False Data Injection Attacks against State Estimation in Electric Power Grids

Yao Liu and Peng Ning
North Carolina State University
and
Michael K. Reiter
University of North Carolina at Chapel Hill

A power grid is a complex system connecting electric power generators to consumers through power transmission and distribution networks across a large geographical area. System monitoring is necessary to ensure the reliable operation of power grids, and *state estimation* is used in system monitoring to best estimate the power grid state through analysis of meter measurements and power system models. Various techniques have been developed to detect and identify bad measurements, including *interacting bad measurements* introduced by *arbitrary, non-random* causes. At first glance, it seems that these techniques can also defeat malicious measurements injected by attackers.

In this paper, we expose an unknown vulnerability of existing bad measurement detection algorithms by presenting and analyzing a new class of attacks, called *false data injection attacks*, against state estimation in electric power grids. Under the assumption that the attacker can access the current power system configuration information and manipulate the measurements of meters at physically protected locations such as substations, such attacks can introduce *arbitrary* errors into certain state variables without being detected by existing algorithms. Moreover, we look at two scenarios, where the attacker is either constrained to specific meters or limited in the resources required to compromise meters. We show that the attacker can systematically and efficiently construct attack vectors in both scenarios to change the results of state estimation in *arbitrary* ways. We also extend these attacks to *generalized false data injection attacks*, which can further increase the impact by exploiting measurement errors typically tolerated in state estimation. We demonstrate the success of these attacks through simulation using IEEE test systems, and also discuss the practicality of these attacks and the real-world constraints that limit their effectiveness.

Categories and Subject Descriptors: K.6.5 [**Management of Computing and Information Systems**]: Security and Protection

General Terms: Algorithms, Security

Additional Key Words and Phrases: Power grids, state estimation, attack

Authors' Addresses: Y. Liu and P. Ning, Department of Computer Science, North Carolina State University, emails: {yliu20,pning}@ncsu.edu; M. K. Reiter, Department of Computer Science, University of North Carolina at Chapel Hill, email: reiter@cs.unc.edu.

A preliminary version of this paper appeared in ACM CCS'09 [Liu et al. 2009].

1. INTRODUCTION

A power grid is a complex system connecting a variety of electric power generators to customers through power transmission and distribution networks across a large geographical area, as illustrated in Figure 1 (adapted from [National Security Telecommunications Advisory Committee (NSTAC) – Information Assurance Task Force (IATF)]). The security and reliability of power grids has critical impact on society. For example, on August 14, 2003, a large portion of the Midwest and Northeast United States and Ontario, Canada, experienced an electric power blackout, which affected an area with a population of about 50 million people. The estimated total costs ranged between $4 billion and $10 billion (U.S. dollars) in the United States, and totaled $2.3 billion (Canadian dollars) in Canada [U.S.-Canada Power System Outage Task Force 2004].

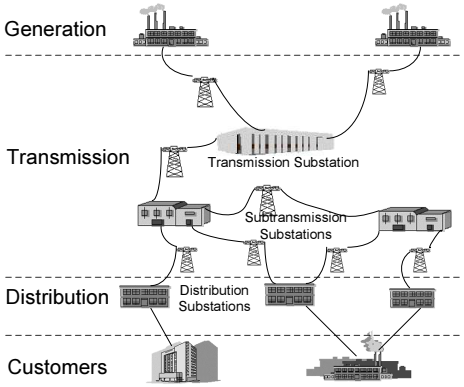

Fig. 1. A power grid connecting power plants to customers via power transmission and distribution networks

1.1 System Monitoring and State Estimation

System monitoring is necessary to ensure the reliable operation of power grids. It provides pertinent information on the condition of a power grid based on the readings of meters placed at important components of a power grid, such as substations. The meter measurements may include bus voltages, bus real and reactive power injections, and branch reactive power flows in every subsystem of a power grid. These measurements are typically transmitted to a *control center*, where the control center staff, with the assistance of computers, collect crucial system data and provide centralized monitoring and control capability for the power grid. Measurements are usually stored in a *telemetry system*, which is also known as *Supervisory Control And Data Acquisition (SCADA)* system.

State estimation is used in system monitoring to best estimate the power grid state through analysis of meter measurement data and power system models. State estimation is the process of estimating unknown state variables in a power grid based on the meter measurements. The control center staff use the output of state estimation as they perform contingency analysis, in which they reason about potential operational problems in the grid, the actions they may take to avoid those problems, and the potential side effects of those actions. For example, they may choose to increase the yield of a power generator in order to maintain reliable operation even in the presence of faults (e.g., a generator breakdown).

State estimation uses power flow models. A *power flow model* is a set of equations that depict the energy flow on each transmission line of a power grid. An *AC power flow model* is a power flow model that considers both real and reactive power and is formulated by nonlinear equations. State estimation using an AC power flow model can be computationally expensive and does not always converge to a solution. Thus, power system engineers sometimes use a linearized power flow model, *DC power flow model*, to approximate the AC power flow model [Li et al. 2008; Hertem et al. 2006].

1.2 Previous Defense against Bad Measurement in State Estimation

It is conceivable that an attacker may attempt to introduce malicious measurements to achieve her goals. For example, the attacker may directly compromise substation meters in a power system or hack computers that store meter measurements to inject malicious data. If these bad measurements affect the outcome of state estimation, the resulting misinformation can reduce the control center operators' level of situational awareness, thus helping the attacker reach or get closer to her malicious goals.

Power systems researchers have realized the threat of bad measurements and developed techniques for processing them (e.g., [Monticelli 1999; Mili et al. 1985; Monticelli and Garcia 1983; Monticelli et al. 1986; Mili et al. 1984; Lin and Pan 2007]). These techniques first detect if there are bad measurements, and then identify and remove the bad ones if there are any. Some of these techniques (e.g., [Monticelli 1999; Monticelli et al. 1986; Mili et al. 1984]) were targeted at *arbitrary*, interacting (i.e., correlated) bad measurements. At first glance, it seems that these approaches can also defeat the malicious measurements injected by attackers, since such malicious measurements can be considered as interacting bad measurements.

1.3 False Data Injection Attacks

However, in the research reported in this paper, we discover that if an attacker knows the current configuration of the power system, all existing algorithms for bad measurement detection and identification in DC power flow models have a common vulnerability that allows an attacker to bypass their safeguards. The fundamental reason for this failure is that all existing algorithms for bad measurement detection in DC power flow models rely on the same assumption that "when bad measurements take place, the squares of differences between the observed measurements and their corresponding estimates often become significant [Lin and Pan 2007]." Our investigation indicates that this assumption is not always true. If the attacker can determine the current power system configuration, she can systematically generate bad measurements so that the above assumption is violated, thus bypassing bad measurements detection.

In this paper, to gain insights of the aforementioned vulnerability, we present and analyze a new class of attacks, called *false data injection attacks*, against state estimation in electric power grids. If the attacker can determine the current configuration of a power system, she can inject malicious measurements that will mislead the state estimation process without being detected by any of the existing techniques for bad measurement detection. We also extend false data injection attacks to a generalized version, which we referred to as *generalized false data injection attacks*. In such an attack, an attacker can utilize the small measurement errors typically tolerated by state estimation algorithms so that she can further increase the impact of false data injection attacks without being detected.

In this paper, as the first step in our research, we focus on attacks against state estimation

using DC power flow models. We present false data injection attacks from the attacker's perspective. We first show that it is possible for the attacker to inject malicious measurements that can bypass existing techniques for bad measurement detection. We then look at two plausible attack scenarios. In the first attack scenario, the attacker is constrained to accessing some specific meters due to, for example, different physical protection of the meters. In the second attack scenario, the attacker is limited in the resources available to compromise meters. For both scenarios, we consider two possible attack goals: *random false data injection attacks*, in which the attacker aims to find any attack vector as long as it can lead to a wrong estimation of state variables, and *targeted false data injection attacks*, in which the attacker aims to find an attack vector that can inject *arbitrary* errors into certain state variables. We show that the attacker can systematically and efficiently construct attack vectors for false data injection attacks in both attack scenarios with both attack goals.

We further look at generalized false data injection attacks, which are extensions to false data injection attacks. The primary objective is to see if an attacker can achieve more impact by taking advantage of the small measurement errors typically tolerated by state estimation algorithms. As we did for false data injection attacks, we show how an attacker can construct a valid attack vector to bypass detection and inject errors to the outcome of state estimation in both attack scenarios with both attack goals. Moreover, we quantify the possible gains that generalized false data injection attacks offer through theoretical analysis.

We validate these attacks through simulation using IEEE test systems, including IEEE 9-bus, 14-bus, 30-bus, 118-bus, and 300-bus systems [Zimmerman and Murillo-Sánchez 2007]. The simulation results demonstrate the success of these attacks. For example, to inject a specific malicious value into one target state variable, the attacker only needs to compromise 10 meters in most cases in the IEEE 300-bus system, which has 1,122 meters in total. In addition, for generalized false data injection attacks, we perform simulation on IEEE test systems to examine the additional impact an attacker achieves beyond false data injection attacks. The simulation results show that even if the attacker fails to launch the original false data injection attacks, she can still inject errors to state estimation through the generalized version of attacks. Moreover, the impacts on large systems are greater than those on small systems, and errors injected to the estimates of certain state variables in large systems (e.g., IEEE 300-bus system) are significantly larger than those injected to the estimates of other state variables.

1.4 Requirements and Practical Implications

False data injection attacks do pose strong requirements for the attackers. First, the attackers must know the current configuration of the target power system, particularly the topology of the system. This system configuration changes frequently due to planned daily maintenance of power grid equipment and unplanned events such as unexpected equipment outage. Normally such information is only available at the control centers of power companies. Physical access to control centers is highly regulated and protected, given the sensitivity of the control centers. Thus, it is non-trivial for the attackers to obtain such configuration information to launch these attacks.

Another requirement for the attackers is the manipulation of the meter measurements. The attackers need to physically tamper with the meters, or manipulate the meter measurements before they are used for state estimation in the control center. Many of these meters

are located in places where there is protection against unauthorized physical accesses (e.g., substations). Thus, it is non-trivial to manipulate the meter measurements.

The primary benefit of studying false data injection attacks is to expose the vulnerability in existing state estimation techniques. The exact impact of such attacks depends not only on the introduced errors, but also how the measurement data (and thus the measurement errors) will be used in the end applications. In a typical application in the power grid today, control center personnel are usually involved in the decision making process. Experienced operators may be able to identify anomalies caused by such attacks. Additional research is necessary to clarify the implication of such attacks in different scenarios.

It should be noted that we assume DC power flow models in state estimation. For large power systems, nonlinearities become prominent, so that the DC power flow model is not accurate anymore. Hence, false data injection attacks based on the DC power flow model may lead to limited impact on large power systems. However, the DC power flow model is the starting point of our research, and the current results can serve as the foundation for future research on more complicated models than the DC model. As an example, following the preliminary version of this paper [Liu et al. 2009], recent work in [Sandberg et al. 2010] considered the AC power flow model and proposed a new targeted false data injection attack, whose goal is to manipulate one power flow measurement without triggering alarms [Sandberg et al. 2010]. This attack requires less knowledge about the system than the targeted attacks presented in this paper.

1.5 Organization

The rest of the paper is organized as follows. Section 2 gives some background information and discusses related work. Sections 3 and 4 present the basic principles of false data injection attacks and generalized false data injection attacks, respectively, and provide approaches for implementing both random and targeted false data injection attacks in the two attack scenarios. Section 5 demonstrates the success of these attacks through simulation. Section 6 concludes this paper and points out some future research directions.

2. PRELIMINARIES

Power System (Power Grid): A *power transmission system* (or simply a *power system*) consists of electric generators, transmission lines, and transformers that form an electrical network [Wood and Wollenberg 1996]. This network is also called a *power grid*. It connects a variety of electric generators together with a host of users across a large geographical area. Redundant paths and lines are provided so that power can be routed from any power plant to any customer, through a variety of routes, based on the economics of the transmission path and the cost of power. A control center is usually used to monitor and control the power system and devices in a geographical area.

State Estimation: In order to ensure that a power system continues to operate even when some components fail, power engineers use meters to monitor system components. Those meters take measurements such as real power injections of buses and real power flows of branches in the power system, and report their measurements to the control center, which then estimates the state variables of power system using meter measurements. Examples of state variables include bus voltage angles and magnitudes[1]. After obtaining estimates of

[1]In DC power flow model, voltage magnitudes and reactive power flows are of little concern, and thus state

state variables, the control center can decide whether or not the power system is operating properly. In simple terms, the state estimation problem is to estimate power system state variables using meter measurements.

A more precise definition of state estimation is given as follows. Let $\mathbf{x} = (x_1, x_2, ..., x_n)^T$ and $\mathbf{z} = (z_1, z_2, ..., z_m)^T$ denote state variables and meter measurements, respectively, where n is the number of state variables, m is the number of meter measurements, and $m \geq n$. Further let $\mathbf{e} = (e_1, e_2, ..., e_m)^T$ denote measurement errors. The state variables are related to the measurements through the model $\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{e}$ [Monticelli 1999], where $\mathbf{h}(\mathbf{x}) = (h_1(\mathbf{x}), ..., h_m(\mathbf{x}))^T$ and $h_i(\mathbf{x})$ is a function of $\mathbf{x}$. Given $\mathbf{z}$, the state estimation problem is to find the estimate $\hat{\mathbf{x}}$ of $\mathbf{x}$ according to this model.

For state estimation using the DC power flow model, the relation between measurements and state variables can be represented by a linear regression model

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

where $\mathbf{H}$ is an $m \times n$ full rank matrix to allow estimating $\mathbf{x}$ from $\mathbf{z}$ [Wood and Wollenberg 1996]. Three statistical estimation criteria are commonly used in state estimation: *the maximum likelihood criterion*, *the weighted least-square criterion*, and *the minimum variance criterion* [Wood and Wollenberg 1996]. When meter error is assumed to be normally distributed with zero mean, these criteria lead to an identical estimator (i.e., minimum mean squared error (MMSE) estimator) with the following matrix solution

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

where $\mathbf{W}$ is a diagonal matrix whose elements are reciprocals of the variances of meter errors. That is,

$$\mathbf{W} = \begin{bmatrix} \sigma_1^{-2} & & & & \\ & \sigma_2^{-2} & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \sigma_m^{-2} \end{bmatrix}, \tag{3}$$

where σ_i^2 is the variance of the i-th meter ($1 \leq i \leq m$).

Bad Measurement Detection: Bad measurements may be introduced due to various reasons such as meter failures and malicious attacks. Techniques for bad measurement detection have been developed to protect state estimation [Wood and Wollenberg 1996; Monticelli 1999]. Intuitively, normal meter measurements usually give an estimate of the state variables close to their actual values, while abnormal ones may "move" the estimated state variables away from their true values. Thus, there is usually "inconsistency" among the good and the bad measurements. Power systems researchers proposed to calculate the *measurement residual* $\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}$ (i.e., the difference between the vector of observed measurements and the vector of estimated measurements), and use its 2-Norm $\|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\|$ to detect the presence of bad measurements. Specifically, $\|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\|$ is compared with a threshold τ, and the presence of bad measurements is inferred if $\|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\| > \tau$. Note that there exist other bad measurement detection methods. For example, the normalized infinity-norm of the residual may be used to detect the presence of bad measurements [Abur

variables are usually voltage angles.

and Expósito 2004]. In this paper, we focus on 2-Norm detector, since it is one of the most commonly used bad measurement detectors.

The selection of τ is a key issue. Assume that all the state variables are mutually independent and the meter errors follow the normal distribution. It can be mathematically shown that $\|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\|^2$, denoted $\mathcal{L}(\mathbf{x})$, follows a $\chi^2(v)$-distribution, where $v = m - n$ is the degree of freedom. According to [Wood and Wollenberg 1996], τ can be determined through a hypothesis test with a significance level α. In other words, the probability that $\mathcal{L}(\mathbf{x}) \geq \tau^2$ is equal to α. Thus, $\mathcal{L}(\mathbf{x}) \geq \tau^2$ indicates the presence of bad measurements, with the probability of a false alarm being α.

2.1 Related Work

Many researchers have considered the problem of bad measurement detection and identification in power systems (e.g., [Mili et al. 1985; Schweppe et al. 1970; Handschin et al. 1975; Monticelli and Garcia 1983; Garcia et al. 1979; Xiang et al. 1982; 1983; Xiang and Wang 1981; Quintana et al. 1982; Monticelli 1999; Monticelli et al. 1986; Mili et al. 1984; Asada et al. 2005; Gastoni et al. 2003; Chen and Abur 2006; Zhao and Abur 2005; Chen and Abur 2005; Zhu and Abur 2007]). Early power system researchers realized the existence of bad measurements and observed that a bad measurement usually led to large normalized measurement residual. After the presence of bad measurements is detected, they mark the measurement having the largest normalized residual as the suspect and remove it [Schweppe et al. 1970; Handschin et al. 1975; Monticelli and Garcia 1983; Garcia et al. 1979; Xiang et al. 1982; 1983; Xiang and Wang 1981; Quintana et al. 1982]. For example, Schweppe et al. [1970] proposed to filter one measurement having the largest normalized residual at each loop, and then rerun the same process on the reduced measurement set until the detection test is passed. Handschin et al. [1975] proposed a grouped residual search strategy that can remove all suspected bad measurements at one time.

It was found that the largest normalized residual criterion only worked well for independent, non-correlated bad measurements called *non-interacting bad measurements* [Monticelli 1999; Monticelli et al. 1986; Mili et al. 1984]. In practice, there exist correlated bad measurements, which make the normalized residual of a good measurement the largest. Such bad measurements are called *interacting bad measurements*. The largest normalized residual method does not work satisfactorily in dealing with interacting bad measurements. To address this problem, Hypothesis Testing Identification (HTI) [Mili et al. 1984] and Combinatorial Optimization Identification (COI) [Monticelli et al. 1986; Asada et al. 2005; Gastoni et al. 2003] were developed. HTI selects a set of suspected bad measurements according to their normalized residuals, and then decides whether an individual suspected measurement is good or bad through hypothesis testing. COI uses the framework from the decision theory to identify multiple interacting bad measurements. For example, Asada et al. [2005] proposed an intelligent bad data identification strategy based on tabu search to deal with multiple interacting bad measurements.

Recently, the focus in bad measurement processing has been on the improvement of the robustness using phasor measurement units (PMUs) [Chen and Abur 2006; Zhao and Abur 2005; Chen and Abur 2005; Zhu and Abur 2007]. For example, Chen and Abur [2006] used PMUs to transform the critical measurements into redundant measurements such that the bad measurements can be detected by the measurement residual testing.

It seems that the approaches targeting at arbitrary, interacting bad measurements (e.g., [Mili et al. 1984; Monticelli et al. 1986; Asada et al. 2005; Gastoni et al. 2003]) can also

defeat the malicious ones injected by attackers, since such malicious measurements are indeed arbitrary, interacting bad measurements. However, despite the variations in these approaches, all of them use the same method (i.e., $\|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\| > \tau$) to detect the existence of bad measurements. Power engineers have realized the vulnerability of this detection approach (e.g., non-detectability of topology errors [Wu and Liu 1989]). However, to the best of our knowledge, our attempt is the first to consider this vulnerability from the perspective of attackers and systematically show how attackers can bypass detection and inject errors into the output of state estimation even if they are restrained in accesses and resources.

The preliminary version of this paper [Liu et al. 2009] has attracted several research groups to investigate how to defend against false data injection attacks (e.g., [Bobba et al. 2010; Sandberg et al. 2010; Kosut et al. 2010a; 2010c; 2010b; Dán and Sandberg 2010]). In particular, Bobba et al. [2010] provided a lower bound on the number of meters that need to be protected to thwart the attacks, Sandberg et al. [2010] introduced indices that quantify the least effort needed to achieve attack goals while avoiding detection by defenders, and Kosut et al. [2010c] proposed a Bayesian framework that leverages the knowledge of prior distribution on the states to detect false data injection attacks. All those works are complementary to ours. It should be noted that concurrent work [Bobba et al. 2010] also points to the existence of generalized false data injection attacks. However, important details such as methods to generate attack vectors and the impacts of the attacks are missing. In this work, we not only show that it is possible for the attackers to take advantage of the small errors tolerated by state estimation to cause extended impact, but also show how the attacker can generate attack vectors for different combinations of scenarios and goals and give a detailed analysis on the impacts of generalized false data injection attacks.

3. FALSE DATA INJECTION ATTACKS

We assume that there are m meters that provide m measurements $z_1, ..., z_m$ and there are n state variables $x_1, ..., x_n$. The relationship between these m meter measurements and n state variables can be characterized by an $m \times n$ matrix $\mathbf{H}$, as discussed in Section 2. In general, the matrix $\mathbf{H}$ of a power system is determined by the topology and line impedances of the system. How the control center constructs $\mathbf{H}$ is illustrated in [Monticelli 1999]. We also assume that the attacker can have access to the matrix $\mathbf{H}$ of the target power system, and can inject malicious measurements into compromised meters to undermine the state estimation process.

As discussed earlier, we consider two possible attack goals: *random false data injection attacks*, in which the attacker aims to find any attack vector as long as it can result in a wrong estimation of state variables, and *targeted false data injection attacks*, in which the attacker aims to find an attack vector that can inject a specific error into certain state variables. While the latter attacks can potentially cause more damage to the system, the former ones are easier to launch, as shown in Section 5.

Besides describing the basic false data injection attacks, we also use the following two plausible attack scenarios to facilitate the discussion on how the attacker can construct attack vectors to bypass the current bad measurement detection approaches. Note, however, that the false data injection attacks are not constrained by these attack scenarios.

—**Scenario I – Limited Access to Meters:** The attacker is restricted to accessing some specific meters due to different physical protections of meters. For example, meters

located in substations with physical perimeter control may be much harder to access than those located in a locked box outside of a building.

—**Scenario II – Limited Resources Available to Compromise Meters:** The attacker is limited in the resources required to compromise meters. For example, the attacker only has resources to compromise up to k meters (out of all the meters). Due to the limited resources, the attacker may also want to minimize the number of meters to be compromised.

In the following, we first show the basic principle of false data injection attacks. We then focus on the two attack scenarios and show how to construct attack vectors for both random and targeted false data injection attacks.

3.1 Basic Principle

Let $\mathbf{z_a}$ represent the vector of observed measurements that may contain malicious data. $\mathbf{z_a}$ can be represented as $\mathbf{z_a} = \mathbf{z} + \mathbf{a}$, where $\mathbf{z} = (z_1, ..., z_m)^T$ is the vector of original measurements and $\mathbf{a} = (a_1, ..., a_m)^T$ is the malicious data added to the original measurements. We refer to $\mathbf{a}$ as an *attack vector*. The i-th element a_i being non-zero means that the attacker compromises the i-th meter, and then replaces its original measurement z_i with a phony measurement $z_i + a_i$.

The attacker can choose any non-zero arbitrary vector as the attack vector $\mathbf{a}$, and then construct the malicious measurements $\mathbf{z_a} = \mathbf{z} + \mathbf{a}$. The traditional bad measurement detection approach computes the 2-Norm of the measurement residual to check whether there exist bad measurements or not. However, as shown in Theorem 1 below, such a detection approach can be bypassed if the attack vector $\mathbf{a}$ is a linear combination of the column vectors of $\mathbf{H}$.

THEOREM 1. *Suppose the original measurements* $\mathbf{z}$ *can pass the bad measurement detection. The malicious measurements* $\mathbf{z_a} = \mathbf{z} + \mathbf{a}$ *can pass the bad measurement detection if* $\mathbf{a}$ *is a linear combination of the column vectors of* $\mathbf{H}$*, i.e.,* $\mathbf{a} = \mathbf{Hc}$*.*

PROOF. $\hat{\mathbf{x}}_{\mathbf{bad}}$, the vector of estimated state variables obtained from $\mathbf{z_a}$, is computed by

$$\begin{aligned}\hat{\mathbf{x}}_{\mathbf{bad}} &= (\mathbf{H^T W H})^{-1}\mathbf{H^T W z_a} = (\mathbf{H^T W H})^{-1}\mathbf{H^T W}(\mathbf{z}+\mathbf{a}) \\ &= \hat{\mathbf{x}} + (\mathbf{H^T W H})^{-1}\mathbf{H^T W a}.\end{aligned}$$

If $\mathbf{a} = \mathbf{Hc}$ (for any $\mathbf{c}$), the 2-Norm of the measurement residual is

$$\begin{aligned}\|\mathbf{z_a} - \mathbf{H}\hat{\mathbf{x}}_{\mathbf{bad}}\| &= \|\mathbf{z} + \mathbf{a} - \mathbf{H}(\hat{\mathbf{x}} + (\mathbf{H^T W H})^{-1}\mathbf{H^T W a})\| \\ &= \|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}} + (\mathbf{a} - \mathbf{H}(\mathbf{H^T W H})^{-1}\mathbf{H^T W a})\| \\ &= \|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}} + (\mathbf{Hc} - \mathbf{H}(\mathbf{H^T W H})^{-1}\mathbf{H^T W Hc})\| \\ &= \|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}} + (\mathbf{Hc} - \mathbf{Hc})\| = \|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\| \le \tau, \qquad (4)\end{aligned}$$

where τ is the detection threshold. Therefore, the 2-Norm of the measurement residual of $\mathbf{z_a}$ is less than the threshold τ, and $\mathbf{z_a}$ can also pass the bad measurement detection. The injected error is $\hat{\mathbf{x}}_{\mathbf{bad}} - \hat{\mathbf{x}} = (\mathbf{H^T W H})^{-1}\mathbf{H^T W a} = \mathbf{c}$ □

In this paper, we refer to an attack in which the attack vector $\mathbf{a}$ equals $\mathbf{Hc}$, where $\mathbf{c}$ is an arbitrary non-zero vector, as a *false data injection attack*. By launching false data injection attacks, the attacker can manipulate the injected false data to bypass the bad measurement

detection and also introduce arbitrary errors into the output of the state estimation (since each element of $\mathbf{c}$ could be an arbitrary number).

3.2 Scenario I – Limited Access to Meters

We assume that the attacker has access to k specific meters. Intuitively, the attacker can only modify the measurements of these k meters. As a result, the attacker cannot simply choose any $\mathbf{c}$ and use $\mathbf{a} = \mathbf{Hc}$ as the attack vector. For those meters that cannot be accessed by the attacker, the injected errors must remain 0.

Formally, we let $\mathcal{I}_{meter} = \{i_1, ..., i_k\}$ be the set of indices of the k meters that the attacker has access to. The attacker can modify measurements z_{i_j}, where $i_j \in \mathcal{I}_{meter}$. To launch a false data injection attack without being detected, the attacker needs to find a non-zero attack vector $\mathbf{a} = (a_1, ..., a_m)^T$ such that $a_i = 0$ for $i \notin \mathcal{I}_{meter}$ (i.e., the attacker cannot change the meters that she cannot access) and $\mathbf{a}$ is a linear combination of the column vectors of $\mathbf{H}$ (i.e., $\mathbf{a} = \mathbf{Hc}$).

3.2.1 *Random False Data Injection Attack.* In a random false data injection attack, the attacker aims to cause wrong estimation of state variables, where the errors injected into the wrong estimation could be any value. Thus, the attack vector $\mathbf{a}$ should satisfy the condition $\mathbf{a} = (a_1, ..., a_m)^T = \mathbf{Hc}$ with $a_i = 0$ for $i \notin \mathcal{I}_{meter}$, where $\mathcal{I}_{meter}$ is the set of indices of the meters that can be accessed by the attacker.

In the following, we develop a method for the attacker to construct such an attack vector. We first show in Theorem 2 that $\mathbf{c}$ is redundant and can be eliminated from our formulation, and $\mathbf{a} = \mathbf{Hc}$ can be transformed into an equivalent but more straightforward form, which only has one variable $\mathbf{a}$. This equivalent form will allow us to easily generate an attack vector $\mathbf{a}$ that satisfies the above condition.

THEOREM 2. $\mathbf{a} = \mathbf{Hc}$ *if and only if* $\mathbf{Ba} = \mathbf{0}$, *where* $\mathbf{B} = \mathbf{H}(\mathbf{H^T H})^{-1}\mathbf{H^T} - \mathbf{I}$.

PROOF. Let $\mathbf{P} = \mathbf{H}(\mathbf{H^T H})^{-1}\mathbf{H^T}$ and $\mathbf{B} = \mathbf{P} - \mathbf{I}$. According to [Brockwell and Davis 1991], for any $\mathbf{a} \in \mathcal{R}^m$, $\mathbf{Pa} = \mathbf{a}$ if and only if $\mathbf{a}$ is a linear combination of column vectors of $\mathbf{H}$ (i.e., $\mathbf{a} = \mathbf{Hc}$). Therefore,

$$\mathbf{a} = \mathbf{Hc} \Leftrightarrow \mathbf{Pa} = \mathbf{a} \Leftrightarrow \mathbf{Pa} - \mathbf{a} = \mathbf{0} \Leftrightarrow (\mathbf{P} - \mathbf{I})\mathbf{a} = \mathbf{0} \Leftrightarrow \mathbf{Ba} = \mathbf{0}. \quad (5)$$

This means $\mathbf{a}$ satisfies $\mathbf{a} = \mathbf{Hc}$ if and only if it satisfies $\mathbf{Ba} = \mathbf{0}$. □

Generating a: The attacker needs to find a non-zero attack vector $\mathbf{a}$ such that $\mathbf{Ba} = \mathbf{0}$ and $a_i = 0$ for $i \notin \mathcal{I}_{meter}$. Represent $\mathbf{a}$ as $\mathbf{a} = (0, ..., 0, a_{i_1}, 0, ..., 0, a_{i_2}, 0, ..., 0, a_{i_k}, 0, ..., 0)^T$, where $a_{i_1}, a_{i_2}, ..., a_{i_k}$ are the unknown variables. Let $\mathbf{B} = (\mathbf{b}_1, ..., \mathbf{b}_m)$, where $\mathbf{b}_i$ ($1 \leq i \leq m$) is the i-th column vector of $\mathbf{B}$. Thus,

$$\mathbf{Ba} = \mathbf{0} \Leftrightarrow (..., \mathbf{b}_{i_1}, ..., \mathbf{b}_{i_2}, ..., \mathbf{b}_{i_k}, ...)(0, ..., 0, a_{i_1}, 0, ..., 0, a_{i_2}, 0, ..., 0, a_{i_k}, 0, ..., 0)^T = \mathbf{0}.$$

Let the $m \times k$ matrix $\mathbf{B}' = (\mathbf{b}_{i_1}, ..., \mathbf{b}_{i_k})$ and the length k vector $\mathbf{a}' = (a_{i_1}, ..., a_{i_k})^T$. We have

$$\mathbf{Ba} = \mathbf{0} \Leftrightarrow \mathbf{B}'\mathbf{a}' = \mathbf{0}.. \quad (6)$$

If the rank of $\mathbf{B}'$ is less than k, $\mathbf{B}'$ is a rank deficient matrix, and there exist infinite number of non-zero solutions $\mathbf{a}'$ that satisfy $\mathbf{B}'\mathbf{a}' = \mathbf{0}$ [Meyer 2001]. According to [Meyer 2001], the solution is $\mathbf{a}' = (\mathbf{I} - \mathbf{B}'^{-}\mathbf{B}')\mathbf{d}$, where $\mathbf{B}'^{-}$ is the Matrix 1-inverse of $\mathbf{B}'$ and $\mathbf{d}$

is an arbitrary non-zero vector of length k. With a non-zero solution $\mathbf{a}'$, the attacker can generate the attack vector $\mathbf{a}$ by filling 0's as the remaining elements in $\mathbf{a}$.

If the rank of $\mathbf{B}'$ is k, then $\mathbf{B}'$ is not a rank deficient matrix and $\mathbf{B}'\mathbf{a}' = \mathbf{0}$ has a unique solution $\mathbf{a}' = \mathbf{0}$ [Meyer 2001]. This means that no error can be injected into the state estimation, and the attack vector does not exist. In other words, the attacker cannot launch the attack. In Section 5, we show that the chance that the attack vector exists increases as k increases. Moreover, we prove in Theorem 3 that the attack vector always exists if $k > m - n$.

THEOREM 3. *Let k be the number of specific meters that can be accessed by the attacker. If $k > m - n$, the attacker can always generate an attack vector that satisfies the condition $\mathbf{a} = (a_1, ..., a_m)^T = \mathbf{Hc}$ with $a_i = 0$ for $i \notin \mathcal{I}_{meter}$, where $\mathcal{I}_{meter}$ is the set of indices of meters that can be accessed by the attacker.*

PROOF. When generating an attack vector, the attacker needs to look at the rank of matrix $\mathbf{B}'$. If rank$(\mathbf{B}') < k$, then the attack vector exists. Otherwise, the attack vector does not exist. Thus, in the following, we prove that if $k > m - n$, rank$(\mathbf{B}')$ is always less than k.

$\mathbf{H}$ is an $m \times n$ full rank matrix and $\mathbf{P} = \mathbf{H}(\mathbf{H^T H})^{-1}\mathbf{H^T}$ is a projection matrix of $\mathbf{H}$. According to [Meyer 2001], rank$(\mathbf{P})$ = rank$(\mathbf{H}) = n$, and rank$(\mathbf{B})$ = rank$(\mathbf{P} - \mathbf{I}) = m - n$. Note that $\mathbf{B}'$ is a submatrix of $\mathbf{B}$. Hence, rank$(\mathbf{B}') \leq$ rank$(\mathbf{B}) < k$.

Therefore, $\mathbf{B}'$ is a rank deficient matrix and there exist infinite number of non-zero solutions for $\mathbf{a}'$ that satisfy $\mathbf{B}'\mathbf{a}' = \mathbf{0}$. With a non-zero solution $\mathbf{a}'$, the attacker can generate the respective attack vector $\mathbf{a}$ by filling 0's as the remaining elements in $\mathbf{a}$. □

When $k > m-n$, the attacker does not need to compute the matrices $\mathbf{B}$ and $\mathbf{B}'$ to obtain the attack vector. Instead, the attacker can perform elementary column operations on $\mathbf{H}$ to generate the attack vector. Appendix A shows the details.

3.2.2 *Targeted False Data Injection Attack.* In a targeted false data injection attack, the attacker not only wants to inject errors into state estimation, but also wants to precisely control the errors injected into the estimation of certain chosen state variables. In some sense, targeted false data injection attacks can be viewed as an advanced form of random attacks.

This attack can be represented mathematically as follows. Let $\mathcal{I}_{variable} = \{i_1, ..., i_r\}$, where $r < n$, denote the set of indexes of the r target state variables chosen by the attacker. (That is, the attacker has chosen $x_{i_1}, x_{i_2}, ..., x_{i_r}$ to compromise.) In this attack, the attacker intends to construct an attack vector $\mathbf{a}$ such that the resulting estimate $\hat{\mathbf{x}}_{\mathbf{bad}} = \hat{\mathbf{x}} + \mathbf{c}$, where $\mathbf{c} = (c_1, c_2, ..., c_n)^T$ and c_i for $i \in \mathcal{I}_{variable}$ is the specific error that the attacker has chosen to inject to $\hat{x}_i$. In other words, the attacker wants to replace $\hat{x}_{i_1}, ...,$ and $\hat{x}_{i_r}$ with $\hat{x}_{i_1} + c_{i_1}, ...,$ and $\hat{x}_{i_r} + c_{i_r}$, respectively.

We consider two cases for the targeted false data injection attack: A *constrained* and an *unconstrained* case. In the constrained case, the attacker wants to launch a targeted false data injection attack that only changes the target state variables but does not pollute the other state variables. The constrained case represents the situation where the control center (software or operator) may know ways to verify the estimates of the other state variables. In the unconstrained case, the attacker has no concerns on the impact on the other state variables when attacking the chosen ones. In the following, we show how an attacker generates an attack vector for the constrained and unconstrained cases, respectively.

Constrained Case: The construction of an attack vector $\mathbf{a}$ becomes rather simple in the constrained case. The error injected into the estimates $\hat{\mathbf{x}}$ of state variables is $\mathbf{c}$. Let $\mathcal{I}_{variable}$ denote the set of indexes of the target state variables chosen by the attacker. Every element c_i in $\mathbf{c}$ is fixed, which is either the chosen value when $i \in \mathcal{I}_{variable}$, or 0 when $i \notin \mathcal{I}_{variable}$. Therefore, the attacker can substitute $\mathbf{c}$ back into $\mathbf{a} = \mathbf{Hc}$, and check if $a_i = 0$ for all $i \notin \mathcal{I}_{meter}$. If yes, the attacker succeeds in constructing the (only) attack vector $\mathbf{a}$. Otherwise, the attack is impossible.

Unconstrained Case: To launch a targeted false data injection attack in the unconstrained case, the attacker need to generate an attack vector $\mathbf{a}$ that satisfies the following three conditions: (1) $\mathbf{a} = \mathbf{Hc}$; (2) $a_i = 0$ for all $i \notin \mathcal{I}_{meter}$; and (3) c_i of $\mathbf{c}$ is the specific value chosen by the attacker, where $i \in \mathcal{I}_{variable}$. To generate such an attack vector, we first show that $\mathbf{a} = \mathbf{Hc}$ can be converted into an equivalent form without having $\mathbf{c}$, and then generate $\mathbf{a}$ based on the equivalent form.

THEOREM 4. *$\mathbf{a} = \mathbf{Hc}$ if and only if $\mathbf{B_s a} = \mathbf{y}$, where $\mathbf{B_s} = \mathbf{H_s}(\mathbf{H_s^T H_s})^{-1}\mathbf{H_s^T} - \mathbf{I}$, $\mathbf{H_s}$ is the submatrix of $\mathbf{H}$ containing columns whose indices are not in $\mathcal{I}_{variable}$, $\mathbf{b} = \sum_{j \in \mathcal{I}_{variable}} \mathbf{h}_j c_j$, and $\mathbf{y} = \mathbf{B_s b}$.*

PROOF. Assume that the number of target variables is r. Let $\mathbf{c_s} = (c_{j_1}, ..., c_{j_{n-r}})^T$, where $j_i \notin \mathcal{I}_{variable}$ for $1 \le i \le n - r$.

$$\begin{aligned}
\mathbf{a} = \mathbf{Hc} \Leftrightarrow\ & \mathbf{a} = \sum_{i \notin \mathcal{I}_{variable}} \mathbf{h}_i c_i + \sum_{j \in \mathcal{I}_{variable}} \mathbf{h}_j c_j = \mathbf{H_s c_s} + \mathbf{b} \Leftrightarrow \mathbf{a} - \mathbf{b} = \mathbf{H_s c_s} \\
\Leftrightarrow\ & \mathbf{H_s}(\mathbf{H_s^T H_s})^{-1}\mathbf{H_s^T}(\mathbf{a} - \mathbf{b}) = \mathbf{H_s}(\mathbf{H_s^T H_s})^{-1}\mathbf{H_s^T H_s c_s} = \mathbf{H_s c_s} = \mathbf{a} - \mathbf{b} \\
\Leftrightarrow\ & \mathbf{H_s}(\mathbf{H_s^T H_s})^{-1}\mathbf{H_s^T} - \mathbf{I})\mathbf{a} = (\mathbf{H_s}(\mathbf{H_s^T H_s})^{-1}\mathbf{H_s^T} - \mathbf{I})\mathbf{b} \\
\Leftrightarrow\ & \mathbf{B_s a} = \mathbf{B_s b} \Leftrightarrow \mathbf{B_s a} = \mathbf{y}. \qquad (7)
\end{aligned}$$

Hence, $\mathbf{a}$ satisfies $\mathbf{a} = \mathbf{Hc}$ if and only if $\mathbf{a}$ satisfies $\mathbf{B_s a} = \mathbf{y}$. □

Generating $\mathbf{a}$*:* The attacker needs to find an attack vector $\mathbf{a}$ such that $\mathbf{B_s a} = \mathbf{y}$ where $a_i = 0$ for $i \notin \mathcal{I}_{meter}$. There are k unknown elements in $\mathbf{a}$ at positions $i_1, ..., i_k$. We follow the same reasoning as in Section 3.2.1 to denote $\mathbf{B'_s} = (\mathbf{b}_{\mathbf{s}_{i_1}}, ..., \mathbf{b}_{\mathbf{s}_{i_k}})$ and $\mathbf{a'} = (a_{i_1}, ..., a_{i_k})^T$. Then we have

$$\mathbf{B'_s a'} = \mathbf{y} \Leftrightarrow \mathbf{B_s a} = \mathbf{y}. \qquad (8)$$

If the rank of $\mathbf{B'}_s$ is the same as that of the augmented matrix $(\mathbf{B'}_s|\mathbf{y})$, $\mathbf{B'_s a'} = \mathbf{y}$ is a consistent equation, and there exist infinite solutions $\mathbf{a'} = \mathbf{B'_s}^{-}\mathbf{y} + (\mathbf{I} - \mathbf{B'_s}^{-}\mathbf{B'_s})\mathbf{d}$ that satisfy $\mathbf{B'_s a'} = \mathbf{y}$, where $\mathbf{B'}_s^{-}$ is the Matrix 1-inverse of $\mathbf{B'}_s$ and $\mathbf{d}$ is an arbitrary non-zero vector of length k [Meyer 2001]. The attacker can generate an attack vector $\mathbf{a}$ from any $\mathbf{a'} \neq \mathbf{0}$.

If the rank of $\mathbf{B'}_s$ is not the same as the rank of the augmented matrix $(\mathbf{B'_s}|\mathbf{y})$, then the relation $\mathbf{B'_s a'} = \mathbf{y}$ is not a consistent equation, and thus has no solution. This means that the attacker cannot generate an attack vector to inject the specific errors into the chosen state variables.

How the attacker chooses specific errors c_j for $j \in \mathcal{I}_{variable}$ affects the feasibility of launching targeted attacks. Note that $\mathbf{y} = \mathbf{B_s b} = \mathbf{B_s} \sum_{j \in \mathcal{I}_{variable}} \mathbf{h}_j c_j$. If the attacker chooses c_j such that $\mathbf{B_s} \sum_{j \in \mathcal{I}_{variable}} \mathbf{h}_j c_j$ is a linear combination of columns of $\mathbf{B'_s}$ or

$\mathbf{B_s} \sum_{j \in \mathcal{I}_{variable}} \mathbf{h}_j c_j = \mathbf{0}$, then the rank of the augmented matrix $(\mathbf{B'_s}|\mathbf{y})$ is the same as that of $\mathbf{B'_s}$ and the attacker can generate an attack vector. Otherwise, the attacker cannot generate an attack vector.

3.3 Scenario II – Limited Resources Available to Compromise Meters

In Scenario II, we assume the attacker has resources to compromise up to k meters. Unlike Scenario I, there is no restriction on what meters can be chosen. For the sake of presentation, we call a length-m vector a *k-sparse vector* if it has at most k non-zero elements. Thus, the attacker needs to find a k-sparse, non-zero attack vector $\mathbf{a}$ that satisfies the relation $\mathbf{a} = \mathbf{H}\mathbf{c}$. As in Scenario I, we consider both random and targeted false data injection attacks in Scenario II.

Note that the existence conditions of the attacks follow the same criteria as in Scenario I. Thus, we focus on investigating how the attacker can construct attack vectors.

3.3.1 *Random False Data Injection Attack.* With the resources to compromise up to k meters, the attacker may use a brute-force approach to construct an attack vector. That is, the attacker may try all possible $\mathbf{a}$'s consisting of k unknown elements and $m - k$ zero elements. For each candidate $\mathbf{a}$, the attacker may check if there exists a non-zero solution of $\mathbf{a}$ such that $\mathbf{B}\mathbf{a} = \mathbf{0}$ using the same method as discussed in Section 3.2.1. If yes, the attacker succeeds in constructing an attack vector. Otherwise, the attacker has to try the next candidate. However, the brute-force approach could be time consuming. In the worst case, the attacker needs to examine $\binom{m}{k}$ candidate attack vectors.

To improve the time efficiency, the attacker may take advantage of the following observation. Since a successful attack vector is a linear combination of the column vectors of $\mathbf{H}$ (i.e., $\mathbf{a} = \mathbf{H}\mathbf{c}$), the attacker can perform column transformations to $\mathbf{H}$ to reduce the number of non-zero elements in the transformed column vectors. As this process continues, more column vectors in the transformed $\mathbf{H}$ will have fewer non-zero elements. The column vectors with no more than k non-zero elements can be used as attack vectors. In particular, when the matrix $\mathbf{H}$ is a sparse matrix (which is usually the case in real power systems), it does not take many column transformations to construct a desirable attack vector.

A Heuristic Approach: We give a heuristic approach to exploit this observation. The attacker can initialize a size-n priority queue with the n column vectors of $\mathbf{H}$. The attacker then repeats the following process: Take the column vector $\mathbf{t}$ with the minimum number of non-zero elements out of the queue. If $\mathbf{t}$ is a k-sparse vector, the algorithm returns and $\mathbf{t}$ can be used as the attack vector. If not, for each column vector $\mathbf{s}$ in the queue, the attacker checks if linearly combining $\mathbf{t}$ and $\mathbf{s}$ can result in a column vector with less zero elements than $\mathbf{t}$. If yes, the attacker appends the resulting vector to the queue. The attacker repeats this process until a k-sparse vector is found or the set is empty. It is easy to see that a k-sparse vector constructed in this way must be a linear combination of some column vectors of $\mathbf{H}$, and can serve as an attack vector.

The heuristic approach could be quite slow for a general $\mathbf{H}$. However, it works pretty efficiently for a sparse matrix $\mathbf{H}$, which is usually the case for real-world power systems. For example, in our simulation, when $k = 4$ in the IEEE 300-bus test system, it takes the heuristic approach about 110ms on a regular PC to find an attack vector.

The heuristic approach does not guarantee the construction of an attack vector even if it exists, nor does it guarantee the construction of an attack vector that has the minimum number of non-zero elements. Nevertheless, it runs pretty quickly when it can construct an

attack vector, and thus could still be a useful tool for the attacker.

Ideally, in order to reduce the attack costs, the attacker would like to compromise as few meters as possible. In other words, the attacker wants to find the optimal attack vector $\mathbf{a}$ with the minimum number of non-zero elements. The attacker may use the brute-force approach discussed at the beginning of Section 3.3.1 with k being 1 initially, and gradually increase k until an attack vector is found. Apparently, such an attack vector gives the optimal solution with the minimum number of compromised meters. There are possibilities to improve such a brute-force approach, for example, using a binary search in identifying the minimum k.

3.3.2 *Targeted False Data Injection Attack.* We follow the notation used in Scenario I to describe the targeted false data injection attack. Let $\mathcal{I}_{variable} = \{i_1, ..., i_r\}$, where $r < n$, denote the set of indexes of the r target state variables chosen by the attacker. In this attack, the attacker intends to construct an attack vector $\mathbf{a}$ to replace $\hat{x}_{i_1}, ...$, and $\hat{x}_{i_r}$ with $\hat{x}_{i_1} + c_{i_1}, ...$, and $\hat{x}_{i_r} + c_{i_r}$, respectively, where $c_{i_1}, ..., c_{i_r}$ are the specific errors to be injected. Similar to Scenario I, we consider both constrained and unconstrained cases.

Constrained Case: As discussed earlier, in the constrained case, the attacker intends to only change the estimation of the chosen target state variables, but does not modify the others. Thus, all elements of $\mathbf{c}$ are fixed. So the attacker can substitute $\mathbf{c}$ into the relation $\mathbf{a} = \mathbf{Hc}$. If the resulting $\mathbf{a}$ is a k-sparse vector, the attacker succeeds in constructing the attack vector. Otherwise, the attacker fails. The attack vector derived in the constrained case is the only possible attack vector; there is no way to further reduce the number of compromised meters.

Unconstrained Case: In the unconstrained case, only the elements c_i of $\mathbf{c}$ for $i \in \mathcal{I}_{variable}$ are fixed; the other c_j for $j \notin \mathcal{I}_{variable}$ can be any values. According to Equation (7), $\mathbf{a} = \mathbf{Hc} \Leftrightarrow \mathbf{B_s a} = \mathbf{y}$. (Note that the derivation of Equation (7) does not assume any specific compromised meters. Thus, Equation (7) also holds in the unconstrained case in Scenario II.)

To construct an attack vector, the attacker needs to find a k-sparse attack vector $\mathbf{a}$ that satisfies the relation $\mathbf{B_s a} = \mathbf{y}$. A closer look at this problem reveals that it is the *Minimum Weight Solution for Linear Equations problem* [Garey and Johnson 1979], which is an NP-Complete problem: Given a matrix $\mathbf{A}$ and a vector $\mathbf{b}$, compute a vector $\mathbf{x}$ satisfying $\mathbf{Ax} = \mathbf{b}$ such that $\mathbf{x}$ has at most k non-zero elements. Several efficient heuristic algorithms have been developed to deal with this problem, for example, the Matching Pursuit algorithm [Natarajan 1995; Pati et al. 1993; Lovisolo et al. 2005], the Basis Pursuit algorithm [Chen 1995; Georgiev and Cichoki 2004], and the Gradient Pursuit algorithm [Blumensath and Davies 2008]. The attacker can use these algorithms to find a near optimal attack vector. In our simulation, we choose the Matching Pursuit algorithm, since it is the most widely used algorithm for computing the sparse signal representations and has exponential rate of convergence [Huggins and Zucker 2007].

The attacker may want to minimize the number of meters to be compromised, i.e., to find an attack vector $\mathbf{a}$ with the minimum number of non-zero elements that satisfies $\mathbf{a} = \mathbf{Hc}$ such that the chosen elements in $\mathbf{c}$ have the specific values. This problem is the MIN RVLS$^{=}$ problem [Amaldi and Kann 1998]: Given a matrix $\mathbf{A}$ and a vector $\mathbf{b}$, compute a vector $\mathbf{x}$ satisfying $\mathbf{Ax} = \mathbf{b}$ such that $\mathbf{x}$ has as few non-zero elements as possible. The Matching Pursuit Algorithm can again be used to find an attack vector, since this problem

is the optimization version of the minimum weight solution for linear equations problem.

3.4 Impact Analysis

In this subsection, we analyze the impact introduced by false data injection attacks. Note that the analysis is not limited to the two scenarios discussed earlier. Instead, it applies to any false data injection attacks that inject errors into state variables through compromised meter measurements.

3.4.1 *Problem Formulation.* Since the state estimation problem is commonly solved as a weighted least squares problem [Monticelli 1999], the vector $\hat{\mathbf{x}}_{\mathbf{bad}}$ of the estimated state variables obtained from observed measurements $\mathbf{z_a}$ can be represented as

$$\hat{\mathbf{x}}_{\mathbf{bad}} = (\mathbf{H^T W H})^{-1}\mathbf{H^T W z_a}, \tag{9}$$

where $\mathbf{W}$ is a diagonal matrix whose elements are reciprocals of the variances of meter errors. (See Section 2.) Therefore,

$$\|\mathbf{z_a} - \mathbf{H}\hat{\mathbf{x}}_{\mathbf{bad}}\| = \|\mathbf{H}\hat{\mathbf{x}}_{\mathbf{bad}} - \mathbf{z_a}\| = \|(\mathbf{H}(\mathbf{H^T W H})^{-1}\mathbf{H^T W} - \mathbf{I})(\mathbf{z}+\mathbf{a})\| = \|\mathbf{F}(\mathbf{z}+\mathbf{a})\|,$$

where $\mathbf{F} = \mathbf{H}(\mathbf{H^T W H})^{-1}\mathbf{H^T W} - \mathbf{I}$. If $\|\mathbf{Fa}\| = \mathbf{0}$, $\|\mathbf{z_a} - \mathbf{H}\hat{\mathbf{x}}_{\mathbf{bad}}\| = \|\mathbf{Fz}\| = \|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\|$. On the other hand, it follows from Equation (9) that

$$\|\hat{\mathbf{x}}_{\mathbf{bad}} - \hat{\mathbf{x}}\| = \|(\mathbf{H^T W H})^{-1}\mathbf{H^T W}(\mathbf{z}+\mathbf{a}) - (\mathbf{H^T W H})^{-1}\mathbf{H^T W z}\| = \|\mathbf{Qa}\|,$$

where $\mathbf{Q} = (\mathbf{H^T W H})^{-1}\mathbf{H^T W}$. Assume $\mathcal{I}_{meter} = \{i_1, ..., i_k\}$ is the set of indices of meters that are compromised by the attacker. Further represent $\mathbf{F}$ as $\mathbf{F} = (\mathbf{f}_1, ..., \mathbf{f}_m)$. Since the attacker can only inject errors into the meters that she compromises (i.e., $a_i = 0$ for $i \notin \mathcal{I}_{meter}$), $\|\mathbf{Qa}\| = \|\mathbf{Q'a'}\|$ and $\|\mathbf{Fa}\| = \mathbf{0} \Leftrightarrow \|\mathbf{F'a'}\| = \mathbf{0}$, where $\mathbf{F'} = (\mathbf{f}_{i_1}, ..., \mathbf{f}_{i_k})$, $\mathbf{Q'} = (\mathbf{q}_{i_1}, ..., \mathbf{q}_{i_k})$, and $\mathbf{a'} = (a_{i_1}, ..., a_{i_k})^T$. Let $\mathcal{I}_{variable} = \{j_1, ..., j_r\}$ denote the set of indexes of target state variables chosen by the attacker. ($\mathcal{I}_{variable}$ consists of the indexes of all state variables when the attacker does not target at any specific state variables.) The error introduced by the attacker can be obtained by solving the following optimization problem:

$$\begin{aligned} &\text{Maximize } \|\mathbf{Q''a'}\| \\ &\text{s.t.} \qquad \|\mathbf{F'a'}\| = 0, \end{aligned}$$

where $\mathbf{Q''}$ is a submatrix of $\mathbf{Q'}$ and is formed by the j_1-th,...,j_k-th row of $\mathbf{Q'}$.

3.4.2 *Injected Error.* Note that when $\mathbf{F'}$ is a full rank matrix (i.e., Rank $(\mathbf{F'}) = k$), $\|\mathbf{F'a'}\| = \mathbf{0}$ has a unique solution $\mathbf{a'} = \mathbf{0}$ [Meyer 2001]. Therefore, $\|\mathbf{Q''a'}\| = 0$ and no error can be injected into the state estimation. However, when $\mathbf{F'}$ is a rank deficient matrix (i.e., Rank $(\mathbf{F'}) < k$), the amount of introduced error is unbounded as shown in Theorem 5.

THEOREM 5. *For false data injection attacks, if Rank $(\mathbf{F'}) < k$, the maximum of the 2-Norm of error an attacker can introduce to the outcome of state estimation is unbounded.*

PROOF. We need to maximize $\|\mathbf{Q''a'}\|$ under the condition that $\|\mathbf{F'a'}\| = 0$. If $\mathbf{F'}$ is a rank deficient matrix, there exist non-zero solutions $\mathbf{a'}$ that satisfy $\mathbf{F'a'} = \mathbf{0}$ and $\mathbf{a'} = (\mathbf{I} - \mathbf{F'^{-}F'})\mathbf{d}$, where $\mathbf{F'^{-}}$ is the Matrix 1-inverse of $\mathbf{F'}$ and $\mathbf{d}$ is an arbitrary non-zero vector of length k. Thus, $\|\mathbf{Q''a'}\| = \|\mathbf{Q''}[(\mathbf{I} - \mathbf{F'^{-}F'})\mathbf{d}]\|$. Note that $\mathbf{d}$ can be any non-zero vector. Therefore, the 2-Norm of injected error $\|\mathbf{Q''a'}\|$ is unbounded. □

4. GENERALIZED FALSE DATA INJECTION ATTACKS

In this section, we extend false data injection attacks to a generalized version, which we referred to as *generalized false data injection attacks*. The primary objective is to see if an attacker can achieve higher impacts by taking advantage of the small measurement errors typically tolerated by state estimation algorithms.

As for false data injection attacks, we consider both *random* and *targeted* generalized false data injection attacks. In random generalized false data injection attacks, the attacker aims to mislead the control center to get wrong estimates of state variables, whereas in targeted generalized false data injection attacks, the attacker aims to make the estimates of selected state variables to be specific values. For both random and targeted false data injection attacks, we show how an attacker constructs an attack vector in Scenarios I and II, respectively.

4.1 Basic Principle

Similar to false data injection attacks, we consider a power system consisting of m meters and n state variables for generalized false data injection attacks. Recall that the compromised measurements $\mathbf{z_a}$ can be represented as $\mathbf{z_a} = \mathbf{z} + \mathbf{a}$, where $\mathbf{z}$ is the vector of original measurements and $\mathbf{a}$ is the attack vector. $\hat{\mathbf{x}}_\mathbf{bad}$, the estimated state variables obtained from $\mathbf{z_a}$, can be represented as $\hat{\mathbf{x}} + \mathbf{c}$, where $\mathbf{c}$ is the introduced error and $\hat{\mathbf{x}}$ is the true estimate. Therefore, the 2-Norm of the measurement residual of $\mathbf{z_a}$ is

$$\begin{aligned} \|\mathbf{z_a} - \mathbf{H}\hat{\mathbf{x}}_\mathbf{bad}\| &= \|\mathbf{z} + \mathbf{a} - \mathbf{H}(\hat{\mathbf{x}} + \mathbf{c})\| \\ &= \|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}} + (\mathbf{a} - \mathbf{H}\mathbf{c})\| \\ &\leq \|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\| + \|(\mathbf{a} - \mathbf{H}\mathbf{c})\|. \end{aligned}$$

Let τ denote the detection threshold and $\tau_a = \tau - \|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\|$. If $\|\mathbf{a} - \mathbf{H}\mathbf{c}\| \leq \tau_a$, then $\|\mathbf{z_a} - \mathbf{H}\hat{\mathbf{x}}_\mathbf{bad}\| \leq \tau$ and the attacker can bypass the detection. We refer to an attack in which the attack vector $\mathbf{a}$ satisfies $\|\mathbf{a} - \mathbf{H}\mathbf{c}\| \leq \tau_a$ as a generalized false data injection attack. That is, in false data injection attacks, the attack vector $\mathbf{a}$ should satisfy the condition $\|\mathbf{a} - \mathbf{H}\mathbf{c}\| = 0$, while the generalized false data injection attacks relax this condition so that any vector $\mathbf{a}$ that satisfies $\|\mathbf{a} - \mathbf{H}\mathbf{c}\| \leq \tau_a$ can be used as the attack vector.

4.2 Scenario I – Limited Access to Meters

Let $\mathcal{I}_{meter} = \{i_1, ..., i_k\}$ represent the set of indices of the k meters whose measurements can be compromised by the attacker. Thus, the attacker can only change the measurement of the i_j-th meter to a wrong value, where $i_j \in \mathcal{I}_{meter}$.

4.2.1 *Random Generalized False Data Injection Attacks.* Assume $\mathbf{a} - \mathbf{H}\mathbf{c} = \mathbf{t}$, where $\mathbf{t}$ is a length-m vector that reflects the difference between $\mathbf{a}$ and $\mathbf{H}\mathbf{c}$. The attacker can bypass detection as long as $\|\mathbf{t}\| = \|\mathbf{a} - \mathbf{H}\mathbf{c}\| \leq \tau_a$. In random generalized false data injection attacks, the vector $\mathbf{c}$ (i.e., the errors introduced to the state variables) can be any value. Note that $\mathbf{a}$ can be represented as $\mathbf{a} = (0, ..., 0, a_{i_1}, 0, ..., 0, a_{i_2}, 0, ..., 0, a_{i_k}, 0, ..., 0)^T$, where $a_{i_1}, a_{i_2}, ..., a_{i_k}$ are the unknown variables to be determined. Following Equations (5) and (6), we can obtain an equivalent form of the relation $\mathbf{a} - \mathbf{t} = \mathbf{H}\mathbf{c}$ as follows:

$$\mathbf{a} - \mathbf{t} = \mathbf{H}\mathbf{c} \Leftrightarrow \mathbf{B}(\mathbf{a} - \mathbf{t}) = \mathbf{0} \Leftrightarrow \mathbf{B}\mathbf{a} = \mathbf{B}\mathbf{t} \Leftrightarrow \mathbf{B}'\mathbf{a}' = \mathbf{B}\mathbf{t}. \tag{10}$$

where $\mathbf{B} = (\mathbf{b}_1, ..., \mathbf{b}_m) = \mathbf{H}(\mathbf{H^T H})^{-1}\mathbf{H^T} - \mathbf{I}$, $\mathbf{B}' = (\mathbf{b}_{i_1}, ..., \mathbf{b}_{i_k})$, $\mathbf{a}' = (a_{i_1}, ..., a_{i_k})^T$, and $\mathbf{t}$ is a vector whose 2-Norm is less than τ_a (i.e., $\|t\| \leq \tau_a$). Thus, the attacker can solve

$\mathbf{a}'$ from equation $\mathbf{B}'\mathbf{a}' = \mathbf{B}\mathbf{t}$ to get the attack vector $\mathbf{a}$. Details are given in Appendix B.

4.2.2 *Targeted Generalized False Data Injection Attacks.* By launching targeted generalized false data injection attacks, the attacker intends to inject specific errors into the estimation of chosen state variables, while resulting in small residuals. We also consider both constrained and unconstrained cases.

In the constrained case, the attacker modifies the target state variables but keeps the other state variables unchanged. Note that the introduced error $\mathbf{c}$ is a fixed vector, and thus the attacker can directly substitute $\mathbf{c}$ into $\mathbf{a} = \mathbf{H}\mathbf{c} + \mathbf{t}$ and adjust $\mathbf{t}$ to obtain the attack vector $\mathbf{a}$. Specifically, the attacker can first use a zero vector as the initial $\mathbf{t} = (t_1, ..., t_m)^T$. Let $\mathbf{f} = (f_1, ..., f_m)^T = \mathbf{H}\mathbf{c}$. For $1 \leq i \leq m$, if $f_i \neq 0$ and $i \notin \mathcal{I}_{meter}$, then the attacker can set t_i to $-f_i$. Finally, the attacker checks whether the 2-Norm of the updated $\mathbf{t}$ is less than τ_a or not. If yes, the attack vector equals to $\mathbf{H}\mathbf{c} + \mathbf{t}$. Otherwise, the attack vector does not exist.

In the unconstrained case, the attacker modifies the target state variables without any concern about the impact on the other state variables. This means only the elements c_i of $\mathbf{c}$ for $i \in \mathcal{I}_{variable}$ are fixed and the other elements c_j for $j \notin \mathcal{I}_{variable}$ can be any values, where $\mathcal{I}_{variable} = \{i_1, ..., i_r\}$ denote the set of indexes of the r target state variables chosen by the attacker. Note that $\mathbf{a} - \mathbf{t} = \mathbf{H}\mathbf{c} = \sum_{i \notin \mathcal{I}_{variable}} \mathbf{h}_i c_i + \sum_{j \in \mathcal{I}_{variable}} \mathbf{h}_j c_j$. Let $\mathbf{b} = \sum_{j \in \mathcal{I}_{variable}} \mathbf{h}_j c_j$ and $\mathbf{H_s} = (\mathbf{h}_{j_1}, ..., \mathbf{h}_{j_{n-r}})$, where $j_i \notin \mathcal{I}_{variable}$ for $1 \leq i \leq n-r$. Following Equations (7) and (8), $\mathbf{a} - \mathbf{t} = \mathbf{H}\mathbf{c}$ can be transformed into the following equivalent forms:

$$\mathbf{a} - \mathbf{t} = \mathbf{H}\mathbf{c} \Leftrightarrow \mathbf{B_s}(\mathbf{a} - \mathbf{t}) = \mathbf{B_s}\mathbf{b} \Leftrightarrow \mathbf{B_s}\mathbf{a} = \mathbf{B_s}(\mathbf{t} + \mathbf{b}) \Leftrightarrow \mathbf{B}'_\mathbf{s}\mathbf{a}' = \mathbf{B_s}(\mathbf{t} + \mathbf{b}), \quad (11)$$

where $\mathbf{B_s} = (\mathbf{b}_{\mathbf{s}_1}, ..., \mathbf{b}_{\mathbf{s}_m}) = \mathbf{H_s}(\mathbf{H_s^T}\mathbf{H_s})^{-1}\mathbf{H_s^T} - \mathbf{I}$, $\mathbf{B}'_\mathbf{s} = (\mathbf{b}_{\mathbf{s}_{i_1}}, ..., \mathbf{b}_{\mathbf{s}_{i_k}})$, $\mathbf{a}' = (a_{i_1}, ..., a_{i_k})^T$, and $\mathbf{t}$ is a vector whose 2-Norm is less than τ_a. Hence, the attacker can solve $\mathbf{a}'$ from equation $\mathbf{B}'_\mathbf{s}\mathbf{a}' = \mathbf{B_s}\mathbf{t} + \mathbf{B_s}\mathbf{b}$ to get the attack vector $\mathbf{a}$. Details are given in Appendix C.

4.3 Scenario II – Limited Resources Available to Compromise Meters

In Scenario II, the attacker can compromise up to k meters, but there is no restriction on what meters can be compromised. The attacker needs to find a k-sparse, non-zero attack vector $\mathbf{a}$ that satisfies the inequality $\|\mathbf{a} - \mathbf{H}\mathbf{c}\| \leq \tau_a$.

For random generalized false data injection attacks and targeted generalized false data injection attacks in unconstrained case, the attacker needs to find a k-sparse vector $\mathbf{a}$ that satisfies equation $\mathbf{B}\mathbf{a} = \mathbf{B}\mathbf{t}$ and $\mathbf{B_s}\mathbf{a} = \mathbf{B_s}(\mathbf{t} + \mathbf{b})$, respectively. The attacker can directly reduce the problem to the Minimum Weight Solution for Linear Equations problem by using any vector with 2-Norm less than or equal to τ_a as $\mathbf{t}$. After the reduction, the attacker can take advantage of existing algorithms such as Matching Pursuit [Natarajan 1995; Pati et al. 1993; Lovisolo et al. 2005], Basis Pursuit [Chen 1995; Georgiev and Cichoki 2004], and Gradient Pursuit [Blumensath and Davies 2008] to find a k-sparse solution for equation $\mathbf{B}\mathbf{a} = \mathbf{B}\mathbf{t}$ or $\mathbf{B_s}\mathbf{a} = \mathbf{B_s}(\mathbf{t} + \mathbf{b})$.

For targeted generalized false data injection attacks in constrained case, the attack vector $\mathbf{a}$ should be a k-sparse vector that satisfies $\mathbf{a} = \mathbf{H}\mathbf{c} + \mathbf{t}$. Note that the introduced error $\mathbf{c}$ is a fixed vector. Thus, if $\mathbf{H}\mathbf{c}$ is k-sparse, then $\mathbf{t} = \mathbf{0}$ and $\mathbf{a} = \mathbf{H}\mathbf{c}$. Otherwise, assume that there are q ($k < q \leq m$) non-zero elements in $\mathbf{H}\mathbf{c}$. The attacker first adjusts $\mathbf{t}$ such that $q - k$ non-zero elements in $\mathbf{H}\mathbf{c}$ can be canceled when $\mathbf{t}$ is added to $\mathbf{H}\mathbf{c}$, and then checks

whether the 2-Norm of $\mathbf{t}$ is less than or equal to τ_a. If yes, $\mathbf{Hc}+\mathbf{t}$ is an attack vector. Otherwise, the attack vector does not exist.

4.4 Impact Analysis

In this subsection, we analyze the impact introduced by generalized false data injection attacks, particularly the additional errors beyond the (original) false data injection attacks. Note that our results are not limited to the above two attack scenarios, but are applicable to any generalized false data injection attacks, where errors are injected into a set of state variables through a set of compromised meters.

4.4.1 *Problem Formulation.* According to Section 3.4.1, $\|\hat{\mathbf{x}}_{\mathbf{bad}}-\hat{\mathbf{x}}\| = \|\mathbf{Qa}\| = \|\mathbf{Q''a}\|$. Note that

$$\|\mathbf{z_a}-\mathbf{H}\hat{\mathbf{x}}_{\mathbf{bad}}\| = \|\mathbf{F}(\mathbf{z}+\mathbf{a})\| \leq \|\mathbf{Fz}\|+\|\mathbf{Fa}\|,$$

where $\mathbf{F}=\mathbf{H}(\mathbf{H^T W H})^{-1}\mathbf{H^T W}-\mathbf{I}$. Let $\tau_a = \tau - \|\mathbf{Fz}\|$. If $\|\mathbf{Fa}\| \leq \tau_a$, then $\|\mathbf{z_a}-\mathbf{H}\hat{\mathbf{x}}_{\mathbf{bad}}\| \leq \tau$. Therefore, the error introduced by the attacker can be obtained by solving the following optimization problem:

$$\begin{aligned} &\text{Maximize } \|\mathbf{Q''a'}\| \\ &\text{s.t.} \qquad \|\mathbf{F'a'}\| \leq \tau_a. \end{aligned}$$

4.4.2 *Injected Error.* When Rank $(\mathbf{F'}) < k$, the amount of error injected by false data injection attacks is unbounded. Note that generalized false data injection attacks include false data injection attacks, and thus the amount of error introduced by generalized attacks is also unbounded.

When Rank $(\mathbf{F'}) = k$, the original false data injection attacks cannot introduce errors to the output of state estimation, as discussed in Section 3.4. However, as shown in Theorem 6, generalized false data injection attacks can still inject non-zero errors, and the 2-Norm of the injected errors is bounded by a constant.

THEOREM 6. *Suppose Rank* $(\mathbf{F'}) = k$. *In generalized false data injection attacks, the maximum of the 2-Norm of injected error is* $\tau_a\sqrt{\lambda_{max}}$, *where* λ_{max} *is the largest eigenvalue of matrix* $\mathbf{D}=[\sqrt{(\mathbf{F'^T F'})^T}]^{-1}(\mathbf{Q''^T Q''})(\sqrt{\mathbf{F'^T F'}})^{-1}$.

PROOF. In generalized false data injection attacks,

$$\|\mathbf{Q''a'}\|^2 = \mathbf{a'^T}(\mathbf{Q''^T Q''})\mathbf{a'} \tag{12}$$

and

$$\|\mathbf{F'a'}\|^2 = \mathbf{a'^T}(\mathbf{F'^T F'})\mathbf{a'}. \tag{13}$$

$\mathbf{F'^T F'}$ is a non-singular $k \times k$ matrix, since $\mathbf{F'}$ is full rank. Let $\mathbf{w}=\sqrt{\mathbf{F'^T F'}}\mathbf{a'}$. Thus, $\mathbf{a'}=(\sqrt{\mathbf{F'^T F'}})^{-1}\mathbf{w}$. Substituting $\mathbf{w}=\sqrt{\mathbf{F'^T F'}}\mathbf{a'}$ and $\mathbf{a'}=(\sqrt{\mathbf{F'^T F'}})^{-1}\mathbf{w}$ into equations (12) and (13), we can obtain

$$\|\mathbf{Q''a'}\|^2 = \mathbf{w^T}[\sqrt{(\mathbf{F'^T F'})^T}]^{-1}(\mathbf{Q''^T Q''})(\sqrt{\mathbf{F'^T F'}})^{-1}\mathbf{w},$$

and

$$\|\mathbf{F'a'}\|^2 = \mathbf{w^T}[\sqrt{(\mathbf{F'^T F'})^T}]^{-1}(\sqrt{\mathbf{F'^T F'}})\mathbf{w} = \mathbf{w^T w}.$$

Let $\mathbf{D} = [\sqrt{(\mathbf{F'^T F'})^T}]^{-1}(\mathbf{Q''^T Q''})(\sqrt{\mathbf{F'^T F'}})^{-1}$ and $\mathbf{y} = \frac{\mathbf{w}}{\tau_a}$. Thus,

$$\|\mathbf{Q''a'}\|^2 = \mathbf{w^T D w} = \tau_a^2(\mathbf{y^T D y}) \tag{14}$$

and

$$\|\mathbf{F'a'}\|^2 = \mathbf{w^T w} = \tau_a^2(\mathbf{y^T y}).$$

Therefore, the maximum amount of injected error can be obtained by solving the following optimization problem:

$$\begin{aligned} &\text{Maximize} \quad \mathbf{y^T D y} \\ &\text{s.t.} \qquad\quad \mathbf{y^T y} \leq 1. \end{aligned}$$

Note that $\mathbf{y} \neq \mathbf{0}$ and matrix $\mathbf{D}$ is a symmetric matrix (i.e., $\mathbf{D^T} = \mathbf{D}$). Let λ_{max} and $\nu_{\mathbf{max}}$ denote the largest eigenvalue of $\mathbf{D}$ and the eigenvector associated with the largest eigenvalue, respectively. According to the Rayleigh-Ritz Theorem [Golub and Van Loan 1989], $\mathbf{y^T D y} \leq \lambda_{max}\mathbf{y^T y}$, and $\mathbf{y^T D y} = \lambda_{max}\mathbf{y^T y}$ when $\mathbf{y} = \nu_{\mathbf{max}}$. The eigenvectors of matrix $\mathbf{D}$ are orthogonal to each other, since $\mathbf{D}$ is a symmetric matrix [Golub and Van Loan 1989]. Hence, $\nu_{\mathbf{max}}^{\mathbf{T}}\nu_{\mathbf{max}} = 1$ and the maximum value of $\mathbf{y^T D y}$ equals λ_{max}. Substituting λ_{max} into Equation (14), we can obtain the maximum error

$$\max \|\mathbf{Q''a'}\| = \tau_a\sqrt{\lambda_{max}} \tag{15}$$

with $\mathbf{a'} = (\sqrt{\mathbf{F'^T F'}})^{-1}\mathbf{w} = (\sqrt{\mathbf{F'^T F'}})^{-1}\tau_{\mathbf{a}}\mathbf{y} = (\sqrt{\mathbf{F'^T F'}})^{-1}\tau_{\mathbf{a}}\nu_{\mathbf{max}}$. □

We have presented false data injection attacks and the generalized versions in this and the previous sections. In the following, we summarize in Table I the main results, particularly the attack existence conditions, to facilitate the understanding of the overall situation.

5. EXPERIMENTAL RESULTS

In this section, we validate both original and generalized false data injection attacks through experiments using IEEE test systems, including the IEEE 9-bus, 14-bus, 30-bus, 118-bus, and 300-bus systems. The IEEE 9-bus, 14-bus, 30-bus, and 118-bus represent portions of American Electric Power System (in the Midwestern US) in the early 1960's, while the IEEE 300-bus system was developed by the IEEE Test Systems Task Force in 1993 [Christie 1999].

In our experiments, we simulate attacks against state estimation using the DC power flow model. We extract the configuration of the IEEE test systems (particularly matrix $\mathbf{H}$) from MATPOWER, a MATLAB package for solving power flow problems [Zimmerman and Murillo-Sánchez 2007][2]. We perform our experiments based on matrix $\mathbf{H}$ and meter measurements obtained from MATPOWER. For each test system, the state variables are voltage angles of all buses, and the meter measurements are real power injections of all buses and real power flows of all branches.

5.1 Objectives of Experimental Evaluation

For false data injection attacks in Scenario I, we have shown that the attacker cannot always generate valid attack vectors to inject random (or specific) errors into estimates of all

[2]In MATPOWER, the shift injection vector is set to $\mathbf{0}$ for state estimation to use the DC power flow model.

Table I. Summary of original and generalized false data injection attacks

Attacks	Scenarios	Goals	Attack existence condition
Basic false data injection attacks	Limited access to meters	Random	$\mathbf{B'a'=0}$, $\mathbf{B'}$ is rank deficient
		Targeted constrained	$\mathbf{a=Hc}$, $\mathbf{c}$ is fixed
		Targeted unconstrained	$\mathbf{B}_s\mathbf{'a'=y}$, $\text{rank}(\mathbf{B}_s\mathbf{'})=\text{rank}(\mathbf{B}_s\mathbf{'|y})$
	Limited resources	Random	$\mathbf{Ba=0}$, $\mathbf{a}$ is k-sparse
		Targeted constrained	$\mathbf{a=Hc}$, $\mathbf{c}$ is fixed
		Targeted unconstrained	$\mathbf{B}_s\mathbf{a=y}$, $\mathbf{a}$ is k-sparse
Generalized false data injection attacks	Limited access to meters	Random	$\mathbf{B'a'=Bt}$, $\|\mathbf{t}\| \le \tau_a$
		Targeted constrained	$\mathbf{a=Hc+t}$, $\mathbf{c}$ is fixed and $\|\mathbf{t}\| \le \tau_a$
		Targeted unconstrained	$\mathbf{B}_s\mathbf{'a'=B}_s\mathbf{(t+b)}$, $\|\mathbf{t}\| \le \tau_a$
	Limited resources	Random	$\mathbf{Ba=Bt}$, $\|\mathbf{t}\| \le \tau_a$, and $\mathbf{a}$ is k-sparse
		Targeted constrained	$\mathbf{a=Hc+t}$, $\mathbf{c}$ is fixed and $\|\mathbf{t}\| \le \tau_a$
		Targeted unconstrained	$\mathbf{B}_s\mathbf{a=B}_s\mathbf{(t+b)}$, $\|\mathbf{t}\| \le \tau_a$ and $\mathbf{a}$ is k-sparse

state variables (or target state variables). Therefore, in our experiments, we are primarily interested in the possibility of generating valid attack vectors, and show how likely the attacker can find valid attack vectors to attack the IEEE test systems.

For false data injection attacks in Scenario II, we pointed out that generating attack vectors for Scenario II is an NP-complete problem. Although it seems difficult for the attacker to find an optimal attack vector in Scenario II due to the NP hardness, we would like to check experimentally if the attacker can take advantage of existing tools to find a near-optimal attack vector within a practical time window. We also want to see the minimum effort the attacker needs to spend compromising meters in order to launch false data injection attacks.

For generalized false data injection attacks, when $\mathbf{F}'$ is a rank deficient matrix, both false data injection attacks and their generalized versions can achieve similar impacts. However, when $\mathbf{F}'$ is a full rank matrix, an attacker cannot launch false data injection attacks but can launch generalized attacks (Theorem 6). Therefore, in our experiments for generalized attacks, we focus on the latter situation (i.e., $\mathbf{F}'$ is a full rank matrix). We would like to investigate how much the attacker can affect the output of state estimation even if the

attacker fails to launch false data injection attacks.

5.2 False Data Injection Attacks: Scenario I

As mentioned earlier, in Scenario I, the attacker is limited to accessing k specific meters. In other words, the attacker can only modify the measurements of these k meters. Our evaluation objective in this scenario is mainly two-fold. First, we would like to see how likely the attacker can use these k meters to achieve her attack goal. Second, we want to see the computational effort required for finding an attack vector. In our evaluation, we consider (1) random false data injection attacks, (2) targeted false data injection attacks in the unconstrained case, and (3) targeted false data injection attacks in the constrained case.

Based on our evaluation objective, we use two evaluation metrics: the *probability* that the attacker can successfully construct an attack vector given the k specific meters, and the *execution time* required to either construct an attack vector or conclude that the attack is infeasible.

We perform the experiments as follows. For random false data injection attacks, we let the parameter k range from 1 to the maximum number of meters in each test system. (For example, k ranges from 1 to 490 in the IEEE 118-bus system.) For each k, we randomly choose k specific meters to attempt an attack vector construction. We repeat this process 100 times for both IEEE 118-bus and 300-bus systems and 1,000 times for the other systems[3], and estimate the *success probability* p_k as $p_k = \frac{\text{\# successful trials}}{\text{\# trials}}$.

Let R_k denote the percentage of the specific meters under the attacker's control (i.e., $\frac{k}{\text{\# meters}}$). Figure 2 shows the relationship between p_k and R_k for random false data injection attacks. We can see that p_k increases sharply as R_k becomes larger than a certain value in all systems. For example, p_k of the IEEE 300-bus system increases quickly when R_k exceeds 20%. Moreover, the attacker can generate the attack vector with the probability close to 1 when R_k is large enough. For example, p_k is almost 1 when R_k passes 60% and 40% in the IEEE 118-bus and 300-bus systems, respectively. Finally, larger systems have higher p_k than smaller systems for the same R_k. For example, p_k is about 0.6 for IEEE 300-bus system and 0.1 for IEEE 118-bus system when the attacker can compromise 30% of the meters in both systems.

For targeted false data injection attacks in the unconstrained case, we also let the parameter k range from 1 to the maximum number of meters in each test system, and perform the following experiments for each k. We randomly pick 10 target state variables for each test system (8 for the IEEE 9-bus system, since it only has 8 state variables). For each target state variable, we use twice its real estimate as the injected error and perform multiple trials (1,000 trials for the IEEE 9-bus, 14-bus, and 30-bus systems, 100 trials for the IEEE 118-bus system, and 20 trials for the IEEE 300-bus system)[4]. In each trial, we randomly choose k meters and test if an attack vector that injects false data into this target variable can be generated. If yes, we mark the experiment as successful. After these trials, we can compute the success probability $p_{k,v}$ for this particular state variable v as $p_{k,v} = \frac{\text{\# successful trials}}{\text{\# trials}}$. Finally, we compute the overall success probability p_k as the average of $p_{k,v}$'s for all the

[3]It takes significantly more time to exhaustively examine the IEEE 118-bus and 300-bus systems with all possible k's. We reduce the number of trials for them so that the simulation can finish within a reasonable amount time.

[4]In this case, it take even more time than random false data injection attacks to exhaustively examine the IEEE 118-bus and 300-bus systems with all possible k's. Thus, we reduce the number of trials for these two systems so that the simulation can finish within a reasonable amount time.

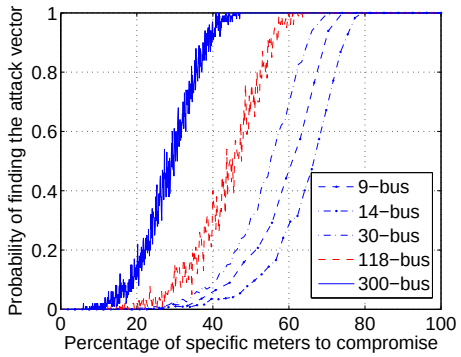

Fig. 2. Probability of finding an attack vector for random false data injection attacks

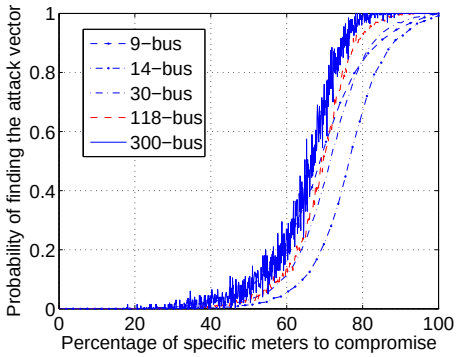

Fig. 3. Probability of finding an attack vector for targeted false data injection attacks (unconstrained case)

chosen state variables.

Figure 3 shows the relationship between p_k and R_k for targeted false data injection attacks in the unconstrained case. We observe the same trend in this figure as in Figure 2, though the probability in this case is in general lower than that in Figure 2. For example, p_k increases sharply as R_k passes 60% for both the IEEE 118-bus and 300-bus systems. Moreover, for both systems, the probability that the attacker can successfully generate the attack vector is larger than 0.6 when R_k passes 70%. For targeted false data injection attacks, larger systems also tend to have higher p_k than smaller systems for the same R_k.

It is critical to note that Figures 2 and 3 represent the success probabilities of "blind trials". In this case, an attacker needs to compromise 30–70% of the meters to get a reasonable probability to construct an attack vector. However, as shown later in Section 5.3.1, when an attacker targets the "weakest link" of a power system, she only needs to compromise a few meters in these test systems.

The targeted false data injection attack in the constrained case is the most challenging one for the attacker. Due to the constraints on the specific meters, the targeted state variables, and the necessity of no impact on the remaining state variables, the probability of successfully constructing an attack vector is in fact very small, though non-zero. We perform experiments for this case slightly differently. We randomly pick 6 sets of meters for the IEEE 118-bus and 300-bus systems. In each set, there are 350 meters and 700 meters for the IEEE 118-bus and 300-bus systems, respectively. We then check the number of individual target state variables that can be affected by each set of meters without affecting the estimation of the remaining state variables. The results show that the attacker can affect 8–11 and 13–16 individual state variables in the IEEE 118-bus and 300-bus systems, respectively. Thus, though the targeted false data injection attack in the constrained case is hard, it is still possible to modify some target state variables.

In Scenario I, all attacks can be performed fairly quickly. When the attack is feasible, it takes little time to actually construct an attack vector. Table II shows the execution times required by the random and the targeted false data injection attacks in the unconstrained case. The time required for the targeted false data injection attack in the constrained case is even less, since the computation is just the multiplication of a matrix and a column vector.

Table II. Timing results in Scenario I (ms)

Test system	Random attack	Targeted attack (unconstrained)
IEEE 9-bus	0.17–2.4	0.21–2.2
IEEE 14-bus	0.16–5.6	0.26–11.3
IEEE 30-bus	0.35–14.9	0.24–31.4
IEEE 118-bus	0.34–867.9	0.42–1,874.5
IEEE 300-bus	0.55–8,549.6	0.73–18,510

For example, the time required for the IEEE 300-bus system ranges from 1.2ms to 11ms.

5.3 False Data Injection Attacks: Scenario II

In Scenario II, the attacker has resources to compromise up to k meters. Compared with Scenario I, the restriction on the attacker is relaxed in the sense that any k meters can be used for the attack. Similar to Scenario I, we would also like to see how likely the attacker can use the limited resources to achieve her attack goal, and at the same time, examine the amount of computation required for attacks. We use two evaluation metrics in our experiments: (1) number of meters to compromise in order to construct an attack vector, and (2) execution time required for constructing an attack vector.

Due to the flexibility for the attacker to choose different meters to compromise in Scenario II, the evaluation of Scenario II generally requires more experiments to obtain the evaluation results. In the following, we examine (1) random false data injection attacks, (2) targeted false data injection attacks in the constrained case, and (3) targeted false data injection attacks in the unconstrained case, respectively.

5.3.1 *Results of Random False Data Injection Attacks.* Random false data injection attacks are the easiest among the three types of attacks under evaluation, mainly due to the least constraints that the attacker has to follow. We perform a set of experiments to construct attack vectors for random false data injection attacks in the IEEE test systems. We assume the attacker wants to minimize the attack cost by compromising as few meters as possible. This means the attacker needs to find the attack vector having the minimum number of non-zero elements.

The brute-force approach is too expensive to use for finding such an attack vector due to its high time complexity. Thus, in our experiments, we use the heuristic algorithm discussed in Section 3.3.1 to find an attack vector that has near minimum number of non-zero elements for the IEEE test systems.

Table III. Random false data injection attacks

Test system	# meters to compromise	Execution time (ms)
IEEE 9-bus	4	0.88
IEEE 14-bus	4	3.47
IEEE 30-bus	4	4.31
IEEE 118-bus	4	19.58
IEEE 300-bus	4	110.51

Table III shows the results. In all test systems, the number of meters that need to be compromised is surprisingly small. For all test systems, the attacker can construct an attack vector for random false data injection attacks by only compromising *4 meters*, with

execution time ranging from 0.88ms to about 110ms. We looked into the experimental data, and found that this is mainly due to the fact that the $\mathbf{H}$ matrices of all these IEEE test systems are sparse. For example, the $\mathbf{H}$ matrix of the IEEE 300-bus system is a $1{,}122 \times 300$ matrix, but most of the entries are 0's. In particular, the sparsest column in $\mathbf{H}$ only has 4 non-zero elements. This column is selected by the algorithm as the attack vector. Note that power systems with sparse $\mathbf{H}$ matrices are not rare cases. In practice, components in a power system that are not physically adjacent to each other are usually not connected.

5.3.2 *Results of Targeted False Data Injection Attacks in Constrained Case.* Similar to Scenario I, targeted false data injection attacks in the constrained case are the most challenging one among all types of attacks due to all the constraints the attacker has to follow in attack vector construction. In the constrained case, the attacker aims to change specific state variables to specific values and keep the remaining state variables as they are.

In our experiments, we randomly choose l ($1 \leq l \leq 10$) target state variables and generate the specific errors for each of them. The specific error is set to be twice as much as the real estimates of the state variables. We then examine how many meters need to be compromised in order to inject the specific errors (without changing the other non-target state variables) into target state variables. For each value of l, we perform the above experiment 1,000 times to examine the distribution of the number of meters that need to be compromised.

Figure 4 shows the results of the IEEE 300-bus system. We use a box plot[5] to show the relationship between the number of target state variables and the number of meters to compromise. In the worst case, to inject specific errors into as many as 10 state variables, the attacker needs to compromise 55–140 meters in the IEEE 300-bus system. Given 1,122 meters in the IEEE 300-bus system, the attacker only needs to compromise a small fraction of the meters to launch targeted false data injection attacks even in the constrained case.

We also exhaustively examine a special situation of targeted false data injection attacks in the constrained case. Specifically, for each state variable, we examine the number of meters that need to be compromised if the attacker aims at this variable. Figure 5 shows the results. We can see that the attacker can inject specific errors into any single state variable using less than 35 meters for the IEEE 118-bus system and less than 40 meters for the IEEE 300-bus system. For all systems, the median values of the number of compromised meters is around 10.

In the constrained case, since $\mathbf{c}$ is fixed, the attack vectors can be directly computed. Thus, the execution time in all the experiments is very short. For example, it costs only 0.45ms on the test computer to generate an attack vector that injects false data into 10 state variables in the IEEE 300-bus system.

5.3.3 *Results of Targeted False Data Injection Attacks in Unconstrained Case.* In the unconstrained case, the attacker wants to inject specific errors into specific state variables, but the attacker does not have to keep the other state variables unchanged. As discussed in Section 3.3.2, we use the Matching Pursuit algorithm [Natarajan 1995; Pati et al. 1993; Lovisolo et al. 2005] to find attack vectors. We perform the same set of experiments as in Section 5.3.2 to obtain the two evaluation metrics: the number of meters to compromise and the execution time. Note that in the unconstrained case, it takes significantly more time

[5]In these box plots, each box shows the first, the second and the third quartiles. The whiskers that extend from the box cover the minimum and maximum points.

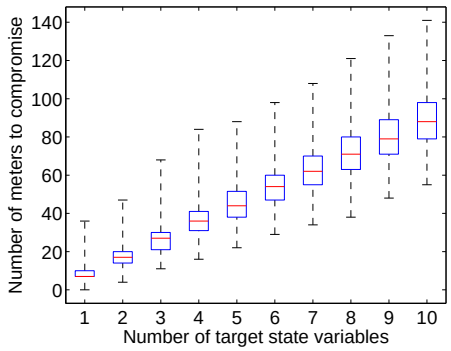

Fig. 4. Constrained case: Number of meters to compromise to inject false data into l target state variables in IEEE 300-bus system

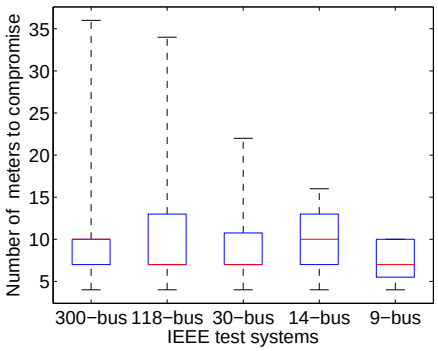

Fig. 5. Constrained case: Number of meters to compromise to inject false data into one target state variable

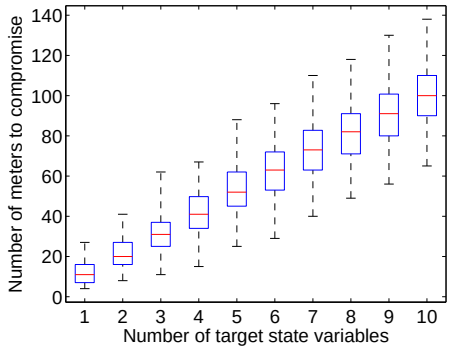

Fig. 6. Unconstrained case: Number of meters to compromise to inject false data into l target state variables in IEEE 300-bus system

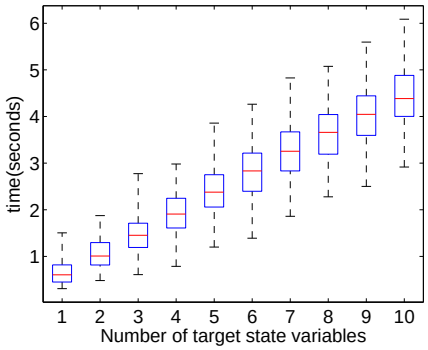

Fig. 7. Unconstrained case: Execution time of finding an attack vector to inject false data into l target state variables in IEEE 300-bus system

to construct an attack vector than the previous experiments. Thus, we show more detailed results on execution time in this case.

Figure 6 shows the relationship between the number of meters to compromise and the number of specific state variables to compromise for the IEEE 300-bus system. Figure 7 shows the corresponding execution time of the Matching Pursuit algorithm for finding an attack vector successfully. We can see that the attacker needs to compromise 60–140 meters for the IEEE 300-bus system, if the attacker wants to inject specific error into as many as 10 state variables. These meters can be quickly identified within 6 seconds.

We also exhaustively examine the special situation of injecting a specific error into a single state variable for all the IEEE test systems, as in the constrained case. Figures 8 and 9 show the number of meters to compromise for these systems and the corresponding execution time, respectively. As shown in Figures 8 and 9, for example, the attacker can inject a specific error into any single state variable of the IEEE 300-bus system by compromising

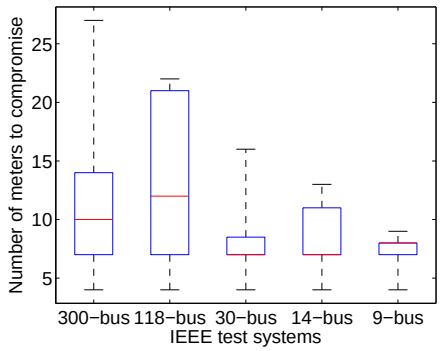

Fig. 8. Unconstrained case: Number of meters to compromise to inject false data into one target state variable

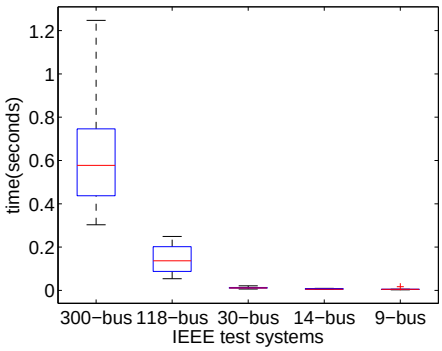

Fig. 9. Unconstrained Case: Execution time of finding an attack vector to inject false data into one target state variable

at most 27 meters, and it costs less than 1.4 seconds to find the attack vector.

These experimental results indicate that the false data injection attacks are practical and easy to launch if the attacker has the configuration information of the target system and can modify the meter measurements.

5.4 Generalized False Data Injection Attacks

For generalized false data injection attacks, we would like to see what an attacker may achieve beyond false data injection attacks. Hence, we focus on the case when an attacker fails in launching false data injection attacks but is still able to launch the generalized version of attacks.

5.4.1 *Experiment Setup.* For false data injection attacks, the attack probability p_k is almost 0 if the attacker cannot compromise more than 10% of the meters. Therefore, to examine the extra impact of generalized false data injection attacks, we require that the number of compromised meters is not larger than $0.1 \times m$ in all the following experiments, where m is the total number of meters in the system. Also, we generate the diagonal matrix $\mathbf{W}$ by using random numbers that range from 250 to 1,000 as diagonal elements[6]. To be consistent with our analysis, we do not limit our attention to any specific attack scenarios and goals, but look at general situations where errors are injected to a set of state variables through a set of compromised meters. In all our experiments, we set τ to 100 and let τ_a range from 0.1τ to τ to see how the change of τ_a affects the impact of the generalized attacks. Note that τ is a parameter chosen by system operators based on the noise in their system. Thus, different systems may have different τ values. Herein, we use a fixed τ for the purpose of illustration. We also use the 2-Norm of injected errors as the evaluation metric.

5.4.2 *Impact on All State Variables.* We first evaluate the impact of generalized false data injection attacks on all state variables. We randomly choose $r_f \times m$ meters and

[6]MATPOWER does not provide $\mathbf{W}$ of the test systems. Hence, we use random numbers close to diagonal elements of $\mathbf{W}$ in example 3.7 of [Monticelli 1999] as diagonal elements of our $\mathbf{W}$.

assume they are compromised by an attacker, where r_f is a parameter between 0.01 and 0.1, denoting the fraction of compromised meters. We then calculate the maximum error that can be injected into the outcome of state estimation using Equation (15). For each r_f, We repeat the above trial 1,000 times and record the average of the results.

Figure 10 shows the average maximum injected error for different r_f and systems when $\tau_a = 0.1\tau$. We can see that the maximum injected error increases as the system becomes large. In particular, the injected error is less than 1 for the IEEE 9-bus system but exceeds 10 for the IEEE 300-bus system. Larger systems have higher injected errors, and thus they are more vulnerable to generalized attacks than smaller systems. Figure 11 shows the maximum injected errors on all state variables as τ_a changes from 0.1τ (i.e., 10) to τ (i.e., 100) for the IEEE 300-bus system. Larger τ_a and higher r_f can result in higher injected error.

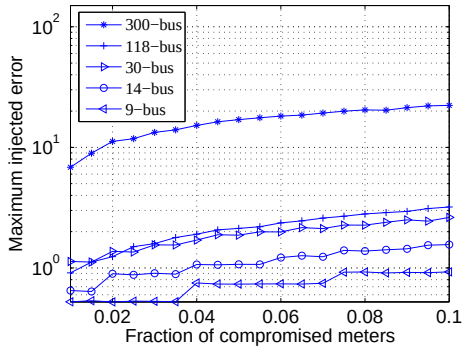

Fig. 10. Maximum impact of generalized false data injection attacks on all state variables

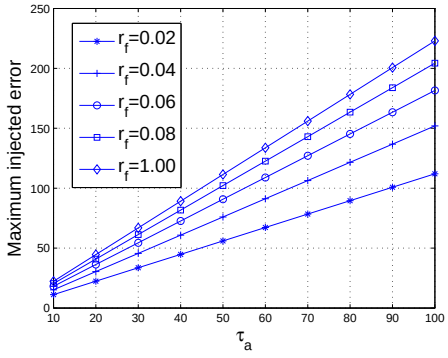

Fig. 11. Maximum impact of generalized false data injection attacks on all state variables as τ_a changes for 300-bus system

According to Theorem 6, the maximum injected error is $\tau_a\sqrt{\lambda_{max}}$, where λ_{max} is the largest eigenvalue of matrix $\mathbf{D} = [\sqrt{(\mathbf{F'^T F'})^T}]^{-1}(\mathbf{Q''^T Q''})(\sqrt{\mathbf{F'^T F'}})^{-1}$. Note that $\mathbf{D}$

has three factor matrices (i.e., $[\sqrt{(\mathbf{F'^T F'})^{\mathbf{T}}}]^{-1}$, $\mathbf{Q''^T Q''}$, and $(\sqrt{\mathbf{F'^T F'}})^{-1}$). We would like to find out which factor matrix is more sensitive to the system change. We randomly select $0.1 \times m$ meters and calculate the maximum eigenvalues of each factor matrix. We repeat this process 1,000 times and record the average of the results as shown in Table IV. Note that the eigenvalue of $[\sqrt{(\mathbf{F'^T F'})^{\mathbf{T}}}]^{-1}$ is equal to that of $(\sqrt{\mathbf{F'^T F'}})^{-1}$, since $[\sqrt{(\mathbf{F'^T F'})^{\mathbf{T}}}]^{-1} = (\sqrt{\mathbf{F'^T F'}})^{-1}$.

Table IV shows that the maximum eigenvalue of $\mathbf{Q''^T Q''}$ increases more quickly than that of $[\sqrt{(\mathbf{F'^T F'})^{\mathbf{T}}}]^{-1}/(\sqrt{\mathbf{F'^T F'}})^{-1}$ as the system becomes large. Such increase is quite significant for the IEEE 300-bus system. In particular, the maximum eigenvalue of $\mathbf{Q''^T Q''}$ is about 450 times more than that in the IEEE 9-bus system, whereas the maximum eigenvalue of $[\sqrt{(\mathbf{F'^T F'})^{\mathbf{T}}}]^{-1}/(\sqrt{\mathbf{F'^T F'}})^{-1}$ is less than twice as much as that in the IEEE 9-bus system. This observation reveals that the factor matrix $\mathbf{Q''^T Q''}$ is more sensitive to the system change than the other factor matrices. Note that the dramatic increase of the maximum eigenvalue of $\mathbf{Q''^T Q''}$ for the IEEE 300-bus system is visually consistent with the sharp raise of the maximum errors injected into the system as shown in Figure 10.

Table IV. Maximum eigenvalues of the three matrices that form $\mathbf{D}$

Test system	$[\sqrt{(\mathbf{F'^T F'})^{\mathbf{T}}}]^{-1}/(\sqrt{\mathbf{F'^T F'}})^{-1}$	$\mathbf{Q''^T Q''}$
IEEE 9-bus	1.3870	0.0055
IEEE 14-bus	1.5170	0.0171
IEEE 30-bus	1.6903	0.0487
IEEE 118-bus	2.0808	0.0713
IEEE 300-bus	2.4353	2.7643

5.4.3 *Impact on Individual State Variables.* We further look at the impact of generalized false data injection attacks on individual state variables. For each state variable, we randomly choose k meters and assume that those meters are compromised by an attacker, where k is set to $0.01 \times m$, $0.05 \times m$, and $0.1 \times m$ in our experiments. We then calculate the maximum injected error based on Equation (15). We repeat this process 1,000 times and use the average of the results as the maximum impact of generalized attacks on that state variable.

Figures 12 and 13 show the empirical cumulative distribution function (CDF) curves of maximum injected errors when $\tau_a = 0.1\tau$. A point (x, y) on the curve indicates that $y\%$ state variables have the maximum injected error less than or equal to x. For all state variables in the IEEE 118-bus system and most of the state variables in the IEEE 300-bus system, the maximum injected error is quite small (e.g., when $k = 0.1m$, the error injected into any state variable of the IEEE 118-bus system is less than or equal to 0.7 and about 90% state variables of the IEEE 300-bus system are injected with errors that are less than or equal to 2). However, some state variables of the IEEE 300-bus system still have large injected errors, which can be as high as 8.5. Figures 14 and 15 show the empirical CDF curves of maximum injected error for different values of τ_a when k equals to $0.1 \times m$. Again, larger τ_a achieves higher injected error.

As revealed in Table IV and Figure 10, the maximum injected error is related to the maximum eigenvalue of $\mathbf{Q''^T Q''}$. Hence, we perform an experiment to examine the maximum eigenvalue of $\mathbf{Q''^T Q''}$. We randomly choose k meters, where k is set to $0.01 \times m$, $0.05 \times m$, and $0.1 \times m$ in our experiment. For each state variable, we generate the corresponding $\mathbf{Q''^T Q''}$ using the method discussed in Section 4.4.1, and calculate the maximum

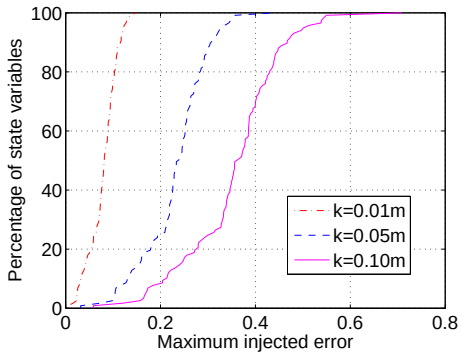

Fig. 12. Empirical CDF curves of maximum injected errors for IEEE 118-bus system

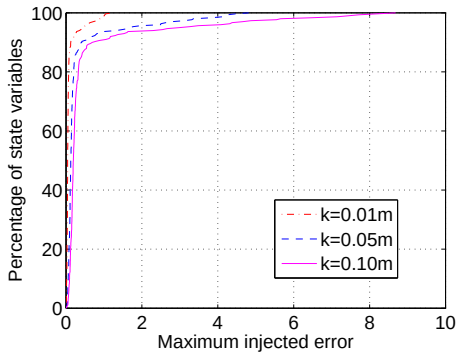

Fig. 13. Empirical CDF curves of maximum injected errors for IEEE 300-bus system

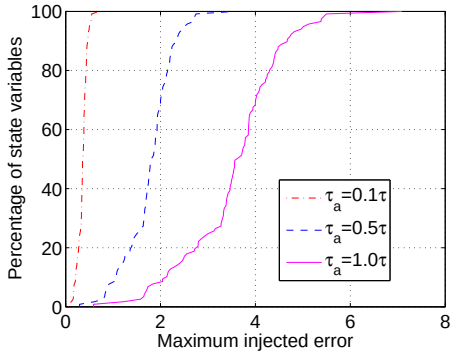

Fig. 14. Empirical CDF curves of maximum injected errors as τ_a changes for IEEE 118-bus system

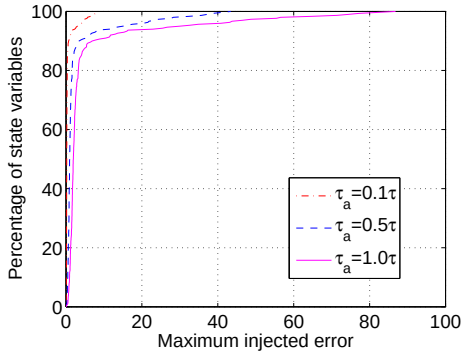

Fig. 15. Empirical CDF curves of maximum injected errors as τ_a changes for IEEE 300-bus system

eigenvalue of $\mathbf{Q''^{T}Q''}$. We repeat this process 1,000 times and record the average of the results.

Figures 16 and 17 show the empirical CDF curves of eigenvalues of $\mathbf{Q''^{T}Q''}$ for the IEEE 118-bus and 300-bus systems when $\tau_a = 0.1\tau$, respectively. We can observe the same tendency as shown in Figures 16 and 17. For the IEEE 118-bus system, the maximum eigenvalue of $\mathbf{Q''^{T}Q''}$ is quite small (e.g., no state variable results in an eigenvalue that is larger than 0.01). However, for the IEEE 300-bus system, a few state variables can achieve large eigenvalues that are near 1.

6. CONCLUSION AND FUTURE WORK

In this paper, we identified a previously unknown vulnerability in the current techniques aimed at detecting and identifying bad measurements for state estimation in electric power grids. We investigated the implications of this vulnerability through presenting and analyzing a new class of attacks, called false data injection attacks, against state estimation

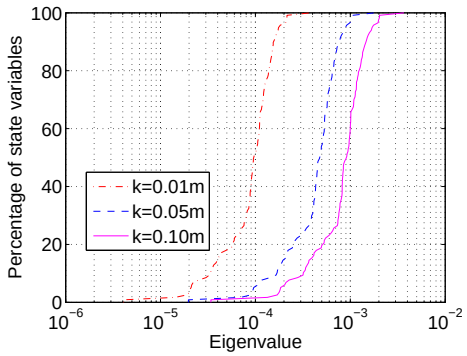

Fig. 16. Eigenvalue of $\mathbf{Q''^{T}Q''}$ for IEEE 118-bus system

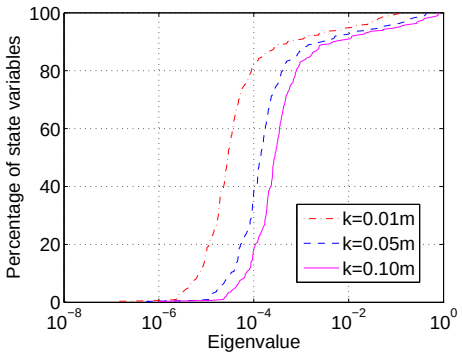

Fig. 17. Eigenvalue of $\mathbf{Q''^{T}Q''}$ for IEEE 300-bus system

in electric power systems. Under the assumption that the attacker can access the current power system configuration information and manipulate the measurements of meters at physically protected locations, such attacks can introduce arbitrary errors into certain state variables without being detected by existing techniques. We considered two attack scenarios, where the attacker is either constrained to some specific meters, or limited in the resources required to compromise meters. We showed that the attacker can systematically and efficiently construct attack vectors in both scenarios, which can not only change the results of state estimation, but also modify the results in a predicted way. We also extended false data injection attacks to generalized false data injection attacks, and used both theoretical analysis and simulation to show that an attacker can gain more impact than false data injection attacks by launching generalized false data injection attacks. Despite the theoretical capability of these attacks, we also pointed out that such attacks are strictly limited by real-world constraints, and do not pose immediate threats to our power grids.

In our future work, we would like to extend our results to state estimation using AC power flow models. Moreover, we would also like to investigate the possibility of adapting network anomaly detection techniques to defend against false data injection attacks.

ACKNOWLEDGMENTS

We would like to thank Dr. Carl Gunter and Dr. Ernst Scholtz for their insightful discussions and comments. We would also like to thank Associate Editor Dr. Marianne Winslett for her extended efforts to facilitate the communication with the anonymous reviewers and the revision of this paper. We are very grateful to the anonymous reviewers from both the power systems and the security communities; their insightful comments have only made this paper better. This work is supported by the National Science Foundation (NSF) under grants CNS-0831302 and CAREER-0447761. The contents of this paper do not necessarily reflect the position or the policies of the U.S. Government.

REFERENCES

ABUR, A. AND EXPÓSITO, A. G. 2004. *Power System State Estimation: Theory and Implementation*. Marcel Dekker Publishers.

AMALDI, E. AND KANN, V. 1998. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science 209*, 1-2 (December), 237–260.

ASADA, E. N., GARCIA, A. V., AND ROMERO, R. 2005. Identifying multiple interacting bad data in power system state estimation. In *IEEE Power Engineering Society General Meeting*. 571–577.

BLUMENSATH, T. AND DAVIES, M. 2008. Gradient pursuits. *IEEE Transactions on Signal Processing 56*, 6 (June), 2370–2382.

BOBBA, R. B., ROGERS, K. M., WANG, Q., AND KHURANA, H. 2010. Detecting false data injection attacks on DC state estimation. In *Proceedings of the First Workshop on Secure Control Systems (SCS '10)*.

BROCKWELL, P. J. AND DAVIS, R. A. 1991. *Time Series: Theory and Methods*, 2nd ed. Springer.

CHEN, J. AND ABUR, A. 2005. Improved bad data processing via strategic placement of PMUs. In *IEEE Power Engineering Society General Meeting*. 509–513.

CHEN, J. AND ABUR, A. 2006. Placement of PMUs to enable bad data detection in state estimation. *IEEE Transactions on Power Systems 21*, 4 (November), 1608–1615.

CHEN, S. S. 1995. *PhD thesis: Basis Pursuit*. Department of Statistics, Stanford University.

CHRISTIE, R. D. 1999. Power systems test case archive. `http://www.ee.washington.edu/research/pstca/`.

DÁN, G. AND SANDBERG, H. 2010. Stealth attacks and protection schemes for state estimators in power systems. In *IEEE 2010 SmartGridComm*. to appear.

GARCIA, A., MONTICELLI, A., AND ABREU, P. 1979. Fast decoupled state estimation and bad data processing. *IEEE Transactions on Power Apparatus and Systems 98*, 5 (September), 1645–1652.

GAREY, M. R. AND JOHNSON, D. S. 1979. *Computer and Intractability: a guide to the theory of NP-Completeness*. W.H.Freeman and Company.

GASTONI, S., GRANELLI, G. P., AND MONTAGNA, M. 2003. Multiple bad data processing by genetic algorithms. In *IEEE Power Tech Conference*. 1–6.

GEORGIEV, P. AND CICHOKI, A. 2004. Sparse component analysis of overcomplete mixtures by improved basis pursuit method. In *the 2004 IEEE International Symposium on Circuits and Systems (ISCAS 2004)*. 5:37–40.

GOLUB, G. H. AND VAN LOAN, C. F. 1989. *Matrix Computation*, 2nd ed. The John Hopkins University.

HANDSCHIN, E., SCHWEPPE, F. C., KOHLAS, J., AND FIECHTER, A. 1975. Bad data analysis for power system state estimation. *IEEE Transactions on Power Apparatus and Systems 94*, 2 (April), 329–337.

HERTEM, D. V., VERBOOMEN, J., PURCHALA, K., BELMANS, R., AND KLING, W. L. 2006. Usefulness of DC power flow for active power flow analysis with flow controlling devices. In *The 8th IEE International Conference on AC and DC Power Transmission*. 58–62.

HUGGINS, P. S. AND ZUCKER, S. W. 2007. Greedy basis pursuit. *IEEE Transactions on Signal Processing 55*, 7 (July), 3760–3772.

KOSUT, O., JIA, L., THOMAS, R. J., AND TONG, L. 2010a. Limiting false data attacks on power system state estimation. In *Proceedings of Conference on Information Sciences and Systems*.

KOSUT, O., JIA, L., THOMAS, R. J., AND TONG, L. 2010b. Malicious data attacks on smart grid state estimation: Attack strategies and countermeasures. In *IEEE 2010 SmartGridComm*. to appear.

KOSUT, O., JIA, L., THOMAS, R. J., AND TONG, L. 2010c. On malicious data attacks on power system state estimation. In *proceedings of 45th International Universities' Power Engineering Conference (UPEC 2010)*.

LI, M., ZHAO, Q., AND LUH, P. B. 2008. DC power flow in systems with dynamic topology. In *Power and Energy Society General Meeting–Conversion and Delivery of Electrical Energy in the 21st Century*. 1–8.

LIN, J. AND PAN, H. 2007. A static state estimation approach including bad data detection and identification in power systems. In *IEEE Power Engineering Society General Meeting*. 1–7.

LIU, Y., NING, P., AND REITER, M. 2009. False data injection attacks against state estimation in electric power grids. In *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS '09)*. 21–32.

LOVISOLO, L., DA SILVA, E. A. B., RODRIGUES, M. A. M., AND DINIZ, P. S. R. 2005. Efficient coherent adaptive representations of monitored electric signals in power systems using damped sinusoids. *IEEE Transactions on Signal Processing 53*, 10 (October), 3831–3846.

MEYER, C. 2001. *Matrix Analysis and Applied Linear Algebra*. SIAM.

MILI, L., CUTSEM, T. V., AND PAVELLA, M. R. 1985. Bad data identification methods in power system state estimation, a comparative study. *IEEE Transactions on Power Apparatus and Systems 103*, 11 (November), 3037–3049.

MILI, L., CUTSEM, T. V., AND RIBBENS-PAVELLA, M. 1984. Hypothesis testing identification: A new method for bad data analysis in power system state estimation. *103*, 11 (November), 3239–3252.

MONTICELLI, A. 1999. *State Estimation in Electric Power Systems, A Generalized Approach*. Kluwer Academic Publishers.

MONTICELLI, A. AND GARCIA, A. 1983. Reliable bad data processing for real-time state estimation. *IEEE Transactions on Power Apparatus and Systems 102*, 5 (May), 1126–1139.

MONTICELLI, A., WU, F. F., AND MULTIPLE, M. Y. 1986. Bad data identification for state estimation by combinatorial optimization. *IEEE Transactions on Power Delivery 1*, 3 (July), 361–369.

NATARAJAN, B. K. 1995. Sparse approximate solutions to linear system. *SIAM Journal on Computing 24*, 2 (April), 227–234.

NATIONAL SECURITY TELECOMMUNICATIONS ADVISORY COMMITTEE (NSTAC) – INFORMATION ASSURANCE TASK FORCE (IATF). Electric power risk assessment.

PATI, Y. C., REZAIIFAR, R., AND KRISHNAPRASAD, P. S. 1993. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *the 27th Asilomar Conference on Signals, Systems and Computers*.

QUINTANA, V. H., SIMOES-COSTA, A., AND MIER, M. 1982. Bad data detection and identification techniques using estimation orthogonal methods. *IEEE Transactions on Power Apparatus and Systems 101*, 9 (September), 3356–3364.

SANDBERG, H., TEIXEIRA, A., AND JOHANSSON, K. H. 2010. On security indices for state estimators in power networks. In *Proceedings of the First Workshop on Secure Control Systems (SCS '10)*.

SCHWEPPE, F. C., WILDES, J., AND ROM, D. B. 1970. Power system static state estimation. parts 1, 2, 3. *IEEE Transactions on Power Apparatus and Systems 89*, 1 (January), 120–135.

U.S.-CANADA POWER SYSTEM OUTAGE TASK FORCE. 2004. *Final report on the August 14, 2003 blackout in the United States and Canada*. `https://reports.energy.gov/B-F-Web-Part1.pdf`.

WOOD, A. AND WOLLENBERG, B. 1996. *Power generation, operation, and control*, 2nd ed. John Wiley and Sons.

WU, F. F. AND LIU, W.-H. 1989. Detection of topology errors by state estimation. *IEEE Transaction on Power Systems 4*, 1, 176–183.

XIANG, N. AND WANG, S. 1981. Estimation and identification of multiple bad data in power system state estimation. In *the 7th Power Systems Computation Conference, PSCC*. 1061–1065.

XIANG, N., WANG, S., AND YU, E. 1982. A new approach for detection and identification of multiple bad data in power system state estimation. *IEEE Transactions on Power Apparatus and Systems 101*, 2 (Febuary), 454–462.

XIANG, N., WANG, S., AND YU, E. 1983. An application of estimation-identification approach of multiple bad data in power system state estimation. In *IEEE Power Engineering Society Summber Meeting*.

ZHAO, L. AND ABUR, A. 2005. Multi area state estimation using synchronized phasor measurements. *IEEE Transactions on Power Systems 20*, 2 (May), 611–617.

ZHU, J. AND ABUR, A. 2007. Bad data identification when using phasor measurements. In *IEEE Power Tech Conference*. 1676–1681.

ZIMMERMAN, R. D. AND MURILLO-SÁNCHEZ, C. E. 2007. *MATPOWER, A MATLAB Power System Simulation Package*. `http://www.pserc.cornell.edu/matpower/manual.pdf`.

A. GENERATING AN ATTACK VECTOR VIA ELEMENTARY OPERATIONS IN RANDOM FALSE DATA INJECTION ATTACKS IN SCENARIO I

In this appendix, we show how the attacker can construct an attack vector using elementary column operations when $k > m - n$. Let $\bar{\mathcal{I}}_{meter} = \{j | 1 \leq j \leq m, j \notin \mathcal{I}_{meter}\}$, and $\mathbf{H} = (\mathbf{h}_1, ..., \mathbf{h}_n)$, where $\mathbf{h}_i = (h_{1,i}, ..., h_{m,i})^T$ for $1 \leq i \leq n$. For a random $j \in \bar{\mathcal{I}}_{meter}$ (i.e., the meter not under the attacker's control), the attacker first scans $\mathbf{H}$ to look for a column vector whose j-th element is not zero. If the attacker can find such a vector,

the attacker swaps it with $\mathbf{h}_1$. Then, the attacker can construct an $m \times (n-1)$ matrix $\mathbf{H^1} = (\mathbf{h^1}_1, ..., \mathbf{h^1}_{n-1})$ by performing column transformations on $\mathbf{H}$ (to zero out the j-th element in all column vectors):

$$\mathbf{h^1}_i = \begin{cases} \mathbf{h}_1 - \frac{h_{j,1}}{h_{j,i+1}}\mathbf{h}_{i+1}, \text{ if } h_{j,i+1} \neq 0, 1 \leq i \leq n-1 \\ \mathbf{h}_{i+1}, \qquad \text{if } h_{j,i+1} = 0, 1 \leq i \leq n-1 \end{cases} \tag{16}$$

If the j-th element is zero for all the column vectors of $\mathbf{H}$, then $\mathbf{h^1}_i = \mathbf{h}_i$ for $1 \leq i \leq n-1$. As a result, the j-th row of $\mathbf{H^1}$ are all zeros. The attacker repeats this process to the reduced matrix $\mathbf{H^1}$ and the reduced matrices thereafter using a different element in $\bar{\mathcal{I}}_{meter}$, until all elements in $\bar{\mathcal{I}}_{meter}$ are exhausted. Finally, the attacker can get a matrix having at least one column vector, since $m - k \leq n - 1$. The column vectors of the final matrix are linear combinations of the column vectors of $\mathbf{H}$, and the $m-k$ rows with index $j \in \bar{\mathcal{I}}_{meter}$ of this matrix consist of all 0's. Any column vector can be used as an attack vector.

Note that the above approach looks similar to traditional Gaussian elimination [Meyer 2001], since they both use elementary matrix operations to eliminate non-zero elements of a matrix. The difference between them is that Gaussian elimination reduces a given matrix to either triangular or echelon form, whereas our approach does not convert the original matrix into triangular or echelon form. We generate a reduced matrix after each iteration instead, attempting to find a linear combination of column vectors of the original matrix.

B. SOLVING $\mathbf{a}'$ FOR RANDOM GENERALIZED ATTACKS IN SCENARIO I

The attacker needs to solve $\mathbf{a}'$ from $\mathbf{B}'\mathbf{a}' = \mathbf{B}\mathbf{t}$ to get the attack vector $\mathbf{a}$. As discussed earlier, if the rank of $\mathbf{B}'$ is not the same as that of the augmented matrix $(\mathbf{B}'|\mathbf{B}\mathbf{t})$, then $\mathbf{B}'\mathbf{a}' = \mathbf{B}\mathbf{t}$ is not consistent and thus has no solution for $\mathbf{a}'$.

To ensure equal ranks, the attacker can manipulate $\mathbf{t}$ such that $\mathbf{B}\mathbf{t}$ is a linear combination of columns of the matrix $\mathbf{B}'$, and thus the rank of $\mathbf{B}'$ is the same as that of the augmented matrix $(\mathbf{B}'|\mathbf{B}\mathbf{t})$. A simple way is to let $\mathbf{t} = (0, ..., 0, t_{i_1}, 0, ..., 0, t_{i_2}, 0, ..., 0, t_{i_k}, 0, ..., 0)^T$, where $(t_{i_1}, ..., t_{i_k})^T$ can be any vector whose 2-Norm is less than τ_a. By choosing a proper $\mathbf{t}$, the attacker can solve $\mathbf{a}'$ from the equation and $\mathbf{a}' = \mathbf{B}'^{-}(\mathbf{B}\mathbf{t}) + (\mathbf{I} - \mathbf{B}'^{-}\mathbf{B}')\mathbf{d}$, where $\mathbf{B}'^{-}$ is the Matrix 1-inverse of $\mathbf{B}'$ and $\mathbf{d}$ is any non-zero vector of length k. The attacker can construct an attack vector $\mathbf{a}$ from any $\mathbf{a}' \neq \mathbf{0}$. Note that if $\mathbf{B}'$ is a full rank matrix, $\mathbf{B}'^{-}\mathbf{B} = \mathbf{I}$ and $\mathbf{a}' = \mathbf{B}'^{-}(\mathbf{B}\mathbf{t})$.

C. SOLVING $\mathbf{a}'$ FOR TARGETED GENERALIZED ATTACKS IN SCENARIO I

The attacker needs to solve $\mathbf{a}'$ from equation $\mathbf{B}'_\mathbf{s}\mathbf{a}' = \mathbf{B}_\mathbf{s}\mathbf{t} + \mathbf{B}_\mathbf{s}\mathbf{b}$ to get the attack vector $\mathbf{a}$. The equation $\mathbf{B}'_\mathbf{s}\mathbf{a}' = \mathbf{B}_\mathbf{s}(\mathbf{t} + \mathbf{b})$ has no solution if the rank of $\mathbf{B}'_s$ is not the same as that of the augmented matrix $(\mathbf{B}'_s|\mathbf{B}_\mathbf{s}(\mathbf{t} + \mathbf{b}))$. This means that the attacker needs to make $\mathbf{B}'_s$ and $(\mathbf{B}'_s|\mathbf{B}_\mathbf{s}(\mathbf{t} + \mathbf{b}))$ have the same rank in order to find an attack vector.

Note that $\mathbf{b}$ is a fixed vector. If the rank of $\mathbf{B}'_s$ is equal to that of the augment matrix $(\mathbf{B}'_s|\mathbf{B}_\mathbf{s}\mathbf{b})$, the attacker can set $\mathbf{t} = (0, ..., 0, t_{i_1}, 0, ..., 0, t_{i_2}, 0, ..., 0, t_{i_k}, 0, ..., 0)^T$, where $(t_{i_1}, ..., t_{i_k})^T$ can be any vector whose 2-Norm is less than τ_a. Consequently, $\mathbf{B}_\mathbf{s}\mathbf{t}$ is a linear combination of columns of the matrix $\mathbf{B}'_s$ and rank$((\mathbf{B}'_s|\mathbf{B}_\mathbf{s}\mathbf{t} + \mathbf{B}_\mathbf{s}\mathbf{b}))$ = rank$((\mathbf{B}'_s|\mathbf{B}_\mathbf{s}\mathbf{b}))$ = rank $(\mathbf{B}'_s)$. Thus, the attack vector $\mathbf{a}$ can be obtained by computing $\mathbf{a}'$ from equation $\mathbf{B}'_\mathbf{s}\mathbf{a}' = \mathbf{B}_\mathbf{s}(\mathbf{t} + \mathbf{b})$.

If the rank of $\mathbf{B}'_s$ is not the same as that of $(\mathbf{B}'_s|\mathbf{B}_\mathbf{s}\mathbf{b})$, the attack vector does not necessarily exist. The attacker can treat $\mathbf{t} + \mathbf{b}$ as a whole to determine the existence of an attack

vector. Specifically, let $\mathbf{t}+\mathbf{b}=\mathbf{w}=(w_1,...,w_m)$ and $\mathbf{b}=(b_1,...,b_m)^T$. The attacker first sets $\mathbf{w}=(0,...,0,\ w_{i_1},0,...,0,w_{i_2},0,...,0,w_{i_k},0,...,0)^T$, where $w_{i_1},...,w_{i_k}$ can be any nonzero value. As a result, $\mathbf{B_s w}$ is a linear combination of columns of the matrix $\mathbf{B}'_s$ and the rank of $\mathbf{B}'_s$ equals that of $(\mathbf{B}'_s|\mathbf{B_s w})$. Then the attacker determines whether the attack vector exists or not by checking the 2-Norm of $\mathbf{t}$. Note that $\mathbf{t}=\mathbf{w}-\mathbf{b}=(-b_1,...,w_{i_1}-b_{i_1},...,w_{i_2}-b_{i_2},...,w_{i_k}-b_{i_k},...,-b_m)^T$. Thus,

$$\|\mathbf{t}\|=\sqrt{\sum_{i\neq i_1,...,i_k} b_i^2+\sum_{i=i_1,...,i_k}(w_i-b_i)^2}\geq\sqrt{\sum_{i\neq i_1,...,i_k} b_i^2}. \tag{17}$$

Therefore, if $\sqrt{\sum_{i\neq i_1,...,i_k} b_i^2}<=\tau_a$, the attacker can choose proper value for w_{i_1},...,w_{i_k} to make $\|\mathbf{t}\|$ less than or equal to τ_a (e.g., set $w_{i_1}=b_{i_1}$,...,$w_{i_k}=b_{i_k}$). Hence, the attack vector can be constructed by solving $\mathbf{a}'$ from equation $\mathbf{B'_s a'}=\mathbf{B_s}(\mathbf{t}+\mathbf{b})$. However, if $\sqrt{\sum_{i\neq i_1,...,i_k} b_i^2}>\tau_a$, $\|\mathbf{t}\|$ is always larger than τ_a and the attacker cannot generate the attack vector.