assuming that the attacker has control over only a set of measurements and that the system has secure measurements that cannot be considered in the construction of the attack vector.

Since power grid networks are large scale networks, system monitoring and security control as envisioned for the smart grid are challenging problems. Therefore, decentralized options in which the computational complexity is distributed throughout the network are desirable. For this reason, distributed estimation techniques arise as strong candidates to incorporate adaptability to dynamic network topologies and flexible reconfiguration in case of sub-network faults. Additionally, distributed estimation techniques do not require all network state information to be available to each group, which facilitates operating with limited knowledge about the state of the network. However, the distributed structure of the networks may lead to critical attacks. For instance, distributed and collective attacks to *active nodes*, which have higher numbers of connections than the rest of the nodes, may cause larger damage to the network (i.e., the group of nodes connected to *active nodes*), because of its scale-free and hierarchical structure [3], [15].

Manuscript received September 21, 2012; revised March 15, 2013. This research was supported in part by the Center for Science of Information (CSoI), an NSF Science and Technology Center, under Grant CCF-0939370, by the U. S. Air Force office of Scientific Research under MURI Grant FA9550-09-1-0643, and by the U. S. Army Research Office under MURI Grant W911NF-11-1-0036 and Grant W911NF-07-1-0185.

M. Ozay, I. Esnaola, S. R. Kulkarni, and H. V. Poor are with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544, USA (e-mail: {mozay, jesnaola, kulkarni, poor}@princeton.edu).

F. T. Y. Vural is with the Department of Computer Engineering, Middle East Technical University, Ankara, Turkey (e-mail: vural@ceng.metu.edu.tr).

Digital Object Identifier 10.1109/JSAC.2013.130713.

We introduce two distributed attack models that make use of the sparsity of the attack vectors. The first model, *Distributed Sparse Attacks*, assumes that the attacks are directed at clusters of measurements. In this setting, attackers have access to a subset of the measurements observed by the nodes in the cluster. The goal is to achieve a consensus on the design of the attack vectors by iteratively computing them for the measurements observed in each cluster. The second model, *Collective Sparse Attacks*, assumes that the network topology is known by the attackers and may access the measurements observed in the whole network. However, attacks occur in groups, i.e., state variables in the same group are attacked by the same attack vectors.

In addition, we introduce two distributed state vector estimation methods from the perspective of the network operator. The first method, *Distributed State Vector Estimation*, considers the scale-free or hierarchical structure of the network, i.e., the observed measurements are grouped into clusters. Then, local state vector estimates are computed using local measurements by either local network operators or *smart* Phasor Measurement Units (PMUs). Using an iterative message-passing sparse optimization algorithm, each local operator or unit sends the estimated state vectors to centralized network processors, which update the state vector estimates. The second method, called *Collaborative Sparse State Vector Estimation*, assumes that different vector operators estimate a subset of state variables. For instance, different network operators may have expertise or special tools in order to estimate specific state variables and as a result, state vector variables are assumed to be distributed in groups and locally accessed by network operators. In this method, each network operator computes an estimate of the subset of the state variables using local data, and these estimates are then sent to a centralized operator in order to update their values. We analyze the proposed state vector estimation methods for attack detection and identification using a residuals test method in Section VII.

All the optimization problems presented in this paper are solved using the Alternating Direction Method of Multipliers (ADMM) algorithm [16]. Parameter and stopping criteria selection methods of ADMM are given in [16] and [17]. Moreover, convergence properties of ADMM are analyzed in [16] and [18].

In the next section, we review the unobservable false data injection and state vector estimation problems. Section III describes centralized sparse attack methods in which the sparse structure of the problem is exploited. In Section IV, we introduce distributed and collaborative state vector estimation methods. We introduce distributed and collective sparse attack models in Section V, and their computational complexity is analyzed in Section VI. We assess the validity of the proposed methods using real-world power systems in Section VII. The paper concludes with Section VIII.

II. Problem Formulation

A. System Model

A review of the problem formulation of false data injection attacks and the state vector estimation problem for attacked systems follows. Consider the DC power flow state acquisition problem [1] given by

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{n}, \tag{1}$$

where $\mathbf{z} \in \mathbb{R}^N$ is the vector of measurements, $\mathbf{x} \in \mathbb{R}^D$ is the state vector which consists of the voltage phase angles at the buses, $\mathbf{H} \in \mathbb{R}^{N\times D}$ is the measurement Jacobian matrix and $\mathbf{n} \in \mathbb{R}^N$ is the measurement noise.

The goal of the network operator is to estimate the state vector and decide whether an attack is present. If the noise is normally distributed with zero mean and independent components, then the following estimator can be employed [1]:

$$\hat{\mathbf{x}} = (\mathbf{H}^T\mathbf{\Lambda}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{\Lambda}\mathbf{z}\,, \tag{2}$$

where $\mathbf{\Lambda}$ is a diagonal matrix whose diagonal elements are given by $\mathbf{\Lambda}_{ii} = \xi_i^{-2}$, and ξ_i^2 is the variance of the i-th measurement for $i = 1, ..., N$. The network operator decides that an attack is present if $\|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\|_2^2 > \tau$, where $\|\cdot\|_2$ is the ℓ_2-norm and $\tau \in \mathbb{R}$ is a given threshold. If $\|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\|_2^2 \leq \tau$ then no attack is declared.

The goal of the attacker is to inject a false data vector $\mathbf{a} \in \mathbb{R}^N$ into the measurements without being detected by the operator. Since the attack is performed by changing the values of a subset of all the measurements, the resulting observation model for the operator is

$$\tilde{\mathbf{z}} = \mathbf{H}\mathbf{x} + \mathbf{a} + \mathbf{n}. \tag{3}$$

Note that for the attack vector $a_i \neq 0$, $\forall i \in \mathcal{A}$, where $\mathcal{A}$ is the set of measurement variable indices with which the attacker tampers. On the other hand, the measurements over which the attacker has no control are the secure variables, $a_i = 0$, $\forall i \in \mathcal{S}$. Note that $\mathcal{S} = \bar{\mathcal{A}}$ where $\bar{(\cdot)}$ is the set complement operator and $|\mathcal{A} \cup \mathcal{S}| = N$ where $|\cdot|$ denotes set cardinality.

Imposing the constraint $\mathbf{a} = \mathbf{H}\mathbf{c}$, where $\mathbf{c} \in \mathbb{R}^D$ is an injected data vector guarantees undetectability via residual tests since it lies in the column space of $\mathbf{H}$ [1], [5]. Note that (3) can be rewritten as

$$\tilde{\mathbf{z}} = \mathbf{H}\tilde{\mathbf{x}} + \mathbf{n}, \tag{4}$$

where $\tilde{\mathbf{x}} = \mathbf{x} + \mathbf{c}$ is what an operator unaware of the attack tries to estimate instead of the actual state vector $\mathbf{x}$.

B. Sparsity in the System

Assuming that an attacker can tamper with a limited number of meters poses the optimization problem in a framework in which the attack vector is sparse. Specifically, if k meters are controlled by the attacker, then $\mathbf{a}$ is at most k-sparse, i.e., $\|\mathbf{a}\|_0 \leq k$, where $\|\cdot\|_0$ is the l_0 norm. In [1] Liu et al. prove the existence of unobservable attack vectors if $k \geq N - D + 1$. Finding the sparsest attack vector that satisfies $\mathbf{a} = \mathbf{H}\mathbf{c}$ is computationally intractable in general. Surprisingly, the solution can be relaxed into a convex optimization problem by using the ℓ_1-norm as the objective function instead of the ℓ_0-norm [19], [20]. Based on sparse reconstruction techniques, Kim and Poor [4] provide a greedy approach for the attack vector construction when a subset of the measurements is controlled by the attacker while the remaining measurements are secured.

TABLE I
RANK VALUES OF MEASUREMENT JACOBIAN MATRICES OF IEEE TEST SYSTEMS AND THE 3375-BUS POLISH SYSTEM PLUS - WINTER 2007-08 EVENING PEAK SYSTEM.

System	N	D	Rank	R
9-Bus	19	9	8	72.00 %
14-Bus	34	14	3	80.25 %
30-Bus	71	30	29	90.89 %
39-Bus	85	39	38	93.27 %
57-Bus	137	57	57	95.22 %
118-Bus	304	118	118	97.64 %
300-Bus	711	300	300	99.09 %
3375-Bus	7536	3375	3375	99.92 %

A second scenario in which the sparsity of the system can be exploited is in the estimation of the state vector. Considering that system states are given by a random process $\{\mathbf{x}_t\}$, the components of the state vector that change *significantly* during an interval (t, t') are defined as

$$\mathcal{X}_{t,t'} = \{i : |\mathbf{x}_t(i) - \mathbf{x}_{t'}(i)| \geq \epsilon\} \tag{5}$$

where ϵ defines the threshold for change *significance*. That being the case, the operator does not need to estimate all state variables for each time t. Assuming that it has a previous state estimate, $\hat{\mathbf{x}}_t$, from time t, it can estimate the values that changed above the ϵ threshold by realizing that

$$\mathbf{y}_{t'} - \mathbf{y}_t = \mathbf{H}\mathbf{x}_{t'} - \mathbf{H}\hat{\mathbf{x}}_t + \mathbf{z}_{t'} - \mathbf{z}_t \tag{6}$$

$$= \mathbf{H}(\mathbf{x}_{t'} - \hat{\mathbf{x}}_t) + \mathbf{z}_{t'} - \mathbf{z}_t \tag{7}$$

$$= \mathbf{H}\boldsymbol{\delta}_{t,t'} + \mathbf{z}_{t'} - \mathbf{z}_t, \tag{8}$$

where $\boldsymbol{\delta}_{t,t'} = \mathbf{x}_{t'} - \hat{\mathbf{x}}_t$ has significantly changed variables given by indices $\mathcal{X}_{t,t'}$. By choosing the significance threshold, ϵ, and the estimation time interval appropriately, $\boldsymbol{\delta}$ becomes a nearly sparse vector whose k largest components can be recovered by solving a standard compressed sensing problem of the form

$$\begin{array}{ll} \text{minimize} & \|\boldsymbol{\delta}\|_1 \\ \text{subject to} & \|\mathbf{y}_{t'} - \mathbf{y}_t - \mathbf{H}\boldsymbol{\delta}\|_2^2 < \gamma, \end{array} \tag{9}$$

where γ is a regularization parameter.

An additional optimization constraint is imposed by the rank deficiency observed in the measurement Jacobian matrix $\mathbf{H}$ of several IEEE test systems, such as the IEEE 39-Bus [21]. In Table I, we show the values of N, D, rank and R (the ratio of the number of nonzero elements of the entries of $\mathbf{H}$) for test systems. We observe that 9-Bus, 14-Bus, 30-Bus and 39-Bus test systems are rank deficient. Although $\mathbf{H}$ matrices of 57-Bus, 118-Bus, 300-Bus and 3375-Bus test systems are not rank deficient, their R values are greater than those of the rank-deficient matrices. Note that the sparseness increases as the system size increases. Following the sparse nature of the system, (3) and (4) are formulated as ℓ_1-norm optimization problems [4].

III. CENTRALIZED DATA SPARSE ATTACKS

A. Sparse Targeted False Data Injection Attacks

Targeted False Data Injection Attacks consist of attackers constructing false data injection vectors $\mathbf{a}$ corresponding to a given attack vector $\mathbf{c}$. In this section, we introduce two models that employ LASSO and regressor selection algorithms to solve the targeted false data injection problem.

1) Targeted LASSO Attacks: The sparseness of $\mathbf{c}$ is exploited for targeted false data injection attacks [1], where $c_j \in \mathbb{R}$ are fixed and defined by attackers $\forall j \in \mathcal{I}$, for a set $\mathcal{I}$ of indices of the state vector variables that will be attacked. However, $c_j \in \mathbb{R}$ are randomly selected by the attacker according to a probability distribution $\forall j \in \bar{\mathcal{I}}$, where $\bar{\mathcal{I}}$ is the set of off-target variables which are not specifically determined by the attackers. In other words, the attackers do not have control on the variables $c_j \in \bar{\mathcal{I}}$. Note that, $|\mathcal{I} \cup \bar{\mathcal{I}}| = D$.

In order to compute the off-target and targeted attack vectors, we employ the following decomposition [1]:

$$\mathbf{a} = \mathbf{H}\mathbf{c} = \sum_{i \in \bar{\mathcal{I}}} c_i \mathbf{h}_i + \sum_{j \in \mathcal{I}} c_j \mathbf{h}_j, \tag{10}$$

where $\mathbf{h}_l$ is the l-th column of $\mathbf{H}$. Then, we define a submatrix $\mathbf{H}^{\bar{\mathcal{I}}}$ of $\mathbf{H}$ as $\mathbf{H}^{\bar{\mathcal{I}}} = (\mathbf{h}_{j_i}, \cdots, \mathbf{h}_{j_{D-|\mathcal{I}|}})$, $\forall j_i \in \bar{\mathcal{I}}$ and $1 \leq i \leq D - |\mathcal{I}|$ [1] and construct a vector $\mathbf{b}$ in the range space of the attacked measurements, such that $\mathbf{b} = \sum_{j \in \mathcal{I}} \mathbf{h}_j \mathbf{c}_j$. Using this construction, we relate $\mathbf{b}$ to the measurements $\mathbf{H}^{\bar{\mathcal{I}}}$, such that $\mathbf{P}^{\bar{\mathcal{I}}} = \mathbf{H}^{\bar{\mathcal{I}}}(\mathbf{H}^{\bar{\mathcal{I}}^T}\mathbf{H}^{\bar{\mathcal{I}}})^{-1}\mathbf{H}^{\bar{\mathcal{I}}^T}$, $\mathbf{B}^{\bar{\mathcal{I}}} = \mathbf{P}^{\bar{\mathcal{I}}} - \mathbf{I}$ and $\mathbf{y} = \mathbf{B}^{\bar{\mathcal{I}}}\mathbf{b}$ [1]. Therefore, we can compute $\mathbf{a}$ by solving $\mathbf{y} = \mathbf{B}^{\bar{\mathcal{I}}}\mathbf{a}$ [1].

We assume that given an attack vector $\mathbf{a}$, the attack strategy of an attacker is to find a sparse $\mathbf{a}$, such that $\mathbf{y} = \mathbf{B}^{\bar{\mathcal{I}}}\mathbf{a}$. Then, using l_1 relaxations for sparse vector estimation [19], [20], [22], we introduce the following optimization problem to model the sparse false data injection attack:

$$\begin{array}{ll} \text{minimize} & \|\mathbf{a}\|_1 \\ \text{subject to} & \mathbf{y} = \mathbf{B}^{\bar{\mathcal{I}}}\mathbf{a}. \end{array} \tag{11}$$

(11) is called basis pursuit and can be employed to find a sparse solution vector $\mathbf{c}$ [19], [20], [22]. In order to solve the optimization problem above using ADMM [16], (11) is formulated as

$$\begin{array}{ll} \text{minimize} & I(\mathbf{a}) + \|\boldsymbol{\beta}\|_1 \\ \text{subject to} & \mathbf{a} - \boldsymbol{\beta} = \mathbf{0}, \end{array} \tag{12}$$

where $I(\mathbf{a})$ is the indicator function for $\{\mathbf{a} \in \mathbb{R}^N : \mathbf{y} = \mathbf{B}^{\bar{\mathcal{I}}}\mathbf{a}\}$ and $\boldsymbol{\beta} \in \mathbb{R}^N$ is the optimization variable. The sparsity of $\mathbf{a}$ is governed by $\|\boldsymbol{\beta}\|_1$ using a scalar real number $\lambda > 0$, which is a regularization parameter. Moreover, in order to reduce the probability of the attack being detected, $\|\mathbf{y} - \mathbf{B}^{\bar{\mathcal{I}}}\mathbf{a}\|_2^2$ can be used as a cost function which results in the optimization problem

$$\begin{array}{ll} \text{minimize} & \|\mathbf{y} - \mathbf{B}^{\bar{\mathcal{I}}}\mathbf{a}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1 \\ \text{subject to} & \mathbf{a} - \boldsymbol{\beta} = 0. \end{array} \tag{13}$$

Problem (13) is a *LASSO* optimization [22] and can be solved via ADMM [16] as follows:

Algorithm 1 (LASSO via ADMM):

- INPUT:
  - Projection matrix defined by secure set $\mathbf{B}^{\bar{\mathcal{I}}}$
  - Projected vector containing injected data $\mathbf{y}$
  - Penalty parameter ρ
  - Maximum number of iterations t'
- OUTPUT:
  - Attack vector candidate $\mathbf{a} \stackrel{\text{def}}{=} \mathbf{a}^t$
- PROCEDURE:

1) Initialize $t = 0$, $\boldsymbol{\beta} = \mathbf{0}$ and $\mathbf{u} = \mathbf{0}$
2) Compute *ridge regression* with penalty parameter ρ:
$$\mathbf{a}^{t+1} = \left((\mathbf{B}^{\bar{\mathcal{I}}})^T\mathbf{B}^{\bar{\mathcal{I}}} + \rho I\right)^{-1}\left((\mathbf{B}^{\bar{\mathcal{I}}})^T\mathbf{y} + \rho(\boldsymbol{\beta}^t - \mathbf{u}^t)\right) \tag{14}$$
3) Perform *soft thresholding* defined by proximity operator $\Pi_\kappa(\phi) = (\phi - \kappa)_+ - (-\phi - \kappa)_+$, where $(\phi)_+ = \max(\phi, 0)$:
$$\boldsymbol{\beta}^{t+1} = \Pi_{\lambda/\rho}(\mathbf{u}^t + \mathbf{a}^{t+1}) \tag{15}$$
4) Update:
$$\mathbf{u}^{t+1} = \mathbf{u}^t + \mathbf{a}^{t+1} - \boldsymbol{\beta}^{t+1} \tag{16}$$
5) Return to step 2 if a stopping criterion is not satisfied and $t < t'$

2) Selective Targeted Attacks: The previous approach provides an implicit control of the sparsity of $\mathbf{a}$ using parameter λ. In the following, the sparsity is explicitly controlled by introducing the constraint $\|\mathbf{a}\|_0 \leq k$ in (11) in the optimization problem as

$$\begin{array}{ll} \text{minimize} & \|\mathbf{y} - \mathbf{B}^{\bar{\mathcal{I}}}\mathbf{a}\|_2^2 \\ \text{subject to} & \|\mathbf{a}\|_0 \leq k\,. \end{array} \tag{17}$$

This optimization problem can also be solved via ADMM with a minor modification of Algorithm 1. Specifically, hard-thresholding $\Pi^*_{\lambda/\rho}(\mathbf{a}^{t+1} + \mathbf{u}^t)$ is employed in the update of $\boldsymbol{\beta}$, such that k largest magnitude elements of $\mathbf{u}^t + \mathbf{a}^{t+1}$ are kept and zeros are assigned to the remaining elements [16].

B. Strategic Sparse Attacks

In this section, we propose two algorithms to compute the attack vector $\mathbf{c}$ based on the formulations of *LASSO Attacks* and *Selective Sparse Attacks* for the strategic sparse attack model case introduced by Kim and Poor [4]. To this end, we first redefine the sparse data injection attack problem for ADMM. Then, we solve the optimization problems using LASSO and Regressor Selection algorithms.

1) Strategic Sparse Attacks with LASSO: In Strategic Sparse Attacks, a row-wise decomposition of the Jacobian measurement matrix is employed based on the set $\mathcal{A}$ of attacked measurement indices denoting the meters to which an attacker has access, and the set $\mathcal{S}$ of secure measurement indices, i.e., the indices of meters which cannot be tampered by an attacker. Specifically, a sub-matrix $\mathbb{H}^{\mathcal{S}} = (\mathbf{H}_{j_i,:}, \cdots \mathbf{H}_{j_{N-|\mathcal{S}|},:})$, $\forall j_i \in \mathcal{S}$, of $\mathbf{H}$ is constructed in order to represent the secure measurements, where $\mathbf{H}_{j_i,:}$ is the j_i-th row of $\mathbf{H}$, such that $\mathbb{H}^{\mathcal{S}}\mathbf{c} = \mathbf{0}$. Similarly, sub-matrix $\mathbb{H}^{\mathcal{A}}$ is defined for attacked measurements. As a result, the attacker's strategy is defined to find a solution $\mathbf{c}$ to the following optimization problem:

$$\begin{array}{ll} \text{minimize} & \|\mathbb{H}^{\mathcal{A}}\mathbf{c}\|_0 \\ \text{subject to} & \mathbb{H}^{\mathcal{S}}\mathbf{c} = \mathbf{0}\,, \\ & \|\mathbf{c}\|_\infty \geq \psi\,, \end{array} \tag{18}$$

where $\psi \geq 0$ is a given constant [4].

Define $\mathbf{h}_i$ as the i-th column vector of $\mathbf{H}$, the sub-matrix $\mathbb{H}_i \in \mathbb{R}^{N\times(D-1)}$ formed by removing the i-th column of $\mathbf{H}$, and $\boldsymbol{\sigma}_i \in \mathbb{R}^{D-1}$ formed by removing the i-th variable of $\mathbf{c}$. Following these definitions, the strategic sparse attack is defined as

$$\begin{array}{ll} \text{minimize} & \|\mathbb{H}_i^{\mathcal{A}}\boldsymbol{\sigma}_i + \mathbf{h}_i^{\mathcal{A}}\|_1 \\ \text{subject to} & \mathbb{H}_i^{\mathcal{S}}\boldsymbol{\sigma}_i + \mathbf{h}_i^{\mathcal{S}} = \mathbf{0}\,. \end{array} \tag{19}$$

Since $\mathbf{H}$ and $\mathbf{c}$ are sparse, it follows that the problem can be reformulated as

$$\begin{array}{ll} \text{minimize} & \|\boldsymbol{\sigma}_i\|_1 \\ \text{subject to} & \mathbb{H}_i^{\mathcal{S}}\boldsymbol{\sigma}_i + \mathbf{h}_i^{\mathcal{S}} = \mathbf{0}\,. \end{array} \tag{20}$$

Since (20) is a LASSO problem, we reformulate (20) as an ADMM optimization problem as follows:

$$\begin{array}{ll} \text{minimize} & \|\mathbb{H}_i^{\mathcal{S}}\boldsymbol{\sigma}_i + \mathbf{h}_i^{\mathcal{S}}\|_2^2 + \lambda\|\boldsymbol{\theta}_i\|_1 \\ \text{subject to} & \boldsymbol{\sigma}_i - \boldsymbol{\theta}_i = \mathbf{0}\,, \end{array} \tag{21}$$

where $\boldsymbol{\theta}_i \in \mathbb{R}^{D-1}$ is the optimization variable. In order to solve (21), Algorithm 1 can be employed with inputs $\mathbb{H}_i^{\mathcal{S}}$, $\mathbf{y} = 1$ and θ_i. This procedure is repeated for $i = 1, ..., D$ in order to compute $\mathbf{c} = (\mathbf{c}_1, ..., \mathbf{c}_D)$.

2) Selective Strategic Sparse Attacks: As discussed in the previous section, the sparsity of $\mathbf{c}_i$ can be bounded explicitly by converting (20) to the equivalent regressor selection problem given by

$$\begin{array}{ll} \text{minimize} & \|\mathbb{H}_i^{\mathcal{S}}\boldsymbol{\sigma}_i + \mathbf{h}_i^{\mathcal{S}}\|_2^2 \\ \text{subject to} & \|\boldsymbol{\sigma}_i\|_0 \leq k\,. \end{array} \tag{22}$$

In this formalism, we relax the constraint in $\mathbb{H}^{\mathcal{S}}\mathbf{c} = \mathbf{0}$ and introduce a sparsity constraint in the construction of attack vectors $\mathbf{c}$, such that we compute an attack vector with at most k non-zero elements. The solution to (22) is the same as the one proposed for (21) except for substituting the soft thresholding operator in step 3 by a hard thresholding.

C. Computational Complexity of Centralized Sparse Attacks

The optimization problems of the centralized sparse attacks are solved using Algorithm 1. The computational complexity of the algorithm is dominated by the attack vector update step in (14) which solves a ridge regression problem [23]. Therefore, the computational complexity of the algorithm is $\Upsilon_1 \in O(t'\alpha^3)$, where

1) $\alpha = \min(N, |\bar{\mathcal{I}}|)$ for targeted attacks given in Section III.A, and
2) $\alpha = \min(N, D-1)$, for strategic attacks given in Section III.B.

Note that, the computational complexity of the algorithm is increased by an additional term D (the dimension of the attack vector) to $O(t'D\alpha^3)$ for strategic attacks, since the algorithm is implemented D times.

In the implementation, the running or iteration time t' can be relaxed by using performance based early stopping criteria as suggested in [16].

IV. Distributed and Collaborative Sparse State Vector Estimation

A sparse state vector estimation model, called *Distributed Sparse State Vector Estimation*, is first introduced in order to estimate the state vectors under attack on the network measurements using an instance distributed *LASSO* algorithm.

In the second model, called *Collaborative Sparse State Vector Estimation*, we assume that the topological information of the network and the measurements are available to the network operator. However, the network operator can choose to process different groups of state vectors using the *Group LASSO* algorithm. We solve the optimization problems using the ADMM algorithm.

A. Distributed Sparse State Vector Estimation

Measurements are distributed in the network and usually form clusters following the topological properties of the network. Additionally, observation vectors and measurement matrices are partitioned into G blocks denoted by $\mathcal{G}_i$ with $|\mathcal{G}_i| = N_i$ for $i = 1, \ldots, G$. As a result, the attacks can also be partitioned. Taking this into account, (3) can be rewritten as

$$\begin{bmatrix} \tilde{\mathbf{z}}_1 \\ \vdots \\ \tilde{\mathbf{z}}_G \end{bmatrix} = \begin{bmatrix} \mathbf{H}_1 \\ \vdots \\ \mathbf{H}_G \end{bmatrix} \mathbf{x} + \begin{bmatrix} \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_G \end{bmatrix} + \begin{bmatrix} \mathbf{n}_1 \\ \vdots \\ \mathbf{n}_G \end{bmatrix}, \tag{23}$$

where $\tilde{\mathbf{z}}_i \in \mathbb{R}^{N_i}$ is the measurement observed in the i-th cluster of nodes through measurement matrix $\mathbf{H_i} \in \mathbb{R}^{N_i \times D}$ and noise $\mathbf{n}_i \in \mathbb{R}^{N_i}$, and which is under attack $\mathbf{a}_i \in \mathbb{R}^{N_i}$ with $i = 1, ..., G$. For each cluster, we consider the penalty function

$$f_i = \|\tilde{\mathbf{z}}_i - \mathbf{H}_i \tilde{\mathbf{x}}_i\|_2^2 , \tag{24}$$

where $\tilde{\mathbf{x}}_i$ is the state vector estimated at cluster i. Note that

$$f \equiv \|\tilde{\mathbf{z}} - \mathbf{H}\tilde{\mathbf{x}}\|_2^2 = \sum_{i=1}^{G} f_i.$$

Therefore, we can write the distributed optimization problem in the following way:

$$\text{minimize} \quad \sum_{i=1}^{G} f_i + g(\boldsymbol{\beta}) \tag{25}$$

$$\text{subject to} \quad \tilde{\mathbf{x}}_i - \boldsymbol{\beta} = \mathbf{0}, \quad i = 1, ..., G , \tag{26}$$

where $\boldsymbol{\beta} \in \mathbb{R}^D$ is the optimization variable, $g(\boldsymbol{\beta}) = \lambda \|\boldsymbol{\beta}\|_1$ is the regularization function and $\lambda \in \mathbb{R}$ is the regularization parameter, which controls the sparsity of the solution vector. Since network operators accessing the local data should agree on the estimated state vector, we introduce a *consensus* constraint in (26). In other words, (26) is considered as a global consensus problem in which $\boldsymbol{\beta}$ is used as the *global* optimization variable.

We solve (26) using an ADMM implementation as described in the following algorithm.

Algorithm 2 (Distributed Estimation via ADMM):

- INPUT:
  - Projection matrix $\mathbf{H}$
  - State measurements $\tilde{\mathbf{z}}$
  - Set of clusters $\{\mathcal{G}_i\}_{i=1}^{G}$
  - Penalty parameter ρ
  - Maximum number of iterations t'
- OUTPUT:
  - Estimated state vector $\tilde{\mathbf{x}} \equiv \tilde{\mathbf{x}}^t$
- PROCEDURE:
  1) Initialize $t = 0$, $\boldsymbol{\beta}^0 = \mathbf{0}$, $\mathbf{u}^0 = \mathbf{0}$.
  2) For $i = 1, \ldots, G$ compute the Tikhonov-regularized least squares estimate with penalty parameter ρ given by
  $$\tilde{\mathbf{x}}_i^{t+1} = \left(\mathbf{H}_i^T \mathbf{H}_i + \rho \mathbf{I}\right)^{-1} \left(\mathbf{H}_i^T \tilde{\mathbf{z}}_i + \rho(\boldsymbol{\beta}^t - \mathbf{u}_i^t)\right). \tag{27}$$
  3) Perform a *soft thresholding* given by
  $$\boldsymbol{\beta}^{t+1} = \Pi_{\frac{\lambda}{\rho G}} \left(\frac{1}{G} \sum_{i=1}^{G} \left(\tilde{\mathbf{x}}_i^{t+1} + \mathbf{u}_i^t\right) \right), \tag{28}$$
  where the ℓ_1 proximity operator is defined as
  $$\Pi_\kappa(\phi) = (\phi - \kappa)_+ - (-\phi - \kappa)_+ \tag{29}$$
  and $(\phi)_+ = \max(\phi, 0)$.
  4) For $i = 1, \ldots, G$ update
  $$\mathbf{u}_i^{t+1} = \mathbf{u}_i^t + \tilde{\mathbf{x}}_i^{t+1} - \boldsymbol{\beta}^{t+1}. \tag{30}$$
  5) Return to step 2 if the halting criterion is not satisfied and $t < t'$.

Note that $\mathbf{H}_i$ is a sparse matrix or vector (depending on $\mathcal{G}_i$). Still, $(\mathbf{H}_i^T \mathbf{H}_i + \rho \mathbf{I})$ is invertible since $\rho > 0$.

Algorithm and optimization variables are initialized in the first step of the algorithm. In the second step, each network operator computes a local estimate using Tikhonov-regularized least squares [24], [25]. Then the local estimates are *gathered* to update the global variable $\boldsymbol{\beta}$ in the third step. Finally, the updated $\boldsymbol{\beta}$ is distributed or *broadcast* to the clusters to update the dual variables $\mathbf{u}_i$, $\forall i = 1, \ldots, G$, in the fourth step, and the halting criterion is checked in the last step.

B. Collaborative Sparse State Vector Estimation

In the distributed sparse attacks scenario, measurements are assumed to be distributed across clusters and operators have access only to local measurements. Alternatively, when collective sparse attacks are considered, operators know the whole topology of the network and the Jacobian measurement matrix $\mathbf{H}$. However, in a distributed framework operators observe a subset of state vector variables, i.e., each operator may observe different groups of buses.

In this setting, the observation model (3) can be rewritten as

$$\tilde{\mathbf{z}} = \left[\hat{\mathbf{H}}_1 \cdots \hat{\mathbf{H}}_G\right] \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_G \end{bmatrix} + \mathbf{a} + \mathbf{n} , \tag{31}$$

where $\tilde{\mathbf{z}} \in \mathbb{R}^N$ is the measurement vector, $\mathbf{x}_i \in \mathbb{R}^{D_i}$ is the state vector, $\mathbf{n} \in \mathbb{R}^N$ is a noise vector and $\hat{\mathbf{H}}_\mathbf{i} \in \mathbb{R}^{N \times D_i}$ is the Jacobian measurement submatrix formed by selecting the columns given by the indices of the subset of state variables assigned to cluster i. Given this structure, the optimization problem can be stated as

$$\text{minimize} \quad \|\mathbf{H}\tilde{\mathbf{x}} - \tilde{\mathbf{z}}\|_2^2 + \lambda \sum_{i=1}^{G} \|\tilde{\mathbf{x}}_i\|_2. \tag{32}$$

By introducing an optimization variable $\mathbf{v} \in \mathbb{R}^D$, it follows that

$$\text{minimize} \quad \|\mathbf{H}\mathbf{v} - \tilde{\mathbf{z}}\|_2^2 + \lambda \sum_{i=1}^{G} \|\tilde{\mathbf{x}}_i\|_2 \tag{33}$$

$$\text{subject to} \quad \tilde{\mathbf{x}}_i - \hat{\mathbf{v}}_i = \mathbf{0}, \quad i = 1, ..., G , \tag{34}$$

is equivalent to (32), where $\hat{\mathbf{v}}_i$ is the estimate of $\mathbf{v}$ for $\tilde{\mathbf{x}}_i$ [16]. In order to solve (34), the proposed ADMM implementation is described below.

Algorithm 3 (Collaborative Estimation via ADMM):

- INPUT:
  - Projection matrix $\mathbf{H}$
  - State measurements $\tilde{\mathbf{z}}$
  - Set of clusters $\{\mathcal{G}_i\}_{i=1}^{G}$
  - Penalty parameter ρ
  - Maximum number of iterations t'
- OUTPUT:
  - Estimated state vector $\tilde{\mathbf{x}} \equiv \tilde{\mathbf{x}}^t$
- PROCEDURE:
  1) Initialize $t = 0$, $\boldsymbol{\beta}^0 = \mathbf{0}$, $\mathbf{v}^0 = \mathbf{0}$, $\boldsymbol{\theta}^0 = \mathbf{0}$ and $\tilde{\mathbf{x}}^0 = \mathbf{0}$.
  2) For $i = 1, \ldots, G$ compute

$$\tilde{\mathbf{x}}_i^{t+1} = \underset{\tilde{\mathbf{x}}_i}{\operatorname{argmin}} \left(\rho \|\boldsymbol{\theta}_i^t\|_2^2 + \lambda \|\tilde{\mathbf{x}}_i\|_2 \right), \qquad (35)$$

  where $\boldsymbol{\theta}_i^t = \hat{\mathbf{H}}_i \left(\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_i^t \right) - \bar{\mathbf{v}}^t + \overline{\mathbf{H}\tilde{\mathbf{x}}}^t + \mathbf{u}^t$ and $\overline{\mathbf{H}\tilde{\mathbf{x}}}^t = \frac{1}{G} \sum_{i=1}^{G} \hat{\mathbf{H}}_i \tilde{\mathbf{x}}_i^t$.

  3) Update

$$\bar{\mathbf{v}}^{t+1} = \frac{1}{G+\rho} (\tilde{\mathbf{z}} + \rho \overline{\mathbf{H}\tilde{\mathbf{x}}}^{t+1} + \rho \mathbf{u}^t), \qquad (36)$$

$$\mathbf{u}^{t+1} = \mathbf{u}^t + \overline{\mathbf{H}\tilde{\mathbf{x}}}^{t+1} - \bar{\mathbf{v}}^{t+1}. \qquad (37)$$

  4) Return to step 2 if the halting criterion is not satisfied and $t < t'$.

V. Distributed and Collective Sparse Attacks

In Section III, we introduced centralized sparse attack methods. In this section, two distributed attack models are proposed in order to employ sparse attacks in a distributed framework. For this purpose, the structure of the measurements and the attack vectors is redefined, followed by a formulation of the false data injection problem as a distributed sparse optimization problem.

The proposed distributed attack models are motivated by two distributed attack scenarios.

1) In *Distributed Sparse Attacks*, *measurements* are assumed to be distributed in the network and may form clusters following the topological properties of the network. Therefore, different attackers located in different clusters can construct attack vectors by just analyzing the local measurements observed in the clusters.
2) *Collective Sparse Attacks* model assumes that attackers know the whole topology of the network and the Jacobian measurement matrix $\mathbf{H}$. However, in this case the attacks are directed at a group of *state vector variables* distributed in the network, *i.e.*, each attack injects false data into the state vector variables of the corresponding cluster.

Although linear sparse attacks are considered for the implementation of distributed attacks in this work, the proposed parallelization and distributed processing strategies can be used as design patterns for developing distributed sparse targeted false data injection attacks and strategic sparse attacks.

A. Distributed Sparse Attacks

A linear sparse attack model is considered, in which given an attack vector, $\mathbf{a}$, the attack strategy is to find a sparse injection vector, $\mathbf{c}$, such that $\mathbf{a} = \mathbf{H}\mathbf{c}$ [1]. Using an ℓ_1 relaxation for sparse vector estimation [19], [20], [22], the following optimization problem is considered:

$$\begin{array}{ll} \text{minimize} & \|\mathbf{c}\|_1 \\ \text{subject to} & \mathbf{a} = \mathbf{H}\mathbf{c}. \end{array} \qquad (38)$$

As noted before, measurements are assumed to be distributed in the network and may form clusters following the topological properties of the network. Similar to the partitioning presented in the previous section, the Jacobian measurement matrix is partitioned into G number of submatrices, which results in a partitioning of the attack vector given by

$$\begin{bmatrix} \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_G \end{bmatrix} = \begin{bmatrix} \mathbf{H}_1 \\ \vdots \\ \mathbf{H}_G \end{bmatrix} \mathbf{c}. \qquad (39)$$

Note that, (39) can also be employed in the distributed false data vector construction described in (23).

In order to solve (38) in the distributed form set by (39) using a distributed optimization algorithm, the loss function is assumed to be separable, such that

$$f_i = \|\mathbf{a}_i - \mathbf{H}_i \mathbf{c}_i\|_2^2. \qquad (40)$$

Note that, $\sum_{i=1}^{G} f_i = f$. Moreover, the optimization problem (38) is assumed to be feasible [16]. Therefore, the distributed optimization problem for (38) can be reformulated as

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{G} f_i + g(\boldsymbol{\phi}) \qquad (41) \\ \text{subject to} & \mathbf{c}_i - \boldsymbol{\phi} = \mathbf{0}, \quad i = 1, ..., G, \qquad (42) \end{array}$$

where $\boldsymbol{\phi} \in \mathbb{R}^{N_i}$ is the optimization variable, $g(\boldsymbol{\phi}) = \lambda \|\boldsymbol{\phi}\|_1$ is the regularization function and $\lambda \in \mathbb{R}$ is the regularization parameter which controls the sparsity of the solution vector. Interestingly, the optimization problem (42) is the same as the one posed in (26) and therefore, Algorithm 2 can be used to solve it.

B. Collective Sparse Attacks

The collective sparse attacks model assumes that attackers know the whole topology of the network and the Jacobian measurement matrix $\mathbf{H}$. However, in this case the attacks are directed at a group of state vector variables, *i.e.*, each attack injects false data into the state vector variables of the corresponding cluster. Within this setting, (38) can be rearranged as

$$\mathbf{a} = \begin{bmatrix} \hat{\mathbf{H}}_1 \cdots \hat{\mathbf{H}}_G \end{bmatrix} \begin{bmatrix} \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_G \end{bmatrix}, \qquad (43)$$

where the injection vector $\mathbf{c}_i \in \mathbb{R}^{D_i}$ is computed by the i-th attacker using Jacobian measurement submatrix $\hat{\mathbf{H}}_i$ for $i = 1, ..., G$. Following this decomposition, the optimization can be posed as

$$\text{minimize} \quad \|\mathbf{H}\mathbf{c} - \mathbf{a}\|_2^2 + \lambda \sum_{i=1}^{G} \|\mathbf{c}_i\|_2. \qquad (44)$$

Introducing the optimization variables, $\boldsymbol{\psi} \in \mathbb{R}^D$, yields a new formulation

$$\text{minimize} \quad \|\mathbf{H}\boldsymbol{\psi} - \mathbf{a}\|_2^2 + \lambda \sum_{i=1}^{G} \|\mathbf{c}_i\|_2 \tag{45}$$

$$\text{subject to} \quad \mathbf{c}_i - \hat{\boldsymbol{\psi}}_i = \mathbf{0}\ , i = 1, ..., G\ , \tag{46}$$

where $\hat{\boldsymbol{\psi}}_i$ is the estimate of $\boldsymbol{\psi}$ for $\mathbf{c}_i$ [16]. In the same fashion as with the previous problem, the optimization problem (46) is the same as (32) and therefore, Algorithm 3 can be used to solve it.

VI. Computational Complexity of Distributed Algorithms

We solve the distributed optimization problems using two main approaches, namely *measurement distributed* and *attribute distributed* optimization as given in Algorithm 2 and Algorithm 3 respectively. In the measurement distributed approach, we assume that the measurements are distributed and the local solutions of the optimization algorithms are computed in the clusters. In the attribute distributed approach, we assume that the state or attack vector variables are distributed and local estimates are computed in the clusters.

If we ignore communication times required to *gather* and *broadcast* the locally estimated vectors $\tilde{\mathbf{x}}_i$ and local variables $\mathbf{u}_i$, then the computational complexity of Algorithm 2 is dominated by the $\tilde{\mathbf{x}}_i$-update step in (27), $\forall i = 1, \ldots, G$. Since the partitioned Jacobian matrix $\mathbf{H}_i$ is used in (27), the computational complexity of (27) is $O(\alpha_i^3)$, where $\alpha_i = \min(N_i, D)$ in each cluster $\mathcal{G}_i$. Then, the complexity of Algorithm 2 is $\Upsilon_2 \in \max\left(t' O(\alpha_1^3), \ldots, t' O(\alpha_G^3)\right)$, since a central processor which employs the third step of the algorithm should wait to gather all the local estimates from the processors in the clusters. If we define the maximum communication complexity of gathering the local data as Υ_g and that of broadcasting as Υ_b, then the complexity of Algorithm 2 is increased to $\Upsilon_2 + \Upsilon_g + \Upsilon_b$.

Similarly, G parallel regularized least squares problems are solved in G_i variables in the $\tilde{\mathbf{x}}_i$-update step (35) of Algorithm 3. Since data partitioning by attribute is employed, the computational complexity of (35) is $O(\alpha_i^3)$, where $\alpha_i = \min(N, D_i)$. Similarly, the complexity of Algorithm 3 is $\Upsilon_3 \in \max\left(t' O(\alpha_1^3), \ldots, t' O(\alpha_G^3)\right)$, and the communication cost increases the complexity to $\Upsilon_3 + \Upsilon_g + \Upsilon_b$.

In the implementations, several practical tricks such as caching can be used to decrease the computational complexity of the local optimization algorithms (27) and (35). For further details, please refer to [16].

VII. Numerical Results

In this section, the validity of the proposed algorithms is assessed by numerically evaluating the performance of the algorithms for IEEE 9-Bus, IEEE 30-Bus, IEEE 57-Bus and IEEE 118-Bus test systems [21]. For each data point 100 realizations are simulated. For all simulation results, λ is fixed as [16]

$$\lambda = C\lambda_{max}, \tag{47}$$

where C is a constant, $\lambda_{max} = \|\mathbf{H}\tilde{\mathbf{z}}\|_\infty$ for distributed sparse state vector estimation methods, and $\lambda_{max} = \|\mathbf{H}\mathbf{a}\|_\infty$ for distributed attack models. In addition, λ_{max} can be considered as a critical value of the regularization parameter λ above which the estimated state and attack vectors take zero values. Consequently, C determines the *sparsity* of the solutions of the optimization problems and the number of iterations required to obtain the solutions, *i.e.* the estimated state and attack vectors. For that reason, an *optimal* $\hat{\lambda}$ or $\hat{C}$ is computed by analyzing the solution (or regularization) path of the optimization algorithms using a given training dataset. A detailed analysis of the impact of C on the number of algorithm iterations required to obtain an optimal solution is given in [16] for ADMM implementations of LASSO type algorithms. We choose the penalty parameter as $\rho = 1$, the absolute tolerance as 10^{-4}, the relative tolerance as 10^{-2} and set the maximum number of iterations $t' = 10000$.

In the experiments, it is assumed that the attacker has access to k measurements. For each realization, a k-sparse attack vector, $\mathbf{a}$, is randomly generated by selecting the non-zero indices following a uniform distribution and Gaussian distributed amplitudes with the same mean and variance values as $\mathbf{z}$. For distributed instances of the problem, the number of clusters, G, is uniformly distributed from the set of all prime divisors of N. On the other hand, for the collaborative instances, G is chosen from the set of all prime divisors of D.

A. Results for Centralized Data Injection Attacks

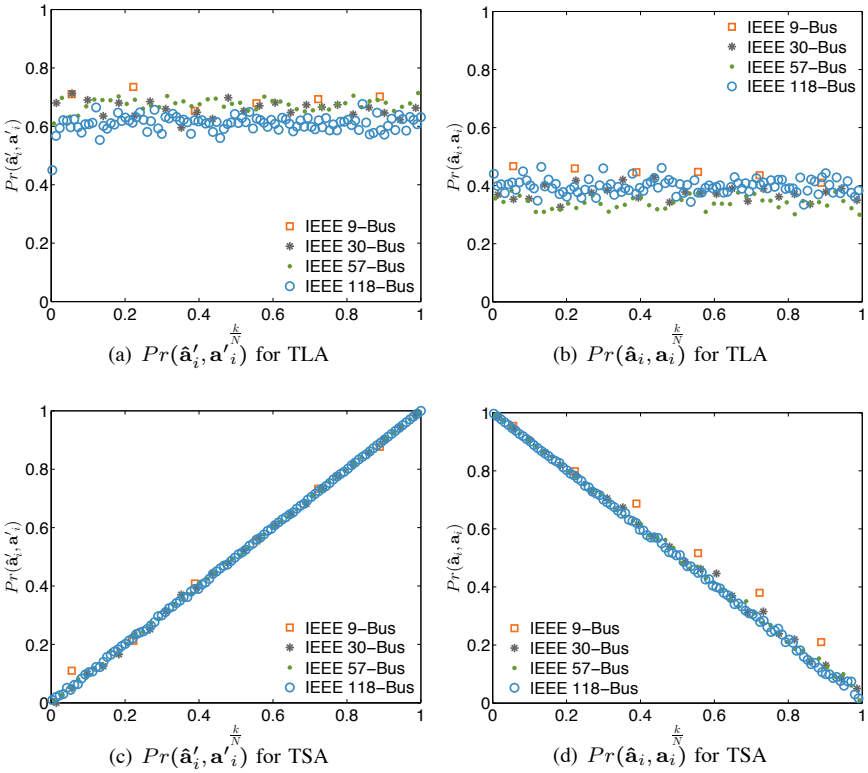
In order to assess the performance, the following parameters are computed in the simulations:

1) $Pr(\hat{\mathbf{a}}_i \neq 0, \mathbf{a}_i \neq 0)$ or simply $Pr(\hat{\mathbf{a}}'_i, \mathbf{a}'_i)$, which is the probability of correctly constructing an attack variable $\hat{\mathbf{a}}_i \neq 0$ of a false data injection vector $\mathbf{a}$.
2) $Pr(\hat{\mathbf{a}}_i = 0, \mathbf{a}_i = 0)$ or simply $Pr(\hat{\mathbf{a}}_i, \mathbf{a}_i)$, which is the probability of correctly constructing a secure variable $\hat{\mathbf{a}}_i = 0$ of a false data injection vector $\mathbf{a}$.

Since $Pr(\hat{\mathbf{a}}'_i, \mathbf{a}'_i) + Pr(\hat{\mathbf{a}}_i, \mathbf{a}'_i) = 1$ and $Pr(\hat{\mathbf{a}}_i, \mathbf{a}_i) + Pr(\hat{\mathbf{a}}'_i, \mathbf{a}_i) = 1$, probabilities of incorrect constructions can be computed from the results.

False data construction probabilities of Targeted LASSO Attacks (TLA), Strategic LASSO Attacks (SLA), Targeted Selective Attacks (TSA) and Strategic Selective Attacks (SSA) are compared in the following.

In Figure 1, the experiments for TLA and TSA are analyzed and the changes of false data vector construction probabilities are depicted for a varying number of attack variables, $\frac{k}{N}$, for each test system. The construction probabilities do not increase or decrease smoothly for TLA (Figures 1.a and 1.b.), since λ is computed dynamically by (47) for each realization and test system. Therefore, sparseness is not explicitly controlled in LASSO Attacks. On the other hand, the dynamic computation of λ using (47) enables estimation of the sparseness of the attacks and the randomness in $\mathbf{H}$. Therefore, the false data vector $\mathbf{a}$ is constructed with similar probabilities independent of the test system and sparsity level $\frac{k}{N}$ of the attack vectors in the TLA case. For instance, $Pr(\hat{\mathbf{a}}'_i, \mathbf{a}'_i)$ obtains values in the range $[0.5, 0.7]$ in Figure 1.a and $Pr(\hat{\mathbf{a}}_i, \mathbf{a}_i)$ obtains



(a) $Pr(\hat{\mathbf{a}}'_i, \mathbf{a}'_i)$ for TLA (b) $Pr(\hat{\mathbf{a}}_i, \mathbf{a}_i)$ for TLA

(c) $Pr(\hat{\mathbf{a}}'_i, \mathbf{a}'_i)$ for TSA (d) $Pr(\hat{\mathbf{a}}_i, \mathbf{a}_i)$ for TSA

Fig. 1. False data vector construction probabilities for for TLA and TSA.

values in the range $[0.3, 0.5]$ in Figure 1.b. Since sparseness can be controlled in Selective Attacks, a smooth change of the construction probabilities of false data vector variables is observed for TSA in Figure 1.c and Figure 1.d.

Note that, if the regularization parameter for LASSO is optimized, the solution vectors for LASSO and Regressor selection algorithms coincide [16]. In Figure 2, it can be seen that similar solutions are attainable, *i.e.* both methods construct similar attack vectors. For instance, we observe that the attacked variable construction probabilities increase similarly in Figures 2.a and 2.c, while secure variable construction probabilities decrease similarly in Figures 2.b and 2.d for SLA and SSA.

B. Results for Attack Detection using Distributed Sparse State Vector Estimation

In this work, our primary interest from the network operator's point of view is the distributed estimation of the state vectors. In this section, we analyze the proposed state vector estimation methods by employing them for the attack detection problem using a modified Normalized Residual Test (NRT) procedure [6].

In the attack detection procedure, we first estimate the state vectors $\hat{\mathbf{x}}$ using the algorithms proposed in Section III. Then, the error of the system is computed as $\|\mathbf{z}-\mathbf{H}\hat{\mathbf{x}}\|_2^2$ and the residual of an observed measurement i is given by $\|z_i-(\mathbf{H}\hat{\mathbf{x}})_i\|_2^2$, where $(\cdot)_i$ denotes the i-th element of the argument vector. Following the classical detection criterion, it is declared that the observation i is attacked if $\|z_i-(\mathbf{H}\hat{\mathbf{x}})_i\|_2^2 > \tau$. Since our goal is to detect the attacks on specific measurements, we do not remove the attacked measurement vectors at each iteration of the algorithm unlike the NRT method proposed in [6]. In addition, such a removal process disturbs the data space. Therefore, the proposed estimation methods should be re-implemented and the regularization parameters should be re-estimated on the updated datasets, leading to additional computational costs.

In the experiments, both algorithms operate with fixed parameter $C = \frac{1}{2}$ for the regularization parameter λ. In addition, τ is chosen as $2\xi_n\|I-\mathbf{H}(\mathbf{H}^T\Sigma_n^{-1}\mathbf{H})^{-1}\mathbf{H}^T\Sigma_n^{-1}\|_\infty$, where ξ_n and Σ_n are the variance and the covariance matrix of the noise $\mathbf{n}$ in (1) respectively, as suggested in [26].

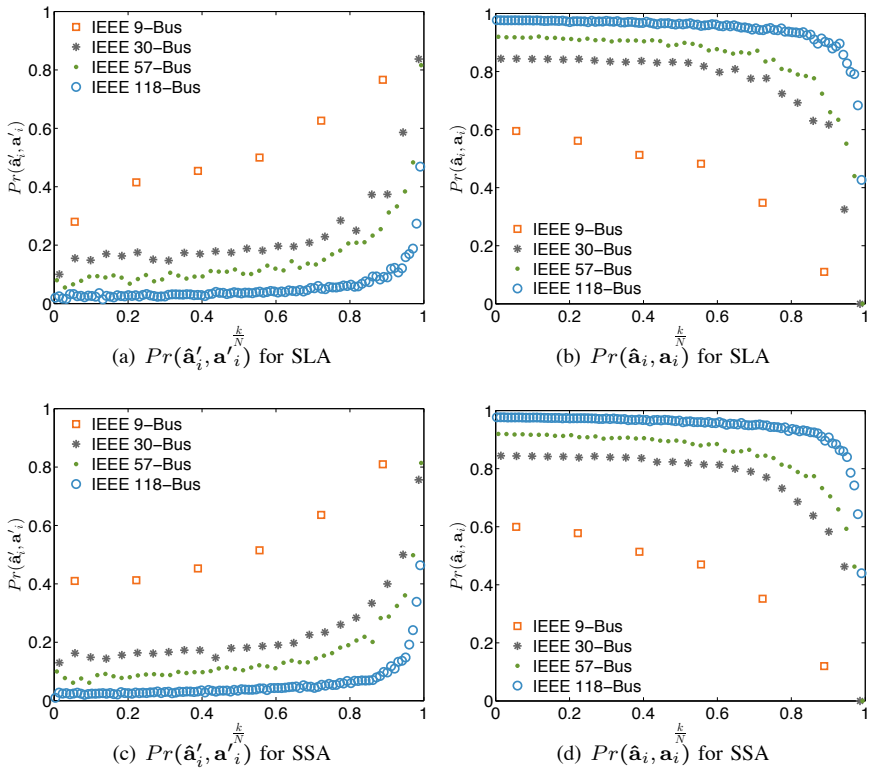
In this section, we construct the attack vectors using *Random False Data Injection Attacks* when the attacker has access to any k meters to construct k-sparse attack vectors $\mathbf{a}$, as suggested by Liu, Ning and Reiter [1].

Performance indices Precision (*Prec*), Recall (*Rec*) and Accuracy (*Acc*) are defined as

$$Prec = \frac{tp}{tp+fp}, \quad Rec = \frac{tp}{tp+fn}, \quad Acc = \frac{tp+tn}{tp+tn+fn+fp}, \quad (48)$$

where true positive (*tp*), true negative (*tn*), false positive (*fp*), and false negative (*fn*) are defined in Table II. For instance, *tp* represents the number of attacked measurements that are correctly detected. On the other hand, *fp* represents the number of secure measurements that are wrongly declared as attacked. Note that, $Prec$ is equal to $Pr(\hat{\mathbf{a}}_i = \mathbf{a}_i)$, which is the probability that a network operator can successfully detect k specific attacks for all $\hat{\mathbf{a}}_i \neq 0$.

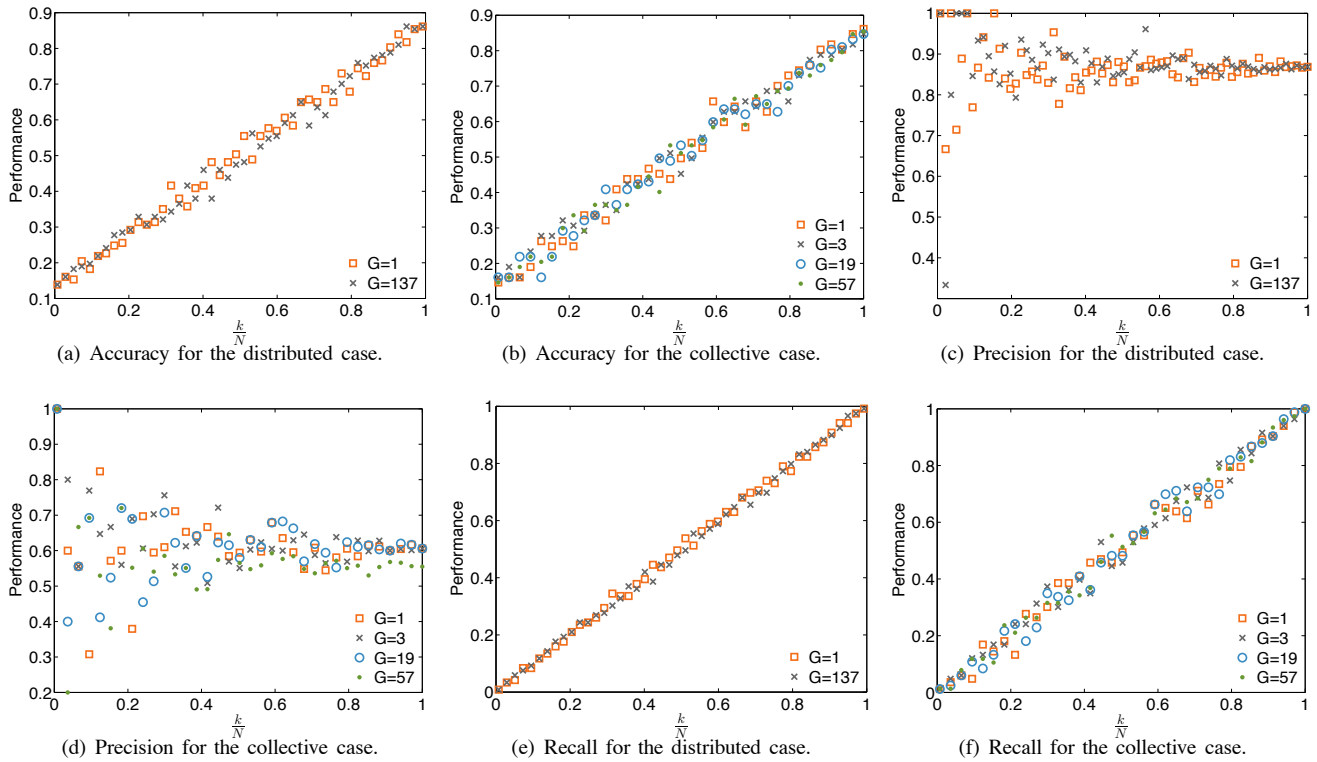
Results for different numbers of measurement clusters, G, are considered in Figure 3. In this experiment, each operator has access to locally observed measurements, state vectors and submatrices. The simulated cases are $G = |N|$ and $G = |D|$, for



(a) $Pr(\hat{\mathbf{a}}'_i, \mathbf{a}'_i)$ for SLA (b) $Pr(\hat{\mathbf{a}}_i, \mathbf{a}_i)$ for SLA

(c) $Pr(\hat{\mathbf{a}}'_i, \mathbf{a}'_i)$ for SSA (d) $Pr(\hat{\mathbf{a}}_i, \mathbf{a}_i)$ for SSA

Fig. 2. False data vector construction probabilities for SLA and SSA.



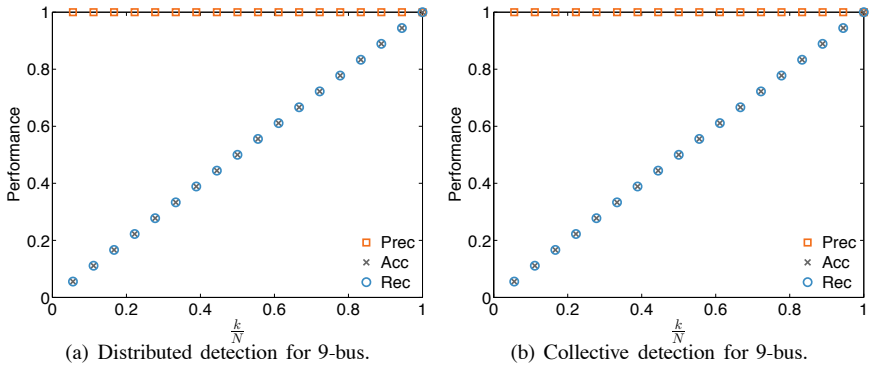
(a) Accuracy for the distributed case. (b) Accuracy for the collective case. (c) Precision for the distributed case.

(d) Precision for the collective case. (e) Recall for the distributed case. (f) Recall for the collective case.

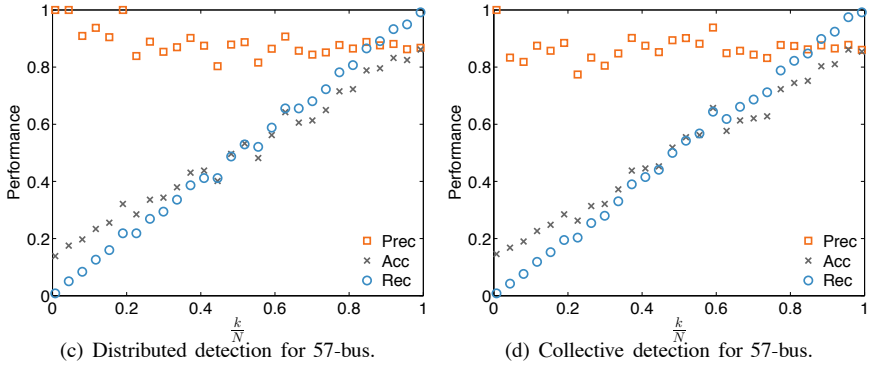
Fig. 3. Experiments for the IEEE 57-bus test system with various G values.

distributed and collective state estimation algorithms respectively. Therefore, $G = |N|$ and $G = |D|$ are the extreme cases for distributed processing scenarios. However, the algorithms have similar performance for different values of G in Figure 3, which shows that the optimality loss with respect to centralized strategies is small in the simulated settings.

Figure 4 shows the results of distributed and collective state vector estimation algorithms for $G = 1$. Note that, the case



(a) Distributed detection for 9-bus. (b) Collective detection for 9-bus.



(c) Distributed detection for 57-bus. (d) Collective detection for 57-bus.

Fig. 4. Distributed estimation performance indices for IEEE 9-bus and IEEE 57-bus test systems.

TABLE II
DEFINITIONS OF *tp*, *fp*, *tn*, AND *fn*

	Attacked	Secure
Classified as Attacked	*tp*	*fp*
Classified as Secure	*fn*	*tn*

$G = 1$ represents a centralized processing scenario in which all of the observed measurements and the whole Jacobian measurement matrix are available to network operators. It can be seen that the performance values of distributed and collective estimation methods are similar for the IEEE-9-bus test system in Figure 4.a and Figure 4.b respectively. However, the results in Figures 4.c and 4.d show that for low values of k/N the precision fluctuates and stabilizes around 0.9 as k/N increases for the IEEE-57-bus test system. In addition, the slopes of the curves representing the increases of accuracy and recall values are slightly smaller in Figures 4.c and 4.d than the ones for the IEEE-9-bus test system case.

C. Results for Distributed and Collective Sparse Attacks

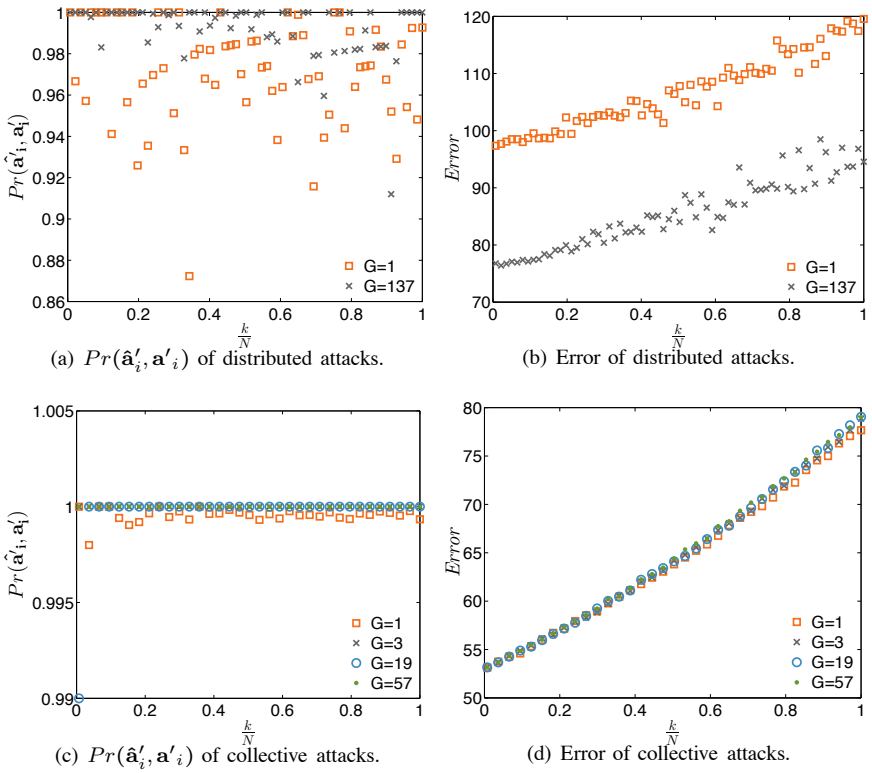
In order to measure the detectability of the attacks from the perspective of the network operators, $Error = \|z_i - (\mathbf{H}\hat{\mathbf{x}})_i\|_2^2$ is considered. Throughout this section, both algorithms operate with fixed parameter $C = \frac{1}{2}$.

In Figure 5, the results of the distributed attack experiments for the IEEE-57-bus test system are shown. Remarkably, the proposed algorithms are capable of successfully injecting data with high probability for a large range of sparsity ranges. However, it can be seen in Figures 5.a and 5.b that $Pr(\hat{\mathbf{a}}_i', \mathbf{a'}_i)$ decreases and $Error$ increases as G decreases. This is due to the fact that the optimization is very sensitive to the optimization of the regularization parameter λ. Interestingly, in the simulation settings evaluated for this paper, it has been observed that the proposed algorithms are more robust to variations of λ when smaller values of G are considered. It is known [27] that the optimization of the regularization parameter in the centralized case is hard. Surprisingly, as the fragmentation of the optimization problem increases, i.e., for lower values of G, the performance of the algorithm is less sensitive to the tuning of the regularization parameter. For instance, the injection vectors are computed and the optimization variables are updated locally in each group with respect to a global regularization parameter in distributed and collective attacks. In other words, the group-wise local regularization paths (i.e., the set of solutions) are computed and used to approximate a global regularization path. Since the paths of Group LASSO are piecewise differentiable, approximating the global path by the local paths may be a challenge as G increases. A solution to this challenge is to compute group-wise parameters in an adaptive scheme [28].

VIII. CONCLUSION

In this paper, we have considered centralized and distributed models for sparse attack construction and state estimation in the smart grid. For a centralized scenario, two methods, LASSO Attacks and Selective Attacks, have been introduced for the construction of false data vectors and attack vectors for a given attack model. The presented methods are used in two well-known attack models, Targeted and Strategic Attacks.

We have shown that Selective Attacks provide control of the sparsity of the attack vectors, explicitly. Therefore, a construction method has been proposed for false data and attack



(a) $Pr(\hat{\mathbf{a}}'_i, \mathbf{a}'_i)$ of distributed attacks.

(b) Error of distributed attacks.

(c) $Pr(\hat{\mathbf{a}}'_i, \mathbf{a}'_i)$ of collective attacks.

(d) Error of collective attacks.

Fig. 5. Experiments for the IEEE 57-bus test system.

vectors, which contain a given number of attacked and secure variables. Incidentally, the randomness of the parameters of the attack models may decrease the unobservability of the attack vectors and the control for the construction of false data vectors. For instance, random construction of the sub-matrices in Targeted Attacks may inject additional randomness into the probabilities of constructing false data vectors $\mathbf{a}$.

For the case in which the distributed nature of the network is considered, new distributed sparse state vector estimation and attack detection methods have been introduced. In the *Distributed State Vector Estimation* method, it is assumed that the observed measurements are distributed in clusters in the network. The state vectors are estimated using local data measurements in the clusters by either local network operators or PMUs. The estimates are then updated by centralized processors. In *Collaborative Sparse State Vector Estimation*, operators estimate a subset of variables of the state vectors. Therefore, state vector variables are assumed to be distributed in groups and accessed by the network operators locally. In this scenario, network operators compute their local estimates and send the estimated values to a centralized network operator in order to update the estimated values.

In the experiments, it has been observed that both state vector estimation methods perform similarly for a varying number of attacks in different test systems. Besides, accuracy and precision values of the proposed methods decrease as the system size increases and the performance values do not change as the number of clusters increases. In other words, we can achieve similar performance when we implement the algorithms in centralized ($G = 1$) and massively distributed scenarios ($G = N$ or $G = D$).

When the *Distributed Sparse Attacks* model is considered, it is assumed that attackers process only local measurements in order to achieve a consensus for attack vectors. In the *Collective Sparse Attacks* case, the topological information of the network and the measurements is available to attackers. However, the attackers employ attacks on variable groups of state vectors.

It has been observed in the experiments that the *Collective Sparse Attacks* model performs better than the *Distributed Sparse Attacks* model for the construction of unobservable attack vectors. Surprisingly, better performance of the algorithm with higher G values is achieved than smaller G values when larger systems are considered. This is due to the fact that one of the challenges of the proposed methods is the estimation of algorithm parameters, e.g., the maximum number of iterations and the regularization parameter λ. For the case in which the sparsity degree, k, of the solution vectors are known *a priori*, regressor selection algorithms can be employed in order to control the sparsity of the solutions.

REFERENCES

[1] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," in *Proc. 16th ACM Conf. Comput. Commun. Security*, ser. CCS '09. New York, NY, USA: ACM, 2009, pp. 21–32.

[2] L. Sankar, S. Kar, R. Tandon, and H. V. Poor, "Competitive privacy in the smart grid: An information-theoretic approach," in *Proc. 2nd IEEE Int. Conf. Smart Grid Communications*, Brussels, Belgium, Oct. 2011, pp. 220–225.

[3] E. Cotilla-Sanchez, P. Hines, C. Barrows, and S. Blumsack, "Comparing the topological and electrical structure of the North American electric power infrastructure," *IEEE Syst. J.*, vol. 6, no. 4, pp. 616–626, Dec. 2012.

[4] T. T. Kim and H. V. Poor, "Strategic protection against data injection attacks on power grids," *IEEE Trans. Smart Grid*, vol. 2, no. 2, pp. 326–333, 2011.

[5] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, "Malicious data attacks on the smart grid," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 645–658, 2011.

[6] A. Abur and A. Expósito, *Power System State Estimation: Theory and Implementation*, ser. Power Engineering. Marcel Dekker, 2004.

[7] A. Tajer, S. Kar, H. V. Poor, and S. Cui, "Distributed joint cyber attack detection and state recovery in smart grids," in *Proc. 2nd IEEE Int. Conf. Smart Grid Communications*, Brussels, Belgium, Oct. 2011, pp. 202–207.

[8] M. Ozay, I. Esnaola, F. T. Y. Vural, S. R. Kulkarni, and H. V. Poor, "Distributed models for sparse attack construction and state vector estimation in the smart grid," in *Proc. 3rd IEEE Int. Conf. Smart Grid Communications*, Tainan City, Taiwan, Nov. 2012.

[9] L. Xie, D.-H. Choi, S. Kar, and H. V. Poor, "Fully distributed state estimation for wide-area monitoring systems," *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1154–1169, Sept. 2012.

[10] F. Pasqualetti, R. Carli, and F. Bullo, "A distributed method for state estimation and false data detection in power networks," in *Proc. 2nd IEEE Int. Conf. Smart Grid Communications*, Brussels, Belgium, Oct. 2011, pp. 469–474.

[11] Q. Yang, J. Yang, W. Yu, N. Zhang, and W. Zhao, "On a hierarchical false data injection attack on power system state estimation," in *Proc. IEEE Global Communications Conference (GLOBECOM 2011)*, Houston, TX, USA, Dec. 2011, pp. 1–5.

[12] O. Vukovic, K. C. Sou, G. Dan, and H. Sandberg, "Network-aware mitigation of data integrity attacks on power system state estimation," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 6, pp. 1108–1118, July 2012.

[13] Y.-F. Huang, S. Werner, J. Huang, N. Kashyap, and V. Gupta, "State estimation in electric power grids: Meeting new challenges presented by the requirements of the future grid," *IEEE Signal Process. Mag.*, vol. 29, no. 5, pp. 33–43, Sept. 2012.

[14] W. Wang and Z. Lu, "Cyber security in the smart grid: Survey and challenges," *Comput. Netw.*, 2013, in press.

[15] D. P. Chassin and C. Posse, "Evaluating north american electric grid reliability using the Barabási-Albert network model," *Physica A: Statistical Mechanics and its Applications*, vol. 355, no. 2-4, pp. 667–677, Sep. 2005.

[16] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[17] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, "On the optimal step-size selection for the alternating direction method of multipliers," in *Proc. 3rd IFAC Workshop on Distributed Estimation and Control in Networked Systems*, Santa Barbara, CA, USA, Sept. 2012.

[18] B. He and X. Yuan, "On the o(1/n) convergence rate of the Douglas-Rachford alternating direction method," *SIAM J. Numer. Anal.*, vol. 50, no. 2, pp. 700–709, Apr. 2012.

[19] E. J. Candès and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.

[20] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[21] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER: Steady-state perations, planning, and analysis tools for power systems research and education," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, Feb. 2011.

[22] R. Tibshirani, "Regression shrinkage and selection via the LASSO," *J. R. Stat. Soc. (Series B)*, vol. 58, pp. 267–288, 1996.

[23] Z. Zhang, G. Dai, C. Xu, and M. I. Jordan, "Regularized discriminant analysis, ridge regression and beyond," *J. Mach. Learn. Res.*, vol. 11, pp. 2199–2228, Aug. 2010.

[24] A. Tikhonov and V. Arsenin, *Solutions of Ill-posed Problems*, ser. Scripta Series in Mathematics. Winston, 1977.

[25] M. de Almeida, A. Garcia, and E. Asada, "Regularized least squares power system state estimation," *IEEE Trans. Power Syst.*, vol. 27, no. 1, pp. 290–297, Feb. 2012.

[26] F. Pasqualetti, R. Carli, and F. Bullo, "A distributed method for state estimation and false data detection in power networks," in *Proc. 2nd IEEE Int. Conf. Smart Grid Communications*, Brussels, Belgium, Oct. 2011, pp. 469–474.

[27] S. Rangan, A. K. Fletcher, and V. K. Goyal, "Asymptotic analysis of MAP estimation via the replica method and applications to compressed sensing," *IEEE Trans Inf. Theory*, vol. 58, no. 3, pp. 1902–1923, 2012.

[28] F. R. Bach, "Consistency of the group LASSO and multiple kernel learning," *J. Mach. Learn. Res.*, vol. 9, pp. 1179–1225, Jun. 2008.

Mete Ozay is with the Department of Computer Engineering at Middle East Technical University. He was a visiting Ph.D. student and research collaborator at the Department of Electrical Engineering, Princeton University in 2011-2012. His research interests include pure and applied mathematics, theoretical computer science, and physics.

Iñaki Esnaola received the M.S. degree in Electrical Engineering from University of Navarra, Donostia, Spain in 2006 and a Ph.D. in Electrical Engineering from University of Delaware, Newark, DE in 2011. He is currently a Postdoctoral Research Associate at Princeton University, Princeton, NJ. In 2010-2011 he was a Research Intern with Bell Laboratories, Alcatel-Lucent, Holmdel, NJ. His research interests include information theory and communication theory.

Fatos T. Yarman Vural received the B.S. degree with honors in electrical engineering from the Technical University of Istanbul in 1973, the M.S. degree in electrical engineering from Bogazici University in 1975, and the Ph.D. degree in electrical engineering and computer science from Princeton University in 1981. From 1981 to 1983, she was a research scientist in Marmara Research Institute in Turkey. From 1983 to 1985, she was a visiting professor at Drexel University. From 1985 to 1992, she was a technical and deputy manager at Yapitel Inc. Since 1992, she has been a professor in the Computer Engineering Department of Middle East Technical University (METU). Her research area covers computer vision, image processing, pattern recognition, and artificial intelligence. She has been involved in teaching, consulting, and organizing conferences in these areas. She was the chair woman in the Computer Engineering Department between 1996-2000. She is a senior member of IEEE and a member of Turkish Informatics Foundation, Turkish Information Association, and chamber of Electrical Engineers in Turkey.

Sanjeev R. Kulkarni (M'91, SM'96, F'04) received the B.S. in Mathematics, B.S. in E.E., and M.S. in Mathematics from Clarkson University in 1983, 1984, and 1985, respectively; the M.S. degree in E.E. from Stanford University in 1985; and the Ph.D. in E.E. from M.I.T. in 1991. From 1985 to 1991 he was a Member of the Technical Staff at M.I.T. Lincoln Laboratory working on the modelling and processing of laser radar measurements. In the spring of 1986, he was a part-time faculty member at the University of Massachusetts, Boston. Since 1991 he has been with Princeton University, where he is currently Professor of Electrical Engineering and an affiliated faculty member in the Department of Operations Research and Financial Engineering and the Department of Philosophy. He spent January 1996 as a research fellow at the Australian National University, 1998 with Susquehanna International Group, and summer 2001 with Flarion Technologies. Prof. Kulkarni received an ARO Young Investigator Award in 1992, an NSF Young Investigator Award in 1994, and several teaching awards at Princeton. He has served as an Associate Editor for the IEEE Transactions on Information Theory. Prof. Kulkarni's research interests include statistical pattern recognition, nonparametric estimation, learning and adaptive systems, information theory, wireless networks, and image/video processing.

H. Vincent Poor (S'72, M'77, SM'82, F'87) received the Ph.D. degree in EECS from Princeton University in 1977. From 1977 until 1990, he was on the faculty of the University of Illinois at Urbana-Champaign. Since 1990 he has been on the faculty at Princeton, where he is the Michael Henry Strater University Professor of Electrical Engineering and Dean of the School of Engineering and Applied Science. He has also held visiting appointments at several other institutions, including most recently Imperial College and Stanford. Dr. Poor's research interests are in the areas of stochastic analysis, statistical signal processing and information theory, and their applications in wireless networks and related fields including social networks and smart grid. Among his publications in these areas are the recent books *Smart Grid Communications and Networking* (Cambridge University Press, 2012) and *Principles of Cognitive Radio* (Cambridge University Press, 2013).

Dr. Poor is a member of the National Academy of Engineering and the National Academy of Sciences, a Fellow of the American Academy of Arts and Sciences, an International Fellow of the Royal Academy of Engineering (U. K), and a Corresponding Fellow of the Royal Society of Edinburgh. He is also a Fellow of the IET, the Optical Society of America, and other scientific and technical organizations. In 1990, he served as President of the IEEE Information Theory Society, and in 2004-07 he served as the Editor-in-Chief of the IEEE TRANSACTIONS ON INFORMATION THEORY. He received a Guggenheim Fellowship in 2002, the IEEE Education Medal in 2005, and the Marconi and Armstrong Awards of the IEEE Communications Society in 2007 and 2009, respectively. Recent recognition of his work includes the 2010 IET Ambrose Fleming Medal for Achievement in Communications, the 2011 IEEE Eric E. Sumner Award, a Royal Academy Distinguished Visiting Fellowship (2012), and honorary doctorates from Aalborg University, the Hong Kong University of Science and Technology, and the University of Edinburgh.