Tomorrow's Smart Power Grid: Crafting Security Measures Using State Estimation

Adam N. Magaña
Computer Science
Youngstown State University
anmagana@student.ysu.edu

Graduate Mentor: **Saurabh Amin**
Research Supervisor: **Dr. Alvaro Cárdenas and Dr. Annarita Giani**
Faculty Mentor: **Prof. S. Shankar Sastry**

July 31, 2009

Summer Undergraduate Program in
Engineering Research at Berkeley (SUPERB) 2009

Department of Electrical Engineering and Computer Sciences
College of Engineering
University of California, Berkeley

Tomorrow's Smart Power Grid: Crafting Security Measures Using State Estimation

Adam N. Magaña

Abstract

The United States power grid is an advanced, robust piece of modern engineering. With the present revolution to make the power grid a 'smarter' system, using two-way data transmission between power usage meters and a control center (Supervisory Control and Data Aquisition - SCADA), comes the threat of falsified data being injected by attackers. Using a technique known as 'False Data Injection,' potential hackers could modify the output of a sensor, or multiple sensors to manipulate the automated reactions of the grid's control center. In this paper we use real-time measurements in order to provide the best estimate of the current operating state and use them to eliminate the abnormal values that may be injected by attackers. While successful attacks are possible, we find that attackers are required to not only infiltrate network communication but are also required to know specific and extensive details regarding network topology and parameters; a pressing challenge that calls for high coordination and timing.

1 Introduction

The United States power grid is a fundamental piece of the nation's critical infrastructure. It is a delicate and expansive system of power generation and distribution to people all over America. Its uses are limitless: business or personal. It is under these well-understood facts that we value and protect our electrical power grid while sparing no expense. We do not harbor any room for failure when it comes to our power grid, which is why the calling of a 'smart' power grid is timely and imperative.

Everyday we draw nearer to the inevitable 'smart grid'. The smart grid is a conceived idea that centers around the power grid's ability to monitor and repair itself in real-time; avoiding power loss and outages before they occur. For these innovations to be possible, we must look at how such a system would acquire real-time data on itself. For a power grid as large as the United States', this would indeed call for millions of measurement sensors at every fraction of the grid. Mainly, as a solution, these sensors can be appended to the already in-place power meters that are found on every electricity-receiving location. After giving these sensors communication to the larger, more computationally-burdened side of the system it is then possible to compute the state of the system. The process known as State Estimation, the use of real-time measurements in order to provide the best estimate of the current operating state, is the main computation that will give the status of the power grid. Using this estimation along with standard deviation techniques and standard distribution patterns, it is possible for the grid to recognize and omit bad data from the estimates. This bad data can be classified as error or noise in the sensor data (standard hardware error) or it can be classified as an attack. These attacks, which would utilize a

hacker's ability to interrupt communication from a sensor to the system and input their own values, is called False Data Injection. With False Data Injection, malicious data may be sent to the system in order to receive some kind of desired effect. These effects have a range of outcomes that include the hacker's self-benefit (such as a cheaper electrical bill) as well as maniacal destruction (cyber terrorism). In either case it is imperative that the grid is able to perceive such attacks and handle them accordingly.

In order to better understand how False Data Injection attacks and State Estimation work, it is important to understand the network topology and architecture. As referenced in Fig. 1, the network's information is collected using millions of measurement devices. These devices communicate directly with network hubs known as Remote Terminal Units (RTU). It is important to think of these RTU's as data aggregators, collecting data from many sensors and sending it along as a single package. These large packets of information are sent to and stored in a system known as SCADA. SCADA, the Supervisory Control and Data Acquisition system, is the median point in the entire system. The SCADA system sends all of the data to the Energy Management System. Specifically, the Energy Management

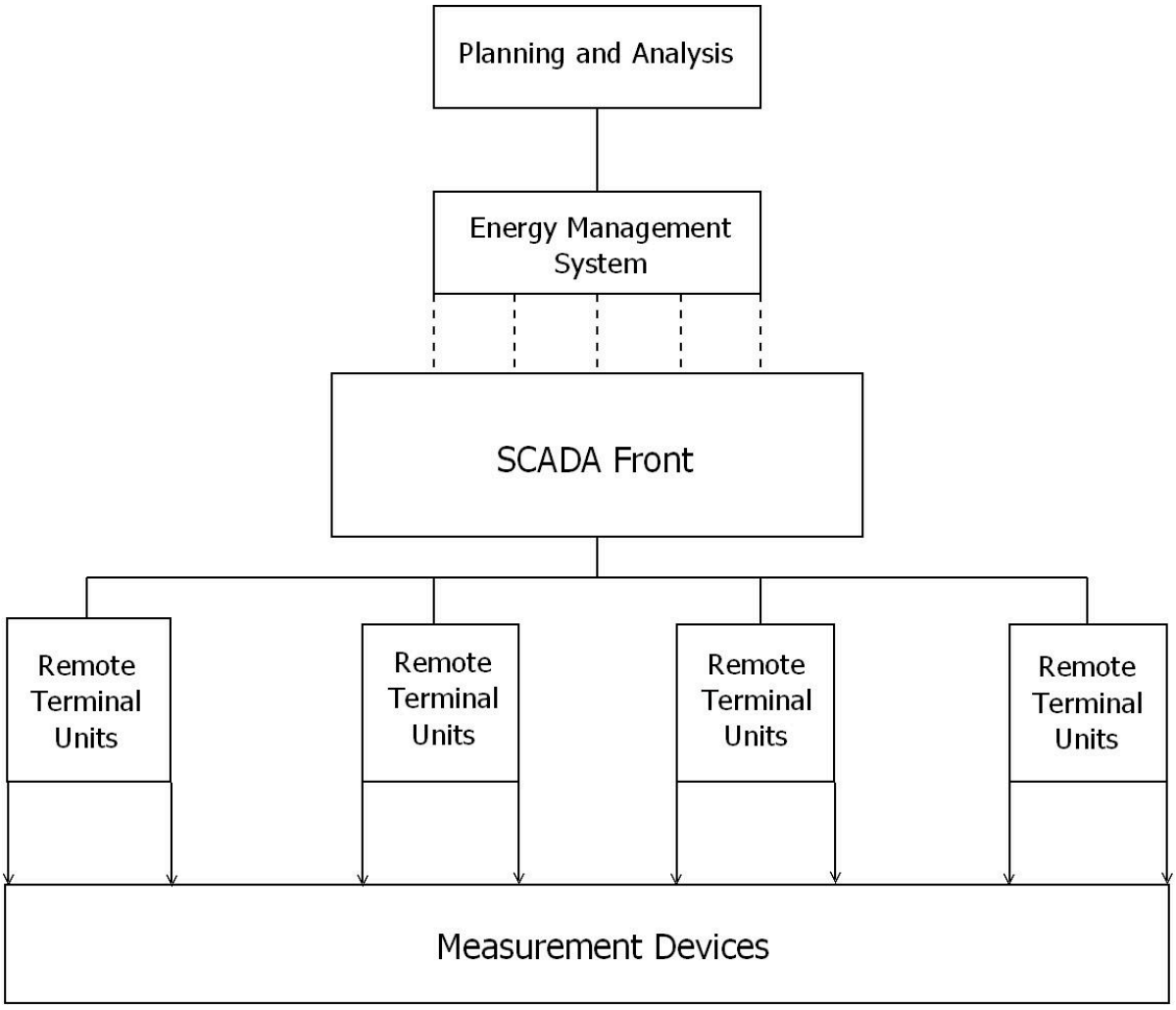

Figure 1. The SCADA network topology and hierarchy model.

System is where the actual State Estimation is calculated using the information that was collected by the measurement devices and passed up through the system. The Energy Management System and SCADA are the two parts of the automated system that react to bad data: identifying and omitting. For analysis on the human administration end, the final reports about the state of the system are transmitted to the Planning and Analysis tier.

Given a full understanding of the system architecture, we can begin to infer how certain False Data Injection attacks would be carried out. It is possible for an attacker to target two major transmission lines easiest: the transmission lines between the RTU's and the SCADA front and the lines between the SCADA front and the Energy Management System. For instance, in a scenario where a hacker has interrupted traffic from an RTU to the SCADA front, we can interject that this interruption is being used for data injection. The hacker has a few attack options. He/she can perform an additive, subtractive, multiplicative, or divisional attack. These attacks involve taking the actual measurement and modifiying it. The algorithm for data manipulation is quite simple and requires little work from the attacking standpoint once the security has been breached. Given a semi-intelligent attacker and some strong propagation/robustness of their attack the system may take a serious blow if it is not able to detect this intrusion.

This can be quite a shocking thought that brings very serious consequences to mind. However, there is still ample time to prepare for such inevitable tests of security. Since the smart technology is not in place yet, it is required that we exclusively run simulations of power flows, attacks, and detection algorithms. For the sake of our experiments we choose to run 14-bus power flow simulations using Matpower, a Matlab plugin suite authored by Ray D. Zimmerman of Cornell University. The software has a State Estimation algorithm with detection methods already in place for us to use; hence, minor modification is all that is necessary. Given this modified software we can then begin to study more modern, dynamic detection algorithms. However, in order to fully understand how counter-measures must be set up we must first investigate the specifics of State Estimation.

2 Specifics of State Estimation

As mentioned earlier we must take into account that the basis for a smart grid is the millions of measurement sensors being placed on every portion of the power grid. If we say that the value of these sensor measurements at any given time in the state can be found in vector y and are enumerated by value j, then we can say that the mean of these values can be found using the following formulation:

$$\bar{y} = 1/n \sum_{j=1}^{n} y_j$$

Using sensor redundancy we can determine whether an attack has occured:

$$\left\{ \begin{array}{c} x \in \Re^n \\ y \in \Re^m \end{array} \right\} : m > n \leftarrow \text{sensor redundancy}$$

Given all of these sensor measurements and their standard distribution, we can now formulate the system model:

$$\begin{bmatrix} z_1 \\ z_2 \\ ... \\ z_m \end{bmatrix} = \begin{bmatrix} h_1(x) \\ h_2(x) \\ ... \\ h_m(x) \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ ... \\ e_m \end{bmatrix}$$

This formulation $h_m(x)$ is a non-linear function of the unknown state x plus the measurement error e_i. z_i is the sum of these two values and the final, observed value is the state based on real meter measurements. Our goal is to obtain an estimate $\hat{x}$ from the received observation z.

Here we define the measurement residual:

$$r_j := z_j - h_j(x)$$

Solving State Estimation as an optimization problem (minimizing cost) where o_j is the standard deviation of the measurement error:

$$\begin{aligned} \hat{x} &= min: \left[1/2 \sum_{j=1}^{m} \frac{(r_j)^2}{(o_j)^2} \right] \\ &= min: \left[1/2 \sum_{j=1}^{m} \frac{[z_j - h_j(x)]^2}{(o_j)^2} \right] \end{aligned}$$

In order to minimize the cost function of state estimation we must calculate

$$\frac{dJ(x)}{dx_1} = \frac{d}{2x_1} * \frac{\sum_{j=1}^{m} [z_j - h_j(x)]^2}{(o_j)^2} = 0$$

solving for x where

$$-H^T(x)R^{-1}(z - h(x))$$

$-H^T(x)$ can be calculated given the Jacobian matrix $H(x)$:

$$H(x) = \begin{bmatrix} \frac{dh_1(x)}{dx_1} & \frac{dh_1(x)}{dx_2} \\ \frac{dh_2(x)}{dx_1} & \frac{dh_2(x)}{dx_2} \end{bmatrix}$$

Next is the update step where k is the index of the iteration, x^k is the state estimate at iteration k, and $G(x^k)$ is a matrix given by $H^T(x^k)R^{-1}H(x^k)$.

$$\begin{aligned} x^{k+1} &= x^k - [G(x^k)]^{-1} g(x^k) \\ (x^{k+1} - x^k) &= -[G(x^k)^{-1}] g(x^k) \\ G(x^k)(\Delta x^{k+1}) &= -G(x^k) G(x^k)^{-1} g(x^k) \\ G(x^k)(\Delta x^{k+1}) &= -g(x^k) \end{aligned}$$

Gauss Newton Steps:

1. $k = 0$
2. Initialize x^k
3. Calculate $G(x^k) = H^T(x^k)R^{-1}H(x^k)$
4. Solve for Δx^{k+1} using step 1.
5. Test for convergence $\left|\Delta x^{k+1}\right| < \in$

3 Potential Threat

The total number of possible attack combinations can be calculated using m and k, where m is the total number of measurement nodes and k is the number of attacks.

$$\binom{m}{k} = \frac{m!}{k!(m-k)!}$$

The attack that we simulated throughout our experimenting was multiplicative. In this attack, the hacker chose a random sensor, j, and a max perturbation value, a. Given the original sensor data, z_j, they would treat a_j as a multiplicative scalar value. For example:

$$\tilde{z}_j = z_j * a_j$$

In this formulation, the value of $\tilde{z}_j$ is the corrupted measurement given by the multiplication of the original data and the random perturbation value, with a max constraint, made by the attacker.

Hypothesis testing, using the largest residual test where $\hat{z} = h(\hat{x})$:

$$z = \begin{bmatrix} z_1 \\ z_2 \\ \dots \\ z_m \end{bmatrix}, \hat{z} = \begin{bmatrix} \hat{z}_1 \\ \hat{z}_2 \\ \dots \\ \hat{z}_m \end{bmatrix}, \hat{z} - z = \begin{bmatrix} \hat{z}_1 - z_1 \\ \hat{z}_2 - z_2 \\ \dots \\ \hat{z}_m - z_m \end{bmatrix}$$

With our calculations, it is easy to see that while corruption is very easily calculated, it requires that the attacker know the topology and parameters of the power system. This is easily plausible with a small power flow, such as a 14-bus system, however, when applying this logic to the entire power grid it seems very unlikely. Any attacks done on such a large, complicated system would undoubtedly be randomized and primitive at best.

4 Simulating the Power Flows

In order to properly simulate a 14-bus power flow system and State Estimator we employed the use of MATPOWER, a Matlab plugin authored by Ray D. Zimmerman of Cornell University [1]. Default configuration settings for this simulation package were not oriented for a 14-bus power flow but rather for a 30-bus system. Modifications were also necessary when building our H and z matrices. Reducing these matrices to only account for voltage angles was the heaviest change made.

Making the listed modifications to the state estimation algorithm, we were able to adapt this simulation to fit our needs. Furthermore, a data aggregation and parent script was custom written with the ability to run the simulation software as many times as desired. This ability to quickly execute the simulation software, even upwards of 10,000 times, allowed for statistical data to be collected on how certain, randomized scaled attacks were processed by the detection algorithm. Given that the state estimator took into account random measurement error it was imperative that we test each perturbation value

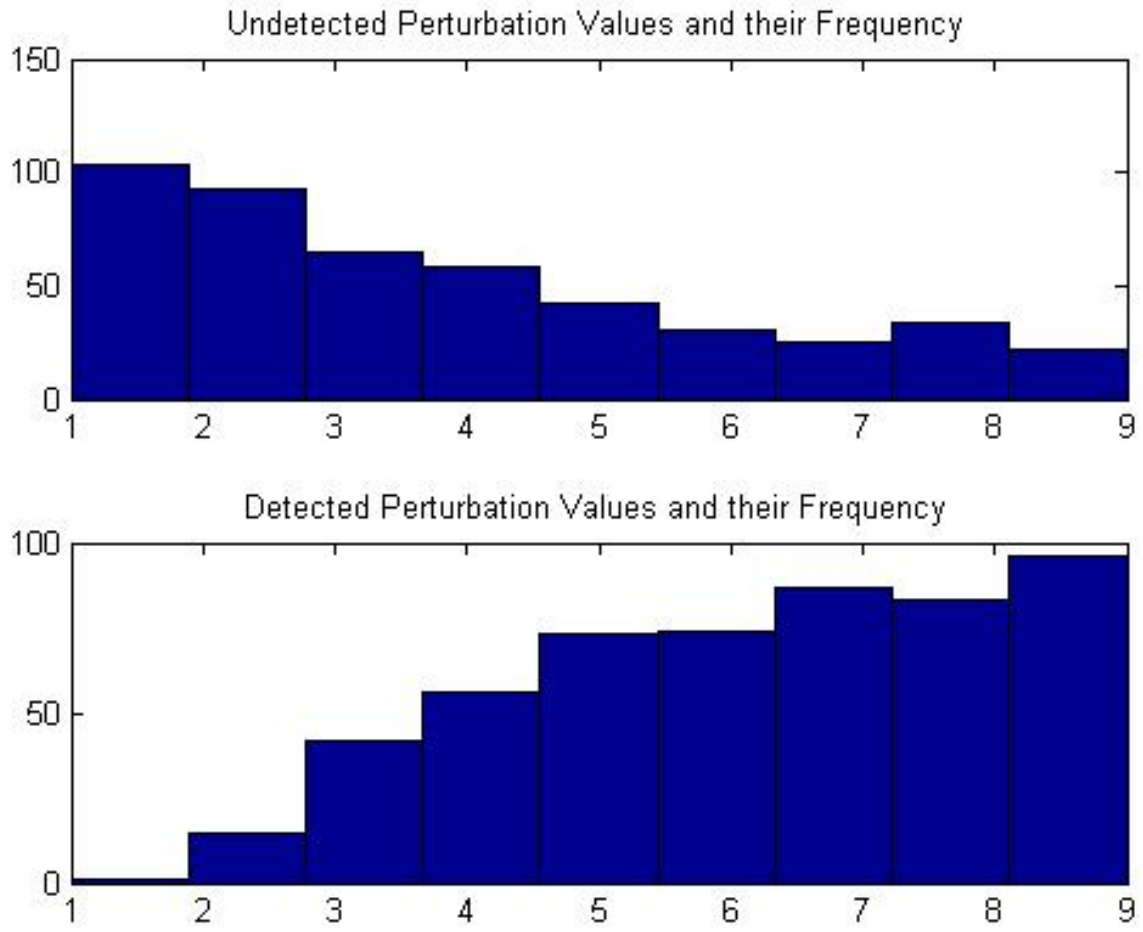

Figure 2. A histogram plot showing the statistical output from 1000 power flow simulations.

thoroughly. Small variations in measurement error can drastically change the detection algorithm's ability to weed out the bad data.

5 Interpreting the Data

After doing 1,000 state estimation simulations, we were able to collect enough statistical data to make some general assumptions regarding the relationship between attack success and the attack value. Referencing Fig. 2, we can see that when performing scaled attacks with a relatively high perturbation value, the attack success goes down drastically. The chi-squared detection algorithm implemented in the State Estimation package is much more likely to detect a value that has been effected by a greater value than that of a smaller value. This can be a very pressing issue for attackers. These findings support the idea that the attacker would be forced to attack with a small value in order to be successful most often. This requirement fends off massive corruption values and, consequently, implies that more than a single measurement be attacked at a time to be successful in stressing the power grid's functional integrity.

6 Conclusions

After reviewing our findings, we can propose that further study be done in certain sects of the research. However, it is important to point out before more work is done that our research was modeled after Yao Liu, Peng Ning, and Michael K. Reiter's 'False Data Injection Attacks against State Estimation in Electric Power Grids' paper: accepted for CCS '09 [2]. Using their analysis and theory we were able to make significant simulation progress in a short amount of time. A fundamental draw-back to this, though, is that the paper makes an impetuous assumption about the knowledge of the hacker. It makes a very strong assumption that the intruder knows the configuration of the Hessian matrix: the configuration and parameters of the entire grid. This stipulation massively changes the magnitude and direction of the research. Even if we humor the authors by saying that an attacker may gain access to the setup of the H matrix, we could easily counter that point by implying that randomized protocols would be in place in the SCADA system that ambiguate the H matrix; stopping the hacker from knowing future configurations. Larger, more plausible security threats should be in focus regarding power grid security. Our results also reinforce this conclusion. Given our high detection percentage for formidable perturbation values (Fig. 2) it is safe to say that the state estimation side of the system is unlikely to experience effective false data injection attacks.

Setting these arguments aside, we can still hope that further improvements will be made in bad data detection. Changing the bad data threshold for the detection algorithms would be a very intriguing next-step. Discovering new, more efficient variations on the bad data threshold may show how a significant portion of undetected attacks can be brought into detection. Furthermore, testing out a larger variety of attacks, not just computationally different, but strategically different, should prove to be a very insightful path. For instance, can the current detection algorithms detect when multiple attacks are carried out at once using different corruption techniques? This would be one of the many questions that could be asked next of the research.

Acknowledgements

I would first like to thank the University of California, Berkeley as well as the Team for Research in Ubiquitous Secure Technology (TRUST) for giving me the opportunity to study this ground-breaking research. Secondly, I would like to thank my advisor and faculty mentor from Youngstown State University, Dr. Graciela Perera, for pushing and encouraging me to do hard, rewarding work. Lastly, I would like to extend my most gracious and sincerest thanks to my graduate mentor at UC Berkeley, Saurabh Amin, and my research supervisors, Dr. Alvaro Cárdenas and Dr. Annarita Giani, for coaching me through this research. Without out all of the listed benefactors this research would not have been possible.

References

[1] MATPOWER (IDE for simulating power flows and state estimation). Technical report, http://www.pserc.cornell.edu/ray/.

[2] Y. Liu, P. Ning, and M. K. Reiter. False data injection attacks against state estimation in electric power grids. Technical report, North Carolina State University, University of North Carolina - Chapel Hill, November 2009.